C3

INSTYTUT ORGANIZACJI I ZARZĄDZANIA
POLITECHNIKI WROCŁAWSKIEJ

Komunikat nr 132

ZAGADNIENIA PRZYDZIAŁU ZADAŃ W OPARCIU O MODELE DYSKRETNE W SYSTEMIE TRANSPORTOWYM

(rozprawa doktorska)

Michał Kulej

Promotor: doc. dr inż. Wiesław M.Grudzewski

Słowa kluczowe: system transportowy, planowanie przewozów, model, algorytm.

WROCŁAW 1976

mgr Michał Kulej

Zakład Systemów Zarządzania w Przemyśle

Komunikat wpłynął do Działu Wydawnictw Pol. Wrocł. dnia 23 I 1976 r.

SPIS TREŚCI

WSTĘP

Znaczna część ciągle rosnących zadań przewozowych w gospodarce narodowej jest realizowana przez transport samochodowy. Wzrost zadań jak również wzrost kosztów przewozu podnoszą wagę ich efektywnego wykorzystania. Współczesne zarządzanie wymaga ciągłego podejmowania wielu decyzji w jednostkach spedycyjnych, których ostatecznym celem jest zwiększenie efektywności wykorzystania transportu samochodowego, przede wszystkim poprzez próbę eliminacji nadmiernej ilości pustych przejazdów samochodów.

Centralizacja zarządzania transportem samochodowym stwarza niewątpliwie większe możliwości bardziej efektywnego wykorzystania środków transportowych, jednocześnie jednak powoduje wzrost stopnia złożoności systemu transportowego, co z kolei komplikuje i stwarza dodatkowe trudności w procesie kierowania takim systemem [16]. Aby nie zaprzepaścić możliwości, jakie niesie koncentracja spedycji, niezbędne jest stałe doskonalenie metod kierowania systemem transportowym.

Jedną z najistotniejszych funkcji kierowania systemem transportowym, która ma decydujący wpływ na efektywność wykorzystania środków transportowych, jest rozdział zadań spedycyjnych między poszczególne środki transportowe. Duża liczba zadań transportowych jak również środków do ich wykonania uniemożliwia efektywną realizację tej funkcji jedynie w oparciu o doświadczenie i intuicję człowieka. Niezbędne jest tutaj zastosowanie odpowiednich naukowych metod i algorytmów, wspartych dużymi możliwościami współczesnych systemów informatycznych.

W niniejszej pracy rozważa się system transportowy, działający na określonej sieci drogowej. System ten posiada środki transportowe rozlokowane w wyróżnionych punktach tej sieci (zwanych dalej bazami) . Zadaniem tego systemu jest przewóz - na rzecz klientów - ładunków z jednych miejsc sieci drogowej do innych, z uwzględnieniem konkretnych ograniczeń. Będzie się badać ten system w ustalonym przedziale czasu, w którym są znane wymagania klientów oraz żąda się od niego, by te wymagania zostały w tym czasie wykonane.

Przyjmuje się, że kryterium jakości działania tego systemu będzie minimalizacja długości drogi przebytej przez wszystkie środki transportowe zaangażowane do realizacji przewozów. Przy tych ograniczeniach realizacja rozważanej funkcji kierowania sprowadza się do zagadnienia określenia dróg, po których mają poruszać się środki transportowe oraz zadań (zleceń) , które powinny być na tych drogach realizowane tak, by uzyskać możliwie najwyższy stopień realizacji przyjętego kryterium. Rozważa się to kryterium pod kątem metod otrzymywania efektywnych rozwiązań optymalnych lub suboptymalnych. Nie będzie się natomiast rozważać losowych zmian popytu lub losowych zakłóceń przewozów typu trudności komunikacyjnych, awarii środków transportowych itp. Zakłada się, że wszystkie występujące wielkości są wielkościami zdeterminowanymi. Badanie rozważanego kryterium jakości działania przeprowadza się z uwzględnieniem pewnego aspektu struktury organizacyjnej systemu transportowego, a mianowicie przy założeniu, że dany system posiada środki transportowe rozlokowane w jednej bazie lub też w wielu bazach.

W części pierwszej pracy sformułowano zagadnienie przydziału zadań dla systemu transportowego, posiadającego jedną bazę i dokonano przeglądu i analizy metod otrzymywania rozwiązań suboptymalnych

[2], [3], [5], [7], [8] . Metody te stanowią efektywne narzędzie w procesie kierowania systemami transportowymi posiadającymi jedną bazę.

Zagadnienie przydziału zadań w systemie transportowym posiadającym wiele baz było w literaturze formułowane [1], [14], [18] . Jednak modele matematyczne, ujmujące wszystkie aspekty tego problemu, są trudne do rozwiązania, natomiast modele przybliżone mogą dawać rozwiązania mało dokładne.

W drugiej części pracy sformułowano model szczególny rozważanego problemu. W zagadnieniu tym system transportowy dysponuje środkami transportowymi, z których każdy może realizować dowolne zlecenie. Środki transportowe mogą poruszać się po zamkniętych trasach o długości nie przekraczającej określonego z góry limitu a ilość tych środków - w każdej z baz - zapewnia realizację wszystkich wymagań klientów. Dla tego modelu podano metodę otrzymywania rozwiązań optymalnych i suboptymalnych. Metoda ta składa się z dwóch części. W pierwszej podaje się metodę cechowania konstrukcji tras o pewnych własnościach dla środków transportowych. Nie konstruuje się wszystkich możliwych tras, a tylko takie, które mogą wchodzić do rozwiązania optymalnego. W części drugiej przedstawia się metodę typu ograniczeń i podziału konstrukcji planu optymalnego, która wykorzystuje trasy skonstruowane w pierwszej części metody.

Część trzecia pracy prezentuje numeryczną realizację algorytmów z części drugiej. W pracy podano program w języku Algol 1204, oznaczony symbolem MC, realizujący metodę cechowania oraz program, oznaczony symbolem MOP, realizujący konstrukcję planu optymalnego. Podano również pewną modyfikację programu MOP pozwalającą w przypadku, gdy nie można, ze względu na czas obliczeń maszyny cyfrowej, otrzymać rozwiązania optymalnego, uzyskać rozwiązanie przybliżone.

Podana w pracy metoda zaprogramowana na szybką, o stosunkowo dużej pamięci operacyjnej, maszynę cyfrową (np. typu ODRA 1305) może stanowić efektywne narzędzie w procesie kierowania systemami posiadającymi wiele baz.

R o z d z i a ł I

ZAGADNIENIE PRZYDZIAŁU ZADAŃ W SYSTEMIE TRANSPORTOWYM POSIADAJĄCYM JEDNĄ BAZĘ

W niniejszym rozdziale rozważy się system transportowy, w którym dana jest sieć drogowa $\langle I, U, c \rangle$, gdzie $I = \{1, 2, \ldots, n\}$ jest zbiorem wierzchołków sieci ponumerowanych liczbami naturalnymi, U jest zbiorem łuków w sieci, przy czym elementy tego zbioru oznaczono przez (i, j), $i \in I$, $j \in I$, $i \neq j$, natomiast $c = c(i, j)$ jest funkcją określoną na elementach (łukach) zbioru U, której wartościami są długości łuków.

W niektórych wierzchołkach $i \in I$ tej sieci istnieje zapotrzebowanie na określony produkt, wynoszące q_i. Zbiór wszystkich takich wierzchołków oznacza się dalej przez Q, gdzie $Q \subset I$. Produkt będzie dostarczany przy użyciu środków transportowych o znanej ładowności Ł, z jednego wierzchołka zbioru I, który będzie się nazywać bazą i oznaczać numerem 1. Ładowność środka transportowego jest mierzona przy pomocy tych samych jednostek co zapotrzebowanie na produkt. Środki transportowe, będące w dyspozycji rozpatrywanego systemu transportowego, mogą poruszać się po trasach, stanowiących zamknięte kontury, przechodzące przez bazę oraz odpowiedni podzbiór wierzchołków zbioru Q. Długość takich tras nie może przekroczyć pewnego, ustalonego z góry limitu, który dalej oznacza się przez dd.

Rozważany system transportowy ma zrealizować wszystkie wymagania odbiorców, tzn. zaspokoić popyt na produkt w każdym z wierzchołków zbioru Q. Za kryterium jakości działania systemu przyjmuje się możliwie jak największe skrócenie łącznej długości tras wszystkich środków transportowych zaangażowanych w realizację przewozów.

Rozważany we wstępie aspekt kierowania takim systemem transportowym można teraz sformułować jako zagadnienie wyznaczania tras dla środków transportowych tak, by nie przekroczyć ładowności tych środków oraz by łączna długość wszystkich tras była minimalna. Dla scharakteryzowania pewnych wariantów tego zagadnienia oraz omówienia głównych idei i metod otrzymywania rozwiązań przybliżonych wprowadzi się następujące oznaczenia i dodatkowe założenia:

1.1. Przez $\mu(i, j)$, dla $i \in I$, $j \in I$ oznacza się najkrótszą drogę $(i, i_1, \ldots, i_k, j)$ od wierzchołka i do wierzchołka j w sieci $\langle I, U, c \rangle$, gdzie $(i_r, i_{r+1}) \in U$ dla $r = 0, 1, \ldots, k$, przy czym $i_0 = i, i_{k+1} = j$. Długość tej drogi oznacza się przez $c(\mu(i, j))$.

1.2. Trasa $\mathcal{K}$ środka transportowego jest konturem zamkniętym

$(i_0, \ldots, i_{r_1}, \ldots, i_{r_2}, \ldots, i_{r_k}, \ldots, i_{r_{k+1}})$,

gdzie $i_{r_1} \in Q, \ldots, i_{r_k} \in Q$

$1 < r_1 < \ldots < r_k < r_{k+1}$

$(i_j, i_{j+1}) \in U$, $j = 0, 1, \ldots, r_{k+1}$.

Trasę tę będziemy mogli jednoznacznie, ze względu na rozpatrywane kryterium, zadać w następujący sposób:

$\mathcal{K} = (1, i_{r_1}, \ldots, i_{r_k}, 1)$ tzn. podać tylko ciąg wierzchołków, do których kolejno środek transportowy dostarcza produkt. Wyjściowy zapis trasy można otrzymać przyjmując, że droga od wierzchołka i_{r_l} do wierzchołka $i_{r_{l+1}}$ jest $\mu(i_{r_l}, i_{r_{l+1}})$ dla $l = 0, 1, \ldots, k$, gdzie $i_{r_0} = i_{r_{k+1}} = 1$.

1.3. Dalsze rozważania prowadzi się w sieci $\langle I', U', c' \rangle$, gdzie

$I \supset I' = Q \cup \{1\}$,

$U' = \{(i, j) : i \neq j, \ i \in I', \ j \in I'\}$

$c'(i, j) = c(\mu(i, j))$.

W dalszym ciągu tego rozdziału przyjmuje się, że graf $\langle I', U' \rangle$ będzie grafem nieskierowanym bez pętli.

1.4. Przyjmuje się dalej, że ładowność środków transportowych jest istotnie większa od popytu w wierzchołkach zbioru Q, jednak nie jest tak duża, by jeden środek transportowy mógł na jednej trasie zrealizować popyt wszystkich odbiorców (wtedy zagadnienie sprowadza się do problemu komiwojażera).

Zatem dalej zakłada się, że

$$(1) \quad \max_{i \in Q} q_i \ll Ł.$$

Uwaga. Przypadek, gdy istnieje wierzchołek $\alpha \in Q$ taki, że $q_\alpha > Ł$ można sprowadzić do rozpatrywanego wyżej. Niech

$$r = \max\left\{ i \in \{1,2,\ldots\} : Ł \cdot i \le q_\alpha \right\},$$

natomiast b będzie resztą z dzielenia modulo r liczby q_α tzn. $q_\alpha = Ł \cdot r + b$.

Na trasę $(1, \alpha, 1)$ przydzielić należy wtedy r środków transportowych i rozwiązywać rozważany problem przyjmując, że popyt w wierzchołku α wynosi b. Warunek (1) jest spełniony.

Metoda Dantziga-Ramsera

Historycznie pierwsze sformułowanie rozważanego problemu zostało podane przez G.B.Dantziga i J.H.Ramsera w roku 1959 w pracy [5]. W ich sformułowaniu nie występuje jedynie ograniczenie, by długość tras była mniejsza od z góry określonego limitu. Założenie to jednak występuje w niejawnej postaci; wynika to z tego, że środek transportowy nie może przejechać zbyt wielu wierzchołków zbioru I', gdyż mogłoby to spowodować przekroczenie jego ładowności (środek transportowy bez ładunku wraca do bazy).

Aby przedstawić ideę metody podanej w pracy [5], zakłada się, że $I' = \{1, i_1, \ldots, i_m\}$, gdzie $i_1, \ldots, i_m$ są tak uporządkowane, że $q_{i_{k-1}} \leq q_{i_k}$ dla $k = 2, \ldots, m$ oraz $N \approx \log_2 t$, gdzie liczba t jest tak dobrana, by spełnione były nierówności:

$$\sum_{k=1}^{t} q_{i_k} \leq Ł, \qquad \sum_{k=1}^{t+1} q_{i_k} > Ł.$$

Metoda Dantziga-Ramsera jest oparta na następującym, oczywistym intuicyjnie fakcie: zbiór tras, które przechodzą przez wszystkie wierzchołki zbioru Q i zaspokajają zapotrzebowanie tych wierzchołków ma tę własność, że każda trasa musi przechodzić przez co najmniej dwa wierzchołki ze zbioru I'.

Wyznaczenie zatem zbioru par punktów leżących możliwie blisko siebie i przeprowadzenie przez te wierzchołki tras, powinno stanowić w miarę dobre przybliżenie rozwiązania optymalnego. Z kolei tak otrzymane pary mogą być znowu grupowane, aby otrzymać zbiory wierzchołków leżących blisko siebie w sensie długości trasy przechodzącej przez te punkty. Postępowanie to powtarza się N-krotnie, przy czym liczba N jest tak dobrana, aby w żadnym z etapów metody liczba wierzchołków w wyznaczanych zbiorach punktów nie przekroczyła liczby t. Liczba t jest maksymalną liczbą wierzchołków, jakie mogą znajdować się na jednej trasie, aby bez przekroczenia ładowności Ł środka transportowego można dostarczyć do tych wierzchołków wymaganą ilość produktu.

W pierwszym kroku metody wyznacza się taki zbiór par wierzchołków ze zbioru I', by każdy wierzchołek z tego zbioru, z wyjątkiem być może wierzchołka o numerze 1, należał do jednej pary oraz by łączna długość wszystkich dróg między wierzchołkami, stanowiącymi pary, była minimalna. Problem wyznaczenia takiego zbioru sprowadza się do rozwiązania następującego zagadnienia programowania całkowitoliczbowego:

Wyznaczyć: minimum $z = \sum_{u \in U'} c''(u) \cdot x_u$

(2) $$\sum_{u \in A(i)} x_u = 1 \quad \text{dla} \quad i \in Q$$

$$x_u = \begin{cases} 1 & \text{gdy wierzchołki, będące końcami krawędzi } u \text{, zostaną wybrane do zbioru par} \\ 0 & \text{w przeciwnym wypadku,} \end{cases}$$

gdzie $A(i)$ jest zbiorem krawędzi grafu $\langle I', U' \rangle$ incydentnych z wierzchołkiem $i \in Q \subset I'$, a $c''(u)$ jest długością krawędzi $u \in U'$, czyli $c''(u) = c'(u)$. Rozwiązanie tego zagadnienia daje zbiór, dalej oznaczany przez I^1, którego elementami są pary wierzchołków ze zbioru $Q \cup \{1\}$. Przeprowadzenie tras przez każdy z elementów zbioru I^1 daje zbiór tras realizujących popyt we wszystkich wierzchołkach zbioru Q. W k-tym kroku jest dany zbiór I^{k-1}, wyznaczony jako rozwiązanie optymalne odpowiedniego zagadnienia programowania całkowitoliczbowego z kroku k-1. Elementami zbioru I^{k-1} są podzbiory wierzchołków ze zbioru $Q \cup \{1\}$. Z elementów zbioru I^{k-1} wybiera się taki podzbiór par jego elementów, dalej oznacza się go przez I^k, by każdy wierzchołek ze zbioru Q należał dokładnie do jednej takiej pary oraz, by łączna długość tras przechodzących przez wierzchołki - należące do elementów zbioru I^k - była minimalna. Prowadzi to zatem do rozwiązywania analogicznego zagadnienia programowania całkowitoliczbowego; z tym, że funkcja $c''(V_1, V_2)$, gdzie $V_1, V_2 \in I^{k-1}$, $V_1 \neq V_2$, jest długością optymalnej trasy komiwojażera przechodzącej przez wierzchołki zbioru $V_1 \cup V_2 \cup \{1\}$.

Proponowane w metodzie postępowanie nie zapewnia otrzymania rozwiązania optymalnego i chociaż w przykładach cytowanych w [5] uzyskano dobre przybliżenia rozwiązania optymalnego, to jednak niżej wymienione mankamenty tej metody nie pozwalają na efektywne jej zastosowania.

(a) Rozwiązanie efektywne zagadnienia (2) nie jest w ogólnym przypadku zapewnione, a proponowana przez autorów przybliżona metoda - oparta na programowaniu liniowym - może dawać rozwiązania, mające ujemny wpływ na jakość ostatecznego rozwiązania.

(b) Metoda wymaga, począwszy od drugiego etapu, rozwiązywania wielu zagadnień komiwojażera w celu obliczenia tablicy c". Ma to istotny wpływ na czas obliczeń; na przykład dla zagadnienia o stuwierzchołkowej sieci, aby obliczyć tablicę c'', może zaistnieć potrzeba rozwiązania trzystu dziewięciowierzchołkowych zagadnień komiwojażera. Przy czasie rozwiązywania jednego takiego zagadnienia co najmniej kilkanaście sekund wydłuża to znacznie czas obliczeń.

(c) W r-tym kroku metody konstruuje się zbiory wierzchołków, w których łączny popyt nie przekracza wielkości $Ł/_2N-r$, co może być powodem nie uwzględniania pewnych tras i tym samym daje gorsze przybliżenia.

Metoda Clarke'a-Wright'a

Mankamenty metody Dantziga-Ramsera wymienione w punktach (a), (b), (c), stanowiące przyczynę małego jej zastosowania praktycznego, zostały wyeliminowane lub częściowo złagodzone w pracy G.Clarke'a i J.W.Wright'a [2].

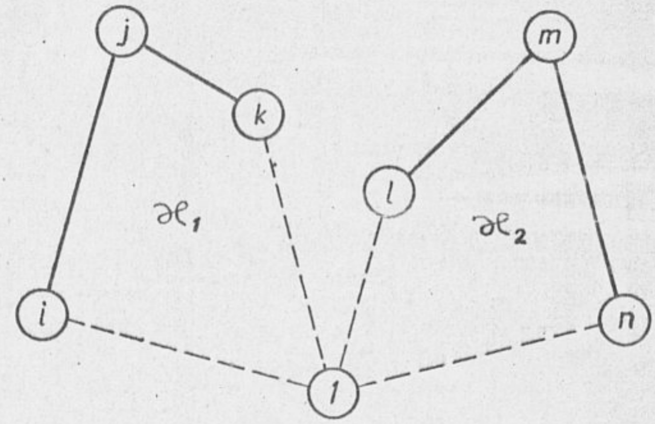
Iteracyjna metoda przez nich zaproponowana daje również przybliżone rozwiązanie zagadnienia sformułowanego na początku tego rozdziału. W pierwszym kroku tej metody, podobnie jak w metodzie Dantziga-Ramsera, wyznaczamy początkowy zbiór tras realizujący wszystkie zapotrzebowania, złożony z tras przechodzących tylko przez jeden wierzchołek zbioru Q oraz wierzchołek 1, tzn. z tras postaci

$$(1, i, 1), \quad i \in Q.$$

W następnych krokach łączy się, w odpowiedni sposób, dwa wierzchołki leżące na różnych trasach tak, by popyt wierzchołków leżących na nowo powstałej trasie i należących do zbioru Q nie przekroczył wielkości L. Tym samym eliminuje się problemy wymienione w punkcie (c).

Nie jest również konieczne rozwiązywanie w każdym kroku serii zagadnień komiwojażera, gdyż łączenie dwóch wierzchołków jest tak dokonywane, aby otrzymać nową trasę, a nie tylko nieuporządkowany zbiór wierzchołków, przez które powinna ona przechodzić. Jedynie w końcowym kroku proponuje się rozwiązać kilka zagadnień komiwojażera na zbiorach wierzchołków, należących do ostatnio wyznaczonego zbioru tras. Może to ulepszyć rozwiązanie, przy czym liczba tych zagadnień jest już znacznie mniejsza. Dzięki temu znacznie złagodzono mankamenty wymienione w (b).

Zasadniczym pomysłem metody, pozwalającym ominąć trudności wymienione w (a), jest sposób wyboru dwóch wierzchołków leżących na różnych trasach, które to wierzchołki zostaną tak połączone, aby należały do jednej nowej trasy. Dla każdych dwóch wierzchołków ze zbioru I' podaje się prosty sposób obliczenia, o ile skróci się lub wydłuży łączna długość tras, gdy dokona się połączenia tych wierzchołków. Skrócenie to lub wydłużenie, gdy rozważa się wierzchołki $j, m \in Q$, oznacza się przez $s(j,m)$ i oblicza następująco:
niech wierzchołek j leży na trasie $\mathcal{H}_1 = (1,\dots,i,j,k,\dots,1)$, a wierzchołek m na trasie $\mathcal{H}_2 = (1,\dots,l,m,n,\dots,1)$ (rys. 1). Trasy $\mathcal{H}_1, \mathcal{H}_2$



Rys. 1.

zadane są przez podanie tylko wierzchołków należących do zbioru $Q \cup \{1\}$, przez które przechodzi dana trasa (jak w punkcie 1.2). Przez każdy wierzchołek $i \in Q$ powinna przechodzić tylko jedna trasa. Z uwagi na to, że łączy się wierzchołki j oraz m, należy dokonać usunięcia jednej z czterech niżej wymienionych par krawędzi:

(α) (i, j), (l, m)

(β) (i, j), (m, n)

(γ) (j, k), (l, m)

(δ) (j, k), (m, n)

oraz odpowiednio w każdym z tych przypadków dodać następujące krawędzie:

(α) (1, i), (1, l), (j, m)

(β) (1, i), (1, n), (j, m)

(γ) (1, k), (1, l), (j, m)

(δ) (1, k), (1, n), (j, m).

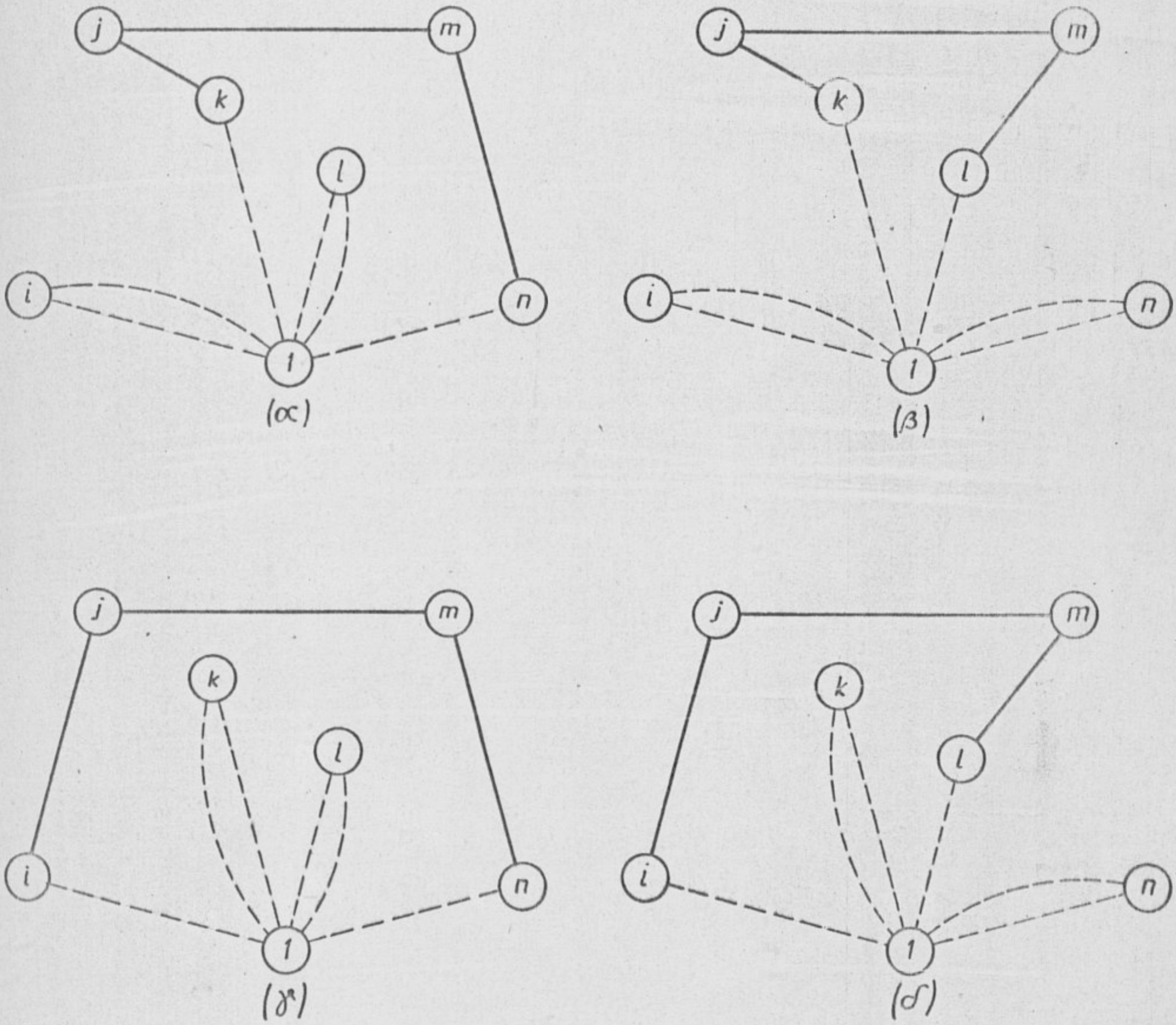
Powstają wtedy trasy (rys. 2), których łączna długość różni się od łącznej długości tras $\varkappa_1$ i $\varkappa_2$ o wielkość wynoszącą odpowiednio:

(α) $c'(i,j) - c'(1,i) + c'(l,m) - c'(1,l) - c'(j,m)$

(β) $c'(i,j) - c'(1,i) + c'(m,n) - c'(1,n) - c'(j,m)$

(γ) $c'(j,k) - c'(1,k) + c'(l,m) - c'(1,l) - c'(j,m)$

$(\delta)\ c'(j,k) - c'(1,k) + c'(m,n) - c'(1,n) - c'(j,m).$



Rys. 2.

Za $s(i,j)$ przyjmuje się maksymalną z tych wartości. Jak wynika z powyższych rozważań, łączenie w ten sposób dwóch wierzchołków, należących do różnych tras, może prowadzić do powstania więcej niż jednej nowej trasy. Aby tego uniknąć, w metodzie, łączy się tylko te wierzchołki j oraz m, które leżą na trasach bezpośrednio przy wierzchołku 1, tzn. wierzchołek j powinien leżeć na trasie $(1,\ldots,j,1)$ lub $(1,j,\ldots,1)$, natomiast wierzchołek m na trasie $(1,\ldots,m,1)$ lub $(1,m,\ldots,1)$, (wtedy $s(j,m) = c'(1,j) + c'(1,m) - c'(j,m)$). W każdym kroku metody wybiera się parę wierzchołków o maksymalnej wartości s(i, j) i dokonuje łączenia tych wierzchołków o ile trasa, która powstanie, ma długość nie przekraczającą dd i przechodzi przez zbiór wierzchołków Q o łącznym popycie nie przekraczającym Ł. Postępowanie kontynuuje się do momentu, gdy w którymś z kroków metody nie będzie już dodatnich wartości s(i, j).

Metoda powyższa jest bardziej wygodna w obliczeniach na maszynę cyfrową i daje wyniki zbliżone do wyników uzyskanych metodą Dantziga-Ramsera. Oprócz tego jest znacznie efektywniejsza, gdyż np. w pracy [2] dla przykładu z siecią o trzydziestu wierzchołkach uzyskano wyniki w czasie o 17% krótszym niż w metodzie uprzednio omawianej.

Metoda Clarke'a-Wrighta pozwala również otrzymywać rozwiązania przybliżone w przypadku, gdy system transportowy dysponuje środkami transportowymi o różnych pojemnościach, oraz wprowadzać częściowo założenia o ograniczonej liczbie środków transportowych danego typu. W tym ostatnim przypadku, dla zapewnienia uzyskania rozwiązania zakłada się, że środków transportowych o najmniejszej pojemności jest dostatecznie dużo.

Rozważana metoda ma jeszcze tę zaletę, że stosunkowo łatwo można wprowadzać do niej pewne modyfikacje, wynikłe z praktycznych doświadczeń, zwiększające efektywność metody. Modyfikacje takie są rozważane np. w pracy Gaskell'a [7].

Metoda trójoptymalnych tras Christofidesa-Eilona

Metodą, która w porównaniu z omówionymi wcześniej, daje rozwiązania najbardziej zbliżone do optymalnych jest metoda trójoptymalnych tras podana przez Christofidesa i Eilona w pracy [3].

W metodzie tej problem wyznaczania zbioru tras, stanowiących rozwiązanie rozważanego problemu, na sieci $\langle I, U, c \rangle$ sprowadza się do wyznaczenia jednej trasy przechodzącej przez wszystkie wierzchołki nowej sieci $\langle I'', U'', c'' \rangle$ dokładnie raz. Nowa sieć powstaje przez usunięcie ze zbioru I' wierzchołka o numerze 1, dołączenie na to miejsce N fikcyjnych wierzchołków, dołączenie nowych połączeń między tymi wierzchołkami i wierzchołkami zbioru $I' \cup \{1\}$ oraz następujące określenie c'' :

$$c''(i,j) = \begin{cases} c'(i,j) & \text{jeśli } i \notin N, \ j \notin N, \ i,j \in Q, \\ c'(1,j) & \text{jeśli } i \in N, \ j \in Q, \\ \infty & \text{jeśli } i \in N, \ j \in N. \end{cases}$$

Rozważany problem sprowadza się zatem do rozwiązania zagadnienia komiwojażera z dodatkowymi ograniczeniami na długość tras i ładowność środków transportowych. Te ograniczenia dają się uwzględnić w przybliżonej metodzie rozwiązywania zagadnienia komiwojażera, tzw. metodzie trójoptymalnych tras. Idea tej metody jest oparta na następującym fakcie: dobre przybliżenie optymalnej trasy komiwojażera stanowią trasy, w których usunięcie trzech dowolnych jej łuków i zastąpienie ich trzema innymi, nie może zmniejszyć długości takiej trasy. Istniejące metody konstrukcji takich tras [4],[15] , użyte do rozwiązywania rozważanego problemu, dały wyniki bardziej zbliżone do rozwiązań optymalnych niż wyniki otrzymane metodą Clarke'a-Wrighta. Czas obliczeń dla przykładów z siecią kilkudziesięciowierzchołkową był jednak kilkakrotnie dłuższy. Dla sieci o dużej liczbie wierzchołków (rzędu 100) rozważana metoda nie jest już tak efektywną i daje wyniki bardzo nieznacznie lepsze od metody Clarke'a-Wrighta. Rozpatrywane wyżej metody nie są zbyt efektywne i nie dają zbyt dobrych przybliżeń dla zagadnień o sieciach mających dużą liczbę wierzchołków.

Metoda Gilleta-Millera

Rozważy się jeszcze jedną metodę, która - na testowanych przykładach dla sieci o liczbie wierzchołków rzędu kilkudziesięciu - dawała wyniki zbliżone do wyników otrzymanych metodą Christofidesa-Eilona, natomiast dla problemów o sieciach z liczbą wierzchołków powyżej pięćdziesięciu wyniki lepsze. Metoda ta tzw. "sweep-algorythm" podana w 1971 roku w [8] przez Gilleta i Millera, podobnie jak i poprzednie metody, opiera się na idei wyznaczania w odpowiedni sposób zbiorów wierzchołków, przez które mają przechodzić trasy stanowiące rozwiązanie problemu. W tym celu na płaszczyźnie zawierającej graf $\langle I', U' \rangle$ wprowadza się układ współrzędnych biegunowych o początku w wierzchołku 1. Każdy wierzchołek $i \in Q$ ma wtedy przyporządkowaną parę liczb: $\varphi(i)$ - współrzędną biegunową kąta oraz $r(i)$ - długość promienia. Wierzchołki zbioru Q numeruje się tak, by wierzchołki o większych numerach miały większe wartości współrzędnej kątowej tzn. jeśli dla wierzchołków i oraz j zachodzi $i < j$, to $\varphi(i) < \varphi(j)$. W przypadku gdy $\varphi(i) = \varphi(j)$ to $i < j$, jeśli $r(i) < r(j)$.

W metodzie wyznacza się zbiory wierzchołków dla kolejnych tras, zaczynając od wierzchołków o najmniejszych wartościach współrzędnej kątowej φ. Do jednego takiego zbioru włącza się maksymalną liczbę wierzchołków, których łączny popyt nie przekracza ładowności Ł środka transportowego. Jednocześnie sprawdza się, czy długość trasy przechodzącej przez wszystkie wierzchołki kolejno wyznaczonego zbioru wierzchołków nie przekracza dd. Sprawdzenie powyższego warunku wymaga rozwiązania zagadnienia komiwojażera. Po wyznaczeniu w wyżej podany sposób trasy rozpatruje się możliwość usunięcia z tej trasy jakiegoś wierzchołka i włączenia do niej jednego lub kilku wierzchołków topologicznie bliskich do niej i dotąd nie należących jeszcze do

20

żadnej z tras. Wybór wierzchołka, który ma być usunięty z trasy, odbywa się przez minimalizację następującej, otrzymanej z praktycznych doświadczeń, funkcji: min $z = r(i) + \varphi(i) \cdot avr$. Minimum bierze się po wszystkich wierzchołkach zbioru Q, należących do danej trasy, a avr jest średnią długością promienia dla wszystkich wierzchołków sieci $\langle I', U', c' \rangle$. Wyżej podaną konstrukcję prowadzi się do momentu, gdy już utworzone trasy przechodzą przez wszystkie wierzchołki zbioru Q. Wynik takiego postępowania może zależeć od kolejności, w jakiej wierzchołki są włączane do kolejno konstruowanych tras, dlatego w metodzie proponuje się kilkakrotne powtórzenie procesu konstrukcji tras, zaczynając każdorazowo od innego wierzchołka jako pierwszego. Wybiera się wariant dający najlepsze rozwiązanie.

Podsumowanie

Rozważane w tym rozdziale metody wyznaczania tras (przydziału zadań) w systemie transportowym cechuje (na przytoczonych przykładach) różna dokładność i efektywność rozumiana jako czas obliczeń EMC. Nie można jednak na tej podstawie ogólnie twierdzić, że któraś z wyżej podanych metod jest najlepsza. W każdym przypadku praktycznym należy przeprowadzić badania, które pozwoliłyby wybrać metodę najlepiej uwzględniającą specyfikę danego problemu np. jeśli sieć $\langle I', U', c' \rangle$ dla konkretnego problemu cechuje się tym, że występują w niej grupy topologicznie bliskich sobie wierzchołków, przy czym liczba wierzchołków w każdej takiej grupie nie jest zbyt duża (chodzi o to, by ładowność środka transportowego wystarczyła dla zaspokojenia popytu w takiej grupie wierzchołków), to prawdopodobnie najefektywniejszą metodą rozwiązania tego problemu będzie metoda Gilleta-Millera.

Główne cechy systemu transportowego

Zasadnicze cechy systemu transportowego, do kierowania którym mogą być użyte dyskutowane wyżej metody:

(i) dokonuje się przewozów z jednego wierzchołka sieci (np. centralny magazyn) do zbioru innych wierzchołków sieci jednorodnego produktu;

(ii) przewozów dokonuje się środkami transportowymi, które po przejechaniu określonej drogi muszą wrócić do miejsca wyjazdu (bazy) , przy czym długość tej drogi jest z góry ograniczona;

(iii) środki transportowe, jakimi dysponuje system transportowy, mają ograniczoną ładowność. Istotne jest, aby wielkość popytu w wierzchołkach sieci była znacznie mniejsza od ładowności tych środków transportowych;

(iv) należy w całości zaspokoić popyt.

W następnym rozdziale będzie się rozważać system transportowy, który scharakteryzowano następująco: zachowano cechy (ii) , (iv) , natomiast zrezygnowano lub znacznie zmodyfikowano cechy (i) , (iii) .

Rozdział II

ZAGADNIENIE PRZYDZIAŁU ZADAŃ W SYSTEMIE TRANSPORTOWYM POSIADAJĄCYM WIELE BAZ

W pierwszym rozdziale rozważano system transportowy, posiadający tylko jedną bazę. W wielu systemach transportowych, np. typu PKS, występuje często więcej niż jedna baza, przy czym kierowanie takimi systemami odbywa się jednak centralnie w zakresie przyjmowania zamówień na przewozy, dysponowania środkami transportowymi w bazach, oceny jakości działania tych systemów itp. Zakłada się, że rozważane w tym rozdziale systemy transportowe posiadają więcej niż jedną bazę i, podobnie jak w pierwszym rozdziale, kryterium jakości działania takich systemów będzie minimalizacja łącznej długości tras wszystkich środków transportowych (z różnych baz), zaangażowanych do realizacji przewozów. Przyjmuje się dalej, że przewozy zwane zleceniami mogą odbywać się między dwoma dowolnymi wierzchołkami w sieci $\langle I, U, c \rangle$ zdefiniowanej jak w rozdziale I. Poza tym rezygnuje się z założenia (iii) z rozdziału I, gdyż w systemach transportowych szczególnie takich, w których stosuje się kontenery, wymagania klientów są w przybliżeniu równe ładowności środków transportowych. Z tego też powodu wprowadza się założenie, że na dowolnym łuku środek transportowy może realizować najwyżej jedno zlecenie.

Rozważany na wstępie aspekt kierowania takim systemem transportowym dotyczy okresu czasu, w którym są zadane zapotrzebowania na przewozy czyli zlecenia. Zlecenia te mają być wszystkie zrealizowane (jak w (iv) rozdziału I).

Problem rozważany w tym rozdziale polega na wyznaczeniu tras dla środków transportowych - przy spełnieniu ograniczeń na dopuszczalną długość tras - tak, by łączna długość tych tras była minimalna. Dla tego zagadnienia sformułuje się model i poda metodę otrzymywania rozwiązania optymalnego.

Pewne warianty rozważanego problemu oraz metody rozwiązywania modeli tych zagadnień podają prace [1], [11], [14], [18] . Z metod przybliżonych, rozwiązujących model zagadnienia ogólniejszego od sformułowanego przez nas, bardziej interesującą jest metoda podana w [1] . Natomiast metodę dokładną rozwiązywania najbardziej ogólnego modelu podaje praca [18] . W modelu tam rozważanym za kryterium jakości działania systemu transportowego przyjmuje się minimalizację kosztów eksploatacji środków transportowych i wprowadza się różne typy środków transportowych. Wprowadza się także dodatkowe ograniczenia wymagające, aby pewne zlecenia były realizowane tylko przez pewne typy środków transportowych, uwzględnia się różne prędkości dla środka transportowego z ładunkiem i bez ładunku. Metoda rozwiązywania tego modelu - oparta na programowaniu liniowym - jest jednak dla realnych zagadnień nieefektywna (mimo prac adaptujących zasadę dekompozycji Dantziga-Wolfa do rozwiązywania pewnych uproszczeń tego modelu np. [14]) a poza tym ma następujące wady:

a) rozwiązanie optymalne nie musi być dane w liczbach całkowitych (zmiennymi decyzyjnymi w tym modelu są ilości "pełnych" i "pustych" przejazdów środka transportowego przez dany łuk), a zaokrąglanie może prowadzić do znacznych błędów,
b) w rozwiązaniu optymalnym może wystąpić dla jakiegoś środka transportowego trasa złożona z rozłącznych konturów,
c) trasy dla środków transportowych będą przechodziły przez bazy w przypadku, gdy bazy będą miejscami początkowymi lub końcowymi realizacji pewnych zleceń.

W wielu systemach transportowych bazy dla środków transportowych nie spełniają warunków z punktu c i wtedy dyskutowany model może być mało przydatny, natomiast bardziej racjonalne może być zastosowanie modelu, który poniżej zostanie sformułowany.

Model zagadnienia

W tym rozdziale, jeżeli nie zdefiniuje się inaczej, będą używane pojęcia i oznaczenia wprowadzone w rozdziale I. Będzie się dalej używać również niżej podanych oznaczeń:

$I \supset B$ - zbiór baz sieci $\langle I, U, c \rangle$,

$Z = \{z_1, z_2, \ldots, z_{lz}\}$ - zbiór zleceń, które rozważany system transportowy ma zrealizować.

Każde zlecenie $z_j \in Z$ określa się następująco:

$z_j = (k_j, l_j)$, gdzie $k_j \in I$ jest wierzchołkiem, skąd zaczyna się, a $l_j \in I$ wierzchołkiem, w którym kończy się realizacja zlecenia dla $j = 1, 2, \ldots, lz$.

Rozważy się zagadnienie wyznaczenia optymalnego planu przydziału zadań, który pozwoliłby zrealizować wszystkie zlecenia. W tym celu wprowadza się pojęcie trasy z bazy i-tej, którą oznacza się przez:

$\mathcal{H}(i) \equiv (i, i_1, \ldots, i_v;\ f(i_1), \ldots, f(i_v))$.

$(i, i_1, \ldots, i_v)$ jest konturem w grafie $\langle I, U, c \rangle$ takim, że $i \in B$ oraz $(i_{j-1}, i_j) \in U$ dla $j = 1, 2, \ldots, v$, przy czym $i_O = i_v = i$; funkcja f jest określona na zbiorze wierzchołków konturu $(i, i_1, \ldots, i_v)$ następująco: dla $j = 1, 2, \ldots, v$ $f(i_j) = r$, gdzie $O \leq r \leq lz$ i posiada następującą własność: jeśli $f(i_j) = r \neq O$, to istnieją takie wierzchołki $i_\alpha = k_r$, $i_\beta = l_r$ konturu $(i, i_1, \ldots, i_v)$ oraz zlecenie $z_r = (k_r, l_r) \in Z$, że $\alpha + 1 \leq j \leq \beta$ a dla każdej liczby naturalnej l spełniającej nierówność $\alpha + 1 \leq l \leq \beta$ wierzchołek i_l należy do rozpatrywanego konturu przy czym $f(i_l) = r$. Kontur $(i, i_1, \ldots, i_v)$ wyznacza drogę środka transportowego a funkcja f określa zlecenia, które są na tej drodze kolejno realizowane. Jeśli dla pewnego j zachodzi $f(i_j) = 0$, to na łuku (i_{j-1}, i_j) nie realizuje się żadnego zlecenia (przejazd pusty). Przez trasę dopuszczalną rozumie się trasę $\mathcal{H}(i)$ spełniającą dodatkowe ograniczenie, a mianowicie taką, dla której łączna długość wszystkich łuków jej konturu $(i, i_1, \ldots, i_v)$ nie przekracza wielkości dd. Oznaczając przez $c(\mathcal{H}(i))$ długość trasy $\mathcal{H}(i)$, mamy dla trasy dopuszczalnej spełniony następujący warunek:

$$c(\mathcal{H}(i)) = \sum_{i=1}^{v} c(i_{j-1}, i_j) \leq dd.$$

Zdefiniuje się teraz plan przydziału zadań w systemie transportowym (nazywany dalej planem dopuszczalnym), który oznaczy się przez P, jako zbiór tras dopuszczalnych, spełniających pewne ograniczenia. Niech:

$\mathcal{H}_k(i)$ - oznacza trasę o numerze k-tym z bazy i-tej,

$H(\mathcal{H}_k(i))$ - będzie zbiorem zleceń realizowanych na trasie $\mathcal{H}_k(i)$, a

$t(i)$ - ilością tras z bazy i-tej występujących w planie P ($t(i) = 0$ - oznacza, że w planie P nie występują trasy wymagające użycia środków transportowych z bazy i-tej).

Wtedy

$$P = \bigcup_{i \in B} \bigcup_{1 \leq k \leq t(i)} \mathcal{H}_k(i),$$

przy tym spełnione są następujące trzy warunki:

2.1. jeśli $j \neq l$ lub $m \neq n$, to

$$H(\mathcal{H}_m(j)) \cap H(\mathcal{H}_n(l)) = \emptyset$$

dla $m = 1, 2, \ldots, t(j)$, $n = 1, 2, \ldots, t(l)$, $j, l \in B$.

2.2. $\bigcup_{i \in B} \bigcup_{1 \leq k \leq t(i)} H(\mathcal{H}_k(i)) = Z.$

2.3. $c(\mathcal{H}_k(i)) \leq dd$

dla $i \in B$, $k = 1,2,\ldots,t(i)$.

Warunek 2.3. zapewnia, że plan P składa się tylko z tras o długości nie przekraczającej dd. Warunek 2.1. gwarantuje, że w planie P żadne zlecenie nie będzie jednocześnie realizowane przez dwa środki transportowe. Natomiast warunek 2.2. zapewnia, by plan P realizował wszystkie zlecenia ze zbioru Z.

Oznaczając przez c(P) łączną długość tras w planie realizacji zadań P, a przez R zbiór wszystkich możliwych planów dopuszczalnych, rozważany problem można sformułować następująco:

wyznaczyć $\min_{P \in R} z = c(P)$.

Plan realizacji zadań P^*, dla którego funkcja celu z przyjmuje wartość minimalną, nazywa się dalej optymalnym planem realizacji zadań lub krótko planem optymalnym.

Podany dalej algorytm wyznaczania planu optymalnego składa się z dwu części. W pierwszej konstruuje się dla baz trasy dopuszczalne, które mogą wchodzić do planu optymalnego. Metoda konstrukcji polega na cechowaniu [9],[17], w odpowiedni sposób, pewnych wierzchołków sieci $\langle I, U, c \rangle$. W drugiej części algorytmu, wykorzystując pewien podzbiór tras skonstruowany w części I, konstruuje się metodę typu "podziału i ograniczeń" [14] plan optymalny.

Własności tras wchodzących do planu optymalnego

W celu uproszczenia zapisu trasy dopuszczalnej (nazywanej dalej trasą) oraz prostszego podania metody cechowania wyznaczania tras dopuszczalnych z dowolnej bazy, podaje się kilka własności tras, które mogą wchodzić do planu optymalnego.

Niech trasa

$\mathcal{H}(i) = (i, i_1, \dots, i_v, \ f(i_1), \dots, f(i_v))$

należy do planu optymalnego i realizuje kolejno l różnych zleceń

$z_{s_1}, \dots, z_{s_l}$. Wtedy istnieją takie wierzchołki

$i_{r_1}, i_{t_1}, \dots, i_{r_l}, i_{t_l}$, gdzie $r_1 < t_1 \leq r_2 < t_2 \leq \dots \leq r_l < t_l$,

należące do konturu $(i, i_1, \dots, i_v)$, że

$z_{s_j} = (i_{r_j}, i_{t_j}) \in Z$

i

$f(i_{r_j+1}) = \dots = f(i_{t_j}) = s_j$

dla $j = 1, 2, \dots, l$.

Kontur $(i, i_1, \dots, i_v)$ ma następujące własności:

2.4. droga łącząca wierzchołek i_{r_j} z wierzchołkiem i_{t_j}, $j = 1, 2, \dots, l$ jest najkrótszą drogą łączącą te wierzchołki w sieci $\langle I, U, c \rangle$.

2.5. drogi $(i_{t_j}, \dots, i_{r_{j+1}})$, $j = 1, 2, \dots, l-1$ są najkrótszymi drogami łączącymi wierzchołki i_{t_j} oraz $i_{r_{j+1}}$.

2.6. drogi $(i, \dots, i_{r_1})$ oraz $(i_{r_l}, \dots, i)$ są najkrótszymi drogami łączącymi wierzchołek i z wierzchołkiem i_{r_1} oraz wierzchołek i_{r_l} z wierzchołkiem l.

Rzeczywiście, gdyby kontur $(i, i_1, \dots, i_v)$ nie posiadał którejś z wymienionych własności, to wtedy można byłoby skonstruować nową trasę o długości mniejszej niż $c(\mathcal{H}(i))$, realizującą w tej samej kolejności i te same, co trasa $\mathcal{H}(i)$, zlecenia.

Powyższe stwierdzenie jest jednak sprzeczne z optymalnością planu, do którego należy rozpatrywana trasa.

Dalej, nie zmniejszając ogólności, ograniczono się tylko do tras posiadających własności 2.4, 2.5, 2.6. W dalszych rozważaniach wygodnie będzie wprowadzić inny zapis trasy. W tym celu rozważa się ciąg liczb naturalnych $(s_1, s_2, \ldots, s_N)$, spełniający następujące dwa warunki:

2.7. $1 \le s_j \le l_z$, $j = 1, 2, \ldots, N$

2.8. $s_i \neq s_j$, jeśli $i \neq j$.

Elementy ciągu $(s_1, s_2, \ldots, s_N)$ traktuje się jak numery zleceń tzn.

$$z_{s_1} = (k_{s_1}, l_{s_1}),\ z_{s_2} = (k_{s_2}, l_{s_2}), \ldots,\ z_{s_N} = (k_{s_N}, l_{s_N})$$

i konstruuje następujący kontur z bazy $i \in B$:

$(\ast)$ $(\mu(i, k_{s_1}),\ \mu(k_{s_1}, l_{s_1}),\ \mu(l_{s_1}, k_{s_2}), \ldots,\ \mu(k_{s_N}, l_{s_N}),\ \mu(l_{s_N}, i_v))$,

gdzie $i = i_v$. Jeśli długość tego konturu nie przekracza dd, to ciągowi $(s_1, \ldots, s_N)$ odpowiada trasa:

$\mathcal{H}(i) = (\mu(i, k_{s_1}), \mu(k_{s_1}, l_{s_1}), \ldots, \mu(l_{s_N}, i_v);\ f(i_1), \ldots, f(i_v))$

przy czym funkcja f przyjmuje dla $j = 1, 2, \ldots, N$ wartość s_j na wszystkich, różnych od k_{s_j}, wierzchołkach drogi $\mu(k_{s_j}, l_{s_j})$, natomiast wartość zero na pozostałych wierzchołkach konturu $(\ast)$. Trasa $\mathcal{H}(i)$ spełnia, z konstrukcji, warunki 2.4, 2.5 i 2.6. Jak z powyższego wynika, każdemu ciągowi liczb naturalnych $(s_1, \ldots, s_N)$ o własnościach 2.7 i 2.8, odpowiada trasa o własnościach 2.4, 2.5, 2.6, o ile długość konturu $(\ast)$ nie przekracza dd. I na odwrót, każdej trasie o własnościach 2.4, 2.5, 2.6 odpowiada ciąg liczb naturalnych $(s_1, \ldots, s_N)$, spełniający 2.7 i 2.8, dla którego kontur $(\ast)$ ma długość nie większą niż dd. Dlatego w dalszych rozważaniach trasę z bazy i-tej (nazwano ją krótko trasą) oznacza się przez

$(\ast\ast)$ $\mathcal{H}'(i) = (s_1, s_2, \ldots, s_N)$,

przy czym liczby $s_1, s_2, \ldots, s_N$ spełniają warunki 2.7 i 2.8 a długość konturu $(\ast)$ nie przekracza dd.

W metodzie cechowania, którą podano poniżej, trasy wyznaczono w postaci $(\ast\ast)$.

Zadanie trasy w postaci (xx) jest bardziej ekonomiczne numerycznie, gdyż zmniejsza liczbę elementów, za pomocą których pamięta się w maszynie cyfrowej daną trasę. Przedstawienie trasy w powyższy sposób jest jednoznaczne, jeśli założy się, że są zadane najkrótsze drogi między każdą parą różnych wierzchołków w sieci $\langle I, U, c\rangle$. Tablicę taką można łatwo wyznaczyć, wykorzystując np. metody podane w [13] . Zakłada się dalej, że taka tablica jest wcześniej obliczona, gdyż w metodzie cechowania będzie się z niej wielokrotnie korzystać, a każdorazowe obliczanie ich wtedy, gdy zachodzi potrzeba, zwiększyłoby niepotrzebnie czas obliczeń.

Metoda cechowania wyznaczania tras z bazy $i \in B$

Trasę środka transportowego z bazy i-tej zadano jak powyżej, tzn. przez podanie ciągu numerów zleceń, jakie dany środek transportowy kolejno realizuje. Podana niżej metoda wyznacza wszystkie trasy z bazy i-tej o własnościach 2.4, 2.5, 2.6. Dla przedstawienia metody wprowadza się pewne dalsze oznaczenia i pojęcia. Niech KZ będzie zbiorem różnych od i wierzchołków sieci $\langle I, U, c\rangle$, w których kończy się co najmniej jedno zlecenie tzn.

$$KZ = \left\{ j \in I : j \neq i \quad \bigvee_{z_s \in Z} \quad z_s = (k_s, l_s) \wedge l_s = j \right\}.$$

Przez cechę wierzchołka $j \in KZ \cup \{i\}$, stanowiącą pewną charakterystykę trasy przechodzącej przez dany wierzchołek, rozumie się następujący układ:

(xxx) $\quad (a(j),\ d(a(j)),\ (s_1, \ldots, s_{N(a(j))})),$

gdzie: $a(j)$ - numer kolejnej cechy wierzchołka j (przez wierzchołek może przechodzić kilka tras, może więc mieć on więcej niż jedną cechę),

$d(a(j))$ - długość drogi od wierzchołka i do wierzchołka j na odpowiadającej (xxx) trasie,

$N(a(j))$ - liczba zleceń realizowanych na odpowiadającej (xxx) trasie,

$(s_1,\ldots,s_{N(a(j))})$ - ciąg numerów zleceń, jakie są realizowane na odpowiadającej (xxx) trasie, kolejno na drodze od wierzchołka i do wierzchołka j (ciąg ten spełnia 2.7, 2.8).

Cechę tę będzie się wykorzystywać do cechowania, tzn. nadawania cech innym wierzchołkom zbioru $\{i\} \cup KZ$ (w tym, być może, rozpatrywanemu wierzchołkowi j). Operacji tej, nazywanej przedłużeniem cechy, dokonuje się w niżej opisany sposób:

niech

$$ZZ(a(j)) = \{s_1\} \cup \{s_2\} \cup \ldots \cup \{s_{N(a(j))}\}$$

i

$$S(a(j)) = \{s \notin ZZ(a(j)) : z_s = (k_s, l_s) \in Z .$$

Zbiór $ZZ(a(j))$ jest zbiorem numerów zleceń zrealizowanych na drodze od bazy i do wierzchołka j przez trasę scharakteryzowaną cechą $a(j)$-tą wierzchołka j-tego (xxx), natomiast $S(a(j))$ jest zbiorem numerów zleceń dotąd przez tę trasę jeszcze nie zrealizowanych.

Dla każdego $s \in S(a(j))$ rozważa się następujące przypadki:

(A) $(k_s = j) \wedge (l_s = i)$.

Wierzchołek i otrzyma kolejną, oznaczoną numerem $a(i)$, cechę:

$(a(i),\ d',\ (s_1,\ldots,s_{N(a(j))},\ s))$,

gdzie: $d' = d(a(j)) + c(\mu(j,i))$.

(B) $(k_s = j) \wedge (l_s \neq i)$.

Jeśli zachodzi relacja $d' \leq dd$, gdzie

$d' = d(a(j)) + c(\mu(k_s, l_s)) + c(\mu(l_s, i))$, to wierzchołek i otrzyma kolejną, oznaczoną numerem $a(i)$, cechę:

$(a(i), d', (s_1, \ldots, s_{N(a(j))}))$,

natomiast wierzchołek l_s otrzyma kolejną, oznaczoną numerem $a(l_s)$ cechę:

$(a(l_s), d(a(j)) + c(\mu(k_s, l_s)), (s_1, \ldots, s_{N(a(j))}, s))$.

(C) $(k_s \neq j) \quad (l_s \neq i)$.

Jeśli zachodzi relacja $d' \leq dd$, gdzie

$d' = d(a(j)) + c(\mu(j, k_s)) + c(\mu(k_s, l_s)) + c(\mu(l_s, i))$,

to wierzchołek i otrzyma kolejną $a(i)$-tą cechę:

$(a(i), d', (s_1, \ldots, s_{N(a(j))}))$,

natomiast wierzchołek l_s kolejną, $a(l_s)$-tą cechę:

$(a(l_s), d(a(j)) + c(\mu(j, k_s)) + c(\mu(k_s, l_s)), (s_1, \ldots, s_{N(a(j))}, s))$.

(D) $(k_s \neq j) \wedge (l_s = i)$.

W przypadku, gdy $d' \leq dd$, gdzie

$d' = d(a(j)) + c(\mu(j, k_s)) + c(\mu(k_s, i))$

wierzchołek i otrzyma kolejną $a(i)$-tą cechę:

$(a(i), d', (s_1, \ldots, s_{N(a(j))}, s))$.

W każdym z czterech powyższych przypadków - do zleceń już zrealizowanych - na drodze od wierzchołka i do wierzchołka j trasy scharakteryzowanej cechą (***) chcemy dołączyć nowe zlecenie z_s. Zlecenie takie dołącza się tylko wtedy, gdy łączna długość następujących dróg:

od wierzchołka i do wierzchołka j trasy scharakteryzowanej (***),

najkrótszej drogi od wierzchołka j do wierzchołka, będącego końcem realizacji zlecenia z_s, poprzez wierzchołek będący początkiem realizacji tego zlecenia,

od wierzchołka, będącego końcem realizacji zlecenia z_s, do bazy w przypadku, gdy wierzchołek ten jest różny od rozważanej bazy i,

nie przekroczy wielkości dd.

W przypadku (A) rozważa się sytuację, gdy zlecenie, które chcemy jeszcze zrealizować, zaczyna się w wierzchołku posiadającym rozpatrywaną cechę (***) i kończy w bazie i. W tym przypadku warunek na długość trasy jest spełniony automatycznie, gdyż dowolny wierzchołek ze zbioru KZ otrzymuje cechę tylko wtedy, gdy można z tego wierzchołka dojechać do bazy i, po drodze trasy scharakteryzowanej pewną cechą. W przypadku (B) staramy się zrealizować zlecenie, które zaczyna się podobnie jak w przypadku (A) lecz kończy w wierzchołku nie będącym bazą i. Sytuację, gdy do zbioru zleceń już zrealizowanych na drodze od bazy i do wierzchołka j trasy scharakteryzowanej cechą (***), staramy się dołączyć nowe zlecenie, które nie zaczyna się w wierzchołku posiadającym rozważaną cechę (***), a kończy w wierzchołku będącym lub nie będącym bazą i, rozpatrujemy odpowiednio w przypadkach (D) i (C).

Oprócz wyżej opisanej operacji przedłużania cechy do formalnego opisu metody cechowania używa się jeszcze pojęć: ocechowanego i nieocechowanego wierzchołka. Wierzchołek należący do zbioru KZ i posiadający co najmniej jedną cechę nazywa się dalej wierzchołkiem ocechowanym, natomiast pozostałe wierzchołkami nieocechowanymi. W trakcie metody cechowania nieocechowane wierzchołki mogą się stawać ocechowanymi i na odwrót.

Metodę cechowania wyznaczania tras z bazy $i \in B$ podaje poniższy schemat.

Krok 0. Wierzchołkowi i nadać cechę $(0, 0, (\phi))$, gdzie ϕ - zbiór pusty. Przyjąć za $S(0)$ zbiór $\{1, \ldots, lz\}$ i dokonać przedłużenia cechy $(0, 0, (\phi))$. Wtedy pewne wierzchołki zbioru $\{i\} \cup KZ$ zostaną ocechowane.

Niech m - numer jakiegoś wierzchołka ocechowanego ze zbioru KZ. Przyjąć $n := m$ i przejść do kroku 1.

Krpk 1. Dokonać przedłużenia wszystkich cech wierzchołka n. Usunąć wszystkie cechy z tego wierzchołka - wierzchołek n stanie się wtedy nieocechowanym. Przejść do kroku 2.

Krok 2. Niech m - numer jakiegoś ocechowanego wierzchołka ze zbioru KZ. Podstawić n: = m i przejść do kroku 1. Jeśli w zbiorze KZ nie ma już ocechowanych wierzchołków, to koniec postępowania. Zbiór cech wierzchołka i, z wyjątkiem cechy $(0,0,(\phi))$, stanowi wyznaczany zbiór tras z bazy i-tej (cechy zawierają dodatkowo informację: numer trasy - pierwszy element cechy, długość trasy - drugi element cechy).

<u>Algorytm wyznaczania optymalnego planu przydziału zadań w systemie transportowym</u>

Wyjściowym zbiorem tras (dalej zastosowano umowę, że przez trasę rozumie się trasę o dopuszczalnej długości, spełniającą warunki 2.4, 2.5, 2.6), który dalej oznacza się przez M, będzie zbiór otrzymany w niżej opisany sposób. Stosuje się metodę cechowania dla każdej bazy $i \in B$. Otrzymuje się w ten sposób pewien zbiór tras, w którym wyszukuje się podzbiory tras realizujących te same zlecenia (być może w różnej kolejności). Do zbioru M z każdego takiego podzbioru włącza się tylko jedną trasę o najmniejszej długości. Jest oczywiste, że ograniczenie się, przy konstrukcji planu optymalnego, do zbioru tras M nie zmniejsza ogólności rozważań, a zmniejsza istotnie liczbę elementów zbiorów, które się dalej rozważa. Ma to duże znaczenie, jeśli idzie o wielkość pamięci maszyny cyfrowej, przeznaczonej na zapamiętanie tych zbiorów oraz na czas obliczeń.

Rozważa się zatem zbiór $R' \subset R$ wszystkich planów dopuszczalnych utworzonych tylko z tras zbioru M. W kolejnych krokach algorytmu zbiór ten, w określony sposób, rozbija się na podzbiory. Dla każdego z takich podzbiorów podaje się oszacowanie z dołu łącznej długości tras tworzących plany dopuszczalne należące do danego podzbioru. Oszacowanie to służy do wskazania podzbioru (podzbiór ten należy dalej rozbijać), który ma największe szanse, że znajdzie się w nim plan optymalny, i pozwala stwierdzić, czy już otrzymany plan jest planem optymalnym. Algorytm wyznaczania planu optymalnego podaje poniższy schemat.

Krok 0. Dla zbioru R' oszacowanie z dołu, które oznacza się przez $\varphi(R')$, definiujemy następująco:

$$\varphi(R') = \sum_{j=1}^{lz} c(\mu(k_j, l_j)),$$

$\varphi(R')$ jest łączną długością dróg, na których realizuje się jakieś zlecenie ze zbioru Z.

Dla każdego planu $P' \in R'$ zachodzi

$c(P') \geqslant \varphi(R')$.

Istotnie z założenia 2.2 oraz z własności 2.5 wynika, że w każdym dopuszczalnym planie te drogi muszą wystąpić, zatem powyższa nierówność jest spełniona.

Zbiór R' rozbija się na r(1) podzbiorów $R_1^1, \ldots, R_{r(1)}^1$, takich, że

$$R' = R_1^1 \cup, \ldots, \cup R_{r(1)}^1.$$

Każdy z tych zbiorów określa się następująco: do zbioru R_j^1 należą te plany dopuszczalne, w których są realizowane trasy należące do podzbioru M_j^1 zbioru M wszystkich tras. W początkowym kroku zbiory M_j^1 są jednoelementowe - M_j^1 zawiera j-tą trasę z ponumerowanego zbioru tras, realizujących ustalone zlecenie z_p (na przykład zlecenie z_1 jak w dalej podanym programie na maszynę cyfrową).

Niech zbiór M_j^1 ma postać:

$$M_j^1 = \left\{ \mathcal{H}'_{l^1_{t(1,j)}} (i^1_{t(1,j)}) \right\}.$$

Elementy rozbicia $R_1^1, \ldots, R_j^1, \ldots, R_{r(1)}^1$ są zatem podane przez wyżej określone zbiory: $M_1^1, \ldots, M_j^1, \ldots, M_{r(1)}^1$, co dalej oznacza się jak poniżej:

$$R_j^1 \longleftrightarrow M_j^1 = \left\{ \mathcal{H}'_{l^1_{t(1,j)}} (i^1_{t(1,j)}) \right\}$$

dla $j = 1, \ldots, r(1)$.

Krok k. Dane jest rozbicie $R_1^k, \ldots, R_{r(k)}^k$, gdzie:

$$R_j^k \longleftrightarrow M_j^k = \left\{ \mathcal{H}'_{l_1^k} (i_1^k), \ldots, \mathcal{H}'_{l^k_{t(k,j)}} (i^k_{t(k,j)}) \right\},$$

przy czym $t(k,j)$ jest ilością tras należących do M_j^k.

Zbiór M_j^k ma następujące własności:

$$\mathcal{H}'_{l^k_{k'}} (i^k_{k'}) \in M,$$

dla $k' = 1, \ldots, t(k,j)$,

$$H(\mathcal{H}'_{l^k_{k'}} (i^k_{k'})) \cap H(\mathcal{H}'_{l^k_{k''}} (i^k_{k''})) = \phi$$

dla $k' \neq k''$, $k', k'' = 1,2, \ldots, t(k,j)$.

Funkcję φ dla zbioru R_j^k, będącą oszacowaniem z dołu długości tras planów dopuszczalnych należących do tego zbioru, określa się następująco:

$$\varphi(R_j^k) = \varphi(R') + c^0(M_j^k),$$

gdzie $c^0(M_j^k)$ jest łączną długością "pustych" przejazdów w trasach $\mathcal{H}'_{l_1^k} (i_1^k), \ldots, \mathcal{H}'_{l^k_{t(k,j)}} (i^k_{t(k,j)})$.

Dla dowolnego planu dopuszczalnego $P' \in R_j^k$ zachodzi:

$c(P') \geqslant (R_j^k)$.

Wynika to stąd, że plan ten - z określenia planu dopuszczalnego - musi realizować wszystkie zlecenia, czyli łączna długość dróg, na których realizuje się te zlecenia, jest równa $\varphi(R')$. Natomiast z określenia zbioru R_j^k plan P' powinien zawierać, między innymi trasy należące do zbioru M_j^k, czyli łączna długość pustych przejazdów w trasach należących do tego planu jest co najmniej równa $c^O(M_j^k)$. Powyższe dwa fakty gwarantują spełnienie wyżej rozpatrywanej nierówności.

Do dalszego rozbicia wybiera się zbiór R_v^k spełniający następującą relację:

$$R_v^k = \min_{1 \leq j \leq r(k)} \varphi(R_j^k).$$

Jeśli zbiór M_v^k zawiera trasy realizujące wszystkie zlecenia tj. $H(M_j^k) = Z$, to plan złożony z tras należących do zbioru M_v^k wyznacza plan optymalny P^{*} i $c(P^{*}) = \varphi(R_v^k)$. W przeciwnym wypadku zbiór R_v^k rozbija się na podzbiory, gdyż są wtedy największe szanse, że wśród planów należących do tego zbioru znajdzie się plan optymalny. W tym celu wyznacza się zlecenie $z_s \in Z$ o najmniejszym wskaźniku s takie, że

$z_s \in H(M_v^k)$, gdzie: $H(M_v^k) = H(\mathcal{H}_{l_1^k}(i_1^k)) \cup \ldots \cup H(\mathcal{H}_{l_{t(k,v)}^k}(i_{t(k,v)}^k))$

oraz wyszukuje się w zbiorze M wszystkie trasy

$\mathcal{H}_{k_1}(i_1), \ldots, \mathcal{H}_{k_n}(i_n)$ zawierające zlecenie z_s i takie, że

$$H(M_v^k) \cap H(\mathcal{H}_{k_j}(i_j)) = \phi \qquad \text{dla } j = 1, 2, \ldots, n.$$

Zbiór R_v^k rozbija się na podzbiory następująco:

$$R_v^k = R_{v,1}^k \cup \ldots \cup R_{v,j}^k \cup \ldots \cup R_{v,n}^k,$$

gdzie: $R_{v,j}^k \longleftrightarrow M_{v,j}^k = \{M_v^k \cup \mathcal{H}_{k_j}(i_j)\}$ dla $j = 1, \ldots, n.$

W następnym kroku rozważa się rozbicie

$$R' = R_1^{k+1} \cup \dots \cup R_{r(k+1)}^{k+1} =$$

$$= R_1^k \cup \dots \cup R_{v-1}^k \cup R_{v,1}^k \cup \dots \cup R_{v,n}^k \cup \dots \cup R_{r(k)}^k.$$

Wyżej opisany algorytm daje po skończonej liczbie kroków rozwiązanie optymalne. Wynika to z faktu, że na każdym kroku suma elementów rozbicia jest zbiorem R' (dzięki temu nie straci się żadnego planu dopuszczalnego utworzonego z tras zbioru M) oraz z tego, że elementy rozbicia R_j^k są generowane przez zbiory tras M_j^k tak konstruowane, aby trasy należące do nich spełniały warunek 1.2 (warunek 2.2 jest spełniony z założenia, gdyż wybiera się tylko trasy o dopuszczalnej długości) .

Algorytm powyższy w realizacji na maszynę cyfrową wymaga dynamicznej rezerwacji pamięci, co jest jego wadą, gdyż z góry nie można oszacować koniecznej dla rozwiązania danego problemu wielkości pamięci maszyny cyfrowej. Aby tego uniknąć, można otrzymywać rozwiązania przybliżone przez używanie powyższego algorytmu z pewną modyfikacją kroku zerowego. Modyfikacja polega na tym, że zamiast postępowania opisanego w algorytmie w kroku 0, należy rozpocząć proces rozbijania nie od zbioru R', ale od zbioru R_1^1 zdefiniowanego następująco:

$$R_1^1 \longleftrightarrow M_1^1 = \left\{ \mathcal{H}_{l_1^1} \; (i_1^1) \right\} \quad ,$$

gdzie: $\mathcal{H}_{l_1^1} (i_1^1) \in M$ spełnia następującą relację:

$$c(\mathcal{H}_{l_1^1} \; (i_1^1)) = \min_{\mathcal{H}_\alpha (i_\beta) \in M} c(\mathcal{H}_\alpha(i_\beta))$$

Wybór takiego zbioru R_1^1 jest uzasadniony tym, że trasa o najmniejszej długości "pustych" przejazdów ma największą szansę wystąpienia w rozwiązaniu optymalnym, dzięki czemu ograniczenie się do zbioru R_1^1 zamiast zbioru R' daje najczęściej rozwiązanie optymalne lub stosunkowo dobre jego przybliżenie.

Rozdział III

NUMERYCZNA REALIZACJA ZAGADNIENIA PRZYDZIAŁU ZADAŃ W SYSTEMIE TRANSPORTOWYM

Podamy obecnie numeryczną realizację zagadnienia przydziału zadań w systemie transportowym. Metodę opisaną w rozdziale II realizuje się w dwóch programach. W pierwszym z nich, oznaczonym symbolem MC, realizuje się konstrukcję zbioru M oraz dodatkowo wykonuje się pewne operacje, dotyczące odpowiedniego umieszczenia informacji o trasach w pamięci zewnętrznej (bębnowej) maszyny cyfrowej. Informację tę wykorzysta się w drugim programie, który oznaczono symbolem MOP. W programie MOP realizuje się metodę "ograniczeń i podziału" konstrukcji planu optymalnego z tras zbioru M.

Realizacja metody z rozdziału II dwoma programami jest podyktowana tym, że maszyna cyfrowa ODRA 1204, na której realizuje się tę metodę, ma stosunkowo małą pamięć operacyjną natomiast znacznie większą pamięć bębnową. Korzystanie jednak z pamięci bębnowej wydłuża znacznie czas działania programu.

Zarówno program MC jak i program MOP wymagają dla dużych zagadnień rezerwacji obszaru pamięci maszyny cyfrowej znacznie przekraczającego wielkość pamięci operacyjnej rozważanej maszyny cyfrowej. Dlatego korzysta się z pamięci bębnowej, z tym, że umieści się w niej tylko informacje wykorzystywane niezbyt często. Informacje wykorzystywane bardzo często w programach MC i MOP umieszcza się w pamięci operacyjnej.

Realizacja rozważanej metody dwoma programami pozwala zwiększyć wielkość dostępnej pamięci operacyjnej kosztem nieznacznego zwiększenia czasu związanego z wydrukiem wyników z programu MC i ponownym ich wczytaniem przez program MOP.

Dla maszyn cyfrowych z dużą pamięcią operacyjną można przez nieznaczną modyfikację programów MC i MOP otrzymać jeden program, działający znacznie szybciej i realizujący zagadnienia większych rozmiarów. W tym celu są konieczne pewne dodatkowe informacje o organizacji tych programów, które podano poniżej w opisach programów MC i MOP.

Opis programu MC

1. Dane

Na płaszczyźnie (mapie) zawierającej sieć $\langle I, U, c \rangle$ rozważanego systemu transportowego wprowadza się układ współrzędnych prostokątnych o odpowiedniej jednostce długości. Każdy wierzchołek $i \in I$ ma wtedy przyporządkowaną parę liczb (x_i, y_i).

Przyjmując, że zbiór zleceń Z, które ma zrealizować rozważany system, zdefiniowano jak na str. 23 . Rozważamy podzbiór $I' \subset I$ złożony z baz oraz wierzchołków będących początkami lub końcami zleceń tzn.

$I' = B \cup \{k_1, l_1, k_2, l_2, \ldots, k_{lz}, l_{lz}\}$.

Wierzchołki zbioru I' numeruje się tak, by pierwsze lb wierzchołków było bazami.

Program MC kolejno wczytuje następujące dane:

lw - liczba wierzchołków w zbiorze I',

lb - liczba baz (elementów zbioru $B \subset I$),

lz - liczba zleceń,

dd - długość maksymalna trasy,

X [1],..., X [lw] - współrzędna x wierzchołków zbioru I',

$\cancel{X}$ y [1],..., $\cancel{X}$ y [lw] - współrzędna y wierzchołków zbioru I',

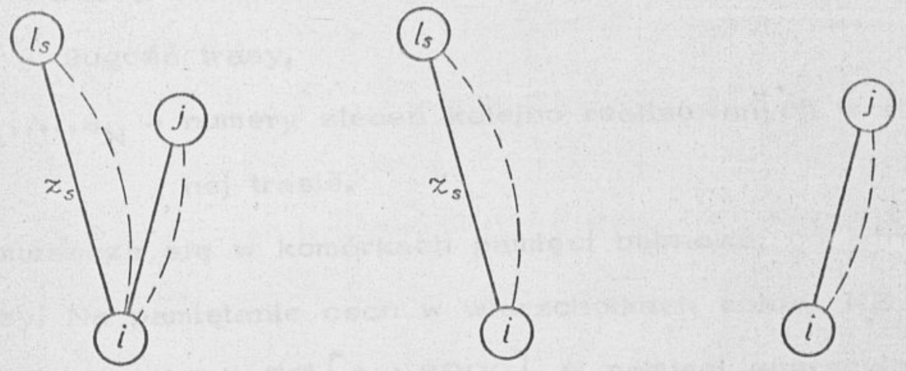
$Z[1], Z[2], \ldots, Z[2 \cdot lz]$ - numery (według numeracji wierzchołków zbioru I') wierzchołków: początkowego i końcowego kolejnych zleceń tzn.

$$z_j = (Z[2 \cdot j - 1], Z[2 \cdot j]), \; j = 1, 2, \ldots lz.$$

2. Numeryczne modyfikacje metody cechowania zastosowane w programie MC

W metodzie cechowania opisanej w rozdziale II dokonuje się następującej modyfikacji:

Jeśli w przypadku D zachodzi dodatkowo relacja $k_s = i$, to nie rozpatruje się tego przypadku (opuszcza się go). Taka modyfikacja zmniejsza liczbę tras w zbiorze M, gdyż zamiast konstruować trasy wszystkich trzech typów a, b, c (schematycznie przedstawione na rys. 3), jak w uprzednio opisanej metodzie, konstruuje się tylko trasy typu b i c (rys. 3). Nie wpłynie to na rozwiązanie optymalne, gdyż jeśli trasa typu a (rys. 3) miałaby wystąpić w rozwiązaniu optymalnym, to po modyfikacji nie wystąpi ona w zbiorze M (i w rozwiązaniu optymalnym), ale w zbiorze M wystąpią dwie trasy typu b i c (rys. 3) o takiej samej łącznej długości jak trasa typu a.



Rys. 3

W programie MC do obliczania najkrótszych dróg między węzłami w sieci $\langle I, U, c \rangle$ używa się procedury ODLEG, gdyż do celów testowania programu jest to wygodniejsze. W procedurze tej korzysta się z metryki liczenia odległości między wierzchołkami sieci komunikacyjnej Polski dla dróg I i II klasy podanej przez Florkiewicza w [6]. Metryka ta pozwala zastępować rzeczywistą długość najkrótszej drogi między wierzchołkami i oraz j sieci komunikacyjnej wyżej podanej poniższymi wyrażeniami:

$$\left| x_i - x_j \right| + \left| y_i - y_j \right|,$$

gdzie x_i, y_i - współrzędne wierzchołka i,

x_j, y_j - współrzędne wierzchołka j.

Procedura ODLEG oblicza w powyższy sposób długość drogi $\mu(i,j)$ i specjalnie nie wydłuża czasu obliczeń. Obliczanie każdorazowo, jeśli zachodzi potrzeba, długości drogi eliminuje konieczność pamiętania tablicy najkrótszych dróg i tym samym zwiększa wielkość, dostępnej dla obliczeń, pamięci operacyjnej.

3. Organizacja programu MC

W programie MC wyznacza się najpierw trasy, metodą opisaną w rozdziale II, w następującej postaci:

$$(le, d, m1, s_1, \dots, s_N),$$

gdzie: le - liczba elementów (komórek) w ciągu,

d - długość trasy,

$s_1, \dots, s_N$ - numery zleceń kolejno realizowanych w rozważanej trasie.

Trasy umieszcza się w komórkach pamięci bębnowej od adresu 0 począwszy. Na pamiętanie cech w wierzchołkach zbioru $KZ \cup \{i\}$ przeznacza się tablicę $TC[1 : 8000]$ w pamięci operacyjnej, przy czym na cechy jednego wierzchołka ze zbioru KZ przeznacza się liczbę sc = entier (7000/pb) komórek, gdzie pb jest aktualną liczbą elementów zbioru KZ. W przypadku, gdy liczba komórek zajętych przez cechy jakiegoś wierzchołka ze zbioru KZ przekroczy sc, program drukuje tekst "Mało komórek na cechy w wierzchołku" oraz numer tego wierzchołka i przerywa obliczenia (stop na monitorze). Cechy nadawane - w trakcie cechowania - wierzchołkowi $i \in B$ pamiętamy w tablicy TC od numeru 1 i dopiero po przedłużeniu danego wierzchołka są przesyłane do pamięci bębnowej. W przypadku, gdy przedłużenie jakiegoś wierzchołka daje zbiór cech, które zajmują więcej niż 1000 komórek program również przerywa obliczenia. Drukuje wtedy tekst "Zbyt dużo cech z" oraz numer odpowiedniego wierzchołka.

Program sygnalizuje sytuację, gdy jakieś zlecenie nie będzie zrealizowane w żadnej trasie. Drukuje się wtedy tekst "Zlecenia", numer tego zlecenia oraz znowu tekst "nie można zrealizować" i program przerywa obliczenia, gdyż wiadomo, że rozpatrywane zagadnienie nie będzie miało rozwiązania.

Eliminację zbędnych tras przeprowadza się w pamięci operacyjnej. Dlatego w przypadku, gdy liczba komórek zajętych przez trasy wyznaczone w programie przekroczy 8000, program drukuje tekst "Za mała tablica TC" i przerywa obliczenia. Trasy należące do zbioru M, otrzymanego po eliminacji, są pamiętane, jak już opisano, z tym, że d jest wtedy długością "pustych" przejazdów dla danej trasy. Adresy początkowych komórek kolejnych tras należących do zbioru M pamięta się w komórkach pamięci bębnowych od adresu 25001, natomiast liczbę tras w zbiorze M w komórce o adresie 25000.

W komórkach od adresu bębnowego 15000 pamięta się ciąg numerów tras k zawierających kolejno zlecenia $z_1, \dots, z_{lz}$. Jeśli liczba elementów tego ciągu przekroczy 10000, program drukuje tekst "Za duży zbiór numerów tras" i przerywa obliczenia.

4. Wyniki

Program MC drukuje (z pulpitu zadać druk na perforator) następujące wielkości:

dp - łączna długość dróg, na których realizuje się jakieś zlecenie tzn.

$$dp = \sum_{j=1}^{lz} c(\mu(k_j, l_j)) \quad (\text{czyli } \varphi(R')),$$

lz - liczba zleceń,

m - ilość elementów ciągu numerów tras, zawierającego kolejno trasy realizujące zlecenie $z_1, z_2, \ldots, z_{lz}$,

$Z[1]$, $Z[2]$ - numer pierwszego i ostatniego elementu segmentu ciągu numerów tras, gdzie są pamiętane numery tras realizujących zlecenie z_1,

$Z[3]$, $Z[4]$ - numer pierwszego i ostatniego elementu segmentu ciągu numerów tras, gdzie są pamiętane numery tras realizujących zlecenie z_2,

$Z[2 \cdot lz - 1]$, $Z[2 \cdot lz]$ - numer pierwszego i ostatniego elementu segmentu ciągu numerów tras, gdzie są pamiętane numery tras realizujących zlecenie z_{lz},

$TC[1], \ldots, TC[m]$ - ciąg numerów tras, w którym segment j-ty, zaczynający się od numeru $Z[2 \cdot j - 1]$ kończący się na numerze $Z[2 \cdot j]$, zawiera numery tras (zbiór tras M ma naturalną numerację) realizujących zlecenie z_j (dla $j = 1, 2, \ldots, lz$), dla $j = 1, 2, \ldots, lz - 1$ zachodzi $Z[2 \cdot j] + 1 = Z[2 \cdot (j + 1) - 1]$.

Opis programu MOP

1. Dane

Danymi do programu MOP jest taśma z wynikami otrzymana z programu MC.

2. Sposób uruchomienia programu

Program MOP podano w dwóch wersjach. Pierwsza z nich pozwala otrzymać rozwiązanie optymalne zagadnienia przydziału zadań w systemie transportowym. Natomiast wersja druga realizuje metodę otrzymywania rozwiązania przybliżonego do rozważanego zagadnienia (opisaną w końcu rozdziału II).

Uruchomienie programu MOP przebiega następująco: wczytuje się taśmę z programem MOP. Czytanie tej taśmy zatrzymuje się na znaku zapytania, po czym, jeśli chcemy otrzymać rozwiązanie optymalne rozważanego zagadnienia, to wczytuje się z osobnej taśmy fragment programu oznaczonym w komentarzu symbolem MOP-A. Natomiast, gdy ze względu na czas obliczeń lub wymiar zagadnienia, należy się zadowolić rozważaniem przybliżonym, wczytuje się fragment programu, z osobnej taśmy, oznaczony w komentarzu symbolem MOP-B. W obu wypadkach wczytuje się następnie dalszą część programu MOP.

3. Organizacja programu MOP

W programie MOP, dla każdego kroku k, informacje o podzbiorach R_j^k dla j = 1,2,..., r(k) pamiętamy w kolejnych komórkach pamięci bębnowej, począwszy od adresu 25000 + f, gdzie f jest liczbą tras w zbiorze M, w sposób opisany poniżej.

Dla zbioru R_j^k w komórkach, począwszy od adresu bębnowego zapamiętanego w j-tym elemencie tablicy ABG pamięci operacyjnej, pamięta się następujący ciąg:

3.0. $(lt, t_1, \ldots, t_{lt}, lzg, z_1, \ldots, z_{lzg})$,

gdzie: lt jest liczbą tras w zbiorze M_j^k (str. 34 , rozdział II),

ciąg $t_1, \ldots, t_{lt}$ jest ciągiem numerów tras należących do M_j^k, natomiast lzg jest liczbą zleceń realizowanych w trasach należących do M_j^k a $z_1, \ldots, z_{lzg}$ jest ciągiem numerów zleceń realizowanych w trasach zbioru M_j^k.

Wartość $\varphi(R_j^k)$ pamięta się, podobnie jak i adres początkowej komórki 3.0, w tablicy DG pamięci operacyjnej, gdyż są to najczęściej używane informacje w programie. Obecnie podamy ograniczenia, wynikające z organizacji programu na maszynę cyfrową ODRA 1204. Ograniczenia te muszą być uwzględnione przy ewentualnej praktycznej realizacji podanej w rozdziale II metody na inną maszynę cyfrową.

3.1. Liczba elementów ciągu 3.0 dla każdego R_j^k $(j = 1, \ldots, r(k))$ nie może przekroczyć 50, gdyż w trakcie obliczeń ciąg 3.0 jest pamiętany w tablicy roboczej TRT[1 : 50] .

3.2. Żadna trasa należąca do zbioru M nie może zajmować (w postaci jak na str. 40) więcej niż 50 komórek. Wynika to stąd, że w programie trasę pamięta się w tablicy roboczej TR [1 : 50] .

3.3. Dane do programu (postać opisana na str. 42) MOP muszą spełniać następujący warunek:

$Z[2 \cdot j] - Z[2 \cdot j - 1] \leq 250$ dla $j = 1, 2, \ldots, lz$.

Ograniczenie to wynika stąd, że numery tras zawierających dane zlecenie, są pamiętane w tablicy TRB[1 : 250] .

3.4. r(k) (ilość podzbiorów R_j^k) nie może, w żadnym kroku, przekroczyć 1500. Wynika to z zadeklarowanych wielkości tablic ABG [1 : 1500] i DG [1 : 1500] .

Przekroczenie któregokolwiek z tych ograniczeń będzie sygnalizowane automatycznie przez translator maszyny cyfrowej ODRA

1204 jako błąd [12] typu SUBSCRIPT lub DRUM PARAMETER (tylko w punkcie 3.2).

4. Wyniki

Program MOP, w przypadku gdy zagadnienie nie ma rozwiązania dopuszczalnego, drukuje tekst "NIE MA ROZWIĄZANIA" i zatrzymuje się na stop. W przypadku, gdy zostanie znalezione rozwiązanie, program drukuje teksty: "ROZWIĄZANIE ZAGADNIENIA" oraz "L.p. trasa" i numerując, wpisuje kolejne trasy należące do planu optymalnego, gdy stosujemy wersję MOP-A lub do planu suboptymalnego, gdy stosujemy wersję MOP-B. Po wydrukowaniu wszystkich tras, należących do planu optymalnego lub suboptymalnego, program MOP drukuje tekst "Łączna długość tras w planie" oraz drukuje łączną długość tych tras (wartość zmiennej min).

Realizacja przykładów i efektywność metody

Sprawdzenia działania programów MC oraz MOP (w obu wersjach) dokonano, przeliczając kilka przykładów. Wyniki obliczeń, czasy działania (łącznie z czasem wczytania danych i drukowania wyników) programów oraz wielkości charakteryzujące przykłady podano w tabeli I.

Wielkości lw, lb, lz, dd charakteryzujące przykłady są określone na str. 24,38. Przykład 4 bardziej dokładnie przedstawiono na rys. 4. Pary liczb w nawiasach okrągłych na tymże rysunku są współrzędnymi prostokątnymi odpowiednich wierzchołków. Każda trasa, należąca do planu optymalnego jest oznaczona odmiennym układem kresek lub kresek i kropek, przy czym zlecenia realizowane na danej trasie są opisane i oznaczone pogrubionymi znakami. W przykładzie tym metoda przybliżona w czasie wielokrotnie krótszym daje

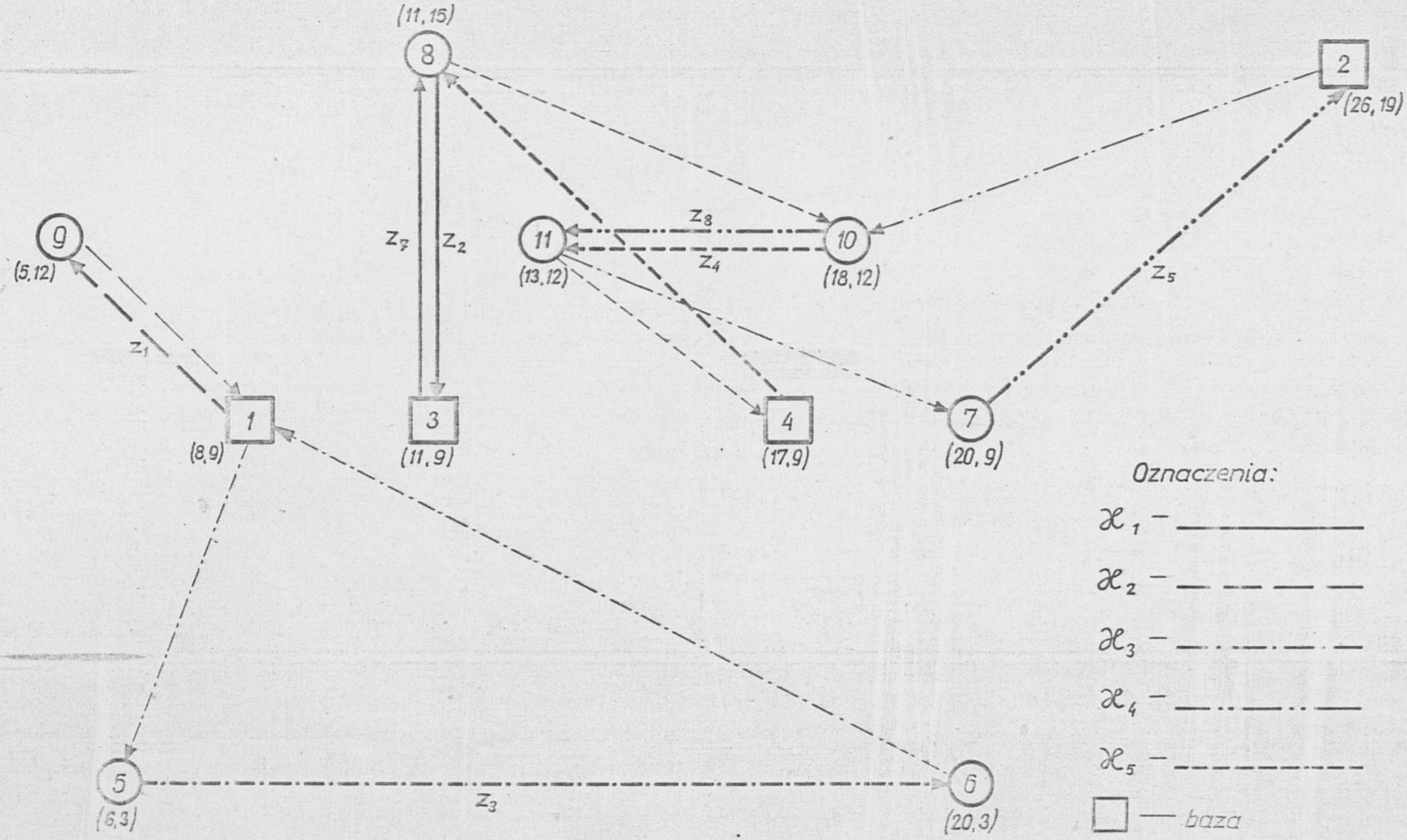
Tabela I

Lp.	lw	lb	lz	dd	Czas działania programu (min sek)			Wartość rozwiązania	
					MC	MOP		optymalnego MOP-A	przybliżonego MOP-B
						MOP-A	MOP-B		
1	8	4	5	60	7"	4"	2"	214	214
2	8	4	5	90	13"	7"	3"	176	190
3	11	4	5	40	19"	12"	5"	102	102
4	11	4	8	40	46"	5'50"	22"	136	136
5	11	4	8	50	2'18"	14'02"	1'21"	128	128
6	11	4	8	60	2'35"	-	2'31"	-	120
7	11	6	10	80	5'39"	8'45"	6'30"	188	200

również rozwiązanie optymalne. Analiza wyników wykazuje, że czas obliczeń zależy od wielkości lw, lb, lz oraz dd.

Zwiększenie czasu obliczeń wraz ze wzrostem wartości lb jest spowodowane między innymi zwielokrotnieniem wykonywania pewnego cyklu obliczeń w programie MC i tym samym bardziej wpływa na czas działania tego programu niż program MOP. Podobnie ma się rzecz z wpływem wzrostu wielkości lw na czas obliczeń.

Jednak najbardziej znaczny wpływ na czasy działania obu programów mają wielkości lz oraz dd. Analizując metodę cechowania, widać, że wzrost liczby zleceń (lz) winien powodować wydłużenie procesu cechowania, gdyż najczęściej trzeba konstruować większą ilość tras, co wpływa na czas obliczeń programu MC. Natomiast w metodzie ograniczeń i podziału przy większej liczbie zleceń - oprócz bardziej licznego zbioru M, z którego konstruowane są trasy, dochodzi większa liczba operacji przy konstrukcji zbiorów R_j^k, co w konsekwencji może znacznie wydłużać czas działania programu MOP.


(11, 15)
8
2
(26, 19)
9
(5,12)
Z7
Z2
11
(13,12)
Z8
Z4
10
(18,12)
Z5
Z1
1
(8,9)
3
(11,9)
4
(17,9)
7
(20,9)
Oznaczenia:
ϰ1 –
ϰ2 –
ϰ3 –
ϰ4 –
ϰ5 –
— baza
5
(6,3)
Z3
6
(20,3)

Rys. 4

Wydłużenie maksymalnej dopuszczalnej długości trasy, czyli wzrost dd, może nawet znacznie wydłużać czas działania programu MC, gdyż może spowodować gwałtowny wzrost liczby tras w zbiorze M i tym samym wydłużyć czas działania programu MOP, spowodowany koniecznością rozpatrywania stosunkowo dużej liczby zbiorów R_j^k. W przykładzie 6 ilość zbiorów R_j^k, jakie należało zapamiętać dla otrzymania rozwiązania optymalnego, przekroczyła zarezerwowaną na ten cel wielkość pamięci operacyjnej maszyny cyfrowej. Wpływ wielkości lz oraz dd na czas obliczeń ma jednak charakter bardziej złożony. Porównując przykłady 5 i 7, widzimy, że wzrost lz oraz dd nie musi powodować wzrostu czasu obliczeń, szczególnie w programie MOP. Fakt ten może zależeć, oprócz własności numerycznych samego programu, od pewnej własności zbioru zleceń, którą rozważa się poniżej, oraz od struktury topologicznej zleceń, rozumianej jako rozmieszczenie zleceń na sieci.

Dla scharakteryzowania pierwszego aspektu parametr, oznaczony slz, zdefiniuje się następująco:

$$slz = \frac{dd}{\frac{1}{lz} \sum_{j=1}^{lz} c(\mu) k_j, l_j))}$$

Wielkość $\frac{1}{lz} \sum_{j=1}^{lz} c(\mu(k_j, l_j))$, będąca mianownikiem ułamka w powyższym wzorze, można interpretować jako "średnią" długość zlecenia, natomiast slz jako praktycznie maksymalną liczbę zleceń, które środek transportowy na trasie o długości dd może zrealizować.

Analiza programów MC oraz MOP wykazuje, że - w przykładach, dla których wartość slz rośnie - czasy obliczeń mogą również wzrastać. Stosunkowo duża wartość slz może powodować w metodzie cechowania konieczność wyznaczania bardzo wielu możliwych tras dla każdej z baz. To może powodować nawet gwałtowny wzrost liczby tras w zbiorze M i tym samym zwiększać czasy obliczeń programów MC i MOP. W przykładzie 4, 5, 6 i 7 wartość slz wynosiła odpowiednio 4.85, 6.06, 7.27 i 5.63, co potwierdza przypuszczenia, że czasy obliczeń dla przykładów o większej wartości slz są większe i rośnie wielkość zajętej pamięci operacyjnej (przykład 7). Można zatem sądzić, że efektywność metody będzie maleć i to gwałtownie dla zagadnień o większych wartościach slz, z tym, że efektywność tej metody będzie zależeć od typu maszyny cyfrowej, na której będzie ona realizowana. Jeśli jednak wielkość slz nie jest zbyt duża i struktura topologiczna zleceń względem baz jest taka, że środek transportowy nie może realizować więcej niż slz zleceń na trasie z danej bazy, to metoda może być efektywną nawet dla zagadnień mających znaczenie praktyczne.

W wielu systemach transportowych np. typu PKS praktycznie liczba zleceń realizowanych na trasie przez środek transportowy jest rzędu kilku i zbliżona jest do wartości slz z przykładów 4, 7. Na taki stan wpływają między innymi operacje związane z załadunkiem i wyładunkiem, które mogą być uwzględnione w rozważanym przez nas modelu (funkcja c(i, j) może być określona nie jako długość lecz czas przejazdu łuku (i, j)).

Dla takiego typu systemów transportowych podana w pracy metoda może być efektywną i praktycznie użyteczną w procesie kierowania takimi systemami. Z tym, że należałoby zaprogramować tę metodę na maszynę cyfrową o znacznie większej pamięci operacyjnej (rzędu 128K). Czasy obliczeń byłyby wtedy znacznie krótsze w porównaniu ze stosunkowo dużymi czasami działania programów MC i MOP na maszynie ODRA 1204, spowodowanymi głównie częstym korzystaniem z pamięci bębnowej w tych programach.

Dysponowanie programem o charakterze użytkowym pozwoliłoby, po przeprowadzeniu odpowiedniej liczby eksperymentów, dokładniej ustalić zależność efektywności tego programu od wielkości slz, ewentualnie od innych parametrów charakteryzujących strukturę zleceń. Poza tym pozwoliłoby ustalić dokładniej klasę realnych problemów, które można by rozwiązywać omawianą metodą.

WNIOSKI

Zagadnienie przydziału zadań w systemach transportowych, posiadających wiele baz, ma bardzo istotne znaczenie praktyczne. Istniejące modele tego zagadnienia starają się uwzględniać więcej lub mniej aspektów tego problemu. Modele uwzględniające większość ograniczeń są trudne do rozwiązania i zazwyczaj dają rozwiązania przybliżone. Model rozpatrywany w niniejszej pracy ujmuje następujące istotne ograniczenia:

a) ograniczoność długości (lub czasu przejazdu) trasy dla każdego środka transportowego,
b) konieczność powrotu środka transportowego do bazy, z której wyjechał, po przejechaniu określonej trasy,
c) realizacja wszystkich zadań nałożonych na system transportowy,
d) minimalizacja pustych przejazdów.

Metoda podana w tej pracy pozwala otrzymać rozwiązanie optymalne tego zagadnienia przy użyciu maszyny cyfrowej (program realizujący tę metodę jest podany dla maszyny cyfrowej ODRA 1204). Rozwiązanie optymalne daje ilość środków transportowych z każdej bazy, zaangażowanych w realizację przewozów, oraz określa trasy dla tych środków i zlecenia, które mają być kolejno na tych trasach zrealizowane.

W modelu rozważanym przez nas zakłada się, że pojemność środków transportowych jest w przybliżeniu równa ciężarowi ładunków, jakie należy przewieźć między pewnymi wierzchołkami w sieci. Ograniczenie to nie jest jednak istotne, gdyż jeśli między pewnymi wierzchołkami w sieci należy np. dokonać przewozu ładunku, który wymaga użycia kilku środków transportowych, to zamiast jednego zlecenia między tymi wierzchołkami daje się odpowiednio większą liczbę zleceń. W przypadku, gdy dla realizacji przewozów między pewnymi parami wierzchołków w sieci transportowej należy użyć znacznej liczby środków transportowych, to wprowadzenie zbyt wielu zleceń między pewnymi wierzchołkami w sieci może znacznie pogorszyć efektywność podanej w pracy metody i wtedy bardziej efektywną będzie np. metoda podana w [1]. Jeśli jednak tak nie jest, to metody podane w [1] mogą dawać rozwiązania mało dokładne i wtedy bardziej przydatną może być metoda podana w niniejszej pracy.

Uwzględnienie, w rozważanym przez nas modelu, ograniczonej liczby środków w każdej z baz jest możliwe, natomiast wprowadzenie środków transportowych o różnych typach wymaga bezspornie dalszych badań w celu wykorzystania pewnych własności grafowych tras oraz struktury topologicznej zleceń dla otrzymania rozwiązań efektywnych.

```
begin
 comment METODA CECHOWANIA  M C ;
 integer dd,lb,lk,lw,lz,p,pb,sc;
 integer array TC[1:8000];
 read(lw,lb,lz,dd);
 p:=lz+lz;
 begin
  integer i,j;
  integer array B,X,Y,XP,YP[1:lw],Z[1:p];
  read(X,Y,Z);
  begin
   integer a1,d,dp,e,f,g,h,k,l,m,m1,n,r,s,t,w;
   Boolean A;
   integer procedure ODLEG(a,b);
   value a,b;
   integer a,b;
   begin
    integer c,c1;
    c:=X[a]-X[b];
    c1:=Y[a]-Y[b];
    ODLEG:=abs(c)+abs(c1)
   end opisu funkcji;
   f:=w:=drumplace:=0;
   format('111␣␣');
   for m1:=1 step 1 until lb do
    begin
     pb:=0;
     for i:=2 step 2 until p do
```

```
      begin
       e:=Z[i];
       if e=m1
         then go to SK;
       h:=pb;
       for j:=1 step 1 until h do
        if TC[j]=e
          then go to SK;
       pb:=pb+1;
       TC[pb]:=e;
       B[e]:=pb;
SK:   end i;
     sc:=7000divpb;
     begin
      integer array KZ,LKZ.TLC,TKR[1:pb];
      copy(pb,TC[1],KZ[1]);
      for i:=1 step 1 until pb do
       begin
        TLC[i]:=0;
        TKR[i]:=LKZ[i]:=1000+sc×(i-1)
       end;
      for s:=1 step 1 until lz do
       begin
        i:=s+s;
        l:=Z[i];
        k:=Z[i-1];
        A:=m1=k;
        if A
          then
```

```
  begin
  d:=ODLEG(m1,l);
  dp:=d+d;
  A:=dp>dd;
  if A
    then go to SK1
end
else
begin
  d:=ODLEG(m1,k)+ODLEG(k,l);
  dp:=d+ODLEG(l,m1);
  A:=dp>dd;
  if A
    then go to SK1;
  A:=l=m1;
  if A
    then
    begin
      f:=f+1;
      w:=w+1;
      TC[w]:=4;
      w:=w+1;
      TC[w]:=d;
      w:=w+1;
      TC[w]:=m1;
      w:=w+1;
      TC[w]:=s;
      go to SK1
```

```
            end
          end;
        f:=f+1;
        w:=w+1;
        TC[w]:=4;
        w:=w+1;
        TC[w]:=dp;
        w:=w+1;
        TC[w]:=m1;
        w:=w+1;
        TC[w]:=s;
        h:=B[l];
        TLC[h]:=TLC[h]+1;
        e:=LKZ[h]+1;
        TC[e]:=4;
        TC[e+1]:=d;
        TC[e+2]:=m1;
        TC[e+3]:=s;
        LKZ[h]:=LKZ[h]+4;
SK1:   end s,krok 0 ;
KONT:  for a1:=1 step 1until pb do
        begin
         j:=KZ[a1];
         e:=TKR[a1];
         lk:=0;
         for m:=1 step 1 until TLC[a1] do
          begin
           e:=e+lk+1;
```

```
lk:=TC[e]-1;
d:=TC[e+1];
for s:=1 step 1 until lz do
 begin
  r:=s+s;
  k:=Z[r-1];
  l:=Z[r];
  for r:=3 step 1 until lk do
   begin
    A:=TC[e+r]=s;
    if A
      then go to SK4
   end r;
  A:=k=j;
  if A
    then
    begin
     g:=d+ODLEG(k,l);
     dp:=g+ODLEG(l,m1)
    end
    else
    begin
     A:=k=m1;
     if A
       then go to SK4;
     g:=d+ODLEG(j,k)+ODLEG(k,l);
     dp:=g+ODLEG(l,m1)
    end ;
```

```
A:=dp>dd;
if A
  then go to SK4;
A:=l≠m1;
if A
  then
  begin
   copy(lk-1,TC[e+2],TC[w+3]);
   f:=f+1;
   w:=w+1;
   TC[w]:=lk+2;
   w:=w+1;
   TC[w]:=dp;
   w:=w+lk;
   TC[w]:=s;
   h:=B[l];
   TLC[h]:=TLC[h]+1;
   t:=LKZ[h]+1;
   copy(lk-1,TC[e+2],TC[t+2]);
   TC[t]:=lk+2;
   TC[t+1]:=g;
   t:=t+lk+1;
   TC[t]:=s;
   LKZ[h]:=t;
   A:=t>TKR[h]+sc;
   if A
     then
```

```
          begin
           print('?Malo komorek na cechy we
                              wierzcholku',h);
           line(1);
           stop
          end A
        end
        else
        begin
         copy(lk-1,TC[e+2],TC[w+3]);
         f:=f+1;
         w:=w+1;
         TC[w]:=lk+2;
         w:=w+1;
         TC[w]:=g;
         w:=w+lk;
         TC[w]:=s
        end l=m1;
SK4:  end s
    end m;
    A:=w>1000;
    if A
      then
      begin
       print('?Zbyt duzo cech z wierzcholka',KZ[a1]);
       line(1);
       stop
```

```
        end;
      A:=w≠0;
      if A
        then
        begin
          todrum(w,TC[1]);
          w:=0
        end;
      TLC[a1]:=0;
      LKZ[a1]:=TKR[a1]
    end a1;
   j:=0;
   for i:=1 step 1 until pb do
    j:=j+TLC[i];
   A:=j≠0;
   if A
     then go to KONT
 end
end m1;
comment Eliminacja tras-konstrukcja zbioru M;
g:=drumplace;
A:=g>8000;
if A
  then
  begin
   print('Za mala tablica TC');
   line(1);
   stop
  end;
```

```
        drumplace:=0;
        fromdrum(g,TC[1]);
        e:=t:=1;
        a1:=0;
E1:     t:=t+a1;
        A:=t-1=g;
        if A
          then go to EN;
        a1:=TC[t];
        e:=t+a1;
        A:=e-1=g;
        if A
          then go to EN;
        copy(a1,TC[t],XP[1]);
 E21:   m1:=TC[e];
        copy(m1,TC[e],YP[1]);
        A:=a1=m1;
        if ¬A
          then go to E2;
        for k:=4 step 1 until a1 do
         begin
          d:=XP[k];
          A:=false;
          for s:=4 step 1 until a1 do
           A:=(d=YP[s])∨A;
          if ¬A
            then go to E2
         end k;
```

```
        A:=XP[2]≤YP[2];
        if A
          then
          begin
            A:=e+m1-1=g;
            f:=f-1;
            g:=g-m1;
            if A
              then go to E1;
            copy(g-e+1,TC[e+m1],TC[e]);
            go to E21
          end;
        f:=f-1;
        A:=e+m1-1=g;
        g:=g-m1;
        if A
          then go to E3;
        copy(g-e+1,TC[e+m1],TC[e]);
        copy(m1,YP[1],XP[1]);
E3:     copy(m1,YP[1],TC[t]);
        if A
          then go to E1
          else go to E21;
E2:     e:=e+m1;
        A:=e-1=g;
        if A
          then go to E1
          else go to E21;
```

```
EN:     t:=1;
        comment Podstawienie pod d ''pustych'' przejazdow
        w trasach zbioru M ,zapamietanie,od adresu bebnowego
        25000 poczawszy,odpowiednio liczby tras-f orazadresow
        poczatkowych komorek tras zbioru M;
        m:=drumplace:=0;
        for i:=1 step 1 until f do
         begin
          e:=TC[t]-1;
          d:=TC[t+1];
          dp:=0;
          r:=e+t;
          for j:=t+3 step 1 until r do
           begin
            s:=TC[j];
            s:=s+s;
            k:=Z[s-1];
            l:=Z[s];
            dp:=dp+ODLEG(k,l)
           end j;
          TC[t+1]:=d-dp;
          todrum(e+1,TC[t]);
          TC[i+1]:=m;
          m:=drumplace;
          t:=r+1;
         end i;
        dp:=0;
        for i:=1 step 1 until lz do
```

```
  begin
   s:=i+i;
   k:=Z[s];
   l:=Z[s-1];
   dp:=dp+ODLEG(k,l)
  end i;
 format('?123456␣');
 print(dp);
 TC[1]:=f;
 drumplace:=25000;
 todrum(f+1,TC[1]);
 comment PRZYGOTOWANIE DANYCH DO PROGRAMU   M O P ;
 drumplace:=0;
 fromdrum(g,TC[1]);
 drumplace:=15000;
 m:=1;
 for i:=1 step 1 until lz do
  begin
   t:=1;
   s:=i+i;
   Z[s-1]:=m;
   lb:=0;
   for j:=1 step 1 until f do
    begin
     e:=TC[t];
     r:=t+e-1;
     for k:=t+3 step 1 until r do
      begin
```

```
          A:=i=TC[k];
          if A
            then
            begin
              lb:=lb+1;
              B[lb]:=j;
              A:=lb=lw;
              if A
                then
                begin
                  todrum(lw,B[1]);
                  lb:=0
                end lb=lw;
              m:=m+1;
              go to endj;
            end A
          end k;
endj:    t:=t+e;
        end j;
       A:=lb>0;
       if  A
        then todrum(lb,B[1]);
       Z[s]:=m-1;
       A:=Z[s]<Z[s-1];
       if A
         then
         begin
          print('?Zlecenia',i,'nie mozna zrealizowac');
```

```
        line(1);
        stop
      end
    end i;
  m:=m-1;
  A:=m>10000;
  if A
    then
    begin
      print('?Za duzy zbior numerow tras');
      line(1);
      stop
    end;
  drumplace:=15000;
  fromdrum(m,TC[1]);
  format('?1234␣␣1234␣␣');
  print(lz,m,Z);
  format('?1111␣␣1234␣␣1234␣␣1234␣␣1111␣␣1111␣␣
                    1111␣␣1111␣␣1111␣␣1111␣␣');
  for i:=1 step 1 until m do
   print(TC[i]);
  space(16)
 end
end
end M C;
```

```
begin
 comment M O P ;
 integer a,aog,b,b1,f,ff,i,j,k,l,le,lg,lgp,lz,lzg,lt,m,r,s,z;
 Boolean B;
 integer array ABG,DG[1:1500],TR,TRP,TRT[1:50],TRB[1:250];
 read(b1,lz,ff);
 drumplace:=25000;
 fromdrum(1,TR[1]);
 f:=TR[1];
 begin
  integer min;
  integer array NT[1:ff],PT,KT[1:lz],ABT[1:f];
  for i:=1 step 1 until lz do
   begin
    PT[i]:=ininteger;
    KT[i]:=ininteger
   end;
  read(NT);
  fromdrum(f,ABT[1]);
 ?
  B:=lg=0;
  if B
    then
    begin
     print('? NIE MA ROZWIAZANIA ');
     line(1);
     stop
    end;
```

```
EP:    min:=1000000;
       for i:=1 step 1 until lg do
        begin
         a:=DG[i];
         B:=a<min;
         if B
           then
           begin
            min:=a;
            k:=i
           end B
        end i;
       drumplace:=ABG[k];
       fromdrum(50,TR[1]);
       lt:=TR[1];
       a:=2+lt;
       lzg:=TR[a];
       for i:=1 step 1 until lz do
        begin
         for j:=1 step 1 until lzg do
          begin
           B:=i=TR[a+j];
           if B
             then go to E
          end;
         s:=i;
         go to E1;
E:     end i;
```

```
     go to D;
E1:  a:=PT[s];
     b:=KT[s];
     m:=0;
     z:=lt+1;
     for j:=a step 1 until b do
      begin
       r:=NT[j];
       for i:=2 step 1 until z do
        begin
         B:=r=TR[i];
         if B
          then go to E2
        end;
       m:=m+1;
       TRB[m]:=r;
E2:   end j;
     B:=m=0;
     if B
       then
       begin
E4:     r:=lg-k;
        copy(r,ABG[k+1],ABG[k]);
        copy(r,DG[k+1],DG[k]);
        lg:=lg-1;
        lgp:=lg;
        go to EP
       end B;
```

```
a:=3+lt;
b:=a+lzg-1;
for i:=1 step 1 until m do
 begin
  r:=TRB[i];
  drumplace:=ABT[r];
  fromdrum(50,TRP[1]);
  le:=TRP[1];
  for j:=a step 1 until b do
   begin
    z:=TR[j];
    for l:=4 step 1 until le do
     begin
      B=z=TRP[l];
      if B
        then go to E3
     end l
   end j;
  copy(lt,TR[2],TRT[2]);
  TRT[1]:=lt+1;
  TRT[2+lt]:=r;
  copy(lzg,TR[a],TRT[4+lt]);
  j:=le-3;
  copy(j,TRP[4],TRT[b+2]);
  z:=lzg+j;
  TRT[a]:=z;
  lg:=lg+1;
  DG[lg]:=DG[k]+TRP[2];
```

```
      z:=z+a;
      ABG[lg]:=drumplace:=aog;
      todrum(z,TRT[1]);
      aog:=drumplace;
E3:   end i;
    B:=lg-lgp=0;
    if B
      then go to E4;
    z:=lg-k;
    copy(z,ABG[k+1],ABG[k]);
    copy(z,DG[k+1],DG[k]);
    lgp:=lg;
    go to EP;
 D: print('?ROZWIAZANIE ZAGADNIENIA ??');
    lt:=lt+1;
    print('?L.p.  trasa');
    format('123␣␣');
    for j:=2 step 1 until lt do
     begin
      print('?',j-1);
      drumplace:=ABT[TR[j]];
      fromdrum(50,TRP[1]);
      le:=TRP[1];
      for i:=1 step 1 until le do
       begin
        B:=i=10;
        if B
          then line(1);
```

```
        print(TRP[i])
      end
    end j;
  format('12345␣␣');
  print('?Laczna dlugosc tras w planie =',min)
end
end M O P ;
```

```
comment M O P - A ;
a:=PT[1];
b:=KT[1];
lg:=0;
aog:=drumplace;
for i:= a step 1 until b do
 begin
  z:=NT[i];
  drumplace:=ABT[z];
  fromdrum(50,TR[1]);
  le:=TR[1];
  lg:=lg+1;
  DG[lg]:=TR[2]+b1;
  TRT[1]:=1;
  TRT[2]:=z;
  copy(le-3,TR[4],TRT[4]);
  TRT[3]:=le-3;
  ABG[lg]:=drumplace:=aog;
  todrum(le,TRT[1]);
  aog:=drumplace
 end M O P - A ;
 ?
```

```
comment M O P -B ;
min:=100000;
aog:=drumplace;
for i:=1 step 1 until f do
 begin
  drumplace:=ABT[i];
  fromdrum(2,TR[1]);
  z:=TR[2];
  le:=TR[1];
  B=z<min;
  if B
    then
    begin
     min:=z;
     lzg:=le;
     k:=i
    end
    else
    begin
     B=z=min;
     if B
       then
       begin
        B=le > lzg;
        if B
          then
          begin
           lzg:=le;
```

```
            k:=i
          end
        end
      end
    end M O P - B;
    ?
```

LITERATURA

[1] Buga J., Nykowski I., Zagadnienia transportowe w programowaniu liniowym, PWN, Warszawa 1972.

[2] Clarke G., Wright J.W., Scheduling of vehicles from a central depot to a number of delivery points. Operations Research 1964, vol. 2, s. 568-581.

[3] Christofides N., Eilon E., An Algorithm for the Vehicle - Lispatching Problem, Operational Research Quarterly 1969, vol. 20, nr 3, s. 309-318.

[4] Croes G.A.; A method for solving travelling salesman problems. Operations Research 1958, vol. 6, s. 791-812.

[5] Dantzig G.B., Ramser J.H., The truck dispatching problem. Management Science 1959, vol. 6, nr 1, s. 80-91.

[6] Florkiewicz A., Modele ekonometryczne w optymalizacji decyzji transportowych (praca doktorska), Wyd. Pol. Wrocł., Wrocław, 1974.

[7] Gaskall T.J., Bases for Vehicle Fleet Scheduling. Operational Research Quarterly 1967, vol. 18, nr 3, s. 281-293.

[8] Gillett B.E., Miller L.R.: A Heuristic Algorithm for the Vehicle - Dispatch Problem. Operations Research 1974, vol. 22, nr 2, s. 340-349.

[9] Jermolew J., Mielnik C., Ekstriemalnyje zadaczi na grafach, Naukowa Dumka, Kijów 1968.

[10] Jerzykiewicz K., Szczepkowicz J., Algol 1204, PWN, Warszawa 1974.

[11] Jurkiewicz A., Zagadnienie transportowe. Sprawozdanie Zakładu Obliczeń Numerycznych i Katedry Metod Numerycznych Uniwersytetu Warszawskiego, Z. 10, Z. 15, 1968.

[12] Korbut A.A., Finkelsztejn J.J., Programowanie dyskretne, PWN, Warszawa 1974.

[13] Kucharczyk J., Sysło M., Algorytmy optymalizacji w języku Algol-60, PWN, Warszawa 1975.

[14] Lasdon L.S., Decomposition of a ship rauting problem. Decomposition of large-scale problems. N.H.P.C. 1973.

[15] Lin S., Kernigham B.W., An effective heuristic algorithm for the traveling - salesman problem. Operations Research 1973, vol. 21, s. 498-516.

[16] Piasecki S., Optymalizacja systemów przewozowych, WKŁ Warszawa 1973.

[17] Praca zbiorowa: Badania operacyjne w nowoczesnym zarządzaniu, PWE, Warszawa 1974.

[18] Zionts R., Allocation of transportation units to alternative trips: a column generation scheme with out-of-Kilter subproblem. Operations Research 1968, vol. 16, nr 1, s. 52-63.