Biblioteka Informatyki
Szkół Wyższych

Information Systems Architecture and Technology

System Analysis Approach to the Design, Control and Decision Support

Library of Informatics of University Level Schools

Series of editions under the auspices
of the Ministry of Science and Higher Education

The ISAT series is devoted to the publication of original research books in the areas of contemporary computer and management sciences. Its aim is to show research progress and efficiently disseminate current results in these fields in a commonly edited printed form. The topical scope of ISAT spans the wide spectrum of informatics and management systems problems from fundamental theoretical topics to the fresh and new coming issues and applications introducing future research and development challenges.

The Library is a sequel to the series of books including Multidisciplinary Digital Systems, Techniques and Methods of Distributed Data Processing, as well as Problems of Designing, Implementation and Exploitation of Data Bases from 1986 to 1990.

Wrocław University of Technology

Information Systems Architecture and Technology

System Analysis Approach to the Design, Control and Decision Support

Editors
Jerzy Świątek
Leszek Borzemski
Adam Grzech
Zofia Wilimowska

Wrocław 2011

Publication partly supported by

Faculty of Computer Science and Management
Wrocław University of Technology

Project editor

Arkadiusz GÓRSKI

The book has been printed in the camera ready form

OFICYNA WYDAWNICZA POLITECHNIKI WROCŁAWSKIEJ
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław
http://www.oficwyd.pwr.wroc.pl;
e-mail: oficwyd@pwr.wroc.pl
zamawianie.ksiazek@pwr.wroc.pl

ISBN 978-83-7493-632-3

CONTENTS

PART 1. DATA MINING SUPPORT AND EXPERT SYSTEMS

PART 2. MATHEMATICAL MODEL AND ITS APPLICATIONS IN DECISION SUPPORT IN TECHNICAL AND NON-TECHNICAL PLANTS

PART 3. IMAGE PROCESSING AND PATTERN RECOGNITION

PART 4. SOFT COMPUTING AND ITS APPLICATONS

PART 5. COMPLEX OF OPERATION SYSTEMS CONTROL

INTRODUCTION

Systems approach to the computer aided design, control and decision requires model of the investigated process. All decisions and project are are based on the knowledge about the object under investigation. That's why models are so important in systems research. Investigation of plants of deferent nature (technical, economical, biomedical or computational) gives us many notifications about observed processes. Collected knowledge, about investigated process gives us the model of observed reality. The mathematical model gives us more precise description. Usually the relation between values characterizing process is given. The static properties are given by functional relations, equations and inequalities. The dynamics of investigated plants are given by differential equations and inequalities – for continuous or difference one for – discrete processes. The set of true sentences gives also mathematical description of the investigated process.

System analysis gives us the proper tools to create further decision about investigated plant based on the collected knowledge, and consequently based on the elaborated model. Base on the model the optimization, control and management task may be formulated. Base on the knowledge about the process the diagnosis may be proposed.

The above mentioned applications of different type tasks we can recognize in selected and revived chapters which have been divided into the following groups:

PART 1. DATAN MINING SUPPORT AND EKSPERT SYSTEMS
PART 2. MATHEMATICAL MODEL AND ITS APPLICATIONS IN DECISION SUPPORT IN TECHNICAL AND NON–TECHNICAL PLANTS
PART 3. IMAGE PROCESSING AND PATTERN RECOGNITION
PART 4. SOFT COMPUTING AND ITS APPLICATONS
PART 5. COMPLEX OF OPERATION SYSTEMS CONTROL

The book provides an interesting representation of research in the area of system analysis in decision aided problems in proposed groups.

PART 1. DATA MINING SUPPORT AND EXPERT SYSTEMS

In the Chapter 1 one of the methods derived from graph theory, which is named Minimal Spanning Tree (MST) is presented. This method is using correlations to calculate distance between pairs of values to show there connection. To present MST effectiveness currencies of several countries were investigates This approach allows to see not only simple currency dependencies, but can also show economic and political connections.

The Chapter 2 presents an application of expert system techniques to the development of a system that can assist advertisers of consumer products in the process of advertising design. The main goal was to create a complete system which produces newspaper advertisement, with fully customized elements such as texts and pictures. An expert system is employed at the stage of designing the layout, selecting colors, images and other graphical forms by analyzing the advertising category, target consumers, advertised product or service. Its knowledge base is established on the analysis of current advertising campaigns observed in the newspapers.

The Chapter 3 describes certain concept of expert system, called STALCOM, which supports management process in the commercial firm. STALCOM system has been designed to support choosing the right grade of steel according to the possibility of using this grade of steel in further processing. Fundamental establishments of systems construction, structure and functional description are presented. Particular attention was paid to knowledge base record taking into account many decisive ways for solving the problem introduced to the system.

The Chapter 4 presents the current state of the on-going project aimed at development of the modular integrated environment MMSD for Managing and Mining Structured Data. The unified modeling scheme for a range of structures extracted from original resources and developed modules for their transformation into prescribed graph types (directed/undirected, labeled/unlabeled) is proposed. For practical reasons, the modules accomplishing visualization and edition of graph models are also provided. The open architecture enables an easy enhancement of system functionalities as well as its potential to cooperate with number of existing structured data repositories.

The Chapter 5 proposes an architecture of the experimental computer program that is adaptable and well suited to keep up with the foreseeable pace of any research work and that makes it possible to maintain the sustainability of the development of the program and to minimize the amount of work that is needed for the program to adjust to subsequent new research projects.

A new strategy for streamlining the process of clustering data with regard to the effect of similarity between the clustered objects in the process of creating groups is presented in the Chapter 6. It provides for the use of local fine-tuning of the parametric error functions, but taking into account the actual degree of order of objects on the grid. Obtained by the author results show an improvement in the quality of clustering obtained for the original, which is an algorithm ATTA.

PART 2. MATHEMATICAL MODEL AND ITS APPLICATIONS IN DECISION SUPPORT IN TECHNICAL AND NON-TECHNICAL PLANTS

The Chapter 7 presents a new approach to the quantifying changes in large social network illustrated by the data from an organizational social network of the Wroclaw University of Technology. We analyse The process of emerging and disappearing of the links for different periods and time windows was analysed. It allows to discover new dynamic patterns and carrying on their structural analysis. The observations are used to propose a novel link prediction algorithm, which shows good performance, especially for sparse networks analysed in short time-scales

A conceptual platform based on microscopic, multiagent approach, dedicated for crowd behavior modeling in different situations is presented in the Chapter 8. Each agent possesses a set of individual properties. An agent can communicate with each other's. Agents in the model are assigned to different groups: the agents who are in the same group cooperate together, while different groups of agents can compete with another groups. If set of simple or more complicated rules of behavior of individual entities are used, a very interesting view on crowd motion can be observed.

In the Chapter 9 the approach based on neutral martingale method and Monte Carlo simulations in order to analyse some model of catastrophe bond is discussed. The example of such bond applying stochastic model of risk-free spot interest rate under assumption of independence between catastrophe occurrence and behaviour of financial market was analysed. Then the numerical simulations to analyse the behaviour of the obtained pricing formula are used.

The aim of Chapter 10 is to provide an overview of an intelligent information system dealing with simulation of tornado damages inflicted in forests. The system is potentially useful for forests managers in delivering information about optimization of newly grown tree stands against tornado damages in regions endangered with severe wind gusts. The system consists of a combined Rankine vortex used for tornado simulation and of HWIND tree damage model used for assessing tornado impact on forests. The HWIND model has been modified to be used for sudden wind blows conflicted by tornadoes in contrast to constant wind speeds for which it was designed

A higher number of new sources of electric energy, in particular wind farms, causes new problems related to forecasting the energy production level. Wind power stations are units, which do not provide a stable level of energy supply. Therefore there exists a need to develop forecasting models, which make it possible to forecast the work of such units in a reliable way. In the Chapter 11 the theory of fractal analysis application to the processes related to the operation of wind power stations is proposed. Research results and suggestions concerning their further possible applications have been given.

In the Chapter 12 the basic problem connected with modeling of waste network are presented. Methods of modeling of basic sewage variables and calculation algorithms are described .The problems concern the gravitation branched off network divided by nodes into sectors. The nodes are the points of connection of several network segments or branches or the points of changing of network parameters as well as the location of sewage inflow to the network. The presented algorithm for hydraulic calculations concerns the housekeeping or combined sewage net.

PART 3. IMAGE PROCESSING AND PATTERN RECOGNITION

In the chapter 13 on of the oldest and most widely used algorithms for contour shape representation – Fourier Descriptors (FD) – is applied to the problem of General Shape Analysis (in short, GSA) and experimentally evaluated. This problem is similar to both the recognition and retrieval of shapes. From the first of those two tasks the searching for similar objects is taken. From the second one – the presentation of more than one resultant object to the user. Moreover, the analysed shape can be similar to one of the template classes and does not have to belong to any of them. It means that the most general information about a shape is concluded by means of the GSA.

Fluency and safety of traffic flow is one of the most important contemporary problems as the number of cars is constantly increasing. In the Chapter 14 two main issues are considered: accelerating immediately after the previous car moved on when the traffic lights change and keeping a safe distance from the previous car in regular traffic e.g. on the motorway. The Driver's Assistance System was proposed to support drivers by displaying messages about braking and accelerating. The system was based on the video image from a camera attached to a car and the License Plate localization algorithm, also introduced in this chapter.

The Chapter 15 describes the developed dynamic recognition algorithm and proposed original method of feature extraction in the pedestrian detection task. The main idea is to follow the observation, that vertical position of people's head oscillates during gait. Distinctive rhythm of human gait is the feature extracted from video sequence and used to distinguish between pedestrians and other objects. Algorithm and implementation details are given. Analysis of incorrect classifications is provided. It is suggested to apply the system to measure traffic at road junctions with traffic lights.

In the Chapter 16 the results in analysing performance of classifiers for missing data in thoracic surgery (TS) risk modelling are reported for experiments made in Statistical Data Miner environment. Brief comments on current results in applications of quantitative modelling for TS data are presented, as well as a comprehensive description of updated and extended TS data bases from Wroclaw TS Centre. Application of Statistical Data Miner Recipes (DMR) is presented with special attention paid to initial data analysis, cleansing, feature reduction and missing values problems.

The Chapter 17 presents the idea of a biometric security system based on the way of using a mouse. The mouse actions parameters are examples of behavioral features, which are not stable at time in contrast to physiological characteristics. Though their advantage is the possibility of applying without using any special hardware and without interrupting users. A set of features, which may be extracted from data coming from a mouse, has been described. Training data from thirteen users has been collected and classifiers have been trained and tested using two machine learning methodologies: support vector machines and decision trees.

In the Chapter 18 system for support athletes training is considered. Architecture of proposed system and suitable application is presented in details. Comprehensive analysis of functional and non-functional properties of system is given. Implementation details and results of experimentation are discussed as well.

PART 4. SOFT COMPUTING AND ITS APPLICATONS

In the Chapter 19 some basic discrete optimization problems such as the shortest path, minimum spanning tree and minimum assignment are discussed. The uncertainty is modeled by specifying a finite scenario set and the min-max criterion is adopted to choose a solution. All the considered problems are known to be NP-hard. Particularly in this chapter the efficiency of the mixed integer programming formulation for these problems is tested. Two popular solvers, namely *glpsol* and *cplex* are used for the tests.

A very important issue in various fields of engineering problem is recognition the actual state of the test object and the related dynamic processes. Recognition of this condition is based on information collected and additional information. However, the accumulated information is burdened with some errors. Therefore, the Chapter 20 presents an attempt to build the neural filter, allowing the identification of a discrete signal, so that it can be assigned by the neural network to a class of signals. The problem of filtering of such signals is not new, but the methods that were used for this purpose so far are mainly based on a simplified and thus quite distant from the reality of the mathematical theory of signals, while the proposed approach involves the use of neural networks, which have the ability to adapt and self-organization during the workout.

The Chapter 21 is a proposal related to partial parallel realization of retrieving phase of Multilayer Perceptron algorithm. The method is based on pipelined systolic arrays – SIMD architecture. The discussion is realized based on operations which create the following steps of the algorithm. The efficiency of proposed approach is discussed based on implementation quality criteria for systolic arrays. The results of discussion show that it is possible to create the architecture which provides massive parallelism and reprogram ability.

In the Chapter 22 a simulated annealing (SA) based solution to the block packing into rectilinear outline problem is considered. The goal is to place a set of blocks into a given rectilinear outline such that no two blocks overlap and a given cost function is minimized. Contrary to the previous approaches, primarily the set of random moves performed by basic SA algorithm is limited to those, which lead to feasible solutions. In each iteration of proposed SA algorithm an incremental SP neighborhood evaluation algorithm is applied, to create the set of all feasible solutions, letting SA to perform a random selection from this set.

The Traveling Salesmen Problem (TSP) is one of the most successful application areas of the Ant Colony Optimization techniques (ACO). The ACO algorithm is controlled by a number of parameters. The selection of proper value parameters for the ACO is done mainly in an experimental manner. In the Chapter 23 the attempts to automate the process parameter optimization using an algorithm which is inspired by a combination if Evolutionally Programming (EP) and Simulated Annealing (SA) is presented. The chapter presents the original versions of the basic algorithms and their proposed modifications.

In the Chapter 24 the school bus route optimization problem is analysed. It is a crucial social issue that concerns faster and more comfortable transport of students to their schools. Moreover, the route optimization allows to decrease the ticket price, i.e., to maximize the profit of the provider. Since the problem belongs to hard optimization problems, thus, four meta-heuristic algorithms are adapted: Tabu Search, Simulated Annealing, Genetic Algorithm, Complete Overview, and invented by the authors algorithm called Constructor, and additionally Bellman-Ford algorithm used as a helper.

PART 5. COMPLEX OF OPERATION SYSTEMS CONTROL

The Chapter 25 deal with the cyclic scheduling problem usually observed in the FMS producing multi-type parts where the AGVS plays a role of a material handling system. Finding the conditions guaranteeing the AGVs deadlock-free and collision-free movement policy is the aim of this work. The AGVs co-sharing the common parts of the transportation route while executing repetitive processes, i.e. being assigned to AGVs passing along machines in a cyclic way, can be modelled in terms of Cyclic Concurrent Process Systems (CCPS). In this chapter a novel approach for schedulability analysis employing the declarative modelling is proposed.

The Chapter 26 presents the results of research on the problem of time-optimal programs scheduling and primary memory pages allocation in computer system consisting of a group of parallel processors for special type of programs processing time function. A multiprocessing computer system consisting of *m* parallel processors, common primary memory and external memory is considered. The primary memory contains *N* pages of identical capacity. This system can execute *n* independent programs. The problem belongs to the class of *NP*-complete problems then an heuristic algorithm to minimize schedule length criterion, which employs some problem properties is proposed

In the Chapter 27 it is shown that the makespan minimization problem in the two-processor flow shop environment becomes strongly NP-hard if the processing time of a job is described by an arbitrary function dependent on its position in a sequence (models learning or aging). Moreover, the fast NEH algorithm with complexity lower than its standard version is proposed. Efficiency of the proposed method was numerically analyzed for the problems with the aging effect.

In many real-life cases the efficiency of a processor can change due to its learning. Therefore, new and more precise models have been proposed that take into consideration the varying nature of processors. On the other hand, the existing solution algorithms are inefficient for these new models. It implies that new methods have to be proposed to manage the real-life problems. Since the scheduling problems with learning are new in the scheduling theory, thus, the significant number of these problems have no efficient solution algorithms. Therefore, in the Chapter 28 an exact methods such as dynamic programming is provided. Furthermore, the fast approximation algorithm NEH that have lower complexity than its standard version, is proposed.

Improvement or a degradation of a system can be modeled by job processing times that are described by non-increasing (improvement) or non-decreasing (degradation) functions dependent on the number of previously processed jobs. The Chapter 29 focus on scheduling problems with such varying processing times and the following minimization objectives: the maximum completion time and the maximum lateness. Although the scheduling problems with varying processing times have attracted particular attention of a research society, the computational complexity of some problems has not been determined.

The significance and hardness of the floorplanning in the VLSI physical design caused that much effort have been taken to address this bottleneck. The floorplanning problem can be expressed as a classic rectangle packing problem: given a set of rectangular modules of arbitrary size the goal is to place them on a two-dimensional space without overlapping, subject to minimize the area of a minimum bounding rectangle. In the Chapter 30, a novel approach based on neural network is proposed. A basis the GIT algorithm that iteratively inserts blocks into initially empty solution is used.

An algorithmic attempt to perform a frame sp-dissection in complex-outline floorplanning problem: given a rectilinear packing outline, enclose it with minimum bounding rectangle and cut the resulting figure into rectangles such that there exists a sequence-pair which codes such obtained placement. Known algorithm starts with an optimal solution of the classical dissection problem, which, however, often cannot be represented with sequence-pair representation. Therefore, in the second stage, an iterative algorithm that, by performing additional cuts of some rectangles, produces a sp-dissection is applied. In the Chapter 31 a counter example is given, showing that known two-phase approach cannot lead to an optimal sp-dissection – a single-phase sp-dissection algorithm is needed. The several properties of optimal sp-dissection problem solution are presented which can lead to an efficient algorithm solving the SP-dissection problem in single-phase.

Wrocław, September 2011

Jerzy Świątek

PART 1

DATA MINING SUPPORT AND EXPERT SYSTEMS

MST, currencies,
correlation coefficient

Anna KIŁYK*, Zofia WILIMOWSKA*

MINIMAL SPANNING TREE OF THE FOREIGN EXCHANGE MARKET (FOREX)

This work presents one of the methods derived from graph theory, which is named Minimal Spanning Tree (MST). This method is using correlations to calculate distance between pairs of values to show there connection. To present MST effectiveness currencies of several countries were investigates. Data used in this work can be divided into two groups: expressed by PLN or expressed by EUR. This approach allows to see not only simple currency dependencies, but can also show economic and political connections.

In addition, quoted method can be also very helpful for the companies operating on the global markets. The information, which one can obtain from analysis of those method, can help with important decision-making processes [5]. One should remember that MST method doesn't give a definite answer, but merely "indicates" possible solutions by describing dependencies between markets.

1. INTRODUCTION

Normal operation of the company is often associated with making financial decision burdened by high risk (import/export of goods and services where there are foreign currency account, tangible investments, equity investment, etc.). In order to overcome those problems they are looking into a variety of methods that might minimize the actual risk by any means available.

The basis of a "good" investment is analyzing the financial market. This analysis of the relationships between pair of shares, currency or other values can help investors with making investment decisions. Literature studies on this subject clearly show how important and difficult is the presented problem, and how many different methods can be created [2, 3, 5, 7, 8, 9, 10, 12]. One of them, which is using the relationships be-

* Wrocław University of Technology, Institute of Industrial Engineering and Management, Smoluchowskiego 25, 50-372 Wrocław.

tween analyzed values (time series) as its most important variable, is the Minimal Spanning Tree (MST) graphs – one of the network programming methods. This method is simple but sometimes results obtained from it, can show an invisible dependency between analyzed values.

The main problem, which has been described in this work, is to identify an easy method that can be very helpful with making financial decisions. This problem is very widespread and affects all companies operating on the basis of barter (one currency exchanged to a different currency of another country to buy/sell a product, etc.). The questions which are raised when tackling these problems (when, which and how much foreign currency to buy) can make the decision more difficult. That's why information from MST analysis, which shows the correlations between currencies can make decision easier (as knowledge about the correlation will provide additional information about possible behavior of the analyzed object).

2. HOW TO BUILD GRAPH

To use MST one should create a time series of a financial value like currency price, their returns or exchange ratios expressed in terms of the base currency. It should be noted that the final graph is dependent on the base currency – thus a different base currency may yield a different tree (showing the problem from a different perspective). This time series is represented by a set of data points $i = 1,2,...,t$, where t is the time span of the analysis (this also represents the number of data points). Since it is ordered series, one can interpret is as a "currency vector" $R_\alpha(i)$ for value α. Analyzing the foreign exchange market one has N time series representing the quoted values (for example: in Poland one has $N = 36$). When the data is ready, the next step is to calculate the distance between two currency vectors. To do this one uses equation (eq.1):

$$d_{\alpha\beta}(i) \equiv d\left(R_\alpha(i), R_\beta(i)\right) = \left\| R_\alpha(i) - R_\beta(i) \right\| \tag{1}$$

where:
i – time steps $(i = 1,2,...,t)$
$d_{\alpha\beta}(i)$ – distance between currency α and currency β at time i,
$R_\alpha(i)$ – currency vector of α at time i.

And to show distance between two currencies one can calculate the sum of all time series distance (eq. 2):

$$d_{\alpha\beta} = \sum_{i=1}^{t} \left\| R_\alpha(i) - R_\beta(i) \right\| \tag{2}$$

This equation (eq. 2) requires some simplification to calculate it in a straightforward way. To do that one should explicitly write the vector's norm in Euclidean space i.e.: raise the elements to the second power, and take its square root and thus calculate the squared difference of the vectors (eq. 3).

$$d_{\alpha\beta} = \left(\sum_i \left(R_\alpha(i) - R_\beta(i) \right)^2 \right)^{\frac{1}{2}} \tag{3}$$

$$d_{\alpha\beta} = \left(\sum_i \left[\left\| R_\alpha(i) \right\|^2 + \left\| R_\beta(i) \right\|^2 - 2\mathrm{R}_\alpha(i) R_\beta(i) \right] \right)^{\frac{1}{2}} \tag{4}$$

Because vectors R are normalized (eq. 5) (the two first elements of the sum yield a 1) formula for the correlation coefficient of two values $C_{\alpha\beta}$ is given by (eq. 6):

$$\sum_i \left\| R \right\|^2 = 1 \tag{5}$$

$$\sum_i R_\alpha R_\beta = C_{\alpha\beta} \tag{6}$$

Now, one can obtain the final formula for the distance between vectors (eq. 7), which is also the “distance” between analysis currencies. As it can be seen the distance between items is dependent only on the correlation coefficient which means it can be calculated in an easily and quickly.

$$d_{\alpha\beta} = \sqrt{2\left(1 - C_{\alpha\beta}\right)} \tag{7}$$

When talking about distance vectors, one should mention three of their properties. Firstly, the distance between a vector and itself is equal zero (eq. 8):

$$d_{\alpha\alpha} = 0 \tag{8}$$

The next property states that order in which the vectors are taken doesn't affect their distance (eq. 9). This fact was visible in (eq. 1), where the distance was defined in terms of a modulus:

$$d_{\alpha\beta} = d_{\beta\alpha} \tag{9}$$

The last property, which is shown in the next equation (eq. 10), describes relations between three different vectors. This is a representation of a very general property of all metrics - subadditivity. In simple 2D space (which is the case here) this the well known triangle inequality:

$$d_{\alpha\beta} \leq d_{\alpha\gamma} + d_{\gamma\beta} \tag{10}$$

Now, with all the needed information, one can create a MST graph for all investigated currencies. First, one should create a distance matrix for all currencies. This matrix is composed of distances between every pair of currencies. Next one should look for a pair of vectors which have the smallest distance between them. This effectively means looking for the smallest value of $d_{\alpha\beta}$. Moreover it should be noted that the smallest value of distance means that correlation coefficient is the biggest of all pairs. This in turn, means that the elements that are closest to each other are more strongly interconnected (highly correlated). With an ordered list of them, one connects consecutive values to create a graph.

3. DESCRIPTION OF THE PROPOSED APPROACH

Using this method one can investigate the foreign currency exchange market (FOREX) which is the biggest financial market associating currencies from all over the word. Because of the world time difference FOREX is operating from 11 p.m. (Sunday) to 10 p.m. (Friday) – Central European Time. The quoted currencies are presented in relation to base currency: currency of the country in which they are noted [14].

The main goal of this research is to present the usefulness of the MST method. To achieve this, two sets of calculations were performed: one with selection of PLN (Polski Zloty) as the base currency and the second with EUR (Euro) as the base. The first group of Polish based currencies was composed of 34 items i.e.: ADU, EUR, HUF, CZK, ZAR, RUB, LTL, LVL, HKD, XDR, USD, SEK, NOK, JYP, GBP, DKK, EKK, CHF, CAD, RON, BGN, TRY, THB, SGD, PHP, MZD, CNY, IDR, HRK, ISK, KRW, MYR, MXN, BRL.

The second group of Euro base currency consisted of 29 items: ADU, PLN, HUF, CZK, ZAR, RUB, HKD, XDR, USD, SEK, NOK, JYP, GBP, DKK, CAD, RON, BGN, TRY, SGD, PHP, NZD, CNY, IDR, ISK, KRW, MYR, MXN, BRL, HRK (the time span of the analysis of both sets is the same: from 05.01.1999 to 03.12.2010). Thanks to having more than one graph one can look for relations which are constant for both base currencies and thus represent a strong and lasting bond.

4. ANALYSIS OF RESULTS

Before graphs of the trees will be shown, one should present the basic rules of how to create a graph. At first, one should declare a variable – in this work we select this variable in the form of currency returns. Next, one should know that introducing links between already connected nodes of the Minimal Spanning Tree is forbidden – loops cannot exist. That's why the last rule states that the number of currency (N) must be larger than the number of links (A) in the graph (eq. 11):

$$A < N \tag{11}$$

Now when all rules presented, using equation 7, one can calculate the distances between all currencies. Furthermore one can be see that the highest correlations are producing the smallest distances.

Figure 1 show the MST for a group of currencies, with the base set in PLN. There are two currencies, which are often combined with other items (they create a cluster). The most connected currency is the SGD (Singapore dollar), which has 7 currencies joined to it. It's also worth noting that the main currencies of Asia are grouped together (China – CNY, Singapore – SGD, Thailand – THB, Japan – JYP, Hong Kong – HKD, Korea S. – KRW, Philippines – PHP, Indonesia – IDR, and Malaysia – MYR).

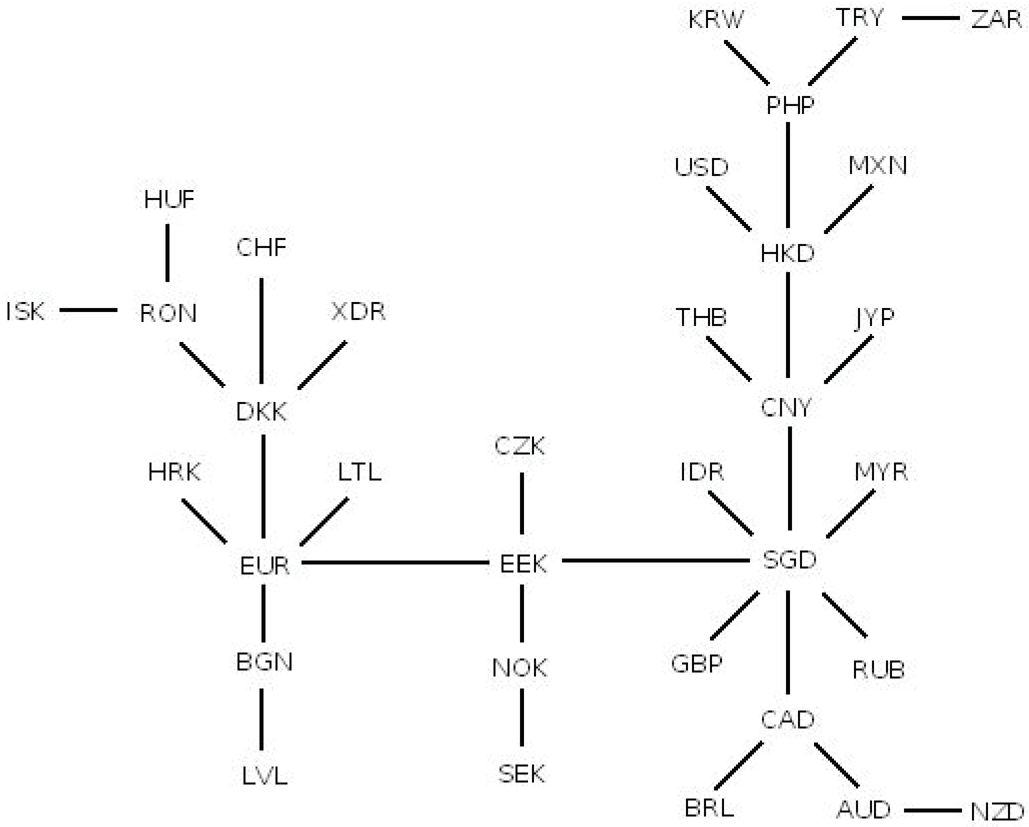


Fig. 1. MST for Polish based currencies

It also should be emphasized that the relatively small distances between Asian currency vectors (which indicates high rates of correlation for individual prices). This is shown more accurately in Table 1.

Table 1. The distance between pair of Asian currencies

Pair of currencies	Distance between
CNY – HKD	0,08
THB – CNY	0,11
SGD – MYR	0,13
JYP – CNY	0,19
HDK – PHP	0,25
CNY – SGD	0,26
IDR – SGD	0,38
PHP – KRW	0,54

Thanks to the Table 1 one can easily notice that there are only two Asian currencies that have a fairly large distance (Indonesia (IDR), Korea S. (KRW)) to other Asian markets, however these distances are still much smaller then the distances between those two and the other world currencies.

It's also na interesting fact, that the shortest distance vectors have the European currencies linked to the EURO (Bulgarian Lev (BGN), Lithuanian Litas (LTL), Danish Krone (DKK), Estonian Kroon (EEK), Croatian Kuns (HRK)) – Table 2.

Table 2. The distance between pair of currencies with EURO

Pair of currencies	Distance between them
EUR – BGN	0,0002
EUR – LTL	0,0003
EUR – EEK	0,0017
EUR – DKK	0,0071
EUR – HRK	0,0137

With such small distances between those currencies, the correlation coefficient will be assumed to be close to value of 1. It's mean that those five currencies will be behave like the Euro trading.

Similar situation one can be observed in case of European markets. Most of the currencies are holding with Euro or with currencies which are related to it. But there are also four exceptions from this "rules" in form: GBP (United Kingdom), RUB (Russia), NZD (New Zealand) and TRY (Turkey). The first three currencies are grouped with SGD (Singapore), with is the one of the most developed country in the region. Turkey on the other hand is strongly related to neither SGD (Singapore) nor EUR.

Looking at American currencies, one can see a quite different situation. The currencies of the United States of America (USA) and Canada (CAD) are "attached" to the one of the biggest Asia currencies like HKD (Croatia) and SGD (Singapore). The position of USA is interesting because previous research show that this currency should be in a central node of the graph [4]. This situation can be caused by the continuing instability of the world market.

The second graph is presenting Euro based currencies and it's more scattered then the first MST. On this figure one can see four major currencies: XDR (Special Drawing Rights[1]), PHP (Philippines), MXN (Mexico), and RON (Romania).

As previously, Asian currencies are keeping together. The components of the other markets are scattered, but one can see some links important between them. It is more

[1] XDR is the international unit of account, existing only in the form of journal entries, which is the currency basket (XDR = 44% USD + 34% EUR + 11% JYP + 11% GBP).

visible in the European currencies, for example: Norway (NOK), Sweden (SEK), Denmark (DKK) and: Turkey (TRY), Romania (ROM), Bulgaria (BGN), Russian Federation (RUB) and Island (ISK).

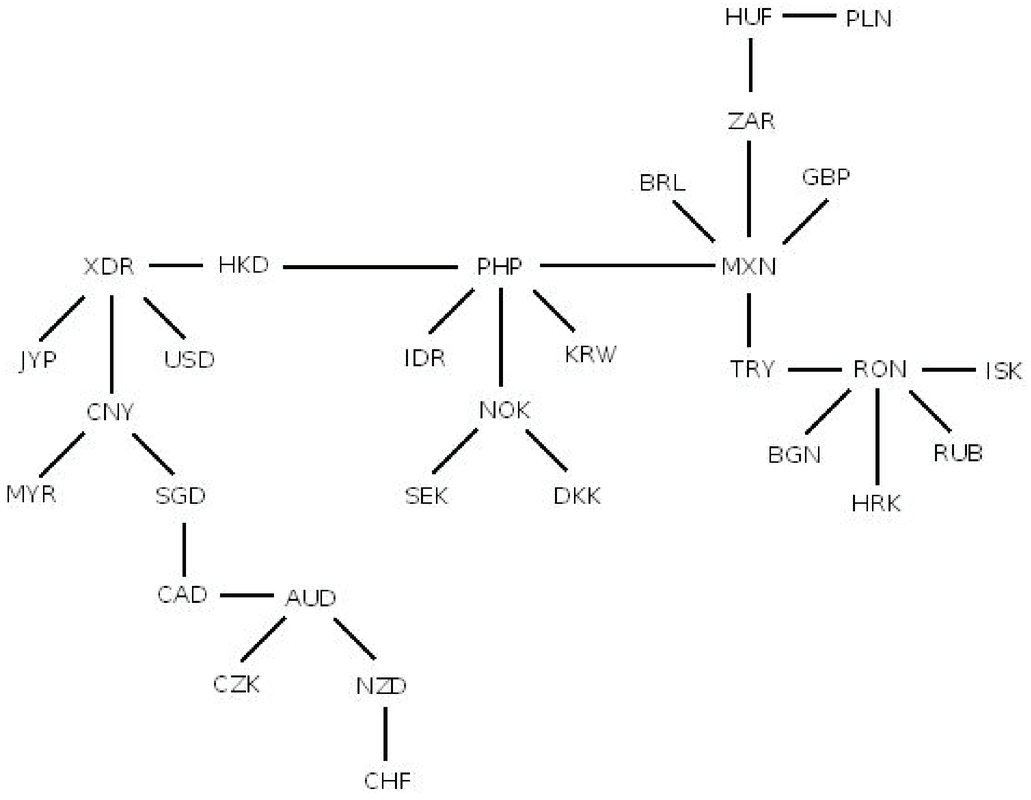


Fig. 2. MST for Euro based currencies

Looking at this graph (Graph 2) one can see that the currency of USA (USD), as in a graph 1, is associated with a group of Asian markets by a link with XDR (Special Drawing Rights). The same situation can be observed for Canada (CAD) and Mexico (MXN). It is also interesting that Polish currency (PLN) is connected with South Africa (ZAR) – through Hungary (HUF).

As in the previous case (graph 1), the distances between Asian currencies are small and therefore highly correlated. But in this case (graph 2) one can not distinguish a single currency closely connected with others (like in the case of graph 1 – Euro), only several different currencies pair which are close to each other.

5. CONCLUSIONS

This method clearly show that the Asian market is strongly interconnected. This situation can be a result of good economic policy and growing demands of emerging economies.

Also, in both cases, currency of the USA (USD) is a marginal element, which may be caused by the financial crash. In addition, most European currencies are trying to keep in a group, which may be seen as a mutual dependence in the spheres of economics, politics, etc.

Graph theory (from which MST is derived) has been used in studies of economic phenomena for several years. The MST is a model, which can be use in all fields of economics: currencies, shares [1, 6, 13], gemstones prices. It can be used for analysis and prediction of asset behavior.

REFERENCES

[1] CHMIEL A.M., SIENKIEWICZ J., SUCHECKI K., HOŁYS J.A, *Networks of companies and branches In Poland.* Physica A, Vol. 383 (2007), pp. 134–138.

[2] COELHO R., GILMORE C.G., LUCEY B., RICHMOND P., HUTZLER S., *The evolution of interdependence in world equity markets – Evidence from minimum spanning trees.* Physica A, Vol. 376 (2007), pp. 455–466.

[3] GILMORE C.G., LUCEY B.M., BOSCiA M, *An ever-closer union? Examining the evaluation of linkages of European equity markets via minimal spanning tree.* Physica A, Vol. 387 (2008), Issue 25, pp. 6319–6329.

[4] GÓRSKI A. Z., DROŻDŻ S., KWAPIEŃ J., OŚWIĘCIMKA P., *Minimal Spanning Tree Graphs and Power Like Scaling in FOREX Networks.* In: Acta Physica Polonica A, Vol. 114 (2008), No. 3, pp. 531–538.

[5] HASHIMOTO Y., ITO T., OHNISHI T., TAKAYASU M., TAKAYASU H., WATANABE T., *Random walk or a run. Market microstructure analysis of foreign exchange rate movements based on conditional probability.* Quantitative Finance (2010), pp. 78–85.

[6] HEIMO T., SARAMAKI J., ONNELA J.P., KASKI K., *Spectral and Network methods in the analysis of correlation matrices of stock returns.* Physica A, Vol. 383 (2007), pp. 147–151.

[7] IINO T., IKEDA Y., IYETOMI H., KAMEHAMA K., OHNISHI T., TAKAYASU H., TAKAYASU M., *Structure analyses of a large-scale transaction network through visualization based on molecular dynamics.* Journal of Physics: Conference Series 221 (2010).

[8] JANG W., LEE J., CHANG W., *Currency crises and the evolution of foreign exchange market. Evidence from minimum spanning tree.* Physica A, Vol. 390 (2011), Issue 4, pp. 707–718.

[9] MCDONALD M., SULEMAN O., WILLIAMS S., HOWISON S., JOHNSON N.F., *Detecting a currency's dominance or dependence using foreign exchange network trees.* Physical Review E, Vol. 72 (2005), No. 4, pp. 1–11.

[10] MIZUNO T., TAKAYASU H., TAKAYASU M., *Correlation networks among currencies.* Physica A, Vol. 364 (2006), pp. 336–342.

[11] TABAK B.M., SERRA T.R., CAJUEIRO D.O., *Topological properties of stock market networks: The case of Brazil.* Physica A, Vol. 389 (2010), No. 16, pp. 3240–3249.

[12] TSENG J.J., LI S.P., *Asset returns and volatility clustering in financial time series.* Physica A, Vol. 390 (2011), pp. 1300–1314.

[13] UENO H., MIZUNO T., TAKAYASU M., *Analysis of Japanese banks' historical Tree diagram.* Physica A, Vol. 383 (2007), pp. 164–168.

[14] http://money.cnn.com/data/currencies/

expert system, knowledge acquiring and representation,
reasoning methods, advertising

Paweł FORCZMAŃSKI*

EXPERT SYSTEM FOR VISUAL ADVERTISING DESIGN

In recent decades, artificial intelligence research has provided new tools and techniques for marketing specialists. These tools, when combined with problem-solving knowledge from a specific domain, can be used to create expert systems. This methodology is most applicable in semi-structured problem domains where the key relationships are logical rather than numerical and problem-solving knowledge is incomplete. Such typical problem in marketing is an advertising design, mostly in area of its visual concept and form. In this chapter, we describe an application of expert system techniques to the development of a system that can assist advertisers of consumer products in the process of advertising design. The main goal was to create a complete system which produces newspaper advertisement, with fully customized elements such as texts and pictures. An expert system is employed at the stage of designing the layout, selecting colors, images and other graphical forms by analyzing the advertising category, target consumers, advertized product or service. Its knowledge base is established on the analysis of current advertising campaigns observed in the newspapers. The system's rules were created using the RETE algorithm, while the implementation was done using Drools engine. Presented application, developed using Java, can be used by advertizing professionals as well as clients of the advertising agencies.

1. INTRODUCTION

1.1. ADVERTISING AS A DOMAIN OF RESEARCH

In general, an expert system is a computer program used to solve problems which require knowledge of an expert in a specific field [7]. It has a specialized knowledge of a specific area arranged in such way that it is possible to perform an interactive dialogue with a user. In other words, an expert system is an adviser to the user, which helps in

* West Pomeranian University of Technology, Szczecin, Faculty of Computer Science and Information Technology, 71-210 Szczecin, ul. Żołnierska 49, e-mail: pforczmanski@wi.zut.edu.pl

making specific decisions and explaining the reasoning [8]. By introducing an expert system we can positively affect the credibility and quality of services offered by the company. This is of great importance, taking into account the rapidly expanding enterprises and increasing competitiveness in the market. Expert systems allow for quick delivery of personalized information, thus they are highly appreciated by many customers [6]. For this reason, there can be seen growing application of expert systems in different areas of life, for example, in medicine, administration and financial sector.

In this chapter we would like to answer the question – is it possible to create an expert system for such an unpredictable field like advertising? Can this type of computer system be adapted for commercial use?

Advertising is an important part of modern free market economy. It is a tool for communication between companies and consumers. The opinion that the first impression is the most important is crucial also the case for advertising. Hence it is worth to invest more in the visually appealing, full of juicy color and energy, professionally done advertising materials. It is very important to send a clear message, but the most important is to accurately hit the tastes of consumers in the sense of visual representation of emotions. Advertising has its own rules and it is not easy to create campaigns that will not only attract attention of potential customers, but also trigger a desire to purchase goods. Often, it is difficult to achieve a compromise between the idea of advertising agency and the agency's customer. In such situations it may be helpful to introduce a tool which will guide the creative team thorough the whole process of advertising design, and explain to the customer why a particular element, like font family or color was used, based on knowledge of the experts.

The chapter is organized as follows. First, we focus on a method for creating a knowledge base and knowledge representation, as well as methods of reasoning and search strategy in the aspect of creating visual advertising. Then we present the initial assumptions and implementation of the expert system, which relies on the analysis of advertising materials taken from various newspapers. Finally we present a description of the program, including user interface and functional characteristics, as well as several tests that were carried out in the environment of specialists in advertising.

1.2. CHARACTERISTICS OF EXPERT SYSTEMS

An expert system (rule-based program) belongs to one of the areas of artificial intelligence described by Feigenbaum as "applied artificial intelligence". Such systems are employed mainly to solve problems, in which it is necessary to have an expert knowledge in the very narrow domain. Their very distinguishable feature is the separation of knowledge from the rest of the application. Expert systems are closely linked with knowledge engineering [4] (i.e. knowledge acquisition and processing), which is considered one of the most promising areas of science [7]. Expert System is defined in many ways, among others as a computer program applied to solve the problem like specialist, consisting of a base of knowledge and system of reasoning [8] or a program used to process knowledge, not data [6] or a program created to perform complex tasks as good as an expert in the same field [7].

Most of the definitions emphasize the role of an expert and the separation of knowledge from other parts of the system [1],[8]. Basic expert system consists of three components [1],[5],[6],[8]:

- Knowledge Base which is a collection of facts and rules containing knowledge needed to answer questions passed by the user. Knowledge can be stored in a declarative (Rules and facts) or procedural (procedures and functions) form.
- Inference module (controller argument) that contains information about how to solve the problem on the basis of the knowledge base and data from the user. It is responsible for finding relevant data in the knowledge database and making a decision on their basis.
- Interface for communication with the user. It provides a user with an input and output of data.

In the developed application we employed Drools engine [2] for the implementation of rule-based expert system, since it allows declarative programming and is very flexible. It is a business rule management system (BRMS) with a forward chaining inference based rules engine, using an enhanced implementation of the RETE algorithm [3]. Sample rule declared using Drools is presented below:

```
rule "Motoryzacja" \\ rule's name
agenda-group "init" \\ group definition
dialect "java"
when
\\condition
c : Company( domain == "Motoryzacja" )
then
\\action
jTextPane1.setText(tipsTexts.getMotoryzacja());
\\which group of rules should be checked
drools.setFocus( "klient" );
end
```

2. SUPPORTING EXPERT SYSTEM

2.1. DEFINITION OF A PROBLEM

Creating an advertisement is a complex process which requires an adequate knowledge. It is also largely dependent on the decisions of people, who sometimes can not enter into the role of agency’s clients, but also do not have a primary knowledge needed to create visual advertisements. The agency is often trying to reconcile the requirements of the market and customer requirements. Creating an advertisement that not only is accepted by the customer, but also increases the sale of the product is a challenging task. If one of above conditions is not realized, the advertisement will not be published and the whole campaign will fail. Both situations are unfavourable, especially for an advertising agency, because dissatisfied customer can not only apply for reimbursement, but also effectively spoil the opinion of the agency and discourage new customers. Hence the idea to create a tool for an advertising agency that should support a customer during the process of creating advertisement, selecting the most optimal solutions related to visual details and consumer preferences.

Of course, the use of such a system, for the purposes of advertising involves some risk – in the opinion of most of the experts, the advertisement should be something unique, while every system designed by a human, even if has a bit of artificial intelligence, is not able to implement a human creativity in advertising.

Having in mind above restrictions, a concept of expert system serving as a support for both customer and the agency was proposed. Its main features are analysing customer needs and requirements in the field of advertisement type, advertised product or service and target group, generating visual advertising prototype with adequate explanation of elements used and possibility of manual modification of each element.

Proposed system consists of knowledge base created on a basis of real advertisements collected from various newspapers, a collection of sample graphical elements and a user-friendly interface.

2.2. VISUAL ADVERTISING ANALYSIS

In the previous section we defined the problem and concept of a system, which relies on the analysis of existing advertisements. In the first stage of creating a knowledge base we should decide what information is needed in the process of visual advertisement design. First of all it is necessary to define the domain of product or service we want to advertise. It is a key issue because different products (services) employ different advertising strategies and means. Then it is very important to confine the advertising media in order to chose an optimal form. For the purpose of this work we decided to divide the analysis into three stages:

1. Select the media type, restricted to the visual form only: television commercial, press advertising, leaflet, poster and billboard;
2. Identify areas such as automotive or cosmetics, allowing to find relationships between various advertising schemes and expected results;
3. In-depth analysis of selected area and medium type.

Due to the large amounts of advertising material of many categories, it was possible to chose only one type of media and only one area. Because radio and television spots are volatile, we decided to focus on physical type such as newspaper advertisements. This type of advertising media occurred to be ideal, since the press is collected by many libraries and is easy to access. Similar advertising media, such as leaflets or billboards are also acceptable, however the amount of materials for analysis is not so easy to gather.

Not all categories of products or services are represented in the equal quantity, hence a next key problem was selecting the areas, which allowed to find common features of advertisements. This phase of research consisted of an analysis of several popular Polish magazines (published in 2009), such as Newsweek, Polityka, Wprost and Forbes, with dominant advertising coming from the automotive industry (in the third issue of Newsweek from 2009, nearly fifty percent of advertisements came from this area).

The analysis made it possible to find analogies between individual advertisements, among others:

- the main object is the advertised product – a car,
- the advertisement shows at least four models, clearly signed,
- an information on the current price or discount which can be obtained is shown,
- a headline highlights what is unique about the offer,
- the advertisement encourages to check further information on the web site and/or in the exhibition room,
- a logo is shown and it is located in the lower right corner,

In other areas of advertised products it is relatively hard to find such common features, however it is not impossible to make similar synthetic description.

The further analysis shown that almost all advertisements could be divided into five categories: new financial offer, new models, stock sale of the previous models, boost brand awareness, and tax deduction.

Each category has its own similar features, which is suitable to make a set of rules used further by the expert system. Hence, advert of the first category shows the car, rarely more than one model. Already in the header it presents financial benefits and reveals the price or discount percentage, together with words “price”, “savings” and the phrase “longer than” and “credit 0%”. The layouts of such advertisements in most cases look similar. In the category of “new model” it presents a picture that illustrates the benefits associated with a new car. What is emphasized is the advanced technology, or novelty, e.g. by using sentences like "fuel consumption 4.5 l/100 km”, “Feel the power” and presents results of various kinds of tests (top-valued and therefore most often used is a crash test Euro NCAP, especially if the car has reached the maximum score of 5 stars).

Advertisements assigned to the category of raising awareness of the brand do not carry any specific information on prices or features of the car. They remind of the existence of the corporation and usually consist of the image of the car, a header and a big logo. If the text appears, it is short and compact. Such advertisements use the most representative model of car that best captures the spirit of the entire brand.

The last category is aimed at business customers, namely people who can deduct VAT from the price of the car. According to the law, this offer may be related only to those models that have truck certification. Such cars come mainly from the higher price category, large in size, allowing the transport of various kinds of materials. The header contains words "Tax Deduction". The text includes car prices, before and after tax or specific information about its amount. It intentionally shows a few models, so the customer has a feeling that he has many possibilities. The more expensive the car, the greater the amount of the deduction, and thus a more tempting offer he gets.

As it was shown, every advertisement category has its own specific features, not only in the field of visual form, but also in the aspect of typical statements and emotional load. All those elements can be grouped and used in the semi-automatic design of advertisement provide we know its category, the product itself and the target group.

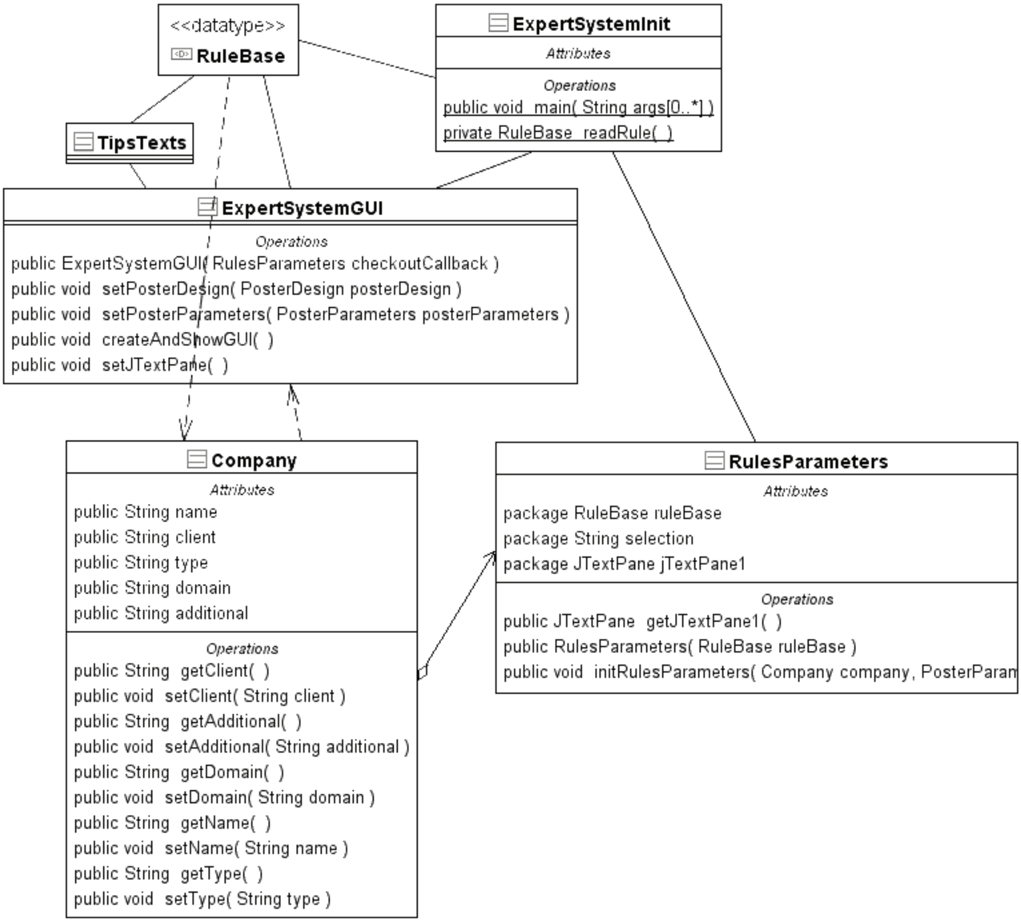
2.3. SYSTEM STRUCTURE

An application based on a concept presented in the previous sections consists of two independent modules: the first one, an expert system, which interviews the user in order to propose a general form of advertisement, and the second one is a interactive visual editor, which allows users to alter all the elements of proposed advertisement, i.e. modify colours, graphical elements, fonts, and edit all texts. In Fig. 1 and Fig. 2 one can observe general diagrams for both modules, presenting most important classes. A class *ExpertSystemInit* invokes graphical user interface (*ExpertSystemGUI*) and calls Drools engine (*RulesParameters*) responsible for passing respective objects to the rules engine. We employ Drools [2] as it is tailored for Java and allows to match the semantics of the problem domain with domain specific languages (DSL) via XML using a schema defined for the problem domain. Since DSLs consist of XML elements and attributes that represent the problem domain, the rules for different categories of advertisements are passed using adequate descriptions stored in XML files.

The class *Company* gathers all the answers given by the user and passes them to the *initRulesParamteres*, responsible for invoking the rules. Another important class is *PosterParameters* containing all the characteristics of resulting advertisement, i.e.

localization of pictures and content of texts, names and folders of the pictures, visibility attributes, font family, type and colors as well as page orientation.

The second module responsible for visual modifications of resulting advertisement involves class *PosterdDesign.* It consists of a control panel and a canvas presenting preview of the advertisement.



<<datatype>>
RuleBase
ExpertSystemInit
Attributes
Operations
public void main(String args[0..*])
private RuleBase readRule( )
TipsTexts
ExpertSystemGUI
Operations
public ExpertSystemGUI(RulesParameters checkoutCallback)
public void setPosterDesign(PosterDesign posterDesign)
public void setPosterParameters(PosterParameters posterParameters)
public void createAndShowGUI( )
public void setJTextPane( )
Company
Attributes
public String name
public String client
public String type
public String domain
public String additional
Operations
public String getClient( )
public void setClient(String client)
public String getAdditional( )
public void setAdditional(String additional)
public String getDomain( )
public void setDomain(String domain)
public String getName( )
public void setName(String name)
public String getType( )
public void setType(String type)
RulesParameters
Attributes
package RuleBase ruleBase
package String selection
package JTextPane jTextPane1
Operations
public JTextPane getJTextPane1( )
public RulesParameters(RuleBase ruleBase)
public void initRulesParameters(Company company, PosterParan

Fig. 1. Classes associated with expert system

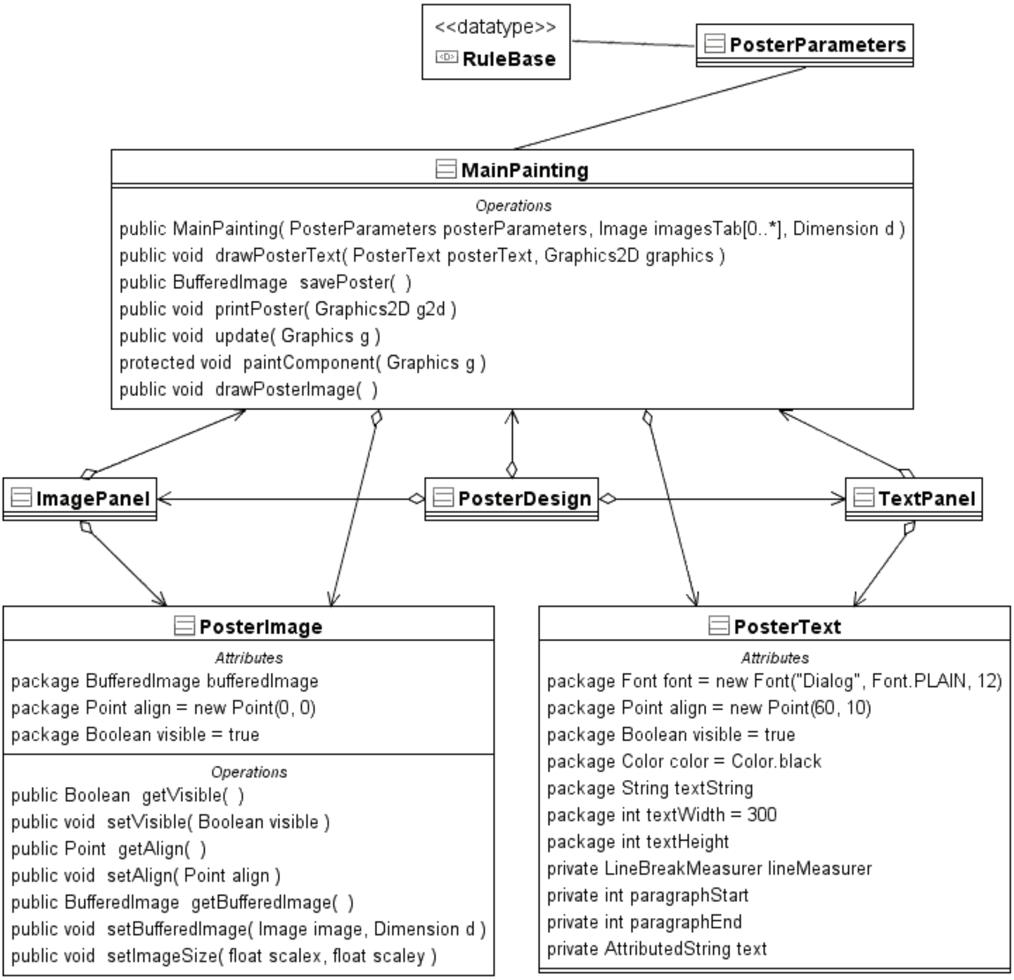

<<datatype>>
RuleBase
PosterParameters
MainPainting
Operations
public MainPainting(PosterParameters posterParameters, Image imagesTab[0..*], Dimension d)
public void drawPosterText(PosterText posterText, Graphics2D graphics)
public BufferedImage savePoster( )
public void printPoster(Graphics2D g2d)
public void update(Graphics g)
protected void paintComponent(Graphics g)
public void drawPosterImage( )
ImagePanel
PosterDesign
TextPanel
PosterImage
Attributes
package BufferedImage bufferedImage
package Point align = new Point(0, 0)
package Boolean visible = true
Operations
public Boolean getVisible( )
public void setVisible(Boolean visible)
public Point getAlign( )
public void setAlign(Point align)
public BufferedImage getBufferedImage( )
public void setBufferedImage(Image image, Dimension d)
public void setImageSize(float scalex, float scaley)
PosterText
Attributes
package Font font = new Font("Dialog", Font.PLAIN, 12)
package Point align = new Point(60, 10)
package Boolean visible = true
package Color color = Color.black
package String textString
package int textWidth = 300
package int textHeight
private LineBreakMeasurer lineMeasurer
private int paragraphStart
private int paragraphEnd
private AttributedString text

Fig. 2. Classes associated with visual editor

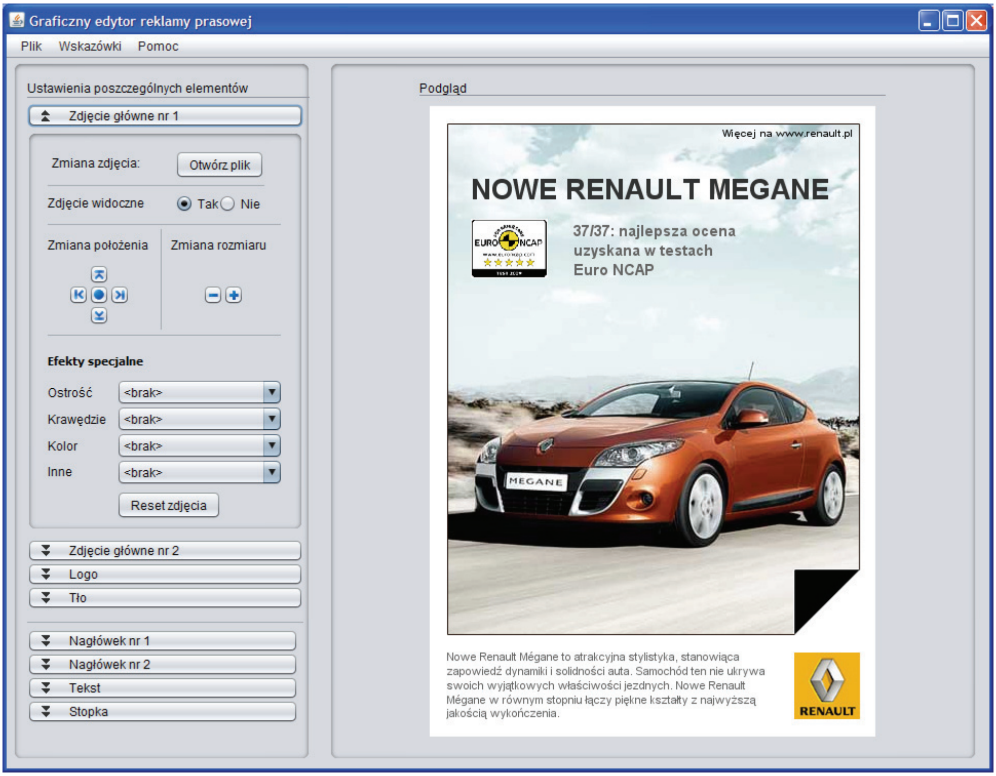
At this stage we assume that each advertisement consists of 8 elements: 4 pictures and 4 text-fields. Each element has its own panel being a composition of *ImagePanel* and *TextPanel.* After any modification of any element associated function *repaint* (*MainPainting*) is invoked, which guarantees high interactivity of the system. Class *MainPainting* allows also to save or print the advertisement.

3. IMPLEMENTATION

The system was implemented using Java and Drools package. It was motivated by the need of cross-platform compatibility and free nature of the final product. For the purpose of research we created a database containing over 400 objects: backgrounds (30), logotypes (8), images of cars (322 with and 102 without background) of 8 car brands, used to produce a final advertisement.

The system is initialized with a dialog presenting five questions from the expert system. They are related to the area of advertised product, type of client, type of commercial, car brand, and car model. After filling the form, the system presents a textual description of proposed advertisement, and goes to the resulting visual form.

The visual editor presents a preview of the advertisement created using rules from the expert system. As it can be seen in Fig. 3, thanks to the standardized form, the system is very easy to operate even by the non-specialists.



Zdjęcie widoczne ● Tak ○ Nie

Ostrość <brak>

Krawędzie <brak>

Kolor <brak>

Inne <brak>

Więcej na www.renault.pl

NOWE RENAULT MEGANE

37/37: najlepsza ocena uzyskana w testach Euro NCAP

Nowe Renault Mégane to atrakcyjna stylistyka, stanowiąca zapowiedź dynamiki i solidności auta. Samochód ten nie ukrywa swoich wyjątkowych właściwości jezdnych. Nowe Renault Mégane w równym stopniu łączy piękne kształty z najwyższą jakością wykończenia.

Fig. 3. Visual editor with resulting advertisement generated on a basis of user's answers

4. SUMAMRY

In this chapter we presented a concept of an expert system supporting advertising professionals and clients of the advertising agencies in the process of creating visual advertising materials. It supports rules created on a basis of investigations involving several Polish newspapers and several areas of products and services. In the practical part we focused on the automotive industry and advertisements of cars. During experiments the developed system was used to create different advertisement for different target groups of customers and was tested by five professionals from the advertising agencies. It was rated very good, as a tool that can dramatically reduce the time and cost of preparing the visual advertising campaign.

The future research will be focused on the implementation of such system for a different types of commercials, i.e. radio spots and internet banners, as well as different areas of advertising.

ACKNOWLEDGMENTS

The system presented in this chapter was designed and implemented under a supervision of the author as a part of Master's Thesis by Mrs Dorota Raczkowska at The Faculty of Computer Science and Information Technology, West Pomeranian University of Technology, Szczecin in 2009.

REFERENCES

[1] CHROMIEC J., STRZEMIECZNA E., *Sztuczna inteligencja: metody konstrukcji i analizy systemów eksperckich*. Akademicka Oficyna Wydawnicza PLJ, Warszawa, 1999.

[2] DROOLS – The Business Logic integration Platform, http://www.jboss.org/drools/ (accessed 27/06/2011)

[3] FORGY C., *Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem*, Artificial Intelligence, 19, 1982, 17–37.

[4] KENDAL S. L., CREEN M. , *An introduction to knowledge engineering*. Springer, 2007.

[5] KNOSALA R., *Zastosowania metod sztucznej inteligencji w inżynierii produkcji*. Wydawnictwo Naukowo Techniczne, Warszawa, 2002.

[6] MICHALSKI A., *Elementy wspomagania decyzji w zintegrowanych systemach kierowania produkcja*. Wydawnictwo Politechniki Śląskiej, Gliwice, 2000.

[7] MULAWKA J., *Systemy ekspertowe*. Wydawnictwo Naukowo-Techniczne, Warszawa, 1997.

[8] NIEDERLINSKI A., *Regułowo-modelowe systemy ekspertowe rmse*. Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, Gliwice, 2006.

artificial intelligence methods, expert systems,
computer decision-making process assist

Zbigniew BUCHALSKI*

DECISION PROCESSES MODELING IN THE COMMERCIAL FIRM DEVELOPMENT STRATEGY FORMATION

This work presents certain concept of expert system, called STALCOM, which supports management process in the commercial firm. STALCOM system has been designed to support choosing the right grade of steel according to the possibility of using this grade of steel in further processing. Fundamental establishments of systems construction, structure and functional description are presented. Particular attention was paid to knowledge base record taking into account many decisive ways for solving the problem introduced to the system. It was tried that the knowledge contained in the basis was detailed and confirmed by the knowledge of many experts. This work describe also computer implementation and testing of STALCOM system..

1. INTRODUCTION

In the era of information and knowledge, when there is an unlimited access to electronic information, contrary to common belief the process of making decisions has become more complex. The use of cheap and easily accessible computer techniques to solve complex decision problems has turned out to be extremely important.

Making decisions is an act of choosing one option (direction) of action out of a given set. This choice can be made based on a specific course of action which leads to finding the most advantageous (optimal) alternative. Intelligent informatics systems, such as expert systems play an important role in the process of supporting the decision-making process [2, 3, 5, 6, 7, 9, 10].

* Institute of Computer Engineering, Control and Robotics, Wrocław Univeristy of Technology, Wrocław, Poland, e-mail: zbigniew.buchalski@pwr.wroc.pl

Expert systems play a big role in informatics and they are used widely in many fields. They are successfully used as diagnostic, consulting, forecasting, classification and monitoring systems [1, 4, 11, 12].

The advantages of expert systems became an inspiration to design and implement the expert system called STALCOM as a response to needs of companies which produce industrial equipment and investment goods using steel. STALCOM system has been designed to support choosing the right grade of steel according to the possibility of using this grade of steel in further processing. The expertise will be based on the percent chemical composition of steel and the amount of contamination in it. After completing the expertise, STALCOM system explains the reason for the decision and the way of handling the steel as well as indicates the possible options of using the investigated grade of steel in further processing.

2. ASSUMPTIONS AND OBJECTIVES OF BUILDING STALCOM SYSTEM

The objective of building STALCOM system is to support the process of choosing a grade of steel and the estimation of possible ways of processing (welding, rolling, nitriding) in accordance to the percent composition of the given grade of steel. The percent composition of elements in the given grade of steel influences the plastic characteristics, hardness and fineness of the steel structure. STALCOM system is to determine the possible ways of processing the grade of steel and putting this system into practice may be of great help to welders.

STALCOM system has been designed to support the production process in services & trade companies which utilize steel, specifically detailed planning of steel processing. A welder who is a user of the system, at first determines the grade of steel that will be processed and when obtaining the STALCOM system expertise can take action without damaging the internal steel structure.

It has been assumed that STALCOM expertise system must meet the following requirements:

- be a tool which makes the process of designing and choosing the right grade of steel for technicians in a company easier which would allow to make the time needed to produce the ready-made goods shorter,
- the use of a language specific for experts of materials science. With regard to the working environment, the system must use the same language as the one used in technical descriptions and specifications,
- provide the possibility of adding new grades of steel along with substitutes to database, which as a result may make finding the right materials for production of ready-made goods easier,

- have a user-friendly interface with an attractive layout which is to interest users, but not to tire and make the work harder,
- be easy to use even for people who do not use computers nor the Internet on a daily basis,
- be ready to explain the decisions,
- be easily extendable and have the possibility of updating rules and facts included in the database.

3. CONSTRUCTION OF THE STALCOM SYSTEM

In this point, there will be a description of the IT environment use to implement STALCOM system and as well as the basic modules that make up this system.

3.1. PROGRAMMING ENVIRONMENT

The .NET platform and WPF (Windows Presentation Foundation) have been used to implement the STALCOM system, which allows the programmer to control in an unprecedented way the look and content of the dialogue boxes. Microsoft Visual Studio 2008 [8] editor and the SQL database have been used, there the information about grades of steel, their percent composition and available substitutes are stored. The system was designed and implemented using the C # programming language in version 3.0 and its distinctive new features such as LINQ to SQL, which make it possible to import the SQL database, all queries, views, stored procedures, etc. and use them as objects and methods.

3.2. GATHERING KNOWLEDGE AND FORMALIZATION OF DATA

After analyzing the issues and identifying the assumptions, the process of gathering knowledge in the field of materials science began. After collecting up-to-date offers of the largest suppliers of steel in Poland, the analysis was carried out. The rules used to classify and estimate properties of the grade of steel were designed.

After gathering all the information the base of rules that steel must meet in order to be admitted to the process of production of individual products was designed. The listing of steel according to these standards is a *table stal_gatunek of dbstal1* database. The following data has been entered in the successive fields:

- *stalID,* which is the identifier of steel, introduced for the implementation,
- listing of steel designations by Polish and international standards, which can be considered as substitutes on the Polish market,
- *skladID,* which is the identifier associated with the chemical composition table.

Each of the presented designation of steel has its own unique chemical composition, which is presented in a *skladchem* table in a *dbstal1* database, which is related with *stal_gatunek*, where the following data was entered in the successive fields:

- *skladID*, which is the identifier of the percent composition of the various elements introduced for the purpose of implementation,
- listing of the percent composition of various elements in a given grade of steel.

3.3. USER

User expectations of STALCOM system are as follows:

- a large number of grades of steel must be accessible, from which a proper grade of steel will be chosen and the expertise will be conducted. Listing of grades of steel must also take into consideration the designation of steel used by their suppliers,
- expertise should be conducted in accordance with the existing standards in Poland,
- user can enter new grades of steel into a database,
- the result of expertise should be provided quickly in an appropriate window.

3.4. USER INTERFACE

The user interface has been designed in a transparent manner and XAML language has been used for implementation, and more specifically speaking, a WPF application window. SQL database has been used as the knowledge base. In connection with the platform.NET and C #programming language, it provides easy access to the knowledge contained therein.

Deployment of options in the user interface has been designed in a logical way and graphical interface is very intuitive. Actions, such as adding grades of steel to the knowledge base, editing and deleting them follow the schemes which work successfully in many well-known applications. The process of designing this type of application, i.e. intended for companies supporting the production process should be guided by the principle of avoiding bright colors and ensuring a clean and simple look.

Inference is based on the contents of the knowledge base, thanks to which the contact of the system with the user is limited to the minimum, which is designed to ensure that working with the system does not become inconvenient. At any time an employee can check the possibility of welding, hardening or nitriding any given grade of steel, which significantly facilitates the work in the company. The system, instead of questioning, asks the user to select the grade of steel that he is interested in to carry out the expertise. After its completion the results are displayed in the dialogue box.

The support in choosing the grade of steel used in further processing was carried out based on the characteristics of the specific grade of steel (e.g. structural steel).

A part of this activity might only be a set of options (options of the user to select certain criteria) according to set requirements. The employee may specify what operations he will want to perform, the environmental conditions to which the steel will be exposed and a final destination of the steel product. Multiple choices are possible in this regard. After pressing the button a specific grade of steel which meets the selected criteria will be displayed.

3.5. DATABASE

SQL Server database of the program is a place from where a set of rules needed for expertise is taken. The database is divided into two parts: a table of grades of steel and a table of their percent composition. To maintain the relation between the grade and the chemical composition of the steel, the relation between the two tables in the form of the relation FK (foreign key) has been introduced. Since LINQ to SQL allows you to use not only the SQL language, but also the views stored in the database. One of these views is applied in this case. Two tables: *stal_gatunek* and *skladchem* create *MaterialyAll* table, introduced for the implementation purposes for viewing or downloading the chemical composition of grade of steel.

It is possible for a user to add, delete and edit records that are used as the fact base, using the appropriate options in the user interface. In the form, after adding the content of the records, firstly the content from the *skladchem* table is added and then from the *stal_gatunek*. This allows to obtain *SkladID*, which is generated automatically when content is added. Then the same, previously generated *SkladID* is downloaded and it is entered in the record of the same name in the table *sklad_gatunek.*. This preserves the relation between the composition and the grade of steel. The price of the grade of steel has been skipped intentionally because it is a matter of individual suppliers of steel and is negotiated depending on the customer and the quantity of steel that is ordered.

3.6. KNOWLEDGE BASE

The knowledge base expert system STALCOM contains information about the so-called acquired knowledge. This is the kind of knowledge which is gained over time by learning and reasoning. As mentioned earlier, the inference only *skladchem* tables used, containing information about the percent composition of elements in a given grade of steel. *Stal_gatunek* table serves only to identify which grade of steel meets the selected criteria. This is the most important part of the knowledge base, without which reaching the appropriate conclusions would not be possible. These are the information based on which the system will be able to choose the appropriate grade of steel.

The way of writing down the rules in the knowledge base STALCOM system varies depending on the choice of inference. The user has a choice of two options:

- Expertise – the user selecting this option the STALCOM system will be asked to select one grade of steel, upon which the expertise will be carried out. In this option more rules for inference will be used and the expertise will be more accurate because there is no need to isolate individual characteristics, operation and use of steel. An employee in this option will only check whether the selected grade of steel is suitable for further processing.

- Selection – after selecting this option, the user will be moved to the dialogue box and will be asked to select the operations that will be performed on the selected grade of steel (welding, rolling, nitriding). The user will also need to determine atmospheric conditions under which the steel will be used (seasonal conditions, high temperature, low temperature, high pressure, moisture). Eventually the user will be able to choose the purpose of the steel with options such as tools, pipes, kettles, structures, casings, drills, bearings. Isolating individual properties is required(e.g. acid resistance is characterized by high levels of chromium in the chemical composition), but inference is not so obvious for all the properties. In the chemical composition fields it is required to enter floating point values specific for grade of steel markings on all descriptions provided by suppliers of steel in Poland.

3.7. INFERENCE PROCEDURES

During conducting the exercise, STALCOM system asks us to choose the grade of steel we are interested in at the beginning and after pressing a button the expertise will be conducted based on the chemical composition of the selected grade of steel. This will be an immediate process(without asking specific questions). We learn at first, whether the selected grade of steel is alloyed or unalloyed. It is an essential categorization, because all further conclusions are made based on this information.

The next step is to calculate the factor, which allows determining the weldability of the grade of steel. Weldability is calculated immediately after pressing a button in Expertise and Choice type of reasoning. If the option to find the appropriate grade of steel is chosen, only selecting the fields of interest to us is needed (operations performed, the conditions of use, destination), then the grades of steel meeting the selected criteria will be displayed in the table.

3.8. STRUCTURE OF STALCOM PROGRAM

STALCOM expert system serves as a support in choosing the accurate grade of steel and in obtaining then expertise based on the chemical composition stored in a knowledge base. The overall goal is to choose the grade of steel, according to the criteria selected by the user. The system examines specific grades of steel which meet the European Union standards, gives a detailed list of uses for a specified grade of steel and the ways to manipulate the selected grade of steel, without affecting its properties. The system is designed for heavy industry workers, namely welding technicians, who use a professional language specific for materials science.

STALCOM system consists of two components: SystemEkspertowy and SystemMaterialy, which inherit objects from each other. The tasks of these components are as follows:

- SystemMaterialy – provides support for database, that is dbstal1.mdf file and Microsoft SQL Server provides a connection to the database Component SystemMaterialy also includes an Admin.cs class, where the procedures for adding, editing and deleting items to the database are described. Da-neMateriały.dbml procedure is another important component, which is generated by LINQ to SQL. It is here where the databases are really declared.
- SystemEkspertowy – this is the main part of the program, which inherits from SystemMaterialy. It runs STALCOM program. There is a main menu and the declarations of all dialogue boxes. The dialogue boxes view is implemented using XAML.

4. COMPUTER IMPLEMENTATION OF STALCOM SYSTEM

STALCOM expert system has been implemented as a program of the same name in C # using Visual Studio 2008 of Microsoft, MSDNAA license. STALCOM system as a tool to carry out the expertise consists of two parts: the application and the SQL database, which task is to collect data in the knowledge base. The application process is dependent on the knowledge base, where there is a set of facts and rules necessary to carry out the expertise. The function of database server is carried out by Microsoft SQL Server 2005.

For proper functioning of the application, installing the following tools is required:

- .NET Framework version 3.5 or later (for PCs equipped with Windows XP SP2 or later),
- .NET Compact Framework version 3.5 or later SP2 (for mobile phones, PDAs equipped with Windows Mobile operating system),
- Microsoft SQL Server version 2005 or later.

Main Menu of STALCOM system, using which the user will use all application functions (such as adding new facts to the database, displaying database content, editing and deleting content), is declared in STALCOM.xaml. Immediately after running the SystemEkspertowy.exe file a dialog box launches.

In the beginning, our attention is drawn to simple and transparent window content. In the centre of the screen we will find a button titled “Start Expertise”. This button invokes the occurrence declared in “btnEkspertyza_Click”. However, before the main function of the program is described, let's look at other options of StlXpert system. These options can be found in the menu bar.

When you move the mouse cursor on the menu bar “View”, a tab with available options displaying the content of the database is shown (a grades of steel, and chemical composition table). After pressing the first option, that is, “View Resources” you will be transferred to the WPF dialogue box, which is called “PokazMaterialy”. Here you can see the headers and a list of all records in the table. The “Show Chemical Composition” button shows the contents of the *skladchem* table, which for being similar to this option is skipped in this part of description.

After selecting the second option in the menu bar “View”, entitled “Show Materials and Chemical Composition”, the user sees the result of a SQL query, which, thanks to LINQ to SQL is used as an object of MaterialyAll during compilation, as well as tables in the database.

An output of this query is creating a view in SQL language, wherein the records of each grade of steel are assigned to each other. Thanks to this, in the inference part, only the grades of steels with IDs assigned to each composition SkladID are taken into account. This allows avoiding a situation in which during inference a grade of steel with no assigned composition is taken into account.

The Expertise using the STALCOM program can be launched by pressing the “Start Expertise button. In the dialog box our attention is drawn to two buttons: “Choose from Materials” and “Adjust Steel”.

The “Select Content” button means an inference based on selecting one item from a list of grades of steel and pressing a button. The Expertise will be carried out on the basis of chemical composition selected of the grade of steel selected by the user. On the basis of a set of rules a steel categorization will be made. The system gives a general description of the selected type of steel, its destination and possible operations which can be performed on this grade of steel. Almost immediately a window with the final expertise opens. The window shows the text box type RichTextField from which you can freely copy the highlighted section of text by pressing Ctrl + C. The STALCOM Expertise appears precisely in this field.

When invoking the “Choose from Materials” it is necessary to pre-select the grade of steel of interest, which is to be investigated. And when you do not make the choice before pressing the button, you will be informed about this in the dialogue box. After finishing working with the program a STALCOM system expertise is displayed in a transparent way.

The “Adjust Steel” button means the inference based on all the elements in the knowledge base, with assigned chemical compositions. In reality this means reasoning backwards, resulting in searching the fact base for selected properties of the steel. By pressing the “Adjust Steel”, the user will be asked to enter the properties of steel of interest. It is possible to make a multiple choice, thus there is a big number of choices. However, it is recommended to add the maximum number of grades of steel to the knowledge base, since the set of proper chemical composition can become really small. After unchecking the properties of steel press “Adjust” so that the grades of steel which meet the requirements indicated in the form appear at the bottom of the screen.

5. TESTING THE STALCOM SYSTEM

STALCOM expert system has been run on two operational systems: Windows XP Service Pack 2 32bit and Windows 7 64bit, so you can check whether the system will work properly on the operating system with 64-bit architecture. It turned out to be necessary to install .NET Framework version 3.5 (on Windows XP), Windows 7 has .NET Framework installed by default.

The purpose of testing the system was to detect and correct errors in the program and get rid of needless lines of code. Most of the testing took place already during the precompile when designing in Visual Studio 2008. It was also necessary to make sure that one part of the program was not dependent on another, and more specifically speaking, the so-called “dependent variables”. In order to ensure a correct display of information it was ensured that all modifications in the database (such as adding, editing and deleting records / facts) are actually changed in the database.

The correctness of inference has also been tested, on the basis of 10 grades of steel downloaded from the manufacturer website, they were put into a database and the expertise was carried out. After analyzing the data from the manufacturer description and from the process of inference conducted by STALCOM no significant differences have been noticed. The program launches after running the SystemEkspertowy.exe file. It is worth mentioning that it is not necessary to first run a client-server application for the SQL database, because it is launched automatically when starting the STALCOM expert system.

6. SUMMARY

The STALCOM expert system presented in this chapter is a confirmation of the possibility of using artificial intelligence methods for practical applications. This system in a satisfactory way fulfills the tasks put before him.

In order to build the system new solutions were used, which will allow to further develop the application in the future. The software used to design an application executed on the platform. NET-Visual Studio 2008, building a knowledge base based on the SQL Server database and the fact that pre-running the SQL Server application to be able to connect to the database is not required are undoubted advantages of the STALCOM system.

Due to the destination of the application, a simple look was used and the programmers were guided by the highest possible functionality. Communication with the database allows for updating the knowledge base of the STALCOM system, and more specifically a set of facts that are entered by the user when adding a new grade of steel. This allows maintaining a fast response time of the expert system.

REFERENCES

[1] BUCHALSKI Z., *Komputerowe wspomaganie podejmowania decyzji z wykorzystaniem regułowego systemu ekspertowego,* In: Komputerowo zintegrowane zarządzanie, t.1, R. Knosala (ed), Warszawa, WNT, 2004, 156–164.

[2] BUCHALSKI Z., *The Role of Symbolic Representation of Natural Language Sentences in Knowledge Acquisition for Expert System.* Polish Journal of Environmental Studies, Vol.16, No.4A, 2007, 40–43.

[3] BUCHALSKI Z., *Zarządzanie wiedzą w podejmowaniu decyzji przy wykorzystaniu system ekspertowego, In:* Bazy danych. Struktury, algorytmy, metody, Warszawa, WKiŁ, 2006, 471–478.

[4] BUCHALSKI Z., *Computer Advisory-Decision System for the Logistics Services Support*. Polish Journal of Environmental Studies, Vol.18, No. 3B, 2009, 53–57.

[5] CHROMIEC J., STRZEMIECZNA E., *Sztuczna inteligencja. Metody konstrukcji i analizy systemów eksperckich,* Warszawa, Akademicka Oficyna Wydawnicza PLJ, 1994.

[6] NIEDERLIŃSKI A., *Regułowo-modelowe systemy ekspertowe,* Gliwice, Pracownia Komputerowa Jacka Skalmierskiego, 2006.

[7] OWOC M., *Elementy systemów ekspertowych, cz,1:Sztuczna inteligencja i systemy ekspertowe,* Wrocław, Wydawnictwo Akademii Ekonomicznej im. Oskara Langego we Wrocławiu, 2006.

[8] RANDOLPH N., *Professional Visual Studio 2008,* Willey, USA, 2008.

[9] RUTKOWSKI L., *Metody i techniki sztucznej inteligencji,* Warszawa, PWN, 2006.

[10] STEFANOWICZ B., *Systemy eksperckie. Przewodnik,* Warszawa, PWN, 2003.

[11] TWARDOWSKI Z., *Inteligentne systemy wspomagania decyzji w strategicznym zarządzaniu organizacją gospodarczą,* Katowice, Wydawnictwo Akademii Ekonomicznej w Katowicach, 2007.

[12] ZIELIŃSKI J., *Inteligentne systemy w zarządzaniu. Teoria i praktyka,* Warszawa, PWN, 2000.

structured data, reusable components,
data preprocessing, graph mining,
visualization, edition, biological networks

Krzysztof ŚWIDER*, Bartosz JĘDRZEJEC*

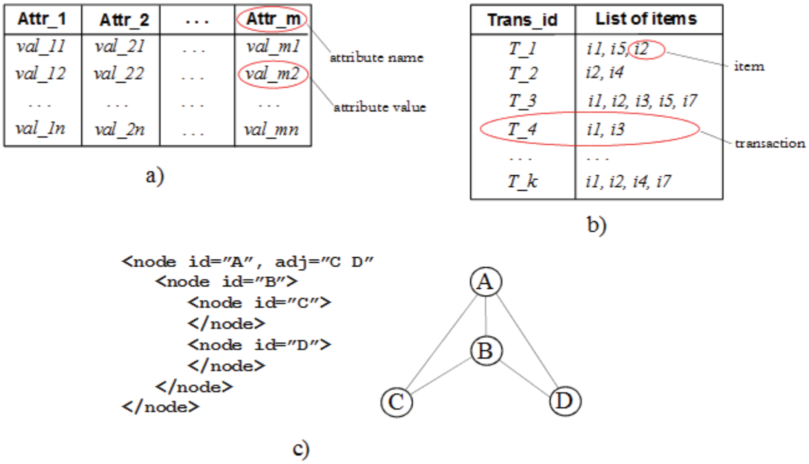
AN INTEGRATED ENVIRONMENT FOR MANAGING AND MINING STRUCTURED DATA

The work presents the current state of the on-going project aimed at development of the modular integrated environment MMSD for Managing and Mining Structured Data. The authors proposed a unified modeling scheme for a range of structures extracted from original resources and developed modules for their transformation into prescribed graph types (directed/undirected, labeled/unlabeled). The cohesive representation enables further processing of the structures including the application and development of graph mining algorithms. For practical reasons, the modules accomplishing visualization and edition of graph models are also provided. This allows, in particular, to draw example graphs from scratch and edit them for experimental and research purposes. Other worthy features are: the automatic coding of graphs in several formats and the ability to generate graphic files for presentation. Due to the application of publicly available program components we achieved the relatively high degree of reusability while implementing MMSD. The open architecture enables an easy enhancement of system functionalities as well as its potential to cooperate with number of existing structured data repositories.

1. INTRODUCTION

An important quality of modern information systems is the continuously increasing rate of data overcoming the commonly used *attribute-value* or *transaction* representations. Such data is commonly referred to as (*semi-*)*structured* because of the highly complex and irregular structure, which, in fact, contains a considerable part of its semantics (Fig. 1). The structured data are usually modeled as trees and graphs, and commonly used in chemistry, bioinformatics, pattern recognition, Web usage, XML documents analysis, software engineering and social processes.

* Rzeszów University of Technology, W. Pola 2, 35-959 Rzeszów.



Attr_1	Attr_2	. . .	Attr_m
val_11	val_21	. . .	val_m1
val_12	val_22	. . .	val_m2
. . .	. . .	. . .	. . .
val_1n	val_2n	. . .	val_mn

Trans_id	List of items
T_1	i1, i5, i2
T_2	i2, i4
T_3	i1, i2, i3, i5, i7
T_4	i1, i3
. . .	. . .
T_k	i1, i2, i4, i7

```
<node id="A", adj="C D"
   <node id="B">
      <node id="C">
      </node>
      <node id="D">
      </node>
   </node>
</node>
```

Fig. 1. The conventional (schema-based) vs. structured data representations; a) a relational attribute value table; b) a set of transactions containing items; c) structured data represented by XML code and a graph

The simple set of interrelated nodes, represented in Fig.1c in parallel as XML code and as an undirected connected graph, intuitively illustrates the nature of structured data and its distinctiveness when compared with schema-based representations. Since most of the real-life structured data usually form a complex and expressive data types, we need methods for representing such data in databases, as well as techniques and tools for manipulating and querying them. Another important issue is the development of effective algorithms for mining graphs and trees typically used to model structured data.

The subject of managing and mining graph data was widely studied and outlined in [1, 2]. In this work we address the problem of effective preprocessing and modeling the structured data stored in existing (e.g. biological) databases and presumed for analysis by graph mining algorithms. In general the term of ‘data preprocessing’ refers to any operation on „raw” data, performed with the aim of setting up the data for further processing (analysis). As applied to conventional data it is a set of operations carried out in the early stage of knowledge discovery process typically connected with the construction of data warehouse and including: cleaning, integration, transformation and reduction. Considering only the ‘analytical’ part of the knowledge discovery process the time rate spent for data preprocessing is estimated to be about 75%. On the other hand the impact of data preprocessing stage on the successful completion of the project is approximately the same. Data preprocessing is an issue often ignored in data mining community, however, without adequate preparation of data, the return on the resources invested in mining is certain to be disappointing.

The effective preprocessing stage becomes even more important for structured data due to the extended abilities to represent semantics (along with attribute values also existing associations and relations may be reflected). Further structured data usually forms huge collections often publicly available as global, typically Web accessed, databases. Since by its definition the structured data has irregular and complex structure, some traditional methods of data cleaning, e.g. the operation of reconstruction of missing values, are no more valid. In addition there is usually a high probability of noise (in many cases data comes from experiments and different sources). In comparison to conventional data, the structured data typically has to be modeled by graphs and the analytical tasks performing on such data are usually more complex (Fig. 2).

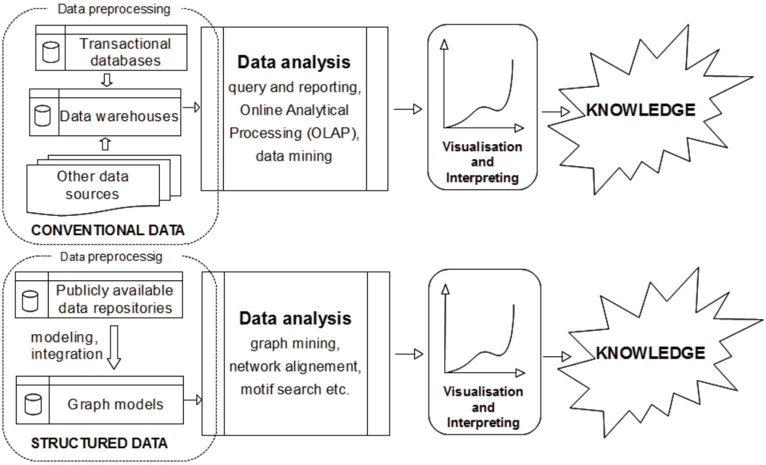

Data preprocessing
Transactional databases
Data warehouses
Other data sources
CONVENTIONAL DATA
Data analysis
query and reporting, Online Analytical Processing (OLAP), data mining
Visualisation and Interpreting
KNOWLEDGE
Data preprocessig
Publicly available data repositories
modeling, integration
Graph models
STRUCTURED DATA
Data analysis
graph mining, network alignement, motif search etc.
Visualisation and Interpreting
KNOWLEDGE

Fig. 2. The global knowledge discovery process for conventional and structured data

As in the case of conventional data types there exists a number of problems in mining structured data. One of the most challenging issues and a focused theme in data mining research is *frequent pattern mining*. One way of formulating the problem for graph datasets is that of discovering subgraphs occurring frequently in a given input graph dataset. More formally, given a set of graphs D, each of which is an undirected labeled graph and a parameter s, such that $0 < s \leq 1$, find all connected, undirected graphs, that are subgraphs in at least $s|D|$ of the input graphs.

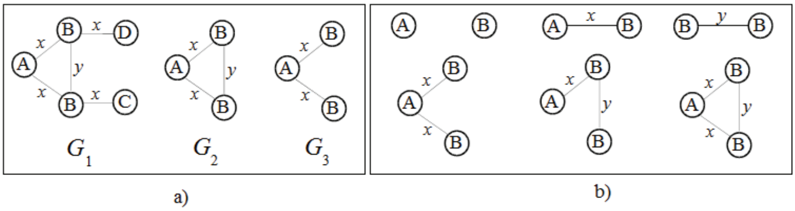


Fig. 3. Mining frequent subgraphs; a) a graph set; b) the discovered frequent subgraphs

The parameter *s* is usually referred to as *minimum support threshold*. For the simple graph database in Fig. 3a and the minimum support threshold set to 0.5 the mining procedure resulted with the set of frequent subgraphs shown in Fig. 3b.

2. SYSTEM CONTEXT AND FUNCTIONALITIES

The context and overall structure of our integrated environment for managing and mining structured data (MMSD) is depicted in Fig. 4.

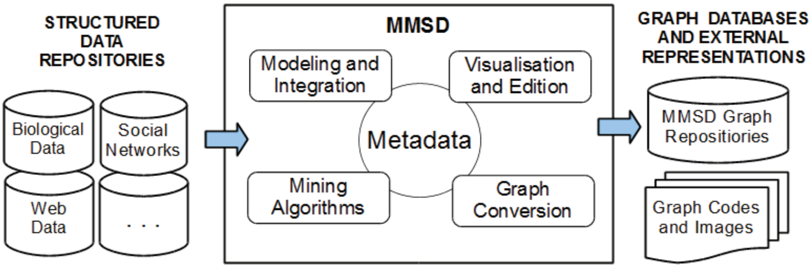


Fig. 4. The context of MMSD

The MMSD application consists of the metadata describing the structure and organisation of graph repositories and the number of modules performing the management and mining operations on structured data. The metadata reflects the structure of the graphs used for modelling as well as the global organisation of the uniquely named graph databases and database groups managed by the system. The unified graph structure provides a flexible way to define and operate with a wide range of graphs. It consists of the three main parts: parameters (graph name, date of creation, graph type, colour palette, etc.), vertices (vertex identifier, position, colour, shape etc.) and edges (edge identifier, a source and destination vertices, etc.). Each graph is coded as an XML document and for efficiency reasons stored in a separate file.

The graph database is a set of homogeneous graphs. Using the graph databases is an effective way to manage a number of graph sets used for analysis. Each graph database is characterized by parameters which prescribe the type of stored graphs (directed/undirected, connected/unconnected, labelled/unlabelled etc.). Another option is to arrange the graph databases into database groups. The metadata is stored in the separate file which contains the parameters of databases and database groups as well as the paths to files describing individual graphs.

For a range of structures extracted from original data resources we proposed a unified modeling scheme and developed modules for their transformation into prescribed graph types. The cohesive representation enables further processing of the structures including the application and development of graph mining algorithms. For practical reasons, the modules accomplishing visualization and edition of graph models are also provided. This allows, in particular, to draw example graphs from scratch and edit them for experimental and research purposes. Other worthy features are: the automatic coding of graphs in several formats and the ability to generate graphic files for presentation. The open architecture enables an easy enhancement of system functionalities as well as its potential to cooperate with number of existing structured data repositories.

3. DESIGN AND IMPLEMENTATION OF MMSD

The global package diagram in Fig. 5 provides some information concerning the MMSD architecture.

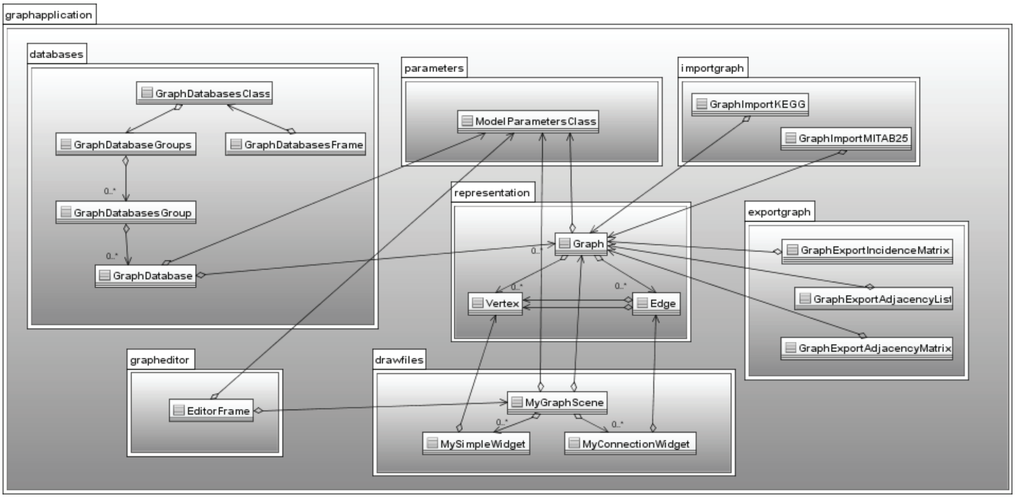


Fig. 5. The packages containing essential MMSD classes

The *representation* package is responsible for an inner graph representation with vertexes and edges. The *Graph* class is connected with the *ModelParametersClass*, which is a part of the *parameters* package and describes the graph type and its parameters. The *database* package is related to graph storing and includes classes which control the particular parts of this process. The key job of the top level class named *GraphDatabasesClass* is to read and write the structure of metadata from/to configuration files. The following class *GraphDatabasesGroups* is responsible for the groups of graph databases and allows to add, remove or modify them. Similar functionalities are provided for graph databases by *GraphDatabasesGroup* class. The functions of *GraphDatabase* class involve: reading, writing and managing information about individual graph files localization. The last class *GraphDatabasesFrame* in the package defines a user interface for metadata structure administration. The next package named *importgraph* allows user to import graphs from external sources, transform into internal form as well as write into specified database. In order to enable communication with external applications the *exportgraph* package was developed. It allows in particular to export graphs into popular representations such as: incidence and adjacency matrices, edge lists, etc.

Due to the application of publicly available program components we achieved the relatively high degree of reusability while implementing MMSD. The system was written in Java language with the application of various libraries including: Xerces, JUNG2, FreeHEP VectorGraphics and others. The application of the reusable components while constructing MMSD considerably reduced the time spent on software development, especially for implementing typical functionalities such as: the access to external repositories via XML files, the calculation of graph layout for visualisation, an export of graph into image files etc. Besides of the reusable components some new libraries were needed to accomplish the novel and uncommon tasks but the overall process of testing proceeded faster, as the existing components were previously verified by their authors and other developers.

4. BIOLOGICAL DATA PREPROCESSING AND MODELING

The investigation of biological networks for their better understanding and making available for practical use is currently the important task in *systems biology*. Graph models for biological networks may be straightforward as in case of protein-protein interactions [8], but sometimes modeling may require more sophisticated operations [9].

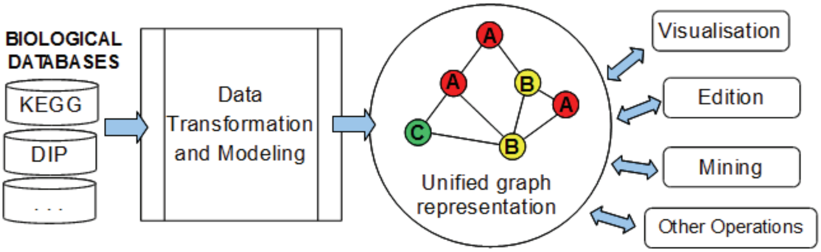
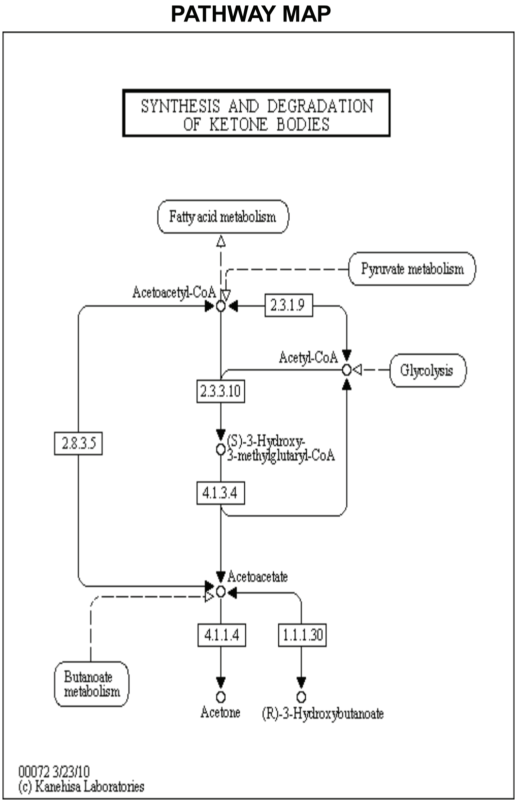


Fig. 6. Managing and mining biological data

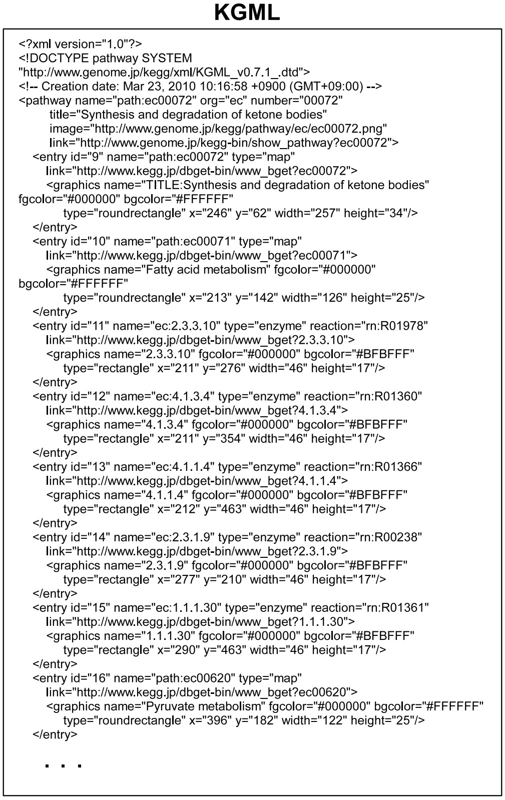
The general schema of preprocessing structured data representing biological networks is depicted in Fig.6. The authors equipped MMSD with a number of functionalities aimed to perform data extraction and modeling operations as well as visualization of resulting graphs. In order to present how the system performs with real data, we used the biological structures representing metabolic networks. A metabolic network may be defined as a collection of objects and relations among them [6]. The objects stand for chemical compounds, biochemical reactions, enzymes and genes.

An example of metabolic pathway network taken out from KEGG (*Kyoto Encyclopedia of Genes and Genomes*) [5] has the form depicted in Fig.7.



PATHWAY MAP

00072 3/23/10
(c) Kanehisa Laboratories



KGML

```
<?xml version="1.0"?>
<!DOCTYPE pathway SYSTEM
"http://www.genome.jp/kegg/xml/KGML_v0.7.1_.dtd">
<!-- Creation date: Mar 23, 2010 10:16:58 +0900 (GMT+09:00) -->
<pathway name="path:ec00072" org="ec" number="00072"
         title="Synthesis and degradation of ketone bodies"
         image="http://www.genome.jp/kegg/pathway/ec/ec00072.png"
         link="http://www.genome.jp/kegg-bin/show_pathway?ec00072">
    <entry id="9" name="path:ec00072" type="map"
        link="http://www.kegg.jp/dbget-bin/www_bget?ec00072">
        <graphics name="TITLE:Synthesis and degradation of ketone bodies"
fgcolor="#000000" bgcolor="#FFFFFF"
             type="roundrectangle" x="246" y="62" width="257" height="34"/>
    </entry>
    <entry id="10" name="path:ec00071" type="map"
        link="http://www.kegg.jp/dbget-bin/www_bget?ec00071">
        <graphics name="Fatty acid metabolism" fgcolor="#000000"
bgcolor="#FFFFFF"
             type="roundrectangle" x="213" y="142" width="126" height="25"/>
    </entry>
    <entry id="11" name="ec:2.3.3.10" type="enzyme" reaction="rn:R01978"
        link="http://www.kegg.jp/dbget-bin/www_bget?2.3.3.10">
        <graphics name="2.3.3.10" fgcolor="#000000" bgcolor="#BFBFFF"
             type="rectangle" x="211" y="276" width="46" height="17"/>
    </entry>
    <entry id="12" name="ec:4.1.3.4" type="enzyme" reaction="rn:R01360"
        link="http://www.kegg.jp/dbget-bin/www_bget?4.1.3.4">
        <graphics name="4.1.3.4" fgcolor="#000000" bgcolor="#BFBFFF"
             type="rectangle" x="211" y="354" width="46" height="17"/>
    </entry>
    <entry id="13" name="ec:4.1.1.4" type="enzyme" reaction="rn:R01366"
        link="http://www.kegg.jp/dbget-bin/www_bget?4.1.1.4">
        <graphics name="4.1.1.4" fgcolor="#000000" bgcolor="#BFBFFF"
             type="rectangle" x="212" y="463" width="46" height="17"/>
    </entry>
    <entry id="14" name="ec:2.3.1.9" type="enzyme" reaction="rn:R00238"
        link="http://www.kegg.jp/dbget-bin/www_bget?2.3.1.9">
        <graphics name="2.3.1.9" fgcolor="#000000" bgcolor="#BFBFFF"
             type="rectangle" x="277" y="210" width="46" height="17"/>
    </entry>
    <entry id="15" name="ec:1.1.1.30" type="enzyme" reaction="rn:R01361"
        link="http://www.kegg.jp/dbget-bin/www_bget?1.1.1.30">
        <graphics name="1.1.1.30" fgcolor="#000000" bgcolor="#BFBFFF"
             type="rectangle" x="290" y="463" width="46" height="17"/>
    </entry>
    <entry id="16" name="path:ec00620" type="map"
        link="http://www.kegg.jp/dbget-bin/www_bget?ec00620">
        <graphics name="Pyruvate metabolism" fgcolor="#000000" bgcolor="#FFFFFF"
             type="roundrectangle" x="396" y="182" width="122" height="25"/>
    </entry>

        . . .
```

Fig. 7. The pathway map and symbolic representations of a metabolic network

The network is provided in parallel as the pathway image file and in the text document written in KGML (KEGG Markup Language) – an exchange format for KEGG pathways based on XML. The graph models of metabolic networks are obtained by MMSD using the KGML files as input.

There are three major entities in KGML: entry, relation, and reaction. The entry element represents various biomolecules in the metabolic pathway, such as enzyme, gene, compound etc. The relation denotes a relationships between two or more enzymes, genes and maps (the map entry denotes the nodes linked to other pathways). The reaction is a biochemical reaction between two or more compounds catalyzed by one or more enzymes. In biochemical semantics entries are nodes of metabolic pathways and relations and reactions are relationships between two or more entries.

In the graph representation used in several studies (e.g. [7,9]) relations and reactions are also representing as vertices in order to better describe the properties of the network. In this case it is necessary to transform the structures coded in KGML document into 'extended' graph representation.

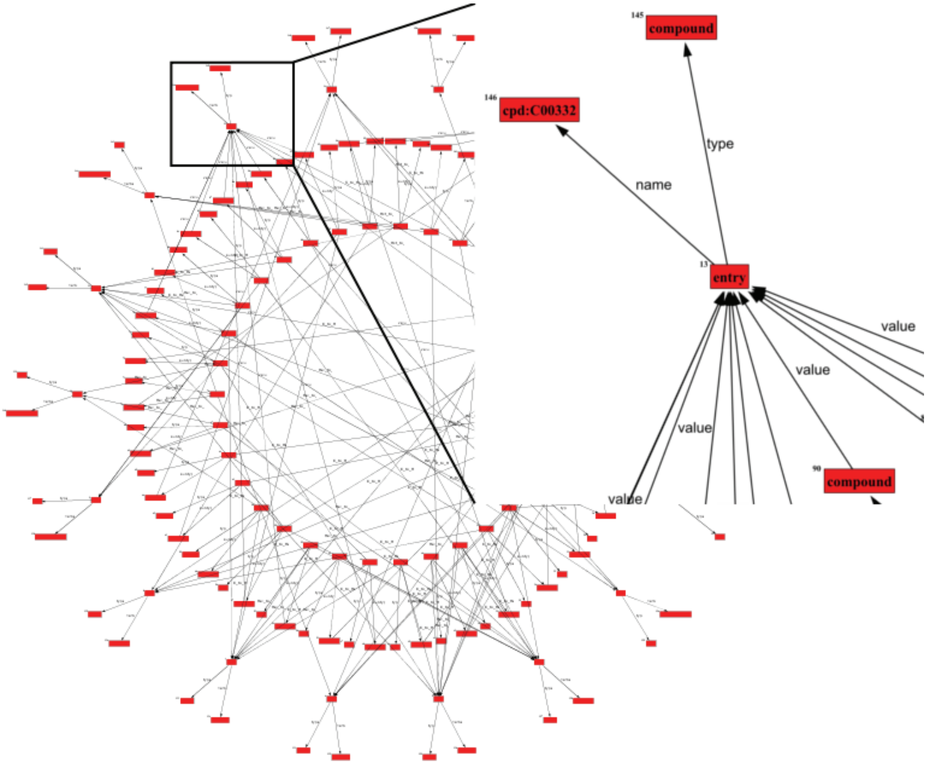


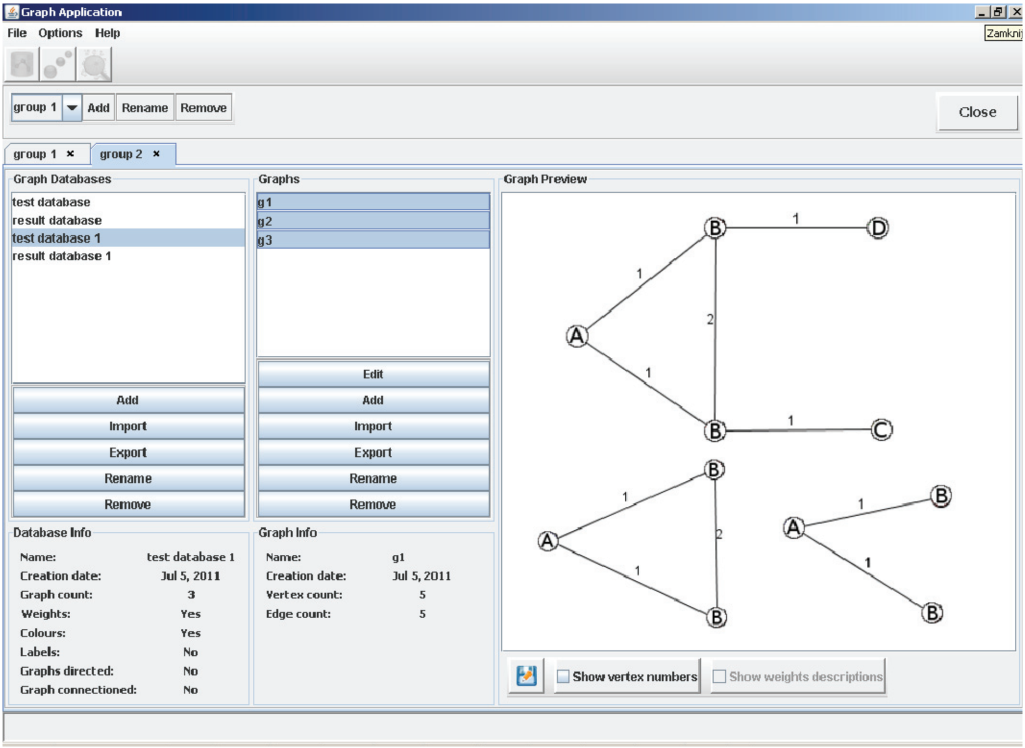
Fig. 8. The graph model of a metabolic pathway network produced and drawn by MMSD

The Figure 8 shows the graph representing the model of the input biological structure produced and visualized by MMSD . The vertices standing for major entities (entries) have two satellite vertices connected by edges labeled as name and type. The type edge connects an entity with a vertex describing its type (eg. compound) while the name edge connects an entity with a vertex containing its KEGG ID (eg. cpd:C00332).

A relation node represents the association between two or more entries (genes or enzymes) and is connected with these entities with edges whose labels describe the direction from one entry to another. The reaction node is also connected with two compounds (substrat and product) and an enzyme as a catalyst. As these entities are represented by entry nodes, the further information is given by labels of the connecting edges.

5. MINING FREQUENT SUBGRAPHS

The essential feature of the MMSD integrated environment is its ability to perform various analytical operations on single graphs or on graph sets representing structured data. In particular the system is ready to be extended by implementing a range of existing graph mining algorithms. In order to demonstrate these analytical potential we use an Apriori-based connected Graph Mining (AcGM) algorithm, intended to discover frequent subgraphs from the set of graphs [4]. A MMSD screen for managing graph repositories is depicted in Fig. 9. For simplicity reasons the set consisting of the three elementary connected graphs shown in the main window will be considered. The graphs were prepared by graph editor using four types of vertices (A, B, C and D) and two numbers (1 and 2) as edge labels.



test database
result database
test database 1
result database 1

g1
g2
g3

Database Info

Name:	test database 1
Creation date:	Jul 5, 2011
Graph count:	3
Weights:	Yes
Colours:	Yes
Labels:	No
Graphs directed:	No
Graph connectioned:	No

Graph Info

Name:	g1
Creation date:	Jul 5, 2011
Vertex count:	5
Edge count:	5

Fig. 9. A MMSD screen for user interaction

The next steps are: to set a minimum support threshold and to press the Process button which starts the mining process. The resulting set of graphs is stored in the destination database and is ready to further processing, visualisation, edition, etc.

6. CONCLUSIONS

The structured data are usually modeled as trees and graphs, and commonly used in chemistry, bioinformatics, pattern recognition, Web usage, XML documents analysis, software engineering and social processes. The current work was focused on the effective extraction, integration and modeling of structured data coming from various real-life repositories. The authors proposed the unified graph representation for the structures extracted from original resources and developed the software for their transformation, visualization and edition. This resulted in a prototype version of an open environment for managing and mining structured data with special emphasize on preprocessing and modeling operations.The representative system capabilities were demonstrated on modelling biological data describing metabolic pathway networks and mining frequent pattern in a graph database.

The project was supported by the grant N516 016 32/1938 of Polish Ministry of Science and Higher Education and the RUT young scientists support funds U-8276/DS/M.

REFERENCES

[1] AGGARWAL C., WANG H., *Managing and Mining Graph Data*, Springer Publishing Company, Inc., 2010.

[2] COOK D., HOLDER L., *Mining Graph Data*, John Wiley and Sons Inc., 2007.

[3] HAN J., KAMBER M., *Data Mining. Concepts and Techniques, 2nd ed.*, Elsevier Inc., 2006.

[4] INOKUCHI A., WASHIO T., MOTODA H., *An Apriori-based Algorithm for Mining Frequent Substructures from Graph Data*, Proc. of the PKDD'00, 2000, 13–23.

[5] KANEHISA M. GOTO S., *Kyoto Encyclopedia of Genes and Genomes*, Nucleic Acids Research, Vol. 28, No. 1, 2000, 27–30.

[6] LACROIX V., COTTRET L., THÉBAULT P., SAGOT M.-F., *An Introduction to Metabolic Networks and Their Structural Analysis*, IEEE/ACM Trans. Comput. Biol. Bioinformatics, Vol. 5, No. 4, 2008, 594–617.

[7] ŚWIDER K., JĘDRZEJEC B., *Modeling and Integration of Biological Networks with BiNArr*, The 8th European Conference on Mathematical and Theoretical Biology, Kraków, 2011 (*http://www.impan.pl/~ecmtb11/showabstract.php?id=Swider_Krzysztof*).

[8] XENARIOS I., RICE D.W., SALWINSKI L., BARON M.K., MARCOTTE E.M., EISENBERG D.: *DIP: the Database of Interacting Proteins*, Nucleic Acids Res., Vol. 28, No. 1, 2000, 289–291.

[9] YOU C., HOLDER L., COOK D., *Substructure Analysis of Metabolic Pathways by Graph-based Relational Learning*, Biomedical Data and Applications, Vol. 224, 2009, 237–261.

*multicast, packet networks, routing,
algorithm comparison software*

Sławomir HANCZEWSKI*, Krzysztof STACHOWIAK*,
Maciej STASIAK*, Piotr ZWIERZYKOWSKI*

THE ARCHITECTURE OF A SIMULATION SOFTWARE THAT ENABLES A COMPARISON OF ROUTING ALGORITHMS

In research studies it is often possible to take advantage of a ready-made computer program that is dedicated to a given type of calculations. In certain cases, however, a necessity to proceed with a self-tailored experimental software may be unavoidable. Writing a reliable program is a challenge in itself. This, however, may lead to a situation where the quality of a program can interfere with the quality of research. Research aspects are obviously of primary importance, but, if as a result of a given decision made by the researcher such a software starts to slow down the research process, the program eventually will not meet the requirements of a very dynamic context that characterizes all innovative experiments. The present work proposes an architecture of the experimental computer program that is adaptable and well suited to keep up with the foreseeable pace of any research work and that makes it possible to maintain the sustainablity of the development of the program and to minimize the amout of work that is needed for the program to adjust to subsequent new research projects.

1. INTRODUCTION

The goal of the works carried out within the framework of the project "Future Internet Engineering" (FIE, Polish: **IIP**) [1] is to construct a prototype of a modern network that would address and eliminate all the shortcomings and flaws of the Internet network that is currently used. The concept of the network in construction is based on a multi-level virtualization [2]. This network architecture is presented in detail in [1]. The **IIP** project envisages construction of as many as three Parallel Internets (Polish: *Równoległe Internety* (**RI**)), within the framework of which virtual networks (i.e.

* Poznań University of Technology, Chair of Communication and Computer Networks.

those characterized by appropriate properties in relation to services offered) are to be created. In the project, the present authors have been assigned to a group responsible for routing in IPv6 QoS Parallel Internet and in virtual networks created within the RI. It has been adopted in the project that in the process of securing routing protocols in packet-switched networks and in the network environment created within different technologies (EZChip, XEN, NetFPGA) route (path) determination will be carried out by Quagga software routing suite [3] and the OSPFv3 protocol [4]. Works on the specification and the implementation of routing for RI IPv6 QoS will be concurrently accompanied by research investigations on new multicast routing algorithms. These algorithms could form a base for new routing protocols that would be used in future networks. A search for effective routing algorithms, however, is not possible without appropriate platform that would enable to compare different algorithms in identical network conditions. This, in turn, will make it possible to reliably and objectively evaluate the practicality and usefulness of algorithms. Currently, a possibility of an introduction of a protocol that would ensure multicast routing is being considered in the project. Within the design phase works it has been adopted that the OSPFv3 protocol implemented in the programmable QUAGGA router will be responsible for a determination of routes for packets, both within RI and between RI. There is currently a discussion going on this issue aimed at working out an appropriate implementation of the foreseen multicast routing protocol. Works on the specification and the implementation of protocols and the routing mechanisms for RI IPv6 QoS are concurrently accompanied with studies on new multicast routing algorithms that could be used in IPv6 QoS networks [5]. Multicast routing algorithms can be treated as a graph algorithms because they frequently used graph representation of the network topology.

Any study of graph algorithms must be accompanied with a requirement to provide a platform for performing simulation experiments. Beside an analytical description of algorithms to be made, an appropriate computer program that makes experiments automatic, and makes obtaining of statistically attractive battery of results possible, is necessary to be implemented. In the case of certain types of research studies it is possible to implement a ready-made software available off the shelf that, after an accurate manual configuration, may bring desired results without the necessity of one’s own implementation of the algorithms under scrutiny [13]. If, however, the number of algorithms to be investigated, as well as their diversity, is very high, some tailoring in implementation of a ready-made computational system may prove to be required [6]. This is evidently so with the case of any study on diversified graph algorithms and, in particular, in the case of multi-criteria algorithms, which is necessitated by the need to apply innovative and not easily available solutions. In addition, the software to be used has to provide an option for an immediate adaptation to new working conditions because any researcher is running a risk of a necessity to abandon a given experimental path and to adopt a new one. Within the research studies on multicast routing, a platform for the study of graph algorithms has been created and has been then dynamically developed. The main task of the platform is to handle the aforementioned difficulties, while its main features include operability within different and varied topologies, mechanisms that enable quick exchange of algorithms that are being used in a given experimental context, and an flexible module for a presentation of obtained results. What distinguishes our platform from other "business" technology solutions is that it does not need to maintain constant and reliable access to a service (installing the program onto specialized servers is not a requirement). In addition, it is possible to adopt a pragmatic assumption that the prospective user has the required knowledge and technical abilities to prepare, or modify, input data on his own, or to manually process or evaluate the results generated by the platform. This being given, it is possible to concentrate primarily on the quality and versatility of computational modules which are critical for a dynamic development of a research project. The process of generating a network topology is usually combined with the use of a topology generator, i.e. BRITE [7], or the generator developed by Ellen Zegura's team [8]. An external computer program, such as for example Gnuplot, is also responsible for the presentation of results [11].

The work is divided into 4 chapters. Chapter 2 presents the architecture of the platform. Chapter 3 discusses the method for conducting a simulation experiment with the application of the platform. Chapter 4 sums up the considerations presented in the work.

2. DESCRIPTION OF THE PLATFORM

All functions and objects of the platform are divided into categories (further on called layers). As this is a classical division for such a computer program, individual layers are described only briefly here, while particular attention is given to their usability relevant to the research studies in question.

- Data access layer: The layer gives access to different data sources that include primarily graphs and archived research results. Its main purpose it to enable reading the topologies stored in the data base, or to retrieve output files of generators, i.e. BRITE [7]. This layer provides a simplified description of objects to be processed by algorithms, i.e. graphs, nodes and edges. It describes object in the form in which they are stored in files or in the data base, i.e. paying special attention to avoid redundancy. The reason for taking into consideration the retrieval of data from different sources is both the specificity of network topology generators and the elasticity of a research study to be performed. Network topology generators record generated topologies usually in text files, whereas in the case of research studies repeated in a large group of networks (graphs), a convenient solution to do that is to store network topologies in a data base.

- Model: the model involves a complex description of objects processed by algorithms. It is with objects of his layer (and not with objects from the data access layer) that algorithms work with. The model layer also includes additional information on objects, i.e. redundant information on the neighbourhood of nodes that improve efficiency of the algorithms, as well as the transparency (hence, the quality and reliability) of the computational code. It may be so that in some particular instances (e.g. operations with very large graphs), a particular implementation will be required that can be less appropriate with the case of work with typical topologies. Therefore, the model has been enhanced with an abstract layer that makes it possible to use different implementations and representations of a graph depending on needs and circumstances.
- Algorithms: Algorithms are the main computational part that includes graph algorithms. The basic categories of the algorithms under investigation by our algorithm team include: path finding algorithms and multicast tree finding algorithms. Individual algorithms can be dependent on one another because some of them are related to and embedded in a multiple run of other algorithms with simultaneous calibration of input parameters, e.g. [12]
- Other tools: This part includes minor objects and functions, i.e. pseudo-random numbers generator, simple statistical functions, etc.
- Remaining modules: The remaining modules of the platform include elements of the functionality of the platform that are less significant from the functional perspective such as, for example, classes for the exceptions management system.

There are general realizations for each layer, e.g. those that concern the core of research investigations, i.e. graphs and graph algorithms. However, for each of subsequent research projects, a set of particular realizations (further on called applications) that are applicable and useful only, or mainly, within the context of a given experiment, is also created. Most frequently these include models for intermediate and ultimate results, as well as operations of their storage in data sources and for statistical analysis. With the addition of a new successive application it may become apparent that part of their particular realizations can be applicable for a repeat use procedure and a generalization can be introduced to the system. A generalization is based on a removal of the dependence between the program fragment under scrutiny and the application in which it has been originated so that it is possible to transfer it to the general realization and reuse it with other applications. Hence, the pool of commonly-applicable realizations of individual functions and objects is enlarged gradually and, consequently, the whole of the platform develops. This is a particularly essential property of the platform because it makes it adequate and appropriate for any research and any academic environment and for carrying out different internal projects in particular. This modularity makes it possible for researchers to reuse once created modules many times. On the other hand, the separation of the general realization from particular realizations of individual projects enables the researcher to maintain a distinct delimitation between proven and reliable modules and particular modules of individual realizations that are not intended for reuse. An important advantage that stems from the adopted approach is the fact that the applications involved do not depend on one another. If, however, any element of the realization of a given application proves to be of advantage for another application, it is transferred to the general realization thus securing mutual independence of applications. This operation is very useful, for example, in a situation where, after termination of a research project, we want some of the applications to be removed. If they were embedded in other applications, their removal would be much hampered by their mutual interrelationship.

Moreover, any modification in general realizations that may ensue (though one of the designed assumptions is to minimize the risk of the occurrence of such changes) potentially influences all applications. Consequently, such a change has to be taken into consideration in all modules on which it exerts some influence, so, generally, the drill is to maintain as few modules of an application as it is possible. In practice, it is sufficient to have two or three modules for each research iteration.

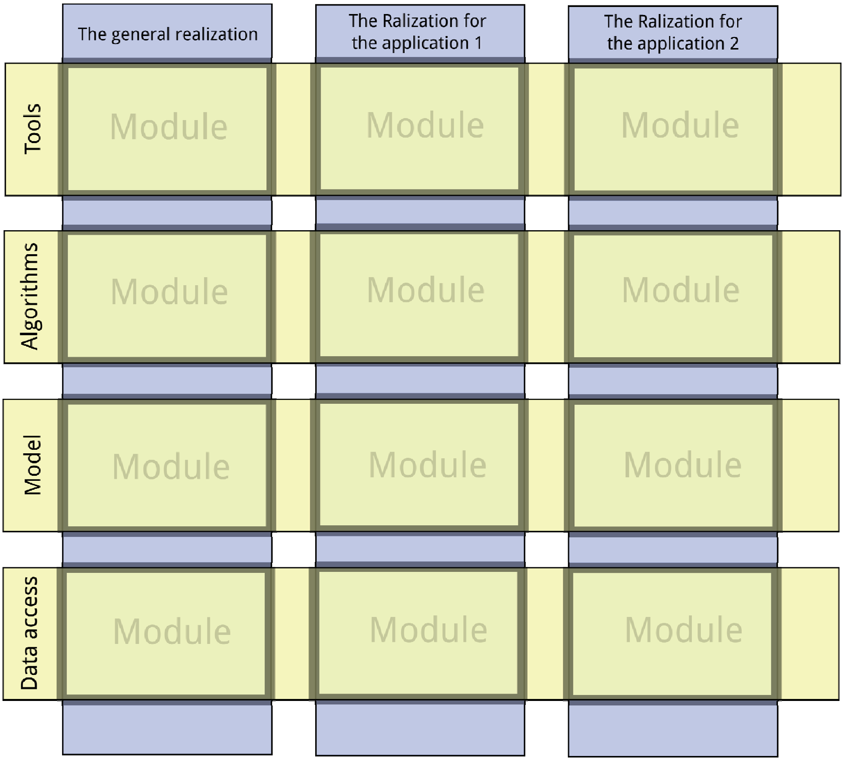

The general realization
The Ralization for the application 1
The Ralization for the application 2
Tools
Algorithms
Model
Data access
Module

Fig. 1. The platform's architecture

Figure 1 shows the architecture described above in terms of its general form. Its purpose is to highlight the existence of the two following planes: layers and realizations that create space for modules. Thanks to the above, each module has its own place both in terms of the function it performs and the context within which it has originated.

Particular layers will be described in detail within the context of their part in the general realization or in a realization of a particular application

2.1. DATA ACCESS LAYER

The layer is composed of two basic parts: object mapping layer (e.g. of graphs or their nodes and edges) and the part responsible for storing this mapping in different data sources. DTO's, *data transfer objects,* are to be particularly considered because of the fact that most of them (e.g. graphs) have their counterparts in the model layer. This duplication results from the fact that the amount of stored information in the data access layer is minimized and redundancy, that can be of some use in computational modules, is to be avoided. DTO objects are only to map the represented elements of the system in an unequivocal way, hence their maximum simplicity is required. The data access layer includes certain improvements that enable to retrieve particular DTO objects from different data sources, but also to record them permanently. Reading is usually necessitated when calculations on a given set of topologies are to be performed, whereas recording is mainly performed in the case of saving and keeping the results of calculations for their future analysis. To perform recordings and readings of DTO objects the DAO (*data access object*) design pattern is used. It provides flexibility in the choice of a data source with a simultaneous provision of a homogenous method for the representation of objects to be processed in calculations. This is particularly important due to the variety of sources mentioned earlier. It is much easier and more convenient to make the data access layer more elastic than to make stored data uniform.

2.2. MODEL

This layer includes a description of the model for calculations. This is where graphs with their nodes and edges are mapped. Objects of the model, as opposed to simplified DTO counterparts, also include additional information, e.g. it is possible to provide a description of the adjacency of nodes in a graph in a number of ways simultaneously (if a given computational module can take advantage of this option). These objects are related to additional operations such as retrieval of neighbours of a given node (this operation would be utterly irrelevant in the data access layer). Apart from graphs, the model layer also includes sub-graphs – used only for calculations and thus not being currently subjected to permanent storage. They include paths as the results of unicast routing and trees as the result of multicast routing. Sub-graphs differ from graphs in that they always ”remember” their parent graph, i.e. the graph on the basis of which they have been created. One of the advantages resulting from this operation is a possibility to create trees out of paths (useful in, for example, the MLRA algorithm[12]), without an excessive complication in the implementation of graph mapping, since such paths are determined within the space of nodes and edges of the same graph.

2.3. ALGORITHMS

This is the most complex layer in which all computational procedures are focused. In the general realization these include primarily path finding algorithms and tree finding algorithms for graphs. In the realizations of particular applications, this layer also includes implementations of statistical processing of results and operations that prepare them to be presented in charts.

2.4. TOOLS

The most essential element in the layer is the pseudo-random number generator. There are many approaches to the problem of generating qualitatively good sequences of random numbers and many reliable algorithms that perform this task. Thus, the pseudo-random number generator has to be addressed with a particular attention and that is why it has a separate place in the platform [9]. Currently, a relatively simple realization, based on a linear feedback shift register, is used. There is, however, a possibility of introducing other implementations in the future, for example, in order to examine their influence on individual research studies. Other elements of the layer include auxiliary classes, e.g. Logger.

3. EXAMPLE OF AN EXPERIMENT IN THE PLATFORM

A typical example of the use of the platform is a run of calculations for the purposes of a research article. With reference to research studies performed by our team it is usually a comparison of different graph algorithms for routing in packet networks. Virtually from the beginning of the development of the platform, implementations of basic path finding algorithms in a graph, such as the Dijkstra’s algorithm, are available in the realization of the general layer of algorithms [10]. Other, more complex, algorithms that make use of the basic algorithms (e.g. in approximation procedures) have been also added to the layer [12]. For the purposes of the experiment, an application module was added to the platform, in which the procedure of an experiment aimed at generating the desired results had been implemented. At this stage, it was only possible to use already existing elements of the system such as the implementations of graph algorithms. Due to the necessity of a calibration of the system, the investigations were repeatedly run with different parameters applied, which necessitated automation of the presentation of results, and which resulted in the appearance of appropriate algorithms that generated code from the Gnuplot program in the realization of the application [11].

Having performed the experiment, it was sufficient to run the graph generator without a necessity to process additionally the data to be presented. Thanks to such an organization of the software architecture it was possible to combine the use of general implementations of graph algorithms with the code specific for a given project that in other applications might have been proved to be completely inappropriate. However, in the realization of this application, a program fragment that could be generalized appeared, i.e. the module that generated graphs on the basis of raw input data. This fragment was "purified" from dependencies from other parts of the application and then transferred to the general realization of the platform. It is thanks to this operation that successive applications will include a possibility of using the module for drawing graphs, which, in consequence, will allow the researcher to focus even more on the research aspects without the necessity to spend time on mechanical and repeatable operations.

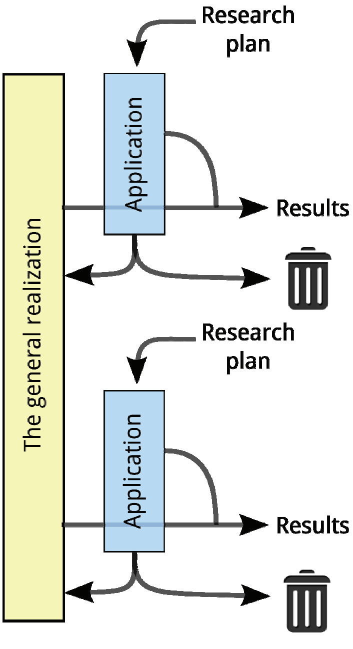


Fig. 2. The platform's lifecycle in the context of the subsequent research projects

Figure 2 presents the cycle of execution for successive research studies and experiments within the context of future development of the platform. One can easily imagine the following decrease in time consumption and the ensuing successive research projects. It is noticeable that after the duration of each of them, the forked arrow denotes the implications of a decision which part of a new implementation is to be incorporated permanently into the platform. The remaining part of the code is irrevocably rejected in order to minimize the risk of the occurrence of dependencies hampering further development.

4. CONCLUSIONS

Creation of a computer program that would exclusively serve purposes of research studies may, under certain circumstances, require a slightly different approach than with the case of utility programs or business platforms. Frequent changes in directions of investigations, numerous experimental paths in the development of a program and the shift of the emphasis to the quality rather than the program itself, can, in consequence, lead to a situation where the software becomes another project problem rather than being a useful and supportive tool. It is necessary then to ensure the platform to be resistant to frequent changes and, at the same time, to have a stable and reliable base safeguarding high quality of results and the accuracy of experimental procedures. It is recommended to reuse repeatable procedures minding at the same time not to excess its complexity or capacity by an introduction of too many generalizations. The architecture proposed in the article has been created with the aim to meet these requirements and it is plain to see that this approach is fit for purpose.

The authors have carried out a large number of simulation experiments with the application of the platform. Currently, work is being done on the development of a multi-criteria multicast routing algorithm based on the Lagrangian relaxation. In the future, the algorithm that is just being developed may be used in a new multicast routing protocol.

ACKNOWLEDGEMENTS

Publication partly supported by European Union within the European Funds 2007–2013, program no. POIG.01.01.02-00-045/09-00 “Inżynieria Internetu Przyszłości”.

REFERENCES

[1] *Projekt Inżynieria Internetu Przyszłości* – https://www.iip.net.pl/project .

[2] *Network Virtualization* – http://www.arl.wustl.edu/netv/main.html .

[3] *Quagga documentation* – http://www.quagga.net/docs/docs-info.php .

[4] *OSPF for IPv6 – RFC 5340* – http://www.rfc-editor.org/rfc/rfc5340.txt .

[5] *Internet Protocol, Version 6 (IPv6) Specification – RFC 2460* – ftp://ftp.rfc-editor.org/in-notes/rfc2460.txt .

[6] STACHOWIAK K., ZWIERZYKOWSKI P., *The Platform of Efficiency Evaluation for Multicast Routing Algorithms.* III International Interdisciplinary Technical Conference of Young Scientists 2010, Poznań, Poland, pp. 250–254.

[7] MEDINA A., LAKHINA A., MATTA I., BEYERS J., *BRITE: Universal Topology Generation from a User's Perspective.*

[8] ZEGURA E.W., CALVERT K. L., BHATTACHARJEE S., *How to Model an Internetwork,* Proceedings of IEEE Infocom '96, San Francisco, CA.

[9] WEGNER B. *Simulation research of the advanced telecommunication systems on the call level, M. Sc. thesis, 2006, Poznan University of Technology.*

[10] DIJKSTRA E.W., *A note on two problems in connexion with graphs*, 1959, Numerische Mathematik, Vol. 1, 269-271.

[11] WILLIAMS T. KELLEY C., et al., *Gnuplot 4.4: an interactive plotting program*.

[12] PIECHOWIAK M., *Badania algorytmów heurystycznych dla połączeń rozgałęźnych w sieciach pakietowych,* PhD *thesis, 2010, Poznan University of Technology.*

[13] *Sinalgo – Simulator for Network Algorithms* - http://disco.ethz.ch/projects/sinalgo/index.html .

ATTA, metaheuristics strategy,
heuristic decision-making system, clustering data, data mining

Arkadiusz LEWICKI*

GREEDY STRATEGY FOR HIERARCHICAL CLUSTERING OF DATA SETS BASED ON ANT COLONY ALGORITHM

The work presented a new strategy for streamlining the process of clustering data with regard to the effect of similarity between the clustered objects in the process of creating groups. It provides for the use of local fine-tuning of the parametric error functions, but taking into account the actual degree of order of objects on the grid. Obtained by the author results show an improvement in the quality of clustering obtained for the original, which is an algorithm ATTA.

1. INTRODUCTION

The explosion of data that are processed by today's network information systems and their loose structure involved with the rapid development of computer technology produces constantly a rise of demand for more efficient data clustering algorithms requiring not only the initial knowledge, but also allowing partition of complex data both symbolic and numerical regardless of their level of disturbances. The process of extraction of data groups, where similar components are to be placed close each other is a difficult combinatorial problem. The complexity of operations for partitioning n objects into m classes in an available data set and their direct comparison is determined by the Stirling number of the second kind expressed by formula (1).

$$S_n^{(m)} = \frac{1}{m!}\sum_{k=0}^{m}(-1)^{m-k}\binom{m}{k}k^n \tag{1}$$

* University of Information Technology and Management in Rzeszów
e-mail: alewicki@wsiz.rzeszow.pl

Therefore, for years, the implementation of solutions of the clustering problem is a subject for studies of many scientists in the field of statistics, databases and learning systems in such important areas as the analysis of geospatial data, the real estate market analysis, web server log analysis, and DNA sequence analysis. Classical algorithms [1] proposed in those areas are inefficient and have too many limitations related to incomplete and imprecise information about the objects being processed, a lack of possibilities for measuring features of tested objects as well as interpretation of results of grouping made by classical numerical algorithms. In this case, solutions of computational intelligence become more popular. However, they have also disadvantages. They depend on the size of processed data sets, what plays an important role in the process of memory allocation. In case of cluster analysis and self-learning neural networks used for this purpose, creation of a functional model requires a well prepared structure and learning error determined during the training process cannot be used to assess the network, because its value can be reduced to any low level by extension of the hidden layer. Increasing a number of hidden neurons causes increasing a number of parameters estimated during the learning process, what has very often a significant impact on the duration of the learning process. Meanwhile, the most serious consequence of using too complex network structures is that the network loses the ability to generalize. Generalization is expressed if the network has the ability to give the correct response for input data not presented earlier in the learning process. Genetic algorithms used often for this purpose are also too slow. They require many iterations and coding, as well as the knowledge of the criterion for new solutions. Other solutions like discriminant functions, decision trees, rule-based systems, rough sets, taxonomic methods, methods of reducing the dimension of a data space, or graphical methods are often insensitive to inconsistent data and a computation time for huge data sets is unsatisfactory. Much better results in this field are expected in case of methods based on the mechanism of autocatalytic social intelligence, which has been successfully implemented by the author to solve combinatorial problems of different types [2, 3, 4, 5]. Algorithms developed until now in this field, founded on the basic Deneubourg model like algorithms: SACA [6, 7, 8, 16], ATTA [8, 9, 10, 11, 13], AntClass [8, 10, 12, 13], ACLUSTER [8, 10, 13, 14, 15] as well as the de Castro algorithm mimic ant behavior using a toroidal grid, on which objects subjected to clustering are placed. Their goal is the distribution of objects in such a way that objects belonging to the same class are placed in the close neighborhood, while dissimilar objects in the feature space are distant also on two-dimensional grid. However, these algorithms are currently characterized by either slow convergence or a lack of stability resulting from the fact that agents do not stop the process of grouping objects, even if they have already reached the optimal allocation, which consequently leads to re-destruction of groups. To avoid this problem and to improve significantly the results, the author of this work propose an algorithm based on a new strategy for the selection of a value of a parameter constituting the relative importance of the difference in similarity between an object moved or claimed by the agent and the objects in its neighborhood, which directly affects the decision of the ants to pick and drop the object.

2. THE APPROACH

Experiments carried out by the author of the study have shown that the major advantage of the most frequently implemented ant clustering algorithm based on the concept given by Handl and Knowles [6] – the ATTA algorithm – is a lack of necessity to predetermine the total number of clusters searched. However, this algorithm has some imperfections determined. One of imperfections is that there is often a smaller number of groups after the grouping process than it should be, especially, for data with a big number of small groups. Another important factor is a lack of stability of the obtained solutions. It causes that that in case of small clusters, they can be destroyed while calculations are continued, even if they have been correctly identified. This is because in the initial stage of computation groups are formed in places of local clusters of similar objects and next, unfortunately, the process of linking these groups of objects at the global level is extremely slow. The probability of dropping and picking x and y depends on the function of local density, expressed by formula (2):

$$\phi_\chi(i) = \frac{1}{\delta^2} \sum_{y \in N_\chi(i)} \left(1 - \frac{\| x - y \|}{\alpha}\right) \tag{2}$$

where α is the scaling factor of the radius of perception of ants during execution of the algorithm. The large value of this parameter reduces the importance of similarities between objects in the process of forming clusters, which leads to formation of large clusters comprising dissimilar elements. On the other hand, too small value significantly impedes the creation of groups because even small differences between objects prevent their placing side by side. To tune this parameter, developers have applied the method of auto-tuning its value, which is determined individually for each thread on the basis of a number f of failed operations of dropping the object during last nstep steps. This value is calculated from formula (3):

$$\alpha_N = \begin{cases} \alpha - 0.01, & when \quad \eta > 0.99, \\ \alpha + 0.01, & when \quad \eta \le 0.99, \end{cases} \tag{3}$$

where η is:

$$\eta = \frac{f}{n_{step}} \tag{4}$$

for $n_{\text{step}} = 100$. Moreover, a larger radius of perception means more computational cost, and it impedes formation of clusters in the initial stage of execution of the algorithm. In the basic algorithm, a radius is increased linearly within the range from 1 to 5, and the related factor δ does not change because, according to author (6), its growth causes the deterioration of the spatial separation of clusters. Our study show not only the possibility of modifying this value depending on the state of the current neighborhood size δ increased in consecutive iterations of the algorithm, but also its impact on increasing the speed of the operation of grouping a greater number of objects. Thus, modification of the local error function, proposed in the publication, associated with the operation carried out to move the object, concerns the density function shown in formula (5):

$$\phi_\chi(i) = \begin{cases} \frac{\vartheta}{\delta^2} \sum_{y \in N_\chi(i)} (1 - \frac{\| x - y \|}{\alpha}), \; when \;\; \phi_\chi(i) > 0 \wedge \forall_y (1 - \frac{\| x - y \|}{\alpha}) > 0 \\ 0, \qquad\qquad otherwise \end{cases} \tag{5}$$

where ϑ is the value determined in the initialization process in the proposed range (from 1 to 5), and δ represents the actual area of the neighborhood, which is increased in consecutive runs of the loop, together with increasing a radius of perception. A factor introduced in this way causes that the sum of distances between objects is less important if the radius of perception of agents is larger, and dependent on it, the size of the neighborhood taken into account in executed calculations.

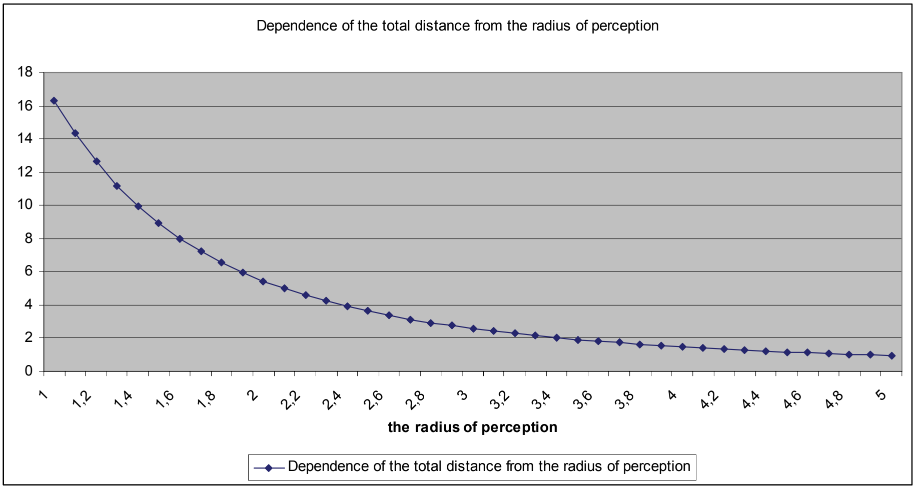


Fig. 1. Dependence of the total distance from the radius of perception

Another change in the proposed approach should be a new rule of tuning coefficient α corresponding to the scaling function of the similarity between the tested objects. This parameter plays a key role in execution of the ATTA algorithm since it determines the relative importance of the difference in the likeness of the object transferred or claimed by an ant to objects in its environment, which corresponds directly to the agent's decisions to pick or leave it. Its right choice depends significantly on the input, which led the author to a new approach to the process of fine tuning its value during execution of the algorithm. It is associated with the rule (6):

$$\alpha_N = \begin{cases} \alpha - \vartheta, & when\ p \le 0, \\ \alpha + \vartheta, & when\ p > q_0, \\ \alpha, & otherwise \end{cases} \tag{6}$$

where q_0 is a random value from the interval [0,1], θ is a constant in the algorithm, associated with a starting value of a radius of perception, based on the similarity of the studied area, while p is calculated from formula (7):

$$p = \frac{\phi_\chi(i) - \phi_\chi(i-1)}{\phi_\chi(i)} \tag{7}$$

for $\phi_\chi(i) - \phi_\chi(i-1)$ of the difference between the current and obtained from the previous step values of the local error function.

3. EXPERIMENTS

A quality of the proposed approach has been determined on three data sets for which the optimal results were known, allowing estimation of the quality of results using external measures of the quality. These collections consisted of 232, 101 and 483 objects, respectively, with a particular emphasis on a diversity of attributes that describe them. Three factors: rand, Dunn and *F*-measure indexes were used to assess the quality of the resulting distribution. The first index is one of the key indicators of a quality of clustering the set *S* = {*S*1,. . . , *Sn*} obtained as a result of execution of grouping algorithm with division *P* and calculated according to formula 8:

$$R = \frac{n1 + n4}{n1 + n2 + n3 + n4} \tag{8}$$

where $n1$ is a number of pairs of objects belonging to the same cluster in divisions *S* and *P*, $n2$ is a number of pairs which belongs to the same cluster in division *S*, but in different clusters in division *P*, $n3$ is a number of objects belonging to the same cluster in division *P*, but in different clusters in division *S*, and $n4$ is a number of objects belonging to different clusters in both division *S* and division *P*.

The second proposed indicator is the Dunn index. This is a relative criterion of assessment of grouping and it consists in execution of the algorithm with different values of parameters for given input data. For each result, a value of the selected index is calculated. It enables us to choose the most appropriate division for an input data set grouped (see formula 9):

$$D_n = \min_{i=1,..,k}\left\{ \min_{j=i+1,..,k} \left(\frac{\| x - y \|}{\max_{m=1,..,k} \max\limits_{x,y \notin C} d(x,y)} \right) \right\} \tag{9}$$

where $\max\limits_{x,y \in C} \max d(x,y)$ is the diameter of the cluster *C*.

The index takes larger values if there exists in a data set groups distinctly drifted apart with small diameters, i.e., with small differences between objects inside clusters. The last indicator used was the *F*-measure defined as (10):

$$F = \sum_{i=1}^{k} \frac{|C_i|}{n} \max_{j=1,..,m} F_{ij} \tag{10}$$

where *n* is the number of objects, and F_{ij} is defined by formula (11):

$$F_{ij} = \frac{2}{\dfrac{1}{p(i,j)} + \dfrac{1}{n(i,j)}} \tag{11}$$

where $p(i, j)$ and $q(i, j)$ are quantities calculated for each cluster C_j of resulting division with respect to each cluster C_i of known division.

Experiments were carried out for random initial values of the designed system, which next were tuned to optimal values in the evaluation process. In this way, a number of ants has been set for 30 for each experiment which was repeated 100 times. Speed of ants (a number of grid cells used to move an agent in one step) has been also tuned experimentally for 24.9. Results of experiments obtained for each examined set are shown in Figures 2, 3 and 4.

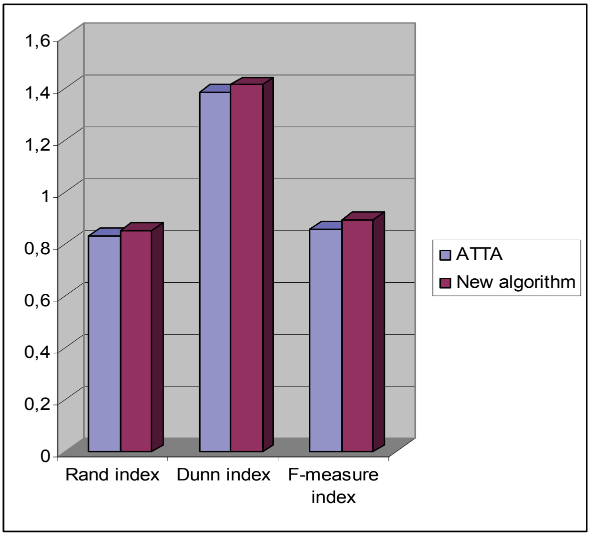


Fig. 2. The best result obtained for a set of 232 objects described by 16 real value attributes

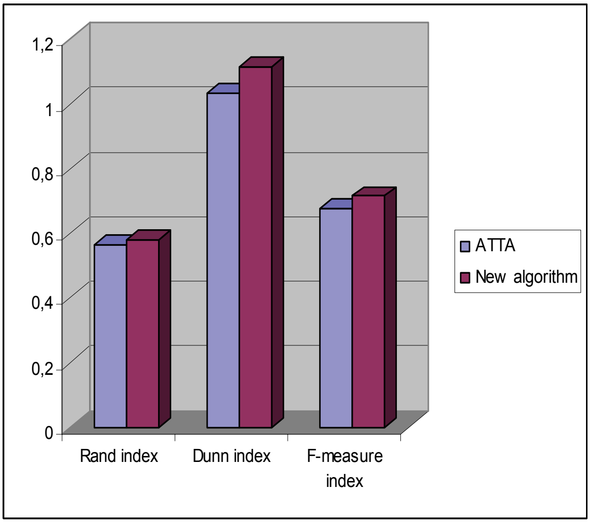


Fig.3. The best result obtained for a set of 101 objects described by 15 integer attributes

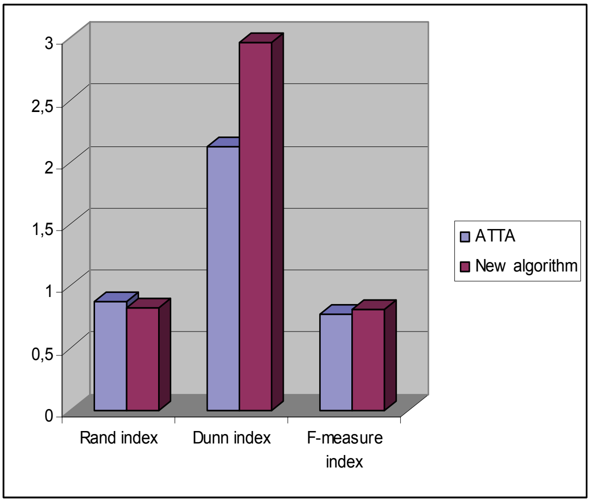


Fig. 4. The best result obtained for a set of 483 objects described by 11 attributes of numeric character

The following conclusions come from experiments carried out by as. Modifications made by us have a quite big impact on the whole grouping process. A selected strategy leads to improving results for simple data sets, in which clusters are distinctly distinguishable. A trend towards creating a greater number of groups of objects, characterized by a strong similarity, can constitute encouragement for an attempt to design an algorithm in which sets treated as a whole are subjected to group.

4. CONCLUSION

Data from A new concept, presented in the publication, of the data aggregation algorithm based on swarm intelligence is an important part of the research on optimization of mechanisms in a very important area of data mining and analysis. The proposed approach is an attempt to improve the algorithm proven many times [6,8,9,10,13], which is the ATTA clustering algorithm. The proposed approach takes into account both changes in the radius of perception of objects searched, accessible in the neighborhood and the impact of similarity between the objects clustered in the process of creating groups. That tactic used taking into consideration the actual degree of the order of objects on the grid has a chance for success. Obtained results reflect on improving a quality of grouping with respect to the initial state.

This proves that the algorithm is suitable for the use in cases when there is no additional knowledge about the objects under consideration in the decision space.

Its result is a map of objects which can be useful for presenting obtained groups of objects.

An important advantage of intelligent systems is the dispersion of the collective and strong parallelism. Ants act simultaneously and independently from each other, usually focusing on a small area of a solution space, so it is important here to exchange an information about the solutions found, to achieve the global optimum. Therefore, developing of methods should be focused on this aspect further. In a current version of the implemented algorithm agents communicate only by modification of the environment, which consists in changing positions of objects. A use of additional mechanisms of communication would be here more advantageous. Therefore, it seems to be advisable to use this area for additional mechanisms.

ACKNOWLEDGEMENTS

The research described in this publication has been supported by the project “The mechanisms of swarm intelligence issues in the discriminant analysis and cluster analysis” of the grant No. 6545/B/T02/2011/40 from the National Science Centre in Poland.

REFERENCES

[1] HORE, P., *Distributed clustering for scaling classic algorithms, Theses and Dissertations*, University of South Florida, 2004.

[2] LEWICKI, A., TADEUSIEWICZ, R., *The recruitment and selection of staff problem with an Ant Colony System*, Backgrounds and Applications 2, Springer-Verlag’s Advances in Intelligent and Soft Computing, 2010.

[3] LEWICKI, A., *Generalized non-extensive thermodynamics to the Ant Colony System*, Information Systems Architecture and Technology, System Analysis Approach to the Design, Control and Decision Support, Wroclaw, 2010.

[4] LEWICKI, A., *Non-Euclidean metric in multi-objective Ant Colony Optimization Algorithms*, Information Systems Architecture and Technology, System Analysis Approach to the Design, Control and Decision Support, Wroclaw, 2010.

[5] LEWICKI, A., TADEUSIEWICZ, R., *An autocatalytic emergence swarm algorithm in the decision-making task of managing the process of creation of intellectual capital*, Springer-Verlag, 2011.

[6] HANDL, J., KNOWLES, J., DORIGO, M., *Ant-based clustering and topographic mapping*, Artif. Life, Vol. 12, No. 1, 2006.

[7] DECASTRO, L., VON ZUBEN, F., *Recent Developments In Biologically Inspired Computing*. Idea Group Publishing, Hershey, PA, USA, 2004.

[8] MOHAMED, O., SIVAKUMAR, R., *Ant-based Clustering Algorithms: A Brief Survey, International Journal of* Computer, Theory and Engineering, Vol. 2, No. 5, October, 2010.

[9] DORIGO, M., DI CARO, G., GAMBARELLA, L., *Ant Algorithms for Discrete Optimization*, Artificial Life, 5(3), 1999.

[10] AZZAG, H., MONMARCH´E N., SLIMANE, M., VENTURINI, G., GUINOT, C., *AntTree: A new model for clustering with artificial ants*, IEEE Congress on Evolutionary Computation, Vol. 4, pp. 2642–2647, Canberra, 2003. IEEE Press.

[11] SCHOLES, S., WILSON, M., SENDOVA-FRANKS, A., MELHUISH, C., *Comparisons in evolution and engineering: The collective intelligence of sorting. Adaptive Behavior – Animals*, Animats, Software Agents, Robots, Adaptive Systems, Vol. 12, No. 3 –4, 2004.

[12] SENDOVA-FRANKS, A., *Brood sorting by ants: two phases and differential diffusion*, Animal Behaviour, 2004.

[13] BORYCZKA, B., *Ant Clustering Algorithm, Intelligent Information Systems*, Kluwer Academic Publishers, 2008.

[14] ABBASS, H., HOAI, N., McKay, R., AntTAG, *A new method to compose computer using colonies of ants*, Proceedings of the IEEE Congress on Evolutianory Computation, Vol. 2, Honolulu 2002.

[15] VIZINE, A., DE CASTRO, L., Hruschka, E., Gudwin R., *Towards improving clustering ants: An adaptive clustering algorithm*, Informatica Journal, Vol. 29, 2005.

[16] OUADFEL, S., BATOUCHE, M., *An Efficient Ant Algorithm for Swarm-based Image Clustering, Journal of Computer Science*, Science Publications, 3(3).

PART 2

MATHEMATICAL MODEL AND ITS APPLICATIONS IN DECISION SUPPORT IN TECHNICAL AND NON-TECHNICAL PLANTS

Social Network, Subgraph Discovery, Link Prediction

Krzysztof JUSZCZYSZYN*, Wojciech FRYS*

THE CHARACTERIZATION OF CHANGES IN STRUCTURAL PATTERNS OF COMPLEX NETWORKS

Social structures built on the basis of records and datalogs of modern information networks are in constant change. We observe various evolutionary patterns corresponding to external events or the evolution of underlying organizations. In this work we present a new approach to the quantifying changes in large social network illustrated by the data from an organizational social network of the Wroclaw University of Technology (6000 employees). We analyse the process of emerging and disappearing of the links for different periods and time windows, discovering new dynamic patterns and carrying on their structural analysis. The observations are used to propose a novel link prediction algorithm, which shows good performance, especially for sparse networks analyzed in short time-scales.

1. INTRODUCTION – SOCIAL NETWORKS

In technology-based networks a relation between two individuals is a result of set of discrete events (like emails, phone calls, blog entries) about which the knowledge is available. Because these events have some distribution, this adds a new dimension to the known problems of network analysis [11]. As shown in [9] for various kinds of human activities related to communication and information technologies, the probability of inter-event times (periods between the events, like sending an email) may be expressed as: $P(t)≈t–α$ where typical values of α are from (1.5, 2.5). This distribution inevitably

* Faculty of Computer Science and Management, Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław,

e-mail: krzysztof.juszczyszyn@pwr.wroc.pl, wojciech.frys@pwr.wroc.pl

results with series of consecutive events ("activity bursts") divided by longer periods of inactivity.

These phenomena have serious consequences when we try to apply the classic structural network analysis (SNA) to the dynamic networks. The most popular approach is to divide the time period under consideration into time windows, then run SNA methods on the windows separately. However, the bursty behaviour of the users (long inactivity periods mixed with the bursts) causes dramatic changes of any measure when switching from one time window to another. There is a trade-off: short windows lead to chaotic changes of network measures, while long windows give us no chance of investigation of network dynamics [13][14]. In order to address this problem, a number of methods, designed to predict changes in the structure of dynamic networks, were proposed [15][16]. The special case is the so-called link prediction problem – the estimation of probability that a link will emerge during the next time window [12].

A broad survey of link prediction methods is presented in [20]. It should be noted that most methods of the link prediction give rather poor results – the best predictors discussed in [12] can identify < 10% of emerging links. It should be emphasised that the networks analysed in [12] were built from arxiv publication record which differs significantly from our test cases (email social networks) presented below. Email networks are highly dynamic in short timescales and – for big networks, the number of disconnected pairs of nodes increases quadratically (the density of real-world networks is small and the graphs are sparse) while the number of links grows only linearly [21].

Basing on our previous experience, which shows that the distribution of subgraphs in complex networks is statistically stable and typical for the considered network even in the face of significant structural changes [17], we claim that it is possible to characterize the network structural changes by statistical data about the evolution of its subgraphs.

In the following sections we propose a method for the description of changes in local connection patterns of complex network, and link prediction algorithm which utilizes these patterns. The approach was evaluated on two email social network datasets which significantly differ in size and dynamics.

2. LOCAL NETWORK TOPOLOGY AND SUBGRAPH MINING

Standard approaches exploiting network analysis by means of listing several common properties, like the degree distribution, clustering, network diameter or average path lengths often fail when applied to complex networks. However, network structures (like social, biological, gene networks) may be investigated with more precise and structure–sensitive methods [1]. During last years we experienced the development of a number of methods investigating complex networks by means of their local structure. The simplest way to characterize the network in the context of local connections is to examine the links between the smallest non-trivial subgraphs, the triads, consisting of three nodes. If we additionally decide to distinguish between the nodes (which is our case, for our network they are corporate email addresses) we get 64 patterns of possible connections between any three identifiable nodes (Fig. 1).

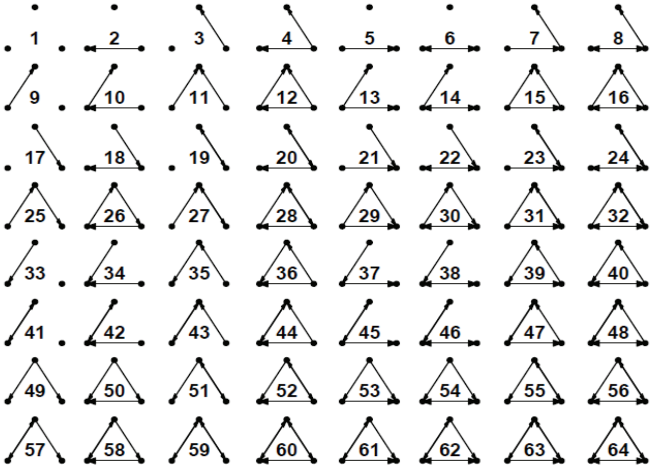


Fig. 1. Three-node triads in a directed labelled graph

Please note the triad ID (the number inside the picture of the subgraph) in Fig. 1, as it will be used further on in this work. The basic method utilizing such subgraphs is the well-knows triad census, which is enumeration of all triads in the network and allows to reason about the functional connection patterns of the nodes [2], [17], [18]. Another is so-called *motif analysis* which aims to characterize the network by the difference between its structures and an ensemble of random networks of the same size and degree distribution [1], [3], [4], [5]. Although the global topological organization of networks is well understood, their local structural organization is still not clear. At the smallest scale, network motifs have been suggested to be the functional building blocks of network biology [6], [7], [8].

We introduce a *Triad Transition Matrix* as an elementary structure for the characterization of local topology changes. It uses data from the history of the network (recorded during past time windows) to derive the probabilities of transitions between triads (patterns of local connections) of chosen network nodes.

The TTM is a matrix of size $g_\Delta \times g_\Delta$, where g_Δ is the number of subgraphs. As we distinguish between the nodes, and for three nodes the connections A-B-C and A-C-B do not constitute the same configuration, the number of directed triads is $g_\Delta = 64$ (see Fig.1). The values of TTM entries are defined as follows:

$$TTM_t(i,j) = P(g_i[t] \rightarrow g_j[\ t+1]) \qquad \mathbf{(1)}$$

TTM$_t$ (*i*,*j*) is the probability (estimated from full triad enumeration for networks created from data gathered in [*t*] and [*t*+1] time windows), that a connection pattern (triad) g_i found during [*t*] will transit into $\mathrm{g_j}$ during [*t*+1]. The sum of values in TTM row is 1. Our goal was to check if the stability of motif distribution is followed by the distinguishable evolutionary patterns of the network triads. For the experiments with the TTM we have chosen the dataset containing the email social network data extracted from Wroclaw University of Technology (WUT) mail server logs. The entire set contains data from 605 days of the operation, with 5834 active email addresses. The networks used in our experiments were created from data for 1-day, 3-days and 7-days time windows. It should be noted that the email social networks undergo rapid structural changes when investigated in short time periods. In our experiment the stability of a single link is quite low: in WUT dataset 53% for 7-day time windows and only 42% for 1-day time windows (which means that on average 42% of the links will still be present in the next time window). On the Fig. 3 we see significant changes in network size which correspond to the pattern of activity at WUT (the network degenerates during weekends, which implies the periodical changes, clearly visible for 1-day windows and influencing longer windows as well – Fig. 2).

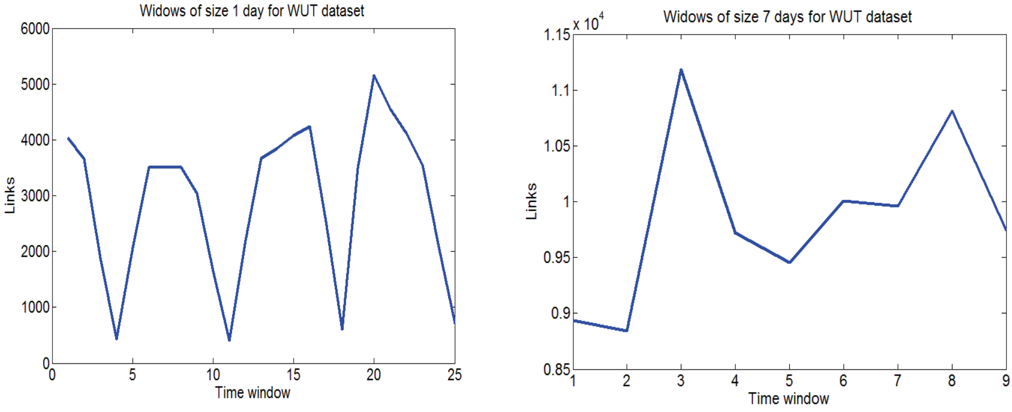


Fig. 2. Network size for time windows of different size for WUT dataset

3. TTM ANALYSIS AND LINK PREDICTION

In Fig. 4 the mean-value TTM derived from 12 time windows for Enron dataset is presented. Despite the changes in network size (Fig. 2), all TTMs computed for neighbouring time windows showed similar values, with standard deviation less than 10%. We may notice that the distribution of transition probabilities is not flat, and there are distinctive patterns (the coordinates of TTM correspond to the triad numbers from the Fig. 1). First of all, the high value of TTM(1,1) reflects the fact that the network is sparse (link density below 1%) which means that most of the possible triads contain no edges. As the result most of the "empty" triads always remain in this state, which gives us a relatively high value of TTM(1,1). Similarly, the full triad (#64 from Fig. 1, containing 6 directed links) is quite stable with TTM(64,64) above 0.3. We should also note the high values in the first column of the TTM. This means that when it comes to disappearing of the links, the probability of resetting the entire triad to zero-connection state is relatively high.

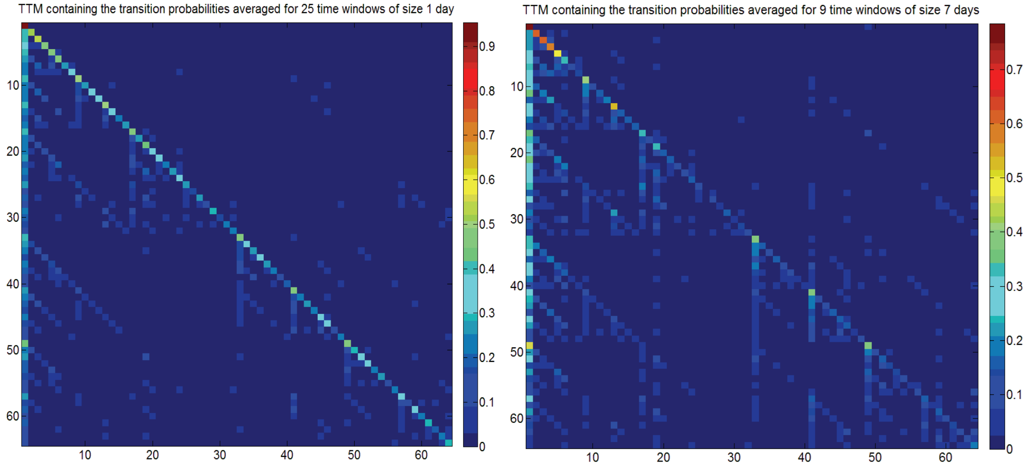


Fig. 3. TTM containing the transition probabilities averaged for all time windows for WUT dataset

From the other hand, it is also visible, that the values on the diagonal of TTM are bigger than values in their neighbourhood, which shows that the already-formed triads tend (in general) to stay in their current state.The last important observation is that some triads are special, they show clearly bigger values in their columns of TTM, which means that they are "sinks" of the connection evolution patterns. For 1-day window the network is more stable – it is visible at the diagonals of averaged TTMs – the probabilities at the diagonal are higher than in other fields of TTMs. At this point we propose the application of the discovered network local evolutionary patterns (TTMs) to the link prediction problem.

The evaluation of link prediction methods in our experiments was based on the principles proposed in [12]. It was assumed there, that all the predictors assign a predicted connection weight *score*(*x, y*) to unlinked pairs of nodes ⟨*x, y*⟩, based on the input graph, and then produce a ranked node pair list *L* in decreasing order of *score*(*x, y*), whose values are treated as proportional to the estimated probability of forming a new link between *x* and *y*. In this way each link predictor outputs a ranked list of node pairs which would eventually form predicted new links. From this list (sorted in decreasing values of scores) the set of first *n* entries is taken, then the size of its intersection with the set of new links (of the same size *n*) is computed. The percentage of the links from the predicted set, which are also present in the set of new links, is the prediction accuracy.

In our link prediction algorithm (from this point called *TTM-predictor*) we assume the following: for the social network graph $G=\langle E, V\rangle$ and a node pair $p = \langle x, y\rangle$; $x, y \in V$; $x\neq y$ let us denote the set of all non-empty triads that *p* belongs to as Δ_p (*non-empty* means: containing at least one directed link, but *not* necessarily the link between *x* and *y*). In this way, in the course of prediction, we check only triads which contain at least one link. This is justified by the huge number of empty triads which, according to the values of TTMs gathered in experiments in most cases will remain empty (the TTM entry (1,1) is typically close to 1). In order to avoid the attraction of the prediction results by zeros (disappearing links) we exclude them from analysis. Moreover we consider the prediction of links which are not connected to existing network (by means of having at least one of their endpoints adjacent to any link, $k_{training} = 1$) unjustified.

Assuming that the TTM is known, we want to check the evolution patterns for all the triads in Δ_p for each *p* adjacent to at least one link in existing network.

Let $t\in\Delta_p$ be a single triad containing *p* (one of the 64 considered), and TTM(*t*) – the row of TTM matrix number *t*. The set of 64 values contained in TTM(*t*) (corresponding to transition probabilities of *t*) may be divided into two disjoint sets $\mathrm{TTM}^0(t)$ and $\mathrm{TTM}^1(t)$, where:

- $\mathrm{TTM}^0(t)$ is a the set of TTM entries from the row *t* corresponding to triads in which there is no link between the nodes of *p*.
- $\mathrm{TTM}^1(t)$ is a the set of TTM entries from the row *t* corresponding to triads in which there is a link between the nodes of *p.*

The algorithm for computing the score for node pair *p* – from here on denoted as $score_{TTM}(p)$, given the known *G* (network graph for current time window) and the TTM is as follows. For a given *p*:

```
$score_{TTM}(p)=0$;
Step 1: Determine $\Delta_p$;
Step 2: For each $t\in\Delta_p$ determine $\mathrm{TTM}^0(t)$ and $\mathrm{TTM}^1(t)$;
Step 3: compute:
$scoreTTM(p) = \sum_{\Delta_p}\sum TTM^1(t)$
```

As one can see, the proposed algorithm is a kind of "voting procedure" in which all the triads from Δ_p vote for the existence of link between the nodes in *p* according to the values in their TTM rows. The votes are weighted and the weight of each vote is equal to the probabilities from the TTM. Having computed the link scores we can propose an algorithm of the TTM-predictor.

The TTM prediction algorithm:
For given G, n, and TTM:
Step 1: Determine the set P of node pairs, such that for each $p \in P$:
- there is no link in G between the nodes of p,
- there is at least one link in G adjacent to the nodes in p.

Step 2: For each $p \in P$ compute $score_{TTM}(p)$
Step 3: Create the list L_P of $p \in P$. Sort L_P in decreasing order of $score_{TTM}(p)$
Step 4: Pick the first n elements from L_P which are predicted new links.

For the evaluation of the prediction algorithms the value of n is assumed to be known, however for practical applications it may be estimated with good accuracy from the time series of the numbers of new and disappearing network links once the history of the network is known. It corresponds to the inference of n from the changing number of links in successive time windows – see Fig. 2.

4. TTM PREDICTION – EVALUATION

The evaluation of link prediction methods in our experiment gave the results presented in Fig.6. We have compared the TTM-predictor (TTM) with the two standard methods: preferential attachment (PA) and common neighbours (CN) predictor, defined exactly as in [12] (where CN was one of the best) and, additionally, the simple random predictor (RN, for which the scores are just random values from [0,1]). The experiments were performed for the time windows 1 and 7 days.

For the WUT dataset the performance of TTM is similar for the 7-day time windows (Fig.4), but for narrowing time windows (1-days corresponding to growing sparsity and the variance of the link number) it can be seen that the performance of all predictors is going down (which is expected), however not equally. When the network undergoes periodic changes connected with rapid reduction of link number (Sundays: 1-day windows 4, 11, 18, 25 and their preceding days – weekends) the TTM performance, however reduced, is still at the level higher than 2% (in contrary to PA and CN). In the periods of “network growth” (from Sundays on), unlike the weekends, the non-zero performance of PA and CN appears. This case may be interpreted as the moment in which the rules of PA and CN (that the unlinked nodes show affiliation to hubs or tend to form a link when having a number of common friends) start to “work” again which results in the increase of the number of accurate predictions.

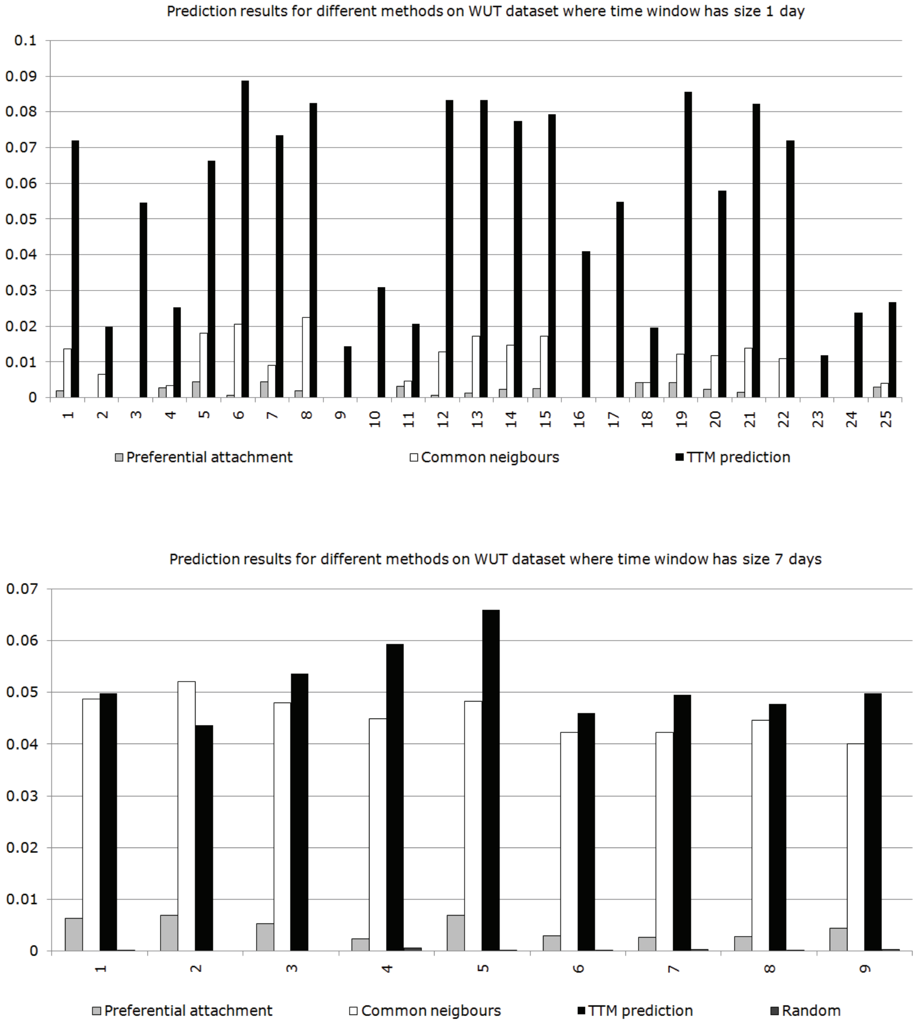


Fig. 4. The performance of TTM-predictor

Summing up, we have checked the performance of the TTM-predictor for two networks (for the second, WUT, for three different timescales) which significantly differ in the size, dynamics and the period covered by their datasets. Its performance was confirmed to be generally better than that of CN and was proved to be relatively immune (in comparison to CN and PA) to the periods when the networks changes its mode of operation which results in rapid structural changes. The observations described in this section suggest the ways of further developing our method.

CONCLUSIONS AND FUTURE WORK

The concept of TTM joins the statistical features of network links with their topological connection patterns. The method, although based on graph analysis, utilizes the inherent network dynamics based on the observations of the recorded network history. For we do not assume any prior knowledge about the nature of relations and network nodes (TTM bases on the structural changes in the network only), this allows also the future classification of the different dynamic networks (social, biological, etc.) according to their local evolutionary schemes expressed by the TTMs.

This work reports the preliminary experiments carried on to check the possibility of modelling network evolution by means of structural changes in its elementary subgraphs. It should be noted that the triad voting scheme applied in the TTM-predictor is one of the simplest possible solutions. The promising results of these experiments open possibilities of further developing of our approach, the most appealing directions are:

1. Including link weight in the analysis; in an e-mail network a link exists as a consequence of sending one or many messages, and in most cases it is far more stable in the second case. This issue will be used to tune our method in the next stage of experiments and should improve the performance of TTM prediction, especially for longer time windows (in which case the incidental communication – links of weight 1 – may be clearly distinguished).

2. Time series analysis of TTM values. From the first experiments we know that the values in TTMs undergo periodic changes which was visible especially for short (1-day) time windows. Accurate estimation of the future TTM values may greatly improve the prediction.

Further experiments on various networked systems of different origin are also planned in order to develop methodologies for modelling the evolution of networks with the dynamic subgraph mining.

ACKNOWLEDGEMENT

The research presented in this work has been partially supported by the European Union within the European Regional Development Fund program no. POIG.01.03.01-00-008/08.

REFERENCES

[1] ITZKOVITZ S., MILO R., KASHTAN N., ZIV G., ALON U., *Subgraphs in random networks,* Physical Review E., 68, 026127, 2003.

[2] JUSZCZYSZYN K., MUSIAŁ K., KAZIENKO P., *Local Topology of Social Network Based on Motif Analysis*, 11th International Conference on Knowledge-Based Intelligent Information & Engineering Systems, KES 2008, Croatia, Springer, LNAI, 2008.

[3] KASHTAN N., S. ITZKOVITZ S., MILO R., ALON U., *Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs*. Bioinformatics, 20 (11), 2004, pp. 1746–1758.

[4] MILO R., SHEN-ORR S., ITZKOVITZ S., KASHTAN N., CHKLOVSKII D., ALON U., *Network motifs: simple building blocks of complex networks,* Science, 298, 2002, pp. 824–827.

[5] MANGAN S. ALON U., *Structure and function of the feedforward loop network motif,* Proc. of the National Academy of Science, USA, 100 (21), 2003, pp. 11980–11985.

[6] MANGAN S., ZASLAVER A. ALON U., *The coherent feedforward loop serves as a sign-sensitive delay element in transcription networks*, J. Molecular Biology, 334, 2003, pp. 197–204.

[7] VAZQUEZ A., DOBRIN R., SERGI D., ECKMANN J.-P., OLTVAI Z.N., BARABASI A.,. *The topological relationship between the large-scale attributes and local interaction patterns of complex networks*, Proc. Natl Acad. Sci. USA 101, 2004, p. 17 940.

[8] YOUNG-HO E., SOOJIN L., HAWOONG J., *Exploring local structural organization of metabolic networks using subgraph patterns*, Journal of Theoretical Biology 241, 2006, pp. 823–829.

[9] BARABÁSI A.-L, *The origin of bursts and heavy tails in humans dynamics*, Nature 435, 2005, p. 207.

[10] GROSS T., SAYAMA H. (Eds.): *Adaptive networks: Theory, models and applications*, Springer: Complexity, Springer-Verlag, Berlin-Heidelberg, 2009.

[11] KLEINBERG J., *The convergence of social and technological networks*, Communications of the ACM, Vol. 51, No.11, 2008, pp. 66–72.

[12] LIEBEN-NOWELL D., KLEINBERG J.M., *The link-prediction problem for social networks*, JASIST (JASIS) 58(7), 2007, pp. 1019–1031.

[13] BRAHA D., BAR-YAM Y., *From Centrality to Temporary Fame: Dynamic Centrality in Complex Networks*, Complexity, Vol. 12 (2), 2006, pp. 59–63.

[14] KEMPE D., KLEINBERG J., KUMAR A., *Connectivity and inference problems for temporal networks*, Journal of Computational System Science, 2002, 64(4), pp. 820–842.

[15] LAHIRI M., TANYA Y. BERGER-WOLF, *Mining Periodic Behavior in Dynamic Social Networks*, ICDM 2008, pp. 373–382.

[16] LISA SINGH, LISE GETOOR: *Increasing the Predictive Power of Affiliation Networks*, IEEE Data Eng. Bull. (DEBU), Vol. 30, No. 2, 2007, pp. 41–50.

[17] JUSZCZYSZYN K, MUSIAL K., KAZIENKO P., GABRYS B., *Temporal Changes in Local Topology of an Email-Based Social Network*, Computing and Informatics 28(6), 2009, pp. 763–779,.

[18] WASSERMAN S., FAUST K., *Social network analysis: Methods and applications*, Cambridge University Press, New York, 1994.

[19] BATAGELJ V., MRVAR A., *A subquadratic triad census algorithm for large sparse networks with small maximum degree*, Social Netw. 23, 2001, pp. 237–243.

[20] GETOOR L., DIEHL C.P., *Link mining: a survey*, ACM SIGKDD Explorations Newslett., Vol. 7, 2005, pp. 3–12.

[21] HUANG Z., LIN D.K.J., *The Time-Series Link Prediction Problem with Applications in Communication Surveillance,* INFORMS Journal on Computing, Vol. 21, No. 2, 2009, pp. 286–303.

Crowd dynamics, Multiagent systems
Pedestrian behavior

Artur KOSZTYŁA*, Jarosław WĄS*

MULTIAGENT SYSTEM FOR CROWD BEHAVIOUR MODELLING

Computer simulations of the crowd dynamics have become popular in recent years. Many of the risks associated with the crowd can be avoided if experts properly use computer simulations of crowd in the process of facilities designing. Another group of application of crowd simulation is entertainment and media industry (movies, computer games etc).

We can distinguish two kind of base models dedicated for crowd simulations: macroscopic approach based on hydrodynamics (no individuality), as well as microscopic models (for instance agent-based models).

The work presents a conceptual platform based on microscopic, multiagent approach, dedicated for crowd behavior modeling in different situations. The platform has been developed and implemented last few years. Each agent possesses a set of individual properties. An agent can communicate with each others. Agents in the model are assigned to different groups: the agents who are in the same group cooperate together, while different groups of agents can compete with another groups. If we use set of simple or more complicated rules of behavior of individual entities, we can observe a very interesting view on crowd motion. The presented tool makes possible to visualize of crowd in 2D, as well as 3D.

1. INTRODUCTION

Continuous growth of computational power in the last years opens new possibilities in many fields of science. One of this fields is crowd dynamics modelling. Over the years scientists developed many kinds of microscopic crowd simulations based on Molecular Dynamics (Social Forces Method), Cellular Automata or Multi-agent approach.

* AGH – University of Science and Technology, Institute of Automatics.

Crowd is defined in literature as “a large group of people who have come together” [6]. According to bibliography we can distinguish different kinds of crowd. For instance gathering is usually a crowd with weak connections between individuals, which met in one place by a chance or for a specified reason like a concert. On the other hand mob is a large crowd, which can be aggressive or panicing [2]. More completed classification of crowds is presented on Fig. 1.

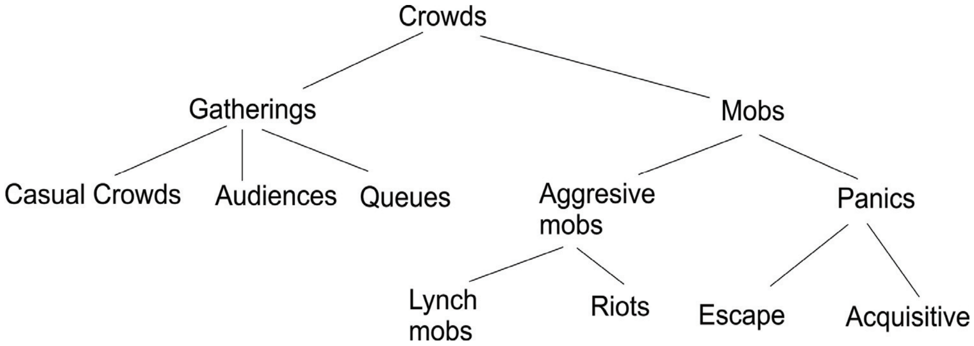


Fig. 1. Classification of crowds according to Forsyth [2]

Actually two approaches in microscopic crowd simulation are very popular [6, 8]. The first one is a continuous based on Molecular Dynamics (Social Forces Model). The second group is based on discrete approach based on Cellular Automata and Multiagent Systems (MAS). Both of them opened new ways and new possibilities in crowd simulations, because particular entity in the system can represent different model of behaviour.

This work describes a system based on Multi-agent theory built for multipurpose crowd behaviour simulations. The system allows to create a 3D rendered animation of crowd, which may be used in a various analyses. Such models find more and more applications: in science, but also in commercial companies designing buildings or in a movie industry.

2. GENERAL ASSUMPTIONS

2.1. SYSTEM

The presented system is built using a combination of MAS (multiagent system) and Cellular Automata. The system is designed to simulate different number of intelligent agents: simulations can be performed with 10, as well as 1000 agents. An important goal was to allow user to create the desired environment, layout of the scene where the simulation is to be processed. The output was designed to be a 3D movie style visualization with good scene representation, animated agents and real looking obstacles.

2.2. AGENT IMPLEMENTATION

Agent implementation used in the system was inspired by an M-Agent theory described in [3] and [5]. Every agent in the system has individual characteristics. Authors put emphasis on different features of an agent such as:

- Autonomic decisions,
- Active communication and cooperation,
- Environment monitoring and reacting to its changes,
- Flexible goal functions.

Formally an agent is defined as:

$$\Lambda = A, S, F \subset S \times a \times S \tag{1}$$

where:

A – set of available actions,

S – set of agents inner states,

F – relation between actions and states, in a state agent has a set of possible actions defined as:

$$(s, a, s_1) \in F \wedge (s, a, s_2) \in F \Rightarrow s_1 = s_2 \tag{2}$$

to keep action – effect relations.

An agent has its own intelligence modelled by a decision tree. Every agent can use a different decision tree. It is based on a simple observation: every human being is different. This is a very effective and intuitive way of personalizing agents. Decisions are made on the base of all the information possessed by the agent and actual goal set by the user.

An agent can possess a better or a worse sight, different average velocity or different criteria in decision making process. Different criteria are reflected in the system.

An agent can be in one of many implemented states as for example: sitting, standing, walking or running. Transitions between states are defined in a state diagram (Fig. 2). Transitions are configurable and they can be extended and modified by the user.

The most important part of an agent is its own goal function. It defines what the agent is supposed to do. The system is designed to allow assigning multiple goal functions to agents, prioritize them statically and dynamically. Such construction gives the user ability to manage agents. Goals defining may also be managed by the agent itself, depending on the information gained from an environment. It can change priorities as well as make some goals disabled for a period of time, if the aims are not currently possible to achieve. Presented system include several goal functions for instance: finding a way to the exit, following the group or risk assessment. Applying of combined goals functions allow to gain complex crowd behaviours. Set of goal functions is kept as a list of them.

To be able to achieve goals in a system all agents during simulation communicate with each other if they are in a close distance. They can exchange learned information about environment, other agents or they common goals. It simulates normal communication between human in a crowd.

Each agent acts in a specific manner shown on Figure 3. An agent first updates its vision of the world (observation), tries to understand and decide whether it can use it in any way. Than the agent observes other agents. Because agents can cooperate or compete it is important to know states and possible actions of other agents in a neighbourhood. An agent in this step can gain new information from other agents or it can share its own knowledge about the world and common goals. Finally, having all the possible information, the agent goes through calculating its goal functions. It can decide to change priorities and temporally deactivate them.

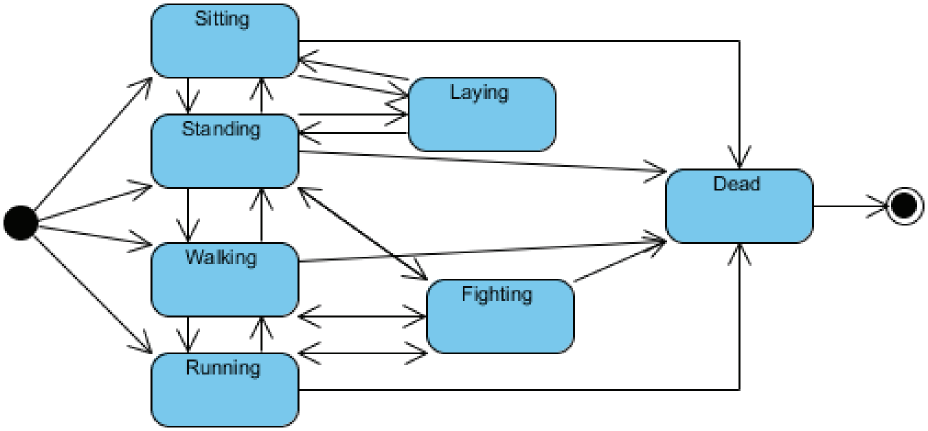


Fig. 2. State diagram of an agent

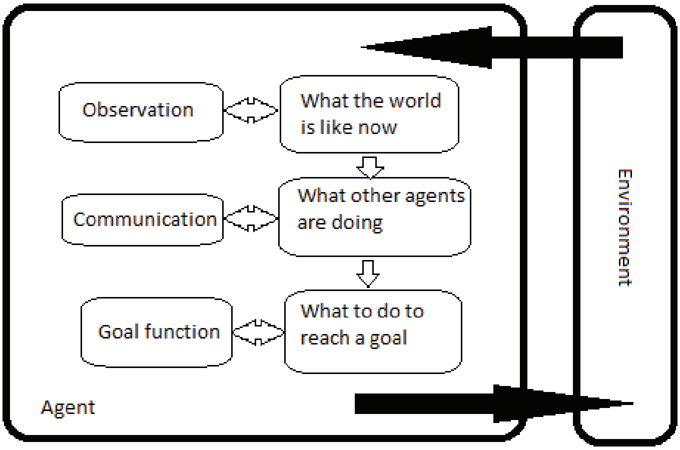

Observation
What the world is like now
Communication
What other agents are doing
Goal function
What to do to reach a goal
Agent
Environment

Fig. 3. Agent reasoning scheme

2.3. SCENE REPRESENTATION

The scene was implemented as a three dimensional Cellular Automaton. Such kind of discretisation of space lead to creating optimised movement algorithms. Our investigation showed that such connection of decision function with rules of inhomogeneous Cellular Automata allows to describes the world of agents in effective and intuitive way.

Introducing 3D CA to the system have opened an easy way to implement multilevel objects: like buildings or complex topography (valleys, hills etc). Furthermore it allows using the system in very specific simulations of multi-storey buildings.

To introduce obstacles into the system, the user puts into the program separate image files representing levels of the scene (Fig 3).

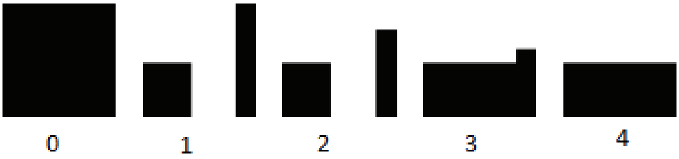


Fig. 3. Image files representing levels 0 to 4 of the scene

The system output is 3D visualization, example view is presented on Figure 6. A view can be displayed online, while the simulation is running. It can also be saved on disk for future analyses or post processing to present the results in more spectacular way.

2.4. MOVEMENT ALGORITHM

The connection of multiagent system with CA gains popularity in the few last years [1], [4], [7], [9]. It has caused speedup of the process of movement calculating and interacion with neighbours.

The system uses movement algorithm based on a field of sight of the agent. Each agent can observe only a piece of the surrounding enviroment. Possible obstacles cause narrowing agent's sight. Figure 4 shows the field of sight of an agent. Green color symbolizes area observable by an agent.

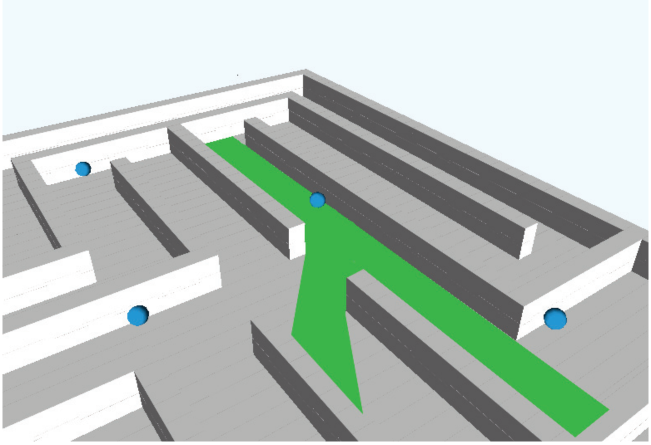


Fig. 4. Field of sight of an agent

An agent also defines two ranges in its neighbourhood: closer and furhter. Depending on the distance between the agent and another object in the enviroment agent can make different decisions.

Movement algorithm uses gradient of path calculated towards the target. Gradients are calculated at the beginning of the simulation, but they can be recalculated. Using this method agent can intelligently choose path omitting obstacles and other dangers.

Steps of the algorithm are shown on Figure 5. Particular agent can continuously adjust the path and change it depending on current situation. Each agent can for example avoid new obstacles or enemies.

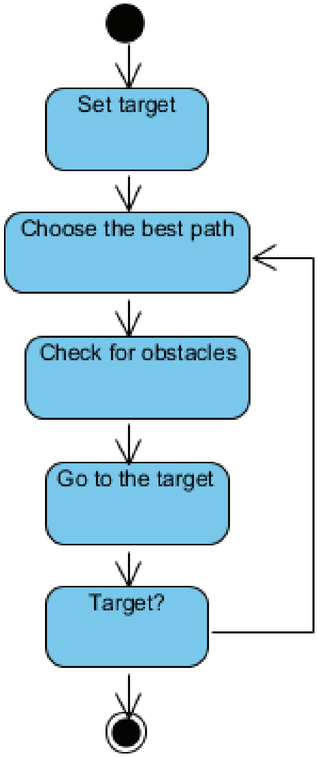


Fig. 5. Movement algorithm

3. MODEL APPLICATIONS

3.1. GROUPING

To demonstrate basic possibilities of the system, a simple simulation of agents grouping will be presented below. All agents are located in a closed square room. They are divided into two concurrent groups. In each group random agent is chosen as a leader. The leader has to stay in the same place during the simulation. Other agents have to surround their leader, as fast as possible. Particular steps and results are shown on Figure 6.

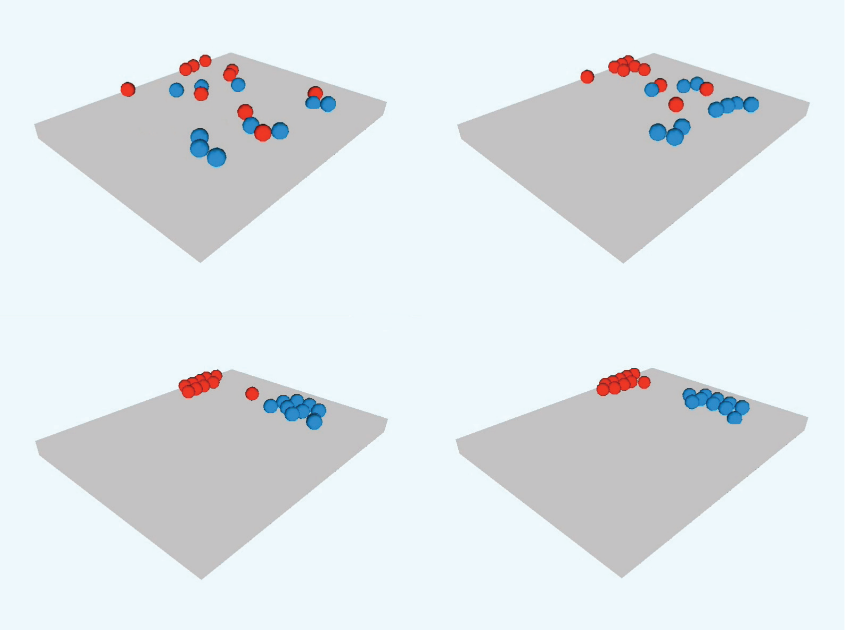


Fig. 6. Grouping agents

3.2. BUILDING EVACUATION

To present more complex behaviours and problems the system was used to implement a simulation of building evacuation. The environment was designed as a four storey building with one staircase in the middle of it. Agents were randomly divided to all flours. Every agent has the same main goal function: to leave the building as fast as possible. Each of them know that the exit is located on the bottom floor in a corner, but each agent has to find, how to get to that exit.

Figure 7 shows the distribution of agents at the beginning of the simulation and a view after 20 seconds. It can be seen that all agents are heading towards the exit in a constant flow. They do not cause blockages on their way and all leave the building.

Such simulations can be also made with more complicated building layouts and bigger number of agents.

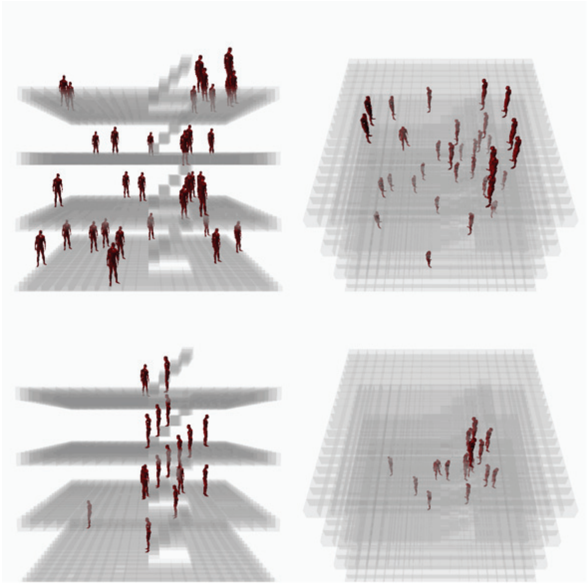


Fig. 7. Building evacuation

4. CONCLUSIONS

The aim of this study was to present the multiagent system dedicated to simulate and visualize the behavior of the crowd. The system allows to simulate crowd behavior in different situations. It can be used for open space simulations, as well as in multi storey buildings or another facilities. Presented system was tested for specific situations like: freeway traffic, evacuation or simulation of battlefield.

Applied connection of multiagent system and Cellular Automaton causes effectiveness and accuracy of simulations.

Presented application is a prototype that can be further developed and customized for specific aims such as simulations to ensure the safety of the crowd (designing objects and organization of mass events), or in the entertainment industry (creation of computer games, movie making).

ACKNOWLEDGEMENTS

The project is partly supported by Polish Ministry of Science

REFERENCES

[1] SIGNORILE R. *A framework combining cellular automata and multiagent in a unified simulation system for crowd control*, European Simulation Symposium, 2004.

[2] KLÜPFEL H.L. *A Cellular Automaton Model for Crowd Movement and Egress Simulation*, Universität Duisburg–Essen, 2003.

[3] CETNAROWICZ K, GRUER P, HILAIRE V, KOUKAM A, *A formal specification of m-agent architecture,* B. Dunin-Keplicz and E. Nawarecki. From Theory to Practice in Multi-Agent Systems. s. l., Springer, 2002.

[4] DIJKSTRA J., JESSURUN A., TIMMERMANS H., *A multiagent cellular automata model of pedestrian movement*, In Pedestrian and evacuation dynamics. 2000.

[5] CETNAROWICZ K, *Problemy projektowania i realizacji systemów wieloagentowych,* Kraków, Uczelniane Wydawnictwa Naukowo-Dydaktyczne Akademii Górniczo-Hutniczej, 1999.

[6] SCHADSCHNEIDER A., KLINGSCH W., KLUEPFEL H., KRETZ T., ROGSCH C., SEYFRIED A., *Evacuation dynamics: Empirical results, modeling and applications*, [in:] Encyclopedia of Complexity and System science Springer 2009.

[7] ZHENG X., ZHONGA T., LIUA M., *Modeling crowd evacuation of a building based on seven methodological approaches*, Building and Environment, Volume 44, Issue 3, March 2009, pp. 437–445.

[8] WĄS J., *Cellular Automata Model of Pedestrian Dynamics for Normal and Evacuation Conditions* 5th International Conference on Intelligent Systems Design and Applications, IEEE CS 2005

[9] WĄS J., KUŁAKOWSKI K., *Agent-based approach in evacuation modeling*, Agent and Multiagent systems. Lecture Notes in Computer Science Springer-Verlag 2010

[10] A. KOSZTYŁA, *Modelowanie tłumu przy pomocy systemów agentowych*, Master thesis, 2011

Finance, Catastrophe bonds,
Neutral martingale measure,
Monte Carlo simulations, CIR model

Piotr NOWAK*, Maciej ROMANIUK**

ON PRICING AND NUMERICAL SIMULATIONS OF SOME CATASTROPHE BOND

The increasing number of natural catastrophes leads to severe losses for insurers. Even one, single catastrophe could cause problems with reserves for many insurers or even bankruptcy of these enterprises. Instead of classical insurance mechanisms, new financial instruments may be used in order to cope with dramatic consequences of such extreme events. The problem is to "package" natural disasters risks (losses) into classical forms of tradable financial assets, like bonds or options. The catastrophe bonds (in abbreviation cat bonds) are the most popular catastrophe-linked securities. We use approach based on neutral martingale method and Monte Carlo simulations in order to analyse some model of catastrophe bond. We price the example of such bond applying stochastic model of risk-free spot interest rate under assumption of independence between catastrophe occurrence and behaviour of financial market. Then we use numerical simulations to analyse the behaviour of the obtained pricing formula.

1. INTRODUCTION

The insurance industry face overwhelming risks caused by natural catastrophes, e.g. the losses from Hurricane Katrina in 2005 are estimated on 40 – 60 billion $ (see [11]). Also other countries are affected by similar problems – e.g. Poland by extreme floods, Japan by enormous earthquakes and tsunamis, etc.

The classical insurance mechanisms adopted by insureds, i.e. application of central limit theorem, are not suitable for such extreme losses caused by natural catastrophes.

* Systems Research Institute Polish Academy of Sciences, e-mail: *pnowak @ibspan.waw.pl*

** Systems Research Institute Polish Academy of Sciences, The John Paul II Catholic University of Lublin, e-mail: *mroman@ibspan.waw.pl*

Even one, single catastrophe could cause problems with reserves for many insurers or even bankruptcy of these enterprises. For example, after Hurricane Andrew more than 60 insurance companies became insolvent (see [11]).

The traditional insurance models (see e.g. [2, 5]) deal with independent risk with rather small claims, like car accidents. Contrary, the sources of losses from natural catastrophes are strongly dependent in terms of time and localization, e.g. single hurricane could blow out many houses, starts fire, leads to robberies, etc. Additionaly, the values of such losses may be extremely high. Therefore new types of financial and insurance instruments are necessary for insurance industry.

So called securization of losses (i.e. "packaging" the risks into tradable asset, like bond or derivative) may be helpful for insurers in dealing with results of extreme natural catastrophes (see e.g. [3, 6, 7, 9]). The catastrophe bonds (cat bonds, see [4, 14, 16]) are the most popular example of catastrophe-linked securities.

Our work is a continuation of Vaugirard's approach (see [17]) to catastrophe bond pricing. We assume no possibility of arbitrage and independence between catastrophe occurrence and behaviour of financial market. An additional assumption is replicability of interest rate changes by financial instruments existing in the market. We price catastrophe bonds applying the Cox-Ingersoll-Ross (CIR) model of risk-free spot interest rate. We also consider a stepwise form of catastrophe bond payoff function. In section 3 we find an analytical valuation formula for catbonds, using the martingale method of pricing. Then in section 4 we use numerical simulations to analyse the behaviour of the obtained pricing formula.

2. CATASTROPHE BONDS

Classical insurance mechanisms are often criticized because of serious problems with adverse selection and moral hazard – e.g., hope for governmental help or possession of insurance policy may change people's attitude. In such case they expose themselves to risk intentionally. Classical reinsurance markets are additional source of problems. In the case of catastrophic events, the reinsurers might not have sufficient capital. The dependency of catastrophic losses in terms of time and localization, and enormous values of such losses should be also mentioned. Therefore the single catastrophic event could cause the bankruptcy of the insurer or serious problems with coverage of losses (see [3, 6, 7, 9]).

Taking into account mentioned problems, applying new kinds of financial or insurance instruments may be profitable. One of possible solutions is to "package" risks (i.e. losses) caused by natural catastrophes into more classical forms of tradable financial assets, like bonds options or other derivatives. The most popular catastrophe-linked security is the catastrophe bond.

Catastrophe bond become wider known in April 1997, when USAA, an insurer from Texas, initiated two new successful classes of cat bonds: A-1 and A-2. According to [10] the cat bond market in year 2003 hit a total issuance of \$1.73 billion, a 42% increase from 2002's record of \$1.22 billion. To the end of 2004 there were about successful 65 emissions of cat bonds. This market is still emerging despite crisis and other problems on financial markets.

Comparing to "classical bond", the payment function of cat bond depends on additional random variable. This variable is connected with so called *triggering point* (see [8, 12, 154]) – occurrence or other properties of specified type of natural catastrophe. Also other parameters like region and time interval for catastrophic event are described in detail for catastrophe bond. The triggering point changes the structure of payments for the cat bond. It may be connected e.g. with occurrence of catastrophe, the issuer's actual losses (e.g. losses from flood), losses modeled by special software based on real parameters of catastrophe, insurance industry index, real parameters of catastrophe (e.g. magnitude of earthquake) or hybrid index related to modeled losses. The structure of payments for cat bonds depends also on some primary underlying asset, like interest rates.

For example, the A-1 USAA bond was connected with losses caused by hurricane on the east coast of USA between July 15, 1997 and December 31, 1997. If the value of losses had been more than \$ 1 billion, the coupon of the bond would have been lost. In case of A-1 USAA bond, the payment depended also on LIBOR.

3. CATASTROPHE BOND PRICING

In this section we introduce pricing formula for catastrophe bonds. We continue our earlier approach from [13 – 15]. We begin with notations and basic definitions concerning catastrophe bonds and their pricing. We define stochastic processes describing dynamics of the spot interest rate and aggregated catastrophe losses. We apply stochastic models with continuous time and time horizon of the form $[0,T']$, where $T'>0$. Date of maturity of a catastrophe bond T is not later than T', i.e. $T\leq T'$. We consider two probability measures: P and Q and we denote by E^P and E^Q the expectations with respect to them. We define stochastic processes and random variables with respect to probability P while probability Q is a martingale measure equivalent to P.

Let $(W_t)_{t\in[0,T']}$ be Brownian motion. It will be used to describe the behaviour of the risk-free interest rate. Let $(U_i)_{i=1}^{\infty}$ be a sequence of identically distributed random variables with bounded second moment. We treat U_i as value of losses during i-th catastrophic event. We define compound Poisson process by formula

$\widetilde{N}_t = \sum_{i=1}^{N_t} U_i$, $t \in [0,T']$, where $(N_t)_{t\in[0,T']}$ is Poisson process with an intensity $\kappa > 0$. For each $t \in [0,T']$ the value of process N_t is equal to the number of catastrophic events till the moment t. In particular, $N_0 = 0$ P - a.s., $E^P N_t = \kappa t$ for $t \in [0,T']$ and $P(N_t - N_s = k) = e^{-\kappa(t-s)} \frac{[\kappa(t-s)]^k}{k!}$, $k = 0,1,2,...$ for $0 \le s \le t \le T'$. Moments of jumps of $(N_t)_{t\in[0,T']}$ are moments of catastrophic events. For each $t \in [0,T']$ process $\widetilde{N}_t$ describes the aggregated catastrophe losses till the moment t. $(\widetilde{N}_t)_{t\in[0,T']}$ is a non-decreasing stochastic process, with right-continuous trajectories of a stepwise form. Heights of its jumps are equal to values of losses during catastrophic events.

All the above processes and random variables are defined on a filtered probability space $(\Omega, F, (F_t)_{t\in[0,T']}, P)$. The filtration $(F_t)_{t\in[0,T']}$ is given by formula

$$F_t = \sigma(F_t^0 \cup F_t^1), \; F_t^0 = \sigma(W_s, s \le t), F_t^1 = \sigma(\widetilde{N}_s, s \le t), \; t \in [0,T'].$$

We assume that $F_0 = \sigma(\{A \in F : P(A) = 0\})$ and that $(W_t)_{t\in[0,T']}$, $(N_t)_{t\in[0,T']}$ and $(U_i)_{i=1}^{\infty}$ are independent. Then the probability space with filtration satisfies standard assumptions, i.e. σ-algebra F is P-complete, filtration $(F_t)_{t\in[0,T']}$ is right continuous and F_0 contains all the sets from F of P-probability zero.

We denote by $(B_t)_{t\in[0,T']}$ banking account satisfying equation $dB_t = r_t B_t dt$, $B_0 = 1$ for a risk-free spot interest rate $r = (r_t)_{t\in[0,T']}$.

We assume that zero-coupon bonds are traded in the market. We denote by $B(t,T)$ the price at the time t of zero-coupon bond with maturity date $T \le T'$ and with face value equal to 1. We price catastrophe bonds under the assumption of no possibility of arbitrage in the market.

We also make two additional assumptions. We first assume that investors are neutral toward nature jump risk (Assumption 1). This assumption has practical confirmations in the market (see e.g. [1, 17]). Secondly (Assumption 2), we assume routinely that changes in interest rate r can be replicated by existing financial instruments (especially zero-coupon bonds).

We consider a catastrophe bond, with a stepwise payoff function. Our aim is to find and prove its valuation formula. Let $0 < K_1 < ... < K_n$, $n > 1$, be a sequence of constants. Let $\tau_i : \Omega \to [0,T']$, $1 \le i \le n$ be a sequence of stopping times of the form

$\tau_i(\omega) = \inf_{t\in[0,T']}\{\widetilde{N}_t(\omega) > K_i\} \wedge T'$, $1 \le i \le n$. Let $w_1 < w_2 < ... < w_n$ be a sequence of nonnegative constants, for which $\sum_{i=1}^{n} w_i \le 1$.

Definition 1. *We denote by $IB_s(T, Fv)$ a catastrophe bond satisfying the following assumptions:*

a) If the catastrophe does not occur in the period $[0, T]$, i.e. $\tau_1 > T$, the bondholder is paid the face value Fv;

b) If $\tau_n \le T$, the bondholder receives the face value minus the sum of write-down coefficients in percentage $\sum_{i=1}^{n} w_i$.

c) If $\tau_{k-1} \le T < \tau_k$, $1 < k \le n$, the bondholder receives the face value minus the sum of write-down coefficients in percentage $\sum_{i=1}^{k-1} w_i$.

d) The cash payments are done at date of maturity T.

We denote by $\upsilon_{IB_s(T,Fv)}$ the payoff function of $IB_s(T, Fv)$.

We assume CIR model of the risk-free spot interest rate. Process r is the solution of stochastic equation

$$dr(t) = a(b - r(t))dt + \sigma\sqrt{r(t)}dW_t$$

for constants $a, b, \sigma > 0$, such that $2ab > \sigma^2$. CIR model was introduced by Cox, Ingersoll and Ross and it is an extension of the Vasicek model. It is often used for modeling of risk-free spot interest rate.

We assume the following stochastic form of risk premium of risk-free bonds $\lambda_u = \lambda\sqrt{r_u}$, $u \in [0, T']$, where λ is a constant. The following theorem gives the no-arbitrage pricing formula for $IB_s(T, Fv)$ with CIR interest rate dynamics. This type of a catastrophe bond was also considered in [13 – 15] for $n = 1$.

Theorem 1. *Let $IB(0)$ be the price of $IB_s(T, Fv)$ at time 0. Let*

$\Phi = \sum_{i=0}^{n} \Phi_i$, *where Φ_i are cumulative distribution function of τ_i. Then*

$$IB(0) = FvP(r,0)(1-\Phi(T)), \tag{1}$$

where $P(r,0) = A(T)e^{-r_0 B(T)}$, $A(T) = \left[\dfrac{\theta_1 e^{\theta_2 T}}{\theta_2\left(e^{\theta_1 T}-1\right)+\theta_1}\right]^{\theta_3}$, $B(T) = \dfrac{e^{\theta_1 T}-1}{\theta_2\left(e^{\theta_1 T}-1\right)+\theta_1}$ *with* $\theta_1 = \sqrt{(a+\lambda)^2 + 2\sigma^2}$, $\theta_2 = \dfrac{a+\lambda+\theta_1}{2}$ *and* $\theta_3 = \dfrac{2ab}{\sigma^2}$.

Proof. Using the above assumptions, we obtain the unique probability measure Q in similar way as in [17]. The following Radon–Nikodym derivative defines measure Q:

$$\frac{dQ}{dP} = \exp\left(\int_0^T \lambda_u dW_u - \frac{1}{2}\int_0^T \lambda_u^2 du\right) \quad P-a.s.$$

For Q the family $B(t,T)$, $t \le T \le T'$, of zero-coupon bond prices with respect to r is arbitrage-free, i.e. for each $T \in [0,T']$ $B(T,T) = 1$ and processes $B(t,T)/B_t$, $t \in [0,T]$, of discounted zero-coupon bond prices are martingales with respect to Q. Then we have the following pricing formula for zero-coupon bond $B(t,T) = E^Q\left(e^{-\int_t^T r_u du} \mid F_t\right)$, $t \in [0,T']$. Using arguments similar as in [17], we obtain the analogous equality for the catastrophe bond:

$$IB(t) = E^Q\left(e^{-\int_t^T r_u du}\upsilon_{IB_s(T,Fv)} \mid F_t\right). \tag{2}$$

From Assumption 1, $\exp\left(-\int_t^T r_u du\right)$ and $\upsilon_{IB_s(T,Fv)}$ are independent under Q. Therefore formula (2) can be written in the form $IB(t) = E^Q\left(e^{-\int_t^T r_u du} \mid F_t\right)E^Q\left(\upsilon_{IB_s(T,Fv)} \mid F_t\right)$ and in particular, for $t = 0$, in the form $IB(0) = E^Q\left(e^{-\int_0^T r_u du}\right)E^Q\left(\upsilon_{IB_s(T,Fv)}\right)$. Since $\upsilon_{IB_s(T,Fv)} = Fv\left\{1 - \sum_{i=1}^{n} w_i I_{\{\tau_i \le T\}}\right\}$, from Assumption 1,

$$E^Q\left(\upsilon_{IB_s(T,Fv)}\right) = FvE^Q\left\{1-\sum_{i=1}^{n} w_i I_{\{\tau_i \le T\}}\right\} = Fv\left\{1-\sum_{i=1}^{n} w_i E^Q I_{\{\tau_i \le T\}}\right\}$$

$$= Fv\left\{1-\sum_{i=1}^{n} w_i E^P\left(I_{\{\tau_i \le T\}}\frac{dQ}{dP}\right)\right\} = Fv\left\{1-\sum_{i=1}^{n} w_i E^P\left(I_{\{\tau_i \le T\}}\right)E^P\left(\frac{dQ}{dP}\right)\right\}$$

$$= Fv\left\{1-\sum_{i=1}^{n} w_i E^P\left(I_{\{\tau_i \le T\}}\right)\right\} = Fv\left(1-\Phi(T)\right).$$

From zero-coupon bond pricing formula for CIR interest rate model (see e.g. [18]) it follows that $E^Q\left(e^{-\int_0^T r_u du}\right) = P(r,0)$ and finally we obtain (1). □

The following lemma from [13] gives the form of the cumulative distribution functions of τ_i and can be applied to computations of the catastrophe bond price.

Lemma 1. *The value of cumulative distribution function* $\Phi_i(\omega)$, $1 \le i \le n$, *at the moment* T *has the form*

$$\Phi_i(T) = 1 - \sum_{j=0}^{\infty} \frac{(\kappa T)^j}{j!} e^{-\kappa T} \Phi_{\tilde{U}_j}(K_i),$$

where $\Phi_{\tilde{U}_j}$ *is the cumulative distribution function of the sum* $\tilde{U}_j = \sum_{p=0}^{j} U_p$, $j = 0,1,\ldots$ *In the above formula we assume that* $U_0 = 0$.

4. MONTE CARLO EXPERIMENTS

In order to analyze the numerical features of the cat bond described in section 3, we use Monte Carlo simulations. In our experiments we price the catastrophe bond according to equation (1). In each case we use *n* = 100 000 000 simulations.

The catastrophe process was modeled by Poisson process with intensity *μ* = 0.05. The value of single catastrophe was described by Gamma random variable with shape parameter *α* and scale parameter *β*. The risk-free interest spot rate was described by CIR model with parameters *a* = 0.025, *b* = 0.01, *r*(0) = 0.05, *σ* = 0.01.

First type of catastrophe bond has parameters *T=5*, *Fv=1*, $w_1 = 0.2$, $k_1 = 50$. We price this bond for various shape parameters with fixed scale parameter *β=10* (see Table 1) and various scale parameters with fixed shape parameter *α=5* (see Table 2).

Tab. 1. Price of the catastrophe bond for various shape parameters

Shape parameter (*α*)	5	6	7	8	9	10
Price	0. 780137	0.778838	0.777759	0.776965	0.776557	0.776258

Tab. 2. Price of the catastrophe bond for various scale parameters

Scale parameter (*β*)	10	12	14	16	18	20
Price	0.780137	0.779023	0.778137	0.777513	0.77709	0.77687

As we could see, for larger values of parameters, the price of catastrophe bond is lower. However, the differences in prices are rather small, as may be seen on Fig. 1 where the price is plotted against values of shape (*x* axis) and scale (*y* axis) parameters.

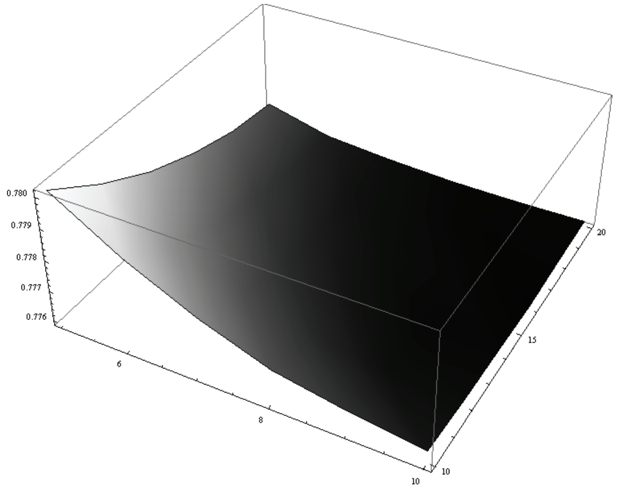


Fig. 1. Price of the catastrophe bond as a function of scale and shape parameters

In next case we analyze price of the catastrophe bond as the function of value of triggering point k_1. We set $\alpha = 5$ and $\beta = 10$. Other parameters (for interest rate model, face value and w_1) were the same as in previous experiment. As we could see on Fig. 2, the differences in prices are relatively low and the appropriate function seems to be hyperbolic curve.

In next case we analyze price of the catastrophe bond as the function of percentage loss of face value w_1. Other parameters (for shape parameter, scale parameter, interest rate model, face value and $k_1 = 50$) were the same as in previous experiment. As we could see on Fig. 3, the differences in prices are relatively low and function seems to be linear.

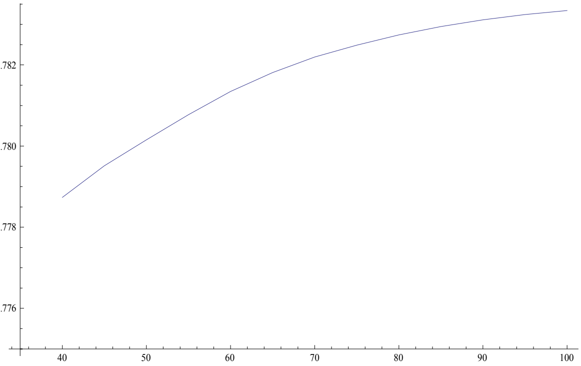


Fig. 2. Price of the catastrophe bond as a function of value of triggering point

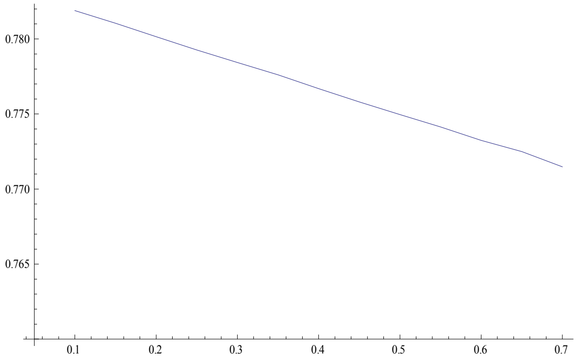


Fig. 3. Price of the catastrophe bond as a function of percentage loss of face value

5. CONCLUDING REMARKS

In this chapter we use approach based on neutral martingale method and Monte Carlo simulations in order to price and analyse some features of example of catastrophe bond. We price the catbond applying CIR model of risk-free spot interest rate. Then we use simulations to analyse the behaviour of the obtained pricing formula in series of numerical experiments.

REFERENCES

[1] ANDERSON R.R., BENDIMERAD F., CANABARRO E., FINKEMEIER M., *Analyzing insurance-linked securities*. Journal of Risk Finance 1(2), 2000.

[2] BORCH, K., *The Mathematical Theory of Insurance*. Lexington Books, 1974.

[3] CUMMINS, J.D., DOHERTY, N., LO, A., *Can insurers pay for the "big one"? Measuring the capacity of insurance market to respond to catastrophic losses*. Journal of Banking and Finance 26, 2002.

[4] ERMOLIEVA, T., ROMANIUK, M., FISCHER, G., MAKOWSKI, M., *Integrated model-based decision support for management of weather-related agricultural losses*. In: Enviromental informatics and systems research. Vol. 1: Plenary and session papers – EnviroInfo 2007, Hryniewicz, O., Studziński, J., Romaniuk, M. (eds.), Shaker Verlag, IBS PAN, 2007.

[5] ERMOLIEV, Y., ERMOLIEVA, T., MACDONALD, G., NORKIN, V., *Problems on Insurance of Catastrophic Risks*. Cybernetics and Systems Analysis, 37 (2), 2001.

[6] FROOT, K.A. (eds.), *The Financing of Catastrophe Risk*. University of Chicago Press, Chicago, 1999.

[7] FROOT, K.A., *The market for catastrophe risk: A clinical examination*. Journal of Financial Economics 60 (2), 2001.

[8] GEORGE, J. B., *Alternative reinsurance: Using catastrophe bonds and insurance derivatives as a mechanism for increasing capacity in the insurance markets.* CPCU Journal, 1999.

[9] HARRINGTON, S.E., NIEHAUS, G., *Capital, corporate income taxes, and catastrophe insurance*. Journal of Financial Intermediation 12 (4), 2003.

[10] MCGHEE, C., *Market Update: The Catastrophe Bond Market at Year-End 2003*. Guy Carpenter & Company, Inc. and MMC Security Corporation, 2004.

[11] MUERMANN, A., *Market Price of Insurance Risk Implied by Catastrophe Derivatives*. North American Actuarial Journal, 12 (3), pp. 221–227, 2008.

[12] NOWAK, P., ROMANIUK, M., ERMOLIEVA T., *Integrated management of weather – related agricultural losses – computational approach.* In: Information Systems Architecture and Technology, Wilimowska, E., Borzemski, L., Grzech, A., Świątek, J. (eds.), Wrocław 2008.

[13] NOWAK, P., ROMANIUK, M., ERMOLIEVA T., *Evaluation of Portfolio of Financial and Insurance Instruments – Simulation of Uncertainty* (to be published).

[14] NOWAK, P., ROMANIUK, M., *Portfolio of financial and insurance instruments for losses caused by natural catastrophes*. In: Information Systems Architecture and Technology, Wilimowska, E., Borzemski, L., Grzech, A., Świątek, J. (eds.), Wrocław 2009.

[15] NOWAK, P., ROMANIUK, M., *Analiza wlasnosci portfela zlozonego z instrumentow finansowych i ubezpieczeniowych.* STUDIA I MATERIALY POLSKIEGO STOWARZYSZENIA ZARZADZANIA WIEDZA, 31, 2010 (in Polish).

[16] ROMANIUK, M., ERMOLIEVA, T., *Application EDGE software and simulations for integrated catastrophe management.* International Journal of Knowledge and Systems Sciences, 2(2), pp. 1 – 9, 2005.

[17] VAUGIRARD, V., E., *Pricing catastrophe bonds by an arbitrage approach*. The Quarterly Review of Economics and Finance, 43, 2003.

[18] XUEPING, W., *A New Stochastic Duration Based on the Vasicek and CIR Term Structures Theories*. Journal of Business Finance and Accounting, 27, 2000.

intelligent information system,
tornado, weather modeling, Rankine vortex,
HWIND tree damage model

Radosław CHMIELARZ*, Jarosław WĄS*

SYSTEM FOR SIMULATING TORNADO DAMAGES IN FORESTS

The aim of this work is to provide an overview of an intelligent information system dealing with simulation of tornado damages inflicted in forests. The system is potentially useful for forests managers in delivering information about optimization of newly grown tree stands against tornado damages in regions endangered with severe wind gusts.

The system consists of a combined Rankine vortex used for tornado simulation and of HWIND tree damage model used for assessing tornado impact on forests. The Rankine vortex equations have been expanded to three dimensions using real tornadoes data. The HWIND model has been modified to be used for sudden wind blows conflicted by tornadoes in contrast to constant wind speeds for which it was designed. Never before had all the equations for Rankine vortex and HWIND tree damage model been gathered in one work. The simulation is visualized by the system and results in a pattern of downed trees from which suitable conclusions for forest managers are made.

The authors believe that the work is important as Europe is struck by tornadoes which conflict damages worth billions of euros every year, mostly in unpopulated areas such as forest. Moreover the scientific literature on this topic in Europe is sparse.

1. INTRODUCTION

It may seem that tornadoes are not frequent events in Poland, but only in 2010 European Severe Weather Database [1] has collected 83 reports of severe wind gusts - sudden winds blowing with speeds over 25km/h and 18 tornadoes. The meaning of tornado's damages is also notable as in 2008 windstorm Emma, crossing Central Europe, conflicted damages estimated by over 1 billion euros.

* University of Science and Technology AGH in Cracow.

The topic of tornado modeling is also not widely described in European literature and up to this date there was no article putting all the equations for HWIND tree damage model in one place. HWIND model plays a central role in estimating effects of wind gusts on trees in forests.

2. SYSTEM OVERVIEW

The system consist of two models: vortex and tree damage. For the vortex model Rankine's method was chosen. Its purpose is to give a wind speed value for a given point in space in the area struck by a tornado. The second model was implemented by HWIND tree damage model which task is to provide maximum bending force for a particular tree which is necessary for that tree to be either broken or uprooted by wind. The result of simulation based on those two combined sub-models is a downed tree pattern. From the downed tree pattern one can infer the impact of changes in model's attributes on trees in the forest and in consequence deduct how to protect forests from tornado effects.

A class diagram for the system is given in figure 1. The classes corresponds directly to models used. **GUI** class is the interface between user and the rest of the system. **Simulation** class is responsible for running a simulation in loop until the center of the vortex is out of the simulated forest area. The **Simulation** class uses **AbstractVortexModel** which is a parent class for Rankine which purpose is to provide calculations from Rankine vortex model. **ForestModel** class is responsible for gathering data about trees in the forest by using calculations for single tree take from **HWIND** class which is a child class of **AbstractTreeModel** and uses **HWINDData** class for particular tree species used in the system.

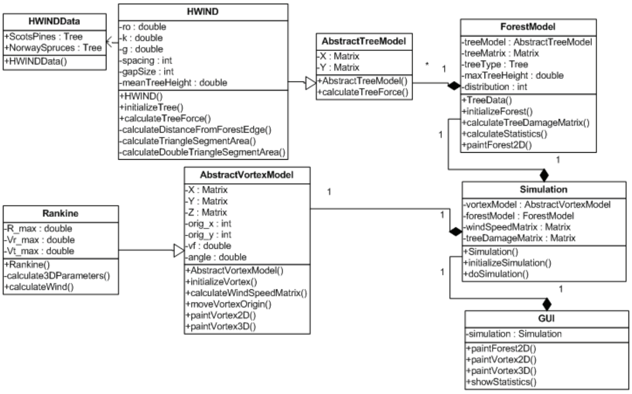


Fig. 1. System’s class diagram. Source: original work

A sequence diagram is given in figure 2. First the user has to provide parameters of the tornado, which include vortex origin coordinates, speed and direction of the move of tornado, maximum vortex radius and maximum wind speeds in the tornado. Next the user has to choose what tree species are growing in the forest. In the forest only one type of trees can be simulated due to HWIND tree damage model's limitation which impose a forest has to be homogenous. The user is also asked to estimate the average age of trees in the forest which is represented by maximum height of a tree. Data on trees' species parameters are stored in **HWINDData** class.

The simulation is done on a grid. On each intersection grows a tree and by manipulating the distance between them and the area of simulated location one can manage forest density.

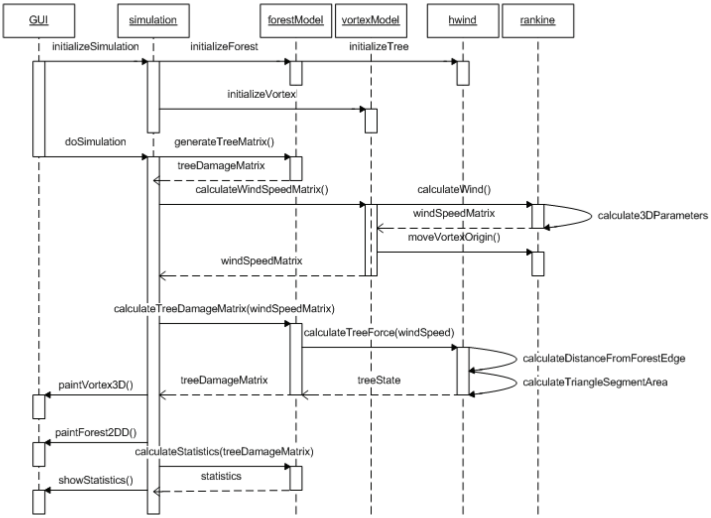


Fig. 2. System's sequence diagram. Source: original work

When the start button in the application is pressed the sequence is started. The grid is initialized by **initializeSimulation()**, forest parameters are set in **initializeForest()**, tree specie is set in **initializeTree()** and finally vortex attributes are set by **initializeVortex()**. When the simulation is started in **doSimulation()** trees attributes are drawn from random number generator in **generateTreeMatrix()** and a blank ***treeDamageMatrix*** is put in ***simulation*** object. Next wind speeds on the intersections of the grid are calculated in **calculateWindSpeedMatrix()**, which in turn uses **calculateWind()** method which calculates wind speeds of a tornado vortex in three dimensions using **calculate3DParameters()**. The new origin point of the vortex center is calculated and moved accordingly to its transition speed vector in **moveVortexOrigin()**. The result of this step is a ***windSpeedMatrix***. The speeds from ***windSpeedMatrix*** are used to check if the wind has inflicted any damage in the forest in **calculateTreeDamageMa-**

trix(). This is done by calling **calculateTreeForce()** on each tree in the forest which calculates equations from HWIND tree damage model and gives an answer on the ***treeState*** which can be either broken, uprooted or standing still. The ***treeDamageMatrix*** is returned to ***simulation*** object which uses ***windSpeedMatrix*** and ***treeDamageMatrix*** to display them to the user by calling **paintVortex3D()** and **paintForest2D()** methods of ***gui*** object. The last step is to gather ***statistics*** of downed trees in the forest and display them to the user. If the origin of the vortex is still in the area of the grid the next step of simulation is computed. If not the simulation ends and the user is presented with complete downed tree pattern.

3. RANKINE VORTEX MODEL

Rankine vortex is a steady-state vortex model developed in the middle of XIX century [2]. It consists of two regions. In the inner region wind speed is rising linearly with the rise of distance from the center of the vortex. Speeds in the outer region are behaving in the opposite way, where they lower with the rise of distance from the vortex center. The border between regions is defined by radius **R**. The situation can be seen in figure 3.

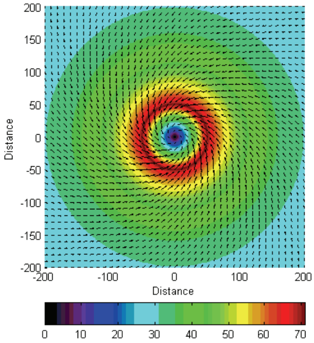


Fig. 3. Rankine vortex 2D visualization. Source: original work

Equations for the vortex are best presented in polar coordination system. Tangential velocity V_φ is calculated from the equation (1) and the radial velocity V_r is calculated according to equation (2).

$$V_\varphi = \begin{cases} V_{\varphi,max}\left(\frac{r}{R_{max}}\right), & r \leq R_{max}, \\ V_{\varphi,max}\left(\frac{R_{max}}{r}\right), & r > R_{max}, \end{cases} \tag{1}$$

$$V_r = \begin{cases} V_{r,max}\left(\frac{r}{R_{max}}\right)^{0.6}, \; r \leq \; R_{max}, \\ V_{r,max}\left(\frac{R_{max}}{r}\right)^{0.6}, \; r > R_{max}. \end{cases} \tag{2}$$

In the equations r is the distance of each point in the grid from the vortex center and R_{max} is the size of inner vortex region. Exponentiation of equation (2) was approximated to 0.6 in [3] by measuring radial velocities of tornadoes by mobile Doppler radar.

Total wind speed of a tornado in each point of the grid is a superposition of tangential and radial velocity vectors plus the forward velocity of vortex center which emulates a moving tornado.

Rankine model is two dimensional but the authors of this publication have expanded it to third plane. It has been done by making the maximum radius R_{max} and tornado center coordinates (x_0, y_0) dependent on the height z. There are many tornado shapes but in this work V-shaped tornadoes were chosen.

Authors have gathered pictures of real V-shaped fully connected to the ground tornadoes. Then by measuring landscape objects on the pictures a scale for each picture was estimated. Next step was to select tornado shapes from the background, convert pictures to black and white and with the help of MATLAB Image Processing Toolbox estimate radius and center position on each height z. Finally all data points where averaged and approximated to functions. Because pictures showed vortexes only from one angle it was decided that the resulting 3D model will computed by using the same equation for both x and y-plane. Estimated functions for radius R_z is shown in equation (3) and the center (x_z, y_z) is shown in equation (4).

$$R_z = (-8.5637 \cdot 10^{-8} z^3 + 0.00018695 z^2 + 0.0078765 z + 0.94933) R_0 \tag{3}$$

$$\begin{aligned} x_z &= x_0 + (1.3142 \cdot 10^{-7} z^4 - 3.864 \cdot 10^{-5} z^3 + 0.0048048 z^2 - 0.10169 z - 0.46675)\frac{R_0}{5} \\ y_z &= y_0 + (1.3142 \cdot 10^{-7} z^4 - 3.864 \cdot 10^{-5} z^3 + 0.0048048 z^2 - 0.10169 z - 0.46675)\frac{R_0}{5} \end{aligned} \tag{4}$$

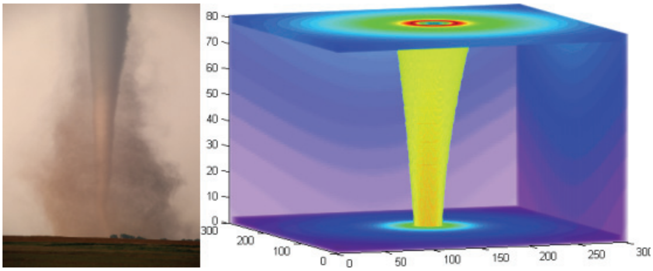


Fig. 4. Comparison between real tornado shape and Rankine vortex 3D visualization. Source of the tornado image: [4]

Figure 4 compares real tornado (not present in the learning set) with the visualization.

4. HWIND TREE DAMAGE MODEL

HWIND model was designed in University of Joensuu in Finland [5]. Its purpose was to simulate trees resistance to constant winds blowing within 10 minutes range 10 meters from the ground on podzolic soil.

Forces acting on the trees are shown in figure 5. First the tree is divided on 1 meter segments and forces are computed for each segment. In the end the segment's forces are summed and the result is the total force acting on a tree.

Total wind-induced forces F_w are written in equation (5). Parameter z is the height (segment number) above the ground in meters, C_d is a dimensionless drag coefficient given in table 1, ρ is the air density, v_h is the wind speed on given height and $A(z)$ is the projected area of each tree segment giving resistance to the wind.

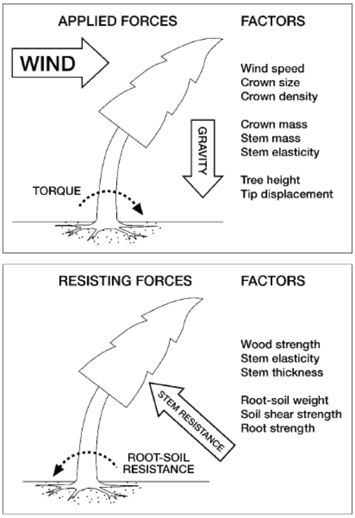


Fig. 5. Forces having impact on a tree. Source: [6]

Two type of tree species are modeled: Scots Pine and Norway Spruce, which are the most popular tree types in Polish forests [7]. Each of them has a distinct crown shape which was approximated by an equilateral triangle in case of Scots Pine and by two equilaterals triangles connected to each other with their bases in case of Norway Spruce. The canopy of a tree was approximated by a rectangle. In consequence segment projected area $A(z)$ is a 1 meter fragment of either the rectangle representing tree canopy or a slice to triangle representing tree crown depending on the height above the ground.

$$F_w(z) = \frac{1}{2} C_d \rho v_h^2 A(z) \tag{5}$$

As the wind blows on the tree the area of the crown is reduced. If the wind blows with speeds below or equal to $11 \frac{m}{s}$ then the area is reduced by 20%. If the wind speed is over $20 \frac{m}{s}$ then the area is reduced by 60 %. And if the wind velocity is in between the shrinking factor S_t is computed from equation (6).

$$S_t(z) = \frac{10}{v(z)} - 0.1 \tag{6}$$

Table 1. Data for HWIND tree damage model. Source: [8]

Parameter	Scots Pine	Norway Spruce
Modulus of rupture (MOR)[MPa]	39.1	30.6
Modulus of elasticity (MOE) [MPa]	7000	6300
Air density (ρ) $\left[\frac{kg}{m^3}\right]$	1.226	1.226
Drag coefficient (C_d)	0.29	0.35
Soil mass to tree mass ratio (f_{RW})	0.3	0.2
Crown approximation shape	Triangle	Double triangle

After the tree is bend by the wind gravitational force F_g in equation (7) starts acting on each segment. Here m_c is the mass of each tree segment.

$$F_g(z) = m_c g \tag{7}$$

Finally total bending moment B_{max} for each segment $z$is the sum of wind-induced force F_w multiplied by the height z and gravitational force F_g multiplied by the horizontal displacement from upright of the segment. The sum is further multiplied by gust factor f_{gust} and gap factor f_{gap} described later in the work.

$$B_{max}(z) = f_{gust} f_{gap} \left[F_w(z) \Delta z + F_g(z) x(z) \right] \tag{8}$$

Horizontal displacement of a segment $x(z)$ is presented in equation (9). Here a means the height from the ground to the middle of the crown, h - total height of the tree, $l(z)$ distance from the tree top to the height z, modulus of elasticity MOE given in table 1, I is the area moment of inertia of the tree stem m^4 which equals to $I = \frac{\pi d_{bh}^4}{64}$, where d_{bh} mean the tree diameter on the breast height (1.3 m). Equation is divided in two parts. In the first part the segment z being calculated is below the height of the middle of the crown and in the other part it is above.

$$x(z) = \begin{cases} \frac{F_w a^2 h\left(3-\frac{a}{h}-\frac{3l(z)}{h}\right)}{6\cdot MOE\cdot I}, & z \leq a, \\ \frac{F_w a^3\left(2-\frac{3(l(z)-b)}{a}+\frac{(l(z)-b)^3}{a^3}\right)}{6\cdot MOE\cdot I}, & z > a. \end{cases} \tag{9}$$

Each segment maximum bending moments $B_{max}(z)$ are summed together resulting in the total bending moment B_{max} for a tree.

In a research conducted in 2005 Gardnier and Stacey [6] showed that trees behave differently under wind pressure if they grow in different distances from the forest edge and apart from each other. Suitable equations are shown in (10) and (11).

$$\begin{gathered} Gust_{mean} = \left(0.68\frac{s}{h} - 0.0385\right) + \left(-0.68\frac{s}{h} + 0.4875\right)\left(1.7239\frac{s}{h} + 0.0316\right)^{\frac{x}{h}} \\ Gust_{max} = \left(2.7193\frac{s}{h} - 0.061\right) + \left(-1.273\frac{s}{h} + 9.9701\right)\left(1.1127\frac{s}{h} + 0.0311\right)^{\frac{x}{h}} \\ f_{gust} = \frac{Gust_{max}}{Gust_{mean}} \end{gathered} \tag{10}$$

$$\begin{gathered} Gap_{mean} = \frac{0.001+0.001p^{0.562}}{0.00465} \\ Gap_{max} = \frac{0.0072+0.0064p^{0.3467}}{0.0214} \\ f_{gap} = \frac{Gap_{max}}{Gap_{mean}} \end{gathered} \tag{11}$$

Furthermore in the tunnel experiment it was noted that the first row of trees in an artificially grown forest has larger resisting force than those trees growing in the middle of the forest. It is suspected that this phenomena is connected to wood density in the outer trees and the airflow inside the forest.

The above equations were connected to the forces applied on a tree. In equations (12) and (13) resistive forces are shown.

$$M_{bk} = \frac{\pi}{32} MOR d_{bh}^3 \tag{12}$$

Stem breakage resisting force M_{bk} is a product of outer fiber layer resistance on the breast height (1.3 m). The tree is assumed to bend to a point of no return which results in breakage of the stem.

$$M_{up} = \frac{g R_{mass} R_{depth}}{f_{RW}} \tag{13}$$

Resistance to a tree uproot M_{up} is a multiplication of gravitational force g, root's mass R_{mass}, root's depth R_{depth} and a ratio of soil mass around the roots to the whole tree mass f_{RW}. If the forced is exceeded the tree is assumed to be uprooted.

Lastly HWIND model assumes that the wind is blowing constantly with the same velocity for at least 10 minutes and in consequence bends the tree to a point of no return. However as tornadoes are a sudden and violent phenomena it was shown by other researchers [9] that its wind blows velocities v_{sudden} correspond to mean wind velocities v_{mean}. The reliance was approximated by authors to equation (14).

$$v_{mean} = -0.0032158v_{sudden}^2 + 0.54722v_{sudden} - 0.1285 \tag{14}$$

5. RESULTS

Downed tree pattern from system's simulation has been compared to an image of a real forest after a tornado had passed through it. As it can be seen in figure 6 the simulation has captured main characteristics of a real downed tree pattern. The damaged area of the forest is narrow as in the simulation and some trees had been broken while others uprooted. Moreover downed trees are aligning in semicircles as in the simulation.

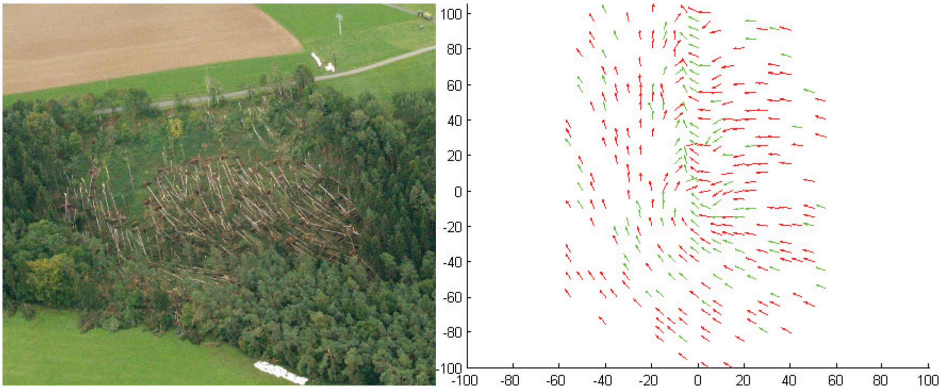


Fig. 6. Comparison between image of a forest after a tornado has passed through it and downed tree pattern from system's simulation. Area of damage is approximately 120 m wide; forest consist of Scots Pine from 12 to 16 m height with distance of about 5 m between trees; tornado is estimated as T6 (73–83 m/s) with 8 m in diameter at the ground. Source of the image: [10]

Downed tree patterns from simulations with different values of each simulation parameter have also been compared to each other and conclusions for forest managers have been drawn:

- Tree height – as expected the force needed to break or uproot a tree grows with the height and diameter of a tree. However the force needed to uproot a tree for thin and small trees is bigger than the force needed to break them, this trend is inverted when it comes to big and thick trees. This dependence can be explained by relatively small root mass growth with tree age and over proportional growth of tree's stem and crown.
- Tree species – Scots pine require greater wind speeds than Norway Spruce in order to be broken or uprooted, therefore forest manager can consider planting trees species which are more resistant to the winds.
- Distance between trees – the lower the distance between trees the greater force is needed to break or uproot them. Therefore densely populated forests are more resistant to tornadoes, however the system in current shape doesn't take into account effects of broken trees falling on each other.
- Upwind gap – the bigger the upwind gap in front of forest edge the more resistant are the trees to tornado damage. The possible cause of this effect may be that winds in narrow spaces are blowing faster than in wide spaces. Therefore forest managers may consider leaving a gap in front of forests in order to strengthen them against sudden winds.
- Mean tree height – the bigger mean height of trees in the forest the more resistant the trees are to tornado damages. The cause of this is that bigger trees are protecting the smaller ones and therefore collectively less trees are downed in the forest. The conclusion for forest managers is that newly grown trees could be planted in between areas with older trees.
- Distance from forest edge – the force needed to down a tree rises with the rise of the distance from the forest edge up to a point and then it starts to fall as the distance is bigger. The cause is believed to be connected with wind flow in narrow spaces [6]. And the conclusion for forest managers is that newly grown trees should be grown in front of an area closed by older trees.

6. CONCLUSIONS AND FURTHER WORK

A system for simulating tornado damage in forests have been shown. The authors have presented all the equations for HWIND tree damage and Rankine vortex models. Rankine vortex has been expanded with equations related to its shape in third dimension based on real tornado data. The HWIND tree damage model has been expanded with equation which translates constant speed of wind blowing in one direction for a period of time to the speeds of sudden winds present in tornadoes. The model, although simple in construction, has met expectations in analysis of tornado characteristics.

From simulation results conclusions for forest managers have been drawn and presented in chapter 5.

In further works the authors will try to overcome current model limitations by introducing second hand effects such as the impact falling trees have on their neighbors. The future work could also benefit from studying airflow in narrow spaces and the difference of wind speeds on different heights above the ground.

REFERENCES

[1] *European Severe Weather Database,* http://www.essl.org/ESWD/.

[2] GIAIOTTI D.B.,STEL F., *The rankine vortex model*, Visco, Italy, University of Trieste, 2006.

[3] BLUESTEIN H., *Mobile Doppler radar observations of tornadoes*, Preprints. 4th European Conference on Severe Storms, 2007.

[4] HUMPAGE M., http://www.markhumpage.com/Mother-Nature/Sever-Weather/, 2008.

[5] PELTOLA H., KELLOMAKI S., *A mechanistic model for calculating windthrow and stem breakage at stand edge*, Silva Fennica, 27:99–111, 1993.

[6] PELTOLA H., *Mechanical stability of trees under static load*, 2006.

[7] *Polskie Lasy Państwowe*, http://www.lasy.gov.pl/o_lasach/polskie_lasy, 2010.

[8] PELTOLA H., KELLOMAKI S., VAISANEN H., IKONEN V., *A mechanistic model for assessing the risk of wind and snow damage to single trees and stands of scots pine, norway spruce and birch*, 1999.

[9] GARDINER B., PELTOLA H., KELLOMAKI S., *Comparison of two models for predicting the critical wind speeds required to damage coniferous trees*, 1999.

[9] SAVERT T., *Tornadolste Deutschland*, http://www.tornadoliste.de/061002quirla_luftbilder.htm , 2006.

wind energy, forecast, Hurst exponent

Tomasz POPŁAWSKI*, Piotr SZELĄG*

A FRACTAL APPROACH TOWARDS WORK OF A WIND POWER STATION

Since several years the percentage of renewable power engineering in the sector of electric energy sector constantly grows. The reasons for this state are various: depletion of fossil fuels, breakdowns of atom power stations, which cause the feeling of anxiety, strong pro-ecological tendencies as well as the possibility to make the country independent on others. A higher number of new sources of electric energy, in particular wind farms, causes new problems related to forecasting the energy production level. Wind power stations are units, which do not provide a stable level of energy supply. Therefore there exists a need to develop forecasting models, which make it possible to forecast the work of such units in a reliable way. In the work an attempt has been undertaken to apply the theory of fractal analysis to the processes related to the operation of wind power stations. Research results and suggestions concerning their further possible applications have been given.

1. INTRODUCTION

Production of electric energy in different countries of our globe has been based on stable sources. Conventional power stations, based on coal, lignite or gas, atom power stations based on fission of atom nuclei or huge hydropower plants are the paradigm of these sources. Their common feature is stability and predictability of their production, a well as the possibility to control the level of generated output power. These features are crucial from the point of view of the operator of distribution system, whose task is to balance the needs of energy consumers with the level of produced energy obtained from the energy suppliers. In practice this is accomplished by adjusting the level of produced electric energy to the actual requirements. In such a system the unknown variable to be forecast is the level of electric energy demand by the con-

* Częstochowa University of Technology, Electrical Faculty, Armii Krajowej 17, 42-200 Częstochowa.

sumers. For many years the prediction methods, used in practice and providing a reliable action of the electric power engineering system, have been worked out [7, 9].

In recent years some tendencies to changes in the power engineering sector have been observed. More and more attention is paid to the production of electric energy with the least negative impact on the natural environment. Therefore the technologies using non-conventional energy sources develop so quickly. This is possible – among other factors – due to a strong and ever increasing political support, which manifests itself in measurable way in financial subsidies to pro-ecologic investments. Another aspect, namely making oneself independent on supplies of energy resources, is also very important, as it stimulates the development of new technologies, in particular in the USA.

One of the fastest growing branches is wind power engineering. Unfortunately, along with the development of this branch of power engineering, new problems emerge. Those are related to instability of production level of electric energy produced by wind farms. Power generated by a wind turbine is dependent to a large extent on the wind velocity. This dependence may be written as [4]

$$P_w = c_p \rho D^2 v^3 \tag{1}$$

where:

c_p – total efficiency of transformation wind energy into mechanical energy

ρ – air density $\left[\frac{kg}{m^3}\right]$

D – diameter of blade circle $[m]$

v – wind velocity $\left[\frac{m}{s}\right]$

The afore-given relationship states, that the variable, which has the fundamental effect on the level of power generated by a wind turbine during its work, is wind velocity. Other components of the relationship do not change so abruptly and unpredictably. It can be assumed that power generated by a wind turbine is a stochastic process, thus it requires development of new forecasting methods, appropriate for such a process. The tools based on fractal theory, already applied in many different branches of science, might become useful for this purpose [1–6, 8, 10, 12].

2. FRACTAL GEOMETRY IN TIME SERIES

For many years mathematics was unable to describe phenomena occurring in Nature. Those seemed to be too complicated until recently. Several years ago a new branch of science has emerged, which offered a new outlook towards natural phenomena. Fractal geometry has become an impulse for many researchers to review many problems unsolved so far. The fundamental notion in this field of science is “fractal”. Currently no exact definition of this object exists. Following [6] it can be stated, that a fractal is an object, whose parts fulfill a certain relationship to the total entity. Fractal is also defined as a set, which has the following features [6, 8]:

- has a nontrivial structure in any scale,
- this structure is difficult to be described using traditional Euclid geometry,
- is self-similar, if not exactly, then in the average or stochastic sense,
- its fractal dimension is larger than its topological dimension.

Two types of fractals may be distinguished [8]:

- deterministic ones (created according to a certain rule, e.g. Cantor set, Koch curve, Mandelbrot or Julia set),
- random ones (coastline, tree, lungs).

Deterministic i.e. mathematical fractals are symmetric and self-similar objects in any scale. They are generated using recurrent formulas or iterated function system (IFS). Such objects may reveal limited self-similarity due to randomness, which influences their creation.

One of the fundamental features of a fractal is its dimension. It gives us the information, how the object fills its space. A problem occurs, however, because there exists a number of methods to determine the fractal dimension, what consequently may imply different results during the analysis of the same object. When comparing fractal dimensions of different objects attention has to be paid to the method of their determination. It will allow us to assess the features of the objects correctly. Below the most often encountered dimensions are summarized [11]:

- Hausdorff
- self-similarity one
- box one
- capacity one
- information one
- Euclid dimension
- Compass one.

Each of them brings us a certain piece of information on the object. In dependence on the given object, the values of individual dimensions may vary, but sometimes they obtain the same value.

The aforementioned features and properties may be applied in studies of time series, which may be envisaged as similar to some extent to a coastline. The more magnified depiction is applied during the analysis of a coastline, the more details may be distinguished. A similar situation happens during the studies of time series, when one considers shorter and shorter time intervals. When studying time series, one can notice their very important feature – self-similarity in the statistical sense. Time series, which exhibit the feature are referred to as fractal time series [6]. It is worth noticing, that such series feature long-term correlation. Therefore it is reasonable to determine their fractal dimension, this is accomplished by studying the ruggedness of their chart using a graphical method, following the definition of the compass dimension or the box one. These methods are time-consuming, therefore in the following part of the work we have decided to use the Hurst exponent in order to determine the fractal dimension of the examined time series.

3. A METHOD FOR DETERMINATION THE HURT EXPONENT

There exists a relationship between the fractal dimension of a time series and the Hurst exponent, expressed as [6]

$$D = 2 - H \tag{2}$$

where:

D – fractal dimension

H – Hurst exponent

Availing of the formula (2), the fractal dimension of the examined time series may be determined, knowing the value of Hurst exponent.

The name of the presented statistics comes from its developer, a British hydrologist Harold Edwin Hurst, who during his work on design and development of water tanks on the Nile has discovered and described the new statistical method. The tool makes it possible to analyze time series and distinguish their kind taking into account the value of Hurst exponent. Three groups of processes may be distinguished:

1. $0 < H < 0.5$ – an anti-persistent series,
2. $H = 0.5$ – a random series, lack of correlation,
3. $0.5 < H < 1$ – a persistent series.

The theory has been extended to take into account the processes related to economy and capital markets [6]. Until today, a number of methods for determination the Hurst exponent has been developed [1– 6,10,12]. They are used in many different branches of science. A particular attention has been paid to the applications of the Hurst statistics to the issues of analysis and forecasting the market rates on capital markets [5,10], noticing their similarity to time series, representing the operation of wind turbines. One of the algorithms [10] has been chosen for adaptation to this purpose, in order to make use of it in the studies on time series, representing wind velocity and power generated by the wind turbine.

For the time series

$$X = X_1, X_2, ..., X_n \tag{3}$$

the mean value has been calculated

$$m = \frac{1}{n}\sum_{i=1}^{n} X_{\mathrm{i}} \tag{4}$$

The centering of the series has been achieved

$$Y_{\mathrm{t}} = X_{\mathrm{i}} - m, \quad t = 1,2,...,n \tag{5}$$

Accumulated time series have been created

$$Z_{\mathrm{t}} = \sum_{i=1}^{t} Y_{\mathrm{i}}, \quad t = 1,2,...,n \tag{6}$$

The range series of accumulated time series have been calculated

$$R_{\mathrm{t}} = \max\left(Z_{1,} Z_{2,}, ..., Z_{\mathrm{t},}\right) - \min\left(Z_{1,} Z_{2,}, ..., Z_{\mathrm{t},}\right), \quad t = 1,2,...,n \tag{7}$$

The average deviations have been determined

$$S_{\mathrm{t}} = \sqrt{\frac{1}{t}\sum_{i=1}^{t}\left(X_{\mathrm{i}} - u\right)^2}, \quad t = 1,2,...,n \tag{8}$$

where u is mean value of the series counted from X_1 to X_t

The mean rescaled range has been calculated

$$\left(\frac{R}{S}\right)_{\mathrm{t}} = \frac{R_t}{S_t}, \quad t = 1,2,...,n \tag{9}$$

of the subsets

$$[X_1, X_t], [X_{t+1}, X_{2t}], ..., [X_{(m-1)(t+1)}, X_{mt}] \tag{10}$$

where m is the ratio $\frac{n}{t}$ rounded down to the nearest integer.

Hurst has formulated the relationship

$$\left(\frac{R}{S}\right)_{\mathrm{t}} = ct^{H} \tag{11}$$

where:

c – a constant

H – the Hurst exponent

Applying the logarithmic transformation of the formula (11) one obtains

$$\log\left(\frac{R}{S}\right)_{\mathrm{t}} = \log c + H \log t \tag{12}$$

Applying linear regression by the least square method, the inclination of the logarithmic chart of the rescaled range with respect to the axis of time logarithms has been determined, what corresponds to the value of the sought Hurst exponent.

4. PRACTICAL APPLICATION OF HURST STATISTICS

Following the discovery by Hurst, that most of natural phenomena is not subject to random walking, as well as the similarity of the chart depicting the operation of a wind turbine to time series of market price rates for market companies, one of the methods to determine the Hurst exponent has been applied to time series, representing the output power generated by the wind turbine, what shall make it possible to use the obtained piece of information for forecasts. The calculations have been carried out using the afore-described algorithm. Fig. 1 depicts an exemplary operation routine of the wind power station (10 min sampling period) as well as the Hurst exponent for this routine, determined stepwise.

Analyzing the examined time dependencies it can be stated, that they exhibit features typical for persistent series. The value of Hurst exponent, in most cases, keeps above 0.8, therefore the process itself has a strong tendency to strengthen its trend.

Another aspect should be considered in detail at this point. High values of Hurst exponent do not always indicate the persistent character of the process. According to [11] there are two scenarios which can be followed to explain the situation, when the value of Hurst exponent differs to a large extent from 0.5. Either in the considered series there exists a memory effect and each observation is correlated to subsequent ones or the carried out analysisis wrong, and the exponent value does not indicate the existence of memory in the process. There is a method for verification of the obtained results [11]. Having determined the exponent value for the considered sample, it is necessary to mix the observations in such a way, that the sequence of the new series would be different for the one in the input series. Next the value of Hurst exponent is determined again for the new series. If the original series is independent, then the result should be the same, because there is no memory effect in it. In the case, when the obtained new exponent value is much closer to 0.5, it follows, that in the examined series there exists a memory effect, because data swap has destroyed the process structure.

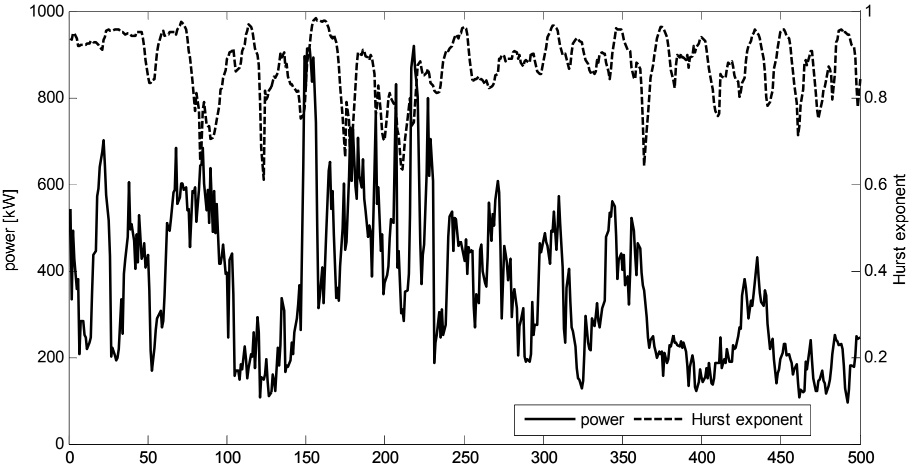


Fig. 1. Power generated by the wind turbine and the determined Hurst exponent

Two tests of this kind have been carried out. They indicate the existence of memory effect in the processes related to work of wind turbines. An example is given in Fig. 2. Data representing power generated in real time by an individual wind turbine are presented. The value of Hurst exponent of the examined sample was equal to 0.78, after data swapping and subsequent calculations the value 0.49 has been obtained. After change in sequences have been introduced, a random series has been obtained, which did not reveal any dependencies between subsequent observations. Therefore the initial process possessed the memory effect, as it follows from the exponent value.

In Fig. 3 the results of analogous analyses carried out for the whole wind farm are presented. After application of the series swapping method, new results have been obtained, which allowed us to draw basically the same conclusions as in the case of an individual turbine.

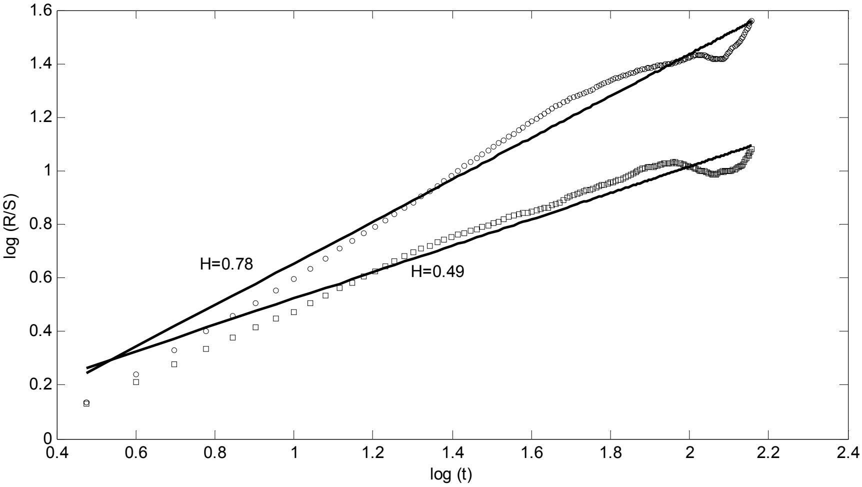


Fig. 2. Hurst exponent of time series describing the work of individual wind turbine in two cases: non-mixed one and swapped one.

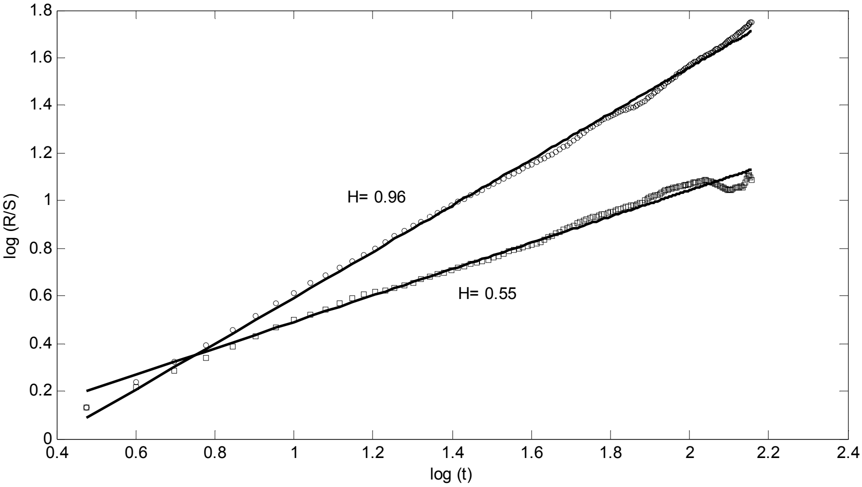


Fig. 3.The Hurst exponent of time series, which describes the work of a wind farm for two cases: a non-mixed one and a swapped one

When comparing Figs. 2 and 3., attention should be paid to the value of Hurst exponent. The value calculated for the whole wind farm is higher than the one determined for the individual turbine in the same time period. It brings us a new piece of information. The Hurst exponent is a measure of raggedness of a time series. The higher is its value, the chart for the process is smoother, thus the series itself approaches a deterministic one. This statement has found its confirmation during the analyses of work of wind turbines. Individual turbines feature much higher variations of generated power in time in comparison to the whole wind farm. It follows from the fact, that factors influencing the level of generated power have a much stronger impact on individual units than on their sets.

5. CONCLUSIONS

The afore-given examples confirm the tendency observed during the carried out analyses. They allow us to state, that in the time series, representing power generated by an individual turbine or a wind farm there exists a memory effect. Therefore there exist different factors, which have an impact on the examined time dependencies. These may result both from technical reasons related to construction and maintenance of individual wind turbines, as well as from atmospheric factors, that influence the operation of the power station. Mostly this is related to wind direction and velocity, but nonetheless other factors are also important, e.g. icing of rotor blades, which limit the control capabilities of the turbine.

The evidence of the memory effect in the examined time series indicates, how important is to carry on with further research on the issue. In the forthcoming work, the strength of loads influencing individual time dependencies should be checked in the first place, because it may be different for individual turbines or farms.

The knowledge of the memory scope in the examined time series and their reasons will make it possible to start development of a forecast model, which takes into account the fractal nature of the processes, what should result in improvement of forecasts for work of wind power stations.

REFERENCES

[1] BASSINGTHWAIGHTE J, B., RAYMOND G. M., *Evaluating rescaled range for time series,* Annals of Biomedical Engineering, Vol. 22, No. 4, 1994, 432–444.

[2] CONNIFFE D., SPENCER J. E., *Approximating the distribution of the R/S statistic,* The Economic and Social Review, Vol. 31, No. 3, 2000, 237–248.

[3] GILMORE M., RHODES T. L., PEEBLES W.A, YU C.X., *Investigation of rescaled range analysis, the Hurst exponent, and long-time correlations in plasma turbulence,* Physics of Plasmas, Vol. 9, No. 4, 2002, 1312–1317.

[4] KRUGER S. E., MATOS O., MARCOS J., MAURICIO J., DE MOURA E.P., REBELLO A., *Rescaled range analysis and fluctuation analysis study of cast irons ultrasonic backscattered signals,* Chaos, Solitons & Fractals, Vol. 19, No. 1, 2004, 55–60.

[5] LO A. W., *Long-term memory in stock market prices,* Econometrica, Vol. 59, No. 5, 1991, 1279–1313.

[6] PETERS E. E., *Chaos And Order In The Capital Markets – A New View Of Cycles, Prices, And Market Volatility, Second Edition,* New York, John Wiley & Sons, 1996.

[7] POPŁAWSKI T., DĄSAL K., ŁYP J., SZELĄG P., *Zastosowanie modeli ARMA do przewidywania mocy i energii pozyskiwanej z wiatru* (*Application of ARMA model*s for prediction of power and energy obtained from wind – in Polish)*,* Polityka Energetyczna (Energy Policy), Vol. 13, No. 2, 2010, 385–400.

[8] POPŁAWSKI T., SZELĄG P., *Analiza fraktalna w prognozowaniu procesów samopodobnych,(Fractal analysis in forecasting of self-similar processes – in Polish),* Śląskie Wiadomości Elektryczne (Silesian Electric News), Vol. 17, No. 5, 2010, 30–33.

[9] POPŁAWSKI T., SZELĄG P., *Wykorzystanie modelu Prigogine'a do długoterminowej predykcji szczytów obciążeń w Krajowym Systemie Elektroenergetycznym*, *(The use of Prigogine's model for long-term peak prediction in The Polish Power System – in Polish)*, Rynek Energii (Energy Market), Vol. 86, No. 1, 2010, 32–36.

[10] QIAN B., RASHEED K., *Hurst exponent and financial market predictability,* from Proceeding: Financial Engineering and Applications, Cambrige, USA, 8–10 November 2004.

[11] STEWART I., *Czy Bóg gra w kości? Nowa matematyka chaosu (Does God play dice? New mathematics of chaos),* Warszawa, PWN, 2001.

[12] TAQQU M. S., TEVEROVSKY V., WILLINGER W., *A critical look at Lo's modified R/S statistic,* Journal of Statistical Planning and Inference, Vol. 80, No. 1–2, 1999, 211–227.

Manning formula, a fulfilling height,
a hydraulic radius, wastewater network,
self – purification process.

Lucyna BOGDAN*, Grażyna PETRICZEK*

PROBLEMS OF WASTEWATER NETWORK MODELLING**

In the work the basic problem connected with modeling of waste network are presented. Methods of modeling of basic sewage variables and calculation algorithms are described.

The problems concern the gravitation branched off network divided by nodes into sectors. The nodes are the points of connection of several network segments or branches or the points of changing of network parameters as well as the location of sewage inflow to the network.

The presented algorithm for hydraulic calculations concerns the houskeeping or combined sewage net. It is also assumed that the net is gravitational in the most of the segments except the cases of canal segments operating under the pressure (pumping stations). It is assumed that the segments parameters such as the shape, the canal dimension, the bottom slope or roughness are constant. The sewage inlet takes place in net nodes. Because of these assuming all the relations concerns the steady state. The calculations of flows velocities and of the fillings heights in the segments of the wastewater net are carried for the known slopes and known diameters of the canals.

Additionally the problem of canal self-purification is considered.

1. CHARACTERISTIC OF A SEWAGE SYSTEM

Taking into account a design and the operating processes we can distinguish the following sorts of sewage: a housekeeping sewage, a industrial sewage, a rain wastewater, a drainage sewage, a ground water.

The following sewage systems can be marked out depending on the kind of wastewater dump:

* Systems Research Institute, Polish Academy of Science, 01-447 Warsaw, Newelska 6.

** The work was prepared within the framework of Project 6521/B/T02/2011/40 of Polish National Center of Science (NCN).

a) a combined sewage system
b) a separate sewage system
c) a semi-separate sewage system

In the universal sewage system (combined system) (Fig. 1) the all kinds of waste water are led using the common channels

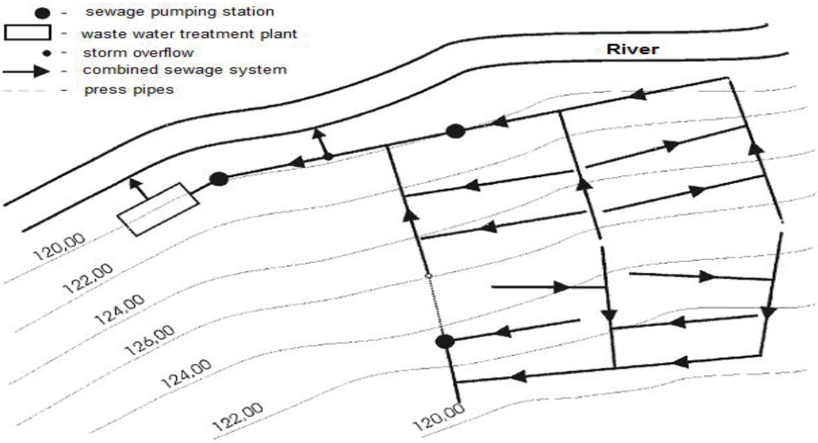

● - sewage pumping station
□ - waste water treatment plant
• - storm overflow
→ - combined sewage system
- - press pipes
River
120,00
122,00
124,00
126,00
124,00
122,00
120,00

Fig. 1. The scheme of combined sewage system

At present time the separate sewage system is mostly used, in which there are two separate sewage nets::

a) a sewage net, used for the housekeeping sewage and for the industrial sewage
b) a rainwater net for carrying out the rain-wastewater

A dimensioning of intersections is done according to a reliable rain. The rain waste-water canals are designed for the 100% filling.

The semi-separate sewage system is a system with two nets: the housekeeping and the rain water one. In this system the sewage net can receive a part of the rain run-off.

2. THE ALGORITHM FOR THE CALCULATION WASTEWATER NETWORK

The presented algorithm for hydraulic calculations concerns the houskeeping or combined sewage net shown in Fig.1, branched, divided on segments by nodes. The nodes are the points in which few segments or branches are connected or the change of parameters or sewage inflow occurs (a sink basin, rain inlets, a connecting basin). In connecting nodes the flow balance equations and the condition of levels consistence are satisfied.

It is assumed that the segments parameters such as the shape, the canal dimension, the bottom slope or roughness are constant. The sewage inlet takes place in net nodes. Because of these all the relations concerns the steady state.

It is also assumed that the net is gravitational in the most of the segments except the cases of canal segments operating under the pressure (pumping stations).

The calculation scheme is presented below for the canals with circular section.
The canal hydraulic calculation concerns in:

- the determining of the canal filling in each wastewater network segment
- determining of the mean flow velocity for the reliable canal segment
- the determining of the hydraulic radius

The calculations are carried for each segment between the nodes taking into account the maximal flow value and canal slope for the down node.

The algorithm of wastewater network should enable the determining of the hydraulic parameters (the canal filling and the mean flow velocity).

The algorithm for the calculation wastewater network consists of following fundamental steps:

Step 1.The net structure data, i.e. number of nodes, number of segments N, the set of nodes, the set of segments, the set of diameters $\{d_i\}$, the set of segment lengths L_i, slopes for the segments J_i, $I = 1,\ldots, N$, the roughness k_i for each segment, initial values of the rate flow Q_i for each segment $I = 1,\ldots,N$, the kinematical flow resistance, gravitational acceleration should be read.

Step 2. Calculate the rate of inflow for respective sewage net nodes. Depending on the kind of the sewage the rate of inflow for each segment is calculated using the proper relation.

For the hauskeeping and industrial sewage the maximum hour inflow Q for the given segment is considered and can be calculated according the relation:

$$Q_{h\max} = \frac{N_{h\max} M \cdot q_{śr}}{24} \tag{1}$$

where:M – the number of residents for the given segment of the net
$q_{śr}$ – the average unit runoff
N_{hmax} – the rate of irregularity for twenty four hours
The rain waste inflow can be expressed:

$$Q = q_d \cdot \psi \cdot F \cdot \varphi \tag{2}$$

where

- Q – the rain waste inflow [dm^{3}l/s]
- F – the area of drainage basin for the canal segment [ha]
- Ψ – the rate of run-off (the ratio of the amount of the rain waste in canals to the amount of rain waste from the given area)
- φ – the rate of the delay
- q_d – the rain intensity

Step 3. For given value of rate inflow Q_i in particular segments i = 1,….,N, one can determine the following values: the filling heights H_i , the hydraulic radius R_h and the flow velocities v_i.

The calculations presented in this step concern segments of gravitational sewage net.

According to the Manning formula the flow velocity depends on hydraulic radius R_h. The hydraulic radius R_h depends on the filling height.

From the Manning formula and taking into account canal geometric one obtains the following relation:

$$J^{\frac{3}{2}} \cdot d^8 \cdot \left(\pi - \frac{\pi}{360} \cdot \varphi + \frac{1}{2} \sin \varphi\right)^5 \bullet \left(\pi - \frac{\pi}{360} \cdot \varphi\right)^{-2} - 4^5 \cdot Q^3 \cdot n^3 = 0 \tag{3}$$

$$\varphi = 2 \cdot arctg\left(\frac{\sqrt{Hd - H^2}}{H - \frac{d}{2}}\right) \tag{4}$$

where:

- H – the filling height
- r – the radius of circular canal
- φ – the central angle
- d – the inside canal diameter.

Solving equation (3) we obtain the filling height as function of rate flow Q. The equation has the form: f(H)=0.

Regarding the formula (3) the Newton method is used for calculating of the canal filling assuming the initial filling value for example $H_p = 0{,}7d$

$$H_{k+1} = H_k - \frac{f(H_k)}{f'(H_k)} \tag{5a}$$

where; the particular elements can be written as:

$$\varphi_k = 2 \cdot \mathrm{arctg}\left(\frac{\sqrt{H_k d - H_k{}^2}}{H_k - \frac{d}{2}}\right) \tag{5b}$$

$$f(H_k) = J^{\frac{3}{2}} \cdot d^8 \cdot \left(\pi - \frac{\pi}{360} \cdot \varphi_k + \frac{1}{2}\sin(\varphi_k)\right)^5 \cdot \left(\pi - \frac{\pi}{360} \cdot \varphi_k\right)^{-2} - 4^5 \cdot Q^3 \cdot n^3 \tag{5c}$$

The convergence condition of this method is the satisfying of the relation:

$$\left|\frac{f(H_k) \cdot f''(H_k)}{(f'(H_k))^2}\right| \leq K < 1 \tag{6}$$

The stop criterion can be written as:

$$|H_{k+1} - H_k| <= \alpha \tag{7}$$

α<1 – the given convergence parameter

The iterations of calculation of the height of fulfilling H are done until the convergence condition is satisfied (7).

Step 4. For the filling *H* calculated in step 2 the hydraulic radius R_h should be determined according to the formula:

$$R_h = \frac{A}{U} = \frac{\pi d - \frac{\pi d}{180} \cdot arctg\left(\frac{\sqrt{Hd - H^2}}{H - \frac{d}{2}}\right) + \frac{d}{2} \cdot \sin\left(2arctg\left(\frac{\sqrt{Hd - H^2}}{H - \frac{d}{2}}\right)\right)}{4 \cdot \left(\pi - \frac{\pi}{180} \cdot arctg\left(\frac{\sqrt{Hd - H^2}}{H - \frac{d}{2}}\right)\right)} \tag{8}$$

where:

A – the collision cross-section area

U – the wetted profile.

Step 5. The velocity should be received from:

$$v=\frac{1}{n}\cdot R_h^{\frac{2}{3}}\cdot J^{\frac{1}{2}} \tag{9}$$

Because of the lack of the Manning rate $K=\frac{1}{n}$ for the canals made of various kinds of plasics the calculation of its value should be done according to the relation:

$$K=4\sqrt{g}\cdot\left(\frac{32}{d}\right)^{\frac{1}{6}}\cdot\log\left(\frac{3{,}7\cdot d}{k}\right) \tag{10}$$

where k is absolute roughness rate.

Step 6. The equations of flow balance $\sum_{j\neq i} Q_j=0$ and the conditions of equality of surfaces levels are calculated in each net node.

Step 7. For the pumping channels for the given flow Q the pressure losses h_s are calculated as follows:

$$h_s=\frac{\lambda}{\pi^2 d_i^5\cdot g}\cdot 8L_i\cdot Q_i^2+\frac{8}{\pi^2 d_i^4\cdot g}\cdot\xi_i\cdot Q_i^2 \tag{11}$$

where:

λ_i – the coefficient of linear losses for the i-th segment calculated using Colebrook–White’s or Walden’s formula

L_i – the length of the i-th segment

ξ_i – the coefficient of local losses in the canal

Because of confounding relation λ coefficient in Colebrook–White’s formula, the Walden’s simplified formula is used:

$$\frac{1}{\sqrt{\lambda}}=-2\cdot\log\left(\frac{6{,}1}{\mathrm{Re}^{0{,}915}}+0{,}268\cdot\varepsilon\right) \tag{12}$$

where :

ε – the related roughness $\varepsilon=\frac{k}{d}$

Re –Reynolds number $\mathrm{Re}=\frac{v\cdot d}{\upsilon}$

υ – the coefficient of kinetic viscosity

The calculations are carried out for each wastewater net segment beginning from the farthest one until the nearest segment, taking into account the wastewater treatment.

The calculations are done for each segment of the net begining from the farthest one taking into account the net outlet. The last calculations are carried for the segment nearest to the wastewater treatment plant.

3. THE CONDITIONS OF SELF-PURIFICATION IN CANAL SYSTEM

The flow velocity can be calculated using the formulas or tables having the diameters values of canal pipes. The diameters should have the minima values: for sewage canals – 160 mm, for rain canals – 250 mm, for general sewage system – 300 mm. The velocity of the self-purification can be interpreted as [3]

- **not sliming velocity**, countering the falling of suspensions and sludge on the canal bottom
- **blurring velocity**, which makes the sludge move off on the canal bottom

The turbulent character of the flow is connected with differentiated velocity distribution in the active canal section. The laminar character of the flow occurs only in near-wall area of the section and only there the velocity vectors are parallel to the canal axis and theirs value is constant.

These velocities are relevant in respect of self-purification because they ensure the movement of sludge particles on the bottom.

In [7] the hydro-transport method is used, assuming solid particles in sewage. They are assumed to have determined dimensions and minimal tangent strains sufficient to initiate the particles movement. The strains are connected with the friction occurring during the particles flow in the canal.

The tangent strain τ understood as the average tangent strain on the whole wetted surface of the canal can be described as:

$$\tau = \frac{A}{U}\rho \cdot g \cdot \sin\alpha \tag{13}$$

where:

A –active section of the canal

U – wetted profile

ρ – the density of sewage

g –gravitational acceleration

α – the angle of inclination of the canal axis

For the small values of the angle α, typical for the the plain area, this formula can be written as:

$$\tau = R \cdot \rho \cdot g \cdot J \tag{14}$$

Transforming this formula we receive the formula for the canal slope

$$J = \frac{\tau}{\rho \cdot g \cdot R} \tag{15}$$

After inserting it into Manning formula we get the expression for the average velocity required in the case of the transport ability of the stream

$$v = \frac{1}{n}\left(\frac{\tau}{\rho \cdot g}\right)^{\frac{1}{2}} \cdot R^{\frac{1}{6}} \tag{16}$$

If the slope in (16) is smaller than slope of the ground, then there is the possibility to design the canal system parallel to the ground gaining the canal slope larger than one required for the self-purification.

Designing the combined sewage system, Yao recommends [8] the tangent strain values larger than 1 N/m2 because then the particles with diameters 0,2–1,0 mm should be removed.

The main canal pipes producers recommend tangent strains larger than 2,25 N/m2 in combined sewage systems and larger than 1,35 N/m2 in rain sewage systems.

Another approach can be found in [3] i [6]. Starting from Apollov formula describing the hydrodynamic thrust pressure

$$P = k \cdot F \cdot \gamma \frac{u_p^2}{2g} \tag{17}$$

where:

u_p – the demersal velocity

k – the coefficient of the particles shape (for spherical k = 0,75, for cubical k = 1,46, for gravel k = 1)

F – the area of projection of particles to the plan perpendicular to the velocity vector

γ – specific gravity

there is possibility of calculation of **not sliming velocity u_1** and **blurring velocity u_2** from the expressions:

$$P_1 = k \cdot F \cdot \gamma \frac{u_1^2}{2g} \tag{18a}$$

and

$$P_2 = (1-b)k \cdot F \cdot \gamma \frac{u_2^2}{2g} \tag{18b}$$

where b<1 is the decreasing coefficient.

Assuming the simplification for small angles of slope of canal axis the following the simplified formulas for limiting demersal velocities for the spherical particle flow with diameter δ and with specific gravity γ_p are received:

$$u_1 = \frac{\sqrt{4g \cdot \delta \cdot (\gamma_p - \gamma) \cdot \eta}}{3k \cdot \gamma} \tag{19a}$$

$$u_2 = \frac{\sqrt{4g \cdot \delta \cdot (\gamma_p - \gamma) \cdot \eta}}{3k \cdot \gamma \cdot (1-b)} \tag{19b}$$

where: η – the friction coefficient.

Using these relations the formulas for the average flow velocities for the δ dimension particle transportation trailing along the canal with the diameter d are received.

The average not sliming velocity is formulated as:

$$v_1 = \frac{u_1}{\varepsilon} \tag{20}$$

and the average **blurring velocity** is given as :

$$v_2 = \frac{u_2}{\varepsilon} \tag{21}$$

where ε – the coefficient of the velocity distribution in the canal cross-section.

The result of the calculations [4] is that the average velocities calculated for the slopes from the formula

$$J = \frac{\tau}{\rho \cdot g \cdot R}$$

for different tangent strains values τ are not smaller than the limiting velocities determining according to formulas (20) and (21).

REFERENCES

[1] BIEDUGNIS S., *Metody informatyczne w wodociągach i kanalizacji*. Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa 1998.

[2] BŁASZCZYK P., *Opory w zamkniętych kanałach ściekowych*. Nowa Technika w Inżynierii Sanitarnej – Wodociągi i Kanalizacja, 3, 1973.

[3] BŁASZCZYK W., STAMATELLO H., BŁASZCZYK P., *Kanalizacja. Sieci i pompownie.*Tom I. Arkady. Warszawa. 1983.

[4] BOGDAN L., PETRICZEK G., *Zagadnienia przepływu zanieczyszczeń w kanalizacji*, w druku.

[5] CHUDZICKI J., SOSNOWSKI S., *Instalacje kanalizacyjne*. Wydawnictwo Seidel-Przywecki Sp. z o.o, Warszawa 2004.

[6] DOLECKI J., USAKIEWICZ A., *Model szacowania wzkaźników zanieczyszczeń w spływach w kanalizacji deszczowej*. Ochrona Środowiska. Nr 3–4 (36–37), pp. 77–80, Wrocław, 1988.

[7] KARNOWSKI J.M., *Warunki transportu wleczonych części mineralnych w przewodach kołowych o dowolnym nachyleniu*. Materiały Konferencji Naukowo-Technicznej PZITS. Poznań. 1973.

[8] KWIETNIEWSKI M., NOWAKOWSKA –BŁASZCZYK A., *Obliczenia hydrauliczne kanałów ściekowych na podstawie krytycznych natężeń stycznych*. Nowa Technika w Inżynierii Sanitarnej – Wodociągi i Kanalizacja, 13, 1981.

[9] PUCHALSKA E., SOWIŃSKI N.: *Wymiarowanie kanałów ściekowych metodą krytycznych naprężeń stycznych*. Ochrona Środowiska. Nr 3–4, pp. 53–62, Wrocław, 1984.

[10] ŁOMOTOWSKI J., SZPINDOR A.: *Nowoczesne Systemy oczyszczania ścieków.* ARKADY. Warszawa 1999.

[11] WARTALSKI A., WARTALSKI J.: *Projektowanie hydrauliczne rurociągów z tworzyw sztucznych*. Ochrona Środowiska. Nr 1(76), pp. 19–24, Wrocław, 2000.

PART 3

IMAGE PROCESSING
AND PATTERN RECOGNITION

*General Shape Analysis, 1-D Fourier Descriptors,
Shape Representation, Coarse Shape Classification*

Dariusz FREJLICHOWSKI*

THE GENERAL SHAPE ANALYSIS BY MEANS OF THE 1-DIMENSIONAL FOURIER DESCRIPTORS

In the chapter one of the oldest and most widely used algorithms for contour shape representation – Fourier Descriptors (FD) – is applied to the problem of General Shape Analysis (in short, GSA) and experimentally evaluated. This problem is similar to both the recognition and retrieval of shapes. From the first of those two tasks the searching for similar objects is taken. From the second one – the presentation of more than one resultant object to the user. Moreover, the analysed shape can be similar to one of the template classes and does not have to belong to any of them. It means that the most general information about a shape is concluded by means of the GSA, i.e. how square, round, triangular, etc. it is.

The results on the application of the 1D Fourier Descriptors to the General Shape Analysis were evaluated by means of the inquiry forms filled out by 187 persons (63 women and 124 men). The result (the most popular indications at the particular places) was treated as a benchmark. The results obtained using the investigated shape descriptor were compared with this benchmark.

1. INTRODUCTION

The idea of General Shape Analysis (in short, GSA) was introduced in [6] and more minutely described and investigated in [7]. It works similarly both to the traditional retrieval and recognition of shapes, when performed by means of the template matching. There is however an important novel property of the GSA – a class of the templates is represented only by one object. Moreover, those templates are very general and there are only a few of them. Usually only about ten to twenty general shapes are applied as the templates. In the General Shape Analysis the main idea is not to exactly identify an ob-

* West Pomeranian University of Technology, Szczecin; Faculty of Computer Science and Information Technology, Żołnierska 52, 71-210 Szczecin

ject under processing, but instead – to conclude some general information about it. For example, if the triangle, the circle, and the square are some of the elements composing the database, using the GSA we can obtain the degree in which the object subject to analysis is similar to the enumerated basic shapes. The discussed problem is briefly presented in Fig.1.

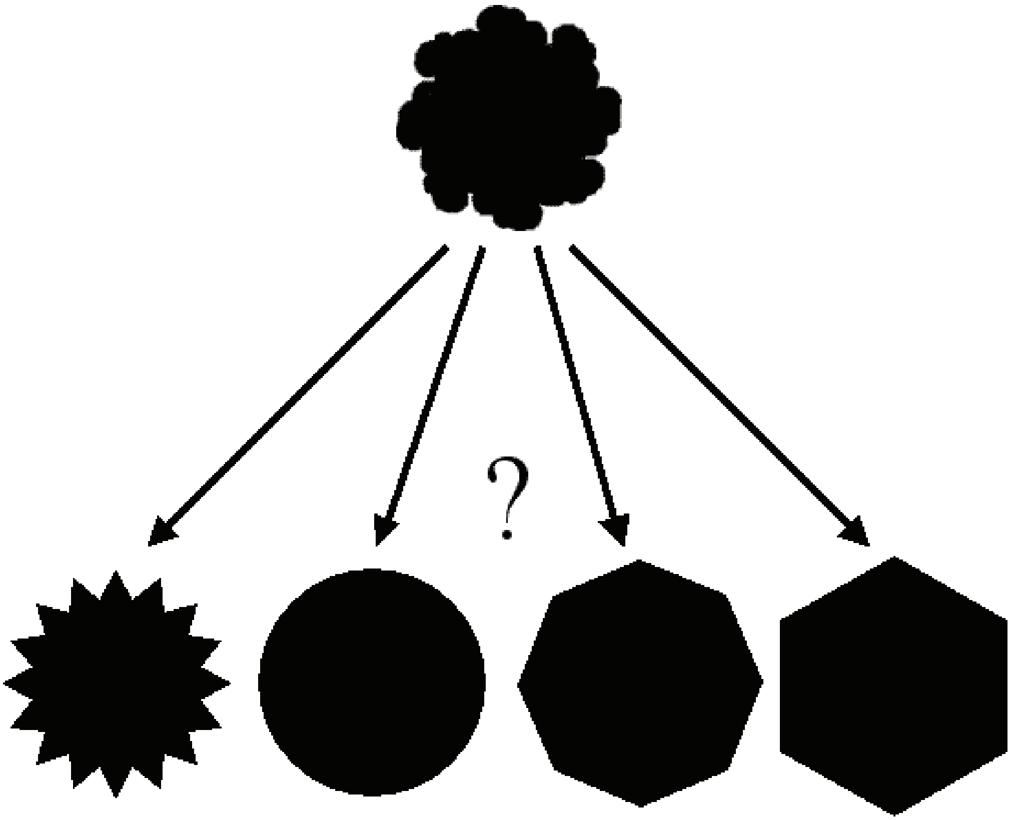


Fig.1. An illustration of the General Shape Analysis – which general shape is the most similar to the one being processed?

The definition of the General Shape Analysis makes some specific applications possible. Roughly speaking, in those applications the analysis of shapes is formulated on a higher level of abstraction. One can conclude the general information describing a shape. One example is the identification of a type of a stamp (e.g. official, medical, public, institutional) in searching for probable false digital documents stored on a hard drive. Some experiments concerning this specific problem have already been performed. Their results were described in [8] and [9]. Obviously, some another examples can be easily found, for example the coarse classification in large databases or the linguistic shape retrieval in multimedia databases, etc.

In the presented research results the Fourier Descriptors were applied to the General Shape Analysis problem. The one-dimensional version of this approach, designed for the contour shape, was used. The idea of the application of the descriptor to the mentioned problem arises from the fact that it was successfully applied in many practical aspects of shape representation, retrieval and recognition. Hence, it was evaluated in relation to a completely new problem.

The remaining part of this contribution is organised as follows. The second section describes the mathematical formulations and exemplary applications of the Fourier Descriptors. The third section provides the experimental results and their discussion. Finally, the last section concludes the chapter.

2. FUNDAMENTALS AND EXEMPLARY APPLICATIONS OF THE FOURIER DESCRIPTORS

The Fourier transform has been successfully applied in the area of pattern analysis and recognition, including shape representation and identification. Although the Fourier theory can be applied to shape recognition in many different ways, its usage can be divided into two main groups, basing on how the represented shape is treated. If the extracted contour of a shape is treated as a signal, the one-dimensional transform is used. If the object within a bitmap is taken, the two-dimensional Fourier transform can be applied. Here, the first case is considered. The most important advantage of the Fourier descriptor in the general shape analysis problem results from an important property of the transform ([13]): “The lower frequency descriptors store the information about the general shape and the higher frequency descriptors store the information about the smaller details of the image. Therefore, the lower frequency components of the Fourier descriptor define the rough shape of the original object.”

The Fourier transform of a continuous function of a variable *x* is given by the formula ([13]):

$$F(\mathrm{u}) = \int_{-\infty}^{\infty} f(x) e^{-\mathrm{j}2\pi ux} dx, \tag{1}$$

For discrete images the Discrete Fourier Transform (DFT) is used and a different formula can be applied ([13]):

$$F(u) = (\frac{1}{N})\sum_{x=0}^{N-1} f(x)e^{\frac{-j2\pi ux}{N}}, \tag{2}$$

where N is the number of equally spaced samples.

The Fourier descriptors have been applied to many problems. The most popular is the representation and recognition of shapes in general (e.g. [14], [24]). Similarly, the retrieval of shapes can be often found as well (e.g. [20], [34], [35], [36]). The 3D object representation is also popular (e.g. [10], [26], [30]). However, some more specific applications can be easily enumerated, e.g. automatic gait recognition ([22]), airplanes identification ([25]), recognition of characters ([15], [18], [21], [31]), or digits ([4]), description of molecular surface shape ([17]), description of objects in speech spectrograms ([27]), shape interpolation ([2]), identification of engineering objects ([23]), image alignment ([3]), particle shape comparison ([29]), silhouette encoding for videoconferencing ([1]), analysis of lip-contours ([11]), classification and numbering of teeth in dental bitewing images ([19]), leaves classification ([28]), human profile ([32]) and silhouette identification ([5]), coin identification ([12]), chromosome classification ([33]), contour map analysis ([16]). As one can notice the area of applications of the Fourier Descriptors is indeed wide, and the above review can be considered as a top of an iceberg only.

3. THE DESCRIPTION OF THE EXPERIMENTS AND THEIR DISCUSSION

The behavior of the Fourier Descriptors in relation to the problem of General Shape Analysis was experimentally evaluated by means of the standard test database, which has been previously used for this purpose, starting from the results described in [6]. This database is composed of fifty various shapes and divided into two groups. The first one includes ten general shapes (the rectangle, the square, the trapezium, the pentagon, the hexagon, the triangle, the circle, the ellipse, the cross and the star). The second group of shapes covers forty test objects. All shapes are stored in bitmaps, 200 × 200 pixel size. In Fig. 2. the division of the objects into templates and test shapes is presented. The Fourier Descriptor for particular shape was calculated by means of the outer contour, as usually in the case of the one-dimensional version of this method.

The method of indicating the templates was based on the template matching approach. Firstly, all general shapes were represented by means of the explored descriptor

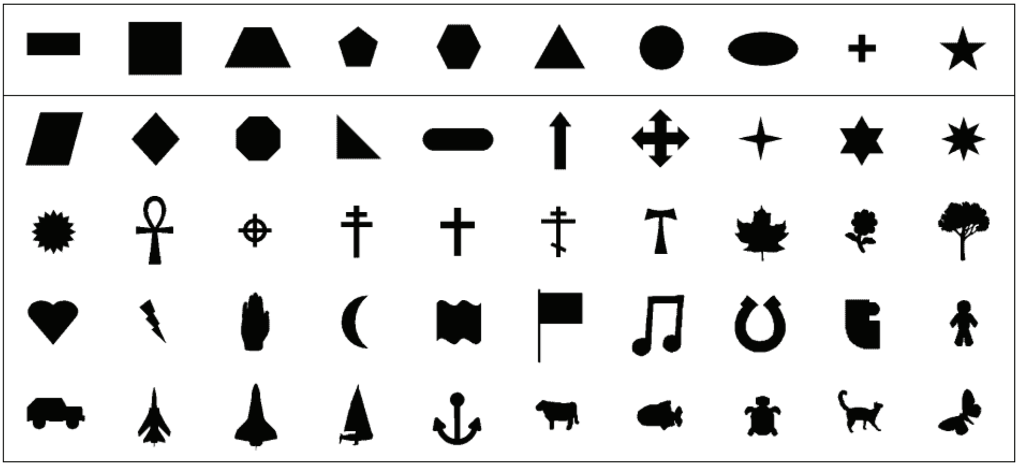


Fig. 2. The division of shapes applied in the experiments into 10 templates (first row) and 40 test objects (rest)

(namely, 1-D Fourier Descriptors) and stored. Later, each tested object was firstly described in the same way as the templates, and then matched with the descriptions of the general shapes. Three templates with the smallest dissimilarity measures, calculated by means of the Euclidean distance, were selected as the general shapes the most similar to the test shape being processed. The results are provided in Fig.3. In each instance a test shape and three general shapes selected by means of the Fourier Descriptors are presented.

As we can see in Fig.3. some of the results seem to be correct. Let the first indications be analysed. For example object no.1 is indeed very similar to the rectangle. The octagon (object no. 3) is most similar to the hexagon. The perpendicular triangle (no.4) is close to the template triangle. The flags (objects no. 30 and 31) can be linked to the square. And so on. However, the above discussion is not objective, since we are only guessing if a given result is correct or not. Therefore, in order to achieve more objective measure 187 persons (63 women and 124 men) have filled out an inquiry form. It was identical with the test performed by the analysed algorithm. It helped in the investigation of the manner in which humans realise the problem of General Shape Analysis. The result (the most popular indications at the particular places) could be used as a benchmark. It is provided in Fig.4.

Having in mind the results of the general shape analysis performed by humans we can attempt to evaluate the investigated shape descriptor. This task will be performed in two different ways. Firstly, only the percentage of proper indications in comparison to benchmark human results will be presented, separately for the three firstly selected templates. However, the analysis of the inquiry forms has provided a conclusion that in many cases the indications made by humans varies. In some of them the sequence_is different, yet the selected set of templates remains the same. Hence, the second method for comparison of the human and artificial results has not taken into account the sequence, only the presence of the particular artificial result in the whole three element benchmark set. Both cases are numerically provided in Tab.1. The result for the most difficult approach is far from the ideal. However, when considering the second approach for the comparison, as a result of the ambiguity of the human results, they are much better.

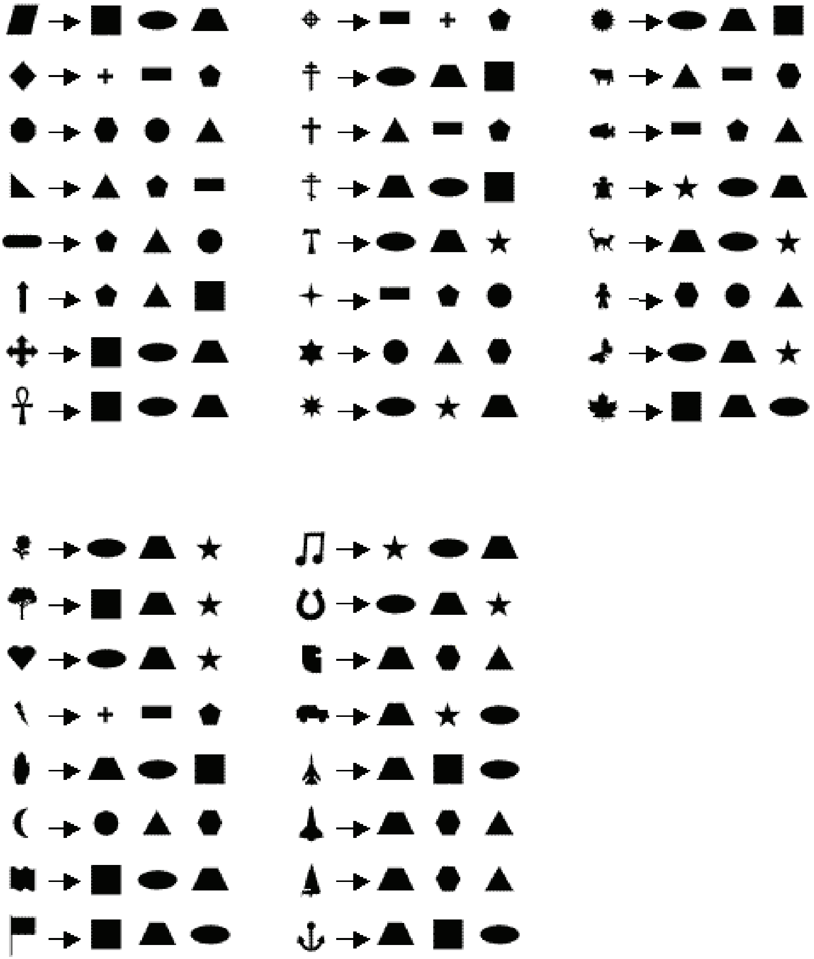


Fig. 3. The results of the experiment on General Shape Analysis using 1D Fourier Descriptors

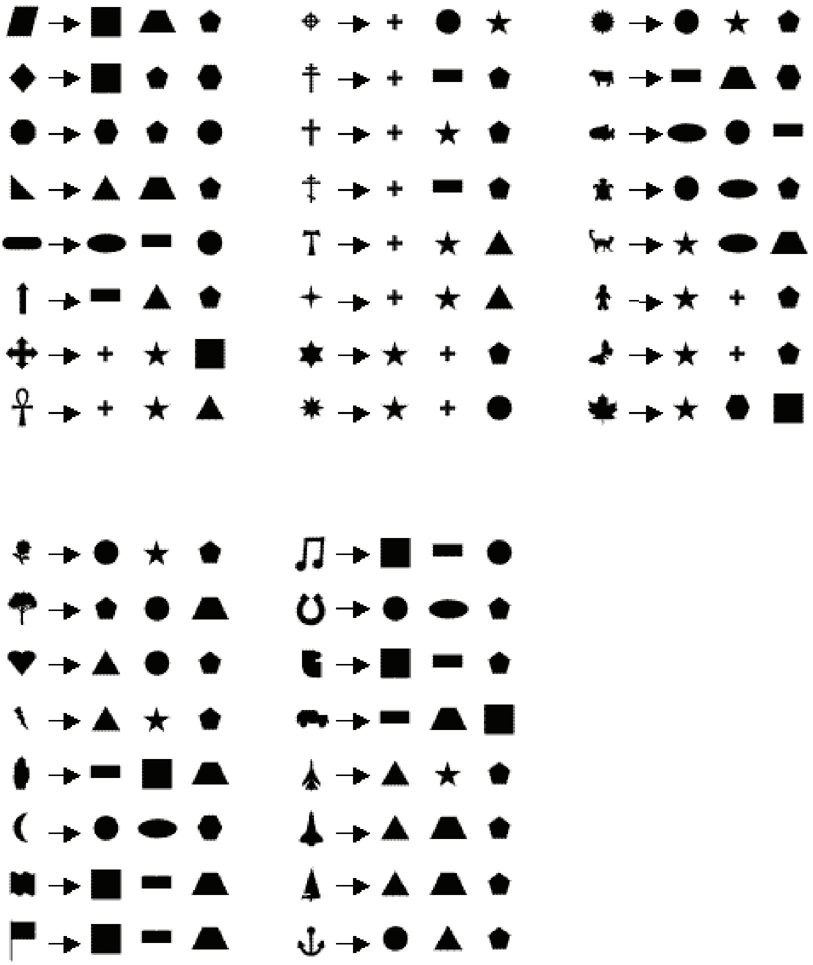


Fig. 4. The results of the General Shape Analysis test performed by humans

Table 1. The percentage of the convergence between the Fourier Descriptors and benchmark results provided by humans

The comparison method	1st indication	2nd indication	3rd indication
1. Taking the sequence in mind	15%	10%	7.5%
2. The sequence is not considered	40%	27.5%	37.5%

4. SUMMARY

In the contribution, the 1-Dimensional Fourier Descriptors, which is a very popular shape description algorithm, was investigated in the problem of General Shape Analysis. The results described in the previous section seem to be correct. In some cases, however, far from the ideal. Hence, future works on the General Shape Analysis problem include mainly the application of other algorithms in order to develop a better method for it.

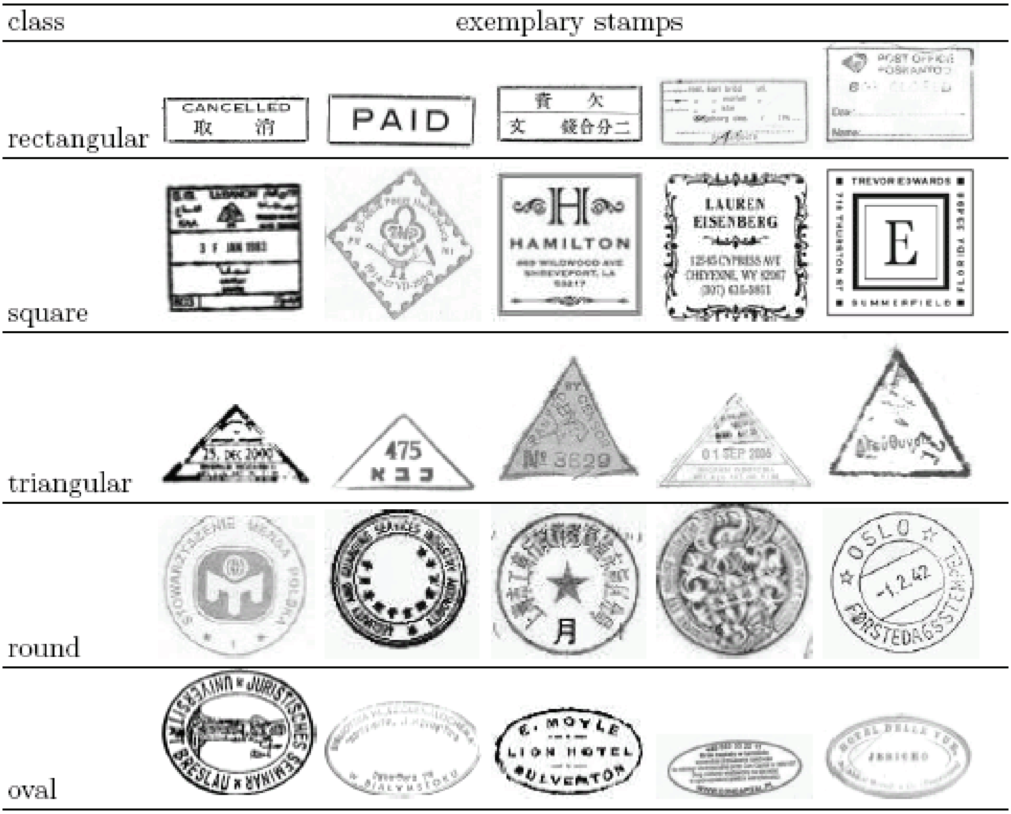


Fig. 5. Sample official stamps, divided into five classes ([9])

The application of the GSA idea in various problems is the second important issue. As it was mentioned, the General Shape Analysis was already successfully applied to the problem of identification of a type of a stamp in searching for probable false digital documents stored on a hard drive. Some shapes of stamps are sometimes typical for particular applications, e.g. official are usually round or elliptical, medical ones are rectangular, etc. Therefore, the coarse identification of a general shape of a stamp could be used. The results of the experiments performed on this problem were provided for example in [8] and [9]. In [8] five algorithms for shape representation were investigated and amongst them the best result was obtained by means of the Point Distance Histogram. In [9] additionally the Discrete Cosine Transform was used in order to reduce the features dimensionality. The above-mentioned problem was depicted in Fig. 5., where some stamps used during the experiments were divided into several main categories. In future, some other possible areas of applications of the General Shape Analysis will be experimentally investigated.

REFERENCES

[1] BALLARO B., ISGRO F., TEGOLO D., *Silhouette Encoding and Synthesis Using Elliptic Fourier Descriptors, and Applications to Videoconferencing*, Journal of Visual Languages & Computing, Vol. 15, Iss. 5, 2004, 391–408.

[2] BERTRAND O., QUEVAL R., MAITRE H., *Shape Interpolation Using Fourier Descriptors with Application to Animation Graphics*, Signal Processing, Vol. 4, Iss. 1, 1982, 53–58.

[3] CHEN C.-S., YEH C.-W., YIN P.-Y., *A Novel Fourier Descriptor Based Image Alignment Algorithm for Automatic Optical Inspection*, Journal of Visual Communication and Image Representation, Vol. 20, Iss. 3, 2009, 178–189.

[4] CHENG D., YAN H., *Recognition of Handwritten Digits Based on Contour Information*, Pattern Recognition, Vol. 31, Iss. 3, 1998, 235–255.

[5] DIAZ DE LEON, R., SUCAR, L.E., *Human Silhouette Recognition with Fourier Descriptors*, Proceedings of the 15th International Conference on Pattern Recognition, Vol. 3, 2000, 709–712.

[6] FREJLICHOWSKI, D., *General Shape Analysis Using Fourier Shape Descriptors*, In Swiątek J., Borzemski L., Grzech A., Wilimowska Z. (Eds.): Information Systems Architecture and Technology — System Analysis in Decision Aided Problems, 2009, 143–154.

[7] FREJLICHOWSKI D., *An Experimental Comparison of Seven Shape Descriptors in the General Shape Analysis Problem*, In: A. Campilho and M. Kamel (Eds.): ICIAR 2010, Part I, Lecture Notes in Computer Science, Vol. 6111, 2010, 294–305.

[8] FREJLICHOWSKI D., FORCZMAŃSKI P., *General Shape Analysis Applied to Stamps Retrieval from Scanned Documents*, In: D. Dicheva and D. Dochev (Eds.): AIMSA 2010, Lecture Notes in Artificial Intelligence, Vol. 6304, 2010, 251–260.

[9] FORCZMAŃSKI P., FREJLICHOWSKI D., *Efficient Stamps Classification by Means of Point Distance Histogram and Discrete Cosine Transform*, In: R. Burduk et al. (Eds.): Computer Recognition Systems 4, Advances in Intelligent and Soft Computing, Vol. 95, 2011, 327–336.

[10] GONZALEZ E., ANTONIO ADAN A., FELIU V., SANCHEZ L., *Active Object Recognition Based on Fourier Descriptors Clustering*, Pattern Recognition Letters, Vol. 29, Iss. 8, 2008, 1060–1071.

[11] HESSELMANN N. L., Structural Analysis of Lip-Contours for Isolated Spoken Vowels Using Fourier Descriptors, Speech Communication, Vol. 2, Iss. 4, 1983, 327–340.

[12] HUBER-MORK R., ZAHARIEVA M., CZEDIK-EYSENBERG H., *Numismatic Object Identification Using Fusion of Shape and Local Descriptors*, In: Bebis G. et al. (Eds.): ISVC 2008, Lecture Notes in Computer Science, 5359, 2008, 368–379.

[13] KEYS L., WINSTANLEY A., *Fourier Descriptors as A General Classification Tool for Topographic Shapes*, Proceedings of the Irish Machine Vision and Image Processing Conference, 1999, 193–203.

[14] KUTHIRUMMAL S., JAWAHAR C. V., NARAYANAN P. J., *Fourier Domain Representation of Planar Curves for Recognition in Multiple Views*, Pattern Recognition, Vol. 37, Iss. 4, 2004, 739–754.

[15] LAI M.T.Y., SUEN C.Y., *Automatic Recognition of Characters by Fourier Descriptors and Boundary Line Encodings*, Pattern Recognition, Vol. 14, Iss. 1–6, 1981, 383–393.

[16] LAM, K. P., *Contour Map Registration Using Fourier Descriptors of Gradient Codes*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 7, Iss. 3, 1985, 332–338.

[17] LEICESTER S.E., FINNEY J.L., BYWATER R.B., *Description of Molecular Surface Shape Using Fourier Descriptors*, Journal of Molecular Graphics, Vol. 6, Iss. 2, 1998, 104–108.

[18] MAHMOUD S.A., *Arabic Character Recognition Using Fourier Descriptors and Character Contour Encoding*, Pattern Recognition, Vol. 27, Iss. 6, 1994, 815–824.

[19] MAHOOR M.H., ABDEL-MOTTALEB M., *Classification and Numbering of Teeth in Dental Bitewing Images*, Pattern Recognition, Vol. 38, Iss. 4, 2005, 577–586.

[20] MEHTRE, B.M., KANKANHALLI, M.S., LEE, W.F., *Shape Measures for Content Based Image Retrieval: A Comparison*, Information Processing and Management, Vol. 33, Iss. 3, 1997, 319–337.

[21] MORNS I.P., DLAY S.S., *Character Recognition Using Fourier Descriptors and a New Form of Dynamic Semisupervised Neural Network*, Microelectronics Journal, Vol. 28, Iss. 1, 1997, 73–84.

[22] MOWBRAY S., NIXON M.S., *Automatic Gait Recognition via Fourier Descriptors of Deformable Objects*, In: Kittler J. and Nixon M.S. (eds.): AVBA 2003, Lecture Notes in Computer Science, Vol. 2688, 2003, 566–573.

[23] NIKRAVAN B., BAUL R.M., GILL K.F., *An Experimental Evaluation of Normalised Fourier Descriptors in the Identification of Simple Engineering Objects*, Computers in Industry, Vol. 13, Iss. 1, 1989, 37–47.

[24] OIRRAK A. EL, DAOUDI M., ABOUTAJDINE D., *Affine Invariant Descriptors Using Fourier Series*, Pattern Recognition Letters, Vol. 23, Iss. 10, 2002, 1109–1118.

[25] OSOWSKI S., NGHIA D.D., *Fourier and Wavelet Descriptors for Shape Recognition Using Neural Networks — a Comparative Study*, Pattern Recognition, Vol. 35, Iss. 9, 2002, 1949–1957.

[26] PARK K.S., LEE N.S., *A Three-Dimensional Fourier Descriptor for Human Body Representation/Reconstruction from Serial Cross Sections*, Computers and Biomedical Research, Vol. 20, Iss. 2, 1987, 125–140.

[27] PINKOWSKI B., *Multiscale Fourier Descriptors for Classifying Semivowels in Spectrograms*, Pattern Recognition, Vol. 26, Iss. 10, 1993, 1593–1602.

[28] PUCHALA D., YATSYMIRSKYY M., *Neural Network in Fast Adaptive Fourier Descriptor Based Leaves Classification*, In: Rutkowski L., Tadeusiewicz R., Zadeh L.A. and Zurada J.M. (Eds.): ICAISC 2008, Lecture Notes in Computer Science, Vol. 5097, 2008, 135–145.

[29] RAJ P.M., CANNON W.R., *2-D Particle Shape Averaging and Comparison Using Fourier Descriptors*, Powder Technology, Vol. 104, Iss. 2, 1999, 180–189.

[30] REEVES, A.P., PROKOP, R.J., ANDREWS, S.E., KUHL, F.P., *Three-Dimensional Shape Analysis Using Moments and Fourier Descriptors*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 10, Iss. 6, 1988, 937–943.

[31] SHRIDHAR M., BADRELDIN A., *High Accuracy Character Recognition Algorithm Using Fourier and Topological Descriptors*, Pattern Recognition, Vol. 17, Iss. 5, 1984, 515–524.

[32] SOMAIE A.A., IPSON S.S., *A Human Face Profile Identification System Using 1-D Real Fourier Descriptors*, International Journal of Infrared and Millimeter Waves, Vol. 16, No. 8, 1995, 1285–1298.

[33] SWEENEY, N., BECKER, R.L., SWEENEY, B., *A Comparison of Wavelet and Fourier Descriptors for a Neural Network Chromosome Classifier*, Proceedings of the 19th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Vol. 3, 1997, 1359–1362.

[34] WONG W.-T., SHIH F.Y., LIU J., *Shape-Based Image Retrieval Using Support Vector Machines, Fourier Descriptors and Self-Organizing Maps*, Information Sciences, Vol. 177, Iss. 8, 2007, 1878–1891.

[35] YADAV R.B., NISHCHAL N.K., GUPTA A.K., RASTOGI V.K., *Retrieval and Classification of Objects Using Generic Fourier, Legendre Moment, and Wavelet Zernike Moment Descriptors and Recognition Using Joint Transform Correlator*, Optics & Laser Technology, Vol. 40, Iss. 3, 2008, 517–527.

[36] ZHANG D., LU G., *A Comparative Study of Curvature Scale Space and Fourier Descriptors for Shape-Based Image Retrieval*, Journal of Visual Communication and Image Representation, Vol. 14, Iss. 1, 2003, 39–57.

image processing, dynamic image recognition,
driver's assistance system

Przemysław ZAJADLAK*, Mateusz LIS*, Jerzy ŚWIĄTEK*

DRIVER'S ASSISTANCE SYSTEM BASED ON LICENSE PLATE LOCALISATION AND DISTANCE ESTIMATION ALGORITHM

Fluency and safety of traffic flow is one of the most important contemporary problems as the number of cars is constantly increasing. In this work two main issues are considered: accelerating immediately after the previous car moved on when the traffic lights change and keeping a safe distance from the previous car in regular traffic e.g. on the motorway. The Driver's Assistance System was proposed to support drivers by displaying messages about braking and accelerating. The system was based on the video image from a camera attached to a car and the License Plate localization algorithm, also introduced in this work.

1. INTRODUCTION

As the number of cars is constantly increasing, the problem of safety and fluency of road traffic is becoming a serious issue. Especially in big cities where the number of traffic jams and road accidents is very large. This situation is caused by two inappropriate driver's behaviours. The first one is too late moving on after the other car, as the traffic lights change to green. This is the cause of a smaller amount of cars passing the crossroad on every light cycle, which is the one of the major reasons of traffic jams. The second problematical situation occurs when the driver does not keep a long enough distance from the car ahead or, for any reason, does not pay attention to the situation in front of the car. In this case, there is a very high possibility of a road accident.

Both of the problematic situations mentioned above can be solved using the automatic assistance system. There are some commercial systems using ultrasounds for

* Wrocław University of Technology, Wrocław.

object location like e.g. Front Assist system in VW cars, but this problem is also very interesting from the viewpoint of the dynamic images analysis.

In this work a **Driver's Assistance System** (**DAS**) is introduced. Another example of such a system can be found in [11]. Reference [11] describes a driver's assistance system that checks the direction in which drivers are looking, to verify if they are not too tired or if they saw a road sign. In [13] a system of autonomous driving based on visual information is introduced. It is also discussing the problem of dynamic image analysis.

DAS presented in this work is based on **License Plate** (**LP**) localisation and distance estimation algorithm. The problem of LP localisation was discussed many times in the literature and there are many approaches for this task. The first group of algorithms uses edge detection, and then applies a morphological operations (such as closure) to merge the edges corresponding to LP regions [14] [15]. The second approach is based on colours information [16] [17]. Searching for regions with colours corresponding to the colours of LP is however very sensitive on different illumination conditions. In [18] a SVM was used to train the classifiers recognising regions with that similar to LP texture. All those methods have one common disadvantage – they have high computational cost, so they cannot be used in real-time systems. They also do not use the dynamic characteristics of the image and they process every single frame from the video separately. That is the reason why an efficient, dynamic method of LP localisation is used in DAS.

DAS realises two main functions:

- On the traffic light crossroad, it is recommending the driver to either accelerate or brake when following the car ahead. The target is to improve the fluency of traffic.
- On regular traffic DAS checks whether the distance from the car ahead is not too short, and if so, it recommends pushing the brake pedal to the driver. The target in this case is to warn a driver about a dangerous situation on the road.

In the subsequent sections of this work, the architecture of the DAS system will be presented, as well as the outline of LP localisation and distance estimation algorithm. Evaluation of the system, its restrictions and future works will be also discussed.

2. OUTLINE OF THE CONTROL ALGORITHM

The aim of the control system was to support the driver in two situations. The first one was accelerating after the previous car on the traffic light and the second was warning the driver in case of too short distance from the car ahead. Both of them can be solved using the size change of the object in front of the controlled car. As the most

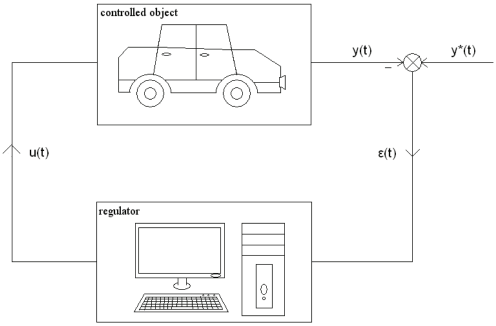


Fig. 1. Driver's Assistance System scheme

characteristic object, the License Plate was chosen. Knowing the size of the LP, distance change can be estimated easily.

Driver's Assistance System consist of two main parts: a video camera attached to a car and a computer system capable of estimating the LP's size and determining, using the video image from the camera, the decision whether to accelerate, brake or keep the current speed. In Fig. 1 a schemata of the basic control system is presented where:

- $y(t)$ – estimated distance from the car ahead.
- $y^*(t)$ – expected distance from the car ahead.
- $\varepsilon(t)$ – control error, equals to $y^*(t) - y(t)$.
- $u(t)$ – decision about speed change.

The set of control decisions was constrained to three possible options: "accelerate", "brake", and "do nothing". The DAS can work in two different modes: starting mode for acceleration on the traffic lights and warning mode for the distance from the car ahead controlling in the regular traffic. The control algorithm proceeds as follows:

1. If the system is in the starting mode, go to 2, otherwise go to 6.
2. If the driver does not push the accelerator or brake pedal, go to 3, otherwise go to 1.
3. If the current distance is less than $d_{s,min}$ or LP size compared to the mean value of the three previous measurements increased by $p_{s,max}$ then "brake".
4. If the current distance is more than $d_{s,max}$ or LP size compared to the mean value of the three previous measurements decreased by $p_{s,min}$ then "accelerate".
5. If the system is in the starting mode, go to 2.
6. If the current distance from the car ahead is less than $d_{w,min}$ then "brake".
7. If system is in the warning mode, go to 6, otherwise go to 1.

All the constant values were determined empirically during the experiments.

3. LICENSE PLATE LOCALISATION AND SIZE ESTIMATION

The most important part of the DAS is the proper estimation of the LP's size. The main idea of the algorithm is based on the fact that the LP is generally located between tail lights [10]. The LP's size estimation algorithm was divided into three main phases described as the following.

3.1. PHASE I: GENERAL REGION OF INTEREST LOCALISATION

For performance reasons **General Region of Interest** (**GROI**) is localised first. License Plate (LP) will be searched only in GROI. This phase consists of three steps.

1. Red colour extraction

For making this step easier a **CIE L*a*b*** color space instead of **RGB** was used [1]. To determine whether the pixel of the input image (A1 in Fig. 2) is red or not, the value of a* parameter is examined. If it is in scope $[a^*_{min}, a^*_{max}]$ the pixel is marked as red. The output image is similar to A2 in Fig. 2.

2. Tail lights localisation

Tail lights localisation is conducted, taking into account the fact that the LP is, in most cases, located between, slightly above or under car's lights. In the image containing only red pixels, all contours are found using the algorithm described in [2]. For every contour the bounding rectangle is marked and if its area is higher than A_{min} then it is added to the candidate's list. For every pair of bounding rectangles in the candidate's list, the difference of their areas is calculated and checked whether it does not exceed the A_{max} value. If not, the angle between the vertical line and line connecting the centres of bounding rectangles is determined and compared to α_{max}. If the angle is less than α_{max} value, the considered pair of bounding rectangles is added to potential tail lights list. From this list, the pair satisfying the following condition was chosen: the length of the line (denoted by l_t) connecting centres of the rectangles is over D_{min} and centre of this line, denoted by (x_c, y_c), is the closest one to the image centre. The chosen pair is marked as the tail lights of the car. Tail lights are marked as red rectangles in A3 in Fig. 2.

3. GROI determination

After determining the position of tail lights in the image, the GROI is then determined as the rectangle with coordinates of the left upper corner given by the following equation:

$$(x_g, y_g) = (x_c, y_c + 0.15 \bullet l_t), \tag{1}$$

width equal to l_t and height equal to $0.8 \bullet l_t$. The example of GROI is marked by the blue rectangle in A in Fig. 2 (red line connects centres of tail lights).

3.2. PHASE II: SPECIFIC REGION OF INTEREST LOCALISATION

Because of the fact that further operations have high computational cost it was imperative to determine a smaller than GROI region of interest. It is called **Specific Region of Interest** (**SROI**). This phase consists of three steps.

1. Edge extraction

For further analysis, it is necessary to extract all vertical edges that are very useful in the LP location finding process [3], [4], [5]. To make this task easier, a vertical Sobel operator is applied. This operator is defined by the following matrix:

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}. \tag{2}$$

After applying a convolution operator for GROI image and the matrix (2), the result image is similar to B2 in Fig. 2. As can be easily seen, all vertical edges have been enhanced. The next step is to find every vertical edge in the image. The **Canny detector**, originally introduced in [6], is used for this issue. As a result of this operation, a binary image (similar to B3 in Fig. 2) is produced, in which white pixels represent the pixels originally corresponding to edges in the processed image.

2. Edge filtration and Marker Line

Not every edge detected in the previous step is useful for License Plate localisation. In this stage, all the edges that are not possibly corresponding to the LP region should be removed as mentioned in [5] [7]. The filtration algorithm checks every edge, if it is satisfying the following condition: its orientation is at the angle of 90 degrees to the horizontal line and its length is greater than L_{min}. All the edges that are not satisfying this condition are removed and then the processed image looks similar to B4 in Fig. 2. As it can be noticed, most of the edges that remained after filtration are corresponding to the LP region (boundaries of the LP or the letters and digits included on the LP).

To make the SROI determination very precise, the **Marker Line** (**ML**) is calculated as the next one. The ML is the horizontal line, for which the sum of distances from the centres of each remained vertical edge is the minimal. Let the ML be denoted by

$$m : y = M, \tag{3}$$

and the set of all points $c = (x_c, y_c)$, that are centres of vertical edges, is denoted by C. Then the M from the equation (3) can be calculated as the minimizing argument of the following equation:

$$M = \arg\min_{M} \left\{ \sum_{c \in C} \left| M - y_c \right| \right\}. \tag{4}$$

As it was mentioned before, most of the vertical edges correspond to the LP region, so there is a high probability that the ML will be somewhere near the centre of the LP region. The ML is marked as the red line in B5 in Fig. 2.

3. SROI

As the European Union LPs are considered, the aspect ratio [5] [8] of these plates can be used for SROI determination. The aspect ratio of the width and height of typical the LP is 5:1. Taking this fact into account, the SROI is then marked as the rectangle with the centre on the Marker Line, width equal to GROI width, and height equal to $\frac{1}{5}$ of GROI width. In other words, the SROI size is equal to the maximum possible size of the LP located in GROI. In the B image in Fig. 2. SROI is marked as a green rectangle, while GROI is marked as a blue rectangle and the red line is the ML.

3.3. PHASE III: LICENSE PLATE LOCATION AND DISTANCE ESTIMATION

After determining an accurate final region of interest, the most precise calculations are made to localise the LP and estimate its size. This phase consists of four steps.

1. Histogram equalization

Before the first step of this phase, the SROI image (C1 in Fig. 2) is first transformed to gray scale (C2 in Fig. 2) as information about colour is not necessary in further processing. The next operation is **histogram equalization**. It increases the global contrast of an image, by spreading out the most frequently pixel intensity values [9]. The result of it is illustrated in C3 in Fig. 2. As can be seen, the contrast between the LP background and the rest of the image is significantly enhanced. This fact will have important meaning in the next processing stages.

2. Gray levels reduction

As it is important to merge the regions with similar pixel intensity, especially the LP background, into consistent ones, the gray level reducing operation was performed. The image was converted to 9 gray levels by taking each pixel and substituting its original value of intensity by the closest allowed one. The allowed values was 0, 31, 63, 95, 127, 159, 191, 223 and 255. The exemplary result of this operation is illustrated in C4 in Fig. 2.

3. SROI binarisation

After reducing the gray levels of the SROI image, the next step is to make a binary image based on SROI. This is simply realised by comparing the intensity of each pixel from the gray level reduced image to a threshold I_{min}. If the intensity has a greater value than I_{min} it is marked as the white pixel on the binary image, and otherwise it is marked as the black one. As a result, an image similar to C5 in Fig. 2 is created. In this picture, most of the white pixels correspond to the LP background, which is very well distinguished from other eventual objects, such as the car's model letters or car's body (especially if the car is e.g. white) presented in the image.

4. LP localisation and distance estimation

For precise localisation of an LP plate and making the decision whether it is located in SROI or not, a few more parameters were introduced. These are as following:

1) *MBL* – a maximum amount of neighbouring vertical black lines in the image.
2) *WPR* – a white pixel to black pixel ratio in the image (similar to introduced in [9]).
3) *SLR* – symmetrical lines to all horizontal lines ratio in the image. The line is symmetrical when the amount of white pixels on its left side differs from the amount of white pixels on its right side by the maximum value of S_{max}.

In the binary image created in the previous step, all the rectangles are detected. If the aspect ratio of the found rectangle is between AR_{min} and AR_{max} it is added to the candidate's list. For every rectangle from the candidate's list, the value of MBL, WPR and SLR parameters are calculated. If the conditions $MBL_{min} < MBL < MBL_{max}$, $WPR_{min} < WPR < WPR_{max}$ and $SLR_{min} < SLR < SLR_{max}$ are satisfied, then the rectangle is added to the possible LPs list. The examples of the possible LPs are marked as blue rectangles in C6 in Fig. 2.

The next step is to decide which of the possible LPs is the real one. For this purpose, some dynamical characteristic of the considered video image is checked. For every rectangle from the possible LPs list, a value from set {*yes, no*} is assigned. If the size of the rectangle does not differ much from the mean value of the LP size on three previous video frames and the position of the rectangle does not differ much from the mean value of the LP position on three previous video frames, the value *yes* is assigned for such a rectangle. If there is more than one rectangle with *yes* value assigned, then the one with the closest to the mean of the LP on the last three video frames size is chosen as the final LP. The example of result image is presented in C7 in Fig. 2 where the orange rectangular is the LP.

The last stage in the algorithm is the estimation of the distance from the car ahead. For this purpose, 22 pictures of a license plate from different distances were taken. The dependence of the distance on the LP size was then estimated by an equation:

$$d = ax^b \,, \tag{5}$$

where *d* denotes the estimated distance between the camera and the LP and *x* denotes the size of the LP in the picture. In this case $a = 360.9$, and $b = –0.48$. The final result image is presented in C in Fig. 2, where the orange rectangular is the LP, the green rectangular is SROI, the numbers in the upper left corner are the coordinates of the centre of the LP and the number in upper right corner is the estimated distance from the car ahead.

3.4. DYNAMICS OF THE IMAGE

It is worth mentioning that the image processed by DAS is not a static image, but is a dynamic image consisting of separate frames captured by the camera. All the frames are obviously connected to each other, as the next frame is a logic consequent of the previous one. These dynamic characteristics have significant meaning for the performance of the system.

When the DAS is detecting a license plate, it is not necessary to perform all the phases of the algorithm for every video frame. In the case that in the previous frame the LP was detected, the processing of the next frame will be started from the third phase. A new SROI will be defined in the same place where the previous one was, taking into account its inertness (its position will be corrected by a difference between two following detected LP coordinates). Such an approach makes the whole process less computational time consuming. But in every 2 second image, processing is started from the first phase, because of some issue connected to inertness. If the car ahead will be slowly moving from one lane to another then the SROI region will be following its LP and a controlled car may drive into another car ahead on the same lane, because the system is following an incorrect LP.

The other useful thing about image dynamics is that possible LP candidates can be verified taking into account a detection history. It is obvious that the LP cannot suddenly change its size or position very much, and that fact is taking into account in the process of verifying whether the candidate region corresponds to the LP or not. The size and position of the last three detected LPs are stored and used for verification. If the DAS does not detect anything for 2.5 seconds, then the history is cleared to avoid the situation of continuing control process using information from the obsolete situation in front of the car.

4. EVALUATION

In this section we present an experimental evaluation of the control system presented previously. The system was evaluated using a video camera (Canon digital IXUS 90 IS, resolution 640x480) attached to front window of the car (video was recorded from inside the car). The created films were analysed afterwards using implemented software. DAS was responsible for displaying the decision message on the screen. System examination was divided into two general parts concerning each of the purposes described in the previous sections. For each goal, we have recorded 13 films including various weather conditions (sunny 45%, cloudy 40%, very cloudy 15%), time of the day (night 15%, day 85%), car model (hatchback, sedan, station wagon etc.), tail lights type (since it was used in the algorithm, it was an important factor) and its colour. All films were recorded in roads of voivodeship Dolnośląskie, Poland. Several measures were introduced to present the objective outcome of tests:

A) Basic effectiveness measures

C_A – ratio of the number of correct accelerations following the car ahead to the number of all test cases.

C_B – ratio of the number of properly displayed warning messages (when car was too close to the car ahead) to all the situations in which such a message ought to be displayed.

B) Additional effectiveness measures

B_T – number of LP not found messages, when the LP is present on the screen.

W_R – ratio of the number of frames on which the LP was found, to all the examined frames of the film.

F_R – ratio of the number of frames on which the LP was incorrectly localised to all frames, where the LP was found.

FPS – mean performance of the system expressed in frames per second.

Basic effectiveness measures were estimated using visual analysis. Additional measures were calculated precisely using the implemented software (except for the F_R which required the visual analysis of about 1200 frames).

The obtained results are presented in compiled form in tab. II. The introduced criterion of effectiveness enabled us to identify the weaknesses of the presented system, which point out future improvements. In the following subsections, groups of criterions and their values will be discussed.

1) *C_A and C_B.*

These two measures were intended to give the overall evaluation of the system. The obtained values of around 90% reject the system as an automatic driver system, but apparently confirm its duty as driver's assistance. More importantly, we managed to identify possible dangerous situations, when the system is not capable of working properly, e. g. in warning mode, when the car ahead is red, the system cannot find GROI because it is difficult to distinguish between the car and its tail lights.

2) *B_R, W_R and F_R.*

Measures related to finding the LP on the screen revealed the way the system works: when it catches a plate, it usually is able to hold it for a long time (this is because of the dynamic image analysis between separate frames). However there are situations in which the system has problems which are not solved yet. Red coloured cars, as mentioned above. Another important problem is a clear sunny day, when sunlight puts shadows on the LP and the system does not work properly. Furthermore GROI localization problems may occur when the car has unusual tail lights (e. g. LED). An additional issue concerning GROI takes place at night time, as night lanterns can create noise which prevents the system from car tail lights localisation. It is important to mention that all the issues concerning GROI localisation come from one step of the algorithm, which is tail lights localization, and is to be improved in the future work.

Discussing the value of W_R it is important to mention that even the lowest value of around 20% was (in most cases) enough to perform proper control system as we see in other criterions.

3) *FPS*

Frames per second show the system's performance in all conditions. The lowest result, around 10 FPS, means that the system would react in about 0.1 second. It is still far more immediate than the fastest human drivers can perform [12].

5. CONCLUSIONS AND FURTHER WORK

In this work a proposed system for driver's assistance in avoiding sudden braking and facilitate accelerating after the traffic lights change is presented. It utilizes an in-vehicle video camera based on which road situation analysis is performed and computer control system is working. Situation analysis is based on the localization and size estimation of the license plate of the car ahead. This information is used to perform a decision support system work. This may enable the computer to warn the driver about the danger or even perform the action itself (as evaluation proved, the system reacts ten times faster than the average driver).

An evaluation of the system covered most of the possible situations on the road (different day times, different weather conditions, different car models etc.). The system empirically proved to be useful as driver assistant and further improvements may even enable it to control the car itself, as all possible flaws were pointed out during the precise evaluation. Further research in this area will consider additional improvements of General Region Of Interest localisation as it has caused most of the errors made by the Driver's Assistance System. Tail lights detection could be supplemented by another method, e. g. car localisation.

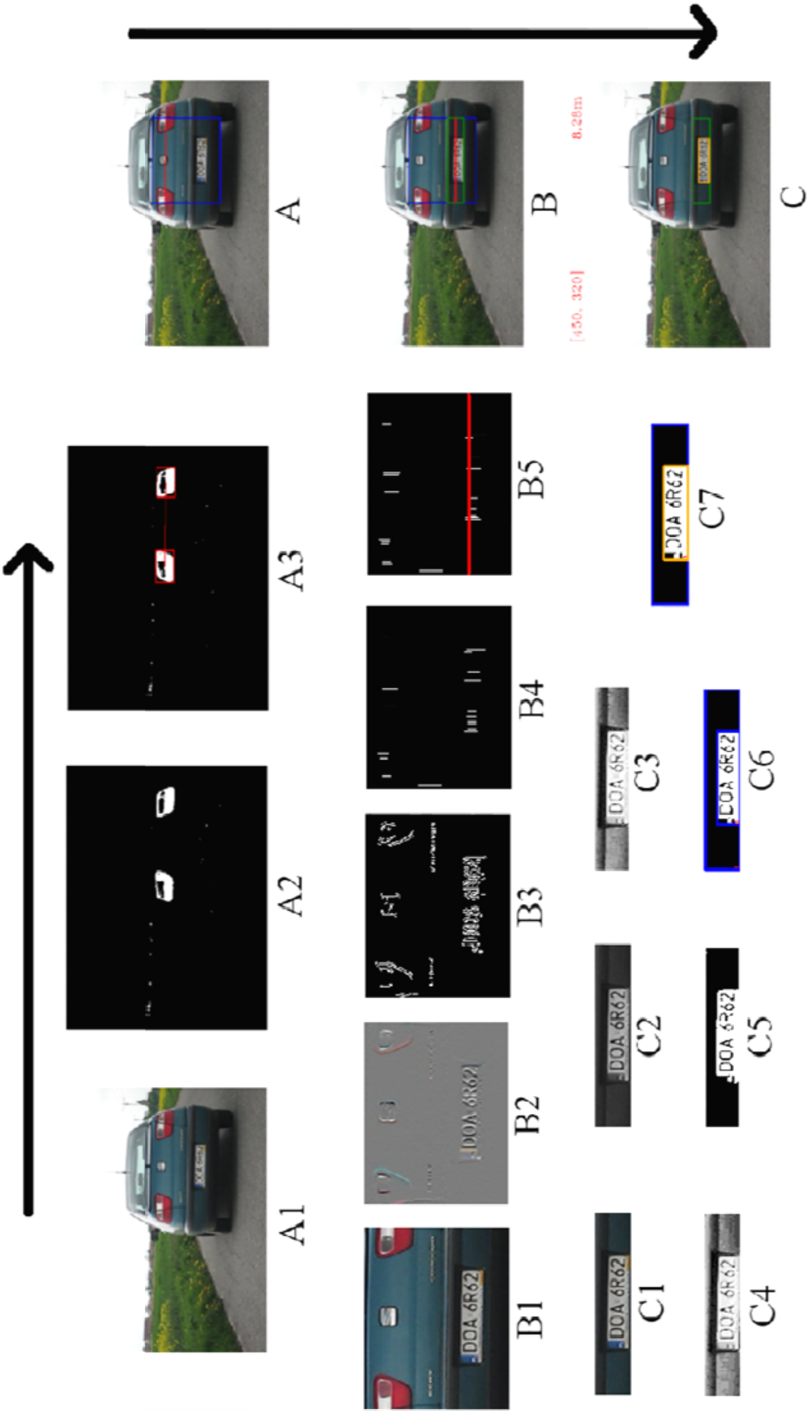


Fig. 2. License Plate size estimation algorithm flow. Different rows refer to main phases of image processing. Overview pictures for each phase are presented on the right side

REFERENCES

[1] YOUSSEF S. M., ABDELRAHMAN S. B., *A smart access control using an efficient license plate location and recognition approach*, Expert Systems with Applications, 2008, 34(1):256–265.

[2] SUZUKI S., ABE K., *Topological structural analysis of digitized binary images by border following*, Computer Vision, Graphics, and Image Processing, 1985, 30(1):32–46.

[3] JIAO J., YE Q., HUANG Q., *A configurable method for multi-style license plate recognition*, Pattern Recognition, 2009, 42(3):358–369.

[4] RATTANATHAMMAWAT P., CHALIDABHONGSE T. H., *A car plate detector using edge information*, Communications and Information Technologies, 2006. ISCIT '06. International Symposium on, 2006 11, pp. 1039–1043.

[5] ZHENG D., ZHAO Y., WANG J., *An efficient method of license plate location*, Pattern Recognition Letters, 2005, 26(15):2431–2438.

[6] CANNY J., *A computational approach to edge detection*, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 10 1986, PAMI-8(6):679–698.

[7] ABOLGHASEMI V., AHMADYFARD A., *Improved image enhancement method for license plate detection*, Digital Signal Processing, 2007, 15th International Conference on, 1–4 2007, pp. 435–438.

[8] JIA W., ZHANG H., HE X., *Region-based license plate detection*, Journal of Network and Computer Applications, 2007, 30(4):1324–1333.

[9] HSIAO-YUN TSENG, CHING-HAO LAI S.-S. Y., *An effective license-plate detection method for overexposure and complex vehicle images*, Convergence and Hybrid Information Technology, ICHIT '08. International Conference on, 2008, pp. 176–181.

[10] HSIEH C.-T., CHANG L.-C., HUNG K.-M., HUANG H.C., *A real-time mobile vehicle license plate detection and recognition for vehicle monitoring and management*, Pervasive Computing (JCPC), 2009 Joint Conferences on, 3–5 2009, pp. 197–202.

[11] ROMBAUT M., *ProLab2: A driving assistance system*, Math. Comput. Modelling, 1995, 22(4–7):103–118.

[12] JOHANSSON G., RUMAR K., *Drivers Brake Reaction Times*, Human Factors: The Journal of the Human Factors and Ergonomics Society, 1971, Vol. 13(5):23–27.

[13] KUNDUR S. R., RAVIV D., *Active vision-based control schemes for autonomous navigation tasks*, Pattern Recognition, 2000, 33(2):295–308.

[14] HONGLIANG B., CHANGPING L., *A hybrid license plate extraction method based on edge statistics and morphology*, Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, 2004, 2:831–834.

[15] ZHENG D., ZHAO Y., WANG J., *An efficient method of license plate location*, Pattern Recognition Letters, 2005, 26(15):2431–2438.

[16] WANG T.-H., NI F.-C., LI K.-T., CHEN Y.-P., *Robust license plate recognition based on dynamic projection warping*, Networking, Sensing and Control, 2004 IEEE International Conference on, 2004, 2:784–788.

[17] SHI X., ZHAO W., SHEN Y., *Automatic license plate recognition system based on color image processing,* Lecture Notes in Computer Science, 2005, 3483(IV):1159–1168.

[18] KIM K. I., JUNG K., KIM J. H. *Color texture-based object detection: An application to license plate localization*. Lee, Seong-Whan (ed.) et al., Pattern recognition with support vector machines. 1st international workshop, SVM 2002, Niagara Falls, Canada, August 10, 2002. Proceedings. Berlin: Springer. Lect. Notes Comput. Sci. 2388, 293–309 (2002).

tracking, dynamic object recognition,
image processing, dynamic images

Bartosz PATER*, Jarosław DRAPAŁA*

DYNAMIC PATTERN RECOGNITION ALGORITHM FOR PEDESTRIANS TRACKING

In order to recognize a moving object by video camera, two general approaches may be applied: static or dynamic pattern recognition. The former one is about processing each frame independently and to recognize an object of interest using features available on a single picture. The more advanced approach is to make use of dynamic patterns, that arise along the sequence of frames. In this work we developed dynamic recognition algorithm and proposed original method of feature extraction in the pedestrian detection task. The main idea is to follow the observation, that vertical position of people's head oscillates during gait. Distinctive rhythm of human gait is the feature extracted from video sequence and used to distinguish between pedestrians and other objects. Algorithm and implementation details are given. Analysis of incorrect classifications is provided. We suggest to apply the system to measure traffic at road junctions with traffic lights.

1. INTRODUCTION

Nowadays, when digital recording methods allows to processing images more easily and effectively, systems based on this kind of information are very common in our everyday life, especially when human safety is under consideration. Popularity of vision-based techniques is associated with the simplicity of interpretation of the effects of these algorithms by human expertise.

Pedestrian crossings with traffic lights are crucial parts of communication systems in the city. In order to increase safety of pedestrians in heavily crowded streets, we propose a system that is able to recognize pedestrians with use of low quality video camera. The system may help to reduce the danger on the road. This solution allows to improve road safety but it requires some additional equipment and procedures. The

* Institute of Informatics, Wrocław University of Technology, Poland.

lengths of the green light is different for road users, it depends on the hour and changes during peak hours matching to the most congested. These times are calculated on the basis of forecasts traffic during construction, often remain immutable for several years. Length of such cycles may vary over years. Proposed algorithm for detection and recognition can be used to measure the traffic of vehicles and pedestrians moving on the road. Data can be analysed in a real-time.

The video stream is a sequence of images generated over specified interval of time. Pattern recognition task performed on a video stream can be reduced to multiple independent recognition problems, each processing a single image to find a pattern in the images. Such an approach requires plenty of computational operations, whereas real-time applications, typically functioning with video camera, need rather simple algorithms. Therefore, it is reasonable to take dynamic pattern recognition into account. Instead of analysing patterns within a single image, dynamic recognition is about detection of patterns arising along a sequence of image. A pattern is no longer assigned to an image, but for video sequence. Extraction of patterns involves processing differences between consecutive images. Moreover, previous recognition may be taken into account, which allows too keep computational burden at reasonable level. It is especially useful, when tracking a moving object. Furthermore, for pedestrian tracking details of appearance are negligible, because they do not provide any important information for the task. Such king of information is mainly used for elimination if the impossible situation at the beginning of image analysis. The crucial part of dynamic pattern recognition solving is to find a class of dynamic pattern, that will be detected during video sequence processing. It must be emphasized, that typical patterns exhibited in single images, may also be considered, but in this work we focus only on dynamic one.

In general, moving objects detection procedures are based on binary movement map that serves to detect any difference in images. But only need to distinguish moving pedestrian from the background. Another moving objects, such as cars, bikes, trains, are not the focus of the algorithm. The key idea behind dynamic recognition algorithm presented in the work is to analyse the frequency of changes between two features: distance from upper edge of the object to the bottom end of the image and the width of the object. Rhythm of walking should enable to distinguish pedestrians from the background and another moving objects.

2. DETECTION OF MOVING OBJECT

Little differences between pixels positions in consecutive images makes difficulties in their distinction and often lead to classifying two or several object as one. This can disrupt detection of features and whole recognition process. To increase differences between pixels and to improve contrast, we propose to start from histogram stretching procedure:

$$J_{x,y} = \frac{J_{max} - J_{min}}{L_{max} - L_{min}} \times \left(L_{x,y} - L_{min}\right), \tag{1}$$

where: $J_{x,y}$ – brightness of the new pixel x, y, J_{max} – maximum brightness, J_{min} - minimum brightness, $L_{x,y}$ – present value of the pixel brightness, L_{max} – largest of the currently used brightness and L_{min} – the smallest brightness.

Binary motion map is defined as a logical picture of the same dimensions as source image, created from differences between corresponding pixels of successive frames in the sequence. A point takes a positive value when the change is greater than the threshold, otherwise it is set to zero. Classical binary motion maps are generated using the following formula:

$$b^{(n)}{}_{x,y} = \begin{cases} 1, & gdy \left|p_{x,y}^{(n)} - p_{x,y}^{(n-1)}\right| > T \\ 0, & gdy \left|p_{x,y}^{(n)} - p_{x,y}^{(n-1)}\right| \le T \end{cases}, \tag{2}$$

where: n – index of current frame, $x,\ y$ – coordinates of the pixel in the image, b – binary motion map element, p – pixel of input image, T – threshold.

After filtering out noise like “salt”, by using median filter, only objects that change their position should remain on binary motion map, see Fig. 1.

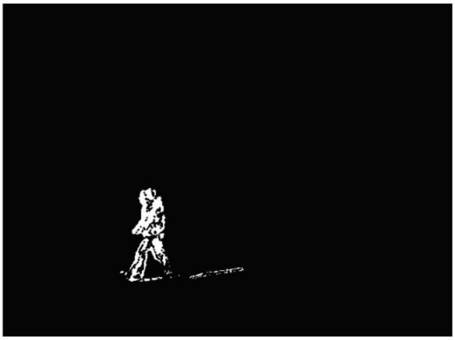


Fig. 1. Example of binary motion map

In single image the positive detection object determined by properties such as size and location is done by matching the rectangle with the following parameters: width, height and attachment points. The rectangle is described by the expression:

$$d = (y_d, x_d, h_y, w_x), \tag{3}$$

where: y_d, x_d – coordinates of point attachment, h_y, w_x – height and width.

Parameters detection process is divided into stages in which two-dimensional image on axes is mapped in the coordinate system of binary motion map. The sum of ones in axis responsible for the object's height is equivalent to the number of pixels in which sufficient change of input value in video sequence is detected.
Projection of binary motion map on horizontal axis is the vector:

$$\overline{w}^{(n)} = \left[u_{1,h}^{(n)}\ u_{2,h}^{(n)} \dots\ u_{k,h}^{(n)} \dots u_{W,h}^{(n)}\right]^T, \tag{4}$$

where: $(n = 1,2, \dots, N)$ – the next images in the sequence, $\overline{w}$ – vector containing vertical sums of binary motion map, $u_{k,h}^{(n)}$ – sum of all pixels with coordinate $x = k$ $(k = 1,2, \dots, W)$. The last sum is defined as follows:

$$u_{k,h}^{(n)} = \sum_{l=1}^{H} b^{(n)}{}_{k,l}, \tag{5}$$

where β_w – the number of pixels from which average sum is calculated, γ_w – the threshold separating the noise from important values , $\overline{\delta}$ – average value of the sum of the interval:

$$\overline{\delta}_{u_{k,h}^{(n)}} = \frac{1}{\beta_w} \sum_{l=k}^{k+\beta_w} u_{l,h}^{(n)}, \ \mu_{k,p}^{W} = \sum_{l=k}^{p} u_{l,h}^{(n)} \Bigg/ \sum_{j=1}^{W} u_{j,h}^{(n)}, \tag{6}$$

where μ_w – the threshold of minimum share of pixels belonging to object, $\mu_{k,p}^{W}$ – a direction of calculating the share of points belonging to an object from the whole vector.

This part of the algorithm is summarized below.

```
START:    $k = 0$, $x_b = 0$, set values: $\mu_w$, $\beta_w$ i $\gamma_w$
1. Calculate the current value of $\overline{\delta}_{u_{k,h}^{(n)}}$
2. If $x_b = 0$ and $\overline{\delta}_{u_{k,h}^{(n)}} \geq \gamma_w$ then $x_d = k$, $x_b = 1$ and go to 4
3. If $x_b = 1$ i $\overline{\delta}_{u_{k,h}^{(n)}} < \gamma_w$ and $\mu_{x_d,k}^{W} \geq \mu_w$ then $w_x = k - x_d$, STOP
4. If $k < W - \beta_w$ then $k = k + 1$ and go to 1
STOP.
```

The second step performs detection of an object in vertical axis. Values calculated in first step are used to changed length of the summed vectors from full width to calculated segment. Reducing the summation vector improves performance considerably. This step is similar to the previous one, so it will not be described here in details.

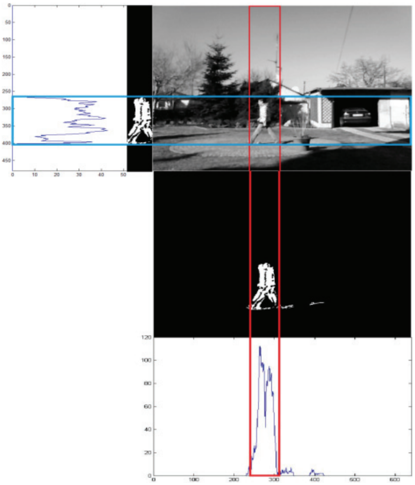


Fig. 2. Typical example of moving object detection

The whole process, from binary motion map to rectangle representing detected object, is illustrated by Fig. 2. Data set containing a sequence of frames similar to (2) is a base for recognition, the next step of proposed algorithm.

3. DYNAMIC RECOGNITION OF PEDESTRIAN

Until this point any try to determinate a class of object was performed. We have analysed video sequence only locally, processing single images individually. Values d_n calculated during detection process for all images containing an object of interest are components of the following matrix:

$$V = (d_n, d_{n-1}, d_{n-2}, \dots, d_{n-l}\ ), \tag{7}$$

where: n – the number of the current frame on which the object is recognized, l – the number of frames from the object detection to the current time instant. Pattern recognition algorithm has the form:

$$i = \psi(V), \tag{8}$$

where: i – number of recognized class matching with:

$$i \in \begin{cases} 1, & for\ positive\ pedestrian\ recognition \\ 0, & in\ others\ case \end{cases}. \tag{9}$$

In this work we narrowed the area of typical objects: walking people and moving vehicles. The challenge was to develop such a set of features, which allows to separate between pedestrian and another object in a real-time. We made interesting observation, that the highest point of moving pedestrian changes vertical position according to the rhythm of his steps. The same point for vehicle looks almost identical in all video difference images. The set of features based on the rhythm of a moving object is the basis for determining the class of detected object. This pattern cannot be observed in a single image, but only along the sequence of images, thus only dynamic approach is applicable here.

Execution of one step for the free walking pedestrian, according to studies included in [6], takes about *0.5-0.69*s, which directly translates into a frequency rhythm of the human gait oscillating about 2 Hz. Knowing the average frequency of human steps is considered here. Fig. 3 shows several segments of image sequence, when registering a moving pedestrian. Analysis of human gait rhythm reveals interesting dependence between images.

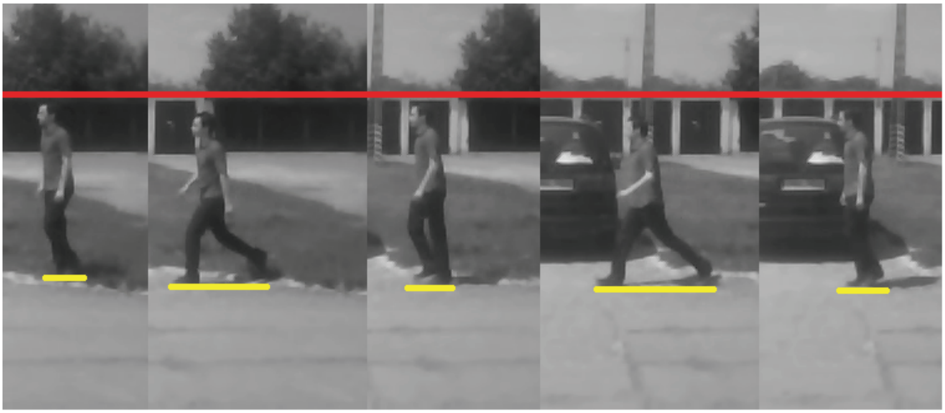


Fig. 3. Dynamic feature expressed by walking pedestrian

Profile of standing man is characterized by a certain height, each step leads to height reduction in the time of pulling one leg forward and back to the original amount at next junction of legs. This process is shown in figure 3 where pedestrian oscillates according to constant upper line.

The second relationship associated with each step is the width of the profiles, which in Fig. 3 is marked with brighter lines.

With series of rectangles that describe the height and width of the object profiles at each frame in the source sequence, and keeping in mind that they change according to the frequency equal to 2 Hz, the recognition algorithm is proposed.

The first method of determining class of an object is based on the frequency of the top of the frame variations. Differences between consecutive points position with a rectangle are described as follows:

$$\Delta y_{d,n-k} = y_{d,n-k} - y_{d,(n-k)-1}, \tag{10}$$

where: $y_{d,n-k}$ – starting point of the $(n-k)$-th rectangle, n – index of current frame, $k = 2,3,\dots,l$ – index of current element. Typical sample of difference vector diagram in time domain is depicted in Fig. 4.

Timing chart showing differences is very unclear, it is affected by the disruption and uncertainty of the measurement. The transition in the frequency domain increases transparency and facilitates their interpretation. Vector of differences transform according to Discrete Fourier Transform is described as follows:

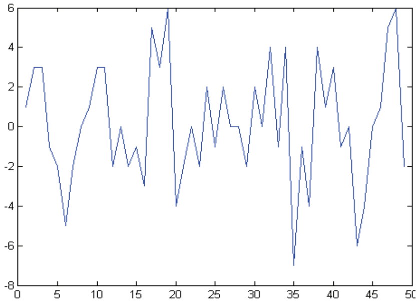


Fig. 4. Oscillations in time domain

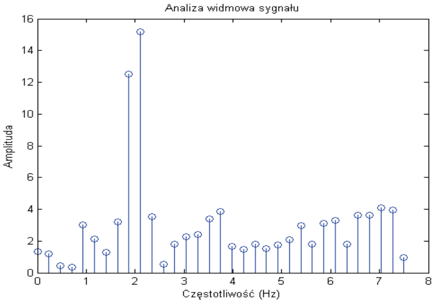


Fig. 5. Oscillations in frequency domain

$$f_{\Delta y_{d,n-k}} = \sum_{j=1}^{l} y_{d,n-j}\omega_N^{((n-j)-1)(k-1)}, \tag{11}$$

where: $\omega_n = e^{(-2\pi i)/N}$.

Transforming the vector of differences we get similar data as those in Fig. 5. Amplitudes for frequencies around a certain point are much higher than value for the remaining ones. This situation occurs when the main component of the signal is characterized by such frequency, but deformations introduced by noise are difficult to identify in the time domain. The point with maximum amplitude is a main rate at which parameter y_d of rectangle surrounding object was changed. To compare the frequency spectrum of a test object and typical pedestrian, we check whether maximum amplitude lies close to typical values to the rhythm of walking man, which is about 2 Hz. This condition can be written as:

$$f_{avg} - \varepsilon < \max\,(f_{\Delta y_d}) < f_{avg} + \varepsilon, \tag{12}$$

where: f_{avg} – average frequency of the rhythm of pedestrian walking, ε – allowable error.

The idea behind the second approach is very similar to the previous one. Both may be used in one procedure, making it more robust. In certain situations, methods can complement each other, measuring rhythm of human gait from different points of view. Now, the width of object profiles is defined by the following equation:

$$\Delta w_{x,n-\,k} = w_{x,n-k} - w_{x,(n-k)-1}, \tag{13}$$

where: $w_{x,n-k}$ – k rectangle width, $k = 2,3,\dots,l$ – index of current element.
The remaining steps are identical to the first method , so there will be ommited.

4. EXPERIMENTAL STUDY

Experimental study aims at determining the effectiveness of detection and object recognition. Reliable criteria and indicators were used to verify the usefulness and range of actual uses of the system under realistic conditions. Studies were conducted on set of previously recorded videos. The collection contained 20 video sequences where subject of recognition are pedestrians with different appearances in terms of dress, manner of walking. Moreover, cars of several brands and different colours were present in video sequences. Each of the objects moved with random speed. The studies also included different distances of an object from the camera. In order to determine the correctness of the algorithm and the system performance based on the proposed solution, more studies need to be performed. The algorithm was implemented in Matlab version 7.10.0.499. Monochromatic videos were recorded by a webcam with autofocus option turned on. The same simulation tests were conducted on a computer with Intel Celeron M 420 1.60 GHz processor and 2.5 GB of memory.

The average efficiency detection for all object from video sequences were 89%. Taking simplicity of the algorithm into account, it seems to be a satisfactory result. Recognition efficiency does not differ significantly from detection value. The average recognition quality is equal to 85,16%. The combination of detection of gait rhythm based on the differences between height of the top point of object and differences in the width of the rectangle occurred a very good solution. The quality of the detection algorithm does not depend on shape and size of object, which is the undoubted advantage of the approach, allowing for efficient detection of different vehicles and pedestrians. The average number of processed frames per second of algorithm was 18.85. Parameter value remains on constant high level for each video from the test set. This feature is a result of comparable number of calculations in each loop of algorithm, independently on situation, which indicates robustness.

5. CONCLUSIONS

When detection relies on difference in color intensity, some detection errors may take place. Typical situations are described below.

The object has colour similar to the background color – if the object does not sufficiently differ from the background, it comes to fault detection (left image on Fig. 5). Shadow may be detected as part of an object (middle image on Fig. 5). If an object moves to slow from frame to frame it may not be detected at all. It may also happen if an objects moves quite fast, but in the direction parallel to the camera (right image on Fig. 5).

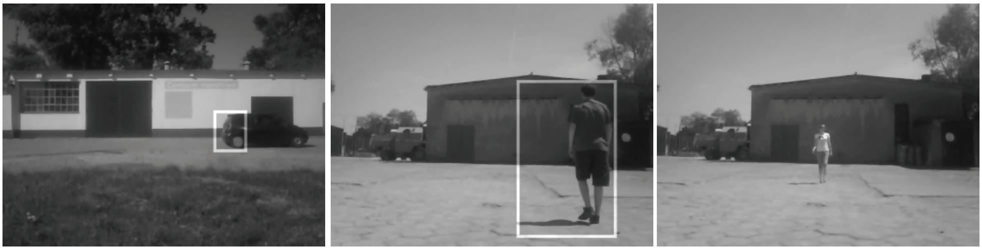


Fig. 5. Frequent mistakes made by the algorithm

Despite those problems, which may be easily handled by more detailed image processing, dynamic pattern recognition is powerful approach in real-time applications. It leads to fast patter recognition algorithms and utilizes interesting patterns, that may not be treated by standard pattern recognition algorithms processing single images.

REFERENCES

[1] ARNELL F., *Vision-based Pedestrian Detection System for use in Smart Cars*. Department of Numerical Analysis KTH and Computer Science Royal Institute of Technology, Sweden, 2005.

[2] BERTOZZI M., BROGGI A., FASCIOLI A., GRAF T., MEINECKE M., *Pedestrian Detection for Driver Assistance Using Multiresolution Infrared Vision*. IEEE Transactions on Vehicular Technology, 6, 53 (Listopad 2004).

[3] CHIA-JUNG P., HSIAO-RONG T., YU-MING L., HONG-YUAN M., SEI-WANG C., *Pedestrian detection and tracking at crossroads*. Pattern Recognition, 37 (2004), 1025–1034.

[4] CURIO C., EDELBRUNNER J., KALINKE T., TZOMAKAS C., SEELEN W., *Walking Pedestrian Recognition*. Institut fur Neuroinformatik, Lehrstuhl fur Theoretische Biologie, Bochum.

[5] DAI C., ZHENG Y., LI X., *Pedestrian detection and tracking in infrared imagery using shape.* Computer Vision and Image Understanding, 106 (2007), 288-299.

[6] EKIMOV A., SABATIER J., *Rhythm analysis of orthogonal signals from human walking.* Journal of the Acoustical Society of America, 3, 129 (Marzec 2011), 1306–1314.

[7] GANDHI T., TRIVEDI M.. *Pedestrian Protection Systems: Issues, Survey, and Challenges.* IEEE Transactions on Vehicular Technology, 8, 3 (Wrzesień 2007).

[8] GAVRILA D., *Pedestrian Detection from a Moving Vehicle.* Image Understanding Systems, DaimlerChrysler Research, Ulm.

[9] GOUBET E., KATZ J., PORIKLI F., *Pedestrian Tracking Using Thermal Infrared Imaging.* Mitsubishi Electric Research Laboratories, Cambridge.

[10] KASPRZAK W. *Rozpoznawanie obrazów i sygnałów mowy.* Politechnika Warszawska, 2009.

[11] MASOUD O., Papanikolopoulos N., *A Novel Method for Tracking and Counting*. IEEE Transactions on Vehicular Technology, 50, 5 (Wrzesień 2005).

[12] NANDA H., DAVIS L., *Probabilistic template based pedestrian detection in infrared videos.* Department of Computer Science University of Maryland, College Park.

[13] NIXON M., AGUADO A., *Feature Extraction Image Processing.* 2008.

[14] PAPAGEORGIOU C., EVGENIOU T., POGGIO T., *A Trainable Pedestrian Detection System.*

Center for Biological and Computational Learning and Artificial Intelligence Laboratory MIT , Cambridge.

[15] TADEUSIEWICZ R., KOROHODA P., *Komputerowa analiza i przetwarzanie obrazów*. Kraków, 1997.

[16] TORRESAN H., TURGEON B., IBARRA-CASTANEDO C., HEBERT P., MALDAGUE X.. *Advanced Surveillance Systems: Combining Video and Thermal Imagery for Pedestrian Detection.* Electrical and Computing Engineering Dept., Universite Laval, Quebec City.

[17] TSUJI T., HATTORI V., WATANABE M., NAGAOKA N., *Development of nightvision.* Transactions on Intelligent Transportation Systems, 3:3 (2002).

[18] VIOLA P., JONES M., SNOW D., *Detecting Pedestrians Using Patterns of Motion and Appearance.* International Journal of Computer Vision, 2, 63 (2005), 153–161.

[19] ZHAO L., THORPE C., *Stereo- and Neural Network-Based Pedestrian.* IEEE Transaction on Intelligent Transportation Systems, 1, 3 (wrzesień 2000).

data mining, missing values,
thoracic surgery, risk modelling

Marek LUBICZ*, Konrad PAWEŁCZYK**

ANALYSING PERFORMANCE OF CLASSIFIERS FOR MISSING DATA IN THORACIC SURGERY RISK MODELLING IN STATISTICA DATA MINER ENVIRONMENT

Further results in analysing performance of classifiers for missing data in thoracic surgery (TS) risk modelling are reported for experiments made in Statistica Data Miner environment. Brief comments on current results in applications of quantitative modelling for TS data are presented, as well as a comprehensive description of updated and extended TS data bases from Wroclaw TS Centre. Application of Statistica Data Miner Recipes (DMR) is presented with special attention paid to initial data analysis, cleansing, feature reduction and missing values problems. A number of techniques for dealing with missing values, and also for feature reduction, available in the DMR environment are compared with TS data classification tasks. The classifiers available in the DMR module include decision trees and boosting trees, Random Forests, Neural Networks and Support Vector Machine. Performance of particular imputation techniques for specific classifiers is compared. Most promising classifiers for the problem domain are suggested.

1. INTRODUCTION

This work deals with Thoracic Surgery (TS) data analysis and risk modelling. The main aims in this domain are concerned with extending traditional retrospective description of samples (data on patients and operations) and statistical analyses of the indices of data structure or potential association between output (e.g. survival, intra or

* Faculty of Computer Science and Management, Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław.

** Department and Clinic of Thoracic Surgery, Wrocław Medical University, Wrocław Thoracic Surgery Centre, Lower-Silesian Centre for Pulmonary Diseases, Wrocław, Poland,

e-mail: wot@thorax.am.wroc.pl

postoperative co-morbidities) and input variables (risk factors), with prospective approaches, which are supposed to support clinicians in their decisions (e.g. suggesting treatment or the most probable course of action). As explained in [8], formal modelling of risks for morbidity and mortality is important in appropriate patient selection for surgery, counselling patients as part of the surgical consent process, stratifying outcomes for research purposes, and assessment of resource utilization. It has been mainly due [2] to anticipation of increasing prediction accuracy, enabling predicting individual outcome of potential resections instead of classifying patients to statistical risk groups, better simplicity in constructing models and ability to detect non predefined relations between predictors and clinical outcomes. Such approaches could have been of interest also to (clinical) management decisions, as they could eventually lead to rational clinical resource allocation.

Data Mining (DM) and Machine Learning (ML) had been explored for years for TS survival prediction [1, 2, 5–6, 10, 12, 14, 16], with most popular techniques being decision trees (DT), artificial neural networks (ANN) and support vector machines (SVM). However, in the published results one can hardly find unified TS risk models, or common agreement on most promising modelling approaches [13]. An important issue influencing the effectiveness of formal modelling is completeness and accuracy of the data used in various models, in particular – missing or unknown data, which are a common drawback of real-life clinical studies. Also in TS risk analysis papers, the issue of missing data is studied more often [3–4, 7–9, 11, 13, 15].

In a previous paper [17] the problem has been studied using models available in WEKA DM environment. A number of approaches used to deal with unknown values of attributes for the TS classification problem were compared resulting in disputable results: accuracy of classification of 70–80% which was inacceptable for the medical classification task. Moreover, no dramatic differences have been observed between different imputation techniques, as well as between predictions made on highly complete (1–2% missing values) and highly incomplete (58–62% missing values) datasets.

Aiming at achieving acceptable levels of at least true positives for postoperative survival, both on-site clinical data preparation and formal data analysis modelling approaches have been reviewed and substantially extended. The data cleansing process and initial clinical data investigations had been extended to reduce the extent of missing values and to include new clinical predictive variables. The modelling work was done using Statistica Data Miner Recipes (DMR) to perform feature reduction, compare approaches to dealing with missing values, and suggest acceptable classifiers for a number of output variables (30 days, 1 year, >1year survival). The purpose of the current study was also to analyze and compare effectiveness of classifiers in relation to the scope of the features vector (data from pre-, peri-, and post-operative period), and the length of the observation period.

2. DATABASES

The main data set used in this study is based on a research database constituting a part of the National Lung Cancer Registry, administered by the Institute of Tuberculosis and Pulmonary Diseases at Warsaw, Poland. The data set includes data on all consecutive first major lung resections for primary lung cancer (LC), performed in Wroclaw Thoracic Surgery Centre (WTSC) in the period 2007–2011. The Centre is associated with the Department of Thoracic Surgery of the Medical University of Wroclaw and Lower-Silesian Centre for Pulmonary Diseases, Poland).

Data sets used in previous studies [10, 17] have been reworked to include new clinical variables (up to the total of 139) and new cases from the period 2007–2011 (the final data set includes 1172 cases).

The cases were divided into three sets, corresponding to the year of operation: patients operated in 2007–2008, 2009, and 2010–2011, with mean observation period of: 42, 24, and 10 months respectively. The size of the sets was 542, 290 and 340 respectively. Most patients come from Lower Silesia (LS) region (75%) and one neighbouring region (20%), and also the vast majority of surgery for LC for LS citizens is performed at WTSC.

Data on each patient include variables referring to the specific stages of surgical hospitalization of an LC patient; the variables were divided into four subgroups: pre-operative, peri-operative, pathology and post-operative. The first group included 35 variables: demographic information, preoperative symptoms, co-morbidities, and – for 50% of patients – preoperative FVC and FEV1. It also included clinical TNM categories. The second group included 29 variables, describing technical features of the resection, detailed location and anatomical structures (including number and location of resected nodules), peri-operative complications, and surgical TNM staging. Next group consists of 48 pathological variables, including grading, histology, local invasion, staging, necrosis, and pathological TNM. Finally the last group consists of 27 post-operative variables, describing post-op treatment, complications, and follow-up.

Additional data on patients operated in the years 2000–2006, as well as more detailed concerning multimodal (radiotherapy, induction/neo-adjuvant chemotherapy) are in preparation.

The rates of missing values for most variables in the final data set were less than 0,5%, however there were six variables with missing rate between 49% (G-factor) and 62% (pre-operative FVC and FEV).

The outcome measures used in this study were: risk of 30days and 1year postoperative mortality, post-operative and overall (from the time of first diagnosis) survival, pulmonary and cardiovascular morbidity. However, in this work the results only for the two binary risks (30 days, 1year) and postoperative survival (treated as a categorical variable) are provided because of the lack of space.

3. STATISTICA DATA MINER RECIPES

Statsoft Statistica is one of the commercial data analysis environments, extensively used in academic institutions. At Wroclaw University of Technology it is available on a campus based license, apart from Matlab or SPSS in particular. In this study we selected Statistica due to a wide selection of Data Mining and Machine Learning techniques. To ensure timely completion of this part of the study we used Data Miner Recipes (DMR) module of Statistica, which constitutes interactive environment for defining and performing data mining experiments through semi automated building advanced analytic models. For categorical variables the following classifiers are available in the DMR module: decision trees (CART) and boosted trees, Random Forests, Neural Networks and Support Vector Machine. There are default parameters for specific classifiers but it is also possible to specify user-defined parameters for modelling. The modelling phase is preceded by preliminary data analysis, including missing data management, pre-processing, dimensionality reduction, and elimination of redundancy in the data set.

In general, most popular approaches to dealing with missing data are: ignoring incomplete cases (default suggestion in some DM packages, though leading to a reduction in the statistical power and biased results), imputation or estimation of missing data and ML using the edited training set, using model-based procedures (e.g. expectation–maximization algorithm), using particular ML procedures, where missing values are incorporated to the classifier.

In DMR module missing data definition and transformation are contained in the preliminary data pre-processing stage (advanced data preparation). The procedures to deal with missing data are defined differently for continuous and categorical variables. For the first type recoding to mean is provided, while for the latter – recoding to most frequent categories. In addition there are two other options: a standard approach of eliminating cases with missing values is available, and an ‘automatic imputations’ option, which consists in automatically substituting missing data with values that are estimated (imputed) from the data. Automatic imputation is based on k-nearest neighbours algorithm.

In addition to the use of DMR module the current study incorporated the use of Feature Selection and Variable Screening (FSL) module, which can be seen as a pre-processor for predictive data mining, to select manageable sets of predictors. This module enables to automatically analyse large sets of continuous and/or categorical predictors, for regression or classification-type problems and select a subset of predictors from a large list of candidate predictors without assuming any relationships between the predictors and the outcome variables. The module uses ILP database connection technology and for categorical variables computes Chi-square statistic and p value for each predictor and for selected output variable.

4. PRESENTATION OF THE RESULTS

The experimental part of the study comprised interactive calculations with Statistica DMR, comparing effectiveness (percentage accuracy of prediction) for particular classifiers and techniques of dealing with missing values, using all available and specific of data, divided into periods of observation and stage of surgical hospitalization. The following techniques for missing values have been considered:

- **MVE**: Instances containing at least one missing value of an attribute are eliminated,
- **MVR**: Missing value is replaced with most time observed value,
- **MVI**: Automatic imputation based on k-nearest neighbours algorithm.

In addition to various approaches of dealing with missing values, different models of classifiers were considered in the study:

- **CART**: decision trees,
- **RF**: random forest ensemble of simple tree classifiers
- **BT**: stochastic gradient boosted trees
- **ANN**: neural networks, in this case: Multilayer Perceptron
- **SVM**: support vector machines.

The output variables taken into account were: 30-days (**Risk30**) and 1-year (**Risk1Yr**) postoperative mortality, and postoperative survival defined in 6 months age bands (**PopSur**).

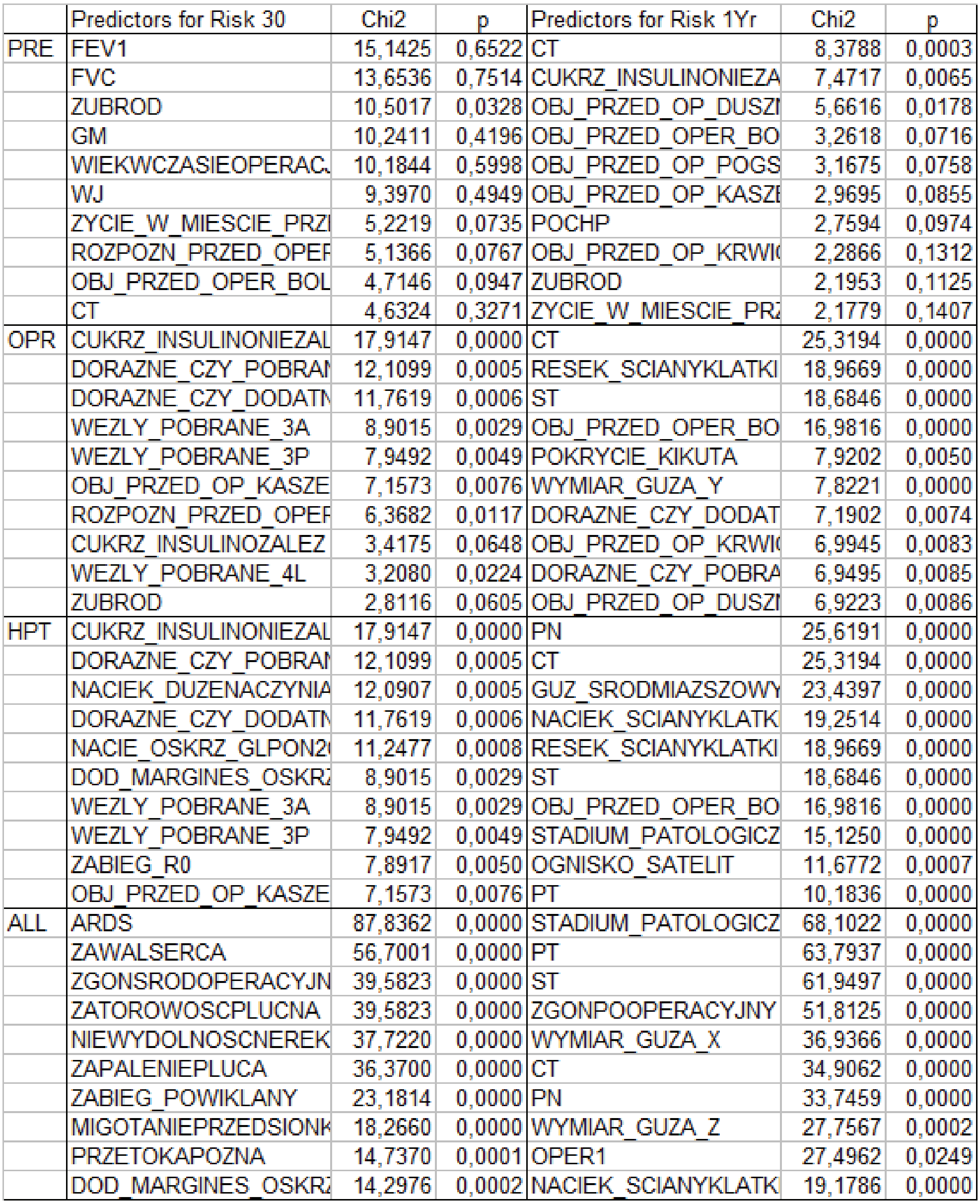
The experiments consisted in repeating calculations for all (categorical) output variables, selected five classifiers and three missing data approaches. The following default parameters have been used for particular classifiers:

- **CART**: maximum number of levels in tree 10, maximum number of nodes 200, minimum child node size to stop 9,
- **RF**: number of trees 100, maximum number of levels in tree 10, maximum number of nodes 100, minimum child node size to stop 5
- **BT**: maximum number of levels in tree 10, maximum number of nodes 13, minimum child node size to stop 1, minimum number to stop 46
- **ANN**: number of hidden units 3–10, networks to train 5, networks to retrain 1
- **SVM**: kernel type RBF, maximum number of iterations 500, stop at accuracy 0,001.

The data subsets denoted as groups **GR1, GR2, GR3** correspond to patients operated in 2007–2008, 2009, and 2010–2011 respectively. The variables (predictors) subsets refer to the specific stages of surgical hospitalization of a LC patient; the variables were divided into four subgroups: pre-operative (PRE), peri-operative (OPR), pathology (HPT) and post-operative (the subgroup ALL includes all available predictors).

The tables below present selected results of the computer experiments using the data and methods specified above. We started from looking at most informative predictors for specific output variables using FSL module of Statistica. Table 1 contains 10 best predictors for specific stages of surgical treatment.

Table 1. Ten best predictors specified by Feature Selection and Variable Screening (FSL) module



	Predictors for Risk 30	Chi2	p	Predictors for Risk 1Yr	Chi2	p
PRE	FEV1	15,1425	0,6522	CT	8,3788	0,0003
	FVC	13,6536	0,7514	CUKRZ_INSULINONIEZA	7,4717	0,0065
	ZUBROD	10,5017	0,0328	OBJ_PRZED_OP_DUSZI	5,6616	0,0178
	GM	10,2411	0,4196	OBJ_PRZED_OPER_BO	3,2618	0,0716
	WIEKWCZASIEOPERAC.	10,1844	0,5998	OBJ_PRZED_OP_POGS	3,1675	0,0758
	WJ	9,3970	0,4949	OBJ_PRZED_OP_KASZI	2,9695	0,0855
	ZYCIE_W_MIESCIE_PRZI	5,2219	0,0735	POCHP	2,7594	0,0974
	ROZPOZN_PRZED_OPEF	5,1366	0,0767	OBJ_PRZED_OP_KRWI(	2,2866	0,1312
	OBJ_PRZED_OPER_BOL	4,7146	0,0947	ZUBROD	2,1953	0,1125
	CT	4,6324	0,3271	ZYCIE_W_MIESCIE_PR.	2,1779	0,1407
OPR	CUKRZ_INSULINONIEZAL	17,9147	0,0000	CT	25,3194	0,0000
	DORAZNE_CZY_POBRAI	12,1099	0,0005	RESEK_SCIANYKLATKI	18,9669	0,0000
	DORAZNE_CZY_DODATI	11,7619	0,0006	ST	18,6846	0,0000
	WEZLY_POBRANE_3A	8,9015	0,0029	OBJ_PRZED_OPER_BO	16,9816	0,0000
	WEZLY_POBRANE_3P	7,9492	0,0049	POKRYCIE_KIKUTA	7,9202	0,0050
	OBJ_PRZED_OP_KASZE	7,1573	0,0076	WYMIAR_GUZA_Y	7,8221	0,0000
	ROZPOZN_PRZED_OPEF	6,3682	0,0117	DORAZNE_CZY_DODAT	7,1902	0,0074
	CUKRZ_INSULINOZALEZ	3,4175	0,0648	OBJ_PRZED_OP_KRWI(	6,9945	0,0083
	WEZLY_POBRANE_4L	3,2080	0,0224	DORAZNE_CZY_POBRA	6,9495	0,0085
	ZUBROD	2,8116	0,0605	OBJ_PRZED_OP_DUSZI	6,9223	0,0086
HPT	CUKRZ_INSULINONIEZAL	17,9147	0,0000	PN	25,6191	0,0000
	DORAZNE_CZY_POBRAI	12,1099	0,0005	CT	25,3194	0,0000
	NACIEK_DUZENACZYNIA	12,0907	0,0005	GUZ_SRODMIAZSZOWY	23,4397	0,0000
	DORAZNE_CZY_DODATI	11,7619	0,0006	NACIEK_SCIANYKLATK	19,2514	0,0000
	NACIE_OSKRZ_GLPON2(	11,2477	0,0008	RESEK_SCIANYKLATKI	18,9669	0,0000
	DOD_MARGINES_OSKR.	8,9015	0,0029	ST	18,6846	0,0000
	WEZLY_POBRANE_3A	8,9015	0,0029	OBJ_PRZED_OPER_BO	16,9816	0,0000
	WEZLY_POBRANE_3P	7,9492	0,0049	STADIUM_PATOLOGICZ	15,1250	0,0000
	ZABIEG_R0	7,8917	0,0050	OGNISKO_SATELIT	11,6772	0,0007
	OBJ_PRZED_OP_KASZE	7,1573	0,0076	PT	10,1836	0,0000
ALL	ARDS	87,8362	0,0000	STADIUM_PATOLOGICZ	68,1022	0,0000
	ZAWALSERCA	56,7001	0,0000	PT	63,7937	0,0000
	ZGONSRODOPERACYJN	39,5823	0,0000	ST	61,9497	0,0000
	ZATOROWOSCPLUCNA	39,5823	0,0000	ZGONPOOPERACYJNY	51,8125	0,0000
	NIEWYDOLNOSCNEREK	37,7220	0,0000	WYMIAR_GUZA_X	36,9366	0,0000
	ZAPALENIEPLUCA	36,3700	0,0000	CT	34,9062	0,0000
	ZABIEG_POWIKLANY	23,1814	0,0000	PN	33,7459	0,0000
	MIGOTANIEPRZEDSIONK	18,2660	0,0000	WYMIAR_GUZA_Z	27,7567	0,0002
	PRZETOKAPOZNA	14,7370	0,0001	OPER1	27,4962	0,0249
	DOD_MARGINES_OSKR.	14,2976	0,0002	NACIEK_SCIANYKLATK	19,1786	0,0000

Preliminary data processing in DMR module started from comparisons of missing data approaches (Table 2). We then compared prediction accuracy for two short term output variables, looking at possible relation between accuracy and the length of the observation period (Table 3). Finally in Tables 4 and 5 we present detailed results of comparing prediction accuracy for Risk 30 and Risk 1Year when predictors available at particular stages of patient hospitalization are taken into account.

Table 2. Percentage accuracy of classification for particular approaches to missing data

	output	**Accuracy – training**			**Accuracy – testing**		
		MVE	**MVR**	**MVI**	**MVE**	**MVR**	**MVI**
CART	**PopSur**	24,3	33,3	35,3	22,1	27,7	23,1
RF		24,3	36,0	49,1	14,7	27,7	37,2
BT		44,2	46,3	67,8	27,9	31,9	36,5
ANN		76,0	68,4	70,9	35,3	16,3	27,6
SVM		38,5	30,0	27,9	33,8	22,9	24,4
CART	**Risk30**	87,6	82,9	83,1	88,2	78,3	82,1
RF		97,4	97,9	97,6	100	96,4	94,9
BT		97,4	97,6	97,9	100	96,4	95,5
ANN		98,7	98,6	98,7	100	96,4	95,5
SVM		97,7	97,6	97,9	100	96,4	95,5
CART	**Risk1Yr**	74,9	47,6	83,1	64,7	48,8	82,1
RF		88,6	75,4	97,6	82,4	71,1	94,9
BT		88,6	83,9	97,9	86,8	79,5	95,5
ANN		92,8	88,7	98,7	86,8	78,9	95,5
SVM		88,9	84,2	97,9	88,2	80,1	95,5

Table 3. Comparison of prediction accuracy in relation to the length of observation (stage ALL)

	output	**Accuracy – training**			**Accuracy – testing**		
		GR1	**GR2**	**GR3**	**GR1**	**GR2**	**GR3**
CART	**Risk30**	94,2	88,4	89,6	92,0	87,4	82,7
RF		97,4	97,8	97,5	98,2	93,7	97,7
BT		97,9	98,2	99,1	98,2	95,6	96,7
ANN		99,3	99,4	98,5	98,2	95,6	96,7
SVM		97,4	97,9	98,1	98,2	95,6	97,7
CART	**Risk1Yr**	78,1	79,6	72,9	65,5	67,9	72,9
RF		78,6	81,7	81,7	63,7	71,1	79,0
BT		84,9	85,4	85,4	74,3	76,1	83,6
ANN		91,1	91,4	90,7	80,5	74,8	84,6
SVM		80,9	83,1	83,3	81,4	78,6	85,1

Table 4. Prediction accuracy for Risk30 and specific classifiers, variables sets and data groups

	stage	**Accuracy – training**			**Accuracy – testing**		
		GR1	**GR2**	**GR3**	**GR1**	**GR2**	**GR3**
CART	PRE	98,8	94,3	87,6	100	88,0	88,2
RF		98,8	98,1	97,4	100	100	100
BT		100	100	97,4	100	100	100
ANN		100	100	98,7	100	100	100
SVM		98,8	98,1	97,7	100	100	100
CART	OPR	98,8	99,0	92,8	100	97,1	73,0
RF		98,8	99,0	97,7	100	97,1	98,4
BT		100	100	97,7	100	100	98,4
ANN		100	100	99,5	100	94,1	98,4
SVM		98,8	99,0	97,4	100	97,1	98,4
CART	PTH	95,7	92,6	82,1	96,0	80,3	74,1
RF		97,7	97,7	97,8	98,0	96,6	95,0
BT		97,7	97,8	98,0	97,0	96,6	95,0
ANN		98,6	99,3	98,6	98,0	96,6	95,0
SVM		97,5	97,7	97,9	98,0	96,6	95,0

Table 5. Prediction accuracy for Risk1Year and specific classifiers, variables sets and data groups

	stage	**Accuracy – training**			**Accuracy – testing**		
		GR1	**GR2**	**GR3**	**GR1**	**GR2**	**GR3**
CART	PRE	88,4	84,0	74,9	84,2	68,0	64,7
RF		84,9	82,1	88,6	79,0	80,0	82,4
BT		88,4	84,9	88,6	79,0	84,0	86,8
ANN		93,0	92,5	92,8	73,7	80,0	86,8
SVM		80,2	98,1	88,9	84,2	76,0	88,2
CART	OPR	75,6	84,4	78,2	40,9	79,4	69,8
RF		79,3	84,4	88,7	77,3	82,4	85,7
BT		95,1	86,5	88,0	77,3	73,5	90,5
ANN		93,9	94,8	95,4	86,4	82,4	92,1
SVM		78,1	81,3	86,9	86,4	82,4	93,7
CART	PTH	80,5	71,2	73,3	70,0	59,9	68,2
RF		81,2	80,4	82,3	70,0	70,1	78,2
BT		88,5	83,7	83,5	77,0	78,9	81,4
ANN		90,7	91,5	92,9	76,0	80,3	82,3
SVM		83,3	80,7	83,0	80,0	81,6	82,3

5. DISCUSSION AND CONCLUDING REMARKS

The results presented here may be interpreted from two contrasting points of view, namely, clinical and analytical (modelling). For a thoracic surgeon the most important issues in the study are concerned with predictive abilities of the modelling approach, especially for pre-operative clinical decision making on short term (30 days) or medium term (1 year) survival after operation. While the results are satisfactory for short term predictions (over 97% accuracy when predicting using only pre-operative variables), they are less acceptable for medium term survival (70–80% accuracy), and completely unacceptable for long term survival (table 1). As expected the accuracy for 30days prediction is almost independent of the length of post-operative observation period (greatly affected on patient's preoperative status), while it deteriorates with shorter periods for medium term survival. What possibly was not expected is that predictions of 1 year survival do not improve with more data on the details of the operation and the results of pathology examination: it is not confirmed in the results presented in table 5, although it could have been expected taking feature selection parameters for OPR and HPT stages (table 1) into consideration.

On the other hand, the results in Table 2 confirm better effectiveness of automatic imputation (MVI) for medium term predictions (1 Year), but not for short (30 days) or long term predictions. However, for long term predictions, the results are far from being acceptable. It should be noted that the number of missing values in the data set used for this study was relatively low; we may expect more distinct results for other data sets used in previous studies (operations from 2000–2006), in which the extent of missing values is much higher. We may also conclude that Decision Trees (especially CART) have done badly comparing to Support Vector Machines and particularly to Neural Networks, independently of data sets (GR1-GR3) or predictor sets (PRE, OPR, HPT stages).

The study confirmed previous opinion of the authors [10] that actual progress in improving prediction accuracy in thoracic surgery decision making, in addition to research on formal modelling approaches, requires first of all – proper, representative, clinical databases of acceptable quality. The authors continue pre-processing work on other available WTSC data bases, as well as theoretical work on specific modelling approaches, which could enable improvements in survival prediction (ensemble modelling, approaches for censored survival data).

REFERENCES

[1] AVCI E. (2011). A New Expert System for Diagnosis of Lung Cancer: GDA-LS_SVM. Journal of Medical Systems (in press).

[2] BELLAZI R., ZUPAN B., *Predictive data mining in clinical medicine: Current issues and guidelines*, International Journal of Medical Informatics (2008), 77(2): 81–97.

[3] BLOUGH D.K., RAMSEY S., SULLIVAN S.D., YUSEN R. (2009). *The impact of using different imputation methods for missing quality of life scores on the estimation of the cost-effectiveness of lungvolume-reduction surgery*. Health Economics, 18(1), 91–101.

[4] DEKKER A., DEHING-OBERIJE C., DE RUYSSCHER D., LAMBIN P. . HOPE A., KOMATI K., FUNG G., YU S., DE NEVE W., LIEVENS Y. , *Survival Prediction in Lung Cancer Treated with Radiotherapy. Bayesian Networks vs. Support Vector Machines in Handling Missing Data*, Proceedings of ICMLA2009 (2009) 494–497.

[5] DELEN D., OZTEKIN A., KONG Z.(J.)., *A machine learning-based approach to prognostic analysis of thoracic transplantations*, Artificial Intelligence in Medicine (2010), 49: 33–42.

[6] ESTEVA H., NÚÑEZ T.G., RODRÍGUEZ R.O., *Neural Networks and Artificial Intelligence in Thoracic Surgery*, Thoracic Surgery Clinics (2007), 17(3): 359–367.

[7] EVANS N.R. III, LI S., WRIGHT C.D., ALLEN M.S., GAISSERT H.A. (2010). *The impact of induction therapy on morbidity and operative mortality after resection of primary lung cancer*. Journal of Thoracic and Cardiovascular Surgery, 139(4): 991–996.e2.

[8] FERGUSON M.K., SIDDIQUE J., KARRISON T., *Modeling major lung resection outcomes using classification trees and multiple imputation techniques*, European Journal of Cardio-Thoracic Surgery (2008) 34(5): 1085–1089.

[9] GARCIA-LAENCINA P.J., SANCHO-GOMEZ J-L., FIGUEIRAS-VIDAL A.R., *Pattern classification with missing data: a review,* Neural Computing and Applications (2010) 19:263–282.

[10] LUBICZ M., RZECHONEK A., PAWEŁCZYK K., KOŁODZIEJ J., *Transforming Data into Knowledge for Lung Cancer Surgical Patients*, in: M. Lubicz ed. Operational Research Applied to Health Services in Action, Oficyna Wydawnicza PWr, Wrocław 2009, 279–299.

[11] MARSHALL A., ALTMAN D.G., ROYSTON P., HOLDER R.L., *Comparison of techniques for handling missing covariate data within prognostic modelling studies: a simulation study*. BMC Medical Research Methodology (2010) 10: 7.

[12] SANTOS-GARCIA G., VARELA G., NOVOA N., JIMENEZ M.F., *Prediction of postoperative morbidity after lung resection using an artificial neural network ensemble*, Artificial Intelligence in Medicine (2004), 30(1): 61–69.

[13] TWALA B., *An Empirical Comparison of Techniques for Handling Incomplete Data Using Decision Trees*, Applied Artificial Intelligence (2009), 23(5): 373–405.

[14] VOZNUKA N., GRANFELDT H., BABIC A., STORM M., LÖNN U., AHN H.C. (2004). *Report Generation and Data Mining in the Domain of Thoracic Surgery,* Journal of Medical Systems, 28(5): 497–509.

[15] WRIGHT C.D., GAISSERT H.A., GRAB J.D., O'BRIEN S.M., PETERSON E.D., ALLEN M.S., *Predictors of Prolonged Length of Stay after Lobectomy for Lung Cancer: A Society of Thoracic Surgeons General Thoracic Surgery Database Risk-Adjustment Model*, Annals of Thoracic Surgery (2008) 85(6): 1857–1865.

[16] YOO I., ALAFAIREET P., MARINOV M., PENA-HERNANDEZ K., GOPIDI R., CHANG J-F., HUA L. (2011). Data Mining in Healthcare and Biomedicine: A Survey of the Literature. Journal of Medical Systems (in press).

[17] ZIĘBA M., LUBICZ M., *Performance of classifiers for missing data in thoracic surgery risk modelling using WEKA-based data mining approaches*, in: Information Systems Architecture and Technology. IT Models in Management Process, Z. Wilimowska et al (ed.). Oficyna Wydawnicza PWr, 2010; 435–445.

biometrics, authentication,
behavioral features,mouse movements

Agata KOŁAKOWSKA*, Rafał PRASAŁ*

IDENTITY VERIFICATION BASED ON MOUSE MOVEMENTS

This work presents the idea of a biometric security system based on the way of using a mouse. The mouse actions parameters are examples of behavioral features, which are not stable at time in contrast to physiological characteristics. Though their advantage is the possibility of applying without using any special hardware and without interrupting users. A set of features, which may be extracted from data coming from a mouse, has been described. Training data from thirteen users has been collected and classifiers have been trained and tested using two machine learning methodologies: support vector machines and decision trees. The results show, that it is possible to authenticate users on the basis of their mouse movements, but it is not as reliable as other known biometric methods. However it may be treated as an additional protection or it may be combined with other methods to improve accuracy.

1. INTRODUCTION

Nowadays more and more computer systems are protected using methods based on biometric techniques, which analyze one of two types of human characteristics to verify one's identity. These are either physiological (face, palm, fingerprint, iris, vein topography) or behavioral (voice, handwritten signatures, keystroke dynamics, mouse movements) features. Physiological features are much more common, because of their stability along time. There are a lot of successful applications based on these parameters, for example notebooks protected by fingerprint scanning. The difficulty with behavioral features is that the samples of one user may vary strongly every time they are recorded. One's way of using a mouse depends not only on the time when it is recorded, but also on the type of hardware used and program operated with a mouse. Generating a profile of a user in such situation becomes really challenging. The advan-

* Gdańsk University of Technology, ul. Narutowicza 11/12, 80-233 Gdańsk.

tage of these characteristics is that these are very natural types of biometrics which do not require any special hardware. Moreover, they are not as intrusive as some other methods. It is possible to collect the necessary data during the normal computer usage, so a user does not have to spend time on any additional actions.

The amount of research focused on analyzing mouse movements in order to identify users is not as high as those on other biometric characteristics, even behavioural ones, like for example keystroke dynamics. However there are some studies reporting promising results [1, 4, 6, 7, 8]. They differ in the way of extracting features, which in general reflect the type, speed and direction of different mouse actions. To build a classifier able to distinguish between an authorised user and an impostor, they apply various learning methods, i.e. decision trees [7], *k* nearest neighbours [4, 8], support vector machines [4], neural networks [6]. Moreover some of the presented experiments are performed only in selected applications, for example Windows Explorer [7] or selected games [4]; some create special baseline programs to collect and test the methods [4, 8]; whereas the others do not make any limitations in this matter. Due to this variety of experimental environment, it is impossible to compare the efficiency of all the mentioned methods.

This work presents a study on user authentication via mouse movements performed in Windows environment without any limitations on the operated software. The feature extraction stage is based on a slightly modified idea reported in [1], which is described in section 2. Then two machine learning approaches, used to authenticate users, are described. Finally the experimental results, and some conclusions on further improvement of the system are presented.

2. EXTRACTING FEATURES

A mouse generates two types of data when it is used, the parameters of the movements and data coming from the buttons. When a mouse is moved, then its position is registered every *t* ms, so a move is defined as a path of points with the times of reaching them, as it is shown on Fig. 1. Moreover all the click times are saved. The raw data is analyzed and features are extracted [1] as it is described in the next sections.

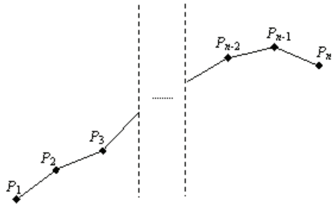


Fig. 1. The points of a mouse move

2.1. ACTION TYPES

Three types of mouse actions may be distinguished: mouse move, point and click, drag and drop. The frequency of these actions, which obviously depends on the application used, is measured. Experiments show that the percentage of mouse move and point and click actions do not vary much between one user's sessions, so these two parameters have been selected for the feature vectors.

2.2. CLICKING

Clicking a button generates a few parameters. In the case of a click it is the time between pressing and depressing a button, no matter if it is left or right button. In the case of the left button double clicks may also occur. In this case the times registered are: the time between the first pressing and the first depressing of the button, the time between the first and the second depressing of the button.

Moreover it is also possible to register the percentage of different clicks in a given period of time. These parameters strongly depend on the software used, for example the number of clicks while writing or reading a document may be really low, whereas the same parameter may reach maximum values while using a graphical tool. However it is also worth noting this to observe possible correlations with others parameters.

2.3. REACTION TIME

A specific parameter strongly dependant on a user is the interval between the end of a move and pressing a button, called reaction time. An opportunity to register this characteristic occurs quite often, for example when one moves a mouse to click the Windows OK button, to start drawing a line, to select an option from a menu, to place a cursor etc. Moves usually end up with an activity indicated by clicks.

2.4. MOVING A MOUSE

Moving a mouse generates much information on the way of using this device. If the path of a move consists of *n* points, then its length is obviously calculated in the following way:

$$l = \sum_{i=1}^{n-1} d(P_i, P_{i+1}) \tag{1}$$

where $d(P_i,P_{i+1})$ is the Euclidean distance between the two points P_i and P_{i+1}.

After analyzing the data three ranges of the path length have been selected: [25px–100px], (100px–200px], (200px–400px]. The percentage of moves of these lengths would be the next three features.

Moreover the distance between the first and the last point of a path may be calculated. This parameter, called direct distance, may differ from the path length, as it is shown on Fig. 2.

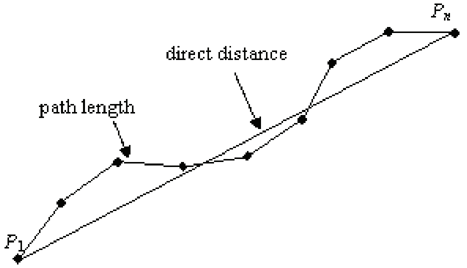


Fig. 2. The path length and the direct distance parameters

The most interesting is to compare the two parameters already mentioned. This comparison may give us some information on users preferences. Some users usually move a mouse along straight lines, which means that the path lengths and the direct distances are almost equal. Whereas the others are used to change the direction during a single move. A quotient parameter has been defined to illustrate this preference:

$$q = \frac{l}{d(P_1, P_n)} \tag{2}$$

Three ranges of quotient values have been regarded as worth analyzing. These are: [1.0-1.1), [1.1-1.2), [1.2-1.5] and they correspond to straight paths, slighty curved and curved. The percentage of the moves for the three ranges of quotient have been added as features.

The next parameter, which might be recorded for further analysis, is the direction of a move defined as an angle between the line connecting the starting and ending points and a horizontal line. This parameter has been discretized to 12 values and the percentage of the moves for those 12 directions have been added to the feature vector.

It has been observed that users, while moving a mouse toward a given point, move away from the shortest route in different ways. Some of the paths keep on one side of the line, some of them cross the line once, and the others cross the line a few times. It has been shown in Fig. 3.

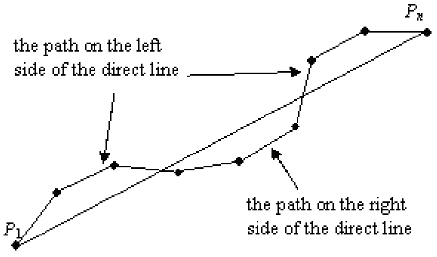

P_n
the path on the left side of the direct line
P_1
the path on the right side of the direct line

Fig. 3. An example of a path placed on the left and right side of the direct line

To measure this tendency, a skewness parameter has been defined. It compares the total length of a path to the length o its part placed on one side of a straight line connecting the starting and the ending points and is calculated as follows:

$$s = \frac{l_L}{l} \tag{4}$$

where l_L is the length of the path on the left side of the direct line. The feature vector contains 12 values reflecting the correlation between the skewness and the direction. These are the average skewness values calculated for the 12 direction ranges mentioned before.

Another characteristic parameter worth noting is the average velocity of a move calculated as follows:

$$v = \frac{l}{t} \tag{3}$$

where t is the time of moving along a path of length l. The velocity itself has not been regarded as a feature, but some features reflecting its correlation with other parameters have been proposed. These are: the average velocity for paths of different lengths calculated for 9 ranges of path lengths, the average velocity of moves of different directions calculated for the 12 possible direction ranges, the average velocity for the mouse moves and finally for the point and click actions.

2.5. CREATING FEATURE VECTORS

Different approaches have been proposed to transform the measurements into feature vectors. In [7] every user was treated individually and for each of them an extraction window consisting of 1500–2500 mouse events was selected during experiments.

The parameters measured in such a window were averaged. In [1] all users were treated equally and the window for all of them contained 2000 events.

In this work the time of collecting data was chosen as an extraction window. The experiments have been performed for different values of this time. The final results strongly depend on this parameter, which has been mentioned in Section 4. All the parameters described in the previous subsections are measured and their values are averaged over the extraction window.

3. TRAINING

Two learning methods have been selected and tested to authenticate users on the basis of their mouse movements: decision trees and support vector machines. In these experiments OpenCV library has been used to implement both methods.

3.1. DECISION TREES

Decision tree is a popular way of representing knowledge. Its popularity arises from the fact, that this representation is really comprehensive, which is important in many applications. They give good results in many classification tasks. Moreover decision trees are very effective from the point of view of space and time complexity. It is also very important that it is possible to use this method in the case of different feature types. The feature vectors may contain both numerical and symbolic parameters. The disadvantage of a tree is that it is difficult to compromise between its accuracy and complexity.

Most tree construction algorithms are top-down methods which create the nodes of a tree starting from the root node. The whole set of training examples is placed in the root. Then the data are split into subsets and directed to child nodes. The same procedure is repeated for the successive nodes until a stopping criterion is met. Splitting the nodes and stopping the construction are the essential stages which influence the size of the final tree and its accuracy. There are a lot o splitting criteria used to divide a set into subsets [5]. These criteria measure how good a feature is in splitting the data. The most popular ones are based on entropy (information measure), probability distributions (chi-square) or impurity estimation (Gini index). In general a good split is when it makes the classification in the child nodes easier than in the parent node. The best feature is selected and used to build a decision rule in the tree node. The number of descendant nodes is equal to the number of different values of the selected feature. In the case of continuous attributes they have to be discretized first.

The splitting is repeated until the nodes contain the examples of only one class or further splitting is impossible. A leaf is assigned the majority class label.

The problem is that such trees may not have good generalisation property. It means that a tree may perfectly classify the training data but show high error rate for unseen examples. This is called overfitting. To avoid such situation a tree should be pruned. There are various pruning algorithms [3], for example reduced error pruning, minimum error pruning, pessimistic error pruning, cost-complexity pruning. Most of them analyze node after node and estimate the error rate in the branch starting from that node with the error in a leaf replacing that branch. If the error rate in the whole branch is higher than in a leaf, then the branch is removed and replaced by a leaf.

To classify an object using a decision tree one has to put the object in the root node and then direct it to one of the descendants according to the results of the test performed in the node. It should be repeated until the object reaches a leaf node, where it is classified to one of the classes.

3.2. SUPPORT VECTOR MACHINES

The idea of support vector machines is to find the greatest possible margin between the examples of two classes. In the case of a linearly separable problems the margin is defined by two hyperplanes. It turns out that the position of the hyperplanes usually depends on a low number of training examples, called support vectors [2]. To find the optimal hyperplane dividing the classes it is necessary to solve the following quadratic programming problem:

$$F = \max_{\alpha}(\sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} \alpha_i \alpha_j d_i d_j x_i^T x_j) \tag{5}$$

where x_i are the training examples, $d_i = 1$ if x_i belongs to class c_1 and $d_i = -1$ if x_i belongs to class c_2, m is the number of training examples. Moreover the following limitations on the sought α_i parameters have to be fulfilled:

$$\sum_{i=1}^{m} \alpha_i d_i = 0, C \geq \alpha_i \geq 0 \tag{6}$$

Only some of the α_i values are not equal to zero. These are the α_i values corresponding to the support vectors, which in turn are taken into account in the final decision rule:

$$\begin{aligned} &c^*(x_j) = c_1 \ \text{ if } \sum_{i=1}^{n} \alpha_i d_i x_j^T x_i + b > 0, \\ &c^*(x_j) = c_2 \ \text{ otherwise} \end{aligned} \tag{7}$$

where n is the number of support vectors.

When the two classes are not linearly separable, then the common approach is to transform the data into a new space defined by a kernel function. The problem in the new space with higher number of dimensions may become linearly separable. The results depend on the type of kernel function. A popular one, also used in our experiments, is the radial function.

4. EXPERIMENTAL RESULTS

To collect the training data a special application was created. Thirteen people installed it on their computers. Two of them used touchpad only. One person took two sets of data separately, once using only a mouse and once using both a mouse and a touchpad. The application was transparent for the users. They were supposed to use their computers in the way they always did. They could use any application they wanted. In the case of each user the data was collected during 512 intervals of 15 minutes. It took them from 10 to 33 days depending on the time they spent using computer.

As it has been mentioned before, it is possible to collect data for different periods of time to create a single feature vector. The experiments were performed for a few time values: 30 seconds, 1 min., 3 min., 5 min., 60 min. It means that the same testing procedure was repeated for five data sets.

The mouse movement analysis may be used to solve one of three tasks. It may be applied either to classify, authenticate or to identify users. Classification means deciding which of the known users is currently using a mouse. In the case of authentication a user using a mouse claims to be the *U* user and the system is supposed to verify whether it is true or not. The last and the most complicated task is the identification. In this case new mouse movement data appears and the system has to decide if the mouse is used by one of the known users (giving his identity as in the case of the classification), or someone unknown.

The problem to solve in these experiments was user authentication. The solution was found by defining this problem as two-class case. If a user claims to be the *U* user, then a classifier trained to discriminate between user *U* and all other users has to be used.

For the two investigated methods (support vector machines and decision trees) the number of classifiers which had to be trained was equal to the number of users. In each of those cases both the false acceptance rate (FAR) and false rejection rate (FRR) were estimated. In the case of decision trees the classifiers were tested in a 10-fold cross validation procedure. Classifiers based on support vector machines required much more time to be trained, so in this case the data was randomly split into the training (90%) and testing set (10%) preserving the proportion of examples of the two classes as in the original set. The results obtained for both methods and for different times of collecting the training data are presented in Table 1 and Table 2.

Table 1. Results obtained using classifiers based on support vector machines

	30 seconds	1 min.	3 min.	5 min.	60 min.
FAR [%]	15.73	13.83	10.44	10.02	4.87
FRR [%]	20.67	17.21	14.79	12.99	15.90

Table 2. Results obtained using classifiers based on decision trees

	30 seconds	1 min.	3 min.	5 min.	60 min.
FAR [%]	3.73	3.63	3.09	2.87	1.24
FRR [%]	51.39	47.75	39.97	35.32	20.79

The results presented in the tables show a great influence of the data collecting time. The more information we gather to construct a feature vector, the better it is. This is due to the fact that if a user is moving a mouse for a long time, his movements become less accidental and therefore the feature values are more stable along time. Unfortunately in some applications it is essential to detect an intruder as quick as possible, so it would be important not to spend so much time on gathering the data. In such cases the creation of a feature vector should terminate after noting prespecified number of mouse actions, no matter how long it took.

The method based on decision trees shows really low values of FAR, but the respective FRR values are high comparing to the method based on support vector machines. In case of SVM the error rates are more balanced, however the choice of the method usually depends on an application. In some systems it is so important not to accept an intruder that one may even allow higher values of FRR.

As it has been mentioned before, a few users used touchpad. One of them used both a mouse and a touchpad. The data collected for different devices have different characteristic, for example different mean values or standard deviations. One user's mouse movements are completely different from the same person's touchpad movements. It means that to create a reliable security system a combination of both devices should be taken into account.

5. SUMMARY

The work presented a set of features which might be extracted from one's mouse movements [1]. Then two common machine learning methods have been described. These methods were applied to authenticate users on the basis of their mouse movements characteristics. The results of the experiments show that this biometric method is less reliable than other known methods based on physiological features, like for example fingerprint. However it might be treated as an additional protection method. The advantage of it is that it might be used without interrupting users. Mouse movements might be analyzed and in the case of rejection a user might be asked to proof his identity in another way, for example by showing his face. Another problem is that the classifiers used to authenticate users should be updated from time to time after gathering more data. If the time complexity of the training method is high, as it was in the case of support vector machines, then this may appear infeasible.

To improve the accuracy of the system, some investigations still may be done. First of all feature selection method may be applied to get rid of the parameters which do not have good discriminative properties. It would reduce the dimension and thus the time of training and applying the decision rule. Moreover other learning algorithms should be tested, especially those designed to detect outliers. Finally a system combining a few decision rules, constructed on the basis of biometric features of different types, might be created to improve the quality of the decision rules used separately.

ACKNOWLEDGEMENTS

This work has been supported by the Polish Ministry of Science and Higher Education as the research project No. N N516 36793.

REFERENCES

[1] BehavioSec company, *Mouse dynamics white paper, Behaviometrics – A Paradigm shift in computer security*, http://www.behaviosec.com.

[2] BURGES C.J.C., *A tutorial on support vector machines for pattern recognition*, Data Min. Knowl. Discov., Vol. 2, No. 2, 1998, 121–167.

[3] ESPOSITO F., MALERBA D., SEMERARO G., *A Comparative Analysis of* Methods *for Pruning Desision Trees*, IEEE Trans. PAMI, Vol.19, No.5, 1997.

[4] KAMINSKY R., ENEV M., ANDERSEN E., *Identifying Game Players with Mouse Biometrics*, University of Washington, 2008.

[5] MINGERS J., *An Empirical Comparison of Selection Measures for Decision-Tree Induction*, Machine Learning, Vol. 3, 1989, 319–342.

[6] NAZAR A., TRAORE I., AHMED A.A.E., *Inverse biometrics for mouse dynamics, Int. Journal of Pattern Recognition and Artificial Intelligence*, Vol. 22, No. 3, 2008, 461–495.

[7] PUSARA M., BRODLEY C.E., *User Re-authentication via mouse movements*, Proceedings of the ACM workshop on visualization and data mining for computer security, 2004.

[8] WEISS A., RAMAPANICKER A., SHAH P., NOBLE S., IMMOHR L., *Mouse Movements Biometric Indentification: A Feasibility Study*, Proceedings of Student/Faculty Research Day, CSIS, Pace University, 2007.

telemedicine, exercise planning, personalization
wearable sensors, pattern recognition

Anna SIEDLARCZYK*, Krzysztof BRZOSTOWSKI*

DEVELOPMENT OF WEARABLE SENSOR-BASED SYSTEM FOR SUPPORT ATHLETES TRAINING

In the work system for support athletes training is considered. Architecture of proposed system and suitable application is presented in details. Comprehensive analysis of functional and non-functional properties of system is given. Implementation details and results of experimentation are discussed as well.

1. INTRODUCTION

Wireless sensor networks are extremely useful tools which are successfully applied in many areas such as industry, healthcare, sport, emergency management and entertainment. Pervasive computing, wearable sensors, artificial intelligence techniques in conjunctions with wireless networks have built interdisciplinary research with open questions, challenges and potential to solve everyday life problems.

According to WHO (*World Health Organisation*) chronic disease such as diabetes, cardiovascular diseases are the leading cause of death worldwide. In 2004 an estimated 3.4 million of people died from consequences of high blood sugar. Cardiovascular disease was cause of death for 17.1 million people in 2004. It was 29% of all global deaths [20]. Risk factor for diabetes (Type II) and cardiovascular disease are e.g. unhealthy diet and low level of physical activity. It means that healthy lifestyle and diet can prevent diabetes and cardiovascular disease. Hence the need delivering easy-to-use and safe system to support management and monitoring of healthy life-

* Institute of Informatics, Wroclaw University of Technology, 50-370 Wroclaw, Wybrzeże Wyspiańskiego 27, {150207, Krzysztof.Brzostowski}@pwr.wroc.pl

style. Main functionalities such systems are recording, transmitting, data processing and support decision making. Advances in wireless technologies allow sensing physiological signals in real time. They are helpful to build sensor platforms called BAN (*Body Area Network*) and PAN (*Personal Area Network*). Usually such platform are composed of devices with Bluetooth and ZigBee interfaces. They help to design sensing platforms that are comfortable to use and not bothersome. It is characteristic the crucial non-functional features for healthcare systems.

On the market few different commercial equipments to wireless sensing are produced e.g. by Zephyr Technology and Shimmer Research manufactures. The first one is a developed construction produced with Bluetooth interface and has ability to sense following physiological signals: heart rate, breath rate, temperature, ECG and other such as acceleration. The second product is open and is still elaborating by many researchers teams and engineers. Technical documents offered by Shimmer Research manufacturer can be used to design various sensing systems. Configuration of the platform and applied sensors depends on application. It means that, for example, platform for athletes and its configuration can be different from the platform for monitoring old and impaired people.

Wireless platforms to record and transmit data can be put together with system to data processing and supporting decision making. It allows to give facilities results of diagnosis, monitoring and decision making in ubiquitous computing environment. It makes healthcare services available for all. Moreover it leads to reduce cost of medical care.

There are many different papers and reports describe solutions where wireless networking and data processing are connected together. General, we can distinguish two main groups of elaborated projects i.e. healthcare and wellness. Researches in Harvard University developing health care system for motion analysis of patients being treated for neuromuscular disorder [13]. To this group of application we can include system for monitoring elderly [12] and long-tern ambulatory health monitoring such as LiveNet [9].

Wellness and sport area is investigated as well. In [11] a system to optimization fitness training are described. Several research teams elaborating systems to remote monitoring athletes while exercising [2, 6, 7, 16]. In these works considered different problems of supporting training planning and monitoring but they have common denominator which is wireless sensing and data processing joined together.

2. SYSTEM FOR SUPPORT ATHLETE’S TRAINING

In this section System for Support Athlete’s Training (SSAT) is presented. This system is composed of two applications i.e.: to acquire data from heart rate, speed and distance monitor and, the second one, to manage athlete's training. First of them is designed as mobile application to be used with mobile phones. It allows to display in real time: (a) measurements transferred from mobile monitor and (b) real time calculation of values such as burned calories and (c) energy expenditure during training. The second application is used to managed training process. The main functionalities of the second one are ability to:

a) fix aim of the training i.e. gaining, losing or maintaining body weight;
b) define maximal level of heart rate during physical activity;
c) support cardiovascular fitness test at physical effort; the results of this test is maximal oxygen uptake (VO_{2max});
d) calculate training zones taking into consideration aim of the training, maximum heart rate, cardiovascular fitness, personal data;
e) measure body weight and calculate body fat, body water and daily caloric intake.

2.1. SSAT REQUIREMENTS SPECIFICATION

Section presents system requirements specifications refer to the capabilities of final system. Presented specification is a results of analysis user needs i.e. athlete and trainer and comply with technical limitations. We divide it on functional and non-functional requirements both for desktop and mobile application.

Functional requirements for desktop application

- Application must get data from body composition analyser;
- User should select appropriate test which estimates maximal volume oxygen uptake;
- User must choose desirable activity level;
- User must choose training goals;
- User must choose training time in specified day (i.e. 30 min., 45 min. or 60 min.);
- User could refer to the exercises while correspond to a specific metabolic equivalent values (MET);
- User could generate charts for the last 30 measurements;
- User could generate chart from present training week. It helps to evaluate number of burnt calories;
- User could fix number of calories to be burnt each day;
- User could save and get data to/from database.

Functional requirements for mobile application

- User must enter personal data;
- System must be connected to a heart rate monitor;

- User could observe the condition of his/her heart rate, training time and burnt calories.

Non-functional requirements for desktop application

- User must pair the body composition analyser with computer;
- User must installed Matlab application.

Non-functional requirements for mobile application

- User should use the desktop application for training support;
- User must wear heart rate monitor;
- User must enable Bluetooth interface in mobile phone;
- System must be comfortable to use and not bothersome.

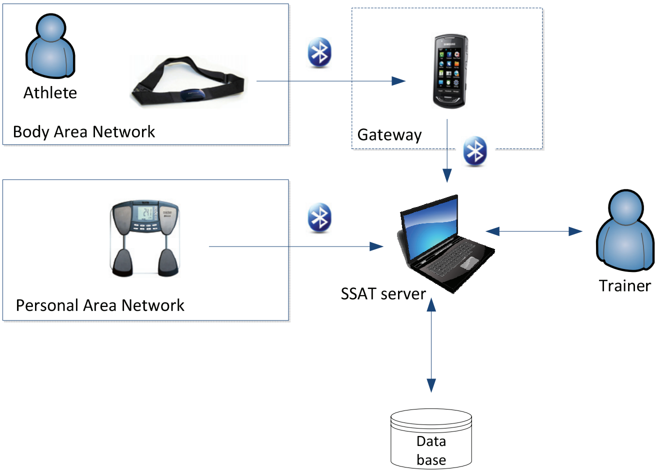


Fig. 1. SSAT architecture

2.2. SSAT SYSTEM ARCHITECTURE

In Fig. 1 architecture of SSAT is presented. The system is composed of SSAT server for main desktop application which is connected with data base and wireless measurement devices such as body composition scale and heart rate monitor. Physiological (i.e. heart rate, body weight) and kinematical signals (i.e. speed) are transferred through Bluetooth to desktop application. Then they are processed and giving facilities both athlete and trainer.

2.3. SSAT APPLICATION ARCHITECTURE

This section presents architecture of desktop and mobile applications. Mobile application is designed for athletes and, on the other hand, desktop application can be used both athletes and trainers. In Fig. 2 main modules of the applications are presented. On the left main modules of desktop application are given. At a coarse level of granularity acquisition, processing and presentation block can be distinguished. Acquisition block is designed to collect data from wireless body composition scale. Before processing transferred measurements additional data related to results of fitness test (i.e. VO_{2max} value) and calories to be burnt must be entered. In the second block following modules are implemented to determine:

a) maximal heart rate during physical activity;

b) training zones;

c) calories to be burnt.

As it can be seen in Fig. 2 input block of desktop application has three modules. The first one is **body composition data** to get data from body composition scale. Next is **VO_{2max} test module.** In order to determine training zones athlete's fitness level is required. One of the most popular method based on measuring volume of oxygen can be consumed by athletes while exercising. There are many different methods to estimate VO_{2max} value. They are suited for athletes in different age, sex and level of fitness. In SSAT many of them are described and short instruction for each test is attached [1, 4, 14].

Next block which is the element of desktop application is data processing block. The first element is module called **maximal heart rate during physical activity module**. The easiest and one of the best known method to determine maximal heart rate is the following formula:

$$HR_{max} = 220 - \text{age} \tag{1}$$

As it was proven this formula is proper only for male. For female obtained results for maximal heart rate determination are overestimated. To this end in [5] modified relationship for female is proposed:

$$HR_{max} = 206 - (0.88 \cdot \text{age}) \tag{2}$$

where age stands for athlete's age in years.

Another component of this block is module to **determine calories to be burnt**. Calculations are based on fitness goals (i.e. gaining or maintaining body weight) calories requirements and of nutrition habit of athlete.

Human energy requirements for males and female are different. For example male between 20 and 60 must intake approximately 2500 kcal per day but female only 2000. It is possible to determine energy requirements both for males and females taking into account physical activity level:

$$\mathrm{TDEE} = (370 + (21.6 \cdot \mathrm{bm})) \cdot \mathrm{al} \tag{3}$$

where *bm* stands for body mass without fat, *al* means activity level: sedentary (al = 1.2), light (al = 1.375), moderate (al = 1.55), high (al = 1.725) and very high (al = 1.9) [19].

In order to elicit a loss of weight between 0.5 to 0.9 kg per week approximately from 500 to 1000 kcal must be burnt each day [8]. Calories to be burnt are determine as difference between TDEE and 500 or 1000 kcal (it depends on training goals). It must be stress that intakes more or less calories in comparison to TDEE is taking into account in SSAT application.

The last component is **training zones determination.** Training zones are calculated based on determined $\mathrm{VO}_{2\mathrm{max}}$ value, age and weight of athlete and calories to be burnt. There are five training zones which allow us to obtain different results [18]. In order to determine proper training zone based on given data algorithm based on classification techniques are presented. Let us group $\mathrm{VO}_{2\mathrm{max}}$ value, age and weight of athlete and calories to be burnt in following feature vector:

$$x = \left[x^{(1)} \quad x^{(2)} \quad x^{(3)} \quad x^{(4)}\right]^T, \tag{4}$$

where $x^{(1)} - \mathrm{VO}_{2\max}$, $x^{(2)} - \mathrm{age}$, $x^{(3)} - \mathrm{calories}$, $x^{(4)} - \mathrm{weight}$.

Now, we can define set of classes to be classified. Because the task is to find proper training zones based on element of feature vector (4) proposed set of classes has five classes connected with five training zones i.e.:

$$\mathrm{J} = \{1,2,...,5\}. \tag{5}$$

In order to solve defined problem we have to proposed classification algorithm such as:

$$j = \Psi(x, S), \tag{6}$$

where $j \in \mathrm{J}$ and S is training set:

$$S = \{(x_1, j_1), (x_2, j_2), ..., (x_N, j_N)\}, \tag{7}$$

where N is length of training set. To solve defined problem in SSAT k-NN method is applied.

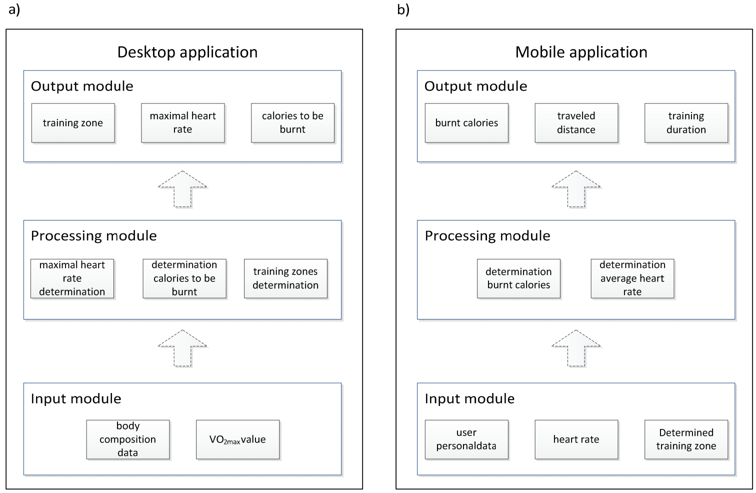


Fig. 2. SSAT application architecture: (a) desktop application, (b) mobile application

The last block is design to present results of data processing. To this end modules to recommend training zones, calories to be burnt and maximal heart rate is implemented.

On the right side of Fig. 2 architecture of mobile application is presented. Acquisition module is used to record data form heart rate, distance and speed monitor. Additionally personal athletes data and determined training zones must be given. Based on this data modules in processing block are used to determine:

a) average value of heart rate;
b) burnt calories during physical activities.

Average value of heart rate module determine mean value of heart rate in each minute. While **burnt calories during physical activities module** is designed to determine burnt calories while exercising. To this end following equations are applied:

$$\mathrm{kcal} = \frac{-59.39 + (-36.38 + 0.27 \cdot \mathrm{age} + 0.39 \cdot \mathrm{weight} + 0.4 \cdot \mathrm{vo2max} + 0.63 \cdot \mathrm{HR})}{4.18} \quad (8)$$

and

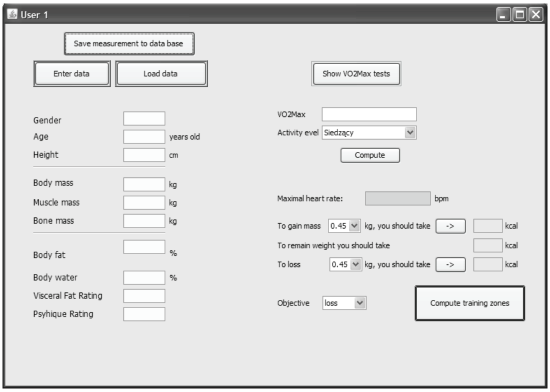
$$\mathrm{kcal} = \frac{-59.39 + (0.27 \cdot \mathrm{age} + 0.1 \cdot \mathrm{weight} + 0.38 \cdot \mathrm{vo2max} + 0.45 \cdot \mathrm{HR})}{4.18} \quad (9)$$

where *kcal* stands for kilogram calories, *age* is athletes age in years, *weight* in [kg], *vo2max* is maximal oxygen uptake and HR stands for average heart rate from proper training zones.

The last block is presentation module designed to display burnt calories while exercising, duration of training and traveled distance are implemented. All of them are very useful for athletes during physical activities.

3. TESTS AND EXPERIMENTS

SSAT system has been implemented and run in laboratory environment. In this section detailed presentation of SSAT is given and results of experiments are considered. Test platform are built by use of Zephyr HxM as heart rate, speed and distance monitor and Tanita BC-590 BT as body composition scale. Both application i.e. desktop and mobile are implemented in Java language. In Fig. 3 main window of desktop application is presented.



Activity evel Siedzący

To gain mass 0.45 kg, you should take

To loss 0.45 kg, you should take

Objective loss

Fig. 3. SSAT: main window

This part of application allows to enter personal data from database and to get data from body composition scale. To determine training zones results of fitness level test (i.e. value of VO_{2max}) and aim of the training (i.e. gaining, losing or maintaining weight to be selected) must be added to this form. Next step in exercise routine planning is connected with determination of training zones. In this step set of following action can be done:

- User can fix number of calories to be consumed at 7 days duration;
- User can choose training time: 30 min., 45 min., 60 min.

Next element of SSAT application is related to computation of training zones resting on data given by user. It helps to establish number of calories to be burnt in order to achieve training goals. Moreover, application recommend number of metabolic equivalences (MET) related to determined values of calories to be burnt. It is useful property and can be utilize in order to plan fitness training.

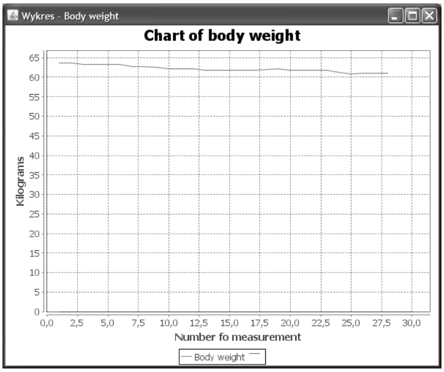


Fig. 4. SSAT: results presentation

The last element of desktop application refers to verification of training results. User can update information on burnt calories based on calculation obtains from mobile application. For comparison purpose it is possible to display on the same chart real and predicted values of burn calories. Moreover, as it is shown in Fig. 4 user can display measured values body weight, muscle mass, body fat and body water of the last 30 measurements.

4. FINAL REMARKS

In the work wearable sensor-based system for support athletes training is presented. Proposed solution based on wearable wireless sensors to record physiological and kinematic data. Transmitting sensed data is made through Bluetooth interface. The second application of the system is called desktop application and it is designed to manage training process. It helps to plan intensity and volume of the training.

Authors design and discussed architecture of the system and application in details. For application comprehensive analysis of functional and non-functional properties are given. Methods of data processing in desktop application are presented. Results of experimentation and system testing are given.

Further works for presented system will be focused upon algorithms for support exercising routine and athlete monitoring while physical activities. It supplies system with additional feedback information which helps to design personalized and user oriented system.

REFERENCES

[1] BURGER S.C., BERTRAM S.R., STEWART R.I., *Assessment of the 2.4 km run as a predictor of aerobic capacity*, South African Medical Journal, Vol. 78, No. 6, 1990, 327–329.

[2] BUTTUSSI F., CHITTARO L., *MOPET: A Context-Aware and User-Adaptive Wearable System for Fitness Training*, Artificial Intelligence in Medicine, Vol. 42, 2008, pp. 153–163.

[3] CHI-CHUN L.O. et al., *Ubiquitous Healthcare Service System with Context-awareness Capability: Design and Implementation*, Expert Systems with Applications, Vol. 38, Issue 4, 2011, pp. 4416–4436.

[4] GEORGE J.D., VEHRS P.R., ALLSEN P.E., FELLINGHAM G.W., FISHER A.G., *VO_{2max} estimation from a submaximal 1-mile track jog for fit college-age individuals*, Medicine & Science in Sports & Exercise, Vol. 25, No. 3, pp. 401–406.

[5] GULATI M., et al., *Heart rate response to exercise stress testing in asymptomatic women: the st. James women take heart project*, Circulation, Vol. 122, No. 1, 2010, pp. 130–137.

[6] HAN D, LEE M, PARK S., *THE-MUSS: Mobile u-health service system*, Journal Computer Methods and Programs in Biomedicine archive, Vol. 97, Issue 2, 2010, pp. 178–188.

[7] HERMENS H.J., VOLLENBROEK-HUTTEN M.M.R., *Towards remote monitoring and remotely supervised training*, J Electromyogr Kinesiol, Vol. 18, Number 6, 2008, pp. 908–919.

[8] JAKICIC J.M., CLARK K., *Appropriate Intervention Strategies for Weight Loss and Prevention of Weight Regain for Adults*, Official Journal of the American College of Sports Medicine, 2001, pp. 2145–2156.

[9] JOVANOV E., *A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation*, Journal of NeuroEngineering and Rehabilitation, Vol. 2, No. 1, 2005.

[10] KEYTEL L.R., GOEDECKE J.H., *Prediction of energy expenditure from heart rate monitoring during submaximal exercise,* Journal of Sports Sciences, 2005, pp. 289–297.

[11] LIM J.E. et al., *A context-aware fitness guide system for exercise optimization in U-health*, IEEE Trans Inf Technol Biomed, Vol. 13, Number 3, 2009, pp. 370–379.

[12] LORENZ A., OPPERMANN R., *Mobile health monitoring for the elderly: Designing for diversity, Pervasive and Mobile Computing*, Vol. 5, Issue 5, 2009, pp. 478–495.

[13] LORINCZ K. et al., *Mercury: A Wearable Sensor Network Platform for High-Fidelity Motion Analysis.*

[14] NOAKESA T.D., MYBURGHA K.H., SCHALL R., *Peak treadmill running velocity during the VO2 max test predicts running performance*, Journal of Sports Sciences, Vol. 8, Issue 1, 1990, pp. 35–45.

[15] PANTELOPOULOS A., BOURBAKIS N.G., *A Survey on Wearable Sensor-Based Systems for Health Monitoring and Prognosis*, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 2010, pp. 1–12.

[16] PREUSCH E. et al., *Mobile Motion Advisor — a feedback system for physical exercise in schools*, Procedia Engineering, Vol. 2, Issue 2, 2010, pp. 2741–2747.

[17] RODRIGUES J.J.P.C. et al., *Biofeedback data visualization for body sensor networks*, Journal of Network and Computer Applications, Vol. 34, Issue 1, 2011, pp. 151–158.

[18] THOMPSON W., GORDON N., *ACSM's Guidelines for Exercise Testing and Prescription*, Lippincott Williams & Wilkins, 2009.

[19] WIDRICK, J., *Treadmill validation of an over-ground walking test to predict peak oxygen consumption*, European Journal of Applied Physiology and Occupational Physiology, Vol. 64, Number 4, 1992, pp. 304–308.

[20] WHO, www.who.int

PART 4

SOFT COMPUTING
AND ITS APPLICATONS

minmax criterion,
robust optimization,
discrete optimization

Bogusz PRZYBYSŁAWSKI*, Adam KASPERSKI*

SOLVING MINMAX DISCRETE OPTIMIZATION PROBLEMS WITH IMPRECISE COSTS

In practical applications of operations research, we are often faced with the problem of uncertain data. This uncertainty is a basic structural feature of the technological and business environment and it must be considered as a part of the decision making process. In this work some basic discrete optimization problems such as the shortest path, minimum spanning tree and minimum assignment are discussed. The uncertainty is modeled by specifying a finite scenario set and the minmax criterion is adopted to choose a solution. All the considered problems are known to be NP-hard. The aim of this work is to test the efficiency of the mixed integer programming formulation for these problems. Namely, what amount of time is required to solve the problems when their size changes, the number of scenarios increases or the structure of the input graph changes. Two popular solvers, namely glpk and cplex are used for the tests.

1. INTRODUCTION

Decision making under uncertainty is an important area of management science. If one tries to describe a particular situation in an organization, some data often appear to be not precisely known. This uncertainty is a basic feature of the nature, so we should accept it and make it part of decision making process [5].

One of the most popular approaches to modeling the uncertainty is a *scenario representation* of uncertain data [5]. Each particular realization of the problem parameters is called a *scenario*. The probability of occurrence of a scenario is only known to

* Institute of Industrial Engineering and Management, Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370, Wrocław.

be positive but its exact value may be unknown. Under uncertainty, the number of scenarios is greater than one and there is a set Γ of all the scenarios, called a *scenario set*. There are two methods of describing the scenario set. In the *interval case,* each parameter can take any value between an upper and a lower bound and Γ is the Cartesian product of all these intervals. In the *discrete scenario case*, the set $\Gamma = \{S_1, S_2, ..., S_K\}$ is defined by explicitly listing all the possible scenarios. In this work only the second uncertainty representation is considered.

In many applications we have to solve a discrete optimization problem (see, e.g.,[2]). In this class of problems we are given a finite set F of feasible solutions and we seek a solution $X \in F$ which minimizes a cost $f(X)$. In an uncertain situation, the solution cost depends on scenario $S \in \Gamma$ and we denote it as $f(X,S)$. In the presence of more than one scenario, the *minmax criterion* for choosing a solution can be used [5]. Under this criterion we seek a solution which minimizes the maximum cost over all scenarios and the corresponding optimization problem is defined as follows:

$$\min_{X \in F} \max_{s \in \Gamma} f(X,S) \tag{2}$$

The problem (2) is equivalent to the following mathematical programming problem:

$$\begin{array}{ll} \text{Minimize} & p, \\ \text{Subject to} & \forall S \in \Gamma: \quad f(X,S) \le p \\ & \qquad\qquad X \in F \end{array}$$

Problem (2) belongs to the class of robust optimization problems (see, e.g.,[5]), where decision makers minimize the cost of a decision in the worst case. The robust approach is very popular and a description of the recent results in this area can be found in [1]. The aim of this work is to recall the formulation of the mixed integer programming (MIP) models for some basic discrete optimization problems and test the efficiency of these models by using two popular solvers *glpk* [8] and *cplex* [7]. We will discuss the models, where F is the set of all paths, spanning trees or assignments in a given graph. All the minmax problems considered in this work are NP-hard [5], so there is no hope to design exact and efficient algorithms for them. Therefore, the MIP formulation can be inefficient for large instances and the aim of this work is to check how large problems can be solved by using this formulation.

This work is organized as follows. Section 2 recalls the formal models of shortest path, spanning tree, and assignment problems, with the minmax criterion. Next sections describe the background of research and contain the results of the tests. The last section contains the conclusions.

2. FORMAL MODELS

2.1. MINIMUM ASSIGNMENT PROBLEM

The assignment problem arises in many situations (see, e.g., [2,3]), when some pairs of elements such as machine-human represent a solution. The data of the assignment problem consist of two equally sized sets M and N, $|N|=|M|=n$, and, in the deterministic case, a cost c_{ij} is associated with each pair (i, j) for $i \in M, j \in N$. We wish to pair, at the minimum possible cost, each object in N with exactly one object in M. Now we will use c_{ijk} to denote the cost of the pairing (i, j) under scenario S_k. The mathematical model for the minmax assignment problem uses a decision binary variable x_{ij} which takes the value of 1 if object i is assigned to object j, and 0 otherwise. The model has the following form:

$$\text{Minimize} \quad p$$

$$\text{Subject to} \quad \forall S_k \in \Gamma: \sum_{i\in M}\sum_{j\in N} c_{ijk}x_{ij} \le p$$

$$\forall i \in M: \sum_{j\in N} x_{ij} = 1$$

$$\forall j \in N: \sum_{i\in M} x_{ij} = 1$$

$$\forall i \in M, j \in N: x_{ij} \in \{0,1\}$$

2.2. SHORTEST PATH PROBLEM

Let $G=(N,A)$ be a directed network defined by the set N of n nodes and the set A of m directed arcs with an associated cost (length) c_{ij} for each arc $(i, j) \in A$. We wish to find a path of the minimum cost (length) from a source node s to a sink node t. We will use c_{ijk} to denote the cost of the arc (i, j) under scenario S_k. The formal model for the shortest path problem with the minmax criterion is the following:

$$\text{Minimize} \quad p,$$

$$\begin{aligned}
\text{Subject to} \quad & \forall S_k \in \Gamma: && \sum_{(i,j)\in A} c_{ijk} x_{ij} \le p\,, \\
& && \sum_{(s,j)\in A} x_{sj} - \sum_{(j,s)\in A} x_{js} = 1 \\
& && \sum_{(t,j)\in A} x_{tj} - \sum_{(j,t)\in A} x_{jt} = -1 \\
& \forall i \in N \setminus \{s,t\}: && \sum_{(i,j)\in A} x_{ij} - \sum_{(j,i)\in A} x_{ji} = 0 \\
& \forall (i,j) \in A: && x_{ij} \in \{0,1\}
\end{aligned}$$

2.3. MINIMUM SPANNING TREE PROBLEM

A spanning tree of a given undirected graph $G = (V, E)$ is an acyclic subgraph of G that contains all the vertices of G. It is important to notice that every spanning tree of n vertices has exactly n-1 arcs. The cost of a spanning tree is the sum of the costs of all its arcs. In the deterministic case, the goal is to identify a spanning tree with the minimum cost. We now use the model for the problem, which is based on the one proposed in [6]. Let A be the set of all arcs defined by $A = \langle (i,j) \in V \times V : \{i,j\} \in E \rangle$ and f_{ij} denote the flow on an the arc (i, j). Let c_{ek} by the cost of edge $e \in E$ under scenario S_k. The minmax model for the considered problem is the following:

$$\text{Minimize} \quad p,$$

$$\begin{aligned}
\text{Subject to} \quad & \forall S_k \in \Gamma: && \sum_{e\in E} c_{ek} x_e \le p \\
& && \sum_{(1,j)\in A} f_{1j} - \sum_{(j,1)\in A} f_{j1} = n-1 \\
& \forall i \in N \setminus \{1\}: && \sum_{(i,j)\in A} f_{ij} - \sum_{(j,i)\in A} f_{ji} = -1
\end{aligned}$$

$$\forall\{i,j\}\in E: \quad f_{ij} \le (n-1)x_{ij}$$

$$\forall\{i,j\}\in E: \quad f_{ji} \le (n-1)x_{ij}$$

$$\sum_{e\in E} x_e = n-1$$

$$\forall e\in E: \quad x_e \in \{0,1\}$$

$$\forall (i,j)\in A: \quad f_{ij} \ge 0$$

3. COMPUTATIONAL RESULTS

In this section we present the results of some computational tests. All the tests were performed on the computer equipped with a Intel Core i5 M430 2,27 GHz processor with 4 GB RAM and a 64-bit operational system. The tests were performed by using the programming environment – IBM OPL IDE 6,3 with CPLEX 12.1 [7] and the open source LP/MILP IDE – Gusek based on the text editor – SciTE 1,76 and a GLPK solver [8]. The aim of the tests was to identify the parameters of the models which have the greatest influence on the efficiency of computations and determine the amount of time which is required to get an optimal solution.

3.2. THE MINIMUM ASSIGNMENT PROBLEM

We used the model from Section 2.1 to obtain the results which are presented in Table 1. If any computation lasted more than 20 minutes, then it was stopped. The number of scenarios varies from 2 to 10 and the value of *n* varies from 10 to 200. As we can see, the computational time increases with the number of assignments and the number of scenarios. The observed increase was, however greater for the number of scenarios. For larger problems, only the problems with a few scenarios can be solved efficiently. If one tries to compare the two solvers based on these tests, it is easy to notice that *cplex* is more efficient.

Table 1. The computational times in seconds for the assignment problem by using *cplex* (*glpk*). The empty place means that the computation time exceeded 20 minutes

	The number of scenarios *K*								
n	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
10	0.0 (0.0)	0.0 (0.2)	0.0 (0.6)	0.1 (1.3)	0.1 (1.7)	0.1 (2.9)	0.1 (6.9)	0.1 (15.0)	0.1 (17.7)
20	0.0 (1.6)	0.1 (7.9)	0.4 (41.8)	0.2 (121.6)	0.65	2.0	2.1	8.1	7.8
30	0.1 (4.1)	0.2. (11.6)	0.2 (60.8)	0.8	7.1	18.4	12.9	90.3	199.6
40	0.3 (41.3)	0.3 (32.9)	1.5	3.8	9.21	62.1	521.5	537.2	
50	0.3 (44.8)	0.4 (118.1)	1.6	10.2	53.8	96.4			
60	0.7 (50.8)	0.8	3.5	47.7	134.8				
80	1.1 (275.8)	0.9	5.2	26.8	249.1				
100	2.2 (379.1)	2.7	27.0	76.8					
120	4.2	4.7	45.2						
140	4.9	6.4	60.7						
160	10.1	14.2	106.7						
180	12.5	9.1							
200	12.6	13.8							

3.3. THE SHORTEST PATH PROBLEM

In order to solve the model constructed in Section 2.2 we defined a class of layered graphs shown in Fig. 1. Each graph consists of a number of vertices *n*, a number of layers *m* and the probability of connection between two vertices *p*. We assume that the connections with the first/last layer from a source node and a sink node always exist. The number of vertices in the last layer is the sum of the number of vertices in each layer and the rest from division between the number of vertices and the number of |layers.

In the first experiment we verified the question how does the size of the input data influence on the computational time. A network with 100 vertices and 10 layers was

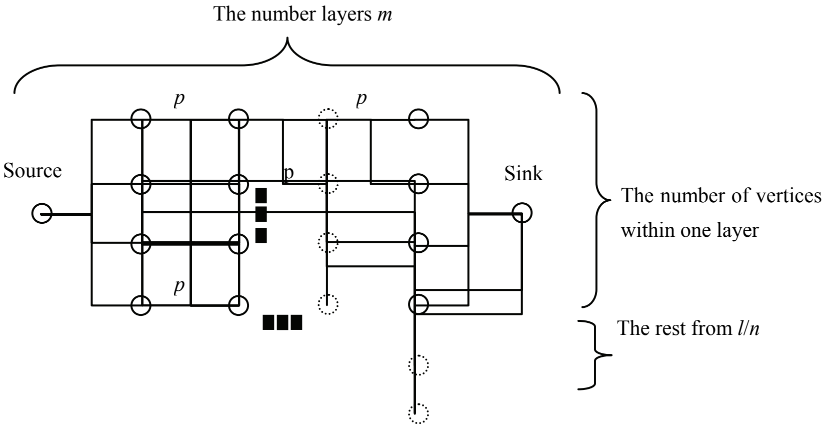


Fig 1. A layered graph

used to obtain the results, which are shown in Table 2. Three different values of the parameter p were chosen. As we can see the computational time increases with the number of scenarios. Although the number of vertices and layers remain the same, the number of connections has great impact on the computation times. The greatest computational times were reported for $p = 1$.

Table 2. The computational times in seconds for the shortest path problem by using *cplex (glpk)*

	The number of scenarios *K*									
p	**2**	**4**	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**
1	0.2 (1.4)	0.3 (6.8)	1.9 (41.7)	3.2 (522.2)	5.1 (596.3)	26.3	27.9	59.7	70.4	168.1
0,5	0.7 (1.4)	1.1 (6.4)	0.9 (19.9)	0.7 (30.3)	3.9 (211.4)	4.6	4.8	4.7	6.5	13.1
0,2	0.0 (0.0)	0.1 (0.0)	0.0 (0.4)	0.1 (0.8)	0.1 (1.0)	0.7	0.2	1.5	0.8	0.5

In the next experiment we explored 54 different graphs with the number of vertices varying from 100 to 800. Each graph had a different number of layers *m=2, 3, 4, 5, 6, 10*. The obtained results are presented in Table 3. According to final results, one can say that the number of layers had no impact on the computational times, which is interesting due to the fact that when this number increases the number of connections decreases.

Table 3. The computational times in seconds for the shortest path problem by using *cplex (glpk)*

	M					
n	**2**	**3**	**4**	**5**	**6**	**10**
100	2.3 (7.5)	0.2 (18.6)	0.3 (0.3)	0.3 (8.3)	2.5 (5.0)	0.2 (3.0)
200	2.2 (5.4)	1.1 (16.1)	1.0 (1.9)	0.8 (83.9)	1.1 (7.1)	0.7 (16.7)
300	6.0 (34.5)	2.8 (70.0)	2.9 (101.7)	2.1 (144.3)	2.0 (46.1)	1.2 (249.1)
400	7.9 (48.2)	4.2 (76.4)	3.6 (22.5)	6.5 (172.9)	5.3 (530.6)	4.1 (2255.7)
500	9.9	7.3	7.3	11.3	10.0	7.5
600	10.8	13.7	13.0	9.4	10.6	12.2
700	11.2	16.8	26.6	26.8	18.0	13.1
800	15.0	25.8	23.9	37.8	31.6	22.6

3.3. THE MINIMUM SPANNING TREE PROBLEM

In this section we investigate the model shown in Section 2.3. This model turned out to be more difficult to solve than the models for the problems discussed in the previous sections. Hence, we used only *cplex* solver, because the performance of *glpk* was poor. In the first experiment the network with 30 vertices and 3 layers was explored (see Fig. 1). The number of scenarios varies from 2 to 6. For each number of scenarios we generated and solved 10 instances of the problem. The obtained results are shown in Table 4. We present the minimum, maximum and average computational times in seconds. The problem seems to be complex due to the enormous deviations from the average in every single test. The average values show that the number of scenarios has great influence on the final computational time.

Table 4. The computation times in seconds for a different number of scenarios.
Min – minimum computation time, max - maximum computation time,
avg – average time for 10 instances by using *cplex*

		min	Max	Avg
K	**2**	0.2	3.6	1.0
	3	1.0	9.1	2.8
	4	1.5	211.4	26.0
	5	0.4	522.7	96.6
	6	5.2	442.7	130.0

A network with 2 layers and 2 scenarios was chosen to the next experiment. The number of vertices varies from 10 to 80. For each number of vertices we generated and solved 10 instances of the problem. If any computation lasted more than 20 minutes, then it was stopped. The obtained results are shown in Table 5. The average values show that the number of vertices has great impact on the final computational time.

Table 5. The computation times in seconds for a different number of vertices. Min – minimum computation time, max - maximum computation time, avg – average time for 10 instances by using *cplex*

		min	max	avg
Number of vertices	10	0,07	0,63	0,28
	20	0,25	0,97	0,52
	30	0,39	1,35	0,96
	40	1,01	5,08	2,99
	50	1,38	86,13	30,78
	60	120,07	>20min	353,27
	70	16,05	>20min	388,59
	80	746,27	>20min	1282,7

3.5. CONCLUSIONS

In this work we have tested the minmax versions of some classical optimization problems such as the minimum assignment problem, minimum shortest path, minimum spanning tree for the discrete scenario case. We performed the computational tests for a special class of layered graphs. The tests included the factors connected to the structure of generated graph. We used two different solvers, namely *cplex* and *glpk,* and compared them. In the discussed cases the presented approach leads to problems which are computationally complex. Having a particular problem, one can try to apply a mixed integer programming formulation to obtain a solution. If it is impossible to achieve in reasonable time, then some approximate algorithms (see, e.g.,[4]) can be applied.

REFERENCES

[1] AISSI H., BAZGAN C., VANDERPOOTEN D., *Min-max and min-max regret versions of combinatorial optimization problems: A survey*, European Journal of Operational Research 197, 2009, 427–438.

[2] AHUJA R., MAGNANTI T., ORLIN J., *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, New Jersey, 1993.

[3] HILLIER F., LIEBERMAN G., *Introduction to Operations Research*, McGraw-Hill Book Co., Fifth edition, Singapore 1990.

[4] KASPERSKI A., *Making robust decisions in discrete optimization problems as a game against nature*, AUCO Czech Economic Review 2(3), 2009, 237–250.

[5] KOUVELIS P., YU G., *Robust Discrete Optimization and Its Applications*, Kluwer Academic Publishers, 1997.

[6] YAMAN H., KARASAN O., PINAR M., *The robust spanning tree problem with interval data*, Operations research letters 29, 2001, 31–40.

[7] http://www-01.ibm.com/software/integration/optimization/cplex-cp-optimizer/; 28.03.2011 r.

[8] http://gusek.sourceforge.net/gusek.html; 28.03.2011 r.

neural networks, discrete signals,
digital signal processing, signal filtering

Arkadiusz LEWICKI*

USING INTELLIGENT DYNAMIC FILTER IN DIGITAL SIGNAL PROCESSING

A very important issue in various fields of engineering problem is recognition the actual state of the test object and the related dynamic processes. Recognition of this condition is based on information collected and additional information. However, the accumulated information is burdened with some errors. Therefore, in this work is presented an attempt to build the neural filter, allowing the identification of a discrete signal, so that it can be assigned by the neural network to a class of signals. The problem of filtering of such signals is not new, but the methods that were used for this purpose so far are mainly based on a simplified and thus quite distant from the reality of the mathematical theory of signals, while the proposed approach involves the use of neural networks, which have the ability to adapt and self-organization during the workout. The results of the evaluation of the solution presented in this publication shows that these networks capable of very high accuracy and probability to identify noisy and distorted signal.

1. INTRODUCTION

The main important advantage of artificial neural network that differs from programs performing deterministic processing of information is their ability to generalize knowledge to new data that were previously unknown, and therefore they were not presented during learning. The advantages of this are mainly due to the ability of neural networks to approximate values of functions of several variables, in contrast to the interpolation possible to obtain the algorithmic processing. Thus, for example, expert systems generally require the assembly and the current access to all knowledge on issues about which they are to rule [1, 2, 3, 4]. Neural networks require only a one-time learning, but they exhibit a high tolerance to discontinuity,

* University of Information Technology and Management in Rzeszow.
e-mail: alewicki@wsiz.rzeszow.pl

random disturbances or even gaps in the training set [4, 5, 6, 7]. This allows you to use them where they cannot solve the problem in any other, efficient manner Due to the specific properties of artificial neural networks, where information processing is incomplete or distorted, after its entry on the same signals only slightly distorted, the output should be set to similar values. This method of detection signals, which rely on assigning a signal to a certain category is called filtering and neural network, which meets this target – neural filter [8, 9, 10].

To be evaluated and the description of filters that are considered by the presented experiments should be done with respect to some discrete dynamical system. Such a system can be described by the following equations:

$$x(k+1) = f\,(x(k), u(k), k) + w(k), \tag{1}$$

$$y(k+1) = h(x\,(k+1), k+1) + v(\mathrm{k}+1), \tag{2}$$

where:

$f()$, $h()$ – nonlinear vector functions,
k – discrete time,
$x(k)$ – object state vector,
$y(k)$ – output vector object

The solution to the equations of the system described above is a linear Kalman [11] filter, whose character can be represented by the following equations:

$$x(k+1 \mid k) = A(\mathrm{k})\, x\ \ k \mid k) + B(k)\, u(k), \tag{3}$$

$$P(k+1 \mid k) = A(k)\, P(k \mid k)\, A^T(k) + Q(k), \tag{4}$$

$$V(k+1) = H(k+1)\, P(k+1 \mid k)\, H^T(k+1) + R(k+1), \tag{5}$$

$$K(k+1) = P(k+1 \mid k)\, H^T(k+1) V^{1}(k+1), \tag{6}$$

$$v(k+1\ ) = y(k+1) - H(k+1)\, x(k+1 \mid k), \tag{7}$$

$$x(k+1 \mid k+1) = x(k+1 \mid k) + K(k+1)\, \mathrm{v}(k+1), \tag{8}$$

$$P(k+1 \mid k+1) = P(k+1 \mid k) - K(k+1)\, H(k+1)\, P(k+1 \mid k), \tag{9}$$

where:

$x(\ i \mid j\ )$ – value of x in the i-th time, determined based on data from the time of j-th,
P – filtering error covariance matrix,
K –strengthening matrix

Defined in this way the filter is an algorithm computing, the convergence, understood as the convergence estimate x (k | k) to the actual value is determined by accurate mathematical model of the test object and the knowledge of the characteristics of the disturbances occurring in this model. Estimation of signals by the described filter is implemented through the appointment of successive approximations of the signal on the basis of previous states and previous and current input values.

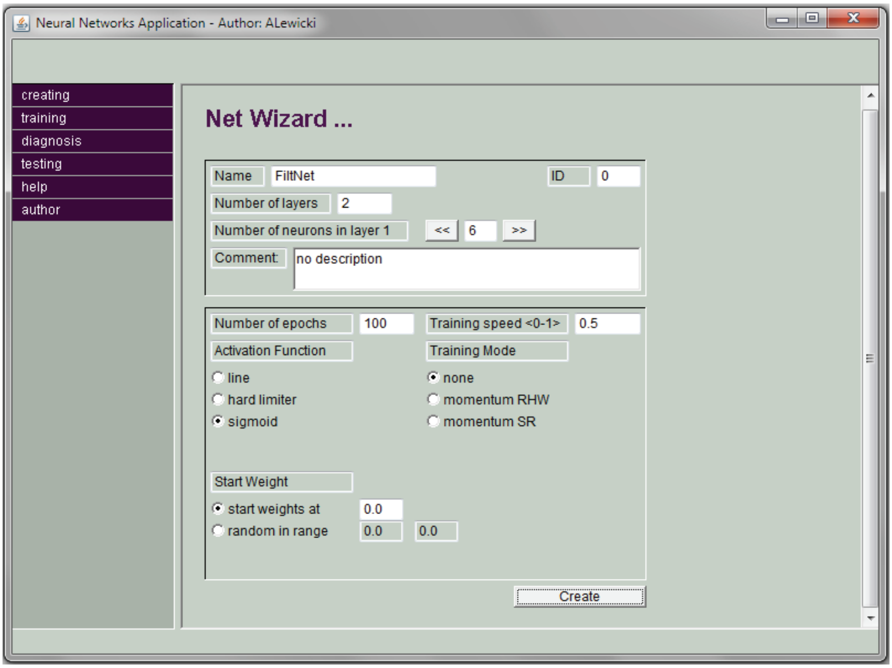
2. THE APPROACH

Adopted a set of trains in the experiments conducted by the author consisted of six pairs of values, where each pair corresponds to a particular signal. For example, if we give input signal sampled sin(x), then the first network output was set to 1, while the remaining to 0. In the case of the network if the input signal is given sampled sine of 2x, the second output of the network has been set at one.

Neural networks are a relatively young field of science and therefore there is no mathematical relations, which can unambiguously determine the optimum network architecture. Therefore, the number of layers and the number of neurons in each layer was determined experimentally by trial and error. The model adopted had the following parameters:

- number of layers: 2
- the number of neurons in each layer: 12, 6
- number of epochs: 100

To assess the quality of filtration of discrete signals (Fig. 2) using neural networks in Java author has created a practical application, whose main objective was the implementation presented in the work linear neural filter. Panel application, which is responsible for creating neural networks and set all its parameters, is presented in Figure 1. It can be seen on all the text fields for the initialization parameters. In the process of designing an application uses both the right approach for the purposes of the basic model and functional model. So that the application has a very high degree of scalability. If the results are unsatisfactory filtration, we may at any time to return to the Network Wizard and verify its performance without changing the training set. All stages of development are available through the menu appearing on the left side of the window. In connection with the specific purpose of neural network implemented when creating the structure of neurons in the output class should be set on 6, because only that number of classes of signals are defined in the training set. This collection can be always modified.



Net Wizard ...

Name: FiltNet
ID: 0
Number of layers: 2
Number of neurons in layer 1: << 6 >>
Comment: no description

Number of epochs: 100
Training speed <0-1>: 0.5

Activation Function
○ line
○ hard limiter
● sigmoid

Training Mode
● none
○ momentum RHW
○ momentum SR

Start Weight
● start weights at 0.0
○ random in range 0.0 0.0

Fig. 1. The panel implementation of the new neural network
(double-layer networking with the number of epochs equal to 100 and learning rate 0.5)

In the process of network learning parameters visible in Figure 1 was obtained characteristics of the error, which is shown in Figure 3. It shows that with the successive epochs of network training error, and thus the difference between the baseline and the expected decreases to a value close to zero. Network results obtained by such learning may be subjected to further testing and diagnosis.

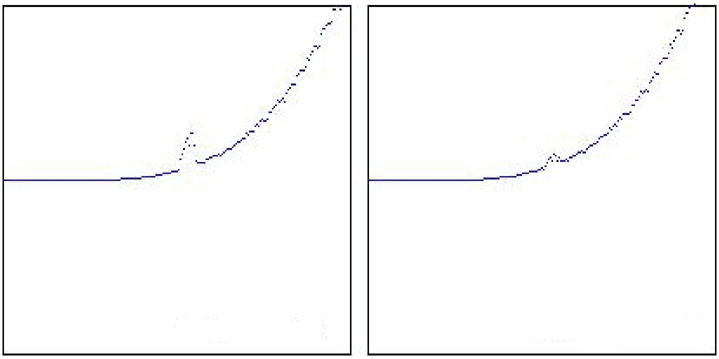
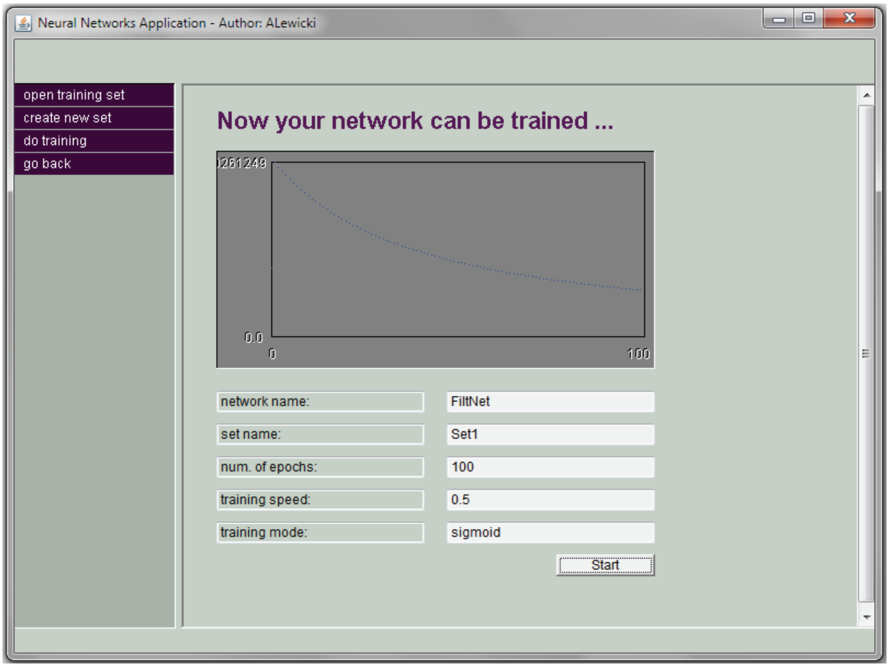


Fig. 2. Computer simulation of a linear filter



Now your network can be trained ...

network name:	FiltNet
set name:	Set1
num. of epochs:	100
training speed:	0.5
training mode:	sigmoid

Fig. 3. Neural network learning process

One of the key elements of the evaluation process is carried out in the process of diagnosis. It involves the administration of the network inputs and outputs of the same patterns that were used to her teaching. Obtain relevant results are the basis for further testing of the network. The network can not be allowed to continue to use (filter) if it does not work correctly with the pre-established vectors of the reference. In the case of the proposed approach, most of the experiments carried out correctly recognize a network of signals provided on input. It should be noted, however, that the selection of appropriate parameters, and construct an ideal, which is fully complete set of patterns is extremely difficult to aim and requires no small experience and time-intensive.

The process of testing the network, associated with the verification, if the network maintains its properties when administered on its entry into the incomplete or distorted is shown in Figure 4.

Made by the author of this publication experimental analysis shows that filtration produces satisfactory results. Only when we superimpose the large signal distortion equal to about 1/3 the amplitude of the input signal is not recognized. The following are illustrations 5 and 6, which show how neural filter filters the distorted signals. The first of these represents an unsuccessful attempt to identify the signal sin (½*x*). In this case, we should consider the selection of other, more relevant parameters, or com-

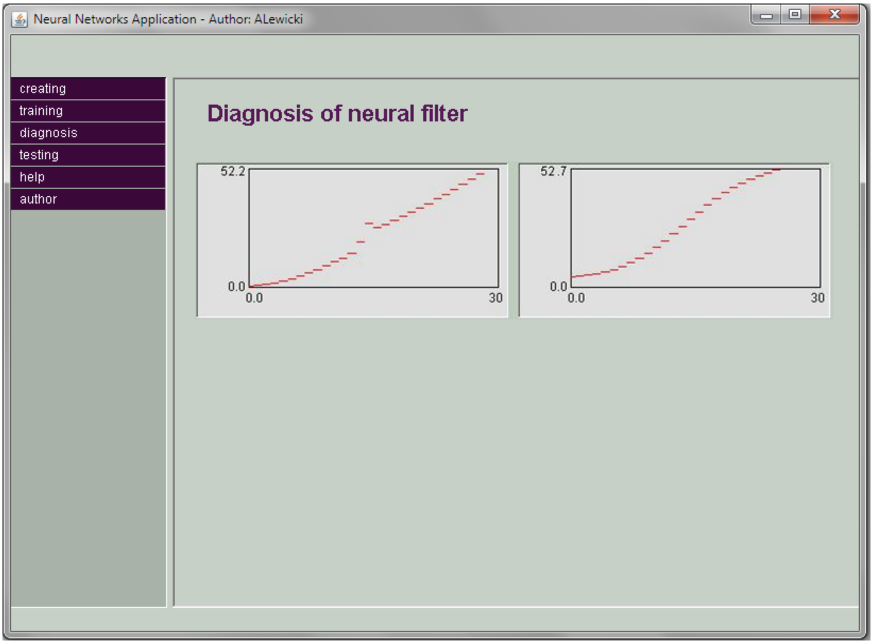
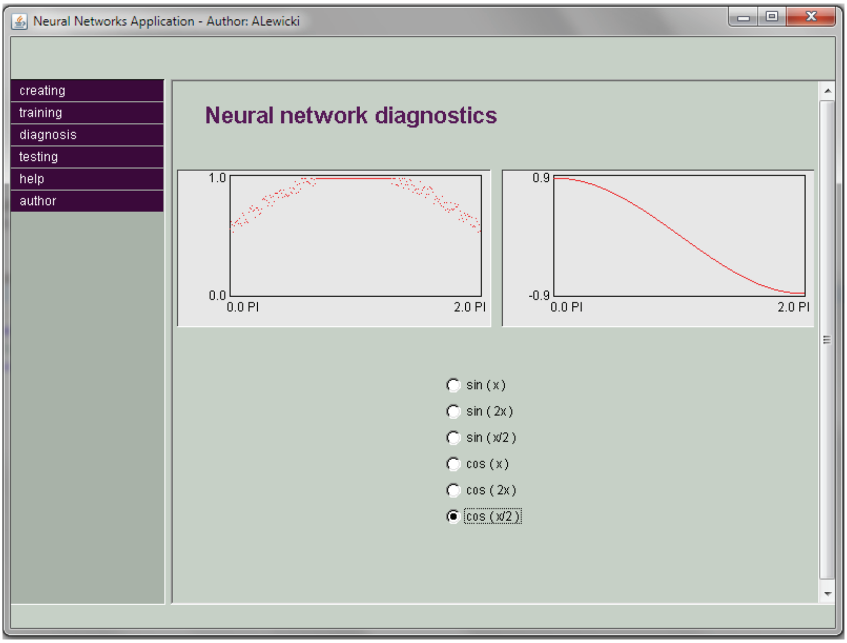

Neural Networks Application - Author: ALewicki
creating
training
diagnosis
testing
help
author
Diagnosis of neural filter
52.2
0.0
0.0
30
52.7
0.0
0.0
30

Fig. 4. Neural network diagnostics panel



○ sin (x)
○ sin (2x)
○ sin (x/2)
○ cos (x)
○ cos (2x)
● cos (x/2)

Fig. 5. Incorrectly recognized input signal sin (1/2 x) for noise level 0.5

pletely change the structure or size of the network's training set. The second figure illustrates a successful attempt to filter the (identification) signal sin(2*x*). We can see how distorted, sampled waveform is smoothed by a network.

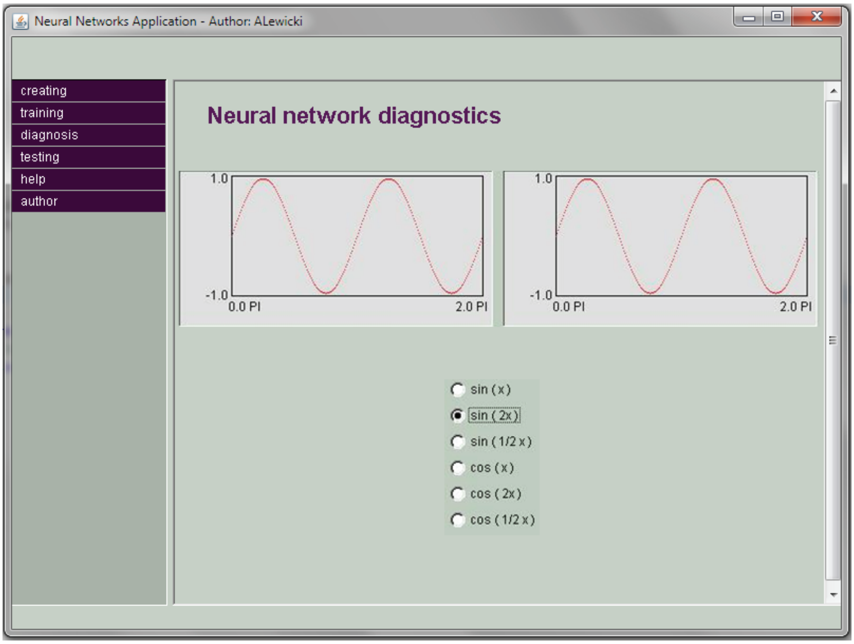


Fig. 6. Correctly identified the input signal cos (2x) for the noise level 0.5

3. EXPERIMENTS

In order to determine the functional quality of the proposed neural filter was made a series of different experiences for different network parameters. All results are summarized in three tables. Each table corresponds to the learning network with different learning algorithm to improve convergence. Tested neural network was constructed of two layers. The first one contains 12 neurons and one sixth. Number of output layer neurons is equal to the number of signals that the network has identified. In this experiment assumed the number of learning epochs equal to 100 and noise level of the amplitude of input signals, which was set to 0.2. In the table “x” indicates that the signal was not recognized. Button “OK.” shows the correct identification of the same course. The following summary of the results are only a small part of the whole field of different combinations. Therefore, in future area of input sets will be expanded to perform a clear assessment of the functionality and performance of the filter presented independent of the type of data.

Table 1. Results of learning without momentum

Speed learning	Error	sin(x)	sin($2x$)	sin($½x$)	cos(x)	cos($2x$)	cos($½x$)
0.001	4.684892	**x**	**x**	**x**	**x**	**x**	**x**
0.003	4.0485783	**x**	**x**	**x**	**x**	**x**	**x**
0.005	3.37754	**x**	**x**	**x**	**x**	**x**	**x**
0.007	2.9859042	**x**	**x**	**x**	**x**	**x**	**x**
0.009	2.5055244	**x**	**x**	**x**	**x**	**x**	**x**
0.01	2.3927903	**x**	**x**	**x**	**x**	**x**	**x**
0.03	1.5676018	**x**	**x**	**x**	**x**	**x**	**x**
0.05	1.634029	**x**	**x**	**x**	**x**	**x**	**x**
0.07	1.6036175	**x**	**x**	**x**	**x**	**x**	**x**
0.09	1.4303061	**x**	**x**	**x**	**x**	**x**	**x**
0.1	1.2519013	**x**	**x**	**x**	**x**	**x**	**x**
0.3	0.66513526	**x**	**OK**	**x**	**x**	**OK**	**x**
0.5	0.1955844	**OK**	**OK**	**OK**	**x**	**OK**	**x**
0.7	0.11131621	**OK**	**OK**	**OK**	**x**	**OK**	**x**
0.9	0.06272577	**OK**	**OK**	**x**	**x**	**OK**	**x**

Table 2. Network learning results from the momentum of RHW (Rumelhart, Hinton, Wiliams)

Speed learning	Error	sin(x)	sin($2x$)	sin($½x$)	cos(x)	cos($2x$)	cos($½x$)
0.001	4.5155964	**x**	**x**	**x**	**x**	**x**	**x**
0.003	3.7249584	**x**	**x**	**x**	**x**	**x**	**x**
0.005	3.1264515	**x**	**x**	**x**	**x**	**x**	**x**
0.007	2.474944	**x**	**x**	**x**	**x**	**x**	**x**
0.009	2.2009308	**x**	**x**	**x**	**x**	**x**	**x**
0.01	2.093015	**x**	**x**	**x**	**x**	**x**	**x**
0.03	1.5740893	**x**	**x**	**x**	**x**	**x**	**x**
0.05	1.611268	**x**	**x**	**x**	**x**	**x**	**x**
0.07	1.4975618	**x**	**x**	**x**	**x**	**x**	**x**
0.09	1.0795106	**x**	**x**	**x**	**x**	**x**	**x**
0.1	1.2386699	**x**	**x**	**x**	**x**	**x**	**x**
0.3	0.37836367	**OK**	**OK**	**OK**	**x**	**OK**	**x**
0.5	0.13880399	**OK**	**OK**	**OK**	**x**	**OK**	**x**
0.7	0.0608855	**OK**	**OK**	**x**	**x**	**OK**	**x**
0.9	0.04913237	**OK**	**OK**	**x**	**x**	**OK**	**x**

Table 3. Network learning results from the momentum of SR (Sejnowski, Rosenberg)

Speed learning	Error	sin(*x*)	sin(*2x*)	sin(½*x*)	cos(*x*)	cos(2*x*)	cos(½*x*)
0.001	4.842132	**x**	**x**	**x**	**x**	**x**	**x**
0.003	3.75667	**x**	**x**	**x**	**x**	**x**	**x**
0.005	2.67768	**x**	**x**	**x**	**x**	**x**	**x**
0.007	1.70070	**x**	**x**	**x**	**x**	**x**	**x**
0.009	1.452459	**x**	**x**	**x**	**x**	**x**	**x**
0.01	0.98926	**x**	**x**	**x**	**x**	**x**	**x**
0.03	0.4723	**x**	**x**	**x**	**x**	**x**	**x**
0.05	0.114842	**x**	**x**	**x**	**x**	**x**	**x**
0.07	0.0785516	**x**	**x**	**x**	**x**	**x**	**x**
0.09	0.0466969	**x**	**x**	**x**	**x**	**x**	**x**
0.1	0.0262080	**x**	**x**	**x**	**x**	**x**	**x**
0.3	0.0166733	**OK**	**OK**	**OK**	**x**	**OK**	**x**
0.5	0.0157773	**OK**	**OK**	**OK**	**OK**	**OK**	**x**
0.7	0.0106855	**OK**	**OK**	**X**	**OK**	**OK**	**OK**
0.9	0.0099274	**OK**	**OK**	**X**	**x**	**OK**	**x**

4. CONCLUSION

Data from different sources are often disrupted. Classical methods allow us to remove noise elimination of a random noise, but do not lead to the elimination of systematic distortions. Meanwhile, artificial neural network can learn, for example identify a number of reference objects. These patterns may be fragments of the series or images. If a variation of one of these patterns, disturbed by the noise will be given to enter the network, which has been properly taught, then this network will be able to recreate the original pattern, which has learned. Increasingly, therefore, the ability to use neural networks as data filters. This approach was also adopted in this work.

After analyzing all the results obtained we can conclude that the error decreases fastest network learning algorithm using momentum SR. In each of the three cases studied, we can see that there is a learning rate value for which the neural network works best. Below and above this limit results that we expect are less common. Both for learning without improving the convergence of the algorithm as well as with algorithms and SR RHW best results were obtained for the speed of learning at the level of 0.5.

REFERENCES

[1] GALUSHKIN A., *Neural networks theory*, Springer Verlag, 2007.

[2] OSSOWSKI S., *Sieci neuronowe do przetwarzania informacji*, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa 2000.

[3] OSSOWSKI S., *Sieci neuronowe do w ujęciu algorytmicznym*, WNT, Warszawa 1996.

[4] JAIN L.C and FANELLI A.M., *Recent Advances in Artificial Neural Networks: Design and Applications*, 2000.

[5] DUDEK-DYDUCH E., TADEUSIEWICZ R., HORZYK A., *Neural network adaptation process effectiveness dependent of constant training data availability,* Neurocomputing, 2009.

[6] TADEUSIEWICZ R., GĄCIARZ T., BOROWIK B., LEPER B., *Elementary introduction to neural networks technology with example programs*, Moscow, 2011.

[7] KORBICZ J, OBUCHOWICZ A., UCIŃSKI D., *Sztuczne sieci neuronowe*, Akademicka Oficyna Wydawnicza, Warszawa 1994.

[8] MILIC L., Multirate *Filtering for Digital Signal Processing: MATLAB Applications*, Informatic Science Reference, 2008.

[9] MARVEN C, EWERS G., Zarys cyfrowego przetwarzania sygnałów, WKŁ, Warszawa 1999.

[10] KACZOREK T., *Teoria sterowania i systemów*, PWN, Warszawa 1993.

[11] LYONS R. G., *Wprowadzenie do cyfrowego przetwarzania sygnałów*, WKŁ, Warszawa 2000.

Multilayer Perceptron, neural network,
SIMD architecture, systolic array

Jacek MAZURKIEWICZ*

MULTILAYER PERCEPTRON REALIZED USING SYSTOLIC ARRAY APPROACH

The chapter is a proposal related to partial parallel realisation of retrieving phase of Multilayer Perceptron algorithm. The method is based on pipelined systolic arrays – SIMD architecture. The discussion is realised based on operations which create the following steps of the algorithm. The efficiency of proposed approach is discussed based on implementation quality criteria for systolic arrays. The results of discussion show that it is possible to create the architecture which provides massive parallelism and reprogrammability.

1. INTRODUCTION

The work is related to partial parallel realisation of retrieving phase of Multilayer Perceptron algorithm. The method proposed is based on pipelined systolic arrays. The described methodology can be used as the theoretical basis for hardware of software simulators of Multilayer Perceptron [2]. The discussion is based on the assumptions:

- the outcome of algorithms realised true to proposed methodology is exactly the same like the outcome of classical Multilayer Perceptron algorithm,
- three-layer Multilayer Perceptron is taking into account, the approach can be easily adopted to more sophisticated Multilayer Perceptron networks,
- the systolic structure is realised using only digital elements, input and output data are represented in proper binary code,
- the number of Multilayer Perceptron neurons is unrestricted, but the maximum number of elementary processors can be limited.

* Wroclaw University of Technology, Institute of Computer Engineering, Control and Robotics, ul. Janiszewskiego 11/17, 50-372 Wroclaw, Poland.

2. MULTILAYER PERCEPTRON ALGORITHM – RETRIEVING PHASE

The Multilayer Perceptron network is composed by sets of neurons which create the layers. The classical net includes three layers: input layer, hidden layer, output layer. Each layer consists of the proper number of neurons. The size of input layer equals to the size of input vector, the size of output layer is related to the code used for the output of the network description and finally the number of hidden layer neurons is estimated by different ways. For our discussion let assume that we have: N neurons in input layer, K neurons in hidden layer, L neurons in output layer. It means that we operate with N-elements size input vectors and L-elements size output vectors. It is possible – of course – to discuss more than single hidden layer, but it does not change the general idea of presented approach. here are no connections among the neurons from the same layer, but output signal from single neuron is transmitted as input signal to all neurons from the next layer. So the outputs from the input layer neurons are inputs from the hidden layer neurons and the outputs from the hidden layer neurons are the inputs for the output layer neurons. The only task related to the neurons from input layer is input vector components transfer to neurons from hidden layer. This way there is no need to discuss their implementation – we have only to guarantee the input vector components transfer to all neurons from hidden layer. Neurons from hidden and output layers realise exactly the same operation, but using different data. For hidden layer neurons we can describe the following equation [2]:

$$u_i = f(\sum_{l=1}^{N} x_l w_{li}^{(1)}) \tag{1}$$

where:

u_i – output value calculated by single neuron from hidden layer – single component of K-elements size vector generated by neurons from hidden layer,

x_l – component of N-elements size input vector,

$w_{li}^{(1)}$– weight associated with connection from component of input vector x_l and neuron from hidden layer indexed by i,

$f(\,)$ – non-linear, usually sigmoid, neuron activation function.

For output layer neurons we can describe the following equation:

$$y_i = f(\sum_{l=1}^{K} u_l w_{li}^{(2)}) \tag{2}$$

where:

y_i – output value calculated by single neuron from output layer – single component of L-elements size vector generated by neurons from output layer,

u_l – component of K-elements size vector generated by neurons from hidden layer,

$w_{li}^{(2)}$ – weight associated with connection from component of vector generated by neurons from hidden layer u_l and neuron from output layer indexed by i,

$f(\,)$ – non-linear, usually sigmoid, neuron activation function.

As we can notice each neuron calculates the weighted sum which is an argument of neuron activation function. Calculations related to neurons from the same layer can be done in parallel mode, but the sequence of operations ought to be preserved in order to succeeding layers. We assume – of course – that the values of weights are fixed and ready to use – as a product of any proper for Multilayer Perceptron learning algorithm.

3. DATA DEPENDENCE GRAPHS FOR MULTILAYER PERCEPTRON DURING RETRIEVING PHASE

A Data Dependence Graph is a directed graph that specifies the data dependencies of an algorithm. In a Data Dependence Graph nodes represent computations and arcs specify the data dependencies between computations. For regular and recursive algorithms, the Data Dependencies Graphs are also regular and can be represented by a grid model. Design of a locally linked Data Dependence Graph is a critical step in the design of systolic array [6]. The Data Dependence Graphs for Multilayer Perceptron retrieving algorithm ought to be discuss individually for each layer. Let's start from the hidden layer. The input layer – as we noticed before is responsible only for proper distribution of input vector components. The hidden layer includes K neurons and each neuron from this layer collects signals from N neurons related to input layer. For such topology there are ($N \times L$) weights – obtained during learning phase [6]. The hidden layer ought to be described by rectangular Data Dependence Graph (Fig. 1.). Each node in this graph – excluding the last horizontal line of nodes – is responsible for elementary multiplication product calculation. This means that each node realises the operations described by rule (1), but without summing using proper value of component of N-elements size input vector and the proper weight and without the value of activation function calculation. The local memory of each node should put the value of the single weight obtained earlier after learning algorithm. The size of the graph equals to the size of the weight matrix plus single extra horizontal line with nodes responsible for activation function calculation. Each node of the graph should be loaded by two signals. The first one is the component of the input vector. The second one is the current value of weighted sum calculated by single neuron of Multilayer Perceptron. So each node ought to add thecalculated product of multiplication to the loaded previous value of weighted sum signal (1) an pass updated value to the next node in the same column. Such proposed solution requires capacity of the local memory of each node large enough to store the single weight, but this way reduces to minimum the number of data which ought to be transmitted during retrieving algorithm realisation by presented Data Dependence Graph. The operations realised by single neuron from Multilayer Perceptron are described by the single column of the graph. The components of input vector are loaded to nodes by horizontal arcs of the Data Dependence Graphs. These values are passed to the next neighbour on the right hand. The current value of weighted sum generated by single neuron of Multilayer Perceptron is loaded by vertical arcs (Fig. 1.). The update this value is passed to the next bottom neighbour. This way we can observe only point-to-point communication among the nodes which means that presented Data Dependence Graph is local graph. This property guarantees construction of well defined systolic array to realise the retrieving algorithm of Multilayer Perceptron [4]. The nodes from the last – extra horizontal line calculates the value of activation function when the previously calculated weighted sum is the argument. We propose to realise this operation using lookup table instead of analytical calculation. The only problem is that the local memory ought to store the lookup table. Now let's try to discuss the Data Dependence Graph for output layer of Multilayer Perceptron. In general steps which ought to be realised are the same, but the number of neurons is different and neurons are loaded by the vector generated by neurons from hidden layer. The output layer includes L neurons and each neuron from this layer collects signals from K neurons related to hidden layer. For such topology there are $(L \times K)$ weights – obtained during learning phase [6]. The output layer ought to be described by rectangular Data Dependence Graph (Fig. 1.). Each node in this graph – excluding the last horizontal line of nodes – is responsible for elementary multiplication product calculation. This means that each node realises the operations described by rule (2), but without summing using proper value of component of K-elements size vector generated by hidden layer neurons and the proper weight, and without the value of activation function calculation. The local memory of each node should put the value of the single weight obtained earlier after learning algorithm [4].

The size of the graph equals to the size of the weight matrix plus single extra horizontal line with nodes responsible for activation function calculation. Each node of the graph should be loaded by two signals. The first one is the component of the vector gen-

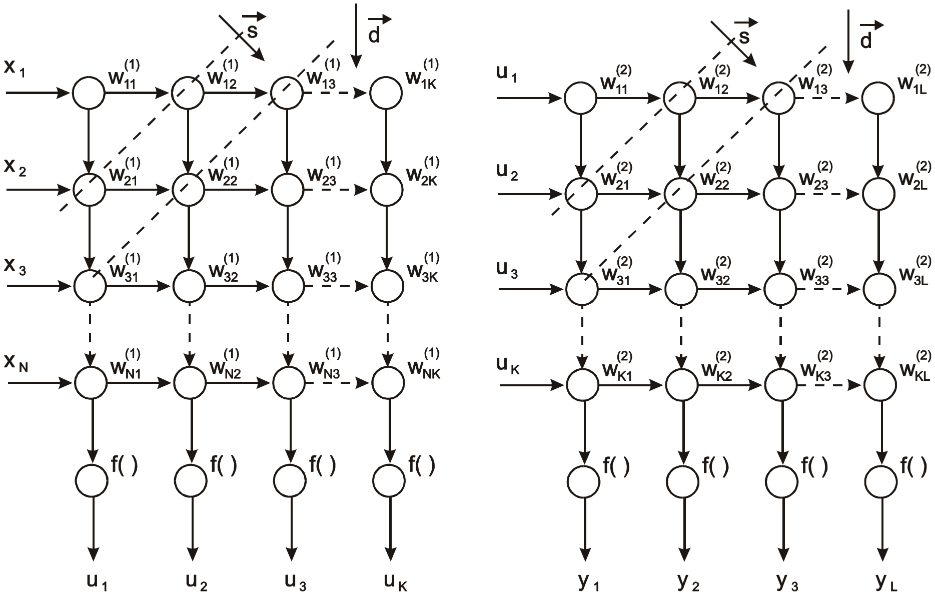


Fig. 1. Data Dependence Graphs of Multilayer Perceptron during retrieving algorithm: for hidden layer (left) and for output layer (right)

erated by hidden layer neurons. The second one is the current value of weighted sum calculated by single neuron of Multilayer Perceptron. So each node ought to add the calculated product of multiplication to the loaded previous value of weighted sum signal (2) an pass updated value to the next node in the same column. Such proposed solution requires capacity of the local memory of each node large enough to store the single weight, but this way reduces to minimum the number of data which ought to be transmitted during retrieving algorithm realisation by presented Data Dependence Graph. The operations realised by single neuron from Multilayer Perceptron are described by the single column of the graph. The components of the vector generated by hidden layer neurons are loaded to nodes by horizontal arcs of the Data Dependence Graphs. These values are passed to the next neighbour on the right hand. The current value of weighted sum generated by single neuron of Multilayer Perceptron is loaded by vertical arcs (Fig. 1.). The update this value is passed to the next bottom neighbour. This way we can observe only point-to-point communication among the nodes which means that presented Data Dependence Graph is local graph. This property guarantees construction of well defined systolic array to realise the retrieving algorithm of Multilayer Perceptron [4]. The nodes from the last – extra horizontal line calculates the value of activation function when the previously calculated weighted sum is the argument. We propose to realise this operation using lookup table instead of analytical calculation. In general Data Dependence Graphs presented above are quite similar and this is a chance to create the single set of Elementary Processors (PE) with switched functions to operate first with hidden layer and next with output layer [5].

4. MAPPING DATA DEPENDENCE GRAPHS ONTO SYSTOLIC ARRAY STRUCTURE

4.1. PROCESSOR ASSIGNMENT VIA LINEAR PROJECTION

Mathematically, a linear projection is often represented by a *projection vector* $\vec{d}$. Because the Data Dependence Graph of a locally recursive algorithm is very regular, the linear projection maps an *n*-dimensional Data Dependence Graph onto an *(n-1)* dimensional lattice of points, known as processor space [3]. We use a linear projection for processor assignment, in which nodes of Data Dependence Graph along a straight line are projected to an Elementary Processor in the processor array (Fig. 1.).

4.2. SCHEDULE ASSIGNMENT VIA LINEAR SCHEDULING

A scheduling scheme specifies the sequence of the operations in all Elementary Processors. More precisely, a schedule function represents a mapping from the *n*-dimensional index space of the Data Dependence Graph onto a 1-D schedule (time) space. Linear scheduling is very common for schedule assignment (Fig. 1.). A linear schedule is based on a set of parallel and uniformly spaced hyperplanes in the Data Dependence Graph. A linear schedule can also be represented by a *schedule vector* $\vec{s}$, which points in the direction normal to the hyperplanes. For any computation node indexed by a vector *n* in the Data Dependence Graph, its scheduled processing time is $\vec{s}n$.

4.3. MAPPING POLICIES

Given a Data Dependence Graph and the projection direction $\vec{d}$ not all schedule vectors $\vec{s}$ are valid for the Data Dependence Graph. Some may violate the precedence relations specified by the dependence arcs. For systolic design, the schedule vector $\vec{s}$ in the projection procedure must satisfy the following two conditions [3]:

– causality condition: $\vec{s}^T\vec{e} > 0$ (3)

$\vec{e}$ – represents any of the dependence arcs in the Data Dependence Graph

– positive pipeline period: $\vec{s}^T\vec{d} \neq 0$ (4)

This way the rectangular Data Dependence Graphs are converted into linear pipelined systolic arrays. This situation we can observe both for hidden and output layers for Multilayer Perceptron (Fig. 3.). The number of elementary processors which are used for array construction equals to the number of neurons in simulated layers of Multilayer Perceptron neural network. Each elementary processor combines all functions described by nodes of Data Dependence Graph placed at the same column of single slab of Data Dependence Graph. If we want to reduce the number of elementary processors we can change the classical systolic structure into ring structure. Of course the reduction of number of elementary processors ought to be done in the way which preserve the same number of neurons for single processor [1].

5. EFFICIENCY OF PROPOSED APPROACH

The computation time is the interval between starting the first computation and finishing the last computation of problem. Given a coprime schedule vector $\vec{s}$, the computation time of a systolic array can be computed as [1]:

$$T = \max_{\vec{p},\vec{q} \in L} \left\{ \vec{s}^{T} \left(\vec{p} - \vec{q} \right) \right\} + 1 \tag{5}$$

where $\boldsymbol{L}$ is the index set of the nodes in the Data Dependence Graph. In the presented architecture the schedule vector is defined as: $\vec{s} = [1,1]$. The total computation time is a sum of computation time related to hidden layer and computation time related to output layer. These two values ought to be calculated independently because operations of each layer are realised in sequence and we have two independent Data Dependence Graph.

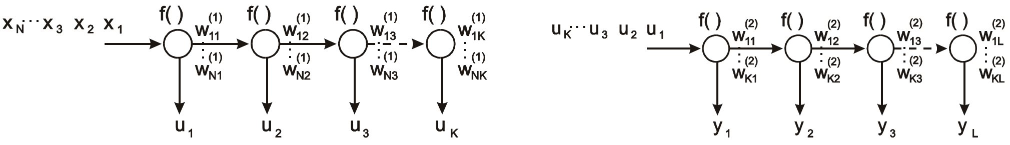


Fig. 2. Systolic array of Multilayer Perceptron during retrieving algorithm: for hidden layer (left) and for output layer (right)

So total computation time equals:

$$T_{systol} = T_{syshid} + T_{sysout} \tag{6}$$

where: T_{syshid} – computation time for hidden layer, T_{sysout} – computation time for output layer

For hidden layer we have *N+1* elements within vertical axis and *K* within horizontal axis of space. For output layer we have *K+1* elements within vertical axis and *L* within horizontal axis of space (Fig. 1.). Based on this remarks the computation time we can estimate as [5]:

$$T_{syshid} = (N + K)\tau \qquad T_{sysout} = (K + L)\tau \tag{7}$$

Based on (6) finally we can say that:

$$T_{systol} = (N + 2K + L)\tau \tag{8}$$

where τ – processing time for elementary processor

5.2. PIPELINING PERIOD

This is the time interval between two successive computations in a processor. As previously discussed, if both $\vec{d}$ and $\vec{s}$ are irreducible, then the pipelining period is [1]:

$$\alpha{=}\vec{s}^{\,T}\vec{d} \tag{9}$$

In the presented approach the schedule vector is defined as: $\vec{s}$ = [1,1], the projection direction vector equals: $\vec{d}$ = [0,1]. The pipelining period is constant: $\alpha = 1$. It means the time interval between two successive computations in an elementary processor is as short as possible. The pipelining period is exactly the same for both Data Dependence Graphs and there is no sense to combine them into single value.

5.3. PROCESSOR UTILIZATION RATE

Lets define the speed-up factor as the ratio between the sequential computation time and the array computation time, then the utilization rate is the ration between the speed-up factor and the number of processors [3].

$$speed-up = \frac{sequential\ computation\ time}{array\ computation\ time} \qquad utilization\ rate{=}\frac{speed-up}{number\ of\ processors} \tag{10}$$

Sequential computation time always is proportional to the number of nodes, which are responsible for calculations. On the other hand time of computations related to each layer ought to be analysed independently because of operation sequence in Multilayer Perceptron. Based on these remarks the total sequential computation time equals:

$$T_{seq} = T_{seqhid} + T_{seqout} \tag{11}$$

For hidden layer sequential computation time we can estimate as:

$$T_{seqhid} = (N+1)K\tau \qquad T_{seqout} = (K+1)L\tau \tag{12}$$

$$T_{seq} = \left((N+L+1)K + L\right)\tau \tag{13}$$

For systolic implementation we need *K* elementary processors to realise calculations for hidden layer and *L* elementary processors to realise calculations for output layer respectively. If we assume that all elementary processors need the same time-period to proceed their calculations the speed-up and utilization rate factors for three layer Multilayer Perceptron can be estimated as [4]:

$$speed-up = \frac{(N+L+1)K+L}{N+2K+L} \qquad utilizationrate = \frac{(N+L+1)K+L}{(N+2K+L)(K+L)} \tag{14}$$

6. CONCLUSION

Summarising, the work proposed a new methodology for retrieving algorithm of Multilayer Perceptron neural network simulation based on systolic array structure. The discussion is focused on operations which are realised during the following steps of algorithm and the data which are transferred among the calculation units. It is clear which operations can be done in parallel way and when the sequence is necessary. We can notice as very promising the results of estimation of total computation time, speed-up and utilization rate parameters. The proposed methodology is presented for three layer network – because such Multilayer Perceptron is very often used in different tasks. On the other hand there are no barriers to adapt the solution for more sophisticated Multilayer Perceptrons. The main idea is to create next Data Dependence Graphs for next layers and transfer them into linear systolic structures. The time parameters in efficiency calculation will increase in additive way. The proposed approach generates no barriers to tune the Multilayer Perceptron to completely new tasks.

REFERENCES

[1] CABESTANY J., IENNE P., MORENO J.M., MADRENAS J., *Is There a future for ANN Hardware?* Workshop on Mixed Design of Integrated Circuits and Systems, Poland, Lodz, 1996.

[2] IENNE P., THIRAN P., VASSILAS N., *Modified Self-Organizing Feature Map Algorithms for Efficient Digital Hardware Implementation*. IEEE Transactions on Neural Networks. 8, No. 2, 1997, pp. 315–330.

[3] KUNG S.Y., *Digital Neural Networks*, PTR Prentice Hall, 1993.

[4] MAZURKIEWICZ J., *Dependable SIMD Architecture for Artificial Neural Networks*, 4th International Conference on Information Technology ICIT 2009, DeSyT Session, Al-Zaytoonah University, Jordan, Amman, 2009, pp. 95–102.

[5] MAZURKIEWICZ J., *SIMD Architecture Approach to Artificial Neural Networks Realisation*, Computational Intelligence and Modern Heuristics, In-Teh, Croatia, Vukovar, 2010, 11, pp. 159–174.

[6] ZHANG D., *Parallel VLSI Neural System Design*, Springer-Verlag, 1999.

Block Packing Problem, Sequence-Pair, VLSI, design automation, algorithm, floorplanning

Andrzej KOZIK*, Bartosz TOMECZKO*

SOLVING BLOCK PACKING INTO RECTILINEAR OUTLINE PROBLEM BY SIMULATED ANNEALING

In this work, we consider a simulated annealing (SA) based solution to the block packing into rectilinear outline problem. The goal is to place a set of blocks into a given rectilinear outline such that no two blocks overlap and a given cost function (involving blocks interconnection structure) is minimized.

Contrary to the previous approaches, we primarily limit the set of random moves performed by basic SA algorithm to those, which lead to feasible solutions. In each iteration of our SA algorithm, we perform a neighborhood pruning, i.e., we apply an incremental SP neighborhood evaluation algorithm, to create the set of all feasible solutions, letting SA to perform a random selection from this set.

Numerical experiment, in which we compare our algorithm with basic SA (without neighborhood pruning) clearly showed that the presented algorithm is stable, i.e., assuming the initial solution is feasible – the delivered solution is also feasible. On the other hand, the competing algorithm often fails to provide a feasible solution at all.

Importantly, the obtained costs of delivered solutions (total length of interconnections between blocks) are far better than those of the competing algorithm, often being better than the best recorded in the literature so far.

1. INTRODUCTION

The recent advances in VLSI (Very Large Scale of Integration) digital circuits production [8, 11] were possible with the shift towards new methods in physical design automation. Namely, the fixed-outline paradigm in floorplanning [1, 2, 3], in which circuits building blocks (transistors, cells or modules) are placed within *bins* of determined shape (rectangle) and size, proved to produce wirelength-optimized designs. As

* Institute of Computer Engineering, Control and Robotics, Janiszewskiego 11/17, 50-372 Wrocław.

the problem is new and not much have been done so far, a Simulated Annealing (SA) [7] approach is used as a solution search framework. In this work we propose a modified SA, in which we limit the set of random moves performed by plain SA algorithm to those, which lead to feasible solutions. The rest of the work is organized as follows. The nest section formulates the problem and describes Sequence Pair solution representation. Section 3 describes Simulated Annealing approach. In Section 4 results of an numerical experiments are presented and discussed. Finally, Section 5 concludes the work.

2. PROBLEM FORMULATION

There is given a set $B = \{1, \ldots, n\}$ of n rectangular blocks. Each block $i \in B$ is characterized by its width w_i, height h_i and area $a_i = w_i \cdot h_i$. The goal of a block packing into rectilinear outline problem (fixed-outline packing problem) is to find such a packing, i.e., placement coordinates of a bottom-left corner (x_i, y_i) and orientation (horizontally or vertically laid) of each block $i \in B$, that no two blocks overlap, all blocks fit into a given rectangular (or more complex rectilinear) outline and a length of circuit interconnections (a wirelength) is minimized [10].

The interconnect structure of the circuit naturally decomposes into separate nets, where each *net* connects together some *terminals* of the circuit. There are two types of terminals: *pins*, attached to the blocks, and *pads*, having fixed locations over the packing, typically along the sides of a rectangle called a *frame*. A sample packingsolution is presented in Fig. 1.

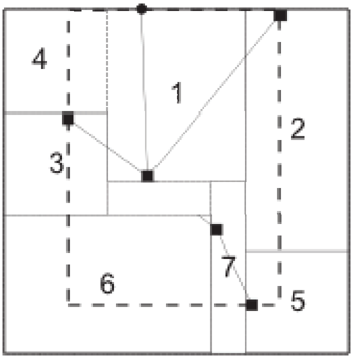


Fig. 1. Sample packing of 7 blocks; black squares represent pins, circle represents a pad, terminals are connected by net; dotted box represents bounding-box of the net

The net wirelength is defined as the length of its routing-tracks in metal layer, and the circuit wirelength is the sum of wirelengths of all nets [4]. Because accurate wirelength is not available at the floorplanning stage, a half-perimeter-wirelength (HPWL) is widely accepted in the floorplanning field as an sufficient estimate. In such an approach, the wirelength of a net is approximated by a half of the perimeter of a minimum rectangle enclosing all net terminals – bounding-box of interconnections (see, Fig. 1). The HPWL of a packing is the sum of the HPWLs over all nets.

To represent a packing solution and to create a solution space to be traversed for the optimal packing we make use of a packing representation called Sequence-Pair (SP), introduced by Murata et al. [6]. A sequence pair, say (Γ_+, Γ_-), is a pair of *n*-element sequences (permutations), each representing some order of elements of the set B of blocks having orientations and dimensions fixed. Sequence-Pair, as well as other known representations, is useful only for reflecting a packing solution into its appropriate coding. The solution space provided by SP consists of $(n!)^2$ sequence-pairs, each of which can be mapped to a packing in polynomial time, and at least one of which corresponds to an optimal solution. SP, however, is not free from redundancies, since there is no one-to-one mapping between floorplan and coding.

3. SIMULATED ANNEALING

The Simulated Annealing (SA) algorithm [7] searches the search-space spanned by Sequence-Pair representation, starting from an initial solution s_0 delivered by GIT algorithm [5]. The GIT constructive heuristic neglects HPWL issues in floorplanning, it provides, however, a feasible solution with low dead-space ratio (*dead-space ratio* (DS) determines a percentage of free space inside a given rectilinear outline in comparison with sum of areas of all blocks).

Then, at each iteration a solution $s' \in \mathrm{N}(s)$ is randomly sampled from the neighborhood of the actual solution s. The new solution s' replaces the old solution s if $f(s') < f(s)$ or, in the case $f(s') \geq f(s)$, with a probability computed following the Boltzmann distribution $e^{-\frac{f(s')-f(s)}{T}}$, where *T* is so-called temperature parameter and $f(s)$ is HPWL of solution s. The temperature *T* is decreased during the solution search process by a *cooling schedule*, which defines the value of *T* at each iteration *k*=1,2,.. . In this work we have applied two cooling schedules: logarithmic law $T_k = \dfrac{T_0}{\ln(k+1)}$ and geometric law $T_k = \alpha T_{k-1}$ where $\alpha \in (0,1)$.

The traditional application of the Simulated Annealing to the fixed-outline packing problem defines a neighborhood of the current solution $N(s)$ as a set of all sequence pairs that can be obtained by performing a single insert move (changing positions of some element in (Γ_+, Γ_-)). This can often lead the algorithm to fail to find a feasible solution (better than s_0) at all. Contrary to the previous approaches, we primarily limit the set of random moves performed by basic SA algorithm to those, which lead to feasible solutions – the neighborhood $N(s)$ consists of feasible solutions only. To this end, in each iteration of our SA algorithm, we perform a neighborhood pruning, i.e., we apply an incremental SP neighborhood evaluation algorithm [9], to create the set of all feasible solutions, letting SA to perform a random selection from this set.

To evaluate the impact of redundancy of Sequence Pair representation, we introduced yet another definition of neighborhood, i.e., $N(s)$ is limited to *contain geometrically different solutions* only – it contains sequence pairs that are decoded into different placements.

4. COMPUTATIONAL RESULTS AND DISCUSSION

This section describes an numerical experiment, in which we compare our algorithms with basic SA (without any neighborhood pruning). Table 3 presents results obtained by Parquet [2], the leading-edge floorplanning algorithm. We recorded minimal HPWL from 10 runs on default algorithm settings. In the case of DS equal 1.13 and 1.15 all runs terminated with feasible solutions; in the case of 1.11 four runs and in the case 1.10 five runs failed to deliver feasible solution, respectively (Parquet starts with random solution).

Table 1. Results of plain SA

Circuit	DS			
	1.10	1.11	1.13	1.15
n50	201222	191131	197424	192654
n100	332941	335597	344671	338437

We developed four variants of SA: A_1 using geometric cooling schedule without neighborhood reduction, A_2 using geometric cooling schedule with neighborhood reduction, A_3 using logarithmic cooling schedule without neighborhood reduction, and A_4 using logarithmic cooling schedule with neighborhood reduction.

To evaluate the quality of proposed algorithms we measured their performance on commonly used GSRC benchmarks (n50 composed of 50 blocks and 485 nets and n100 composed of 100 blocks and 885 nets). The experiments were ran on Intel Core 2 Quad 2,66GHz CPU with 4GB of RAM memory.

In the case of n50 we set number of iterations $= 10000$, $T_0 = 1000$, $\alpha = 0{,}9997$, and in the case of n100 we set number of iterations $= 1000$, $T_0 = 1000$, $\alpha = 0{,}997$.

Results are gathered in Table 2 and 3 and presented in Fig. 2 and 3. In both cases (n50 and n100) we noticed a big improvement in HPWL comparing to the initial value. Importantly, the obtained costs of delivered solutions are better than those of the competing algorithm, often being better than the best recorded in the literature so far, especially in the range of low dead-space ratios, where the neighborhood size is small due to area constraint and competing algorithm is not stable.

Surprisingly, algorithms with reduced neighborhood fare badly in comparison with algorithms without reduction. This contradicts the common belief, that SP redundancy is a major flaw of SP. Observe, that even SA seems to be deadlocked in the same solution (in the geometrical sense), the underlying sequence pairs differ, allowing to escape from local minima after some number of moves.

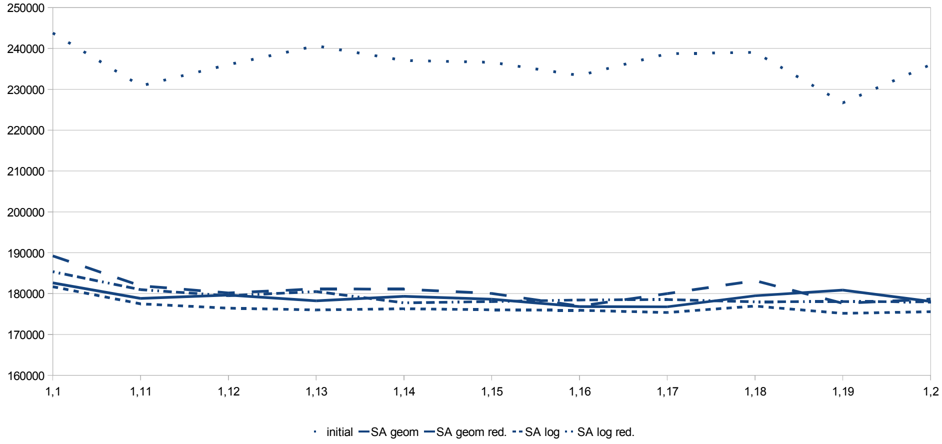


Fig. 2. HPWL results for n50 circuit as a function of dead space ratio

In Fig. 4 the sizes of the neighborhood with and without reduction for different DS ratios are presented. Note, that in plain SA approach, the sizes equal to 25000 and 60000, in the case of n50 and n100, respectively. In comparison with our approach, a plain SA traverses rather unfeasible region of solution-space. Observe also a big reduction of neighborhood size caused by excluding geometrically identical solutions from the search process. Apart of the high computational overhead needed to prune the neighborhood (depicted in Fig. 5), such obtained solutions are far worse than solutions delivered by SA without reduced neighborhood.

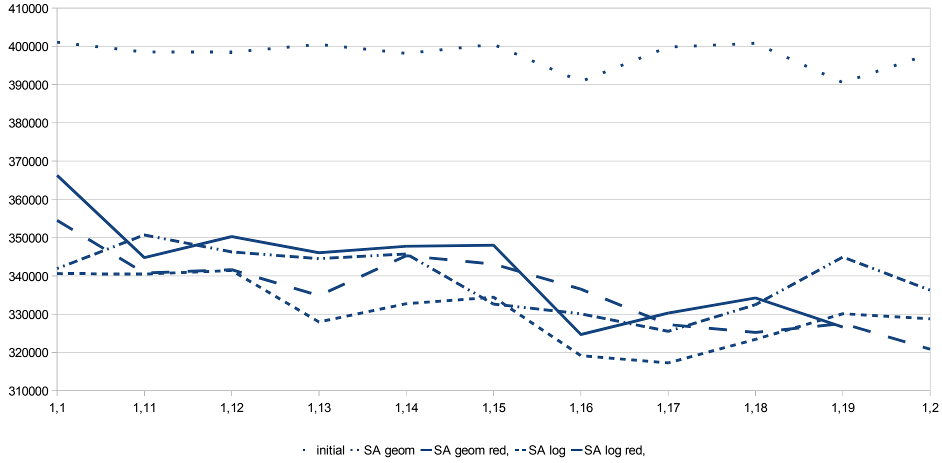


Fig. 3. HPWL results for n100 circuit
as a function of dead space ratio

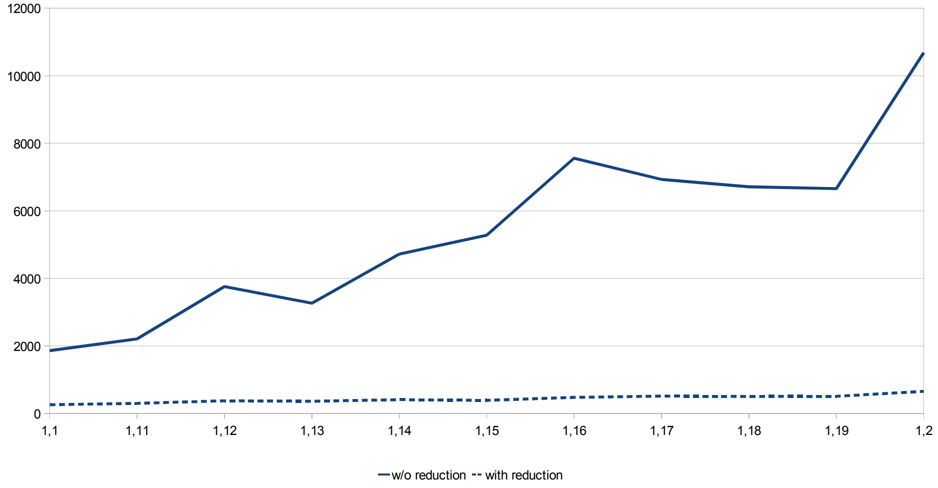


Fig. 4. Sizes of neighborhoods with and without reduction

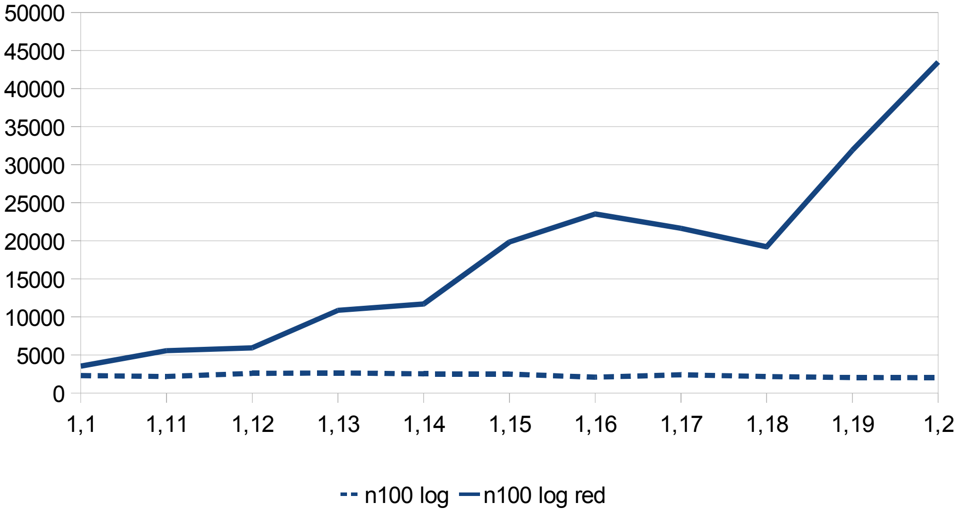


Fig. 5. Computation time for SA with and without neighborhood reduction

5. CONCLUSIONS

In this work we developed a novel Simulated Annealing based solution to the block packing into rectilinear outline problem. Contrary to the previous approaches, we limited the set of random moves performed by basic SA algorithm to those, which lead to feasible solutions. In each iteration of our SA algorithm, we perform a neighborhood pruning, i.e., we apply an incremental SP neighborhood evaluation algorithm, to create the set of all feasible solutions, letting SA to perform a random selection from this set.

Numerical experiment, in which we compare our algorithm with basic SA (without neighborhood pruning) clearly showed that the presented algorithm is stable, i.e., assuming the initial solution is feasible – the delivered solution is also feasible. On the other hand, the competing algorithm often fails to provide a feasible solution at all.

Importantly, the obtained HPWL values are far better than those of the competing algorithm, often being better than the best recorded in the literature so far.

Table 2. Computation results for n50 test circuit

algorithm	DS	initial	minimal	average	maximal	time (s)
A1	1,10	243773	182613	185844	196836	2846
	1,11	230829	178827	182369	195869	3812
	1,12	236029	179649	180537	185464	3222
	1,13	240630	178241	179853	186745	2869
	1,14	237047	179318	179012	181457	2859
	1,15	236594	178624	180192	186789	2880
	1,16	233370	176819	181946	193637	3146
	1,17	238704	176762	179068	185307	3216
	1,18	239046	179465	180756	183102	2845
	1,19	226683	180849	179649	181845	3189
	1,20	236019	178071	178906	181109	3920
A2	1,10	243773	189213	192138	203529	3187
	1,11	230829	181916	184804	200558	3682
	1,12	236029	180161	182933	188901	4590
	1,13	240630	181143	184335	191511	4606
	1,14	237047	181076	181707	182593	5811
	1,15	236594	180027	182299	185485	5653
	1,16	233370	176937	181177	185599	7242
	1,17	238704	179982	181090	183523	9869
	1,18	239046	183161	181803	189930	11209
	1,19	226683	177608	180126	181772	11432
	1,20	236019	178653	180911	182850	15178
A3	1,10	243773	181681	188055	206482	2353
	1,11	230829	177487	182545	195898	3757
	1,12	236029	176431	182641	192371	3736
	1,13	240630	175999	178793	195757	4444
	1,14	237047	176297	179485	190324	3756
	1,15	236594	176035	177908	186899	3374
	1,16	233370	175921	177941	185389	3527
	1,17	238704	175370	176949	184508	4353
	1,18	239046	176943	178556	183329	4487
	1,19	226683	175160	177588	183061	3977
	1,20	236019	175570	176504	181119	3461
A4	1,10	243773	185348	191797	206089	3161
	1,11	230829	180964	183544	192547	2869
	1,12	236029	179487	185111	203354	4150
	1,13	240630	180493	184633	194827	3306
	1,14	237047	177738	179088	182609	4832
	1,15	236594	178035	179641	190545	6195
	1,16	233370	178440	181325	191765	4992
	1,17	238704	178549	178467	184075	6352
	1,18	239046	177986	182895	189655	6988
	1,19	226683	178058	180017	191886	8613
	1,20	236019	177974	179038	181570	9567

Table 3. Computation results for n100 test circuit

algorithm	DS	initial	minimal	average	maximal	time [s]
A1	1,10	401058	341918	353886	373573	2282
	1,11	398517	350700	359857	373048	1888
	1,12	398476	346261	355069	367390	1903
	1,13	400488	344512	356282	375489	1942
	1,14	398154	345740	353806	366758	2329
	1,15	400483	332628	343025	367287	1930
	1,16	390797	330091	341259	356862	1947
	1,17	399805	325516	339418	361847	1917
	1,18	400777	332466	347310	363985	2321
	1,19	390517	344909	346489	356551	2588
	1,20	397976	336269	341329	351156	2341
A2	1,10	401058	366282	375301	385658	3672
	1,11	398517	344774	360367	373968	5031
	1,12	398476	350266	360584	381828	6482
	1,13	400488	346066	362342	375774	12103
	1,14	398154	347744	359161	370067	11947
	1,15	400483	348011	356561	371924	17539
	1,16	390797	324680	340192	367177	28073
	1,17	399805	330293	342263	370657	29451
	1,18	400777	334241	349594	368481	26540
	1,19	390517	326644	340381	367295	48953
	1,20	397976	321351	335271	354462	47644
A3	1,10	401058	340597	358666	382840	2294
	1,11	398517	340469	354646	377055	2168
	1,12	398476	341391	350926	369568	2579
	1,13	400488	327964	342130	366280	2635
	1,14	398154	332740	346236	367282	2348
	1,15	400483	334430	349199	377742	2480
	1,16	390797	319145	332660	350864	2079
	1,17	399805	317264	331122	359445	2391
	1,18	400777	323412	335034	356265	2161
	1,19	390517	330116	337192	359441	2040
	1,20	397976	328779	342067	358889	2014
A4	1,10	401058	354490	366769	383247	3533
	1,11	398517	340748	354585	372989	5556
	1,12	398476	341610	356264	373785	5932
	1,13	400488	334851	353412	378751	10857
	1,14	398154	345275	358902	378799	11702
	1,15	400483	343081	353095	370513	19842
	1,16	390797	336536	347610	366369	23520
	1,17	399805	327277	339740	367361	21636
	1,18	400777	325220	341193	371138	19217
	1,19	390517	327543	339470	360370	31904
	1,20	397976	320847	337666	359871	43487

ACKNOWLEDGMENTS

Andrzej Kozik's work was supported by Polish Ministry of Science and Higher Education under grant no. N519 408337. Bartosz Tomeczko's work was supported by Polish National Science Centre under grant no. N516 494140.

REFERENCES

[1] ADYA S. N., CHAN H. H., LU J. F., MARKOV I. L., NG A. N., PAPA D. A., ROY J. A. *CAPO: Robust and Scalable Open-Source Min-Cut Floorplacer*, ISPD '05, 2005.

[2] ADYA S. N., MARKOV I. L. *Fixed-outline floorplanning: enabling hierarchical design*. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2003, 11(6): 1120–1135.

[3] ADYA S. N., MARKOV I.L. *Combinatorial techniques for mixed-size placement*. ACM Transactions on Design Automation of Electronic Systems, 2005, 10(5):58–90.

[4] BASARAN B., GANESH K., LEVIN A., MCCOO M., RANGARAJAN S., RAU R., SEHGAL N. *GeneSys – a layout synthesis system for GHz VLSI designs*. Proc. Int. Conf. in VLSI Design, 1999, 458–472.

[5] ENSCORE E. E. JR, HAM I., NAWAZ M. *A heuristic algorithm for m-machine, n-job flow-shop sequencing problem*. OMEGA International Journal of Management Science (11), 1983, 91–95.

[6] FUJIYOSHI K., MURATA H., KAJITANI Y. AND NAKATAKE S. *VLSI module placement based on rectangle-packing by the sequence pair*. IEEE Trans. on CAD of ICs., 1996, 15:1518–1524.

[7] GELATT C.D., KIRKPATRICK S., VECCHI M. P. *Optimization by Simulated Annealing*. Science 220 (4598), 1983, 671–680.

[8] HAYES J.P. AND MURRAY B.T. *Testing ICs: getting to the core of the problem*. IEEE Computer Magazine, 1996, 11:32–38.

[9] JANIAK A., KOZIK A., LICHTENSTEIN M. *New perspectives in VLSI design automation: deterministic packing by Sequence Pair*. Annals of Operations Research. vol. 179, 2010, nr 1: 35–56.

[10] JANIAK A., KOZIK A., TOMECZKO B. *Complex outline packing problem in VLSI physical design.* Information systems architecture and technology: system analysis in decision aided problems Wrocław: Oficyna Wydawnicza Politechniki Wrocławskiej, 2009. s. 343–354.

[11] KUEHLMANN A. *The best of ICCAD – 20 years of excellence in computeraided design*. Kluwer Academic Pub., 2003.

*Travelling Salesman Problem. Ant Colony System,
Simulated Annealing, Genetic Programming,
parameter optimization*

Andrzej SIEMIŃSKI*

TSP/ACO PARAMETER OPTIMIZATION

The Traveling Salesmen Problem (TSP) is one of the most successful application areas of the Ant Colony Optimization techniques (ACO). The ACO algorithm is controlled by a number of parameters. The selection of proper parameters values for the ACO is done mainly in an experimental manner. This is due to the complexity of both the TSP problem and the ACO algorithm itself. The work attempts to automate the process parameter optimization using an algorithm which is inspired by a combination if Evolutionally Programming (EP) and Simulated Annealing (SA). Each of genes making up a chromosome represents a single parameter of the ACO algorithm. The population of chromosomes evolves using the typical for EP selection and mutation mechanisms. The mutation is controlled by an algorithm adopted from Simulated Annealing. The work first presents the original versions of the basic algorithms and their proposed modifications. The usefulness of the proposed algorithm is verified by a number of experiments.

1. INTRODUCTION

Metaheuristics are approaches to solve a complex problem by iteratively improving candidate solutions. They are not guaranteed to find the optimum or even a satisfactory near-optimal solution. Despite this deficiency, in the case of the NP-hard or even NP-complete problems they are the often the only available choice. Theoretical analysis of metaheuristics is very difficult. Usually such analyses are not available for every method at all. The existing analysis make often a number of simplifying assumptions which limit their validity in real world optimization scenarios. Therefore the performance and convergence aspects of metaheuristic optimizers are usually demonstrated empirically.

* Institute of Informatics, Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław.

The work deals with the optimization of the parameters of the AntTSP metaheuristics. It is designed to solve the well known Traveling Salesmen using Ant Colony Optimization. The problem consists in finding a shortest possible tour that visits each city exactly once. It is a NP-hard problem. It is the problem that had originally inspired the Ant Colony Optimization techniques (ACO) and still remains one of their most successful application areas. An ACO algorithm is controlled by a number of parameters. The selection of proper parameters values for the ACO is done mainly in an experimental manner. The work attempts to automate the process parameter optimization using other mataheuristics namely Simulated Annealing (SA) and Evolutionally Programming (EP).

The work is organized as follows. The second Section describes the basic version of the used metaheuritics: Ant Colony Optimization, Evolutionally Programming and Simulated Annealing. The Section 3 presents the implementation details and proposed modifications to the basic metaheuristics and the describes the way in which they cooperate. The 4th Section summarizes the results of the conducted experiments. The work ends with a brief presentation of future research work.

2. USED METAHEURISTICS

2.1. ANT COLONY SYSTEM

The Ant Colony System (ACS) was first proposed by M. Dorigo in his PhD thesis [4] in 1992. The ACS was inspired by the behavior of real ants. An ant could be regarded as an extremely simple agent. It is capable only of going from one node to another laying a pheromone trail on its way. The mental capabilities of an ant are very limited. It can remember its current position, the nodes it has visited so far and sense the direct distances from its current position to other nodes together with the amount of pheromone laid on them. The selection of a next node is takes into account the above mentioned factors but is at the same time to certain extend random. The amount of pheromone laid by an ant depends on the length of the route it follows. As in real life, the pheromone laid on routes evaporates with time.

U. Chirico [2] described and made available to the research community the JACSF – Java Ant Colony System Framework. The following description refers to his implementation of the framework on TSP. The implementation with only marginal modifications is used also in the experiment discussed in the Section 4.

The behavior of an ant is controlled by the following rules:

- a State Transition Rule used for selecting the next node to visit;
- a Local Updating Rule which updates the pheromones deposited by the ant on the route it walked in;

- a Global Updating Rule which is used to update the pheromones deposited on the routes when an ant ends its trip;

  In what follows:
- *r*, *t* and *u* denote graphs' nodes;
- $\eta(r, t)$ denote the function of the distance separating the r and t nodes, the values of the function are in the range [0..1].
- $\tau(r, t)$ denotes the function of pheromone deposit on the that was laid on the route from nodes r to t; its values are guaranteed to be >=0; there is no predefined upper limit of its values.
- $Av(a)$ denotes the set of available that is not yet visited nodes for ant a.

The State Transition Rules selects an available (not yet visited node) using one of two algorithms: exploration or exploitation. The parameter *Q*0 specifies the probability of employing the exploitation.

The exploitation algorithm is a deterministic one and it selects a node which maximizes the route quality function *qf*:

$$qf(r,t) = \tau(r,t) * \eta(r,t)^{\beta} \tag{1}$$

Where *r* and *t* are the two nodes and β is a parameter. The *qf* function selects the best node exploiting the gained so far collective knowledge of the ACS. The influence of the distance depends of the parameter I. It is greater than 1. As the direct distance between nodes is always <=1 then increasing β gives more prominence to pheromone level.

The exploration on the other hand is non deterministic process. It selects t – the next node with the probabilities defined by the Formula 2:

$$pr(r,t) = \frac{qf(r,t)}{\sum_{u \in A(r)} qf(r,u)} \tag{2}$$

The exploration algorithm prefers to choose nodes with high values of the qf function but it could select any other available node. Exploration is used to search for alternative solutions and mitigate the danger of being trapped in a local minimum.

Both local and global updating rules control the amount of pheromone deposited on the route from *s* to *t*. Let *GB* denote the global (found by any ant) best path found so far and *L*(*GB*) its length. The local updating function is evoked after each movement of an ant and it changes the amount of pheromone in the following way:

$$\tau(r,s) = (1-\rho) * \tau(r,s) + \rho * \Delta(r,s) \tag{3}$$

Where:

$$\Delta(r,s) = \begin{cases} \dfrac{(L(GB)^{-1} \quad if(r,s) \in global-best-tour}{0 \quad otherwize} \end{cases} \tag{4}$$

If an ant takes a route that does not belong to *GB* then its pheromone level decreases, what mimics the process of pheromone evaporation. Otherwise its level dos not change or increases. The evaporating intensity is controlled by the parameter *ρ*.

The global updating function is evoked after each iteration, that is whenever all ants have completed their search. It changes the pheromone level on all routs in the graph. The Formula 5 specifies the resulting level:

$$\tau(r,s) = \quad (1-\alpha) * \tau(r,s) + \alpha * \Delta(r,s) \tag{5}$$

The Formula 5 is very much like the Formula 3, the only difference is that it evaporation intensity is controlled by still another parameter *α*.

The initial values of the pheromone level for all paths (*τ*0) of a network with n nodes is not parameterized and is calculated using the Formula 6:

$$\tau_0 = \frac{2n}{\sum \delta(r,s)} \tag{6}$$

Another not optimized parameter is the maximal number of iterations. It was set to 300. Performed tests have shown that increasing the value is not justified as it does not lead to the decreasing of the average value for the best route.

Despite its simplicity of each individual ant, the ACS is capable of efficiently solving NP-hard tasks such as the TSP. This is due to the pheromone trails which represents the collective search experience of all ants.

2.2. SIMULATED ANNEALING

Simulated annealing was independently described by S. Kirkpatrick at al. in 1983 [8] and by V Černý in 1985 [1]. It is often used for discrete and huge search spaces. Each point of space represents a possible solution. The usability function *Uf*(*p*) defines the quality of solutions represented by point p. As the name suggests inspiration of this metaheuristics comes from the metallurgical process of annealing. As in real life annealing process the temperature play an important role. At start of the process a single particle is put an randomly selected a point in the search space. The temperature is set to a high value (see Section 3.3). The particle attempts then to move to another point. When the temperature is high a particle has much freedom and moves in an almost random mode. As it lowers the freedom of movement is more and more restricted.

Let p and r denote respectively the current ant the next prospective point in search space. It *Uf*(*r*) > = *Uf*(*p*) then the movement always takes place. In the opposite case the probability of moving to r is defined by the Formula 7.

$$p(p,r) = \left\{ \frac{1: \quad Uf(r) >= Uf(p)}{\exp(\frac{(Uf(r) - Uf(p))}{T}) \quad : \quad otherwize} \right\} \tag{7}$$

Where T is the temperature. If it is high enough the particle has a good chance of moving to a "worse" point. This helps the algorithm to avoid being taped in a local minima. As usual the best so far solution is memorized. The process of moving from one point (solution) to the next one continues till the temperature drops to 0 and then the values of the *Uf* function does not decrease.

To implement the SA mataheuristics the following elements must be specified:

- *S* – a space of all possible solutions;
- Gen(*s*) – a function that for a starting point s generates next possible element of *S*.
- *Uf*(*s*) – a function that specifies a value of the element $s \in S$
- Anneal – an annealing schema that specifies the initial temperature and the rules of lowering it.

The selection of an annealing schema ant other implementation details are discussed more deeply in the Section 3.1.

2.3. EVOLUTIONARY PROGRAMMING

Evolutionary programming is one of the oldest metaheuristics. It was first introduced in the early 1960's by L. Fogel [6]. A solid works summarizing the past and present development in the field can be found in [7]. The aim of the approach is to exploit the mechanisms of evolution to optimize complex tasks. Now it is regarded as member of a broader field of evolutionary algorithm paradigm. Common to all of approaches in the field is the use of a (large) population of possible solutions that evolve concurrently. A solution is represented by a sequence of genes, each gene being assigned to a single property. Such a sequence defies a specimen. A fitness function defines the quality of the solution represented by a specimen. The evolutionary principle "survival of the fittest" controls the chances of a specimen to stay in life or to evolve. The principle is implemented e.g. by a simple yet effective roulette algorithm. It picks the specimens for mutation or crossover with probabilities proportional to their usefulness.

The process goes on until a candidate solution with sufficient quality is found or a previously defined computational limit is reached.

What differs the Evolutionary Programming from other similar approaches is that to generate a new solution the mutation of genes in a single population member is used. The other Evolutionary Algorithms rely heavily on the crossover of genes of two population members. The mutation plays a marginal role or is eliminated altogether.

Each gene represents one of the parameters of the JACSF algorithm described in Section 3.1. Their values are floating point numbers and using the crossover is not sensible and in some cases may lead to generating not valid numbers. The remedy is to digitize the values but a far more natural way to obtain new values is to work directly with floating point numbers. The Simulated Annealing offers a proven method for generating new parameter values

3. PUTTING IT ALL TOGETHER

Optimizing of the parameters for JACSF proceeds as follows. A specimen is defined by a sequence of genes. Each of the genes describes one parameter of the JACSF implementation, see the Section 3.1. Let TspAl(p) denote the minimal average length found by JACSF implementation that is controlled by set of parameters p. Algorithm 1 specifies the steps of the optimization process.

Algorithm 1.

1. Set initial conditions.
   - Assign 0 to generation number.
   - Generate a set of *N* specimens with random gene values gene.
   - Store the first generated specimen as *BS* – best so far specimen.
2. Calculate the value of TspAl function for each member of the generation i.
3. Find the *GB* – the best specimen in the generation *i*.
4. Store the *GB* in generation *i*+1.
5. If (TspAl(*GB*)<TspAl(*BS*)) then store *GB* as *BS*.
6. Generate *N*-1 elements in the following manner
   - Select a specimen from generation i using the roulette rule. The number of best specimens used for selection declines with lowering the temperature.
   - Modify the genes of the selected specimen using the Formula 8. As in pt. a the formula includes temperature.
   - Place the modified specimen in the i+1 generation.
7. Change the temperature according to the annealing schema.
8. It there in no change in the *GB* in two consecutive steps then STOP the processing.
9. Increment i.
10. Go to step 2.

The implementation details are described below.

3.1. ACS

The ACS has many variations as described e.g. in a recent paper [5]. In the study we have not tried to propose still another one. Instead of that the standard version of JACSF for TSP is used is used and the and the aim it to optimizing its parameters. Due to the complexity of the TSP problem and the JACSF itself there is no analytical solutions. Their values proposed in the literature are the result of several test.

The Table 1 recapitulates the description for of the JACSF Tsp parameters. It includes the both their suggested values and the tested range of values. The ranges were selected to encompass the suggested value and a broad range of possible, making sense values.

Table 1. ACO parameters description

Name	Description	Suggested Value	Tested Range
N	Number of Ants	20	10–50
$Q0$	Probability of selecting exploitation over exploration	0.8	0.10–0.99
α	Aging factor used in the global updating rule	0.1	0.01–0.5
β	Moderating factor for the cost measure function	2.0	1.0–4.0
ρ	Aging factor in the local updating rule	0.1	0.01–0.5

3.2. GENETIC ALGORITHM

The parameters of the TSP Run are represented by a sequence of genes. Each gene is characterized by three floating point values:

- min V – minimal allowed value and
- max V – maximal allowed value.
- *C*urr V – current gene value;

The first two values are taken from the Table 1.

Each gene supports the following operations:

- Randomizing – assigning a random value to currVal from within the range. The operation is performed at the start of work for each gene.
- Annealing – changing the value of a gene. The operation could increase, decrease or keep the old value and is controlled by the Formula 8:

$$cV_n = \begin{cases} cV_p \; if \quad Rand_1 < {}^1\!/_3 \\ (\max V - currV) * Rand_2 \qquad for\, 1/3 \le Rand_1 < 2/3 \\ (cV_i - cV_i) * Rand_2 \qquad otherwize \end{cases} \tag{8}$$

Where:

cVp and *cVn* are respectively the previous and next value a gene.

Rand$_1$ and Rand$_2$ are a randomly generated values in the range from 0.0 to 1.0. The Formula 8 guaranties that the new value will not trespass the range allowed for a gene and the scope of change decreases with lowering of temperature. In one third of all cases the value does not change.

3.3. SIMULATED ANNEALING

Standard SA procedure works with a single particle. The proposed solution differs from the standard one in three important points:

1. It is applied not to 1 but to M specimens representing parameters;
2. It is uses on two levels:
   - To select the number of specimens for reproduction
   - To influence the modification of value of individual genes.
3. All generated specimens or genes are accepted, the annealing is used just to gradually limit the range of possible changes their values.

Each SA algorithm starts with the selection of *T*0 the initial temperature. In the study it was set to 1.0. The consecutive values of Ti are calculated according to the Formula 9.

$$T_{i+1} = \alpha T_i \tag{9}$$

As it is suggested in the literature the value of α was set to 0.95. The temperature is used in the Step 8 of Algorithm 1. This is a simplified version of annealing mechanism it is meant as a reference solution during further studies.

4. VERIFICATION

The verify the approach series of experiments were performed. The optimization process was applied to two networks with 30 and 50 nodes. *M* – the number of specimens testes in each step was set to 20. Every one JACSF specimen optimization was performed by a separate process controlled by a its set of parameters. The maximal number of iterations had a fixed equal to 300.

A supervising Simulating Annealing process controlled the JACSF sub processes.

The Tables 2 and 3 show the values of average route best iteration number and the parameters values for of three best specimens for the networks with 30 and 50 nodes respectively. The first raw is devoted to the “optimal” set as it is defined in [2]. The path length of all three generated solutions is better than the reference path. It should be also stressed that in both cases the number of annealing steps was very small, it was equal to just 5.

The data clearly show how tricky the task of parameter setting is. It seems rather unlikely, that an analytical solution to that task could be found. The solution proposed in the work is capable of producing in an automated way that the proposed metahetristics is capable of obtaining parameters that are better then the standard solutions.

Table 2. Best specimens for a network of 30 nodes

Path length	Best Iter.	Ant No.	Alpha	Beta	*ρ*	*Q*0
2.2055	114	30	0.100	2.00	0.100	0.800
2.1338	31	37	0.410	1.64	0.092	0.478
2.1475	115	49	0.433	3.94	0.04	0.200
2.1755	16	38	0.359	3,79	0.167	0.676

Table 3. Best specimens for a network of 50 nodes

Path length	Best Iter.	Ant No.	Alpha	Beta	*ρ*	*Q*0
2.2055	36	30	0.100	2,00	0.100	0.800
2.1338	19	40	0.459	1.75	0.313	0.776
2.1475	45	40	0.433	2.95	0.127	0.776
2.1755	6	40	0.359	2.90	0.089	0.907

5. CONCLUSIONS AND FUTURE WORK

The JACSF algorithm is a carbon copy Chirico algorithm. The modifications that were introduced were solely for providing an interface to other parts of the software used in the experiment. This is a deliberate decision. The aim was to optimize the parameter setting of an existing algorithm not to find a new variants. he annealing mechanism is an adaptation of general SA principles to a specific problem. What differs it from conventional cases is the way it is applied. It is used on twice. On the first place it controls the whole process of parameter setting. It is also applied to change the genes values in all ants. It was hoped, that this feature would make the process more robust.

Both the simulated annealing and TSP are do not differ significantly from standard versions. The choice of Evolutionary Programming over other versions of Genetic Algorithms was due to the fact that the aim is to find optimal values of floating point numbers. The crossover mechanism works best with discrete values of genes. One can still argue that this does not block the evolution. The asexual reproduction has started it and it is present even now. The mutation mechanism is far more elaborate than usual.

While evaluating the TSP algorithm performance many factors could be taken into account. The two most obvious are the average route length and the computational effort necessary find a solution. The latter one could be estimated by the best iteration number. During the test the quality function took only the first criteria into account. This was solely driven by the desire to compare the experiments results with the results obtained by other researches. In future we plan to study more elaborate evaluation functions that would encompass the mentioned above factors and e.g. stability of results achieved in different environments.

Each metaheuristics is complex on its own. The interplay of several metarauristics is even more intricate. The work describes an early stage of study of their interplay. Far more experiments are necessary to obtain clues or insights into that complex problem.

REFERENCES

[1] BUSETTI F., *Simulated Annealing Overview*, Report, 2003.

[2] CHIRICO U., *A Java Framework for Ant Colony Systems*, Ants2004: Forth International Workshop on Ant Colony Optimization and Swarm Intelligence, Brussels, 2004.

[3] 3. ČERNY, V., *Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm*", Journal of Optimization Theory and Applications 45: 41–51., 1985.

[4] DORIGO M., *Optimization, Learning and Natural Algorithms*, PhD thesis, Politecnico di Milano, Italie, 1992.

[5] DORIGO M., STUETZLE T., *Ant Colony Optimization: Overview and Recent Advances*, IRIDIA – Technical Report Series, Technical Report No. TR/IRIDIA/2009-013, May 2009.

[6] FOGEL, L.J., OWENS, A.J., WALSH, M.J., *Artificial Intelligence through Simulated Evolution*, John Wiley, 1966.

[7] EIBEN, A.E., SMITH, J.E., *Introduction to Evolutionary Computing*, Springer, 2003.

[8] KIRKPATRIK, S., GELATT, C. D., VECCHI, M. P., *Optimization by Simulated Annealing*, Science 220 (4598): 671–680, 1983.

*Algorithms, meta-heuristic, optimization,
Vehicle Routing Problem, transport*

Krzysztof GOLONKA*, Leszek KOSZAŁKA*

BUS ROUTE OPTIMIZATION: EVALUATION OF ALGORITHMS

In this work, we analyze the school bus route optimization problem. It is a crucial social issue that concerns faster and more comfortable transport of students to their schools. Moreover, the route optimization allows to decrease the ticket price, i.e., to maximize the profit of the provider. Since the problem belongs to hard optimization problems, thus, we adapted four meta-heuristic algorithms: Tabu Search, Simulated Annealing, Genetic Algorithm, Complete Overview, and invented by the authors algorithm called Constructor, and additionally Bellman-Ford algorithm used as a helper. In order to measure the efficiency of the considered algorithms we create our own evaluating function called Balance and compare results given by algorithms to the maximum found with Complete Overview. Finally, we designed and implemented an experimentation system to test these algorithms on various problem instances, to emerge the most efficient one.

1. INTRODUCTION

Since banking crisis from 2008 many companies were forced to cut expenses and look for more savings. Moreover, nowadays a lot of pressure is put on being green – environmentally-friendly, especially when it comes to industry or transport. A lot of effort must be put in analysis and planning process to come across these challenges. This work, based on [6], focuses on optimization of a school bus route and proposes the direction of searching optimal solutions. The main goal is to find the most profitable route.

This optimization belongs to non-polynomial problem and has a huge solution space, meaning we cannot find the best solution in polynomial time. For small instances it is easy to search through the whole solution space but when instance begins to grow, the required time may become unacceptable. The only one reasonable way to

* Wroclaw University of Technology.

solve this is to use meta-heuristic algorithms that were invented to struggle with such problems. An idea to consider such algorithms based on artificial intelligence like Tabu Search (e.g. described in [4]), Simulated Annealing (e.g. explained in [8]) and Genetic Algorithm (e.g. illustrated in [2]) seems to be promising.

In addition, we tried to invent on our own a new algorithm and we created the algorithm called Constructor which is described in this work. To determine the shortest path from the starting point to the ending point through all possible bus routes we implemented Bellman-Ford Algorithm.

To assess algorithms' efficiency we implemented a simulation system that allows user to perform series of tests, returns the average values and presents them on plots.

The rest of work is organized as follows: Section 2 defines the problem to be solved. Section 3 describes the considered algorithms and their roles in optimization process. Section 4 shortly presents the implemented simulation system. The results of the research appear in Section 5. Last but not least Section 6 provides some remarks and conclusion.

2. PROBLEM STATEMENT

There is given a certain urban area (Fig. 1), that consists of links and Bus Stops (BSs). Lines represent links, numbers next to them represent their lengths and red circles symbolize Bus Stops. The beginning of the route is marked by a green rectangle, but ending point (blue rectangle) represents the location of the school. It is necessary to determine the most profitable constant for a certain time period route of a school bus to maximize profits of a bus provider. Once the route is planned the bus may omit some BSs which are not on this specified route.

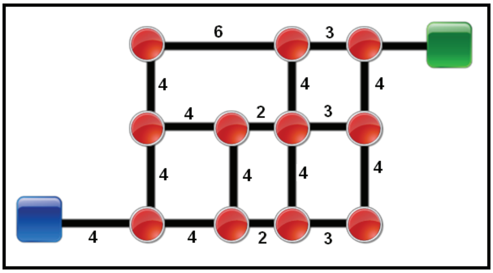


Fig. 1. An example of an instance

The basis for making decision on which BS should stop is the observation of statistics that deliver the number of students (pupils) waiting at a BS on a given time. These values are represented by a matrix, in which each row corresponds to the next hour in bus transit and the columns show the number of pupils on each BS (4).

2.1. BASIC TERMS

Good understanding of this problem requires an adequate mathematical description, similar to ones encountered in [1], [10]. This section provides such sufficient explanation.

Bus Stop (*BS*) is a point on a map where pupils wait for a bus to school. Each BS has its coordinates *x* and *y* on a map.

$$BS = [x, y] \tag{1}$$

List of Potential *BSs* (*LBS*) is a collection of all *BSs* on a map, where *LBS*(*N*) may be defined by (2).

$$LBS(N) = \begin{bmatrix} BS_1 \\ BS_2 \\ \vdots \\ BS_N \end{bmatrix} = \begin{bmatrix} x_1, y_1 \\ x_2, y_2 \\ \vdots \\ x_N, y_N \end{bmatrix} \tag{2}$$

where:

N – number of *BSs* in *LBS*

Link (*L*) is an array defined by (3) that describes lengths of links between BS_i and BS_j. Some *BSs* may not be directly linked to others but each *BS* must have at least one link.

$$L(N, N) = \begin{bmatrix} l_{1,1} & \cdots & l_{1,j} \\ \vdots & \ddots & \vdots \\ l_{N,1} & \cdots & l_{N,N} \end{bmatrix} \tag{3}$$

where:

li,j = *lj,i*

Pupils Statistics (*P*) is an array that provides the number of pupils waiting for a bus on each BS for a Bus on specified time.

$$P(K, N) = \begin{bmatrix} p_{1,1} & \cdots & p_{1,N} \\ \vdots & \ddots & \vdots \\ p_{K,1} & \cdots & p_{K,N} \end{bmatrix} \tag{4}$$

where:

K – number of transits per day

Route (*R*) is a path consisting starting point, ending point and going through *BSs* chosen from *LBS*. A Route may contain smaller amount of *BSs* than *LBS*.

$$R(N,N) = \begin{bmatrix} r_{1,1} & \cdots & r_{1,N} \\ \vdots & \ddots & \vdots \\ r_{N,1} & \cdots & r_{N,N} \end{bmatrix} \tag{5}$$

where:

$$ri, j= \begin{cases} 1 \text{ if } BSi \text{ is on a route } R \text{ directly before } BSj,\ i, j \in \{1, 2, 3, \ldots, N\} \\ 0 \text{ otherwise} \end{cases}$$

Ticket Price (*T*) informs how much each pupil must pay for taking a bus.

Driver's Cost (*DC*) equals money paid to a driver for driving one unit of a length of a Route.

Bus Exploitation Cost (*BC*) equals expenses for fuel used by a bus after driving one unit of a length of a Route.

2.2. EVALUATING FUNCTION

The evaluating function introduced by us is called the Balance (6) interpreted as a daily balance – obtained after one day of work. If *Q(R)* is less than 0 the provider gets such loses, if greater than 0 the provider gets profits.

$$Q(R) = \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{K} p_{k,i} \cdot r_{i,j} \cdot T - (DC + BC) \cdot K \cdot \sum_{i=1}^{N}\sum_{j=1}^{N} r_{i,j} \cdot s_{i,j} \tag{6}$$

where:

T – ticket price

N – number of *BSs* in *LBS*

K – number of transits per one day

Moreover, to make sure that the bus will not go from starting point right to the destination point, we introduce a constraint. The constraint describes the minimal percentage number of all pupils from the statistics that should be delivered to the school (7).

$$100\ \% \cdot \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{K} p_{k,i} \cdot r_{i,j}}{\sum_{i=1}^{N} \sum_{k=1}^{K} p_{k,i}} \geq P_{\%}(R) \tag{7}$$

3. ALGORITHMS

3.1. BASIC IDEAS

For experiment purposes we implemented four meta-heuristic algorithms as the main algorithms, including three known algorithms (but specially adopted) : TS – Tabu Search, SA – Simulated Annealing, GA – Genetic Algorithm, and the Constructor (originally proposed by the authors of the work). The Route Length is calculated by us using Bellman-Ford algorithm.

3.2. SIMULATED ANNEALING

In each iteration the solution is replaced by a new one randomly chosen from the neighbourhood if the new one is better. If the new solution is worse it has (100–π)% of chances to replace the previous one. In this particular implementation we do not have such thing as temperature that changes the probability of replacing solutions. Here probability is constant. Moreover, SA has parameter called *%precision* that indicates the minimal percentage difference value of evaluation function between two solutions. These are the only two differences between our SA and the one described in [9].

3.3. TABU SEARCH

Tabu Search is more complex algorithm than SA because it is searching through the whole neighbourhood of a solution and choosing the best one unlike SA. Moreover TS is more resistant to loops thanks to the taboo list. The best found solution may replace the previous one only if it differs from all the records in a taboo list more than a certain percentage value. The length of the taboo list is limited and when it is full, the old records are overwritten.

3.4. GENETIC ALGORITHM

This algorithm is based on evolutionary mechanisms. The main idea is to create a population of a constant size and observe its evolution meanwhile registering the best ones. DNA chain is represented in this situation as a single solution. All the solutions have a chance to hand over gens but the higher the Balance function value of the solution, the higher possibility of being picked as a parent. As the population size is constant, the newborns must replace the old solutions regardless of their breeding history.

3.5. CONSTRUCTOR

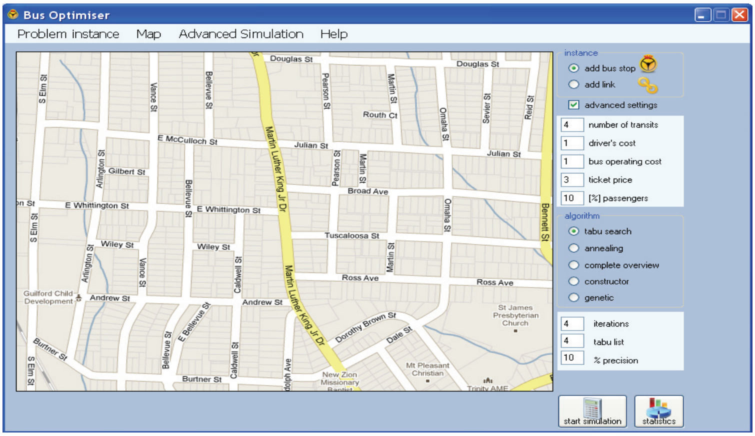
The Constructor was invented by us – an inspiration was the idea of searching through a whole neighbourhood while one iteration as in TS. Moreover, we assumed that splitting a big instance to smaller ones, solving them separately and joining all together may come up with quite good results.

At the beginning the Constructor splits the whole instance to smaller instances. The next step is to search through a neighbourhood of each small instance and modify them. After this operation, the algorithm combines in pairs small instances making them bigger. These operations last until the joining gives back the initial instance.

4. SIMSULATION SYSTEM

The application was designed, mainly in order to visualize the tested algorithms. It was created using Visual Studio 2008. The implementation language was C#. Figure 2 shows a screenshot of the application window.

At the beginning the user defines an instance of problem, next selects the considered algorithm and fixes its parameters. After clicking on “start simulation” button the application is searching for an optimal solution.



instance
● add bus stop
○ add link
☑ advanced settings

4 number of transits
1 driver's cost
1 bus operating cost
3 ticket price
10 [%] passengers

algorithm
● tabu search
○ annealing
○ complete overview
○ constructor
○ genetic

4 iterations
4 tabu list
10 % precision

Fig. 2. Application window

5. RESEARCH

At the beginning we had to determine appropriate parameters of problem instance which would give reliable results and allow us to use CO in order to be able to observe exact differences between used algorithm and the optimal solution. The second step was to find inner parameters of each algorithm such that the performance of all algorithms would be as good as possible. The last but not least phase, was to compare all calibrated algorithms and emerge one that gives results the closest to the optimal solution. The performance indicator for algorithms is called the average inaccuracy (8).

$$\Delta A = \left| \frac{Q_{CO} - Q_A}{Q_{CO}} \right| \cdot 100\% \qquad (8)$$

where:

Q_{CO} – solution calculated using CO

Q_A – solution calculated using selected Algorithm (TS, SA, GA, Constructor).

5.1. INSTANCE PARAMETERS, ALGORITHMS' PARAMETERS

These parameters were defined after numerous simulations. Main goal while setting instance parameters was to determine ones, that would give authoritative and objective results. Algorithms' parameters were set such that the algorithms would give results as good as possible. Instance parameters are presented in Table 1, algorithms' parameters are shown in Table 2.

Table 1. Instance parameters

parameter	|LBS|	instances to test	test iterations	K	DC	BC	T	P% [%]
value	18	40	10	4	1	1	1	50

Table 2. Inner parameters of algorithms

SA		**TS**		**GA**		**Constructor**	
parameter	**value**	**parameter**	**value**	**parameter**	**value**	**parameter**	**value**
iter	500	iter	500	iter	500	temp_l	2
π	60	tabu_l	6	Pop_l	256	repeat	1
prec	1	prec	1	par_l	192	repeat_p	3

5.2. COMPARISON – ALL ALGORITHMS

The next part of research was to compare to CO all four other algorithms with the best inner parameters. Instances of parameters were the same as in previous part

(Table 1) apart from minimal percentage passengers served (% passengers) which is variable in this test.

According to Fig. 3 the smallest inaccuracy was found for TS – from 10% to less than 1% for 90% passengers. The second efficiency takes GA – from 0% to less than 20%. Third was Constructor and the last place for SA. SA presents the biggest inaccuracy for 10% passengers (almost 35% inaccuracy) but its performance improves when constraint becomes more strict – 90% passengers.

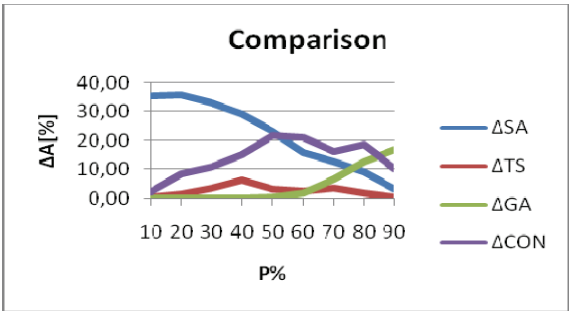


Fig. 3. The average inaccuracy (1)

This test presented the performance of algorithms according to inner parameters set in Section 5.1 but one aspect was not taken into account: number of solutions visited by each algorithm during the calculation. This number is equal (9) for SA, (10) for TS, (11) for GA, and (12) for GA.

$$Z_{SA} = iter \tag{9}$$

$$Z_{TS} = iter \cdot (|LBS| - 2) \tag{10}$$

$$Z_{GA} = iter \cdot Pop\_l \tag{11}$$

$$Z_{CON} = |LBS| \cdot repeat \cdot (1 + repeat\_p) \cdot (|LBS| : temp\_l) \tag{12}$$

Table 3. Number of visited solutions

size	Z_{SA}	Z_{TS}	Z_{GA}	Z_{CON}
65536	500	8000	128000	648

According to equations (9), (10), (11), (12) Table 3 presents the number of solutions checked by algorithms and size is actual size of a solution space.

This information forces to make another tests, this time setting such iterations parameter that all the algorithms would search through comparable amount of point in solution space as TS. The iterations parameter for algorithms are presented in Table 4 and the results are shown in Fig. 4.

Table 4. Iter parameter

	SA	TS	GA
iter	8000	500	32

This test brings us closer to answer for the question which algorithm is the most efficient. There is no difference in Constructor's results because there was no change in iteration number (this algorithm lacks such parameter). GA is below expectations which we could have had after previous test. SA and TS give similar results and there is needed one more comparison, but this time only TS and SA will be compared.

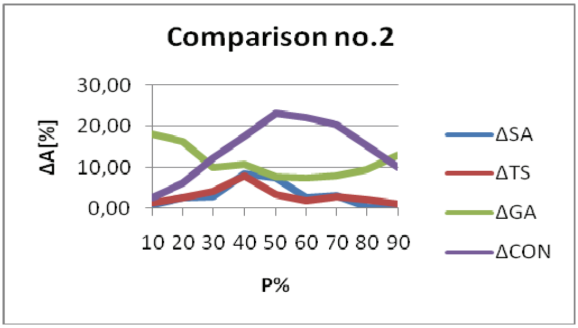


Fig. 4. The average inaccuracy (2)

5.3. TS AND SA COMPARISON

This part is supposed to emerge the algorithm which is capable of finding solutions closer to optimum searching through less points in solution space.

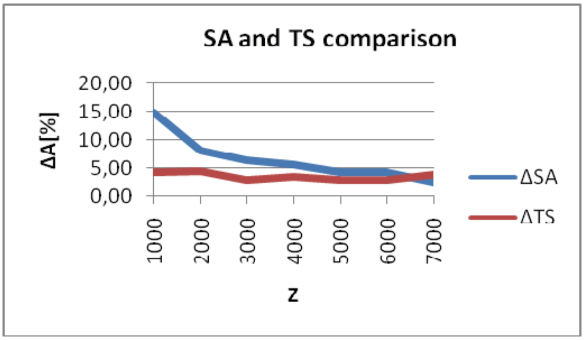


Fig. 5. Comparison of alghoritms: TS vs SA

Results are presented on Fig. 5. This test justifies an observation that the number of iterations has significant impact on the obtained results. The more iterations, the better results is given by the algorithm. The main observation is that TS gives better results than SA regardless of the number of iterations, and the results are more constant and stable (inaccuracy always below 5%).

6. CONCLUSION

To sum up, series of performed experiments revealed that, from all implemented meta-heuristic algorithms – Tabu Search, Simulated Annealing, Genetic Algorith and Constructor – the two first give the best results. The final simulation emerged the most efficient algorithm: Tabu Search.

Genetic Algorithm in this version appeared to be the worst. In comparison to result of TS it was far below expectations. GA needed to search through too much point in solution space. Constructor turned out to be not as bad as we anticipated – inaccuracy never reached 30% even for the most difficult constraints. Unfortunately this result cannot be improved in any other way but introducing somehow the iterations number parameter and further work on this algorithm should be focused on this aspect.

The main goal for solving school bus problem is to provide an opportunity to every single pupil to reach school on time. According to this statement and research presented in this work, the proposed and recommended algorithm for route planning is Tabu Search (TS).

REFERENCES

[1] BOWERMAN R., HALL B., CALAMAI P., *A Multi-Objective Optimization Approach to Urban School Bus Routing: Formulation and Solution Method*, Transportation Research Part A: Policy and Practice, Vol. 29, No. Issue 2, pp. 107–123, 1995.

[2] DAVIES L.D., *Genetic Algorithms and Simulated Annealing*, Morgan Kaufmann Publ, 1987.

[3] GENDREAU M., *An Introduction to Tabu Search*, Universite de Montreal, 2003.

[4] GLOVER F., Tabu Search – part I. *ORSA Journal on Computing*, Vol. 1, no. 3, 1997.

[5] GLOVER F., KOCHENBERGER, G.A., *Handbook of Metaheuristics*, Springer, Heidelberg, New York, 2002.

[6] GOLONKA K., KOSZAŁKA L., POŹNIAK-KOSZAŁKA I., KASPRZAK A., *An Experimentation System for Bus Route Planning and Testing Metaheuristics Algorithms*, ICONS IARIA Proceedings, St. Maarten (Netherlands), 2011.

[7] GRANVILLE V., KRIVANEK M., RASSON J.P., Simulated Annealing: a proof of convergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, pp. 652–656, 1994.

[8] KIRKPATRICK S., GELATT C.D., VECCHI M.P., Optimization by Simulated Annealing. *Science, New Series*, Vol. 220, pp. 671–680, 1983.

[9] LAARHOVEN J.M., EMILE H., AARTS L., *Simulated Annealing: Theory and Applications*, Springer, Berlin, 1987.

[10] LYO Li, Z FU, *The school bus routing problem: case study*, Journal of The Operational Research Society, Vol. 53, pp. 552–558, 2002.

PART 5

COMPLEX OF OPERATION SYSTEMS CONTROL

Cyclic behavior, cyclic scheduling,
constraint satisfaction problem, declarative modeling

Grzegorz BOCEWICZ*, Robert WÓJCIK**, Zbigniew BANASZAK***

A REFERENCE MODEL OF AGVS CYCLIC SCHEDULING PROBLEMS: DECLARATIVE APPROACH

In this work, we deal with the cyclic scheduling problem usually observed in the FMS producing multi-type parts where the AGVS plays a role of a material handling system. Finding the conditions guaranteeing the AGVs deadlock-free and collision-free movement policy is the aim of this work. The AGVs co-sharing the common parts of the transportation route while executing repetitive processes, i.e. being assigned to AGVs passing along machines in a cyclic way, can be modeled in terms of Cyclic Concurrent Process Systems (CCPS) [13]. Schedulability analysis for a given CCPS answers the question whether a cyclic schedule exists in the reachability space of the modeled AGVS system or not. The work suggests a novel approach for schedulability analysis employing the declarative modeling. A reference model of constraint satisfaction cyclic scheduling problem shows that unschedulability can be caused by a relation among an initial state and dispatching rules selected. The sufficient conditions guaranteeing CCPS schedulability are discussed and the recursive approach to their designing is proposed.

1. INTRODUCTION

The Flexible Manufacturing System (FMS) [9], [12] produces multi-type parts, in which the Automated Guided Vehicle System (AGVS) is used as a material handling system. The Automated Guided Vehicles (AGVs) scheduling problem is a special case

* Koszalin University of Technology, Department of Computer Science and Management, Śniadeckich 2, 75-453 Koszalin, Poland, bocewicz@ie.tu.koszalin.pl

** Wrocław University of Technology, Institute of Computer Engineering, Control and Robotics, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland, robert.wojcik@pwr.wroc.pl

*** Warsaw University of Technology, Department of Business Informatics, Narbutta 85, 02-524 Warsaw, Poland, z.banaszak@wz.pw.edu.pl

of the cyclic blocking flow-shop one, where the jobs might block either the machine or the AGV at the processing time. A cyclic schedule [11], [17] is one in which the same sequence of states is repeated over and over again. A cycle in such a schedule begins at any state and ends when that state is encountered next. Finding the conditions guaranteeing the AGVs deadlock-free and collision-free movement policy is the aim of this work.

The AGVs co-sharing the parts of the transportation route and executing repetitive tasks assigned to the vehicles passing along machine tools in a cyclic way, can be modeled in terms of Cyclic Concurrent Process Systems (CCPS) [2], [3], [4], [13]. The existing constraints connected with the available traveling route width (not allowing for vehicle passing by), the topology of travelling routes and itineraries of individual vehicles, lack of simultaneous access to the stations, etc. imply the necessity to investigate conditions leading to possible vehicle collisions and deadlocks [7], [16]. This means that the cyclic scheduling, i.e. guaranteeing the AGVs, collision-free and deadlock-free schedules problem belongs to the class of NP-hard problems [10].

The existing approach to solving the problem base usually upon the simulation models, e.g. the Petri nets [1], [16], the algebraic models, e.g. upon the (max,+) algebra [13] or the artificial intelligent methods e.g. upon the genetic algorithms [6]. In this context, this work constitutes some continuation of the investigations conducted in [2], [3], [4], [13].

Moreover, since the cyclic scheduling of CCPS can be seen as a kind of Diophantine problem [5], hence its solvability, i.e. schedulability, plays a pivotal role [8], [15]. Schedulability analysis for a given (CCPS) answers the question whether a cyclic schedule exists in the reachability space of this system or not. Therefore the problem considered in this work reduces itself to determination of the sufficient conditions ensuring the collision-free and deadlock-free execution of the concurrent cyclic processes. In that context, we suggest a novel approach for schedulability analysis employing the declarative modeling as well as the recursive approach to sufficient conditions designing.

The work is organized as follows. In the next section, the reference model of the AGVS scheduling problem is defined. In the third section, the schedulability of AGVs is discussed. Sufficient conditions for CCPS deadlock avoidance are proposed. Following that, the cyclic steady states space and a way of precedence digraphs designing are described. In the fourth section, an illustrative example of AGVS redeveloping is shown.

2. A REFERENCE MODEL

2.1 AGVS MODEL

The AGVs can be modeled in terms of a CCPS, wherein the cyclic processes (vehicles) are interconnected one with another by use of the AGVS common resources. In Fig. 1, there is presented an illustration of the FMS with distinguished AGVS Fig. 1 a), and its CCPS model Fig. 1 b).

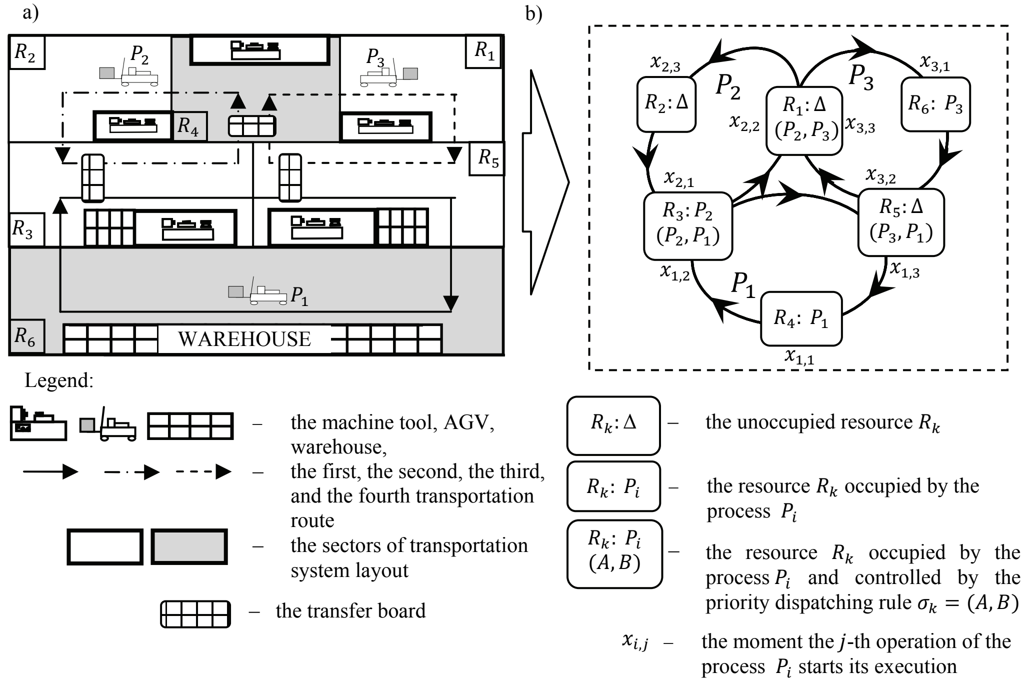

a)
b)
R_2
P_2
P_3
R_1
R_4
R_5
R_3
P_1
R_6
WAREHOUSE
$x_{2,3}$
P_2
P_3
$x_{3,1}$
$R_2\colon\Delta$
$R_1\colon\Delta$ (P_2, P_3)
$R_6\colon P_3$
$x_{2,2}$
$x_{3,3}$
$x_{2,1}$
$x_{3,2}$
$R_3\colon P_2$ (P_2, P_1)
$R_5\colon\Delta$ (P_3, P_1)
$x_{1,2}$
P_1
$x_{1,3}$
$R_4\colon P_1$
$x_{1,1}$
Legend:
– the machine tool, AGV, warehouse,
– the first, the second, the third, and the fourth transportation route
– the sectors of transportation system layout
– the transfer board
$R_k\colon\Delta$ – the unoccupied resource R_k
$R_k\colon P_i$ – the resource R_k occupied by the process P_i
$R_k\colon P_i$ (A, B) – the resource R_k occupied by the process P_i and controlled by the priority dispatching rule $\sigma_k = (A, B)$
$x_{i,j}$ – the moment the j-th operation of the process P_i starts its execution

Fig. 1. An example of the FMS: AGVS a), CCPS model of AGVS b)

Three processes are considered P_1, P_2, and P_3, that reflect operation of individual vehicles. The transportation route consists of six parts, treated as resources R_1– R_6 the vehicles have to pass by. For the systems of that type, it is assumed that the cooperation of the processes is determined by the following constraints [13]:

- the processes share the common resources in the mutual exclusion mode,
- commencement of a successive process operation happens immediately after completing of the current operation provided that there is a possibility of making use of the successive resource requested by the given process,
- during waiting for a busy resource, the process does not release the resource allocated for execution of the previous operation,

- the process in not pre-emptive, i.e. the resource may not be taken of the process while it is using it,
- the processes are executed cyclically,
- in one cycle, a process may pass via any resource once only.

In the model of CCPS the following definitions are used [4], [13]:

- A sequence $p_i = (p_{i,1}, p_{i,2}, \ldots, p_{i,lr(i)})$ specify the route of the process P_i, its components define the numbers of the resources used in course of process operations execution, where: $p_{i,j} \in R$ (set of resources: $R = \{R_1, R_2, \ldots, R_m\}$), in the rest of the work the j-th operation executed on the resource $p_{i,j}$ along the process P_i is denoted as follows $o_{i,j}$; $lr(i)$ – denotes a length of cyclic process route.
- $x_{i,j}(k) \in \mathbb{N}$ the moment operation $o_{i,j}$ starts its execution in the k-th cycle.
- $T_i = (t_{i,1}, t_{i,2}, \ldots, t_{i,lr(i)})$ specifies the operations time in the i-th process, where $t_{i,j}$ denotes the time of execution of the j-th operation by the i-the process. For the sake of simplicity let as assume the all operation times are the same equal to the 1 unit of time.
- $\Theta = \{\sigma_1, \sigma_2, \ldots, \sigma_m\}$ the set of the priority dispatching rules, where $\sigma_i = (s_{i,1}, \ldots, s_{i,lp(i)})$ – is the sequence, the components of which determine the order process can execute on the i-th resource, $s_{i,j} \in P$ (set of processes: $P = \{P_1, P_2, \ldots, P_n\}$).

In that context a CCPS can be defined as the following quadruple [4]:

$$SC = (R, \Pi, T, \Theta) \tag{1}$$

where:

$R = \{R_1, R_2, \ldots, R_m\}$ – the set of resources,
$\Pi = \{p_1, p_2, \ldots, p_n\}$ – the set of process routes,
$T = \{T_1, \ldots, T_n\}$ – the set of process operations times,
$\Theta = \{\sigma_1, \sigma_2, \ldots, \sigma_m\}$ – the set of dispatching priority rules.

The model considered (1) specifies parameters characterizing the CCPS's structure. In that context the main question concerns of CCPS cyclic behavior and a way this behavior depends on direction of local process routes Π as well as on priority rules Θ, and a set of initial states, i.e. initial processes allocation to the system resources. Assuming such a cyclic steady state the next question regarding of its periodicity evaluation plays a pivotal role.

2.2. CSP-DRIVEN CYCLIC SCHEDULING

Since parameters describing the CCPS are usually discrete, and linking them relations can be seen as constraints, hence related to them cyclic scheduling problems can be presented in the form of the Constraint Satisfaction Problem (CSP) [2], [14]. More formally, CSP is a framework for solving combinatorial problems specified by pairs:

(a set of variables and associated domains, a set of constraints restricting the possible combinations of the variable values). In this context, the CSP is defined as follows:

$$CS = ((V, D), C) \tag{2}$$

where:

$V = \{v_1, v_2, \dots, v_{lv}\}$ – a finite set of discrete decision variables,

$D = \{D_i | D_i = \{d_{i,1}, d_{i,2}, \dots, d_{i,ld}\}, i = 1, \dots, lv\}$ – a family of finite domains,

$C = \{C_i | i = 1, \dots, l\}$ – a finite set of constraints limiting the domains of variables.

The solution to the CS is a vector $\left(d_{1,i}, d_{2,k}, \dots, d_{n,j}\right)$ coordinates of which satisfy each constraint of the set C. The inference engine consists of the following two components: constraint propagation and variable distribution [14].

In the considered case of CCPS the relevant CSP can be stated as follows [14]:

$$CS = ((\{R, \Pi, T, \Theta, X, Tc\}, \{D_R, D_\Pi, D_T, D_\Theta, D_X, D_{Tc}\}), C) \tag{3}$$

where:

- $R, \Pi, T, \Theta$are the decision variables see (1), and $X = \{X_1, X_2, \dots, X_n\}$ is the set of sequences of $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,lr(i)})$, and Tc is the CCPS periodicity. Each variable $x_{i,\mathrm{j}}$ enables to determine the moment of $o_{i,j}$ beginning in any (the k-th) cycle: $x_{i,\mathrm{j}}(k) = x_{i,\mathrm{j}} + k \cdot Tc$.
- the following domains of decision variables are considered:
  D_R– the set of resources,
  D_Π–the family of sets of admissible routings,
  D_T–the family of sets of admissible operation times,
  D_Θ– the family of sets of admissible dispatching priority rules,
  D_X–the family of sets of admissible coordinate values X_i, $x_{i,\mathrm{j}} \in \mathbb{Z}$,
  D_{Tc}– the sets of admissible values of variables Tc,
- constraints are specified as follows:

$$x_{i,1} = max\{\left(x_{i,lr(i)} + t_{i,lr(i)} - Tc\right), \left(x_{\alpha(i,1)} + 1 + \beta(i,1)\right)\}, i = 1, .., n;$$
$$x_{i,j} = max\{\left(x_{i,(j-1)} + t_{i,(j-1)}\right), \left(x_{\alpha(i,j)} + 1 + \beta(i,1)\right)\}, i = 1, .., n;\ j = 2, \dots, lr(i);$$

where:

$$\alpha(i,j) = \begin{cases} (i, j-1) \text{ if } o_{i,j} \text{ executes on unshared resource} \\ (a, b+1) \text{ if } o_{a,b} \text{ executes on shared resource} \\ \qquad\qquad\qquad \text{before } o_{i,j} \end{cases}$$

$$\beta(i,j) = \begin{cases} 0 \text{ if } o_{\alpha(i,j)} \text{executes in the same cycle as } o_{(i,j)} \\ -Tc \text{ if } o_{\alpha(i,j)} \text{ has been executed in the cycle preceding the cycle of } o_{(i,j)} \text{ execution} \\ Tc \text{ if } o_{\alpha(i,j)} \text{ is executed in the cycle cuceeding the cycle of } o_{(i,j)} \text{ execution} \end{cases}$$

Solution to the problem (2) aimed at determination of X sequences' values guaranteeing CCPS cyclic behaviour while following the set of constraints C is of primary importance. The conditions sufficient for collision-free and deadlock-free processes execution (deadlock-freeness for any k-th cycle) are presented in [3].

Because the CCPS model consists of a set of local cyclic processes, the considered CS problem can be stated in a reference structure shown in Fig. 2. The reference model allows to consider the CCPS designing process as recurrent designing of constraints linking elementary problems CS_1, CS_2 and CS_3.

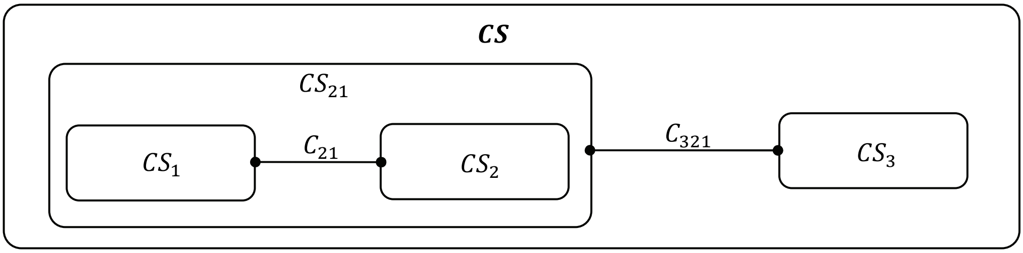


Fig. 2. The CSP reference model following CCPS from Fig. 1

2.3. PROBLEM STATEMENT

Consider a CCPS specified by a given set of dispatching rules and initial processes allocation. The main question concerns of CCPS periodicity. In case it behaves periodically the next question regards of the CCPS's period. Other questions regard of admissible initial processes allocation (i.e. the possible AGV dockings), the dispatching rules guaranteeing a given CCPS periodicity while preserving assumed frequency of local processes execution within a global period Tc. In general, besides of straight problem formulation the reverse ones can be considered, e.g. Does there exist the CCPS's structure such that an assumed steady cyclic state can be achieved?

3. AGVS SCHEDULABILITY

3.1. STATE SPACE

Consider the k-th state S^k (4) composed of the sequence of processes allocation A^k, the sequence of semaphores (encompassing the rights of process's access to a resource) Z^k, and the sequence of semaphore indices Q^k:

$$S^k = (A^k, Z^k, Q^k) \tag{4}$$

where:

- $A^k = \left(a_1{}^k, a_2{}^k, \dots, a_m{}^k\right)$ – the processes allocation (m – a number of CCPS's resources). $a_i{}^k \in P \cup \{\Delta\}$ means the process allotted to the i-th resource R_i in the k-th state, $a_i{}^k = P_g$ means, the i-th resource R_i is occupied by the process P_g, and $a_i{}^k = \Delta$ – the i–th resource R_i is unoccupied; $P = \{P_1, P_2, \dots, P_n\}$ – the set of processes. In the case considered (see Fig. 1) the processes allocation is specified by the sequence: $A^0 = (P_3, \Delta, P_2, \Delta, \Delta, P_1)$.
- $Z^k = \left(z_1{}^k, z_2{}^k, \dots, z_m{}^k\right)$ – the sequence of semaphores corresponding to the k-th state, where $z_i{}^k \in P$ means the name of the process (specified in the i-th dispatching rule σ_i, allocated to the i-th resource) allowed to occupy the i-th resource R_i. For instance $z_i{}^k = P_g$ means that at the moment the process P_g is allowed to occupy the i-th resource R_i. For the CCPS from Fig. 1 the sequence of semaphores has the following form: $Z^0 = (P_3, P_2, P_2, P_2, P_3, P_1)$.
- $Q^k = \left(q_1{}^k, q_2{}^k, \dots, q_m{}^k\right)$ – the sequence of semaphore indices corresponding to the k-th state, where $q_i{}^k$ means the position of the semaphore $z_i{}^k$ in the priority dispatching rule σ_i: $z_i{}^k = crd_{(q_i{}^k)}\sigma_i$, $q_i{}^k \in \mathbb{N}$ ($crd_i D = d_i$, for $D = (d_1, \dots, d_i, \dots, d_w)$. For instance, $q_2{}^k = 2$ means the value of semaphore $z_2{}^k$ which is placed at the 2nd position in the priority dispatching rule σ_2. For the CCPS from Fig. 1 the sequence of semaphores Z has the following form: $Q^0 = (1,1,1,1,1,1)$.

The state $\boldsymbol{S^k}$ is feasible only if for any of its co-ordinate $a_i{}^k$ the following conditions hold:

$$\forall_{i\in\{1,2,\dots,n\}} \exists!_{j\in\{1,2,\dots,m\}} \left(P_i = crd_j A^k\right) \tag{5}$$

$$\forall_{i\in\{1,2,\dots,m\}} \left(crd_i\, A^k \in P \cup \{\Delta\}\right). \tag{6}$$

The set of all feasible states is called a state space $\mathbb{S}$, i.e., $S^k \in \mathbb{S}$. Consider two feasible states S^k and S^l:

$$S^k = (\left(a_1{}^k, a_2{}^k, \dots, a_m{}^k\right), \left(z_1{}^k, z_2{}^k, \dots, z_m{}^k\right), \left(q_1{}^k, q_2{}^k, \dots, q_m{}^k\right)), \tag{7}$$

$$S^l = ((a_1{}^l, a_2{}^l, \dots, a_m{}^l), (z_1{}^l, z_2{}^l, \dots, z_m{}^l), (q_1{}^l, q_2{}^l, \dots, q_m{}^l)). \tag{8}$$

The state $\boldsymbol{S^l}$ is directly reachable from the state $\boldsymbol{S^k}$ if the following conditions hold:

$$\forall_{i\in\{1,2,\dots,m\}} \forall_{j\in\{1,2.,,,,.n\}} \left[(a_i{}^k = \Delta) \wedge (a_{\beta_i(P_j)}{}^k = z_i{}^k) \Rightarrow (a_i{}^l = z_i{}^k) \right], \tag{9}$$

$$\forall_{i\in\{1,2,\dots,m\}} \forall_{j\in\{1,2.,,,,.n\}} \left[(a_i{}^k = \Delta) \wedge (a_{\beta_i(P_j)}{}^k \neq z_i{}^k) \Rightarrow (a_i{}^l \neq P_j) \right], \tag{10}$$

$$\forall_{i\in\{1,2,\dots,m\}} \left[(a_i{}^k = \Delta) \Rightarrow [(z_i{}^l = z_i{}^k) \wedge (q_i{}^l = q_i{}^k)] \right] \tag{11}$$

$$\forall_{i\in\{1,2,\dots,m\}} [(a_i{}^k \neq \Delta) \wedge (a_i{}^l \neq \Delta) \Rightarrow [(z_i{}^l = z_i{}^k) \wedge (a_i{}^l = a_i{}^k) \wedge (q_i{}^l = q_i{}^k)]], \tag{12}$$

$$\forall_{i\in\{1,2,\dots,m\}} \left[(a_i{}^k \neq \Delta) \wedge (a_i{}^l = \Delta) \Rightarrow [(z_i{}^l = crd_{(q_i{}^l)} \sigma_i) \wedge (q_i{}^l = \gamma_i(q_i{}^k))] \right], \tag{13}$$

$$\forall_{i\in\{1,2,\dots,m\}} \left[(a_i{}^k \neq \Delta) \wedge (z_{\alpha_i(a_i{}^k)}{}^k = a_i{}^k) \Rightarrow (a_{\alpha_i(a_i{}^k)}{}^l = a_i{}^k) \wedge (a_i{}^l = \Delta) \right], \tag{14}$$

$$\forall_{i\in\{1,2,\dots,m\}} \left[(a_i{}^k \neq \Delta) \wedge (z_{\alpha_i(a_i{}^k)}{}^k \neq a_i{}^k) \Rightarrow [(a_i{}^l = a_i{}^k) \wedge (q_i{}^l = q_i{}^k)] \right], \tag{15}$$

where:

m – a number of resources, n – a number of processes,

$\beta_i(P_j)$ – the index of resource directly proceeding the resource R_i, in the j-th process route p_j,$\beta_i(P_j) \in \{1,2,\dots,m\}$,

$\alpha_i(P_j)$ –the index of resource directly succeeding the resource R_i, in the j-th process route p_j,$\alpha_i(P_j) \in \{1,2,\dots,m\}$,

$\gamma_i(q_i{}^k)$ – the function defined by (16):

$$\gamma_i(a) = \begin{cases} a+1 & for\ a < lp(i) \\ 1 & for\ a = lp(i) \end{cases} \tag{16}$$

where: $lp(i)$ – the number of processes dispatched by the rule σ_i.

Consider the CCPS and its state space $\mathbb{S}$ (the set of all feasible states defined by (4)). The set $Sc = \{S^a, S^b, S^c, \dots, S^d\}$, $Sc \subset \mathbb{S}$ is called a cyclic steady state generated by an initial state $S^a \in \mathbb{S}$ if the following condition holds:

$$S^a \rightarrow S^b \rightarrow S^c \rightarrow \dots \rightarrow S^d \rightarrow S^a \tag{17}$$

where: $S^a \rightarrow S^b$ – the transition defined by (9)–(15).

In other words a cyclic steady state consists of such a set of states in which starting from any distinguished state it is possible to reach the rest of states and finally reach this distinguished state again. Each cyclic steady state is determined by so called period of cyclic steady state Tc.

A cyclic steady state period Tc is defined in the following way: $Tc = \|Sc\|$. Of course, for any $S^k \in Sc$ the following property holds $S^k \xrightarrow{Tc-1} S^k$.

Therefore, searching for a cyclic steady state Sc in a given CCPS can be seen as a reachability problem where for an assumed initial state S^0 the state S^k, such that following transitions $S^0 \overset{i}{\rightarrow} S^k \xrightarrow{Tc-1} S^k$ holds, is sought.

Note that cyclic steady state behavior of the CCPS (SSB_{CCPS} for short) follows from assumption that the quadruple (1) has been extended by an initial state $S^0 \in Sc$ and the state transition function δ. So, in general case one may consider the cyclic steady state space defined as sixtuple, where SS consists of states belonging to a cyclic steady state ($Sc \in SS$).

$$SSB_{\text{CCPS}} = (\Pi, T, R, \Theta, SS, \delta) \tag{18}$$

The graphical illustration of the state space following assumption from Fig. 1 is shown in Fig. 3. There are two cyclic steady states periodicity of which are equal to 5 (Sc_1) and 7 (Sc_2), respectively.

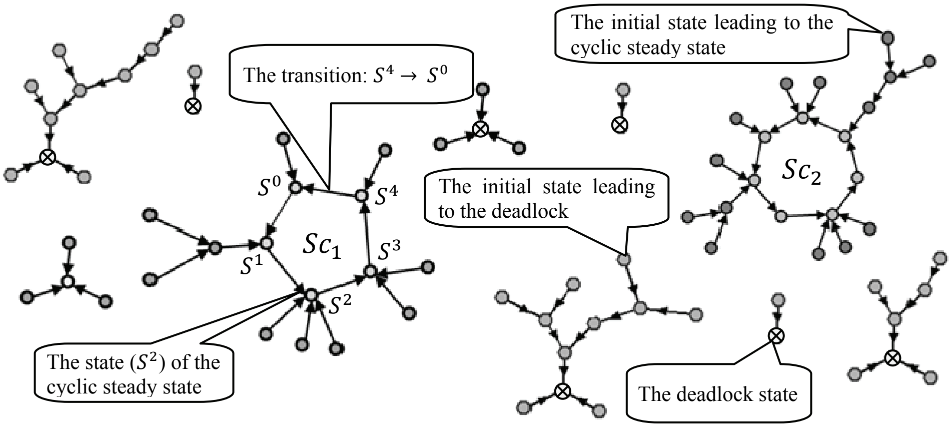

The initial state leading to the cyclic steady state
The transition: $S^4 \rightarrow S^0$
S^0
S^4
S^1
Sc_1
S^3
S^2
Sc_2
The initial state leading to the deadlock
The state (S^2) of the cyclic steady state
The deadlock state

Fig. 3. The state space of the CCPS from Fig. 1, following dispatching rules: $\sigma_1 = (P_2, P_3)$, $\sigma_3 = (P_2, P_1)$, $\sigma_5 = (P_3, P_1)$ and assumption limiting the all operation times to the same equal to 1 unit of time

3.2. OPERATIONS PRECEDENCE DIGRAPH

Let us consider the operations precedence digraph associated to the cyclic steady state Sc_1 (see Fig. 3) shown in Fig. 4. States from the cyclic steady state Fig. 4 a) correspond to the stages of prisms in operations precedence digraph (see Fig. 4 b)). Prisms about the triangular basis correspond to particular local cyclic processes P_i, and linking them arcs correspond to particular moments $x_{i,j}$ of operations o_{ij} beginning. In the case considered there are following potential initial states:

$$\ldots \rightarrow S^0 \rightarrow S^1 \rightarrow S^2 \rightarrow S^3 \rightarrow S^4 \rightarrow S^0 \rightarrow \ldots$$

corresponding to operations:

$$\ldots \rightarrow (o_{1,4}, o_{2,3}) \rightarrow (o_{2,1}, o_{3,5}) \rightarrow (o_{1,3}, o_{2,2}) \rightarrow (o_{3,1}) \rightarrow (o_{1,5}, o_{3,6}) \rightarrow \ldots$$

Each operation is repeated within the same period equal to 5 units of time.

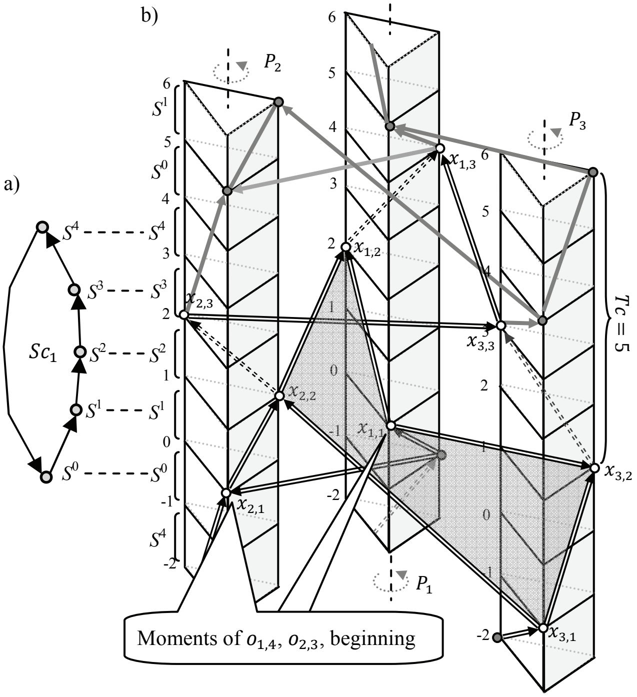
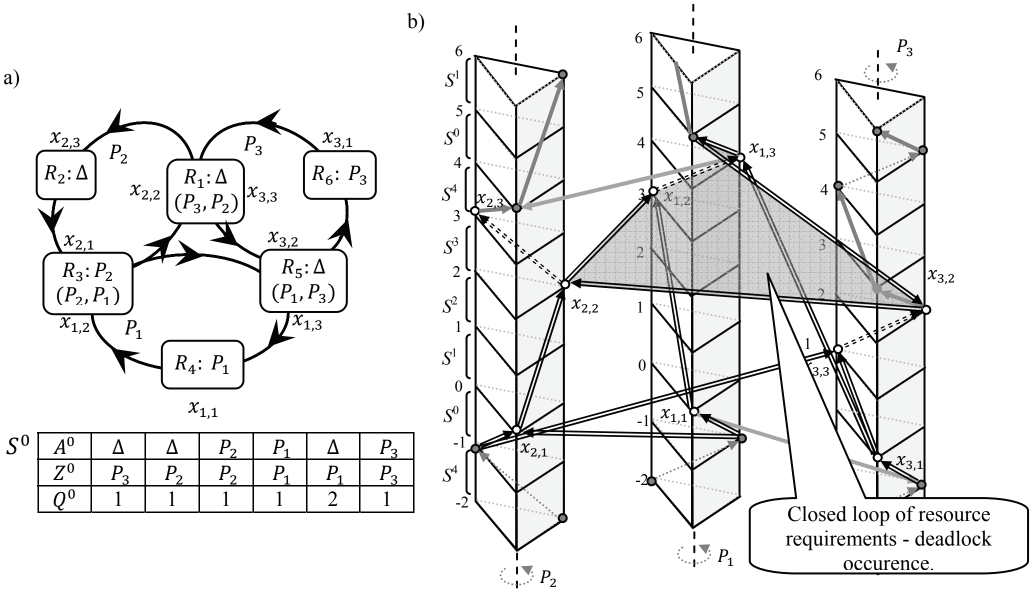


Fig. 4. The cyclic steady state of the CCPS following dispaching rules: $\sigma_1 = (P_2, P_3)$, $\sigma_3 = (P_2, P_1)$, $\sigma_5 = (P_3, P_1)$ a), the operations precedence digraph associated with Sc_1 b)



S^0	A^0	Δ	Δ	P_2	P_1	Δ	P_3
	Z^0	P_3	P_2	P_2	P_1	P_1	P_3
	Q^0	1	1	1	1	2	1

Fig. 5. The CCPS following dispaching rules: $\sigma_1 = (P_3, P_2)$, $\sigma_3 = (P_2, P_1)$, $\sigma_5 = (P_1, P_3)$ and the initial state S^0 a), the operations precedence digraph corresponding to a deadlock state occurrence b)

3.3. DEDALOCK AVOIDANCE CONSTRAINTS

Let us note that arcs within the period Tc zone, and distinguished by double line arcs create contours.

It can be proved, in case the one of the contours become a cycle, e.g., $x_{2,2}$, $x_{1,2}$, $x_{1,3}$, $x_{1,1}$, $x_{3,2}$ (see Fig. 5b) the relevant initial state, e.g., S^0, leads to a deadlock [2].

That follows from the set of contradictory conditions (i.e., the set of constraints of CS (3)) below:

$$x_{1,1} = \max\{x_{1,3} + 1 - Tc; x_{1,3} + 1 - Tc\},\ x_{1,2} = \max\{x_{1,1} + 1;\ x_{2,1} + 1\},$$
$$x_{1,3} = \max\{x_{1,2} + 1;\ x_{3,1} + 1\},\ x_{2,1} = \max\{x_{2,3} + 1 - Tc;\ x_{1,3} + 1 - Tc\},$$
$$x_{2,2} = \max\{x_{2,1} + 1; x_{3,2} + 1 - Tc\},\ x_{2,3} = \max\{x_{2,2} + 1; x_{2,2} + 1\}$$
$$x_{3,1} = \max\{x_{3,2} + 1 - Tc; x_{3,2} + 1 - Tc\},\ x_{3,2} = \max\{x_{3,3} + 1;\ x_{1,6} + 1 + Tc\}$$
$$x_{3,3} = \max\{x_{2,3} + 1 - Tc;\ x_{3,1} + 1\}$$

4. ILLUSTRATIVE EXAMPLE

4.1. AGVS REDEVELOPING

There is a given FMS of the structure depicted in the Fig. 1. Consider two processes P_1, P_2, corresponding to two AGVs. The relevant AGVS$_{12}$ is modeled by CCPS shown in Fig. 6 a). The relevant CSP (3) has the following form:

$$CS_{12} = ((\{R, \Pi, T, \Theta, X_{12}, Tc\}, \{D_R, D_\Pi, D_T, D_\Theta, D_X\}), C_{12})$$

where:

- decision variables X_{12} are following:

$$X_{12} = \{X_1, X_2\},$$
$$X_1 = (x_{1,1}, x_{1,2}, x_{1,3},), X_2 = (x_{2,1}, x_{2,2}, x_{2,3})$$

- the set of constraints C_{12} consists:

$$x_{1,1} = \max\{x_{1,3} + 1 - Tc;\ x_{1,3} + 1 - Tc\}, x_{1,2} = \max\{x_{1,1} + 1;\ x_{2,2} + 1\}$$
$$x_{1,3} = \max\{x_{1,2} + 1;\ x_{1,2} + 1\}, x_{2,1} = \max\{x_{2,3} + 1 - Tc;\ x_{1,3} + 1 - Tc\}$$
$$x_{2,2} = \max\{x_{2,1} + 1; x_{2,1} + 1\}, x_{2,3} = \max\{x_{2,2} + 1;\ x_{2,2} + 1\}$$

Assume the AGVS$_{12}$ considered has to be modernized by adding extra AGV, i.e. AGVS$_3$ (see Fig. 6 b).

$$CS_3 = ((\{R, \Pi, T, \Theta, X_3, Tc\}, \{D_R, D_\Pi, D_T, D_\Theta, D_X\}), C_3)$$

where:

- decision variables X_{12} are following:

$$X_3 = (x_{3,1}, x_{3,2}, x_{3,3})$$

- set of constraints C_3

$$x_{3,1} = \max\{x_{3,3} + 1 - Tc; x_{3,3} + 1 - Tc\}, x_{3,2} = \max\{x_{3,1} + 1;\ x_{3,1} + 1\}$$
$$x_{3,3} = \max\{x_{3,2} + 1;\ x_{3,2} + 1\}$$

The AGVS$_{12}$ and AGVS$_3$ have to be connected by common shared resources R_1, and R_5. Let us assume the following priority dispatching rules: $\sigma_1 = (P_2, P_3)$, $\sigma_5 = (P_1, P_3)$. The question regards of the resultant (see Fig. 6c)) AGVS's periodicity.

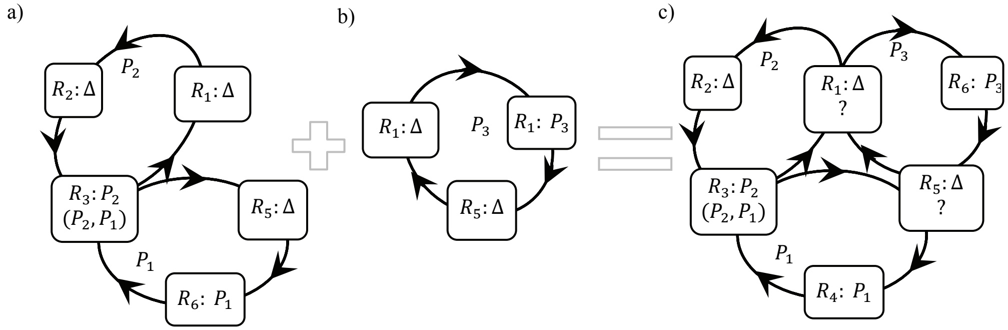


Fig. 6. An illustration of CCPS redeveloping: components a) b), the final structure c)

4.2. AGVS DOCKING PLACES SELECTION

The process of considered AGVS redeveloping can be illustrated using component operations precedence digraphs (see Fig. 7 a) b)) and the resultant operations precedence digraph shown in Fig. 7 c).

$$CS = \big((\{R, \Pi, T, \Theta, X, Tc\}, \{D_R, D_\Pi, D_T, D_\Theta, D_X\}), C\big),$$

where: $X = X_{12} \cup X_3$ – the set of decision variables following constraints:

$$x_{1,1} = \max\{ x_{1,3} + 1 - Tc;\ x_{1,3} + 1 - Tc\}, x_{1,2} = \max\{x_{1,1} + 1;\ x_{2,2} + 1\},$$
$$x_{1,3} = \max\{x_{1,2} + 1;\ x_{3,3} + 1\}, x_{2,1} = \max\{x_{2,3} + 1 - Tc;\ x_{1,3} + 1 - Tc\}$$
$$x_{2,2} = \max\{x_{2,1} + 1; x_{3,1} + 1 - Tc\}, x_{2,3} = \max\{x_{2,2} + 1;\ x_{2,2} + 1\}$$
$$x_{3,1} = \max\{x_{3,3} + 1 - Tc; x_{3,3} + 1 - Tc\}, x_{3,2} = \max\{x_{3,1} + 1;\ x_{1,1} + 1\}$$
$$x_{3,3} = \max\{x_{2,3} + 1;\ x_{3,2} + 1\}$$

Let us note that introduced newly obtained constraints $x_{1,3}$, $x_{2,2}$, $x_{3,2}$, $x_{3,3}$ are recursive representation of the previous ones. The modernized AGVS is deadlock-free and its periodicity is equal to 7.

The possible initial states of CCPS model, i.e. AGVs docking places, being the solution of the CS problem (where: $X_1 = (0, 5, 6), X_2 = (0, 4, 5)$, $X_3 = (3, 8, 9)$) considered are provided by the following subsets:

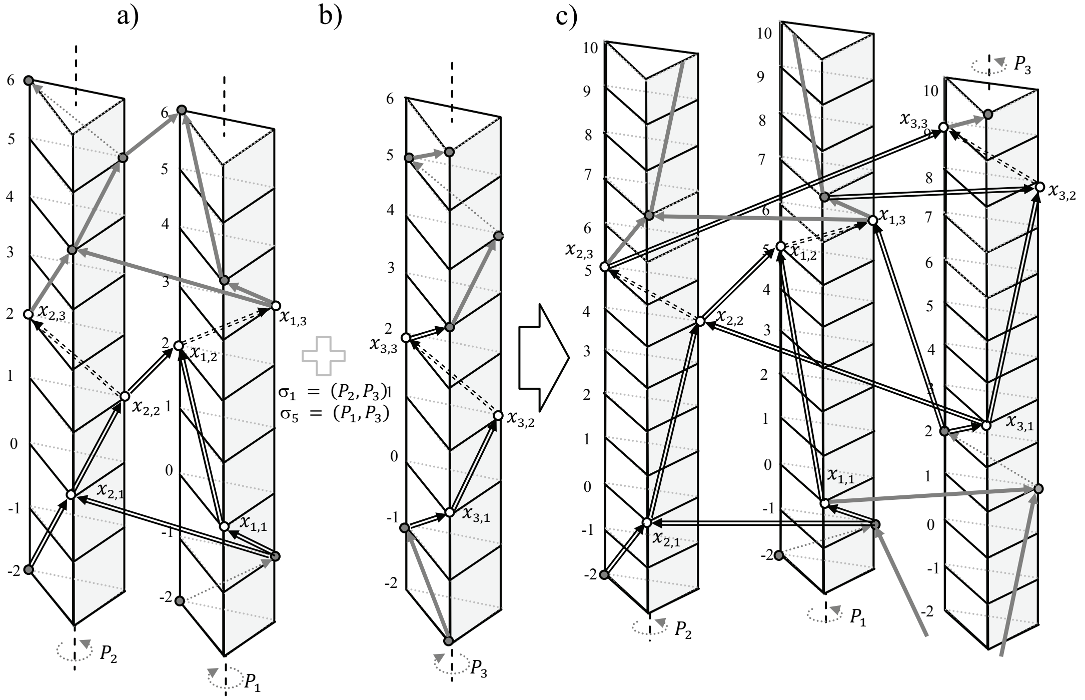


Fig. 7. An illustration of CCPS redeveloping: component operations precedence digraphs a) and b), the final structure of the operations precedence digraph c)

S^0: $R_1, R_2, \boxed{R_3}, \boxed{R_4}, R_5, \boxed{R_6}$

$A^0 = (\Delta, \Delta, \boxed{P_2}, \boxed{P_1}, \Delta, \boxed{P_3})$

$Z^0 = (P_3, P_2, P_2, P_1, P_3, P_3)$

$Q^0 = (2, 1, 1, 1, 2, 1)$

Zasób R_6 jest elementem podzbioru potencjalnych miejsc dokowania wózków

S^1:

$A^1 = (\Delta, \Delta, P_2, P_1, P_3, \Delta)$

$Z^1 = (P_3, P_2, P_2, P_1, P_3, P_3)$

$Q^1 = (2, 1, 1, 1, 2, 1)$

S^2:

$A^2 = (P_3, \Delta, P_2, P_1, \Delta, \Delta)$

$Z^2 = (P_3, P_2, P_2, P_1, P_1, P_3)$

$Q^2 = (2, 1, 1, 1, 1, 1)$

S^3:

$A^3 = (\Delta, \Delta, P_2, P_1, \Delta, P_3)$

$Z^3 = (P_2, P_2, P_2, P_1, P_1, P_3)$

$Q^3 = (1, 1, 1, 1, 1, 1)$

S^4:

$A^4 = (P_2, \Delta, \Delta, P_1, \Delta, P_3)$

$Z^4 = (P_2, P_2, P_1, P_1, P_1, P_3)$

$Q^4 = (1, 1, 2, 1, 1, 1)$

S^5:

$A^5 = (\Delta, P_2, P_1, \Delta, \Delta, P_3)$

$Z^5 = (P_3, P_2, P_1, P_1, P_1, P_3)$

$Q^5 = (2, 1, 2, 1, 1, 1)$

S^6:

$A^6 = (\Delta, P_2, \Delta, \Delta, P_1, P_3)$

$Z^6 = (P_3, P_2, P_2, P_1, P_1, P_3)$

$Q^6 = (2, 1, 1, 1, 1, 1)$

5. CONCLUDING REMARKS

CCSP that can be seen as a model of AGVS lead to two fundamental questions: Does there exist a control procedure (i.e. a set of dispatching rules and an initial state) enabling to guarantee an assumed steady cyclic state (e.g. following requirements caused by AGVS at hand) subject to AGVS's structure constraints? Does there exist the AGVS's structure such that an assumed steady cyclic state (e.g. following requirements caused by AGVS at hand) can be achieved? Response to these questions determines our further works.

We believe that this approach leads to solutions based on sufficient conditions that allow the designer to compose elementary systems in such a way as to obtain the final AGVS scheduling system with required quantitative and qualitative behavior features. So, we are looking for a method allowing one to replace the exhaustive search for the admissible control by a step-by-step structural design guaranteeing the required system behavior.

REFERENCES

[1] ALPAN G., JAFARI M.A., *Dynamic analysis of timed Petri nets: a case of two processes and a shared resource*, IEEE Trans. on Robotics and Automation, Vol. 13, No. 3, 1997, pp. 338–346.

[2] BOCEWICZ G., BACH I., BANASZAK Z., *Logic-algebraic method based and constraints programming driven approach to AGVs scheduling*. In: International Journal of Intelligent Information and Database Systems, Vol.3, No 1, 2009, pp. 56–74.

[3] BOCEWICZ G., BANASZAK Z., WÓJCIK R., *Design of admissible schedules for AGV systems with constraints: a logic-algebraic approach*, Lecture Notes in Artificial Intelligence 4496, Springer-Verlag, 2007, pp. 578–587.

[4] BOCEWICZ G., WÓJCIK R., BANASZAK Z., *Cyclic Steady State Refinement*. In: International Symposium on Distributed Computing and Artificial Intelligence, (Eds.) Abraham, A., Corchado, J.M., Rodríguez González, S., de Paz Santana, J.F. Series: Advances in Intelligent and Soft Computing, Vol. 91, Springer, 2011, pp. 191–198.

[5] BOCEWICZ G., WÓJCIK R., BANASZAK Z., *On undecidability of cyclic scheduling problems*. In: Mapping Relational Databases to the Semantic Web with Original Meaning, Lecture Notes in Computer Science, LNAI, Springer-Verlag, Vol. 5914, 2009, pp. 310–321.

[6] CAI X., LI K.N., *A genetic algorithm for scheduling staff of mixed skills under multi-criteria,* European Journal of Operational Research, 125, 2000, pp. 359–369.

[7] GAUJAL B., JAFARI M., BAYKAL-GURSOY M., ALPAN G., *Allocation sequences of two processes sharing a resource*, IEEE Trans. on Robotics and Automation, 11(5), 1995, pp. 748–353.

[8] GUY, R.K., *Diophantine Equations. Ch. D in Unsolved Problems in Number Theory*, 2nd ed. Springer-Verlag, New York, 1994, pp. 139–198.

[9] LAWLEY M.A., REVELIOTIS S.A., FERREIRA P.M., *A correct and scalable deadlock avoidance policy for flexible manufacturing systems,* IEEE Trans. on Robotics and Automation, Vol. 14, No. 5, 1998, pp. 796–809.

[10] LEVNER E,. KATS V., ALCAIDE D. PABLO L., CHENG T.C.E., *Complexity of cyclic scheduling problems: A state-of-the-art survey*, Computers and Industrial Engineering, Vol. 59, Issue 2, 2010, pp. 352–361.

[11] LIEBCHEN, C., MÖHRING, R.H., *A case study in periodic timetabling*. In: Electronic Notes in Theoretical Computer Science, Vol. 66 (6), 2002, pp. 21–34.

[12] PINEDO, M. L., *Planning and scheduling in manufacturing and services*, Springer-Verlag, New York, 2005.

[13] POLAK, M., MAJDZIK P., BANASZAK Z., WÓJCIK R., *The performance evaluation tool for automated prototyping of concurrent cyclic processes*, Fundamenta Informaticae. ISO Press, 60(1–4), 2004, 269–289.

[14] SCHULTE CH., SMOLKA G., WURTZ J., *Finite Domain Constraint Programming inOz*, DFKI OZ documentation series, German Research Center for Artificial Intelligence, Saarbrucken, Germany, 1998.

[15] SMART NIGIEL P., *The Algorithmic Resolution of Diophantine Equations*, London Mathematical Society Student Text, 41. Cambridge University Press, Cambridge, 1998.

[16] SONG J.-S., LEE T.-E., *Petri net modeling and scheduling for cyclic job shops with blocking*. In: Computers & Industrial Engineering, Vol. 34, No. 2, 1998, pp. 281–295.

[17] VON KAMPMEYER T., *Cyclic scheduling problems*, Ph.D. Dissertation, Fachbereich Mathematik/ Informatik, Universität Osnabrück, 2006.

Multiprocessing computer system,
programs scheduling, resources allocation

Zbigniew BUCHALSKI*

SCHEDULING SETS OF PROGRAMS AND MEMORY PAGES ALLOCATION IN MULTIPROCESSING COMPUTER SYSTEM

The aim of the work is present results of research on the problem of time-optimal programs scheduling and primary memory pages allocation in computer system consisting of a group of parallel processors for special type of programs processing time function. We consider an multiprocessing computer system consisting of *m* parallel processors, common primary memory and external memory. The primary memory contains *N* pages of identical capacity. This system can execute *n* independent programs. Because our problem belongs to the class of *NP*-complete problems we propose an heuristic algorithm to minimize schedule length criterion, which employs some problem properties. Some results of executed computational experiments for basis of this heuristic solution procedure are presented.

1. INTRODUCTION

In last years the time-optimal problems of tasks scheduling and resources allocation are intensive developing [4, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]. Scheduling problems can be understood very broadly as the problem of the allocation of resources over time to perform a set of tasks. By resources we understand arbitrary means tasks compete for. They can be of a very different nature, e.g. energy, tools, money, manpower. Tasks can have a variety of interpretation starting from machining parts in manufacturing systems up to processing information in computer systems. The further development of the research has been connected with applications, among other things in multiprocessing computer systems [1, 2, 3, 5, 7, 20].

* Institute of Computer Engineering, Control and Robotics, Wrocław University of Technology, Wrocław, Poland, e-mail: zbigniew.buchalski@pwr.wroc.pl

In multiprocessing computer systems is usually used common primary memory with limited capacity and external memory. The external memory has significantly longer access time and this is why minimization of the number of demands to the external memory during programs processing is necessary.

In this work the problem optimization of programs scheduling and optimal allocation of primary memory pages to the processors are considered. We propose an heuristic algorithm for solving of a optimization problem. In the second section formulation of optimization problem is presented. In the third section an heuristic algorithm is given and in the fourth section several experimental results on the base this heuristic algorithm are presented. Last section contains final remarks.

2. FORMULATION OF OPTIMIZATION PROBLEM

We consider an multiprocessing computer system (as shown in Fig.1) containing *m* processors, common primary memory and external memory. This system can execute *n* independent programs.

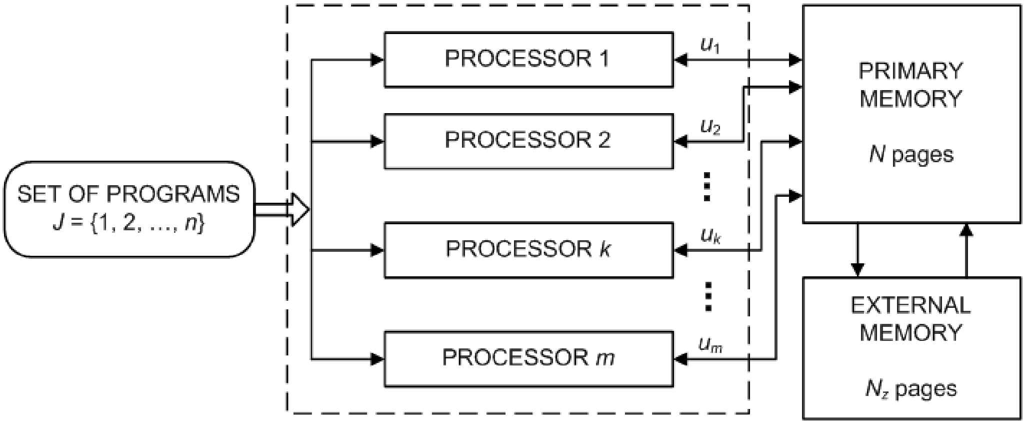


Fig. 1. Multiprocessing computer system

This system can execute *n* independent programs. We assume about this system, that it is paged virtual memory system and that:

- the primary memory contains *N* pages of identical capacity,
- each the processor has access to every one of *N* primary memory pages and may execute every one of *n* programs,
- the external memory contains N_z pages (the external memory pages capacity is equal to the primary memory pages capacity), $N_z > N$,

- during execution of all *n* programs, the number of u_k primary memory pages is allocated to the *k*-th processor; $\sum_{k=1}^{m} u_k \leq N$. Each processor may use only allocated to him the primary memory pages.

Let $J = \{1,2,...,n\}$ be the set of programs, $U = \{1,2,...,N\}$ – set of primary memory pages, P denotes the set of processors $P = \{1,2,...,m\}$. Processing time of *i*-th program on *k*-th processor is given by following function:

$$T_i(u_k, k) = a_{ik} + \frac{b_{ik}}{u_k}, \quad u_k \in U, \quad 1 \leq k \leq m, \quad i \in J, \tag{1}$$

where $a_{ik} > 0$, $b_{ik} > 0$ – parameters characterized *i*-th program and *k*-th processor.

This programs scheduling and primary memory pages allocation problem in multi processing computer system can be formulated as follows: find scheduling of *n* independent programs on the *m* processors running parallel and partitioning of *N* primary memory pages among *m* processors, that schedule length criterion is minimized.

Let $J_1, J_2, ..., J_k, ..., J_m$ be defined as subsets of programs, which are processing on the processors 1, 2,..., *k*,..., *m*. The problem is to find such subsets $J_1, J_2, ..., J_k, ..., J_m$ and such pages numbers $u_1, u_2, ..., u_k, ..., u_m$, which minimize the T_{opt} of all set *J*:

$$T_{opt} = \min_{\substack{J_1, J_2, ..., J_m \\ u_1, u_2, ..., u_m}} \max_{1 \leq k \leq m} \left\{ \sum_{i \in J_k} T_i(u_k, k) \right\} \tag{2}$$

under the following assumptions:

$$(i) \quad J_s \cap J_t = \emptyset, \quad s,t = 1,2,...,m, \quad s \neq t, \quad \bigcup_{k=1}^{m} J_k = J,$$

$$(ii) \quad \sum_{k=1}^{m} u_k \leq N, \quad u_k \in U, \quad k = 1,2,...,m,$$

$$(iii) \quad u_1, u_2, ..., u_m \text{ – positive integer.}$$

The assumption (iii) is causing, that the stated problem is very complicated therefore to simplify the solution our problem we assume in the sequel that primary memory pages are continuous. The numbers of pages obtained by this approach are rounded to the integer numbers (look **Step 12** in the heuristic algorithm) and finally our problem can formulated as following minimizing problem:

$$T_{opt} = \min_{\substack{J_1,J_2,\dots,J_m \\ u_1,u_2,\dots,u_m}} \max_{1\le k\le m} \left\{ \sum_{i\in J_k} \widetilde{T}_i\left(u_k,k\right) \right\} \tag{3}$$

under the following assumptions:

$$(i)\quad J_s \cap J_t = \varnothing, \quad s,t = 1,2,\dots,m, \quad s \ne t, \quad \bigcup_{k=1}^{m} J_k = J,$$

$$(ii)\quad \sum_{k=1}^{m} u_k \le N, \quad u_k \ge 0, \quad k = 1,2,\dots,m,$$

where $\widetilde{T}_i : [0,N] \times \{1,2,\dots,m\} \to R^+$ is the extension of function $T_i : \{1,2,\dots,N\} \times$ $\times \{1,2,\dots,m\} \to R^+$ and formulated by function:

$$\widetilde{T}_i(u_k,k) = a_{ik} + \frac{b_{ik}}{u_k}, \quad u_k \in [0,N], \quad 1 \le k \le m, \quad i \in J. \tag{4}$$

Taking into account properties of the function $\widetilde{T}_i(u_k,k)$, it is easy to show the truth of the following theorem:

Theorem 1.
If the sets u_k^*, J_k^*, $k = 1,2,\dots,m$ are a solutions of minimizing problem (3), then:

$$(i)\quad \sum_{k=1}^{m} u_k^* = N; \quad u_k^* > 0, \quad k : J_k^* \ne \varnothing, \quad k = 1,2,\dots,m;$$

$$u_k^* = 0, \quad k : J_k^* = \varnothing, \quad k = 1,2,\dots,m;$$

$$(ii)\quad \sum_{i\in J_k^*} \widetilde{T}_i(u_k^*,k) = \text{const}, \quad k : J_k^* \ne \varnothing, \quad k = 1,2,\dots,m.$$

We define function $F(J_1 J_2,\dots,J_m)$, which value is solution following system of equations:

$$\begin{cases} \sum\limits_{i\in J_k} a_{ik} + \dfrac{\sum\limits_{i\in J_k} b_{ik}}{u_k} = F\left(J_1, J_2, ..., J_m\right), \quad k : J_k \neq \varnothing, \quad k = 1,2,...,m \\ \sum\limits_{k:J_{k\neq\varnothing}} u_k = N; \quad u_k > 0 \qquad\qquad k : J_k \neq \varnothing, \quad k = 1,2,...,m \end{cases} \tag{5}$$

On the basis of **Theorem 1** and (5), problem (3) will be following:

$$T_{opt} = \min_{J_1, J_2, ..., J_m} F(J_1, J_2, ..., J_m) \tag{6}$$

under the assumptions:

$$(i) \quad J_s \cap J_t = \varnothing; \quad s,t = 1,2,...,m, \quad s \neq t$$

$$(ii) \quad \bigcup_{k=1}^{m} J_k = J; \quad k = 1,2,...,m,$$

If $J_1^*, J_2^*, ..., J_m^*$ are solutions of problem (6), it u_k^*, J_k^* $k = 1, 2, ..., m$ are solutions of problems (3), where:

$$u_k^* = \begin{cases} \dfrac{\sum\limits_{i\in J_k^*} b_{ik}}{F(J_1^*, J_2^*, ..., J_m^*) - \sum\limits_{i\in J_k^*} a_{ik}}; & k : J_k^* \neq \varnothing, \quad 1 \leq k \leq m, \\ 0 \qquad ; & k : J_k^* = \varnothing, \quad 1 \leq k \leq m\,. \end{cases} \tag{7}$$

3. THE HEURISTIC ALGORITHM

We assume that the first processor from the set *P* has highest speed and the last processor from the set *P* has least speed. We assume also if be of assistance in pages allocation so-called partition of pages coefficient α; $\alpha > 1$. To the last *m* processor is allocated u_m pages according to the following formula:

$$u_m = \frac{N}{1 + \sum\limits_{k=1}^{m-1} \left[(m-k) \cdot \alpha\right]} \tag{8}$$

To the remaining processors are allocated pages according to the formula:

$$u_k = (m-k)\cdot\alpha\cdot u_m;\quad k = 1,2,...,m-1. \tag{9}$$

The proposed heuristic algorithm is as follows:

Step 1. For given $u_k = \frac{N}{m}$ and random generate parameters a_{ik}, b_{ik} calculate the processing times of programs $T_i(u_k, k)$ according to the formula (1).

Step 2. Schedule programs from longest till shortest times $T_i(u_k, k)$ and formulate the list *L* of these programs.

Step 3. Calculate mean processing time T_{mean} every processors according to follows formula:

$$T_{mean} = \frac{\sum_{i=1}^{n} T_i(u_k, k)}{m};\quad i \in J,\quad k \in P,\quad u_k = \frac{N}{m}.$$

Step 4. Schedule first *m* longest programs from the list *L* to the succeeding *m* processors from first processor to the *m*-th processor and eliminate these programs from the list *L*.

Step 5. Allot in turn shortest and longest programs from the list *L* to the succeeding processor from first processor to the *m*-th processor for the moment, when the sum of processing times these programs to keep within the bounds of time T$_{mean}$ and eliminate these programs from the list *L*. If list *L* is not empty go to the next step, if is empty go to the **Step 7**.

Step 6. Remainder of programs in the list *L* allotte to the processors according to the algorithm *LPT* (Longest Processing Time) to moment of finish the list *L*.

Step 7. Calculate total processing time T_{opt} of all programs for scheduling $J_1, J_2, ..., J_m$, which was determined in the **Steps 3÷6** and for given numbers of pages $u_k = \frac{N}{m}$.

Step 8. For given partition of pages coefficient α allot pages u_k, $k \in P$ to succeeding processors as calculated according formula (8) and (9).

Step 9. For programs scheduling which was determined in **Steps 3÷6** and for numbers of pages u_k, $k \in P$ allotted to processors in the **Step 8** calculate total processing time T_{opt} of all programs.

Step 10. Repeat the **Step 8** and **Step 9** for the next nine augmentative succeeding another values of coefficient α.

Step 11. Compare values of total processing times T_{opt} of all programs calculated after all samples with different values of coefficient α (**Steps 8÷10**). Take this coefficient α when total processing time T_{opt} of all programs is shortest.

Step 12. Find the discrete numbers $\hat{u}_k$ of pages, k = 1, 2, ..., m according to follows dependence:

$$\hat{u}_{\beta(k)} = \begin{cases} \lfloor u_{\beta(k)} \rfloor + 1 & ; \quad k = 1,2,...,\Delta \\ \lfloor u_{\beta(k)} \rfloor & ; \quad k = \Delta+1, \Delta+2, ..., m \end{cases}$$

where $\Delta = N - \sum_{j=1}^{m} \lfloor u_j \rfloor$ and β is permutation of elements of set P ={1, 2, ..., m} such, that $u_{\beta(1)} - \lfloor u_{\beta(1)} \rfloor \geq u_{\beta(2)} - \lfloor u_{\beta(2)} \rfloor \geq \ldots \geq u_{\beta(m)} - \lfloor u_{\beta(m)} \rfloor$.

4. NUMERICAL EXAMPLES

On the base this heuristic algorithm were obtained results of computational experiments for ten another values of coefficient α = 3, 6, 9, … , 30. For the definite number of programs n = 50, 100, 150, 200, 250, number of processors m = 4, 8, 12, 16, 20, 24 and number of primary memory pages N =10.000 were generated parameters a_{ik}, b_{ik} from the set $\{0.3, 0.6, ..., 9.6, 9.9\}$. For each combination of n and m were generated 40 instances. The results of comparative analysis of heuristic algorithm proposed in this work and the algorithm *LPT* are showed in the Table 1.

Table 1. The results of comparative analysis of heuristic algorithm and algorithm *LPT*

n/m	Number of instances, when:			Δ^H	S^H	S^{LPT}
	$T_{opt}^H < T_{opt}^{LPT}$	$T_{opt}^H = T_{opt}^{LPT}$	$T_{opt}^H > T_{opt}^{LPT}$	%	sec	sec
50/4	20	1	19	1,9	2,5	1,8
100/4	22	0	18	2,2	4,7	3,2
150/4	23	1	16	3,4	8,8	7,2
200/4	23	2	15	4,6	12,6	9,2
250/4	24	2	14	5,9	14,3	10,6
50/8	21	1	18	2,1	2,8	2,1
100/8	23	2	15	3,1	5,6	4,2
150/8	24	0	16	3,7	9,8	7,9

continue Table 1						
200/8	25	1	14	4,2	13,9	11,5
250/8	26	0	14	5,5	15,6	13,1
50/12	19	3	18	2,3	2,9	2,2
100/12	21	1	18	3,5	7,2	5,9
150/12	23	1	16	4,6	10,5	8,4
200/12	24	2	14	4,8	14,7	10,8
250/12	26	2	12	5,9	16,2	13,4
50/16	20	2	18	2,3	3,9	2,8
100/16	22	1	17	3,6	8,1	6,5
150/16	24	1	15	4,7	12,5	9,9
200/16	26	2	12	5,5	16,4	13,2
250/16	28	1	11	5,9	18,5	14,8
50/20	21	1	18	2,6	5,6	4,3
100/20	23	0	17	3,8	8,4	5,9
150/20	24	2	14	4,6	11,6	9,5
200/20	25	1	14	5,4	16,4	13,6
250/20	26	2	12	6,4	18,7	15,2
50/24	21	1	18	2,9	6,2	4,9
100/24	23	0	17	3,8	9,4	7,1
150/24	24	2	14	4,9	12,4	9,5
200/24	26	2	12	6,1	17,4	13,9
250/24	29	1	10	7,6	19,8	16,4

In the Table 1 there are the following designations:

n – number of programs,

m – number of processors,

T_{opt}^{H} – total processing time of all set of programs J for the heuristic algorithm,

T_{opt}^{LPT} – total processing time of all set of programs J for the algorithm LPT,

Δ^{H} – the mean value of the relative improvement T_{opt}^{H} in relation to T_{opt}^{LPT} :

$$\Delta^{H} = \frac{T_{opt}^{LPT} - T_{opt}^{H}}{T_{opt}^{H}} \cdot 100\% \,,$$

S^{H} – the mean time of the numerical calculation for the heuristic algorithm,

S^{LPT} – the mean time of the numerical calculation for the algorithm LPT.

5. FINAL REMARKS

Computational experiments presented above show, that quality of programs scheduling in parallel multiprocessing computer system based on the proposed in this work heuristic algorithm increased in compare with simple *LPT* algorithm. The few percentages improvement of time T^{H} in compare with T^{LPT} can be the reason why heuristic algorithms researches will be successfully taken in the future.

Application of presented in this work heuristic algorithm is especially good for multiprocessing computer systems with great number of programs because in this case the Δ^{H} improvement is the highest. Proposed heuristic algorithm can be used not only to programs scheduling in multiprocessing computer systems but also to task scheduling in parallel machines or even to operations scheduling in workplaces equipped with production machines.

REFERENCES

[1] BIANCO L., BŁAŻEWICZ J., DELL'OLMO P., DROZDOWSKI M., *Preemptive scheduling of multiprocessor tasks on the dedicated processors system subject to minimal lateness*. Information Processing Letters, 46, 1993, 109–113.

[2] BIANCO L., BŁAŻEWICZ J., DELL'OLMO P., DROZDOWSKI M., *Linear algorithms for preemtive scheduling of multiprocessor tasks subject to minimal lateness,* Discrete Applied Mathematics, 72, 1997, 25–46.

[3] BŁAŻEWICZ J., DROZDOWSKI M., WERRA D., WĘGLARZ J., *Scheduling independent multiprocessor tasks before deadlines*. Discrete Applied Mathematics 65 (1–3), 1996, 81–96.

[4] BŁAŻEWICZ J., ECKER K., SCHMIDT G., WĘGLARZ J., *Scheduling in Computer and Manufacturing Systems*. Springer-Verlag, Berlin-Heidelberg, 1993.

[5] BŁAŻEWICZ J., LIU Z., *Scheduling multiprocessor tasks with chain constraints*. European Journal of Operational Research, 94, 1996, 231–241.

[6] BOCTOR F., *A new and efficient heuristic for scheduling projects will resources restrictions and multiple execution models*. European Journal of Operational Research, vol. 90, 1996, 349–361.

[7] BRAH S.A., LOO L.L., *Heuristics for scheduling in a flow shop with multiple processors*, European Journal of Operational Research, Vol. 113, No. 1, 1999, 113–122.

[8] BUCHALSKI Z., *Application of heuristic algorithm for the tasks scheduling on parallel machines to minimize the total processing time*. Proceedings of the 15th International Conference on Systems Science , vol. 2, Wrocław, 2004.

[9] BUCHALSKI Z., *Minimising the Total Processing Time for the Tasks Scheduling on the Parallel Machines System*. Proc. of the 12th IEEE International Conference on Methods and Models in Automation and Robotics, Domek S., Kaszyński R. (Eds.), Międzyzdroje, Poland, MMAR 2006, 28–31 August 2006, 1081–1084.

[10] CHENG J., KARUNO Y., KISE H., *A shifting bottleneck approach for a parallel-machine flowshop scheduling problem*, Journal of the Operational Research Society of Japan, Vol. 44, No. 2, 2001, 140–156.

[11] GUPTA J.N.D., HARIRI A.M.A., POTTS C.N., *Scheduling a two-stage hybrid flow shop with parallel machines at the first stage*, Annals of Operations Research, Vol. 69, No.0, 1997, 171–191.

[12] JANIAK A., KOVALYOV M., *Single machine scheduling subject to deadlines and resources dependent processing times*. European Journal of Operational Research, , vol. 94, 1996, 284–291.

[13] JÓZEFCZYK J., *Task scheduling in the complex of operation with moving executors*, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 1996 (in Polish).

[14] JÓZEFCZYK J., *Selected Decision Making Problems in Complex Operation Systems*, Monografie Komitetu Automatyki i Robotyki PAN, t. 2, Oficyna Wydawnicza Politechniki Wrocławskiej, Warszawa–Wrocław, 2001 (in Polish).

[15] JÓZEFOWSKA J., MIKA M., RÓŻYCKI R., WALIGÓRA G., WĘGLARZ J., *Discrete-continuous scheduling to minimize maximum lateness*, Proceedings of the Fourth International Symposium on Methods and Models in Automation and Robotics MMAR'97, Międzyzdroje, Poland, 1997, 947–952.

[16] JÓZEFOWSKA J., MIKA M., RÓŻYCKI R., WALIGÓRA G., WĘGLARZ J., *Local search metaheuristics for discrete-continuous scheduling problems*, European Journal of Operational Research, 107, 1998, 354–370.

[17] JÓZEFOWSKA J., WĘGLARZ J., D*iscrete-continous scheduling problems – mean completion time result*, European Journal of Operational Research, vol. 94, No. 2, 1996, 302–310.

[18] JÓZEFOWSKA J., WĘGLARZ J., *On a methodology for discrete-continous scheduling*, European Journal of Operational Research, Vol. 107, No. 2, 1998, 338–353.

[19] NOWICKI E., SMUTNICKI C., *The flow shop with parallel machines*. A Tabu search approach. European Journal of Operational Research 106, 1998, 226–253.

[20] WĘGLARZ J., *Multiprocessor scheduling with memory allocation – a deterministic approach.* IEEE Trans. Comput., C-29, 1980, 703–710.

scheduling, learning effect, aging effect, computational complexity, heuristic

Agnieszka RUDEK*, Radosław RUDEK**

FLOWSHOP SCHEDULING WITH POSITION DEPENDENT JOB PROCESSING TIMES

In this work, we show that the makespan minimization problem in the two-processor flowshop environment becomes strongly NP-hard if the processing time of a job is described by an arbitrary function dependent on its position in a sequence (models learning or aging). Moreover, we construct the fast NEH algorithm with complexity lower than its standard version. Efficiency of the proposed method was numerically analysed for the problems with the aging effect.

1. INTRODUCTION

Flowshop scheduling problems constitute a significant part of scheduling theory, since they describe settings, where a task has to flow through all production/processing stages to be completed (see [3]). However, the accuracy of solving real-life problems strongly depends on a reliability of mathematical models used during designing process of solution algorithms. Therefore, flowshop scheduling problems that assume constant job processing times are not sufficient for modelling problems that occur in the real-life systems ([3]), inter alia where the processing times of jobs depend on the number of earlier performed jobs that model learning or aging effects (see [1], [5], [6]).

In this work, we analyse a flowshop scheduling problem with processing times dependent on the number of performed jobs (that are also called position dependent job processing times). If job processing times decrease with the number processed jobs then it is called the learning effect, otherwise we called such phenomenon the aging

* Institute of Computer Engineering, Control and Robotics, Wrocław University of Technology, Poland, e-mail: agnieszka.wielgus@pwr.wroc.pl.

** Wrocław University of Economics, Poland, e-mail: radoslaw.rudek@ue.wroc.pl.

effect. We show the two-processor flowshop problem with the makespan minimization becomes strongly NP-hard if the processing time of a job is not a constant value, but it depends on its position in a sequence (i.e., the number of already performed jobs) for learning and aging effects. Furthermore there is a lack of efficient heuristic algorithms for the makespan flowshop problem with learning or aging effects. Therefore, we also provide the fast NEH algorithm that solves the general version of the problem, i.e., with arbitrary position dependent job processing times that can describe both learning and aging effects.

2. PROBLEM FORMULATION

There are given a set $J = \{1,..., n\}$ of n jobs and m processors, namely $M = \{M_1,..., M_m\}$. Each job j consists of a set $O = \{O_{1,\mathrm{j}},..., O_{m,j}\}$ of m operations. Each operation $O_{z,j}$ has to be processed on processor M_z ($z = 1,..., m$). Moreover, operation $O_{z+1,j}$ may start only if $O_{z,j}$ is completed. It is assumed that processors have to process jobs in the same order that is called a permutation flowshop. Moreover, each processor can process one operation at a time, and there are no precedence constraints between jobs. Operations are non-preemptive and available for processing at time 0 on M_1. Further, instead of operation $O_{z,j}$, we say job j on processor M_z.

Each job j is characterized by its processing time $p_j^{(z)}(v)$ that is a non-increasing (learning) or non-decreasing (aging) function of its position v in a sequence on processor M_z ($z = 1,...,m$), in other words a function dependent on a number of previously processed jobs. In this work, we consider two models of job processing times. In the first one that describes learning, the processing time of job j on processor Mz is given by the following function:

$$p_j^{(z)}(v) = a_j^{(z)} - b_j^{(z)} \min\{v-1, g_j^{(z)}\}, \qquad j,v = 1,\ldots,n \tag{1}$$

where $a_j^{(z)}$ is the normal processing time of job j on M_z, $b_j^{(z)}$ is the learning ratio of job j on M_z and $g_j^{(z)}$ is the learning threshold of this job on M_z (z = 1, 2); moreover $a_j^{(z)} - b_j^{(z)}(n-1) > 0,$ for $j = 1,..., n$ and $z = 1, 2$.

The second model describes the aging effect and the processing time of job j on processor Mz is given as follows:

$$p_j^{(z)}(v) = a_j^{(z)} + b_j^{(z)} \min\{v-1-g_j^{(z)}, 0\}, \qquad j,v = 1,\ldots,n \tag{2}$$

where $b_j^{(z)}$ and $g_j^{(z)}$ denote the aging ratio and the aging threshold of job *j* on *Mz* (*z* = 1, 2), respectively.

Note that both functions (1) and (2) depend on a position *v* of job *j*. But it can be observed that it is a function dependent on a number of previously performed jobs (*v*-1), and this interpretation suits better for a description of real-life problems. However, for convenience and simplification of the further mathematical considerations, we will use the "position dependent" terminology and interpretation.

In (1) and (2), we use the same symbols (i.e., $b_j^{(z)}$ and $g_j^{(z)}$) for learning and aging parameters due to the similar meaning. If a function describing processing times is non-increasing (1), then the symbols describe learning parameters otherwise they describe aging parameters (2).

For the considered flowshop problem, the schedule of jobs on the processors can be unambiguously defined by their sequence (permutation) *π*. Thus, for each job $\pi(i)$, i.e., scheduled in the *i*-th position in *π*, we can determine its completion time $C_{\pi(i)}^{(z)}$ on machine *Mz*:

$$C_{\pi(i)}^{(z)} = \max\{C_{\pi(i)}^{(z-1)}, C_{\pi(i-1)}^{(z)}\} + p_{\pi(i)}^{(z)}(i) \tag{3}$$

where $C_{\pi(1)}^{(0)}$ = $C_{\pi(0)}^{(z)}$ = 0 for *z* = 1,...,*m* and $C_{\pi(i)}^{(1)} = \sum_{l=1}^{i} p_{\pi(l)}^{(1)}(l)$ is the completion time of a job placed in position *i* in the permutation *π* on M_1. The objective is to find such schedule *π* of jobs on the processors that minimize the makespan $C_{\max} = \max_{j \in J}\{C_j\}$ (in the considered problems $C_{\max} = C_{\pi(n)}^{(m)}$). The considered problem, according to the standard three field notation scheme $\alpha \mid \beta \mid \gamma$, with model (1) will be denoted as $F2|LE|C_{\max}$, with (2) as $F2|AE|C_{\max}$ and with the general model as $Fm|\, p_j^{(z)}(v)\,|C_{\max}$.

3. COMPUTATIONAL COMPLEXITY

In this section, we will show that the problems $F2|LE|C_{\max}$ and $F2|AE|C_{\max}$ are strongly NP-hard. In order to do that, we will show that the strongly NP-complete problem 3-PARTITION ([2]) can be transformed in a pseudopolynomial time to the decision versions of these scheduling problems.

3-PARTITION (3PP) ([2]): There are given positive integers *m*, *B* and $x_1,...,x_{3m}$ of 3*m* positive integers satisfying $\sum_{q=1}^{3m} x_q = mB$ and $B/4<x_q<B/2$ for *q* = 1,..., 3*m*. *Does there exist a partition of the set X* = {1,...,*m*} *into m disjoint subsets* $X_1,..., X_m$ *such that* $\sum_{q \in X_i} x_q = B$ *for i* = 1,...,*m?*

3.1. AGING EFFECT

Theorem 1 *The problem $F2|AE|C_{\max}$ is strongly NP-hard.*

Proof. First, decision version of the problem $F2|AE|C_{\max}$, (Decision Problem with Aging Effect, DPAE) is defined. *Does there exist such a schedule π of jobs on the processors M_1 and M_2 for which $C_{\max}(\pi)$ is lower or equal than the given value y?*

A pseudopolynomial time reduction from 3PP to DPAE is given. The constructed instance of DPAE contains $3m$ *partition* jobs and m^2B *enforcer* jobs. The parameters of partition jobs are defined as follows:

$$a_j^{(1)} = mBN(m+1)b + 2mBb; \qquad g_j^{(1)} = 0;$$
$$b_j^{(1)} = mBb;$$
$$a_j^{(2)} = 4mBN(m+1) + 4m(m+1)Nx_j + 12mx_j; \qquad g_j^{(2)} = 0;$$
$$b_j^{(2)} = 4m(B - x_j);$$

for $j = 1,...,3m$ and m^2B enforcer jobs defined as follows:

$$a_{l_i}^{(1)} = D_{i-1} + 4N\big(m+1-(i-1)\big) + 3; \qquad b_{l_i}^{(1)} = y; \qquad g_{l_i}^{(1)} = Ni - 4;$$
$$a_{l_i}^{(2)} = 3Nb\big(m+1+i\big) + 3; \qquad b_{l_i}^{(2)} = 0; \qquad g_{l_i}^{(1)} = 0;$$

where: $l_i = (mB)(i-1)+1,\ldots,mBi$ (for $I = 1,...,$ m), $N = (mB+3)$, $b = N^2m^3$, $D_i = 12N(m+1+i)$ and $y = mB\left[\sum_{i=1}^{m}\big([D_{i-1} + 4N(m+1-(i-1)) + 3] + [3Nb(m+1-i)]\big) + [D_m + 4N + 3] + 3m\right].$

From the reduction follows that before each enforcer job $e_{l_i} \in l_i$ at last v = Ni -4 jobs can be scheduled, otherwise for M_1 we have $C_{\max} = C_{\max}^{(1)} > p_{l_i}^{(1)}(v) > y$, where $C_{\max}^{(1)}$ is the completion time of all jobs on M_1. Note also that $p_{l_i}^{(1)}(v) = a_{l_i}^{(1)}$ for $v \le g_{l_i}^{(1)}$ and $l_i \in E_i$ (i = 1,..., m), hence the processing times of the enforcer jobs on M_1 are constant in an optimal solution π, otherwise $C_{\max} > y$.

On the other hand, assuming that processing times of enforcer jobs are constant in π (i.e., before $e_{l_i} \in l_i$ at last $Ni - 4$ jobs are scheduled), from the reduction follows that $C_{\max}^{(1)}$ is smaller if partition jobs occupy the lowest possible positions in schedule π. It follows from $p_j^{(1)}(v) = p^{(1)}(v)$ and from the fact that $p^{(1)}(v)$ is a non-decreasing function of v.

Therefore, using inductive principle, we can easily prove that $C_{\max}^{(1)}$ is minimum for such a schedule π, which satisfy the following conditions. The enforcer jobs are partitioned into m sets E_i, such that $E_i = \{(mB)(i-1)+1,\ldots,mBi\}$ for i=1,..., m and jobs from E_i are scheduled one after another. Since jobs from E_i are identical they are scheduled in an arbitrary order within E_i. Next, between each pair E_i and E_{i+1} for i=1,..., m-1 exactly one set X_i of partition jobs is scheduled, such that the cardinality of X_i is equal 3 for i=1,..., m (i.e., $|X_i|$=3) and jobs from X_m are scheduled as the last one. Therefore, for the schedule π, we have the following sequence of performed jobs (E_1, X_1, E_2, X_2,..., E_i, X_i,..., E_m, X_m), where $E_i = \{(mB)(i-1)+1, \dots, mBi\}$ and $|X_i|$=3 for i=1,..., m. For such schedule π, we have $p_{l_i}^{(1)}(v) < y$ and the contribution of partition jobs to the criterion value is minimal.

Let us now consider a schedule π' that is different than π, but from the reduction follows for π' that before each enforcer job $e_{l_i} \in l_i$ at last Ni - 4 jobs must be scheduled, otherwise $C_{\max}(\pi') > C_{\max}^{(1)}(\pi') > y$. However, in a contradiction to π, let us assume that in π' there exists a set of enforcer jobs E_i ($i \in \{2,..., m\}$) that is preceded by less than Ni - 4 jobs, thereby by less than $3(i-1)$ partition jobs. In other words, for such E_i there exists a corresponding set X_{i-1} for which $|X_{i-1}| < 3$, thus, $C_{\max}^{1}(\pi') > C_{\max}^{(1)}(\pi) + b_j^{(1)}$, where $j \in \{1,..., 3m\}$. On this basis, we can estimate that $C_{\max}(\pi') > C_{\max}^{(1)}(\pi') \geq y$.

Thus, any schedule π' that is not consistent with schedules such as π cannot be optimal. Therefore, there exists an optimal solution π for DPAE, where jobs must be scheduled as follows: (E_1, X_1, E_2, X_2,..., E_i, X_i,..., E_m, X_m), where $E_i = \{(mB)(i-1)+1, \dots, mBi\}$ and $|X_i| = 3$ for i = 1,..., m. The sequence of jobs within each set X_i is arbitrary as well as in E_i. To make the calculations easier, we renumber the jobs in these sets, i.e., $X_i = \{3i\text{-}2, 3i\text{-}1, 3i\}$ for i = 1,..., m. In the further part, we consider only schedules consistent with π.

Let us introduce useful expressions: $V^{(z)}(X_i)$ and $W^{(z)}(E_i)$ denote the sum of processing times of the partition jobs from X_i and the enforcer jobs from E_i on M_z (z={1,2}), respectively, that are ordered according to π. Based on the reduction, these expressions are defined as follows:

$$V^{(1)}(X_i) = 3mBNb(m+1+i),$$

$$V^{(2)}(X_i) = mBD_i + 4mN(m+1-i)\sum\nolimits_{q \in X_i} x_q + 8mx_{3i-2} + 4mx_{3i-1},$$

$$W^{(1)}(E_i) = mBD_{i-1} + 4mN(m+1-(i-1))B + 3mB,$$

$$W^{(2)}(E_i) = 3mBNb(m+1+i)$$

Since $B/4 < x_q < B/2$ (for i = 1,..., $3m$), then $V^{(2)}(X_i)$ can be estimated as follows:

$$mBD_i + 4mN(m+1-i)\sum\nolimits_{q \in X_i} x_q + 3mB < V^{(2)} X_i$$

$$< mBD_i + 4mN(m+1-i)\sum\nolimits_{q \in X_i} x_q + 6mB$$

On this basis, the completion times of the last scheduled jobs in E_i and X_i on M_1 are:

$$C_{E_i}^{(1)} = \sum\nolimits_{l=1}^{i-1} \left(W^{(1)}(E_l) + V^{(1)}(X_l) \right) + W^{(1)}(E_i),$$
$$C_{X_i}^{(1)} = \sum\nolimits_{l=1}^{i} \left(W^{(1)}(E_l) + V^{(1)}(X_l) \right),$$

for i=1,..., m, where $C_{X_0}^{(1)} = 0$, and on M_2:

$$C_{E_i}^{(2)} = \max\left\{ C_{E_i}^{(1)}, C_{X_{i-1}}^{(2)} \right\} + W^{(2)}(E_i),$$
$$C_{X_i}^{(2)} = \max\left\{ C_{X_i}^{(1)}, C_{E_i}^{(2)} \right\} + V^{(2)}(X_i),$$

for i=1,..., m, where $C_{X_0}^{(2)} = 0$.

Based on the above considerations, it can be shown that the answer for DPAE is *yes* (i.e., $C_{\max}(\pi) = C^{(2)}{}_{X_m} \le y$ **if and only if** the answer for 3PP also is *yes*. However, due to the lack of space the rest of the proof is omitted. □

3.2 THE LEARNING EFFECT

The proof of the strong NP-hardness for $F2|LE|C_{\max}$ is similar to the previous one.

Theorem 2 *The problem* $F2|LE|C_{\max}$ *is strongly NP-hard.*

Proof. A decision version of the problem $F2|LE|C_{\max}$, (Decision Problem with Learning Effect, DPLE) is defined as follows: *Does there exist such a schedule* π *of jobs on the processors* M_1 and M_2 *for which* $C_{\max}(\pi) \le y$?

The pseudopolynomial reduction from 3PP to DPLE is given as follows. The constructed instance of DPLE contains 3m *partition* jobs and m^2B *enforcer* jobs. The parameters of partition jobs are:

$a_j^{(1)} = mBN(m+1)b - 2mBb$ $\qquad g_j^{(1)} = mN$

$b_j^{(1)} = mBb$

$a_j^{(2)} = 4mBN(m+1) + 4m(m+1)Nx_j - 12mx_j$ $\qquad g_j^{(2)} = mN$

$b_j^{(2)} = 4m(B + x_j)$

for j=1,..., 3m and the enforcer jobs are defined as follows:

$a_{l_i}^{(1)} = D_{i-1} + 4N\left(m+1-(i-1)\right) + 3 + yg_{l_i}^{(1)};$ $\qquad b_{l_i}^{(1)} = y;$ $\qquad g_{l_i}^{(1)} = N(i-1);$

$a_{l_i}^{(2)} = 3Nb\left(m+1-i\right);$ $\qquad b_{l_i}^{(2)} = 0;$ $\qquad g_{l_i}^{(1)} = 0;$

where $l_i = (mB)(i-1)+1,\ldots,mBi$ (for i = 1,..., m), $N = (mB+3)$, $b = N^2m^3$, $D_i = 12N(m+1-i)$ and

$$y = mB\left[\sum_{i=1}^{m} \left(\left[D_{i-1} + 4N\left(m+1-(i-1)\right) + 3 \right] + \left[3Nb\left(m+1-i\right) \right] \right) + \left[D_m + 4N + 3 \right] + 3m \right].$$

For the given instance of DPLE, we will provide the property of an optimal solution that is similar to the previous proof. Since the considered problem is a permutation flowshop, then the solution can be defined by the sequence of jobs (permutation) π.

At first observe that from the reduction follows that before each enforcer job $e_{l_i} \in l_i$ minimum $N(i-1)$ jobs have to be scheduled, otherwise $C_{\max}^{(1)}(\pi) > y$, where $C_{\max}^{(1)}(\pi)$ is the completion time of jobs on M_1. Therefore, processing times of enforcer jobs for π are lower than y if they are constant in reference to v, i.e., $p_{l_i}^{(1)}(v) = a_{l_i}^{(1)}$.

On the other hand, $p_j^{(1)}(v) = p^{(1)}(v)$ ($j = 1,..., n$) and it is non-increasing in reference to v. Thus, it is easy to notice that $C_{\max}^{(1)}(\pi)$ is smaller if partition jobs are performed after as many enforcer jobs as it is possible, but keeping $C_{\max}^{(1)}(\pi) < y$, thereby $p_{l_i}^{(1)}(v) = a_{l_i}^{(1)}$ for $i = 1,..., m$.

Using inductive principle, we can easily prove that the completion time of jobs on M_1 is minimum for such a schedule π, which satisfy the following conditions. First, enforcer jobs are partitioned into m sets E_i, such that E_i = {$mB(i$-1)+1, ..., mBi} for $i = 1,..., m$ and jobs from E_i are scheduled one after another. Since jobs from E_i are identical they are scheduled in an arbitrary order within E_i.

Second, between each pair E_i and $E_{\text{i}+1}$ for $i = 1,..., m$-1 exactly one set X_i of partition jobs is scheduled, such that $|X_\text{i}|$=3 for i=1,..., m.

On this basis, jobs in π are scheduled as follows: (E_1, X_1, E_2, X_2,..., E_i, X_i,..., E_m, X_m) where $E_i = \{(mB)(i-1)+1,\ldots,mBi\}$ and $|X_i| = 3$ for $i = 1,..., m$. The sequence of jobs within each set X_i is arbitrary, however, to make the calculations easier, renumber the jobs in these sets, i.e., X_i = {3i-2, 3i-1, 3i} for $i = 1,..., m$.

The criterion value for π can be estimated as follows:

$$C_{\max}(\pi) > C_{\max}^{(1)}(\pi) = y - mB(D_m + 4n + 3 + 3m).$$

Similar as in the proof to Theorem 1, consider a schedule π', which is different than π. Thus, we have $C_{\max}^{(1)}(\pi') \geq C_{\max}^{(1)} + b_j^{(1)}$ (for $j \in$ {1,..., 3m}, hence $C_{\max}(\pi') > C_{\max}^{(1)}(\pi') \geq C_{\max}^{(1)}(\pi) + b_j^{(1)} > y$. Therefore, we conclude that the optimal solution of DAEF is consistent with π. Hence, in the further part, we consider only schedules as π.

Now we introduce useful expressions. Let $V^{(z)}(X_i)$ and $W^{(z)}(E_i)$ denote the sum of processing times of partition jobs from X_i and enforcer jobs from E_i, respectively, on M_z for π (z = {1, 2}). Based on the reduction, these expressions are defined as follows:

$$V^{(1)}(X_i) = 3mBNb(m+1-i),$$
$$V^{(2)}(X_i) = mBD_i + 4mN(m+1-i)\sum\nolimits_{q \in X_i} x_q + 8mx_{3i-2} + 4mx_{3i-1},$$

$$W^{(1)}(E_i) = mBD_{i-1} + 4mN(m+1-(i-1))B + 3mB,$$
$$W^{(2)}(E_i) = 3mBNb(m+1-i),$$

for $i = 1,..., m$. Note that $V^{(1)}(X_i)=W^{(2)}(E_\mathrm{i})$ for $i = 1,..., m$.

The rest of the proof is exactly the same as the proof to Theorem 1. □

4. ALGORITHMS

In this section, we propose Fast NEH algorithm (denoted as FNEH) with complexity O(mn^2) that is the improved version of the classical NEH with complexity O(mn^3) ([4]). The proposed FNEH is presented (Algorithm 1).

Algorithm 1 Fast NEH

```
Determine initial solution $\pi_{init}, \pi^*=\emptyset;$ $q_v^{(z)} = 0$ For $v = 2,..., n+1$ and $z = 1,..., m$ + 1,
    $f_v^{(z)} = 0$ For $v = 1,..., n$ and $z = 0,..., m$, $C_v^{(z)} = 0$ For $v = 1,..., n$ and $z = 0,..., m$,
For $i = 1$ To $n$
   Insert job $\pi_{init}(i)$ in the last (i.e., $i$th) position in $\pi^*$
   Assign $\pi = \pi^*$
   For $z = 1$ To $m$
      $C_i^{(z)} = \max\{C_{i-1}^{(z)}, C_i^{(z-1)}\} + p_{\pi^*(i)}^{(z)}(i)$
   Assign $v = i - 1$ and $C^* = C_i^{(m)}$
   For $v = i - 1$ To $1$
      Swap jobs in positions $v$ and $v+1$ in the permutation $\pi$
      For $z = 1$ To $m$
         $f_v^{(z)} = \max\{C_{v-1}^{(z)}, f_v^{(z-1)}\} + p_{\pi(v)}^{(z)}(v)$
      For $z = m$ To $1$
         $q_{v+1}^{(z)} = \max\{q_{v+2}^{(z)}, q_{v+1}^{(z+1)}\} + p_{\pi(v+1)}^{(z)}(v+1)$
      $C_{\max} = \max_{1 \le z \le m}\{f_v^{(z)} + q_{v+1}^{(z)}\}$
      If $C_{\max} < C^*$   Then
         Assign $v^* = v$ and $C^* = C_{\max}$
   If $v^* \neq i$   Then
      Insert job $\pi_{init}(i)$ in position $v^*$ in the permutation $\pi^*$
      For $v = v^*$ To $i$
         For $z = 1$ To $m$
            $C_v^{(z)} = \max\{C_{v-1}^{(z)}, C_v^{(z-1)}\} + p_{\pi^*(v)}^{(z)}(v)$
$\pi^*$ is the given solution
```

5. NUMERICAL EXPERIMENT

The algorithm FNEH is designed for the general problem $Fm| p_j^{(z)}(v) |C_{\max}$, i.e., processing times are arbitrary functions dependent on a job position in a sequence $p_j^{(z)}(v)$ and the fixed number of processors m.

The proposed algorithm FNEH is evaluated for the following problem sizes n=10 and $m = \{2, 5, 10\}$. For each pair of n and m, 100 random instances were generated from the uniform distribution in the following ranges of parameters: $p_j^{(z)}(v) \in [1, 10]$ or [1, 30], where $z \in \{1,..., m\}$ and $v = 1,..., n$ such that processing times are non-decreasing. The initial solution of FNEH is a random permutation.[1]

The considered FNEH is evaluated, for each instance I, according to the relative error $\delta_A(I)$ that is calculated in the following way: $\delta_A(I) = \left(\frac{C_{\max}(\pi_I^A)}{C_{\max}(\pi_I^*)} - 1 \right) * 100\%$, where $C_{\max}(\pi_I^A)$ denotes the criterion value provided by FNEH for instance I and $C_{\max}(\pi_I^*)$ is the optimal solution of instance I (for n = 10).

The results concerning mean and maximum relative errors provided by the analysed algorithms and their mean running times are presented in Table 1 and 2, respectively.

It can be seen in Table 1 that the mean and maximum relative errors for FNEH do not exceed 14% and 34%, respectively. Thus, the proposed algorithm is relevant to provide an initial solution for more advanced methods such as tabu search. Furthermore, the running times of FNEH are clearly shorten than for the standard NEH, i.e., for n = 100 FNEH is over 30 times faster than NEH (see Table 2).

Table 1. Mean and maximum (in square brackets) relative percentage errors of the algorithms

n	m	$p_j^{(z)}(v)$	FNEH	
10	2	[1, 10]	13.42	[33.53]
		[1, 30]	12.45	[28.34]
	5	[1, 10]	10.22	[21.04]
		[1, 30]	10.20	[24.02]
	10	[1, 10]	6.86	[14.50]
		[1, 30]	7.62	[15.46]

It can be seen in Table 1 that the mean and maximum relative errors for FNEH do not exceed 14% and 34%, respectively. Thus, the proposed algorithm is relevant to

[1] The algorithms were coded in C++ and simulations were run on PC, Processor Intel®Core™ i7-930 2.80 GHz and 4GB RAM.

provide an initial solution for more advanced methods such as tabu search. Furthermore, the running times of FNEH are clearly shorten than for the standard NEH, i.e., for n = 100 FNEH is over 30 times faster than NEH (see Table 2).

Table 2. Mean running times (in seconds) of the algorithms for $p_j^{(z)}(v) \in [1, 10]$

n	m	FNEH	NEH
10	2	<0.001	<0.001
	5	<0.001	<0.001
	10	<0.001	<0.001
100	2	0.001	0.025
	5	0.002	0.074
	10	0.005	0.158

6. CONCLUSIONS

In this work, we showed that the makespan minimization problem in the two-processor flowshop environment becomes strongly NP-hard if the processing time of a job is described by a piecewise linear function dependent on its position in a sequence. Moreover, we constructed the fast NEH with complexity $O(mn^2)$. Efficiency of the proposed method was numerically analysed for the problem with the aging effect. The numerical analysis revealed that Fast NEH is characterized by acceptable relative errors and short running times that are significantly shorter than for the standard NEH.

Our further research will focus on a construction of metaheuristic algorithms for the analysed problems.

REFERENCES

[1] BISKUP D., *A state-of-the-art review on scheduling with learning effects*. European Journal of Operational Research, Vol. 188, 2008, pp. 315–329.

[2] GAREY M.R. and JOHNSON D.S., Computers and intractability: *A guide to the theory of NP-completeness*. Freeman: San Francisco, 1979.

[3] GUPTA J.N.D., STAFFORD Jr E.F., *Flowshop scheduling research after five decades*. European Journal of Operational Research, Vol. 169, 2006, pp. 699–711.

[4] NAWAZ M., ENSCORE Jr E.E. and HAM I.A., *A heuristic algorithm for m-machine, n-jobs Flowshop sequencing problem*. Omega, Vol. 11, 1983, pp. 91–95.

[5] RUDEK A., RUDEK R., *A note on optimization in deteriorating systems using scheduling problems with the aging effect and resource allocation models*. Computers & Mathematics with Applications, 2011, doi:10.1016/j.camwa.2011.06.030 (in press).

[6] RUDEK R., *Computational complexity and solution algorithms for flowshop scheduling problems with the learning effect*. Computers & Industrial Engineering, Vol. 61, 2011, pp. 20–31.

[7] TAILLARD E.D., *Some efficient heuristic methods for the flow shop sequencing problem*. European Journal of Operational Research, Vol. 47, 1990, pp. 65–74.

scheduling, learning effect,
computational complexity, algorithm

Radosław RUDEK*, Agnieszka RUDEK**

EXACT AND APPROXIMATION ALGORITHMS FOR A SCHEDULING PROBLEM WITH LEARNING

In many real-life cases the efficiency of a processor can change due to its learning. Therefore, new and more precise models have been proposed that take into consideration the varying nature of processors. On the other hand, the existing solution algorithms are inefficient for these new models. It implies that new methods have to be proposed to manage the real-life problems. Since the scheduling problems with learning are new in the scheduling theory, thus, the significant number of these problems have no efficient solution algorithms. Therefore, in this work, we provide exact methods such as dynamic programming. Furthermore, we also propose fast approximation algorithm NEH that have lower complexity than its standard version. Numerical experiments revealed the high efficiency of the algorithms.

1. INTRODUCTION

The learning effect usually causes decreasing time or cost of processed tasks (e.g., [2], [3]), therefore, it has a significant impact on the objectives of manufacturing and computer systems. Biskup [1] and Cheng and Wang [4] were among the pioneers who showed that objectives of a system can be improved if the presence of learning is taken into consideration during determining the sequence of processed tasks. Thus, this direction of research has attracted particular attention, since the proposed approach does not interfere in the system, but allows to utilize its learning ability, e.g., [2], [7], [11].

To exploit the learning effect efficiently, it is crucial to precisely describe this phenomenon. In scheduling theory, it is modelled by the processing time of a job (task)

* Wrocław University of Economics, Poland, e-mail: radoslaw.rudek@ue.wroc.pl

** Wrocław University of Technology, Poland, e-mail: agnieszka.wielgus@pwr.wroc.pl

that is expressed as a non-increasing function (called learning curve) of an experience possessed by a processor (system). The experience is usually assumed to be equivalent to the number of previously performed jobs or to the sum of their normal processing times, where the normal processing time of a job is defined as the time required to process it if no learning exists, i.e., when the system is not learnt. More details concerning the differences between these two approaches as well as the discussion on their application to model real-life systems are presented in [2].

Although the single processor total weighted completion time scheduling problem with the sum-of-processing time learning model was introduced into scheduling over three years ago (see [2] and [7] for survey) and it was analysed in the number of papers, its complexity status is not determined yet. Moreover, in the literature there is lack of solution algorithms for this problem. Therefore, to fill this gap, in this work, we will prove the considered problem is at least NP-hard. Moreover, we will construct a pseudopolynomial time algorithm based on dynamic programming that optimally solves the problem with step learning functions. Furthermore, for the general problem (where job processing times are described by stepwise functions dependent on the sum of the normal job processing times) efficient approximation algorithms are proposed, *inter alia*, fast version of NEH.

The remainder of this work is organized as follows. The problem is formulated in the next section, whereas its computational status is determined in Section 2. The exact and approximation algorithms dedicated for the considered problem are presented subsequently and their analysis is provided in Section 5. Finally, the last section concludes the work.

2. PROBLEM FORMULATION

There is given a single processor (an autonomous agent, an algorithm, a human, an intelligent system or a learning system in general) and a set $J = \{1,\ldots, n\}$ of n jobs (e.g., tasks, packets or products) that have to be performed by the processor; there are no precedence constraints between jobs. The processor is continuously available and can process at most one job at a time. Once it begins processing a job it will continue until this job is finished.

Each job j is characterized by its weight parameter w_j and the processing time of job j if it is scheduled as the vth in a sequence is given as follows:

$$\widetilde{p}_j(v) = p_j \cdot f\left(\sum_{l=1}^{v-1} p_{[l]}\right), \tag{1}$$

where p_j is the normal processing time that is the time required to perform the job if the processor is not influenced by learning (i.e., $p_j = p_j(1)$) and f : $[1,+\infty) \to (0, 1]$ is the non-increasing function (learning curve) common for all jobs that depends on the sum of the normal processing times of jobs performed before job j, i.e., $\sum_{l=1}^{v-1} p_{[l]}$, where $p_{[l]}$ denotes the normal processing time of a job scheduled in the lth position in a sequence.

In particular, we focus on the following function characterizing the learning ability of the processor:

$$f(x) = \begin{cases} 1, & x < g \\ \alpha, & g \le x \end{cases}, \tag{2}$$

where $g \in \left(0, \sum_{j=1}^{n} p_j\right)$ is the learning threshold and α ∈ (0; 1] is a learning index common for all jobs, which determines the value of the job processing times after the processor is learnt, i.e., the percentage value of the normal processing time. Note that α = 1 means that there is no learning and the job processing times are constant. The value (1 – α) can be perceived as the learning ratio of the processor (the higher value the better learning performances of the processor). The considered shape of the learning curve characterizes *inter alia* computer systems working on the basis of machine learning algorithms (see [12]). It also can be used as an approximation of other learning curves in manufacturing or computer systems (see [6] or [8]).

Let $\pi = \langle \pi(1),\ldots,\pi(i),\ldots\pi(n) \rangle$ denote the sequence of jobs (permutation of the elements of the set J), where $\pi(i)$ is the job processed in position i in this sequence. By Π we will denote the set of all such permutations. For the given sequence (permutation) $\pi \in \Pi$, we can easily determine the completion time $C_{\pi(i)}$ of a job placed in the ith position in π from the following formulae:

$$C_{\pi(i)} = C_{\pi(i-1)} + p_{\pi(i)}(i) = \sum_{l=1}^{i} p_{\pi(l)}(l), \tag{3}$$

where $C_{\pi(0)} = 0$ and according to (1), we have $p_{\pi(i)} = p_{\pi(i)} \cdot f\left(\sum_{l=1}^{i-1} p_{[l]}\right)$.

The objective is to find a sequence (schedule) π of jobs on the single processor, which minimizes the total weighted completion time criterion:

$$TWC(\pi) = \sum_{i=1}^{n} w_{\pi(i)} C_{\pi(i)}. \tag{4}$$

Formally the optimal schedule $\pi^* \in \Pi$ for the considered minimization objective is defined as follows $\pi^* = \arg\min_{\pi\in\Pi}\{TWC(\pi)\}$.

For convenience and to keep an elegant description of the considered problems we will use the standard three field notation scheme $X \mid Y \mid Z$, where X describes the machine environment, Y describes job characteristics and constraints and Z represents the minimization objectives. According to this notation, the problems will be denoted as $1 \mid LE \mid \sum w_j C_j$.

3. COMPUTATIONAL COMPLEXITY

In this section, we show that the considered scheduling problem with learning $1 \mid LE \mid \sum w_j C_j$ is NP-hard. To prove it, we transform EQUAL CARDINALITY PARTITION problem [5], that is known to be NP-complete, to the decision version of the considered scheduling problem with learning.

Theorem 1 *The problem* $1 \mid LE \mid \sum w_j C_j$ *is NP-hard.*

Proof. At first the definition of EQUAL CARDINALITY PARTITION problem [5] is given.

EQUAL CARDINALITY PARTITION (ECP) ([5]): There are given positive integers m, B and $x_1,\ldots, x_{2m}$ of $2m$ positive integers satisfying $\sum_{q=1}^{2m} x_q = 2B$ for $q = 1,\ldots, 2m$. *Does there exist a partition of the set* $X = \{1,.., 2\mathrm{m}\}$ *into two disjoint subsets* X_1 *and* X_2 *such that* $\sum_{q \in X1} x_q = \sum_{q \in X2} x_q = B$ *and* $|X_1| = |X_2| = m$*?*

The decision version of the scheduling problem $1 \mid LE \mid \sum w_j C_j$ is given as follows (DPTWC): *Does there exist such a schedule* π *of jobs on the single processor for which the criterion value* $TWC(\pi) = \sum_{i=1}^{n} w_{\pi(i)} C_{\pi(i)}$ *is not greater than the given value y?*

On this basis, the polynomial time reduction from ECP to DPTWC is constructed:

$$n = 2m, \qquad p_j = A + x_j, \qquad w_j = A + x_j, \qquad g = mA + B,$$

$$f(x) = \begin{cases} 1, & x < g \\ \alpha = 1/2, & g \le x \end{cases},$$

for $j = 1,\ldots, 2m$ and

$$y = \frac{1}{2}(1+\alpha)\left[\left(m^2 + m\right)A^2 + 2A(m+1)B + 2B^2\right] + (mA + B)^2,$$

where $A = 4B^2$. Obviously, this transformation is polynomial.

Observe that each element from X corresponds to one job from J. Therefore, we construct an arbitrary schedule π for DPTWC on the basis of elements from X. Note that for the considered scheduling problem two states of the processor can be distinguished: before it is learnt and after that. Therefore, for any schedule π the set of all elements X can be partitioned into two disjoint subsets X_1 and X_2 that correspond to jobs J_1 and J_2 that are performed before and after the processor is learnt, respectively.

The further part of the proof is omitted due to the lack of space. □

4. ALGORITHMS

In this section, we construct a dynamic programming pseudopolynomial time algorithm that solves optimally the considered problem $1 | LE | \sum w_j C_j$. Furthermore, the fast NEH approximation algorithm is also provided.

Algorithm 1 Dynamic Programming (DP)

```
Initialize $TWC^* = +\infty$ and $\pi^* = \emptyset$
Schedule jobs within $J$ according to non-increasing $p_j = w_j$
For each job $q \in J$
    For $S_q = \max\{g - p_q, 0\}$ To $\min\{g - 1, \sum_{j=1}^{n} p_j - p_q\}$
    $C_q = S_q + p_q$, $P^B_{\max} = S_q$, $P^A_{\max} = \sum_{j=1}^{n} p_j - C_q$
    For $P^B = 0$ To $P^{\mathrm{B}}_{\max}$
        For $P^A = 0$ To $P^{\mathrm{A}}_{\max}$
            $F(0, P^B, P^A) = +\infty$
    $F(0, 0, 0) = C_q w_q$
    For each $j \in J \setminus \{q\}$
        For $P^B = 0$ To $P^{\mathrm{B}}_{\max}$
            For $P^A = 0$ To $P^{\mathrm{A}}_{\max}$
                Calculate $F(j, P^B, P^A)$:
    $$F(j, P^B, P^A) = \min \begin{cases} F(j-1, P^B - p_j, P^A) + P^B w_j, & if \quad P_B - p_j \geq 0 \\ F(j-1, P^B, P^A - p_j) + (C_q + \alpha \cdot P^A) w_j, & if \quad P_A - p_j \geq 0 \\ F(j-1, P^B, P^A) \end{cases}$$
    $TWC = F(n - 1, P^{\mathrm{B}}_{\max}, P^{\mathrm{A}}_{\max})$
    If $TWC < +\infty$ and $TWC < TWC^*$ Then
        $TWC^* = TWC$
        Find the schedule $\pi^*$ that corresponds to $TWC^*$ using
        backtracking from the state $(n - 1, P^{\mathrm{B}}_{\max}, P^{\mathrm{A}}_{\max})$
$\pi^*$ is the optimal schedule with the criterion value $TWC^*$
```

4.1. DYNAMIC PROGRAMMING

In this part, we construct an exact pseudopolynomial time algorithm that is based on the dynamic programming. Note that the dynamic programming approach assumes the finite number of states, therefore, only integer values of job processing times and of the threshold g are considered in the proposed algorithm. If it is not a case, then each p_j (for $j = 1,\ldots,n$) and g has to be calibrated to an integer value.

Consider an arbitrary schedule of jobs. In such a schedule there can be distinguished job q such that $\sum_{l=1}^{\langle q\rangle-1} p_{[l]} < g$ and $q < \sum_{l=1}^{\langle q\rangle-1} p_{[l]} + p_q$ where $\langle q\rangle$ denotes position of job q in a sequence. Therefore, if job q is fixed, then the schedule can be represented by a partition of other jobs $J\backslash\{q\}$ into two disjoint subsets J_B and J_A of jobs that are scheduled before and after job q, respectively.

In other words, we say that job $j \in J\backslash\{q\}$is assigned to the set J_B if $\sum_{l=1}^{\langle q\rangle-1} p_{[l]} < g$ otherwise it is assigned to the set J_A. For each job $q \in J$ and for each its possible start time $S_q = g - p_q,\ldots, g - 1$ that holds $S_q < g$ and $g \le S_q + p_q$, we calculate the partition of other jobs into two subsets J_B and J_A with the minimal criterion value.

Let jobs within $J\backslash\{q\}$ be renumbered such that $J\backslash\{q\} = \{1,\ldots j,\ldots, n - 1\}$ and P_B and P_A denote the sum of processing times of jobs scheduled before and after job q, respectively. Define a function $F(j. P^B, P^A)$ that represents the minimal value of the criterion for the partial schedules of the first j jobs in $J\backslash\{q\}$ where the sum of processing times of jobs assigned to the set J_i is $P^i = \sum_{j\in J_i} p_j$ for $i \in \{B, A\}$.

Consider a partial schedule in the state $(j. P^B, P^A)$ and corresponding value of the criterion function $F(j, P^B, P^A)$. If the last scheduled job (say j) is assigned to the set J_B then P^B is increased by p_j (i.e., from $P^B - p_j$ to P^B), thereby the criterion value $F(j - 1, P^B - p_j , P^A)$ is increased by $P^B \cdot w_j$ to $F(j, P^B, P^A)$. On the other hand, if job j (scheduled as the last one) is assigned to the set J_A then P^A is increased by p_j (i.e., from $P^A - p_j$ to P^A), thereby the criterion value is increased from $F(j - 1, P^B, P^A - p_j)$ to $F(j, P^B, P^A)$ by $(C_q+\alpha\cdot P^A)\cdot w_j$ that considers the learning effect, where $C_q = S_q+p_q$ is the completion time of job q. The proposed dynamic programming algorithm (DP) is presented.

4.2. FAST NEH

The standard NEH algorithm is based on the method introduced in [9]. It starts with an initial sequence that determines the order of tasks that are inserted subsequently into the resulting solution. Namely, in each iteration the algorithm gets the first task from the initial sequence and adds it to the recent partial sequence of tasks such that it is inserted into a position in this partial sequence that minimizes the criterion value and next this task is removed from the initial sequence. This new partial sequence is the starting sequence in the next iteration of NEH. This process is continued until the initial sequence is empty.

The computational complexity of the primary NEH algorithm is $O(n^3)$ whereas the proposed Fast NEH is $O(n^2)$. The formal description of Fast NEH (FNEH) is given (see Algorithm 2).

Algorithm 2 Fast NEH (FNEH)

```
Initialize:
Determine initial solution $\pi_{init}$, $\pi^*=\langle \pi_{init}(1)\rangle$, $P_{init} = p_{\pi(1)}$, $P = 0$, $W = 0$
For $i = 2$ To $n$
    Insert job $\pi_{init}(i)$ in the last (i.e., $i$th) position in $\pi^*$
    $P_{init} = P_{init} + p_{\pi^*(i)}$
    Calculate the criterion value $TWC$ for the permutation $\pi^*$ and
    store $TWC^* = TWC$ and $v^* = i$
    For $v = i - 1$ To $1$
        If $v \neq i - 1$ Then
            $P = P - p_{\pi^*(v)}$ and $W = W + w_{\pi^*(v+2)}$
        Calculate the criterion $TWC$ for the new permutation
        $\langle \pi^*(1), \ldots, \pi^*(v-1), \pi^*(i), \pi^*(v), \ldots, \pi^*(i-1) \rangle$ as follows:
        $TWC = TWC - p'_{\pi^*(v)}(P) \cdot (W + w_{\pi^*(v)} + w_{\pi^*(v+1)}) - p'_{\pi^*(v)}(P + p_{\pi^*(v)}) \cdot (W + w_{\pi^*(v+1)})$
        $+ p'_{\pi^*(v+1)}(P) \cdot (W + w_{\pi^*(v)} + w_{\pi^*(v+1)}) + p'_{\pi^*(v)}(P + p_{\pi^*(v+1)}) \cdot (W + w_{\pi^*(v)})$
        By swapping jobs in positions $v$ and $v + 1$ in the permutation $\pi^*$
        obtain the new permutation $\langle \pi^*(1), ..., \pi^*(v-1), \pi^*(i), \pi^*(v), ..., \pi^*(i-1) \rangle$
        If $TWC < TWC^*$ Then
            $TWC = TWC^*$ and $v^* = v$
    Insert job $\pi^*(1)$ in position $v^*$ in the new permutation $\pi^*$
    $P = P_{init} - p_{\pi^*(i)}$ and $W = 0$
$\pi^*$ is the given solution with the criterion value $TWC^*$
```

5. NUMERICAL EXPERIMENTS

The analysed algorithms WSPT (scheduling jobs according to the non-decreasing order of $p_j = w_j$) and Fast NEH (FNEH) are evaluated for the following problem sizes n = {10, 25, 50}. For each n, 100 random instances were generated from the uniform distribution in the following ranges of parameters: $pj \in [1, 10]$, $wj \in [1, 10]$; and the parameters α and g were chosen from the following sets: $\alpha \in \{0.2, 0.5, 0.8\}$ and $g' \in \{0.2, 0.5, 0.8\}$, where $g = g' \cdot \sum_{j=1}^{n} p_j$.[1]

[1] The algorithms were coded in C++ and simulations were run on PC, Processor Intel®Core™ i7-930 2.80 GHz and 4GB RAM.

The algorithms WSPT and FNEH are evaluated, for each instance *I*, according to the relative error $\delta_A(I)$ that is calculated in the following way: $\delta_A(I) = \left(\frac{TWC(\pi_I^A)}{TWC(\pi_I^*)} - 1\right) * 100\%$, where $TWC_{\max}(\pi_I^A)$ denotes the criterion value provided by algorithm $A \in$ {WSPT, FNEH} for instance *I* and $TWC_{\max}(\pi_I^*)$ is the optimal solution of instance *I* provided by the dynamic programming exact algorithm (DP).

The results concerning the percentage values of mean and maximum relative errors provided by the analysed algorithms as well as the number of instances for which each algorithm found the optimal criterion value (per 100 instances) and their mean running times are presented in Table 1 and 2, respectively.

Table 1. Percentage values of mean and maximum (in square brackets) relative errors of algorithms and the number of instances (in round brackets) for which each algorithm found the optimal criterion value (where $p_j \in [1, 10]$ and $w_j \in [1, 10]$)

n	g	α	NEH			WSPT		
10	0.2	0.2	9.92	[35.20]	(23)	1.83	[11.59]	(51)
		0.5	3.50	[15.12]	(41)	0.61	[5.57]	(69)
		0.8	0.83	[5.48]	(53)	0.16	[2.23]	(77)
	0.5	0.2	2.91	[8.90]	(22)	0.54	[4.54]	(52)
		0.5	1.33	[5.19]	(34)	0.30	[2.11]	(65)
		0.8	0.23	[2.02]	(64)	0.08	[0.82]	(81)
	0.8	0.2	0.51	[3.38]	(52)	0.10	[1.24]	(80)
		0.5	0.21	[2.41]	(68)	0.03	[0.96]	(90)
		0.8	0.03	[0.60]	(87)	0.01	[0.04]	(98)
25	0.2	0.2	4.56	[12.83]	(13)	0.49	[2.56]	(31)
		0.5	1.99	[5.80]	(24)	0.41	[3.60]	(40)
		0.8	0.47	[2.16]	(25)	0.16	[0.87]	(46)
	0.5	0.2	1.77	[5.58]	(4)	0.37	[1.68]	(22)
		0.5	0.75	[2.67]	(19)	0.14	[0.91]	(41)
		0.8	0.26	[0.95]	(25)	0.08	[0.50]	(51)
	0.8	0.2	0.25	[1.08]	(25)	0.06	[0.46]	(57)
		0.5	0.12	[0.70]	(37)	0.03	[0.30]	(64)
		0.8	0.02	[0.19]	(62)	0.01	[0.12]	(82)
50	0.2	0.2	2.51	[6.61]	(7)	0.12	[0.84]	(17)
		0.5	1.18	[3.35]	(16)	0.12	[0.92]	(29)
		0.8	0.36	[1.09]	(20)	0.08	[0.42]	(38)
	0.5	0.2	0.78	[2.30]	(4)	0.09	[0.54]	(15)
		0.5	0.48	[1.43]	(11)	0.11	[0.67]	(27)
		0.8	0.17	[0.53]	(16)	0.05	[0.32]	(33)
	0.8	0.2	0.14	[0.37]	(15)	0.03	[0.19]	(33)
		0.5	0.07	[0.38]	(33)	0.01	[0.10]	(59)
		0.8	0.03	[0.13]	(36)	0.01	[0.05]	(64)

Note that WSPT and FNEH provide good results, i.e., their greatest mean relative errors were about 9.92% and 1.83%, respectively, and the maximum relative errors were 35.20% and 11.59%, respectively (see Table 1). Moreover, the running times of WSPT and FNEH were less than 1ms even for instances with 200 jobs (see Table 2). It is worth noticing that the constructed FNEH is over 24 times faster than the primary NEH with complexity $O(n^3)$. Therefore, the WSPT and FNEH can be applied in real-time systems (e.g., adaptive routers), where the calculation time is a crucial criterion.

Table 2. Mean running times (in seconds) of the algorithms ($p_j \in [1, 10]$ and $w_j \in [1, 10]$, $g' = 0.5$ and $\alpha = 0.5$); < 0.001 denotes a value lower than 1ms

n	DP	NEH	FNEH$_{\text{DS}}$
10	0.003	<0.001	<0.001
25	0.128	<0.001	<0.001
50	2.065	<0.001	<0.001
100	32.008	0.004	<0.001
150	176.281	0.012	<0.001
200	497.621	0.024	<0.001

Finally, it can be seen (Table 2), that the constructed DP is a very efficient algorithm that can be applied for systems, where the calculation time is not fundamental (few minutes for 200 jobs).

6. CONCLUSIONS

In this work, we analysed the single processor scheduling problem with the learning effect (where the processing time of each job is a non-increasing function dependent on the sum of the normal processing times of preceding jobs) to minimize the total weighted completion time objective. We showed that the considered problem is at least NP-hard. Furthermore, we constructed the optimal pseudpolynomial time algorithm for the analysed problem. We also provided fast approximation algorithms for the general version of the problem. The extensive numerical analysis revealed that the algorithms are characterized by low relative errors and running times.

Our further research will focus on a construction of more precise approximation algorithms, especially metaheuristics such as simulated annealing and tabu search.

REFERENCES

[1] BISKUP D., *Single–machine scheduling with learning considerations*. European Journal of Operational Research, vol. 115, 1999, 173–178.

[2] BISKUP D., *A state-of-the-art review on scheduling with learning effects*. European Journal of Operational Research, vol. 188, 2008, 315–329.

[3] Buşoniu L. and Babuška R. and De Schutter B., *A comprehensive survey of multi-agent reinforcement learning*. IEEE Transactions On Systems, Man, And Cybernetics – Part C: Applications And Reviews, vol. 38, 2008 pp. 156–172.

[4] CHENG T.C.E. and WANG G., Single *machine scheduling with learning effect considerations.* Annals of Operations Research, vol. 98, 2000, 273–290.

[5] GAREY M.R. and JOHNSON D.S., *Computers and intractability: A guide to the theory of NP-completeness*. Freeman: San Francisco, 1979.

[6] JABER Y.M. and BONNEY M., *The economic manufacture/order quantity (EMQ/EOQ) and the learning curve: Past, present, and future*. International Journal of Production Economics, vol. 59, 1999, 93–102.

[7] JANIAK A. and RUDEK R., *Experience based approach to scheduling problems with the learning effect.* IEEE Transactions on Systems, Man, and Cybernetics – Part A, vol. 39, 2009, 344–357.

[8] JANIAK A. and RUDEK R., *A note on the learning effect in multi-agent optimization*. Expert Systems with Applications, vol. 38, 2011, 5974–5980.

[9] NAWAZ M. and ENSCORE Jr E.E. and HAM I.A., *A heuristic algorithm for m-machine, n-jobs Flow-shop sequencing problem*. Omega, vol. 11, 1983, 91–95.

[10] RUDEK A. and RUDEK R., *A note on optimization in deteriorating systems using scheduling problems with the aging effect and resource allocation models*. Computers & Mathematics with Applications, 2011, doi:10.1016/j.camwa.2011.06.030 (in press).

[11] RUDEK R., *Computational complexity and solution algorithms for flowshop scheduling problems with the learning effect.* Computers & Industrial Engineering, vol. 61, 2011, 20–31.

[12] WHITESON S. and STONE P., *Adaptive job routing and scheduling*. Engineering Applications of Artificial Intelligence, vol. 17, 2004, 855–869.

scheduling, learning effect,
aging effect, computational complexity

Radosław RUDEK*, Agnieszka RUDEK**

COMPUTATIONAL COMPLEXITY OF SCHEDULING PROBLEMS WITH VARYING PROCESSING

Improvement or a degradation of a system can be modelled by job processing times that aredescribed by non-increasing (improvement) or non-decreasing (degradation) functions dependent on the number of previously processed jobs. In this work, we focus on scheduling problems with such varying processing times and the following minimization objectives: the maximum completion time and the maximum lateness. Although the scheduling problems with varying processing times have attracted particular attention of a research society, the computational complexity of some problems has not been determined. Therefore, we fill this gap and we show that these problems are NP-hard.

1. INTRODUCTION

The classical scheduling problems assume that the processing times of jobs are constant. However, in many real-life cases the efficiency of a processor can change due to its aging or learning (e.g., [2], [5], [8]). Therefore, to precisely describe and efficiently solve such problems more advanced models have to be analysed. In the scheduling literature, the phenomena of learning and aging are usually modelled by job processing times dependent on the number of processed jobs, where the relation between the processing time and the number of jobs (function) is non-increasing for learning (see [1], [7]) and non-decreasing for aging (e.g., [6]). Nevertheless, the computational complexity of some scheduling problems with these effects are still an open issue. Therefore, in this work, we analyse computational complexity of some single processor scheduling problems with learning and aging effects with the following minimization objectives: the maximum lateness, the makespan with release dates. We

* Wrocław University of Economics, Poland, e-mail: radoslaw.rudek@ue.wroc.pl
** Wrocław University of Technology, Poland, e-mail: agnieszka.wielgus@pwr.wroc.pl.

prove that these problems are strongly NP-hard. Furthermore, we show that the minimization of the makespan with release dates is equivalent to the minimization of the maximum lateness if job processing times are described by functions dependent on the number of processed jobs. The remainder of this work is organized as follows. Section 2 contains formulation of problems and their computational complexity is determined in Section 3. Finally Section 4 concludes the work.

2. PROBLEM FORMULATION

In this section, we will formulate scheduling problems with two phenomenon: aging (fatigue) and learning.

There is given a single processor and a set $J = \{1,...,n\}$ of n jobs (e.g., tasks, products, cleaned items) that have to be processed by a processor; there are no precedence constraints between jobs. The processor is continuously available and can process at most one job at a time. Once it begins processing a job it will continue until this job is finished. Each job is characterized by its aging/learning curve $p_j(v)$ that describes increasing/decreasing of the time required to perform this job depending on the number of jobs completed before it. In other words, we will say that $p_j(v)$ is a processing time of job j if it is processed as the vth job in a sequence. Moreover, each job j is also characterized by the normal processing time a_j that is the time required to perform the job if the processor is not influenced by aging/learning (i.e., a_j , $p_j(1)$). Other job parameters are the release date r_j that is the time at which the job is available for processing and the due-date d_j when it should be completed.

For the aging effect, the processing time (aging/fatigue curve) of job j is described by a function of its position v in a sequence:

$$p_j(v) = a_j v^{\alpha} \tag{1}$$

where α > 0 is the aging index common for all jobs that describes the aging characteristics. On the other hand, for the learning effect the processing time (learning curve) is given as follows:

$$p_j(v) = a_j - b_j v \tag{2}$$

where b_j is a learning ratio of job j.

Let $\pi = \langle \pi(1),\ldots,\pi(i),\ldots\pi(n) \rangle$ denote the sequence of jobs (permutation of the elements of the set J), where $\pi(i)$ is the job processed in position i in this sequence. By Π we will denote the set of all such permutations. For the given sequence (permutation) π, we can easily determine the completion time $C_{\pi(i)}$ of a job placed in the i-th position in π from the following recursive formulae:

$$C_{\pi(i)} = \max\{C_{\pi(i-1)}, r_{\pi(i)}\} + p_{\pi(i)}(i) \tag{3}$$

where $C_{\pi(0)} = 0$ and the lateness $L_\pi(i)$ is defined as follows:

$$L_{\pi(i)} = C_{\pi(i)} - d_{\pi(i)} \tag{4}$$

We will say that job $\pi(i)$ is late if $L_{\pi(i)} > 0$. The objective is to find such an optimal sequence (schedule) $\pi^* \in \Pi$ of jobs on the single processor, which minimizes one of the following objective functions: the maximum completion time (makespan) $C_{\max} = \max_{i=1,\dots n}\{C_{\pi^*(i)}\}$, (i.e., $C_{\max} = C_{\pi^*(n)}$) and the maximum lateness $L_{\max} = \max_{i=1,\dots n}\{L_{\pi^*(i)}\}$.

Formally the optimal schedule $\pi^* \in \Pi$ for the considered minimization objectives is defined as follows $\pi^* = \arg\min_{\pi\in\Pi}\{C_{\pi(n\}}\}$ and $\pi^* = \arg\min_{\pi\in\Pi}\{\max_{i=1,\dots n}\{L_{\pi(n\}}\}\}$, respectively.

For convenience and to keep an elegant description of the considered problems we will use the standard three field notation scheme $X \mid Y \mid Z$, where X describes the processor environment, Y describes job characteristics and constraints and Z represents the minimization objectives. According to this notation, the problems will be denoted as follows: $1 \mid r_j, ALE \mid C_{\max}$ and $1 \mid ALE \mid L_{\max}$, where $ALE \in \{p_j(v) = a_j v^\alpha, p_j(v) = a_j - b_j v\}$. If $r_j = 0$ for $j = 1,\dots, n$ then it is omitted in the given notation.

3. COMPUTATIONAL COMPLEXITY

In this section, we will prove that the considered problems are strongly NP-hard. First, we will determine the computational complexity of the maximum lateness minimization problem with the aging effect and then with the learning effect. Next we will prove, on the basis of a problem equivalency, the makespan minimization with release dates is also strongly NP-hard with aging/learning models.

3.1. AGING EFFECT

We will show that the problem $1 \mid p_j(v) = a_j v^\alpha \mid L_{\max}$ is strongly NP-hard. At first, we will provide the pseudopolynomial time transformation from the strongly

NP-complete problem 3-PARTITION ([4]) to the decision version of the considered scheduling problem, $1|p_j(v) = a_j v^{\alpha}|L_{\max}$.

3-PARTITION (3PP) ([4): There are given positive integers m, B and $x_1,...,x_{3m}$ of $3m$ positive integers satisfying $\sum_{q=1}^{3m} x_q = mB$ and $B/4 < x_q < B/2$ for $q = 1,..., 3m$. *Does there exist a partition of the set $X = \{1,...,m\}$ into m disjoint subsets $X_1,..., X_m$ such that $\sum_{q \in X_i} x_q = B$ for $i = 1,...,m$?*

The decision version of the scheduling problem with aging, $1|p_j(v) = a_j v^{\alpha}|L_{\max}$ (DSPA) is given as follows: *Does there exist such a schedule π of jobs on the processor for which $L_{\max} \le y$?*

The transformation from 3PP to DSPA is given as follows. The instance of DSPA contains the set $X = \{1,..., 3m\}$ of $3m$ *partition* jobs (constructed on the basis of the elements from the set Y of 3PP) and the set $E = \{1,...., mN\}$ of mN *enforcer* jobs (where $N = mB$). The enforcer jobs can be partitioned into m sets $E_i = \{e_{N(i-1)+1}, \dots, e_{Ni}\}$ for $i = 1,..., m$, such that jobs within each set E_i have the same parameters, i.e., $a_k = a_l$ and $d_k = d_l$ for $k, l \in E_i$ for $i = 1,..., m$.

The parameters of the enforcer jobs are defined as follows:

$$a_{eN(i-1)+1} = \dots = a_{eN\,i} = a_{Ei} = a_E = \frac{1}{m(N+3)},$$

$$d_{eN(i-1)+1} = \dots = d_{eN\,i} = d_{Ei} = \sum_{l=1}^{i-1} (W_l + V_l) + W_i,$$

for i = 1,..., m and α = 1. The parameters of the partition jobs are:

$$a_j = (M + x_j),$$

$$d_j = D = \sum_{i=1}^{m} (W_i + V_i),$$

for $j = 1,..., 3m$, where $N = mB$, $M = (m+1)^2 (N+3)B$, $V_i = 3M(i(N+3)-1) + i(N+3)B - \frac{3}{4}B$,

$$W_i = a_E \left((i-1)N(N+3) + \sum_{l=1}^{N} l \right), \tag{5}$$

for $i = 1,..., m$ and $y = 0$.

Let X_i denote the set of the partition jobs that are processed just after jobs from the set E_i (for $i = 1,..., m$). Define a schedule π^* where jobs are scheduled as follows: (E_1, X_1, E_2, X_2, E_3..., E_i, X_i, E_{i+1}..., X_{m-1}, E_m, X_m), where $X_i=\{3i-2, 3i-1, 3i\}$ for i=1,..., m, if it is not a case we can always renumber the partition jobs. Let $V(X_i)$ and W_i denote the sum of processing times of the partition jobs from X_i and the enforcer jobs from E_i, respectively, for the schedule π^*. Based on the transformation $V(X_i)$ is defined as:

$$V(X_i) = 3M\big(i(N+3)-1\big) + i(N+3)\sum_{q\in X_i} x_q - 2x_{3i-2} - x_{3i-1},$$

for $i = 1,..., m$ and it can be estimated as follows:

$$3M\big(i(N+3)-1\big) + i(N+3)\sum_{q\in X_i} x_q - \frac{3}{2}B < V(X_i) <$$

$$3M\big(i(N+3)-1\big) + i(N+3)\sum_{q\in X_i} x_q - \frac{3}{4}B$$

It is easy to observe that the sum of processing times of the enforcer jobs from the set E_i, i.e., W_i, ($i = 1,..., m$) in schedule π^* is given by (5). The completion time of the last job in E_i is C_{E_i} and of the last job in X_i is C_{X_i} for $i = 1,..., m$.

Let us also define useful inequalities:

$$V(X_i) > V_i - i(N+3)B - \frac{3}{4}B,$$

$$Wi < a_E N\left(\big(m-1\big)\big(N+3\big) + \frac{N+1}{2}\right) < a_E mN(N+3) = N,$$

$$M > m(m+1)(N+3)B + mB + N > \sum_{l=1}^{m}\left(l\big(N+3\big)B + \frac{3}{4}B\right) + W_i,$$

for $i = 1,..., m$. Note also that the processing times of the partition jobs can be estimated as follows $p_j(v) > M$ for $j = 1,..., 3m$ and $v = 1,..., m(N+3)$.

On this basis, we will provide properties of an optimal solution for DSPA, but due to the lack of space the proofs are omitted.

Lemma 1 *The optimal sequence of jobs for the problem* $1|\, p_j(v) = av^{\alpha}$, d_j=d|$L_{\max}$ *is arbitrary.*

Lemma 2 *The problem* $1|\, p_j(v) = av^{\alpha}$ |$L_{\max}$ *can be solved in* $O(n \log n)$ *steps by scheduling jobs according to the non-decreasing order of their due dates* (*the EDD rule*).

Lemma 3 *The problem* $1|\, p_j(v) = a_j v^{\alpha}$ |$C_{\max}$ *can be solved in* $O(n \log n)$ *steps by scheduling jobs according to the non-increasing order of their normal processing times* (*LPT rule*).

Lemma 4 *There is an optimal schedule π, for the given instance of DSPA, in which, before the enforcer jobs from E_i (i = 1,..., m) at last 3(i – 1) partition jobs can be scheduled.*

Lemma 5 *Jobs in each block E_i are processed one after another and between E_i and E_{i+1} exactly 3 partition jobs are scheduled for i = 1,..., m – 1.*

Based on the above lemmas, the following theorem can be proved.

Theorem 1 *The problem* $1|p_j(v)=a_jv|L_{\max}$ *is strongly NP-hard.*

Proof. Based on the given transformation from 3PP to DSPA and on Lemma 5 we construct a schedule π for DSPA, that is given as follows: (E_1, X_1, E_2, X_2, E_3..., E_i, X_i, E_{i+1}..., X_{m-1}, E_m, X_m). Recall that blocks of the enforcer jobs are scheduled according to the EDD rule and the schedule of jobs within each E_i is immaterial and the sequence of jobs within each set X_i is arbitrary.

On this basis, it can be shown that the answer for DSPA is *yes* (i.e., $L_{\max} \le y$) **if and only if** it is *yes* for 3PP (i.e., $\sum_{q\in Y_i} x_q = B$ for i = 1,..., m). Due to the lack of space, the rest of the proof is omitted. □

3.2. LEARNING EFFECT

Cheng and Wang [3] proved that the problem $1|p_j(v)=a_j-b_j\min\{v-1,g_j\}|L_{\max}$ is strongly NP-hard. We will show that already the problem $1|p_j(v)=a_j-b_jv|L_{\max}$ is strongly NP-hard. It will be proved in the similar manner as in case of the problem $1|p_j(v)=a_jv^{\alpha}|L_{\max}$.

At first, we will provide the pseudopolynomial time transformation from the strongly NP-complete problem 3-PARTITION ([4]) to the decision version of the considered scheduling problem, $1|p_j(v)=a_j-b_jv|L_{\max}$.

The decision version of the scheduling problem with learning $1|p_j(v)=a_j-b_jv|L_{\max}$ (DSPL) is given as follows: *Does there exist such a schedule π of jobs on the processor for which $L_{\max} \le y$?*

The pseudopolynomial time transformation from 3PP to DSPL is given. The constructed instance of DSPL contains 3m *partition* jobs and mN *enforcer* jobs partitioned into m sets E_i, such that $E_i = \{e_{N(i-1)+1},\ldots,e_{Ni}\}$ for i = 1,..., m, where $N = mB$ (i.e., number of jobs in each set E_i). The parameters of the enforcer jobs are defined as follows:

$$b_{Ei} = b_E = M(N+1),$$

$$a_{Ei} = a = 2mM(N+3) + 2mN(N+3)b_E,$$

$$d_{Ei} = \sum_{l=1}^{i-1}\left(W_l + V_l\right) + W_i,$$

for $i = 1,..., m$ and of the partition jobs:

$$b_j = (M - x_j),$$

$$a_j = a,$$

$$d_j = D = \sum_{i=1}^{m}\left(W_i + V_i\right),$$

for $j = 1,..., 3m$, where

$$N = mB,$$

$$M = (m+1)^2(N+3)B,$$

$$V_i = 3a - 3M\left(i(N+3)-1\right) + i(N+3)B - \frac{3}{4}B,$$

$$W_i = aN - b_E\left(\left(i-1\right)N(N+3) + \sum_{l=1}^{N} l\right), \tag{6}$$

for $i = 1,..., m$ and $y = 0$.

Let *Xi* denotes the set of the partition jobs that are processed just after jobs from the set E_i (for $i = 1,..., m$). Define a schedule π^*, where jobs are scheduled as follows: (E_1, X_1, E_2, X_2, E_3..., E_i, X_i, E_{i+1}..., X_{m-1}, E_m, X_m), where X_i={3i–2, 3i–1, 3i} for $i = 1,..., m$, if it is not a case we can always renumber the partition jobs. Let $V(X_i)$ and W_i denote the sum of processing times of the partition jobs from X_i and the enforcer jobs from E_i, respectively, for the schedule π^*. Based on the transformation $V(X_i)$ is defined as:

$$V(X_i) = 3a - 3M\left(i(N+3)-1\right) + i(N+3)\sum_{q\in X_i} x_q - 2x_{3i-2} - x_{3i-1},$$

for $i = 1,..., m$ and it can be estimated as follows:

$$3a - 3M\left(i(N+3)-1\right) + i(N+3)\sum_{q\in X_i} x_q - \frac{3}{2}B < V(X_i) <$$

$$3a - 3M\left(i(N+3)-1\right) + i(N+3)\sum_{q\in X_i} x_q - \frac{3}{4}B.$$

It is easy to observe that the sum of processing times of the enforcer jobs from the set E_i, i.e., W_i, ($i = 1,..., m$) in schedule π^* is given by (6). The completion time of the last job in E_i is C_{E_i} and of the last job in X_i is C_{X_i} for $i = 1,..., m$.

Let us also define useful inequalities:

$$V(X_i) > V_i - i(N+3)B - \frac{3}{4}B,$$

$$M > \sum_{l=1}^{m}\left(l(N+3)B + \frac{3}{4}B\right),$$

$$a < b_E mN(N+3) + MmN(N+3) + \sum_{l=1}^{m}\left(l(N+3)B + \frac{3}{4}B\right),$$

for $i = 1,..., m$.

On this basis, we will provide properties of an optimal solution for DSPL, but due to the lack of space the proofs are omitted.

Lemma 6 *The optimal sequence of jobs for the problem* $1|p_j(v) = a_j - bv, d_j = d|L_{\max}$ *is arbitrary.*

Lemma 7 *The problem* $1|p_j(v) = a_j - bv|L_{\max}$ *can be solved in* $O(n \log n)$ *steps by scheduling jobs according to the non-decreasing order of their due dates* (*the EDD rule*).

Lemma 8 *The problem* $1|p_j(v) = a_j - b_j v|C_{\max}$ *can be solved in* $O(n \log n)$ *steps by scheduling jobs according to the non-decreasing order* b_j / a_j *parameters.*

Lemma 9 *There is an optimal schedule* π*, for the given instance of DSPL, in which, before the enforcer jobs from* E_i ($i = 1,..., m$) *at last* $3(i - 1)$ *partition jobs can be scheduled.*

Lemma 10 *Jobs in each block* E_i *are processed one after another and between* E_i *and* E_{i+1} *exactly 3 partition jobs are scheduled for* $i = 1,..., m - 1$.

Theorem 2 *The problem* $1|p_j(v) = a_j - b_j v|L_{\max}$ *is strongly NP-hard.*

Proof. Based on the given transformation from 3PP to DSPL and on Lemma 10 we construct a schedule π for DSPL, that is given as follows: (E_1, X_1, E_2, X_2, E_3..., E_i, X_i, E_{i+1}..., X_{m-1}, E_m, X_m). Recall that blocks of the enforcer jobs are scheduled according to the EDD rule and the schedule of jobs within each E_i is immaterial and the sequence of jobs within each set X_i is arbitrary.

The further part of the proof can be done exactly the same as for Theorem 1. □

3.3. PROBLEM EQUIVALENCY

In classical scheduling theory, the following problems $1||L_{\max}$ and $1|r_j|C_{\max}$ are equivalent with respect to the criterion value. Moreover, an algorithm solving problem $1||L_{\max}$ can be taken as an algorithm solving the problem $1|r_j|C_{\max}$. Now, we will show that this equivalency still holds in the presence of learning and aging, but if the corresponding job processing times are symmetric for the both phenomena. The proof is omitted.

Theorem 3 *The problems* $1|p_j(v)|L_{\max}$ *and* $1|r'_j, p'_j(v)|C'_{\max}$ *are equivalent in the following sense: the optimal schedules are inverse and the criterion values differ only by a constant, if* $p_j(v)$ *is a positive function of a job position and* $p'_j(v) = (n - v + 1)$ *for* $v, j = 1,..., n$.

On this basis, we can easily prove the complexity of the following problems.

Corollary 1 *The problem* $1|r_j, p_j(v) = a_j(n+1-v)|C_{\max}$ *is strongly NP-hard.*

Corollary 2 *The problem* $1|r_j, p_j(v) = a'_j + a_j v,\ a'_j = (c - a_j(n+1)),\ c > a_j n|C_{\max}$ is strongly NP-hard.

4. CONCLUSIONS

In this work, we showed that the minimization of the maximum lateness or of the makespan with release dates is strongly NP-hard even if job processing times are described by functions dependent on a number of processed jobs (i.e., a job position in a sequence). Moreover, we showed that the minimization of the makespan with release dates is equivalent to the minimization of the maximum lateness if job processing times are described by functions dependent on the number of processed jobs.

REFERENCES

[1] BISKUP D., *A state-of-the-art review on scheduling with learning effects*. European Journal of Operational Research, Vol. 188, 2008 pp. 315–329.

[2] BUŞONIU L., BABUŠKA R., De SCHUTTER B., *A comprehensive survey of multiagent reinforcement learning*. IEEE Transactions On Systems, Man, And Cybernetics – Part C: Applications And Reviews, Vol. 38, 2008 pp. 156–172.

[3] CHENG T.C.E., WANG G., *Single machine scheduling with learning effect considerations*. Annals of Operations Research, Vol. 98, 2000 pp. 273–290.

[4] GAREY M.R., JOHNSON D.S., *Computers and intractability: A guide to the theory of NP-completeness.* Freeman: San Francisco, 1979.

[5] JABER Y.M., BONNEY M., *The economic manufacture/order quantity (EMQ/EOQ) and the learning curve: Past, present, and future.* International Journal of Production Economics, Vol. 59, 1999, 93–102.

[6] RUDEK A., RUDEK R., *A note on optimization in deteriorating systems using scheduling problems with the aging effect and resource allocation models.* Computers & Mathematics with Applications, 2011, doi:10.1016/j.camwa.2011.06.030 (in press).

[7] RUDEK R., *Computational complexity and solution algorithms for flowshop scheduling problems with the learning effect*. Computers & Industrial Engineering, Vol. 61, 2011, pp. 20–31.

[8] WHITESON S., STONE P., *Adaptive job routing and scheduling*. Engineering Applications of Artificial Intelligence, Vol. 17, 2004, 855–869.

*Block Packing Problem, Sequence-Pair,
VLSI, design automation, algorithm,
floorplanning*

Adam KURPISZ*, Andrzej KOZIK**

VLSI MODULE PLACEMENT BASED ON RECTANGLE-PACKING BY THE NEURAL NETWORK BASED ALGORITHM

The significance and hardness of the floorplanning in the VLSI physical design caused that much effort have been taken to address this bottleneck. The floorplanning problem can be expressed as a classic rectangle packing problem: given a set of rectangular modules of arbitrary size the goal is to place them on a two-dimensional space without overlapping, subject to minimize the area of a minimum bounding rectangle.

In this work we propose a novel approach based on neural network. We take as a basis the GIT algorithm that iteratively inserts blocks into initially empty solution. Considering the solution space is spanned by Sequence-Pair layout representation, in each step of the GIT algorithm the best solution from the neighborhood have to be selected as a representative.

In our approach we use a neural network to pick such a solution. Neural is trained using previously prepared optimal instances of the problem. Training is based on inverse GIT algorithm method – we iteratively remove modules from optimal placement and train neural network with such partial solutions.

Numerical experiments showed that GIT algorithm with such trained neural network can solve efficiently any instances of rectangle packing problem. We show that a well trained neural network can be used as a technique of choosing the best element from a neighborhood. Such a method seems to be very useful in case of combining it with some heuristic algorithms such as tabu search, what is a topic of our further research.

* Institute of Mathematics and Computer Science Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland, adam.kurpisz@pwr.wroc.pl

** Institute of Computer Engineering, Control and Robotics, Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland, andrzej.kozik@pwr.wroc.pl

1. INTRODUCTION

The significance and hardness of the floorplanning in the VLSI physical design [1, 8] caused that much effort have been taken to address this bottleneck. Recent advances in this field [5] include introduction of strong properties of solution representation, as incremental evaluation of solution neighborhood and, on this basis, successful application of greedy insertion technique.

This floorplanning problem can be expressed as a classic Rectangle (Block) Packing Problem [2]: given a set of rectangular modules, representing parts of integrate circuit, e.g., transistors, cells, (…), of given sizes the goal is to place them on a two-dimensional space without overlapping, subject to minimize the area of a minimum bounding rectangle.

In this work we propose a novel approach based on neural network. We take as a basis the GIT algorithm [5] that iteratively inserts blocks into initially empty solution. Considering the solution space is spanned by Sequence-Pair layout representation, in each step of the GIT algorithm the best solution from the neighborhood have to be selected as a representative. In our approach we use a neural network to pick such a solution. Neural network has been trained using previously prepared optimal instances of this problem. Training is based on inverse to GIT algorithm method – we iteratively remove modules from optimal placement and train neural network with such partial solutions. Numerical experiments clearly showed that GIT algorithm with such trained neural network can efficiently solve the rectangle packing problem.

The rest of the work is organized as follows. The next section formulates the problem and describes Sequence Pair solution representation. Section 3 describes the Neuron-GIT approach. In Section 4 results of numerical experiments are presented and discussed. Finally, Section 5 concludes the work.

2. PROBLEM FORMULATION

There is given a set $B = \{1, \ldots, n\}$ of n rectangular blocks. Each block $i \in B$ is characterized by its width w_i, height h_i and area $a_i = w_i \cdot h_i$. The goal of the Block Packing Problem (BPP) is to find such a packing, i.e., placement coordinates of a bottom-left corner (x_i, y_i) and orientation (horizontally or vertically laid) of each block $i \in B$, that no two blocks overlap and the area of minimum enclosing rectangle of the packing is minimized.

To represent a packing solution and to create a solution space for BPP we make use of a concise and elegant packing representation called Sequence-Pair (SP), introduced by Murata et al. [9]. Sequence-Pair, as well as other known representations, is useful only for reflecting a packing solution into its appropriate coding. A sequence pair, say (Γ_+, Γ_-), is a pair of n-element sequences (permutations), each representing some order of elements of the set B of blocks having orientations and dimensions fixed.

The solution space provided by SP consists of $(n!)^2$ sequence-pairs, each of which can be mapped to a packing in polynomial time, and at least one of which corresponds to an optimal solution. The SP casts BPP as a discrete optimization problem [4], allowing finding a coding that represents an optimal solution, which could be solved by variety of methods developed in the past four decades [3].

3. THE NEURON-GIT ALGORITHM

In this section we present a novel approach to solve BPP. We take as a basis the GIT algorithm that iteratively inserts blocks into initially empty solution. Considering the solution space is spanned by Sequence-Pair layout representation, in each step of the GIT algorithm the best solution from the neighborhood have to be selected as a representative. In our approach we use a neural network to pick such a solution. Neural network is trained using previously prepared optimal instances of the problem. Training is based on inverse to GIT algorithm method – we iteratively remove modules from optimal placement and train neural network with such partial solutions.

The GIT algorithm. The GIT algorithm, presented in [5], resembles the insertion technique, primarily applied to flow-shop scheduling [10].

Let $O = (o_1, \dots, o_n)$ be such an order of blocks from $B = (1, \dots, n)$ that $a_{o_1} \geq a_{o_2} \geq \dots \geq a_{o_n}$. Denote by $\sigma = \langle \sigma_1, \dots, \sigma_n \rangle$ a sequence of partial solutions, where $\sigma_i = (\Gamma_+^i, \Gamma_-^i)$ and Γ_+^i and Γ_-^i are sequences of blocks from $B_i = (o_1, \dots, o_i)$.

The solution is constructed by successively inserting blocks $o_1, \dots, o_n$ into an initially empty solution. In the insertion step, all possible positions of inserting block o_i into σ_{i-1} are evaluated, considering both cases of block orientation. The representative solution (evaluated as best in the neighborhood) creates a base solution σ_i for the next step. The pseudo-code of the GIT algorithm is given as Algorithm 1.

Algorithm 1: GIT (B)

```
Define order $O$
FOR $i$ = 1 TO n DO:
  $\sigma_i$ = EvalNeighborhood$(B^{i-1}, \Gamma_+^{i-1}, \Gamma_-^{i-1}, o_i)$
  $B^i = B^{i-1} \cup \{o_i\}$
ENDFOR
RETURN: $\sigma_n$
```

The EvalNeighborhood function evaluates one of the solutions in the neighborhood of the actual base solution as the best. Classically, it is done by considering some function giving a *score* to each solution from the neighborhood; solution with the greatest score is selected as the representative. The problem is, however, with proper choice of such a function. The still open question is what aspects are crucial in differentiating between good and bad choices. Knowing, that criterion function only is insufficient to build an evaluating function, should the neighborhood search concentrate on sequence pair view on solution or rather on the geometrical view on the packing solution? Trying to answer this question, in this work we propose a novel approach to develop such a scoring function based on neural network.

The Neuron-GIT algorithm. In our approach, we substitute classical scoring function by a neural network. The idea is based on the possibility that a properly developed and learned neural network can approximate unknown function. In our case we use neural network as a *black-box* that classifies solutions from the neighborhood as bad or promising choices from the perspective of future iterations of GIT algorithm.

The learning algorithm. Learning a neural network requires knowledge of the desired output for any input in the training set. In our approach we reverse the choices of the GIT algorithm assuming, its result was an optimal packing solutions.

Let $\sigma_n = (\Gamma_+^n, \Gamma_-^n)$ be an optimal packing of blocks from $B = (1, \dots, n)$. Let $O = (o_1, \dots, o_n)$ be such an order of blocks that $a_{o_1} \geq a_{o_2} \geq \cdots \geq a_{o_n}$; note the GIT proceeds blocks in this order. Knowing σ_n and O we may reverse the steps taken by GIT, obtaining the sequence $\langle \sigma_n, \dots, \sigma_1 \rangle$ of partial solutions (choices) that lead GIT to the optimal solution. Consider the GIT inserts the block o_i into σ_{i-1} picking σ_i as a best solution from the neighborhood. On this basis we can assign to each solution from a neighborhood the desired output – solutions similar to σ_i are labeled as *good*, while all remaining solutions are labeled as *bad*.

Construction of the training set. In case to train our neural network and to test efficiency of our algorithm we prepare some instances of BPP problem. We create sets of modules with different cardinality in case of checking how the neural network and the algorithm cope with it. Each of prepared input set of modules can be placed on a two-dimensional space without overlapping such that wasted space is either equal zero of relatively small in comparison to the size of bounding rectangle. Optimal instances received from given input sets can be divided into two groups. An instance is called slicing [11] if modules can be received by the division of a bounding rectangle of an optimal schedule into smaller rectangles by parallel lines. The operation can be applied to each of the resulting rectangles (slices) with lines perpendicular to the previous set of dividing lines. Slicing can be repeated to any depth alternating the orientation of the dividing lines. A non-slicing is an instance which modules can by constructed with above dividing formula but do not have to. Instances of a floorplan design problem based on non-slicing structures are much more difficult to solve. Slicing structures have more tools to deal with, despite the problem of the floorplanning in the VLSI physical design has found its description rather in non-slicing based structures.

We prepare slicing sets of cardinality from 5 up to 100 modules and total area from 10^4 and 10^6 square units. These instances were received from described above dividing formula from square figure, thus wasted area of optimal solution to these instances is equal zero. Non-slicing instances have also cardinality from 5 to 100. We use some empirical method to get such instances. The method was fully random to get a wide range of instances. This causes that filling coefficient (quotient of bounding rectangle to sum of modules area) is bigger than 1. This also better matches the real floorplan design problem.

Previously prepared sets of modules were used as testing sets for neural network. Thus we have certainty of comparability between differently trained networks. As training set we use other instances prepared in the same way as testing ones. Despite having the knowledge of the proportion between cardinality of training and testing sets we use as the training set only one instance of a problem. This was caused by limited computational power of used machine. Even though, by the construction of the training set, it was very big.

Training is based on inverse to GIT algorithm method – we iteratively remove modules from optimal schedule and train neural network with such partial schedules. To prepare a testing set, denoted as Φ, we follow the following steps.

Algorithm 2: TrainingSet(B)

```
Define order $O$
FOR $i$ = n DOWNTO 1 DO:
 FOR $j$ = 1 TO $i^2$ DO:
   $\left(\Gamma_{+}^{i-1},\Gamma_{-}^{i-1}\right)$=nextPerm$(B^{i-1},\Gamma_{+}^{i-1},\Gamma_{-}^{i-1},o_i)$
   $w_{out}^{j,i-1} = countOutput(B^{i-1},\Gamma_{+}^{i-1},\Gamma_{-}^{i-1},o_i)$
```

```
        $\Phi = \Phi \cup \left(w_{out}^{j,i-1}, \left(\Gamma_{+}^{i-1}, \Gamma_{-}^{i-1}\right)\right)$
    ENDFOR
    $B^{i-1} = B^{i} - \{o_i\}$
ENDFOR
RETURN: $\Phi$
```

Where `nextPerm` is a function which changes the sequence-pair of partial solution to the next permutation in neighborhood. $countOutput$ clearly counts the desired output.

Neural network. Now we will describe the properties of neural network used in our algorithm. Since slicing based structures and non-slicing are different we use different settings for each one. Differences are especially in input neurons, both in cardinality and the form. We use a unidirectional neural network with backward propagation of errors and supervised learning method. As the activation function we use hyperbolic tangent sigmoid transfer function.

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{1}$$

Since backpropagation requires that the activation function used by the artificial neurons to be differentiable, we have

$$\tanh'(x) = \frac{4}{(e^{x} + e^{-x})^{2}} \tag{2}$$

Kalman and Kwasny in [6] show that the fastest learning process is under hyperbolic tangent sigmoid transfer function. In [7] it was shown that small value of derivative function makes a learning process slower, hyperbolic tangent also fits this assumption.

As a backward propagation of errors we use the Levenberg–Marquardt Algorithm which is widely known to be very good for the unidirectional neural network. This algorithm modifies weights of neuron inputs in groups. This combines the convergence algorithm, Gauss–Newton near the minimum, with the fastest descent, which quickly reduces the error when the solution is far.

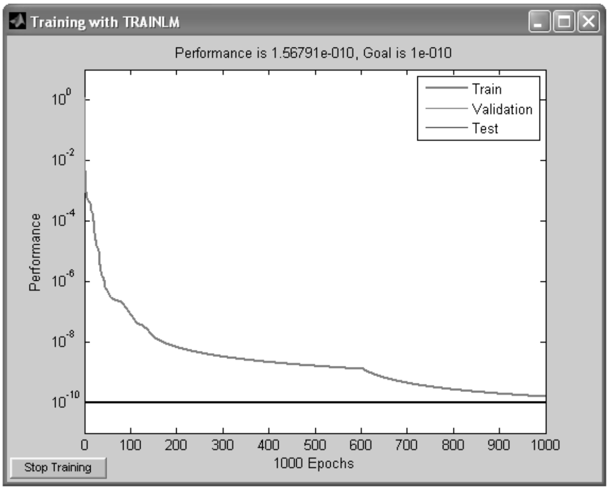


Fig. 1. Decrease the gradient while training with Levenberg–Marquardt Algorithm

In our algorithm we use a neural network with three input neurons. These neurons are described as follow:

- first input neuron:

$$w_1 = \frac{a_0}{2\sum_{i \in B} x_i y_i} \tag{3}$$

Where a_0 denotes the area of bounding rectangle of input instance,

- second input neuron:

$$w_2 = \frac{i}{n} \tag{4}$$

Describes the quotient of number of modules in partial instance to total number of modules,

- third input neuron:

$$w_3 = \frac{w_i}{h_i} \tag{5}$$

Describes the quotient of width of i-th module to its height. The i-th module is the smallest module over all modules in partial instance.

We set numbers of hidden layer to one with eight neurons. Number of epoch was set to 1000 and performance goal was set to 10^{-5}. The desired output which is calculated for every input instance has the form:

$$w_{out} = \frac{a_C - |a_0 - a_C|}{\sum_{i \in B} x_i y_i} \tag{6}$$

Where a_C denotes the area of bounding rectangle of partial instance received by removing proper number of modules from optimal schedule, a_0 as before.

It is easily seen that w_{out} gets the largest value for partial instance that is received from optimal one by removing proper number of modules. Any other instance with such module cardinality has smaller w_{out} factor. Thus we can label the best partial solutions as good and all remaining as bad ones.

4. COMPUTATIONAL RESULTS AND DISCUSSION

Having the neural network trained with the learning set described above, we use our Neuron-GIT algorithm to solve testing instances. The following results were received.

As we can see from above table the filling coefficient is very high, more than 85%. Non-slicing based structures as we supposed, turn out to be more difficult for our algorithm.

Table 1. Statistics of the results

Type of structure	Average value of filling percentage(VoFP)	Variance VoFP	Maximum VoFP	Minimum VoFP
Slicing	0,89	0,036	1,00	0,52
Non-slicing	0,86	0,007	1,00	0,76
All	0,88	0,020	1,00	0,52

Worth noticing is that algorithm over non-slicing instances is much more stable and the variance is smaller for such sets of modules. There were no solutions worse than 76% percent of filling. Following figures show sample solution received with our algorithm. Both slicing and non-slicing instances are presented.

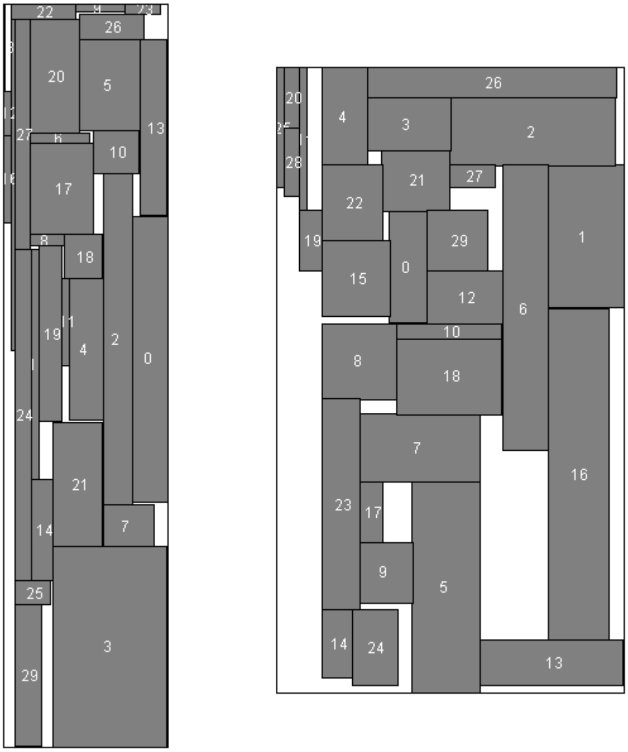


Fig. 2. Solutions received from Neuron-GIT algorithm, slicing instance on the left-90 Vofp, non-slicing on the right-76 VoFP.

5. CONCLUSIONS AND TOPIC FOR FURTHER RESEARCH

In this work we develop a novel algorithm based on neural network. We modify widely known GIT algorithm. Neural network as a method of evaluation the best solution in the neighborhood turns out to be very effective. The obtained results can be compared to results of best known heuristic algorithms. The advantage is that computational complexity of presented algorithm is polynomial. Nonetheless this work shows that there are some improvements that can be applied to this algorithm. Especially analyzing inputs of neural network is interesting topic. Finding coefficients which better describe differences between partial solution corresponding to optimal solution and its neighborhood is a core to improve such algorithm.

Another interesting topic is using our method of evaluation the best solution in the neighborhood in some heuristic algorithms such as tabu search.

ACKNOWLEDGMENTS

Andrzej Kozik's work was supported by Polish Ministry of Science and Higher Education under grant No. N519 408337.

REFERENCES

[1] ADYA S.N., MARKOV I.L., *Combinatorial techniques for mixed-size placement*. ACM Transactions on Design Automation of Electronic Systems, 2005, 10(5):58–90.

[2] BAKER B.S., COIFMAN E.G., RIVEST R.L., *Orthogonal packings in two dimensions*. SIAM J. Comput., 1980, 9(4): 846–855.

[3] BLUM C., ROLI A., *Metaheuristics in combinatorial optimization: Overview and conceptual comparison.* ACM Computing Surveys, 2003, 35(3):268–308.

[4] CORMEN T., LEISERSON C., RIVEST R., *Introduction to algorithms.* McGraw-Hill Book Company, 1990.

[5] JANIAK A., KOZIK A., LICHTENSTEIN M., *New perspectives in VLSI design automation: deterministic packing by Sequence Pair.* Annals of Operational Research 179(1), 2010, 35–56.

[6] KALMAN B.L., KWASNY S.C., *Why tanh: choosing a sigmoidal function.* Neural Networks, 1992, IJCNN., International Joint Conference on, Baltimore, 1992, 4:578–581.

[7] KENUE S.K. *Efficient activation functions for the back-propagation neural network.* Neural Networks, 1991, IJCNN., International Joint Conference on, Seattle, 1991.

[8] KUEHLMANN A., *The best of ICCAD – 20 years of excellence in computer aided design.* Kluwer Academic Pub., 2003.

[9] MURATA H., FUHIZOSHI S., NAKATAKE S., KAJITANI Y., *VLSI module placement based on rectangle-packing by the sequence pair*. IEEE Trans. on CAD of ICs., 1996, 15: 1518–1524.

[10] NAWAZ M., ENSCORE E.E. JR., HAM I., *A heuristic algorithm for m-machine, n-job flow-shop sequencing problem.* OMEGA International Journal of Management Science, 1983, 11:91–95.

[11] OTTEN R.H.J.M., *Automatic Floorplan Design*. Proc. ACM/ IEEE Design Automation Conf., 1982, 261–267.

*Block Packing Problem, Sequence-Pair, VLSI,
design automation, algorithm, floorplanning*

Andrzej KOZIK*, Bartosz TOMECZKO*

ON AUTOMATIC SP-DISSECTION IN COMPLEX OUTLINE FLOORPLANNING

Recently, we have presented an algorithmic attempt to perform a frame sp-dissection in complex-outline floorplanning problem: given a rectilinear packing outline, enclose it with minimum bounding rectangle and cut the resulting figure (called frame) into rectangles such that there exists a sequence-pair (defined on set of resulting rectangles) which codes such obtained placement (it decodes into frame).

Our previous algorithm started with an optimal solution of the classical dissection problem, which, however, often cannot be represented with sequence-pair representation. Therefore, in the second stage, we applied an iterative algorithm that, by performing additional cuts of some rectangles, produces a sp-dissection.

In this work, we first give a counterexample, showing that our previous two-phase approach cannot lead to an optimal sp-dissection (sp-dissection of minimum number of rectangles) – a single-phase sp-dissection algorithm is needed. We present several properties of optimal sp-dissection problem solution. Presented properties can lead to an efficient algorithm solving the SP-dissection problem in single-phase.

1. INTRODUCTION

The success of nowadays min-cut floorplanning techniques in VLSI [3, 4, 9, 14] is based on algorithms solving fixed-outline problem [2], in which the integrated circuit elements like transistors, cells or modules have to be placed without overlapping inside given rectangular outline. The further development, however, is frustrated by additional requirements that have appeared with billion-transistor circuits: first and foremost new circuits demand the outlines to be of rectilinear shapes, and second,

* Institute of Computer Engineering, Control and Robotics, Janiszewskiego 11/17, 50-372 Wrocław.

there are rectilinear obstacles in the layout, e.g., modules of predefined positions in the circuit layout [1]. In our previous works [11, 12] we presented a methodology to cope with such *complex-outline* problem, showing, that any set of geometric placement constraints could be represented within Sequence-Pair [6] representation.

Rest of the work is organized as follows. The next section describes the sp-dissection problem. In Section 3 properties of an optimal sp-dissection are presented. Section 4 describes a new concept in sp-dissection – zero-blocks, which introduction allows to achieve optimal sp-dissections. Section 5 concludes the work.

2. FRAME SP-DISSECTION

In our previous paper [12], we have presented a methodology that allows to tackle the complex-outline floorplanning problem: having a set of n rectangular blocks, the goal is to place them inside a rectilinear complex-outline without any overlaps between blocks and rectilinear obstacles in the layout area (fixed positions). We showed that any complex outline and rectilinear obstacles can be represented using Sequence-Pair representation. The Sequence-Pair, introduced by Murata et al. [6], as well as other known representations, is useful for reflecting a packing solution into its appropriate coding and creates a solution space to be traversed for the optimal packing solutions. A sequence pair, say (Γ_+, Γ_-), is a pair of n-element sequences (permutations), each representing some order of elements of the set of blocks having orientations and dimensions fixed. Such a packing representation encodes pairwise relative placements, i.e., the topological relation between each pair of blocks (i, j), $i \neq j$, $i, j \in B$ imposed by their relative order in the (Γ_+, Γ_-), is either horizontal (1) or vertical (2):

$$\left(\langle \ldots, i, \ldots, j, \ldots \rangle, \langle \ldots, i, \ldots, j, \ldots \rangle\right) \Rightarrow i \xrightarrow{h} j \tag{1}$$

$$\left(\langle \ldots, j, \ldots, i, \ldots \rangle, \langle \ldots, i, \ldots, j, \ldots \rangle\right) \Rightarrow i \xrightarrow{v} j \tag{2}$$

Horizontal relation between pair of blocks imposes their left-to-right arrangement in a packing, i.e., $i \xrightarrow{h} j$ forces block j to be placed horizontally on the right of block i. In the case of vertical relations, the arrangement is bottom-up. If a block has no relations oriented to it, it is assumed to be aligned to the reference axis.

The solution space provided by SP consists of $(n!)^2$ sequence-pairs, each of which can be mapped to a packing in polynomial time, and at least one of which corresponds to an optimal solution.

To make this mapping, a geometrical description of placement constraints (outline and obstacles) must be represented by artificial blocks added to the original problem instance. The construction of such a set of artificial blocks is the sp-dissection: given a rectilinear packing outline, enclose it with minimum bounding rectangle and cut the resulting figure (called frame) and rectilinear obstacles (if any) inside the packing area into rectangles such that there exists a sequence-pair (defined on the set of resulting rectangles) which codes such obtained placement.

Recently, we have presented an algorithmic attempt to perform a frame sp-dissection in complex-outline floorplanning problem [12]. Our algorithm started with an optimal solution of the classical dissection problem [8, 15], which, however, often cannot be represented with Sequence-Pair representation. Therefore, in the second stage, we applied an iterative algorithm that, by performing additional cuts of some rectangles, produces a sp-dissection. An example of frame sp-dissection is presented in Fig. 1a. Observe a frame composed of artificial blocks. In Fig. 1b an obstacle in the layout is added. Note, that in order to fix the position of the obstacle, some of the frame blocks have to be divided (in the second stage of our algorithm).

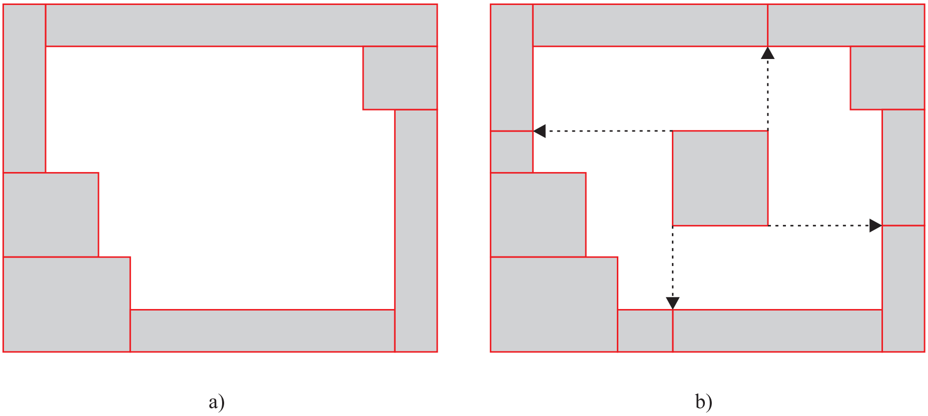


Fig. 1. Examples of sp-dissection of complex-outline (a) and complex outline with obstacle in the layout

Our previous two-phase approach, unfortunately, cannot lead to an optimal sp-dissection (sp-dissection of minimum number of rectangles). The counter-example is presented in Fig. 2, where classical dissection of rectilinear obstacle (a) is composed of 11 rectangles (b), its sp-dissection produced by our two-phase algorithm is composed of 19 rectangles (c), while optimal sp-dissection is of only 13 rectangles (d).

Property 1. The optimal sp-dissection must be produced by a single-phase algorithm.

3. PROPERTIES OF OPTIMAL SP-DISSECTION

3.1. DISSECTION OF THE FRAME BOUNDARY

Consider a fragment of frame left boundary given in Fig. 4. It can be divided as in Fig. 4a into 6 *frame* blocks (blocks composing the frame, for details see [12]) or into 11 blocks as in Fig. 4b (2 frame blocks, 4 *inside*-blocks, and 5 cut-lines producing 5 additional blocks). This shows, that the frame should be divided horizontally. Now, consider a simple rectangular obstacle in the layout (inside-block). If the frame was divides as in Fig. 4a, then fixing an obstacle results in 19 blocks in total (Fig. 4c); if the frame was divides as in Fig. 4b, then fixing an obstacle results in 16 blocks in total, what contradicts the earlier statement.

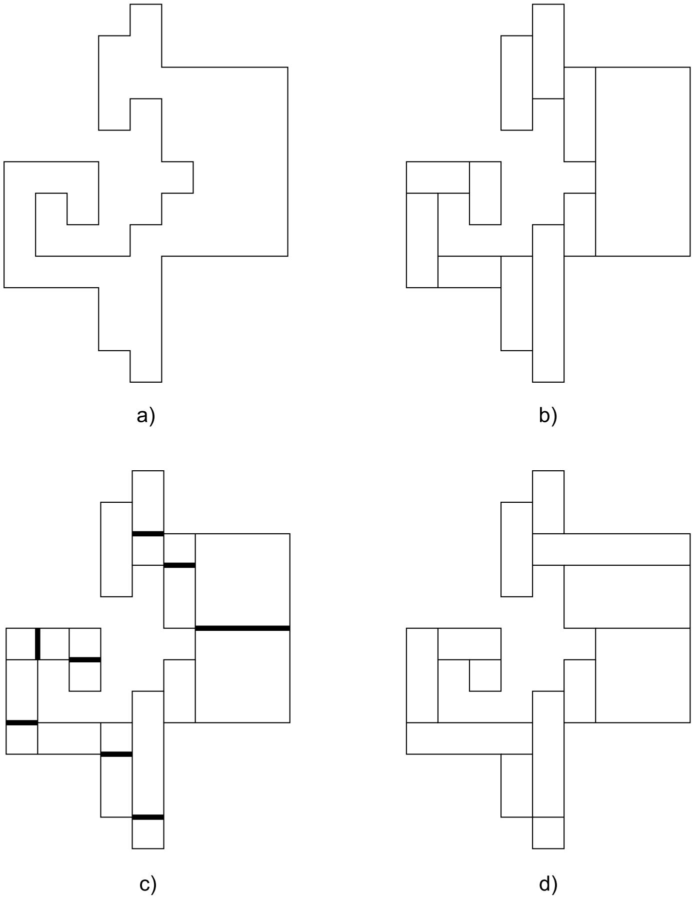


Fig. 2. Example of obstacle dissection – shape of rectilinear obstacle (a), optimal classical dissection (b), sp-dissection result of two-phase algorithm (c), optimal sp-dissection (d)

Property 2. The sp-dissection of the frame to be optimal have to consider the obstacles in the layout.

3.2. DISSECTION OF THE FRAME BOUNDARY

Consider the obstacle in the layout presented in Fig. 3. It can be internally divided in many ways, e.g., into 5 blocks in Fig. 3a or 7 blocks in Fig. 3b. In order to fix its position in the layout, it generates cut-lines, that divide the frame. Note, in both cases the set of cut-lines (denoted as arrows in Fig. 3) is the same.

Property 3. The internal dissection of obstacle has no effect on dissection of the frame. The frame dissection depends only on the shape of the obstacle. Obstacle composed of *k* corners can produce at most *k*/2 + 2 cut-lines.

3.3. CUT-LINES ABSORPTION

Consider a frame with two obstacles given in Fig. 5. Observe that some cut-lines (denoted as dotted arrows) result in division of other obstacles (they not divide frame), i.e., some cut-lines of Property 3 are absorbed by other obstacles.

Property 4. The set of obstacles can be preprocessed in such a way, that some cut-lines can be absorbed. Resulting reduced set of cut-lines, according to Property 2, should be considered in the division of frame.

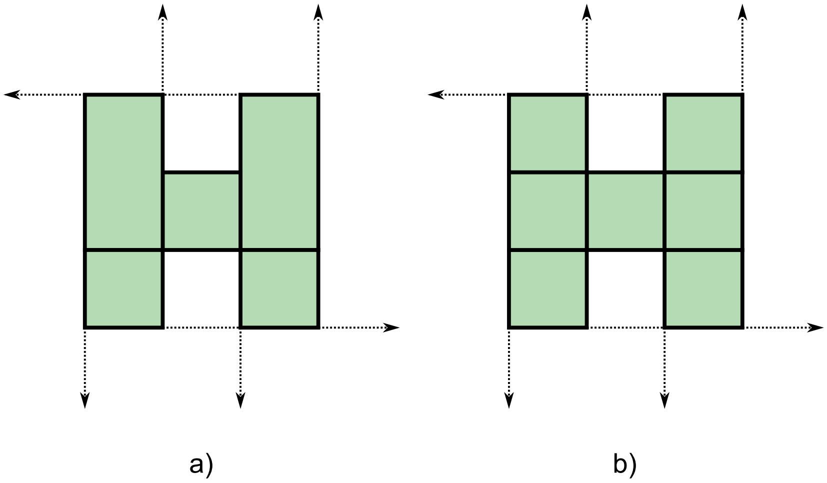
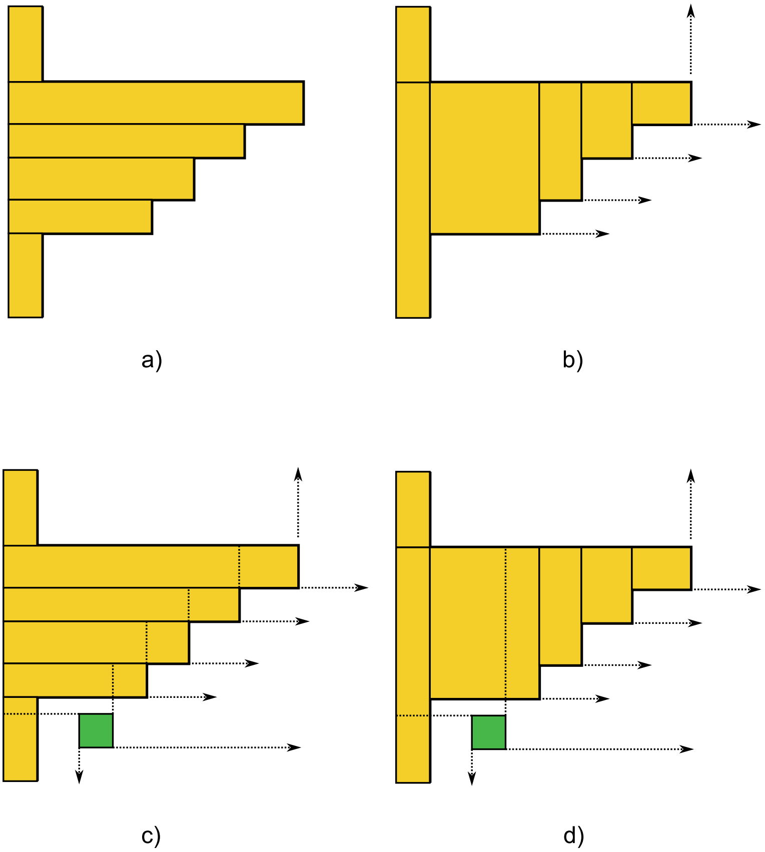


Fig. 3. Example of internal division of obstacle in the layout



a)

b)

c)

d)

Fig. 4. Example of frame division

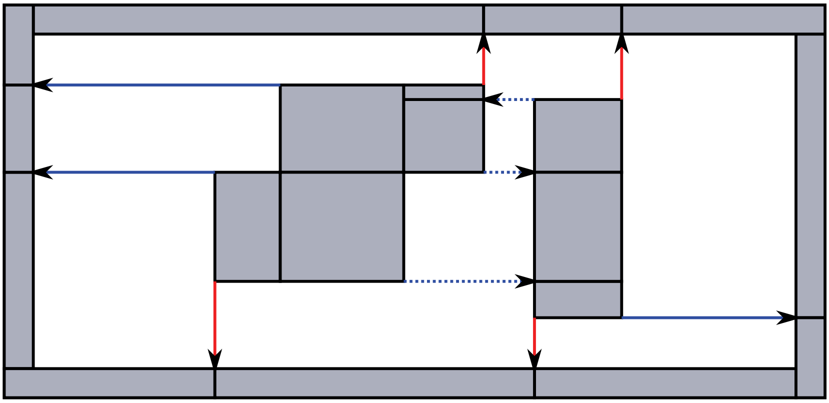


Fig. 5. Example dissection of frame with two obstacles

3.3. RECURSIVE CUT-LINES

Consider the frame and obstacle presented in Fig. 6. Fixing the obstacle I1 produces the cut-line C1 which, in turn, cuts off inside-block I2 from the frame-block. The new inside-block I2 generates two additional cut-lines in order to fix its position in the layout. This process can be recursively continued – an extreme example is given in Fig. 4c.

Similar situation to that of Fig. 6 is depicted in Fig. 7. In Fig. 7a an inside block divides a frame-block what results in 10 blocks in total (6 blocks and 4 cut-lines). Note, that the frame-block that remain after the division by cut-line forms a *left base* for inside-block, i.e., there is a horizontal relation in sequence pair between those two blocks.

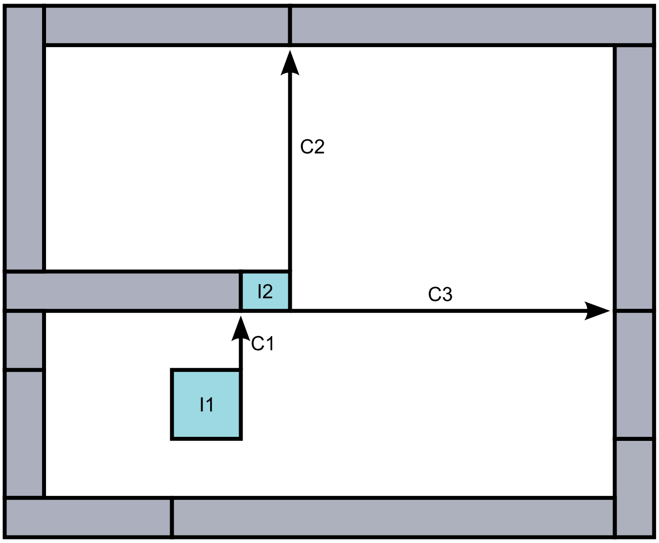


Fig. 6. Division of frame-block generates a new inside-block

4. PROPERTIES OF ZERO-BLOCKS

In Fig. 7b, an artificial *zero-block*, of size zero in one dimension, is introduced to form such a left-base. Observe, original frame-block does not to be divided anymore (forming a left-base does not introduce new inside-blocks with new cut-lines).

Property 5. The zero-blocks can stop the recurrent division of blocks caused by formation of left-base for inside-block.

There are situation, however, that introduction of zero-block does not result in less block in total. Consider a frame presented in Fig. 8, where two example dissections are depicted. Observe, layout of Fig. 8a needs 13 blocks (with 2 zero-blocks), while layout of Fig. 8b needs 12 blocks. This suggests that introducing a zero-block does not always brings desired results.

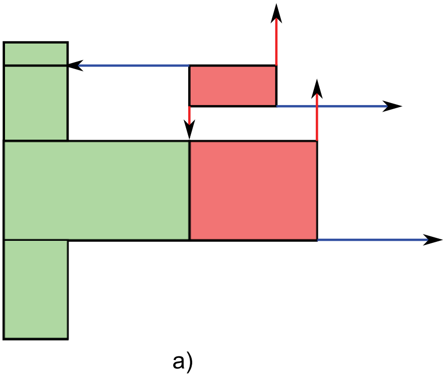


Fig. 7. Examples of forming a left-base for inside-block: by block division (a) and by zero-block (b)

There are, however, situations, where introducing more than one zero-block can help in obtaining better sp-dissection. In Fig. 9 a situation similar to that of Fig. 8 is presented. Note, that in this case, layout of Fig. 9a needs 13 blocks, while layout of Fig. 9b needs 14 blocks.

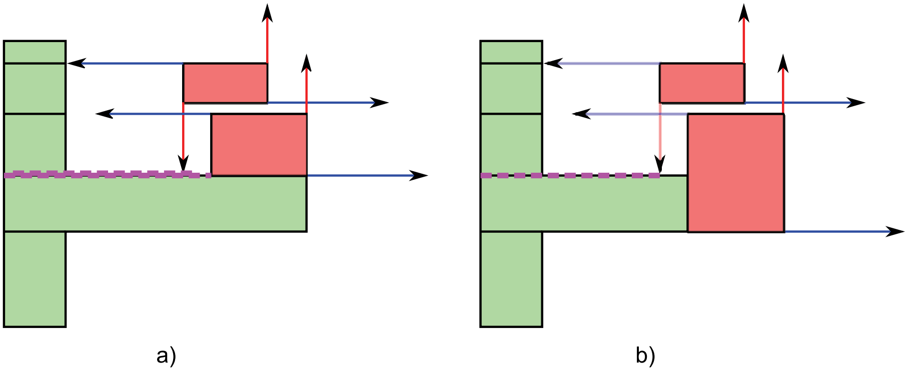


Fig. 8. Examples of forming a left-base for inside-block: by two zero-blocks (a) and by single zero-block and division (b)

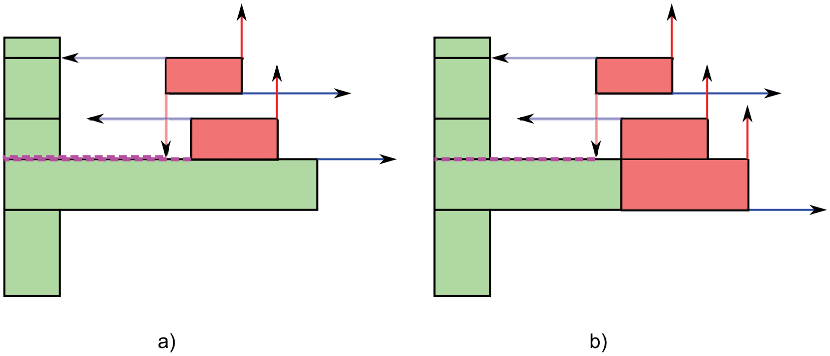


Fig. 9. Examples of forming a left-base for inside-block: by two zero-blocks (a) and by single zero-block and division (b)

5. CONCLUSIONS

In this work we considered a problem of automatic translation of complex geometric layout constraints into Sequence-Pair representation. We showed, that our previous (two-phase) algorithmic approach to sp-dissection problem cannot lead to dissections of minimum sizes, and showed several properties of optimal sp-dissections. We introduced a concept of zero-blocks, that artificially added to the problem instance, lead to solutions with fewer blocks. Presented properties can form a basis for a single-phase algorithm solving sp-dissection problem optimally.

Further research, apart from development of such algorithm, will be concentrated on application of proposed methodology to algorithms solving complex-outline floorplanning problem. Ongoing research includes enhancing simulated annealing approach [7] and building constructive heuristic based on greedy insertion technique [5, 10] and fast Sequence-Pair neighborhood evaluation [13]. Already obtained results are promising.

ACKNOWLEDGMENTS

Andrzej Kozik's work was supported by Polish Ministry of Science and Higher Education under grant No. N519 408337. Bartosz Tomeczko's work was supported by Polish National Science Centre under grant No. N516 494140.

REFERENCES

[1] ADYA S.N., CHAN H.H., LU J.F., MARKOV I.L., NG A.N., PAPA D.A., ROY J.A. *CAPO: Robust and Scalable Open-Source Min-Cut Floorplacer*, ISPD ’05, 2005.

[2] ADYA S.N., MARKOV I.L. *Fixed-outline floorplanning: enabling hierarchical design*. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2003, 11(6): 1120–1135.

[3] ADYA S.N., MARKOV I.L. *Combinatorial techniques for mixed-size placement*. ACM Transactions on Design Automation of Electronic Systems, 2005, 10(5):58–90.

[4] BASARAN B., GANESH K., LEVIN A., MCCOO M., RANGARAJAN S., RAU R., SEHGAL N. *GeneSys – a layout synthesis system for GHz VLSI designs*. Proc. Int. Conf. in VLSI Design, 1999, 458–472.

[5] ENSCORE E.E. JR, HAM I., NAWAZ M. *A heuristic algorithm for m-machine, n-job flow-shop sequencing problem*. OMEGA International Journal of Management Science (11), 1983, 91–95.

[6] FUJIYOSHI K., MURATA H., KAJITANI Y. AND NAKATAKE S. *VLSI module placement based on rectangle-packing by the sequence pair*. IEEE Trans. on CAD of ICs., 1996, 15:1518–1524.

[7] GELATT C.D., KIRKPATRICK S., VECCHI M. P. *Optimization by Simulated Annealing*. Science 220 (4598), 1983, 671–680.

[8] GORPINEVICH A., SOLTAN V. *Minimum dissection of a rectilinear polygon with arbitrary holes into rectangles*. Discrete & Computational Geometry, Vol. 9, No. 1, 1992, 57–79.

[9] HAYES J.P. AND MURRAY B.T. *Testing ICs: getting to the core of the problem*. IEEE Computer Magazine, 1996, 11:32–38.

[10] JANIAK A., KOZIK A., LICHTENSTEIN M. *New perspectives in VLSI design automation: deterministic packing by Sequence Pair*. Annals of Operations Research. vol. 179, 2010, nr 1: 35–56.

[11] JANIAK A., KOZIK A., TOMECZKO B. *Complex outline packing problem in VLSI physical design*. Information systems architecture and technology: system analysis in decision aided problems, Oficyna Wydawnicza PWr., Wrocław 2009, pp. 343–354.

[12] JANIAK A., KOZIK A., TOMECZKO B. *Algorithms for encoding rectilinear outline constraint in Sequence-Pair representation for VLSI floorplan design.* Information systems architecture and technology: system analysis in decision aided problems Wrocław: Oficyna Wydawnicza Politechniki Wrocławskiej, 2010, pp. 223–233.

[13] JANIAK A., KOZIK A., TOMECZKO B. *On the Sequence-Pair neighborhood evaluation.* Information systems architecture and technology: system analysis in decision aided problems Wrocław: Oficyna Wydawnicza Politechniki Wrocławskiej, 2009. pp. 355–368.

[14] KUEHLMANN A. *The best of ICCAD – 20 years of excellence in computeraided design*. Kluwer Academic Pub., 2003.

[15] LEE R.C.T., LIOU W.T., TAN J.J.-M. *Minimum rectangular partition problem for simple rectilinear polygons*. Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, 1990, 720–733.

BIBLIOTEKA INFORMATYKI SZKÓŁ WYŻSZYCH

Information Systems Architecture and Technology, ISAT 2005, pod redakcją Adama GRZECHA, Zofii WILIMOWSKIEJ, Wrocław 2005

Information Systems Architecture and Technology. Information Models, Concepts, Tools and Applications, pod redakcją Leszka BORZEMSKIEGO, Adama GRZECHA, Jerzego ŚWIĄTKA, Zofii WILIMOWSKIEJ, Wrocław 2006

Information Systems Architecture and Technology. Information Technology and Web Engineering: Models, Concepts & Challenges, pod redakcją Leszka BORZEMSKIEGO, Adama GRZECHA, Jerzego ŚWIĄTKA, Zofii WILIMOWSKIEJ, Wrocław 2007

Information Systems Architecture and Technology. Application of Information Technologies in Management Systems, pod redakcją Zofii WILIMOWSKIEJ, Leszka BORZEMSKIEGO, Adama GRZECHA, Jerzego ŚWIĄTKA, Wrocław 2007

Information Systems Architecture and Technology. Decision Making Models, pod redakcją Jerzego ŚWIĄTKA, Leszka BORZEMSKIEGO, Adama GRZECHA, Zofii WILIMOWSKIEJ, Wrocław 2007

Information Systems Architecture and Technology. Information Systems and Computer Communication Networks, pod redakcją Adama GRZECHA, Leszka BORZEMSKIEGO, Jerzego ŚWIĄTKA, Zofii WILIMOWSKIEJ, Wrocław 2007

Information Systems Architecture and Technology. Web Information Systems: Models, Concepts & Challenges, pod redakcją Leszka BORZEMSKIEGO, Adama GRZECHA, Jerzego ŚWIĄTKA, Zofii WILIMOWSKIEJ, Wrocław 2008

Information Systems Architecture and Technology. Information Systems and Computer Communication Networks, pod redakcją Adama GRZECHA, Leszka BORZEMSKIEGO, Jerzego ŚWIĄTKA, Zofii WILIMOWSKIEJ, Wrocław 2008

Information Systems Architecture and Technology. Models of the Organisations Risk Management, pod redakcją Zofii WILIMOWSKIEJ, Leszka BORZEMSKIEGO, Adama GRZECHA, Jerzego ŚWIĄTKA, Wrocław 2008

Information Systems Architecture and Technology. Designing, Development and Implementation of Information Systems, pod redakcją Leszka BORZEMSKIEGO, Adama GRZECHA, Jerzego ŚWIĄTKA, Zofii WILIMOWSKIEJ, Wrocław 2008

Information Systems Architecture and Technology. Model Based Decisions, pod redakcją Jerzego ŚWIĄTKA, Leszka BORZEMSKIEGO, Adama GRZECHA, Zofii WILIMOWSKIEJ, Wrocław 2008

Information Systems Architecture and Technology. Advances in Web-Age Information Systems, pod redakcją Leszka BORZEMSKIEGO, Adama GRZECHA, Jerzego ŚWIĄTKA, Zofii WILIMOWSKIEJ, Wrocław 2009

Information Systems Architecture and Technology. Service Oriented Distributed Systems: Concepts and Infrastructure, pod redakcją Adama GRZECHA, Leszka BORZEMSKIEGO, Jerzego ŚWIĄTKA, Zofii WILIMOWSKIEJ, Wrocław 2009

Information Systems Architecture and Technology. Systems Analysis in Decision Aided Problems, pod redakcją Jerzego ŚWIĄTKA, Leszka BORZEMSKIEGO, Adama GRZECHA, Zofii WILIMOWSKIEJ, Wrocław 2009

Information Systems Architecture and Technology. IT Technologies in Knowledge Oriented Management Process, pod redakcją Zofii WILIMOWSKIEJ, Leszka BORZEMSKIEGO, Adama GRZECHA, Jerzego ŚWIĄTKA, Wrocław 2009

Information Systems Architecture and Technology. New Developments in Web-Age Information Systems, pod redakcją Leszka BORZEMSKIEGO, Adama GRZECHA, Jerzego ŚWIĄTKA, Zofii WILIMOWSKIEJ, Wrocław 2010

Information Systems Architecture and Technology. Networks and Networks Services', pod redakcją Adama GRZECHA, Leszka BORZEMSKIEGO, Jerzego ŚWIĄTKA, Zofii WILIMOWSKIEJ, Wrocław 2010

Information Systems Architecture and Technology. System Analysis Approach to the Design, Control and Decision Support, pod redakcją Jerzego ŚWIĄTKA, Leszka BORZEMSKIEGO, Adama GRZECHA, Zofii WILIMOWSKIEJ, Wrocław 2010

Information Systems Architecture and Technology. IT TModels in Management Process, pod redakcją Zofii WILIMOWSKIEJ, Leszka BORZEMSKIEGO, Adama GRZECHA, Jerzego ŚWIĄTKA, Wrocław 2010

Wydawnictwa Politechniki Wrocławskiej
są do nabycia w księgarni „Tech”
plac Grunwaldzki 13, 50-377 Wrocław
budynek D-1 PWr., tel. 71 320 29 35
Prowadzimy sprzedaż wysyłkową
zamawianie.ksiazek@pwr.wroc.pl

ISBN 978-83-7493-632-3