

Krzysztof Targiel

Akademia Ekonomiczna w Katowicach

ZASTOSOWANIE JĘZYKÓW OPISU PROBLEMÓW OPTIMALIZACYJNYCH W ARKUSZACH KALKULACYJNYCH

Streszczenie: Obserwowane zainteresowania poszukiwaniem najlepszych rozwiązań biznesowych z wykorzystaniem metod matematycznych owocuje powstaniem bogatego oprogramowania implementującego metody badań operacyjnych. Powstał problem zapisu modelu w formie akceptowalnej przez oprogramowanie oraz sposobu wprowadzania parametrów. Pojawiły się dwie odmienne koncepcje rozwiązania tego problemu. Zrodziło to z jednej strony pomysły wbudowania solvera w arkusz kalkulacyjny, z drugiej strony powstanie zintegrowanych środowisk do rozwiązywania takich problemów. Zauważalna jest także powszechność stosowania arkusza kalkulacyjnego jako interfejsu we wszelkich zastosowaniach biznesowych. W pracy przedstawiono możliwość wykorzystania języków opisu problemów optymalizacyjnych w arkuszu kalkulacyjnym oraz zwrócono uwagę na zalety takiego podejścia.

Słowa kluczowe: język opisu problemu optymalizacyjnego, język LINGO, optymalizacja z MS Excel

1. Wstęp

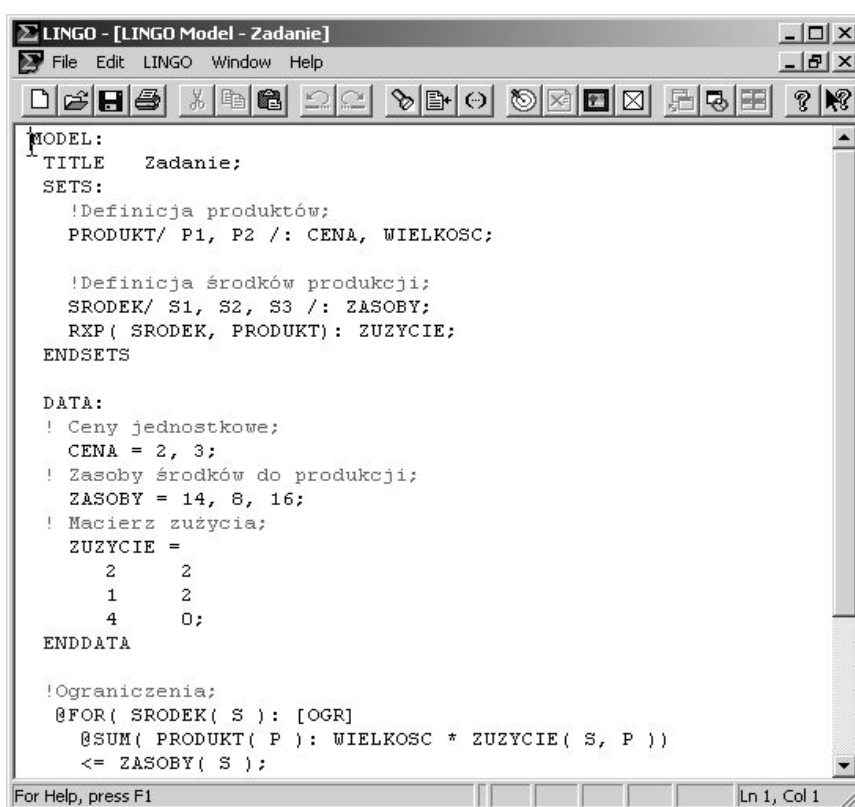
Wzrost konkurencji wiąże się z koniecznością podejmowania trafniejszych decyzji. W ich wyborze mogą pomóc metody matematyczne. Wykorzystanie metod matematycznych w procesie podejmowania decyzji jest przedmiotem badań operacyjnych. Złożoność współczesnego środowiska gospodarczego powoduje złożoność modeli rozpatrywanych w ramach badań operacyjnych. Do rozwiązywania zadań optymalizacyjnych konieczne są wyspecjalizowane pakiety oprogramowania.

Współczesne oprogramowanie do rozwiązywania problemów optymalizacyjnych to najczęściej zintegrowane środowiska. Składają się na nie:

- środowisko graficzne integrujące wszystkie funkcje systemu,
- język opisu problemów optymalizacyjnych
- program do rozwiązywania konkretnego zadania optymalizacyjnego (solver)

Środowisko graficzne integrujące całość posiada mechanizmy wprowadzania i edycji zadania optymalizacyjnego oraz narzędzia do obsługi zbiorów dyskowych.

Poprzez to środowisko można uruchamiać solvery oraz prezentować otrzymane rozwiązania. Drugim elementem zintegrowanego środowiska jest język opisu problemu optymalizacyjnego. Najczęściej jest to specyficzny język formalny, dzięki któremu można opisać problem. Problem optymalizacyjny zapisany w tym języku jest następnie tłumaczony na format zrozumiały dla ostatniego z elementów zintegrowanych środowisk, czyli dla solverów. Są to programy, w których zaimplementowano algorytmy rozwiązywania zadań optymalizacyjnych. Najczęściej są to programy zewnętrzne dla producenta środowiska, choć nie jest to regułą. Programy te rozwiązują zadanie, a następnie przekazują wyniki do zintegrowanego środowiska.

The image shows a screenshot of the LINGO software interface. The window title is "LINGO - [LINGO Model - Zadanie]". The menu bar includes "File", "Edit", "LINGO", "Window", and "Help". The toolbar contains various icons for file operations and editing. The main text area contains the following LINGO model code:

```
MODEL:
TITLE    Zadanie;
SETS:
    !Definicja produktów;
    PRODUKT/ P1, P2 /: CENA, WIELKOSC;

    !Definicja środków produkcji;
    SRODEK/ S1, S2, S3 /: ZASOBY;
    RXP( SRODEK, PRODUKT): ZUZYCIE;
ENDSETS

DATA:
    ! Ceny jednostkowe;
    CENA = 2, 3;
    ! Zasoby środków do produkcji;
    ZASOBY = 14, 8, 16;
    ! Macierz zużycia;
    ZUZYCIE =
        2   2
        1   2
        4   0;
ENDDATA

!Ograniczenia;
@FOR( SRODEK( S ): [OGR]
    @SUM( PRODUKT( P ): WIELKOSC * ZUZYCIE( S, P ))
    <= ZASOBY( S );
```

The status bar at the bottom indicates "For Help, press F1" and "Ln 1, Col 1".

Rys. 1. System LINGO

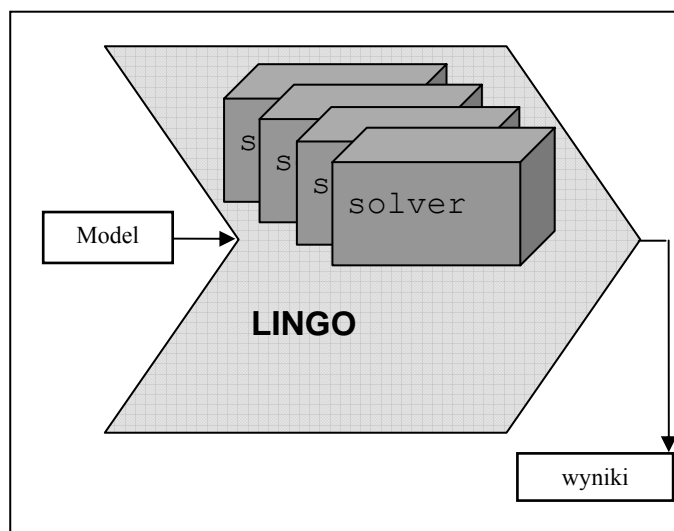
Początkowo zintegrowane środowiska powstawały w celu ułatwienia dostępu do istniejących solverów, zatem nacisk kładziono przede wszystkim na sposób wprowadzania danych wejściowych – modelu i jego parametrów. Obecne tendencje w rozwoju takiego oprogramowania zmierzają ku rozwojowi sposobów prezentacji wyników. Tak jest w komercyjnym środowisku AIMMS oferowanym przez

holenderską firmę Paragon Decision Technology B.V. Celem niniejszej pracy jest pokazanie, jak można te współczesne tendencje zrealizować w środowisku LINGO (rys. 1), którego zaletą jest to, iż posiada wersję darmową, choć o ograniczonych możliwościach.

2. Środowisko LINGO

Przykładem zintegrowanego środowiska jest System LINGO, oferowany przez firmę LINDO Systems Inc. [Schrage, Cunningham 1988]. Jest to ulepszona wersja poprzednika – systemu LINDO, który nie oferował jeszcze możliwości indeksowania [Fourer, Gay, Kernighan 1990]. Na rysunku 1 pokazano przykład ekranu środowiska. Środowisko ma postać prostego edytora, w którym zapisywany jest model. Dostępne są opcje w menu oraz ikonki, poprzez które można wywołać solver.

Współdziałanie elementów systemu oraz jego strukturę przedstawiono na rys. 2.



Rys. 2. Struktura systemu LINGO

Model zapisywany jest w edytorze dostępnym w systemie w postaci pliku tekstowego. Poprzez funkcje systemu jest on przekazywany solverom do rozwiązania. System LINGO korzysta z własnych wbudowanych implementacji algorytmów optymalizacyjnych. Dzięki takiemu rozwiązaniu system sam potrafi rozpoznać typ zadania i dobrać odpowiedni solver. Otrzymywane wyniki są prezentowane w środowisku w postaci raportów pojawiających się w oknie wyników. Jak zostanie to pokazane w dalszej części artykułu, możliwe jest przekazywanie wyników do arkusza kalkulacyjnego, a w nim prezentacja graficzna.

Na stronie <http://www.lindo.com/> jest dostępna darmowa testowa wersja oprogramowania, która posiada pewne ograniczenia. Zadanie może mieć: 150 zmiennych i 300 ograniczeń, 30 zmiennych dyskretnych, 30 ograniczeń nieliniowych i 5 zmiennych globalnych. Za pomocą tego systemu można rozwiązywać zadania liniowe, całkowitoliczbowe, kwadratowe oraz nieliniowe. Dla systemu stworzono język LINGO.

3. Język LINGO

Język LINGO daje szerokie możliwości modelowania problemów decyzyjnych. System umożliwia definiowanie zbiorów obiektów, na których podstawie tworzone są indeksy – symbole obiekty reprezentujące. Zbiory mogą być proste oraz złożone. Możliwe jest opisywanie relacji występujących w modelowanym systemie. Zdefiniowano operatory iteracyjne, które bazują na indeksacji. Język opiera się na licznych zdefiniowanych funkcjach. Jako przykład modelu zapisanego w języku LINGO prześledzimy zadanie z podręcznika [Trzaskalik 2008], które prowadzi do następującego zadania programowania liniowego:

$$\begin{aligned} 2x_1 + 2x_2 &\rightarrow \max, \\ 2x_1 + 2x_2 &\leq 14, \\ x_1 + 2x_2 &\leq 8, \\ 4x_1 &\leq 16, \\ x_1 \geq 0, x_2 &\geq 0. \end{aligned}$$

Jest to zadanie, w którym chcemy dobrać optymalną wielkość produkcji dwu wyrobów (produktów), które są wytwarzane z trzech surowców. Odpowiadają im kolejne ograniczenia. Kryterium oceny planów produkcji jest maksymalizacja zysku z produkcji. W języku LINGO model ten został zakodowany tak jak pokazano to w przykładzie 1.

Przykład 1. Zadanie w składni języka LINGO

```
MODEL:
  TITLE Zadanie;
  SETS:
    !Definicja produktów;
    PRODUKT/ P1, P2 /: CENA, WIELKOSC;
    !Definicja środków produkcji;
    SRODEK/ S1, S2, S3 /: ZASOBY;
    RXP( SRODEK, PRODUKT): ZUZYCIE;
  ENDSETS
  DATA:
    ! Ceny jednostkowe;
    CENA = 2, 3;
    ! Zasoby środków do produkcji;
    ZASOBY = 14, 8, 16;
```

```
! Macierz zużycia;
ZUZYCIE =
      2      2
      1      2
      4      0;
ENDDATA

!Ograniczenia;
@FOR( SRODEK( S ): [OGR]
@SUM( PRODUKT( P ): WIELKOSC * ZUZYCIE( S, P ))
<= ZASOBY( S );
);

! Funkcja celu;
[CEL] MAX =
@SUM( PRODUKT : WIELKOSC * CENA * CENA );
END
```

Problem zapisano pomiędzy słowem kluczowym MODEL a słowem kluczowym END. Po słowie kluczowym TITLE zapisano nazwę zadania, które stworzono dla rozpatrywanego problemu. W zapisie problemu, który możemy traktować jako program do zrealizowania przez maszynę cyfrową, występują sekcje definicji zbiorów (SETS: ... ENDSETS) oraz sekcja danych (DATA: ... ENDDATA). W sekcji definicji zbiorów zadeklarowano i zdefiniowano zbiory tworzące indeksy wykorzystywane w dalszej części. Jest to zbiór produktów (PRODUKTY), zbiór surowców (SRODEK) oraz zbiór będący iloczynem kartezjańskim tych zbiorów (RXP). Każdy z dwu elementów zbioru PRODUKTY wymienionych po ukośniku (są to elementy P1 i P2) ma dwa atrybuty, są to: cena tego produktu (pole CENA) oraz wielkość produkcji (pole WIELKOSC). Elementy zbioru SRODEK (jest ich trzy: S1, S2, S3) posiadają tylko jeden atrybut ZASOBY, opisujący wielkość zasobu tego środka do produkcji. Podobnie elementy zbioru RXP mają jeden atrybut ZUZYCIE, opisujący ilość surowca na produkcję jednostkową. Elementami tego zbioru są pary stworzone z elementów zbiorów produktów i surowców. Wartości atrybutom elementów nadano w sekcji danych. Następnie zdefiniowano ograniczenia, które muszą być spełnione.

```
@FOR( SRODEK( S ): [OGR],
@SUM( PRODUKT( P ): WIELKOSC * ZUZYCIE( S, P ))
<= ZASOBY( S );
```

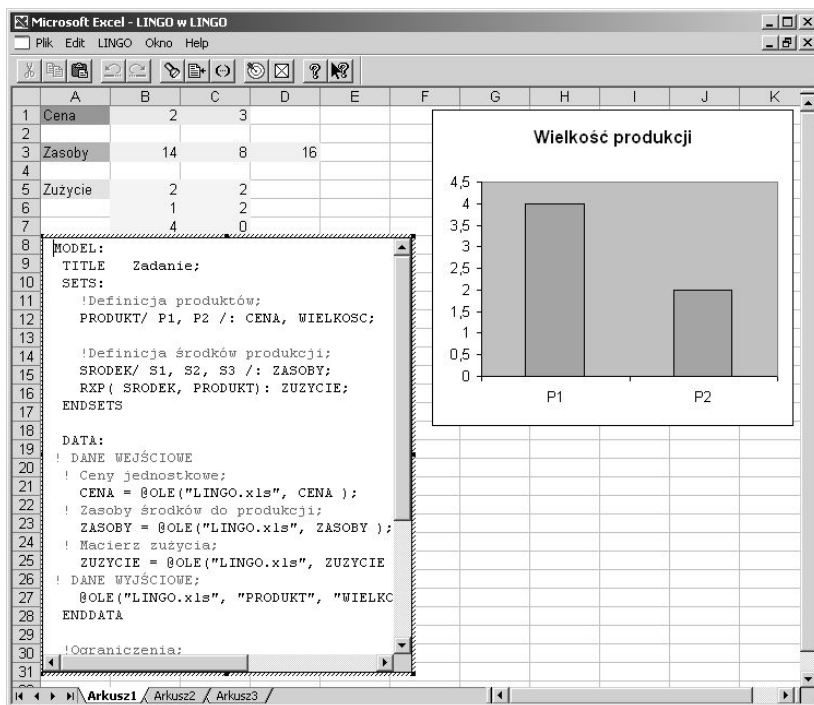
Wykorzystano instrukcje iteracyjne: przebiegającą elementy zbioru (@FOR) oraz sumującą po elementach zbioru (@SUM). W nawiasach kwadratowych nadano ograniczeniom nazwę, która później pojawia się w raportach generowanych przez solver. Wreszcie zdefiniowano funkcję celu w następujących liniach:

```
[CEL] MAX =
    @SUM( PRODUKT : WIELKOSC * CENA * CENA );
```

Słowo kluczowe MAX, które poprzedza nazwa funkcji celu umieszczona w nawiasach kwadratowych, informuje system, iż następująca po znaku równości formuła ma osiągnąć wartość maksymalną, dopuszczalną uprzednio zdefiniowanymi ograniczeniami. System LINGO sam dobierze odpowiedni program do rozwiązania tego zadania.

4. Osadzanie w arkuszu

W przedstawionym przykładzie dane zostały zapisane w sekcji DATA. Możliwe jest jednak zintegrowanie modelu z arkuszem kalkulacyjnym, który będzie stanowił interfejs, poprzez który dane są wprowadzane i poprzez który prezentowane są wyniki. W wersji 10.0 systemu LINGO możliwe było zintegrowanie jedynie z arkuszem MS Excel. Model jest osadzany jako obiekt w arkuszu kalkulacyjnym przy wykorzystaniu techniki OLE (*object linking and embedding*). Wygląd arkusza pokazano na rys. 3. W arkuszu tym zostały zdefiniowane następujące zakresy:



Rys. 3. Model osadzony w arkuszu Excel

- o nazwie CENY: obszar komórek B1:C1,
- o nazwie ZASOBY: obszar komórek B3:D3,
- o nazwie ZUZYCIE: obszar komórek B5:C7.

Nazwy zakresu zostały następnie wykorzystane w modelu. Do przekazywania danych wykorzystana została funkcja @OLE o następującej składni:

```
OBIEKTY = @OLE( ['ZBIOR.XLS'] [, ZAKRESY]);
```

Poprzez OBIEKTY oznaczono listę symboli, którym zostaną przekazane dane z arkusza do modelu, ZBIOR.XLS oznacza nazwę skoroszytu Excela, natomiast ZAKRESY to lista uprzednio zdefiniowanych w arkuszu zakresów.

W podobny sposób można przekazywać wyniki z modelu do arkusza, także wykorzystując funkcję @OLE, lecz w następującej składni.

```
@OLE( ['ZBIOR.XLS'] [, ZAKRESY]) = OBIEKTY;
```

Oznaczenia są identyczne jak przy imporcie danych. W arkuszu należało zdefiniować odpowiednie zakresy, w które zostaną wpisane eksportowane dane. W rozważanym przykładzie zdefiniowano zakres WIELKOSC. Zakres ten jest wyświetlany w postaci wykresu widocznego na rys 3.

Zapis modelu wykorzystanego w arkuszu kalkulacyjnym pokazano w przykładzie 2.

Przykład 2. Zadanie osadzone w arkuszu

```
TITLE Zadanie;
SETS:
!Definicja produktów;
PRODUKT/ P1, P2 /: CENA, WIELKOSC;

!Definicja środków produkcji;
SRODEK/ S1, S2, S3 /: ZASOBY;
RXP( SRODEK, PRODUKT): ZUZYCIE;
ENDSETS

DATA:
! DANE WEJŚCIOWE
! Ceny jednostkowe;
CENA = @OLE("LINGO.xls", CENA );
! Zasoby środków do produkcji;
ZASOBY = @OLE("LINGO.xls", ZASOBY );
! Macierz zużycia;
ZUZYCIE = @OLE("LINGO.xls", ZUZYCIE );
! DANE WYJŚCIOWE;
@OLE("LINGO.xls", "PRODUKT", "WIELKOSC" ) = PRODUKT, WIELKOSC;
ENDDATA

!Ograniczenia;
@FOR( SRODEK( S ): [OGR]
@SUM( PRODUKT( P ): WIELKOSC * ZUZYCIE( S, P ))
```

```

<= ZASOBY( S );
);

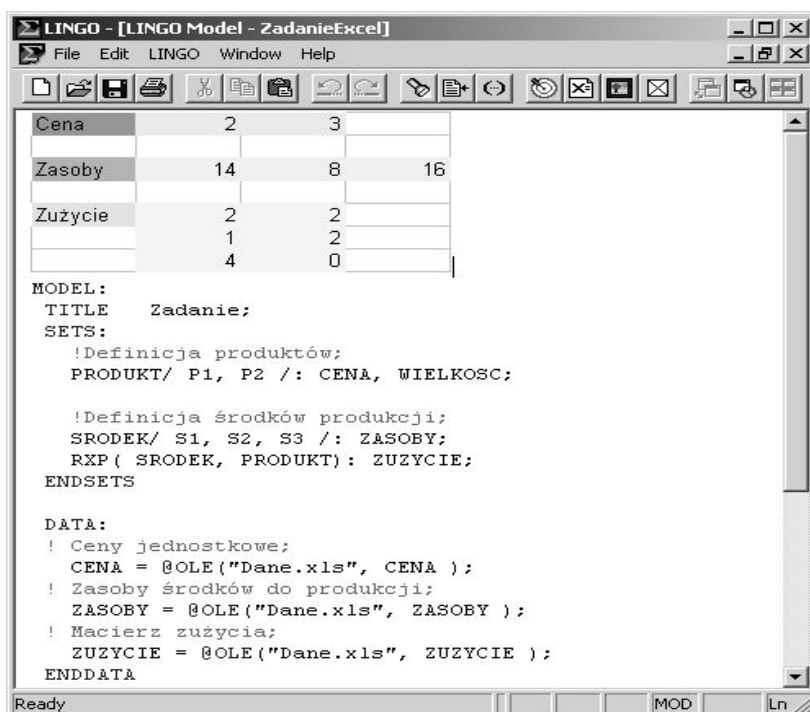
! Funkcja celu;
[CEL] MAX =
@SUM( PRODUKT : WIELKOSC * CENA );
END

```

Różnice widoczne są w sekcji DATA, w której wykorzystano opisane wcześniej funkcje. Dane wczytywane są z zakresów CENA, ZASOBY, ZUZYCIE. Rezultaty są przekazywane do arkusza do zakresu PRODUKT, w którym wpisywany jest kolejny element zbioru produktów, oraz do zakresu WIELKOŚĆ, do którego jest wpisywana obliczona optymalna wielkość produkcji.

5. Osadzanie arkusza

Możliwa jest także sytuacja odwrotna – osadzanie arkusza w środowisku LINGO. Dzięki takiemu rozwiązaniu można usprawnić wprowadzanie danych wejściowych, na przykład podczas testowania modelu. Wygląd systemu LINGO po umieszczeniu w nim arkusza Excela przedstawiono na rys. 4, natomiast wykorzystany model jako przykład 3.



Rys. 4. Arkusz Excel osadzony w LINGO

Przykład 3. Model z danymi czerpanymi z arkusza

```
MODEL:
  TITLE Zadanie;
  SETS:
    !Definicja produktów;
    PRODUKT/ P1, P2 /: CENA, WIELKOSC;

    !Definicja środków produkcji;
    SRODEK/ S1, S2, S3 /: ZASOBY;
    RXP( SRODEK, PRODUKT): ZUZYCIE;
  ENDSETS

  DATA:
    ! Ceny jednostkowe;
    CENA = @OLE("Dane.xls", CENA );
    ! Zasoby środków do produkcji;
    ZASOBY = @OLE("Dane.xls", ZASOBY );
    ! Macierz zużycia;
    ZUZYCIE = @OLE("Dane.xls", ZUZYCIE );
  ENDDATA

  !Ograniczenia;
  @FOR( SRODEK( S ): [OGR]
    @SUM( PRODUKT( P ): WIELKOSC * ZUZYCIE( S, P ))
    <= ZASOBY( S );
  );

  ! Funkcja celu;
  [CEL] MAX =
    @SUM( PRODUKT : WIELKOSC * CENA );
END
```

Różnice pojawiają się tylko w sekcji DATA, w której widoczne są funkcje wczytujące dane z zanurzonego obiektu, jest nim skoroszyt Dane.xls. Podobnie jak poprzednio wykorzystana została funkcja @OLE. Dane są wczytywane do uprzednio zdefiniowanych zakresów: CENA, ZASOBY, ZUZYCIE.

6. Podsumowanie

W pracy przedstawiono możliwość wykorzystania języka LINGO do opisu problemu optymalizacyjnego w arkuszu kalkulacyjnym MS EXCEL. Jest to alternatywa względem dodawanego standardowo do arkusza solvera firmy Frontline Systems, Inc. Przewaga tego podejścia polega na tym, iż zapis modelu matematycznego jest dokonywany w specjalnie zdefiniowanym języku, nazywanym tutaj języ-

kiem LINGO. Dzięki jego zastosowaniu można modelować złożone relacje zachodzące pomiędzy obiektami w modelowanym systemie, zachowując przy tym przejrzystość zapisu modelu. Dzięki osadzaniu modeli LINGO, co zostało pokazane w niniejszym artykule, możliwe jest podążanie za trendami pojawiającymi się w systemach optymalizacyjnych, polegającymi na kładzeniu nacisku na prezentację otrzymanych wyników. Wyniki uzyskiwane za pomocą solverów środowiska LINGO można prezentować w arkuszu kalkulacyjnym w postaci rysunków i tabel.

Literatura

- Fourer R., Gay D.M., Kernighan B.W., *A modeling language for mathematical programming*, "Management Science" 1990 vol. 36, s. 519-554.
- Gutenbaum J., *Modelowanie matematyczne systemów*, Omnitech Press, Warszawa 1992
- LINGO 10.0 User's Manual*, LINDO Systems Inc. Chicago 2006, <http://www.lindo.com>.
- Schrage L., Cunningham K., *Demo LINGO/PC: Language for Interactive General Optimization, version 1.04a*, LINDO Systems Inc., Chicago 1988.
- Trzaskalik T., *Wprowadzenie do badań operacyjnych z komputerem*, wyd. 2 zm., PWE, Warszawa 2008.

ON USING OPTIMIZATION LANGUAGES IN SPREADSHEETS

Increased competition involves the need for making better decisions. In their selection can help mathematical methods. The use of mathematical methods in the decision-making process is the subject of operations research. The complexity of modern economic environment leads to the complexity of the models in operations research. To solve the tasks of optimization, specialized software packages are necessary. This paper presents the possibility of using in spreadsheets specially defined in those packages languages. Advantages and disadvantages are also discussed.