STUDIA INFORMATICA 2009
Volume 30 Number 1 (82)

Jacek WIDUCH
Politechnika Śląska, Instytut Informatyki

ROZWIĄZANIE PROBLEMU WYZNACZANIA POŁĄCZEŃ W SIECIACH KOMUNIKACYJNYCH ZA POMOCĄ ZMODYFIKOWANEGO ALGORYTMU WYZNACZANIA K NAJKRÓTSZYCH ŚCIEŻEK

Streszczenie. Problem wyznaczania połączeń w sieciach komunikacyjnych jest przykładem zadania optymalizacji wielokryterialnej, którego rozwiązaniem jest zbiór rozwiązań niezdominowanych. Wyznaczanie połączeń polega na rozwiązaniu dwukryterialnego problemu wyznaczania najkrótszej ścieżki w grafie ważonym. W pracy przedstawiono algorytm umożliwiający wyznaczenie wszystkich połączeń należących do zbioru rozwiązań niezdominowanych. Podstawą do opracowania algorytmu był algorytm wyznaczania *K* najkrótszych ścieżek. Problem wyznaczania *K* najkrótszych ścieżek został omówiony w pracy oraz przedstawiono algorytm umożliwiający jego rozwiązanie.

Słowa kluczowe: problem transportowy, optymalizacja wielokryterialna, zbiór rozwiązań niezdominowanych, dwukryterialny problem wyznaczania najkrótszej ścieżki, problem wyznaczania *K* najkrótszych ścieżek, drzewo ścieżek

SOLVING THE COMMUNICATION NETWORKS ROUTING PROBLEM USING MODIFIED ALGORITHM FOR FINDING K SHORTEST PATHS

Summary. The communication networks routing problem is an example of multicriteria optimization which the solution is the set of non-dominated solutions. Establishing routes consists in solving the bicriterion shortest path problem. In the paper an algorithm which determines all routes belong to the set of non-dominated solutions is shown. The algorithm is based on the algorithm for finding *K* shortest paths. It addition, the *K* shortest paths problem and an algorithm for solving it are presented.

Keywords: transportation problem, multicriteria optimization, set of non-dominated solutions, bicriterion shortest path problem, the *K* shortest path problem, paths tree

1. Wstęp

Jednym z wielu problemów związanych z problemami transportowymi, dystrybucyjnymi czy też komunikacyjnymi jest dobór odpowiedniej trasy przejazdu. Wpływ na dobór trasy przejazdu mają takie czynniki, jak czas i koszt przejazdu, komfort podróży czy też bezpieczeństwo. Są też inne czynniki, na które nie mamy wpływu. Do takich czynników bez wątpienia należą korki uliczne, roboty drogowe, awaryjność środków transportu. Przy wyborze trasy przejazdu można uwzględnić tylko jeden z czynników, np. czas lub koszt podróży lub większą ich liczbę. Wybór trasy przejazdu ma znaczenie praktyczne, np. aby dotrzeć na miejsce jak najszybciej, ale także znaczenie ekonomiczne – aby ponieść jak najmniejsze koszty dojazdu.

W pracy został przedstawiony algorytm umożliwiający wyznaczenie połączeń w sieci komunikacyjnej przy jednoczesnym uwzględnieniu dwóch kryteriów: czasu i kosztu podróży. Rozwiązanie niniejszego problemu polega na rozwiązaniu dwukryterialnego problemu wyznaczania najkrótszej ścieżki w grafie ważonym [7] i [8]. Podstawą do opracowania algorytmu był algorytm wyznaczania K najkrótszych ścieżek przedstawiony w pracy [9]. Niniejszą pracę można traktować jako kontynuację prac [1] i [2], w których omówiono problem wyznaczania połączeń w sieciach komunikacyjnych oraz przedstawiono algorytmy rozwiązywania tego problemu.

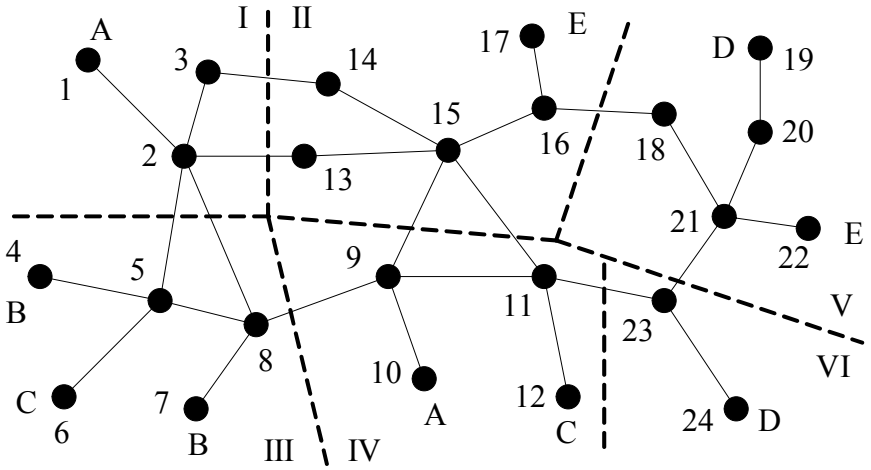
2. Problem wyznaczania połączeń w sieciach komunikacyjnych

Dana jest sieć komunikacyjna opisana za pomocą grafu (rys. 1). Sieć podzielona jest na strefy i kursują w niej pojazdy linii komunikacyjnych, przy czym niektóre z nich oznaczone są jako pospieszne (czas przejazdu pojazdem takiej linii jest mniejszy od czasu przejazdu pojazdem linii zwykłej). Wierzchołki grafu reprezentują przystanki lub stacje końcowe pojazdów komunikacyjnych, natomiast krawędzie trasy przejazdu pojazdów pomiędzy przystankami. Pojazdy każdej linii kursują w obydwu kierunkach, dlatego każdą krawędź grafu należy traktować jako parę łuków przeciwnie skierowanych.

Pojazdy linii komunikacyjnych kursują z określoną częstotliwością według rozkładu jazdy. Trasa przejazdu pojazdów, czasy przejazdu pomiędzy przystankami oraz godziny rozpoczęcia kursu z przystanku początkowego są dane. Zakładamy, że pojazdy kursują zgodnie z rozkładem jazdy, pomijamy także czas oczekiwania pojazdu na przystanku, tzn. godzina wyjazdu z przystanku jest równa godzinie przyjazdu na przystanek.

Dla tak zdefiniowanej sieci komunikacyjnej dany jest przystanek początkowy p_p i przystanek końcowy p_k, pomiędzy którymi ma się odbyć podróż. Dana jest także godzina rozpo-

częcia podróży na przystanku p_p. Zadanie polega na wyznaczeniu linii komunikacyjnych oraz ewentualnych przystanków przesiadek, tak aby czas i koszt przejazdu z przystanku p_p do przystanku p_k były minimalne.



Rys. 1. Przykładowa sieć komunikacyjna
Fig. 1. A sample communication network

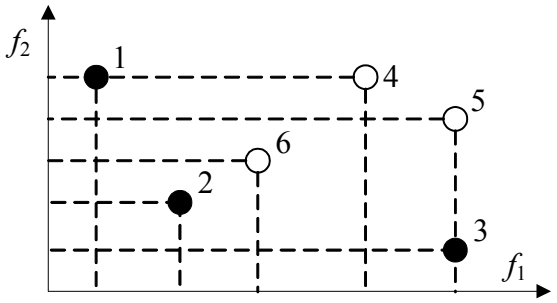
Czas i koszt podróży wynikają z trasy przejazdu. Czas przejazdu jest równy sumie czasów przejazdu pomiędzy przystankami, ewentualnego czasu oczekiwania na przystanku początkowym oraz ewentualnych czasów oczekiwania na przesiadkę. Koszt przejazdu obliczany jest w następujący sposób. Koszt przejazdu pojazdem linii zwykłej bez przesiadki w granicach jednej strefy wynosi c_1 jednostek, w granicach dwóch stref c_2 jednostek, a w granicach trzech lub więcej stref c_3 jednostek. Dla pojazdów linii pospiesznej koszt przejazdu jest C ($C > 1$) razy większy od kosztu przejazdu pojazdem linii zwykłej. W przypadku przesiadki łączny koszt przejazdu jest równy sumie kosztów przejazdu na poszczególnych odcinkach.

3. Podstawy teoretyczne problemu

3.1. Optymalizacja wielokryterialna

Problem będący tematem niniejszej pracy należy do klasy problemów optymalizacji wielokryterialnej, a dokładniej optymalizacji dwukryterialnej [3]. Zadanie optymalizacji wielokryterialnej polega na optymalizacji zadanego zbioru kryteriów jakości (funkcji kryterialnych) f_1, ..., f_n. Kryteria jakości stanowią kryteria porównawcze, względem których odbywa się porównywanie rozwiązań. W przypadku problemu wyznaczania połączeń komunikacyjnych kryteriami jakości są czas i koszt przejazdu. Obydwa kryteria są kryteriami minimalizowanymi, tzn. w procesie optymalizacji dąży się do tego, aby osiągnęły minimalną wartość [3].

Rozwiązaniem zadania optymalizacji wielokryterialnej w ogólnym przypadku jest zbiór rozwiązań, zwany zbiorem rozwiązań niezdominowanych (zbiorem rozwiązań Pareto-optymalnych [4]).



Rys. 2. Przykładowy zbiór rozwiązań niezdominowanych
Fig. 2. A sample set of non-dominated solutions

Niech będzie danych n minimalizowanych funkcji kryterialnych $f_1, ..., f_n$. Mówimy, że rozwiązanie X dominuje rozwiązanie Y, jeżeli spełniona jest zależność:

$$\forall i \in \{1, ..., n\}\ f_i(Y) \le f_i(X) \wedge \exists j \in \{1, ..., n\}\ f_j(Y) < f_j(X).$$

Zbiór rozwiązań niezdominowanych jest zbiorem takich rozwiązań, dla których nie istnieją rozwiązania, które je dominują. Przykładowy zbiór rozwiązań niezdominowanych został przedstawiony na rys. 2. Zbiór wszystkich rozwiązań składa się z 6 rozwiązań, jednak tylko rozwiązania 1, 2 i 3 należą do zbioru rozwiązań niezdominowanych.

3.2. Dwukryterialny problem wyznaczania najkrótszej ścieżki w grafie ważonym

Wyznaczenie połączeń w sieciach komunikacyjnych sprowadza się do rozwiązania dwukryterialnego problemu wyznaczania najkrótszej ścieżki w grafie ważonym. Dodatkowo, żadna ze ścieżek nie może zawierać cykli [5].

Dany jest skierowany graf ważony $G = (V, E)$, gdzie V jest skończonym zbiorem wierzchołków, a E skończonym zbiorem łuków. Z każdym łukiem $(v_i , v_j) \in E$; $v_i, v_j \in V$; związane są dwie dodatnie wagi: $w_1(v_i , v_j)$ i $w_2(v_i , v_j)$. Każda ścieżka:

$$p = <v_0, (v_0, v_1), v_1, (v_1, v_2), ..., (v_{n-1}, v_n), v_n>$$

w grafie G ma dwie wagi, które są równe sumie odpowiednich wag łuków ją tworzących [5]:

$$W_1(p) = \sum_{i=1}^{n} w_1(v_{i-1}, v_i),\ W_2(p) = \sum_{i=1}^{n} w_2(v_{i-1}, v_i).$$

Rozwiązaniem dwukryterialnego problemu wyznaczania najkrótszej ścieżki pomiędzy wierzchołkiem początkowym $v_p \in V$ i końcowym $v_k \in V$, w ogólnym przypadku nie jest jedna ścieżka, ale zbiór ścieżek stanowiący zbiór rozwiązań niezdominowanych. Wagi ścieżek stanowią kryteria, według których ścieżki są porównywane ze sobą.

4. Problem wyznaczania *K* najkrótszych ścieżek

Problem wyznaczania ścieżki w grafie jest jednym z problemów analizy grafów [5 i 10]. W niektórych przypadkach, oprócz wyznaczenia najkrótszej ścieżki, pojawia się także pytanie o dwie najkrótsze ścieżki, trzy najkrótsze ścieżki itd. Ogólnie, konieczne jest wyznaczenie K ($K > 1$) najkrótszych ścieżek. W literaturze omawiane są dwie kategorie problemów wyznaczania K najkrótszych ścieżek. W pierwszej zakłada się, że żadna ścieżka nie może zawierać cyklu, natomiast druga dopuszcza ścieżki zawierające cykle.

Niech będzie dany skierowany graf ważony $G = (V, E)$, w którym zdefiniowana jest dodatnia waga $w(v_i , v_j)$ dla każdego łuku $(v_i , v_j) \in E$. Dane są także wierzchołki początkowy $v_p \in V$ i końcowy $v_k \in V$, pomiędzy którymi należy wyznaczyć ścieżki. Niech P oznacza zbiór wszystkich ścieżek niezawierających cykli z wierzchołka v_p do wierzchołka v_k. Każda ścieżka $p = <v_0, (v_0, v_1), v_1, (v_1, v_2), ..., (v_{n-1}, v_n), v_n>$ ma wagę $W(p)$, będącą sumą wag łuków ją tworzących:

$$W(p) = \sum_{i=1}^{n} w(v_{i-1}, v_i).$$

Problem wyznaczania K najkrótszych ścieżek z wierzchołka v_p do wierzchołka v_k polega na wyznaczeniu zbioru ścieżek $P_K = \{p_1, ..., p_K\} \subseteq P$ spełniającego następujące warunki:

$\forall i \in \{1, ..., K - 1\}\ W(p_i) \le W(p_{i+1})$,

$\forall p \in P \setminus P_K\ \ W(p_K) \le W(p)$,

$\forall i \in \{1, ..., K - 1\}$ ścieżka p_i jest wyznaczana przed wyznaczeniem ścieżki p_{i+1}.

Wyznaczanie K najkrótszych ścieżek z wierzchołka v_p do wierzchołka v_k jest podobne do konstruowania drzewa, którego korzeniem jest wierzchołek v_p, a liśćmi są wierzchołki v_k. Otóż można wykazać, że wszystkie ścieżki $p_i \in P_K$ rozpoczynają się od tej samej podścieżki [5]. Innymi słowy, jeżeli mamy dwie ścieżki:

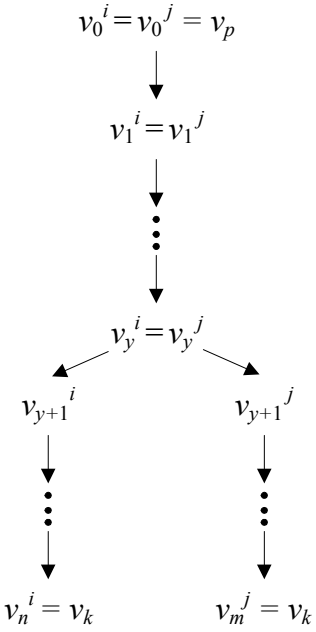
$$p_i = <v_0^{\,i} = v_p, (v_0^{\,i}, v_1^{\,i}), v_1^{\,i}, (v_1^{\,i}, v_2^{\,i}), ..., (v_{n-1}^{\,i}, v_n^{\,i}), v_n^{\,i} = v_k> \in P_K$$

$$p_j = <v_0^{\,j} = v_p, (v_0^{\,j}, v_1^{\,j}), v_1^{\,j}, (v_1^{\,j}, v_2^{\,j}), ..., (v_{m-1}^{\,j}, v_m^{\,j}), v_m^{\,j} = v_k> \in P_K$$

to spełniona jest następująca zależność:

$$\exists\ 0 \le y < min\{n, m\}\ \forall\ x = \{0, ..., y\}\ v_x^{\,i} = v_x^{\,j}. \tag{1}$$

Zależność (1) można zilustrować za pomocą rys. 3, na którym ścieżki p_i oraz p_j zostały przedstawione w postaci drzewa, zwanego drzewem ścieżek. Wierzchołki $v_y^{\,i}$ oraz $v_y^{\,j}$ są ostatnimi wierzchołkami we wspólnej podścieżce. Wierzchołek, będący ostatnim wierzchołkiem na wspólnej podścieżce, nosi nazwę wierzchołka odchylającego (ang. *deviation node*) i jest oznaczany przez $d(p)$.



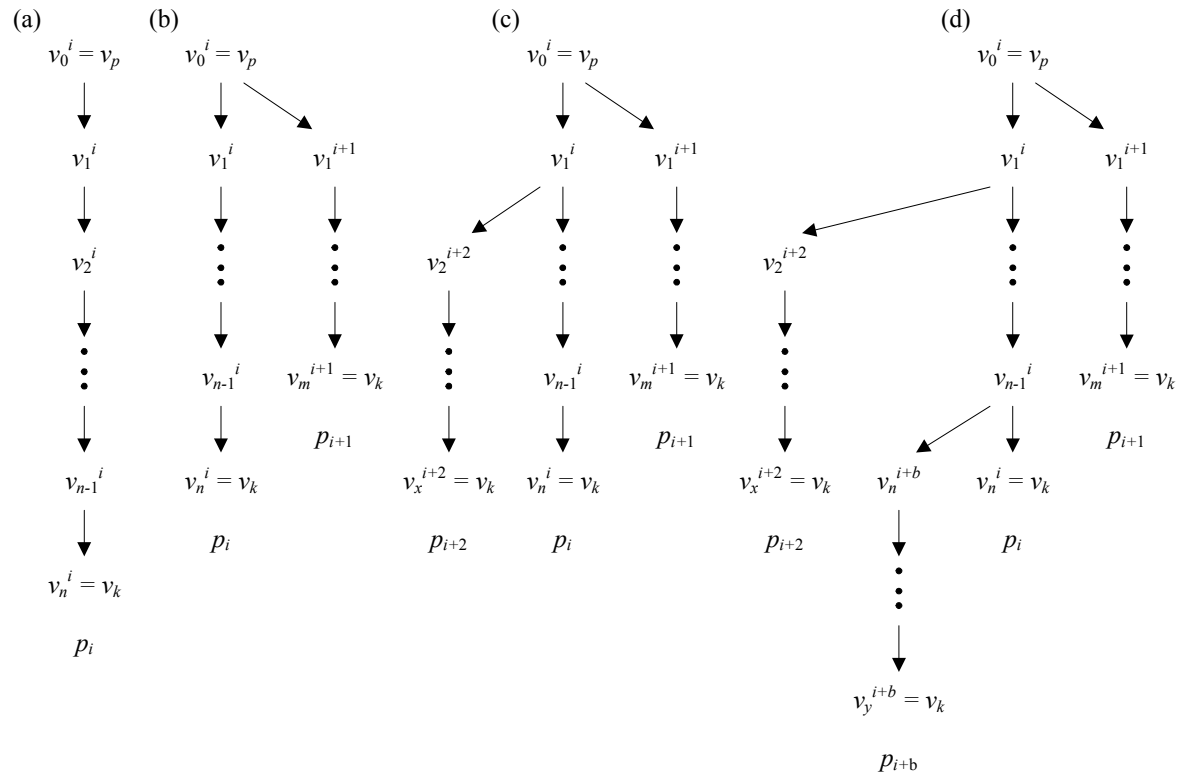
Rys. 3. Przykładowe drzewo ścieżek
Fig. 3. A sample paths tree

Aby skonstruować drzewo ścieżek, konieczne jest wyznaczenie dowolnej ścieżki p_i (rys. 4a) z wierzchołka v_p do v_k. Kolejne ścieżki będą tworzone na podstawie ścieżki p_i w następujący sposób. Z grafu usuwane są kolejne łuki $\{(d(p_i), v_j^i), (v_j^i, v_{j+1}^i), ..., (v_{n-1}^i, v_n^i)\}$ i po usunięciu każdego łuku wyznaczana jest nowa ścieżka prowadząca odpowiednio z wierzchołków $\{d(p_i), v_j^i, ..., v_{n-1}^i\}$ do wierzchołka końcowego v_k. Początkowe kroki konstruowania drzewa ścieżek przedstawione są na rys. 4. Niech będzie dana ścieżka p_i (rys. 4a), której wierzchołkiem odchylającym $d(p_i)$ jest wierzchołek v_0^i. Kolejną ścieżkę p_{i+1} (rys. 4b) wyznaczamy usuwając z grafu łuk (v_0^i, v_1^i) i wyznaczając ścieżkę z wierzchołka v_0^i do v_k. Następnie usuwając z grafu łuk (v_1^i, v_2^i) wyznaczana jest ścieżka prowadząca z v_1^i do v_k, którą jest ścieżka p_{i+2} (rys. 4c). Wierzchołkiem odchylającym ścieżki p_{i+2} jest wierzchołek v_1^i. W identyczny sposób postępujemy ze wszystkimi łukami w ścieżce p_i aż do usunięcia z grafu łuku (v_{n-1}^i, v_n^i), uzyskując kolejne ścieżki. W ten sposób ze ścieżki p_i zostało utworzonych b nowych ścieżek $\{p_{i+1}, ..., p_{i+b}\}$ (rys. 4d). Z nowo utworzonymi ścieżkami postępujemy podobnie i na ich podstawie tworzymy kolejne ścieżki.

Opisany sposób konstruowania drzewa ścieżek został zastosowany do wyznaczenia K najkrótszych ścieżek w grafie ważonym. Jako rozwiązanie startowe, na podstawie którego tworzone są kolejne ścieżki, nie jest przyjmowana dowolna ścieżka, tylko najkrótsza ścieżka z wierzchołka v_p do v_k. Podobna sytuacja jest przy wyznaczaniu kolejnych ścieżek na podstawie aktualnie analizowanej ścieżki p_i – za każdym razem wyznaczana jest najkrótsza ścieżka z aktualnie rozpatrywanego wierzchołka v_j^i do wierzchołka końcowego v_k. Wyznaczanie ścieżek kończymy po zakończeniu analizy ścieżki P_{K-1}. Algorytm, w dalszej części

pracy nazywany algorytmem KSP (od nazwy *K shortest paths*), w którym zastosowano opisaną metodę, został przedstawiony w pracy [9] i można go przedstawić za pomocą pseudokodu:

1: p := najkrótsza ścieżka z wierzchołka v_p do wierzchołka v_k;
2: $d(p) := v_p$;
3: $X := \{p\}$; // wstaw ścieżkę p do zbioru wyznaczonych ścieżek
4: $P_K := \varnothing$; $k := 0$; // zbiór K najkrótszych ścieżek
5: **while** $X \neq \varnothing$ **and** $k < K$ **do**
6: $k := k + 1$;
7: $p_k := <v_0^k, ..., v_j^k>$; // najkrótsza ścieżka ze zbioru X
8: $X := X \setminus \{p_k\}$; $P_K := P_K \cup \{p_k\}$; // usuń ścieżkę p_k ze zbioru X i wstaw do zbioru P_K
9: usuń z grafu wierzchołki znajdujące się w ścieżce $sub_{pk}(v_p, d(p_k))$;
10: usuń z grafu łuki $(d(p_k), v_i) \in p_1, ..., p_{k-1}$;
11: **for all** $v_i^k \in \{d(p_k), \ldots, v_{j-1}^k\}$ **do**
12: usuń z grafu łuk (v_i^k, v_{i+1}^k);
13: q := najkrótsza ścieżka z wierzchołka v_i^k do wierzchołka v_k;
14: $p := sub_{pk}(v_p, d(p_k)) + q$; // nowa ścieżka z wierzchołka v_p do wierzchołka v_k
15: $d(p) := v_i^k$;
16: $X := X \cup \{p\}$; // wstaw ścieżkę p do zbioru wyznaczonych ścieżek
17: usuń z grafu wierzchołek v_i^k;
18: **end for**
19: umieść w grafie usunięte łuki i wierzchołki;
20: **end while**
21: **if** $X \neq \varnothing$ **and** $k = K$ **then**
22: $p_k := <v_0^k, ..., v_j^k>$; // najkrótsza ścieżka ze zbioru X
23: $P_K := P_K \cup \{p_k\}$; // wstaw ścieżkę p_k do zbioru K najkrótszych ścieżek
24: **return** P_K;
25: **else** // nie ma w grafie K ścieżek
26: **return** ERROR; // zwróć informację o błędzie
27: **end if**



Rys. 4. Początkowe kroki konstruowania drzewa ścieżek
Fig. 4. A initial steps of construction paths tree

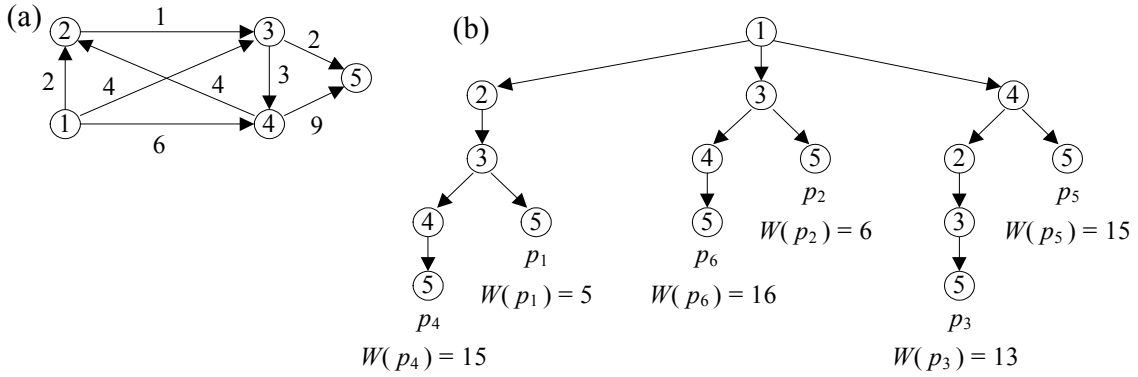
Wyjaśnienia wymaga operacja sub_{pk} zastosowana w krokach 9 i 14 algorytmu KSP. Otóż, niech będzie dana ścieżka: $p_i = <v_0^i, ..., v_{j-1}^i, (v_{j-1}^i, v_j^i), v_j^i, (v_j^i, v_{j+1}^i), v_{j+1}^i, ..., v_n^i>$. Operacja $sub_{pi}(v_0^i, v_j^i)$ polega na wyznaczeniu podścieżki ścieżki p_i zawierającej sekwencję wierzchołków i łuków od v_0^i do wierzchołka v_{j-1}^i, tj.: $sub_{pi}(v_0^i, v_j^i) = <v_0^i, (v_0^i, v_1^i), ..., (v_{j-2}^i, v_{j-1}^i), v_{j-1}^i>$.

W kroku 14 algorytmu zastosowano operację składania dwóch ścieżek, która jest zdefiniowana w następujący sposób:

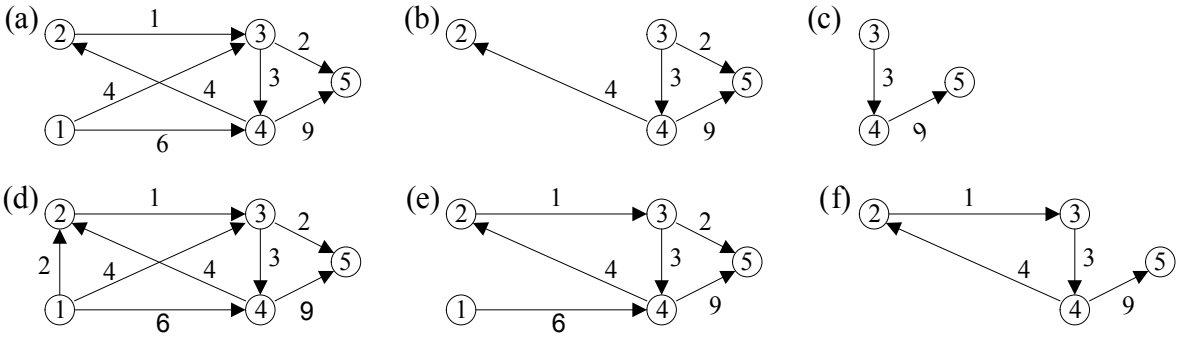
$$p_i = <v_0^i, ..., v_n^i>;\ p_j = <v_0^j, ..., v_m^j>;\ p_i + p_j = <v_0^i, ..., v_n^i, (v_n^i, v_0^j), v_0^j, ..., v_m^j>.$$

Do wyznaczenia najkrótszych ścieżek można zastosować algorytm Dijkstry o złożoności $O(n^2)$ lub zmodyfikowany algorytm Dijkstry o złożoności $O(m + n \cdot log(n))$, gdzie m jest liczbą łuków, a n liczbą wierzchołków [5]. Usunięcie z grafu odpowiednich wierzchołków (kroki 9 i 17 algorytmu) zapewnia uzyskanie ścieżki niezawierającej cyklu. Natomiast usunięcie z grafu odpowiednich łuków (kroki 10 i 12 algorytmu) zabezpiecza nas przed uzyskaniem jako nowego rozwiązania ścieżki, która została wyznaczona w jednej z poprzednich iteracji.

Zasadę działania algorytmu prześledzimy wyznaczając zbiór $K = 3$ najkrótszych ścieżek z wierzchołka 1 do 5 (rys. 5a). Dodatkowo, na rys. 5b przedstawiono drzewo zawierające wszystkie ścieżki pomiędzy wierzchołkami 1 i 5 wraz z ich wagami.



Rys. 5. Przykładowy graf (a) i drzewo ścieżek z wierzchołka 1 do wierzchołka 5 (b)
Fig. 5. A sample graph (a) and paths tree from node 1 do node 5 (b)



Rys. 6. Grafy użyte do wyznaczania K najkrótszych ścieżek
Fig. 6. Graphs used to determine K shortest paths

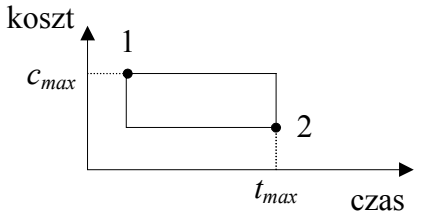
Najkrótszą ścieżką w grafie z rys. 5a jest ścieżka p_1 o wadze $W(p_1) = 5$ i wierzchołku odchylającym $d(p_1) = 1$ (rys. 5b). Ścieżka p_1 stanowi rozwiązanie startowe, na podstawie którego będą uzyskiwane pozostałe ścieżki. Z grafu usuwany jest łuk (1, 2) i w grafie (rys. 6a) wyznaczana jest najkrótsza ścieżka z wierzchołka 1 do 2, którą jest ścieżka p_2 (rys. 5b). Po wyznaczeniu ścieżki p_2 usuwane są z grafu (rys. 6a) wierzchołek 1 i łuk (2, 3). W grafie (rys. 6b) podejmowana jest próba wyznaczenia najkrótszej ścieżki z wierzchołka 2 do 5. Ponieważ taka ścieżka nie istnieje, więc nie została uzyskana żadna nowa ścieżka. Następnie usuwane są z grafu (rys. 6b) wierzchołek 2 i łuk (3, 5) i wyznaczana jest w grafie z rys. 6c najkrótsza ścieżka z wierzchołka 3 do 5, a jest nią ścieżka p_4 o wierzchołku odchylającym $d(p_4) = 3$. W ten sposób została przeanalizowana cała ścieżka p_1 i utworzono na jej podstawie nowe ścieżki p_2 i p_4. Kolejnym krokiem jest umieszczenie w grafie wszystkich usuniętych wierzchołków i łuków (rys. 6d) i próba utworzenia nowych ścieżek na podstawie ścieżki p_2. Postępujemy w podobny sposób jak w przypadku ścieżki p_1, uzyskując na podstawie grafów z rys. 6e i 6f ścieżki p_3 i p_6. Po zakończeniu analizy ścieżki p_2 algorytm kończy swoje działanie, gdyż zostały wyznaczone $K = 3$ najkrótsze ścieżki. Rozwiązaniem problemu jest zbiór ścieżek $P_K = \{p_1, p_2, p_3\}$.

5. Algorytm wyznaczania połączeń w sieci komunikacyjnej

Algorytm wyznaczania połączeń w sieci komunikacyjnej został opracowany przez wprowadzenie odpowiednich modyfikacji w algorytmie KSP opisanym w punkcie 4. Algorytm KSP uwzględnia tylko jedną wagę ścieżki, natomiast ścieżka reprezentująca połączenie komunikacyjne ma dwie wagi – czas i koszt przejazdu. Zmodyfikowany algorytm umożliwia wyznaczenie wszystkich ścieżek należących do zbioru rozwiązań niezdominowanych.

Algorytm KSP umożliwia wyznaczenie jedynie K najkrótszych ścieżek ze względu na czas przejazdu. Jest to niewystarczające, gdyż ścieżki (połączenia komunikacyjne) należące do zbioru rozwiązań niezdominowanych mogą mieć czas większy niż K najkrótszych ścieżek. Z tego powodu działanie algorytmu nie jest przerywane w momencie wyznaczenia K najkrótszych ścieżek, tylko jest kontynuowane aż zostaną przeanalizowane wszystkie ścieżki w zbiorze rozwiązań X.

W zbiorze rozwiązań X nie są przechowywane rozwiązania, co do których jest pewność, że na ich podstawie nie ma możliwości uzyskania rozwiązania niezdominowanego. Aby była możliwość oceny, które rozwiązania można pominąć, konieczne jest wyznaczenie w części inicjalizacyjnej algorytmu, oprócz ścieżki o minimalnym czasie, także ścieżki o minimalnym koszcie przejazdu. Obydwie ścieżki wyznaczają obszar, w którym znajdują się potencjalne rozwiązania niezdominowane. Rozwiązanie 1 (rys. 7) o minimalnym czasie przejazdu określa maksymalny koszt przejazdu c_{max}, a rozwiązanie 2 o minimalnym koszcie przejazdu określa maksymalny czas przejazdu t_{max} rozwiązania niezdominowanego. Do zbioru rozwiązań jako rozwiązanie startowe jest wstawiana tylko ścieżka o minimalnym czasie, gdyż ścieżka o minimalnym koszcie jest wyznaczana wyłącznie w celu określenia obszaru, w którym znajdują się rozwiązania niezdominowane.



Rys. 7. Obszar rozwiązań niezdominowanych
Fig. 7. A space of non-dominated solutions

Stosując algorytm Dijkstry, każdorazowo wyznaczana jest ścieżka o minimalnym czasie przejazdu. Zatem, z każdego rozwiązania (ścieżki) p_i można uzyskać rozwiązanie (ścieżkę) p_{i+1} o czasie większym bądź równym czasowi przejazdu w rozwiązaniu p_i. W grafie może istnieć wiele ścieżek o minimalnym czasie, ale różniących się kosztem przejazdu. Niestety, stosując algorytm Dijkstry nie jest możliwe, aby spośród wszystkich ścieżek o minimalnym czasie przejazdu została wyznaczona ścieżka mająca minimalny koszt [11]. Zatem, koszt

przejazdu w rozwiązaniu p_{i+1}, uzyskanym na podstawie rozwiązania p_i, może być mniejszy od kosztu przejazdu w rozwiązaniu p_i. Wynika z tego, że z rozwiązania zdominowanego można uzyskać rozwiązanie niezdominowane. Zatem, w zbiorze X będą pamiętane także rozwiązania zdominowane, ale co do których nie możemy wykluczyć, że na ich podstawie można utworzyć rozwiązanie niezdominowane. Takie rozwiązania będziemy nazywać rozwiązaniami akceptowalnymi (każde rozwiązanie niezdominowane jest także rozwiązaniem akceptowalnym). Niech rozwiązanie $p_i = <v_0^i = v_p, ..., v_n^i = v_k>$ ma czas przejazdu równy $t(p_i)$ i koszt przejazdu równy $c(p_i)$. Rozwiązanie jest rozwiązaniem akceptowalnym, jeżeli spełnione są warunki: $t(p_i) \le t_{max}$, $c(p_i) + \Delta c \le c_{max}$, gdzie wartość Δc jest określona zależnością:

$$\Delta c = \sum_{j=0}^{n-1} c_{j,j+1}$$

Wartość $c_{j,j+1}$ jest równa:

- 0, jeżeli nie ma przesiadki na przystanku v_j^i,
- kosztowi przejazdu z przystanku v_j^i do v_{j+1}^i w przypadku przesiadki, jeżeli przystanek v_j^i nie jest przystankiem końcowym w trasie linii, którą dojechano do v_j^i,
- połowie kosztu przejazdu z przystanku v_j^i do v_{j+1}^i w przypadku przesiadki, jeżeli przystanek v_j^i jest przystankiem końcowym w trasie linii, którą dojechano do v_j^i i do przystanku v_{j+1}^i jedziemy pojazdem linii pospiesznej,
- połowie kosztu przejazdu z przystanku v_j^i do v_{j+1}^i, jeżeli przystanek v_j^i jest przystankiem początkowym v_p i do przystanku v_{j+1}^i jedziemy pojazdem linii pospiesznej.

W algorytmie KSP analizowane są kolejne wierzchołki v_i^k należące do ścieżki p_k. Z każdego wierzchołka v_i^k jest wyznaczana tylko jedna ścieżka do wierzchołka v_k (krok 14), po czym analizowany jest kolejny wierzchołek należący do ścieżki p_k. Ponieważ wyznaczana jest tylko jedna ścieżka z wierzchołka v_i^k, więc każdorazowo przed rozpoczęciem analizy nowej ścieżki p_k konieczne jest usunięcie z grafu wszystkich łuków wychodzących z wierzchołka $d(p_k)$ i należących do wcześniej przeanalizowanych ścieżek p_1, ..., p_{k-1} (krok 10). Usuwanie łuków jest konieczne, aby ta sama ścieżka nie została wyznaczona wielokrotnie. Aby uniknąć usuwania łuków w zmodyfikowanym algorytmie, nie jest wyznaczana tylko jedna ścieżka prowadząca z wierzchołka v_i^k do v_k, tak jak w algorytmie KSP, tylko wyznaczane są także kolejne ścieżki. Ścieżki są wyznaczane dopóki czas przejazdu tak uzyskanej ścieżki nie przekroczy dopuszczalnej wartości maksymalnej t_{max}, co oznacza, że uzyskane rozwiązanie nie jest rozwiązaniem akceptowalnym.

W trakcie analizy wierzchołka v_i^k nie jest wyznaczana jedna ścieżka prowadząca do wierzchołka v_k, ale wszystkie ścieżki będące rozwiązaniami akceptowalnymi. Z tego powodu każda nowo utworzona ścieżka p_i nie musi być analizowana od wierzchołka odchylającego $d(p_i)$, tylko od kolejnego wierzchołka. Wszystkie rozwiązania akceptowalne, jakie można by

uzyskać analizując wierzchołek $d(p_i)$, zostały już wyznaczone w trakcie analizy ścieżki, na podstawie której została utworzona ścieżka p_i. Wierzchołek znajdujący się w ścieżce za wierzchołkiem $d(p_i)$ będzie oznaczany jako $d_1(p_i)$. Dla rozwiązania startowego, tj. ścieżki o minimalnym czasie przejazdu wyznaczanej w części inicjalizacyjnej algorytmu, jest on równy przystankowi początkowemu v_p.

Uwzględniając wymienione modyfikacje w algorytmie KSP, algorytm MKSP umożliwiający rozwiązanie problemu wyznaczania połączeń w sieci komunikacyjnej można opisać za pomocą pseudokodu:

1: p := połączenie o minimalnym koszcie z wierzchołka v_p do wierzchołka v_k; $t_{max} := t(p)$;
2: p := połączenie o minimalnym czasie z wierzchołka v_p do wierzchołka v_k; $c_{max} := c(p)$;
3: $d_1(p) := v_p$;
4: $X := \{p\}$; $k := 1$; // wstaw połączenie o minimalnym czasie do zbioru rozwiązań
5: **while** $\exists\, p_k \in X$ **do**
6: $p_k := <v_0^k, ..., v_j^k>$; // ścieżka należąca do zbioru X, która będzie analizowana
7: usuń z grafu wierzchołki znajdujące się w ścieżce $sub_{pk}(v_p, d_1(p_k))$;
8: **for all** $v_i^k \in \{d_1(p_k), \ldots, v_{j-1}^k\}$ **do** // przeanalizuj wierzchołki ścieżki p_k
9: usuń z grafu łuk (v_i^k, v_{i+1}^k);
10: **repeat**
11: wyznacz ścieżkę $q := <v_0^q = v_i^k, v_1^q, ..., v_k>$ o minimalnym czasie z v_i^k do v_k;
12: **if** $\exists\, q$ **then**
13: $p := sub_{pk}(v_p, d(p_k)) + q$; // nowa ścieżka z wierzchołka v_p do v_k
14: **if** ścieżka p jest rozwiązaniem akceptowalnym **then**
15: $d_1(p) := v_1^q$; $X := X \cup \{p\}$; // wstaw ścieżkę p do zbioru rozwiązań
16: **end if**
17: usuń z grafu łuk (v_0^q, v_1^q);
18: **end if**
19: **until** (**not** $\exists\, q$) **cor** $(t(p) > t_{max})$;
20: usuń z grafu wierzchołek v_i^k;
21: **end for**
22: umieść w grafie usunięte łuki i wierzchołki;
23: $k := k + 1$; // następna ścieżka do przeanalizowania ze zbioru X
24: **end Chile**
25: usuń ze zbioru X rozwiązania zdominowane;
26: **return** X;

O złożoności czasowej algorytmu decyduje liczba wszystkich połączeń należących do zbioru rozwiązań niezdominowanych. W przypadku pesymistycznym liczba połączeń należących do zbioru rozwiązań niezdominowanych jest równa s^{n-1}, gdzie s jest maksymalnym

stopniem wyjściowym wierzchołka grafu reprezentującego sieć komunikacyjną [7]. Wobec tego złożoność czasowa algorytmu jest równa $O(s^n)$. W badanym problemie maksymalny stopień wyjściowy s jest równy liczbie linii komunikacyjnych.

6. Wyniki badań eksperymentalnych

Badania eksperymentalne algorytmu przedstawionego w punkcie 5 zostały przeprowadzone dla sieci składającej się z 1211 przystanków, która jest podzielona na 26 stref, w której kursują pojazdy 500 linii komunikacyjnych. Maksymalna długość trasy linii komunikacyjnej jest równa 29, a minimalna długość trasy jest równa 6. W badaniach przyjęto, że koszt przejazdu pojazdem linii pospiesznej jest dwa razy większy od kosztu przejazdu pojazdem linii zwykłej. Sieć nie reprezentuje żadnej rzeczywistej sieci komunikacyjnej, została stworzona wyłącznie dla potrzeb testów.

Tabela 1

Wyznaczone rozwiązania

Przystanek		Długość trasy		Liczba rozwiązań	Liczba rozwiązań
początkowy	końcowy	minimalna	maksymalna	niezdominowanych	akceptowalnych
1211	672	21	25	3	6225
703	1095	23	28	6	7954
337	667	27	32	12	73706
484	1074	32	34	2	3028
1155	512	22	25	9	2558
1095	1160	12	16	6	1554
296	353	18	25	6	1883
602	738	20	24	7	10340
574	741	19	28	6	45647
147	1153	28	31	4	3980
90	1	32	39	12	2822

W tabeli 1 została przedstawiona liczba wyznaczonych rozwiązań dla przykładowych par przystanków. Ostatnia kolumna zawiera liczbę wszystkich wyznaczonych rozwiązań akceptowalnych. Natomiast w kolumnach 3 i 4 zostały przedstawione odpowiednio minimalna i maksymalna długość ścieżki należącej do zbioru rozwiązań niezdominowanych. Długość ścieżki jest rozumiana jako liczba wierzchołków należących do ścieżki.

W tabeli 2 przedstawiono porównanie czasu wyznaczania połączeń za pomocą algorytmu MKSP, będącego modyfikacją algorytmu KSP (kolumna 5 tab. 2) oraz algorytmu DFSB przedstawionego w pracy [1], w którym użyto metody powrotów i przeszukiwania grafu w głąb oraz algorytmu BFS przedstawionego w pracy [2], w którym zastosowano metodę przeszukiwania grafu wszerz.

Tabela 2

Czas wyznaczania połączeń

Przystanek		Czas wyznaczania		
początkowy	końcowy	Algorytm DFSB	Algorytm BFS	Algorytm MKSP
1211	672	0:00.00	0:00.10	0:00.31
703	1095	0:00.01	0:00.58	0:00.35
337	667	0:00.00	0:00.31	0:09.25
484	1074	0:00.00	0:00.11	0:00.23
1155	512	0:00.00	0:00.14	0:00.09
1095	1160	0:00.00	0:00.02	0:00.01
296	353	0:00.00	0:00.06	0:00.02
602	738	0:00.00	0:00.36	0:00.36
574	741	0:00.01	0:00.51	0:04.22
147	1153	0:00.04	0:05.35	0:00.19
90	1	0:00.00	0:00.39	0:00.05

7. Podsumowanie

W niniejszej pracy dokonano analizy problemu wyznaczania połączeń w sieciach komunikacyjnych, będącego przykładem zadania optymalizacji wielokryterialnej. Przedstawiony został algorytm umożliwiający wyznaczenie wszystkich połączeń należących do zbioru rozwiązań niezdominowanych. Rozwiązania niezdominowane wyznaczane są przy użyciu algorytmu MKSP, będącego modyfikacją algorytmu wyznaczania *K*-tej najkrótszej ścieżki w grafie ważonym. Problem wyznaczania *K*-tej najkrótszej ścieżki jest rozszerzeniem klasycznego problemu wyznaczania najkrótszej ścieżki, przy czym rozpatrywane są dwa warianty tego problemu: w pierwszym wariancie wyznaczane są wyłącznie ścieżki niezawierające cykli, natomiast w drugim dopuszcza się wyznaczenie ścieżek zawierających cykle. W problemie będącym tematem pracy do zbioru rozwiązań niezdominowanych należą wyłącznie ścieżki, które nie zawierają cykli [7], dlatego zastosowano i zmodyfikowano algorytm rozwiązujący problem pierwszego wariantu.

Rozwiązania w algorytmie MKSP wyznaczane są z użyciem drzewa ścieżek. Podczas konstruowania drzewa ścieżek korzysta się ze ścieżek dotychczas wyznaczonych i na ich podstawie wyznaczane są nowe ścieżki. W algorytmie dokonywana jest bieżąca ocena każdej wyznaczonej ścieżki. Celem oceny jest pominięcie ścieżek, na podstawie których nie ma możliwości utworzenia nowej ścieżki należącej do zbioru rozwiązań niezdominowanych. Podczas oceny użyte zostały ścieżki o minimalnym czasie i minimalnym koszcze przejazdu. Bieżąca ocena zmniejsza obszar rozwiązań, który podlega przeszukaniu podczas wyznaczania rozwiązań.

Algorytm MKSP został zaimplementowany oraz przeprowadzono badania eksperymentalne. Celem badań było porównanie algorytmu z dwoma innymi algorytmami rozwiązującymi badany problem. Czas obliczeń dla zmodyfikowanego algorytmu KSP jest mniejszy od czasu obliczeń dla algorytmu BFS, w którym stosowana jest metoda przeszukiwania grafu wszerz, natomiast jest on większy od czasu obliczeń dla algorytmu DFSB opartego na metodzie powrotów i przeszukiwanie grafu w głąb. Opracowany algorytm MKSP ma wykładniczą złożoność czasową. Z tego powodu stosowanie go do sieci o dużych rozmiarach może być kłopotliwe ze względu na znaczny czas wyznaczania połączeń, który może być nieakceptowany przez użytkownika. Jak wykazały badania, opracowany algorytm jest możliwy do zastosowania dla sieci komunikacyjnych, których liczba przystanków jest rzędu 1000.

BIBLIOGRAFIA

1. Widuch J.: Problem wyznaczania połączeń w sieciach komunikacyjnych. ZN Pol. Śl. Studia Informatica, Vol. 22, No. 4(46), Gliwice 2001, s. 117÷134.
2. Widuch J.: Wyznaczanie połączeń w sieciach komunikacyjnych o zmiennych wagach. ZN Pol. Śl. Studia Informatica, Vol. 23, No. 4(51), Gliwice 2002, s. 85÷104.
3. Peschel M., Riedel C.: Polioptymalizacja. Metody podejmowania decyzji kompromisowych w zagadnieniach inżynieryjno-technicznych. WNT, Warszawa 1979.
4. Stadler W.: A survey of Multicriteria Otimization or the Vector Maximum Problem. Part I: 1776-1960, Journal of Optimization Theory & Aplication, tom 29, nr 1, USA wrzesień 1979, s. 1÷52.
5. Cormen T. H., Leiserson Ch. E., Rivest R. L.: Wprowadzenie do algorytmów. WNT, Warszawa 2000.
6. Lipski W.: Kombinatoryka dla programistów. WNT, Warszawa 1989.
7. Skriver A.J.V., Andersen K.A.: A label correcting approach for solving bicriterion shortest-path problems. Computers & Operations Research, Vol. 27, 2000, s. 507÷524.
8. Brumbaugh-Smith J., Shier S.: An empirical investigation of some bicriterion shortest path algorithms. European Journal of Operational Research, Vol. 43, North-Holland 1989, s. 216÷224.
9. Martins E. Q. V., Pascoal M. M. B.: An algorithm for ranking optimal paths. Departamento de Matematica, Universidade de Coimbra, Technical Report 01/004, CISUC, 2000 (http://www.mat.uc.pt/~marta/Publicacoes/rank_optimal.ps.gz).
10. Jungnickel D.: Graphs, networks and algorithms. Springer, Berlin 1999.
11. Widuch J.: Algorytmy optymalizacji wielokryterialnej w problemach komunikacyjnych. Rozprawa doktorska, Politechnika Śląska, Gliwice 2007.

Recenzent: Dr hab. inż. Mariusz Boryczka

Wpłynęło do Redakcji 19 października 2009 r.

Abstract

The communication networks routing problem (CNRP) is an example of multicriteria optimization problem (MOP). In the paper we present fundamental informations about MOP and we present the CNRP description and analysis. In general case the solution of the MOP is not a single solution but a set of solutions called the set of non-dominated solutions. We present the MKSP algorithm to find the routes between two given stops: the start stop and the final stop, and the time of start of journey at the start stop. We have two optimization criteria: time and cost of travel. Computed routes belonging to the set of non-dominated solution. The algorithm bases on the KSP algorithm for solving *K*-th shortest path problem [9]. The path in the MKSP are computed using the path tree which stores paths from the start node to the final node.

The MKSP algorithm was tested on the communication network consisted of 1211 stops and 500 communiaction lines. The network did not describe any from the real communication-systems and it was created only for tests. The test results were compared with the results obtained by the DFSB algorithm based on depth-first search and backtracking [1] and the results obtained by the BFS algorithm based on breadth-first search [2]. The time of computations using the MKSP is greater than time of computation using the DFSB but it is lower than the time of computations using the BFS. The result of the tests show that it is possible to use the MKSP to find the routes in small communication networks.

Adres

Jacek WIDUCH: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16,
44-100 Gliwice, Polska, jacek.widuch@polsl.pl.