

Jacek WIDUCH
Politechnika Śląska, Instytut Informatyki

ROZWIĄZANIE PROBLEMU WYZNACZANIA POŁĄCZEŃ W SIECIACH KOMUNIKACYJNYCH Z ZASTOSOWANIEM METODY SKALARYZACJI

Streszczenie. Problem wyznaczania połączeń w sieciach komunikacyjnych jest przykładem zadania optymalizacji wielokryterialnej, którego rozwiązaniem jest zbiór rozwiązań niezdominowanych. Wyznaczenie połączeń polega na rozwiązaniu dwukryterialnego problemu najkrótszych ścieżek w grafie ważonym o zmiennych wagach. W pracy przedstawiono algorytm umożliwiający wyznaczenie połączeń należących do zbioru rozwiązań niezdominowanych. Rozwiązania są wyznaczane z użyciem metody skalaryzacji, w której k ($k > 1$) funkcji kryterialnych jest zastępowanych jedną ważoną funkcją kryterialną. Zaprezentowano także przykładowe wyniki przeprowadzonych badań eksperymentalnych.

Słowa kluczowe: problem transportowy, zbiór rozwiązań niezdominowanych, optymalizacja wielokryterialna, metoda skalaryzacji, dwukryterialny problem wyznaczania najkrótszej ścieżki, drzewo ścieżek

SOLVING THE COMMUNICATION NETWORK ROUTING PROBLEM USING SCALARIZATION METHOD

Summary. The communication networks routing problem is an example of multicriteria optimization problem which the solution is the set of non-dominated solutions. Establishing routes consists in solving the bicriterion shortest path problem in the weighted graph with non-constant weights. In the paper an algorithm which determines routes belong to the set of non-dominated solutions is shown. The solutions are computed using scalarization method where the k ($k > 1$) criterion functions are replaced with single weighted criterion function. Apart from that a sample results of experimental tests are presented.

Keywords: transportation problem, set of non-dominated solutions, multicriteria optimization, scalarization method, bicriterion shortest path problem, paths tree

1. Wstęp

Jednym z badanych problemów teorii grafów jest problem wyznaczania najkrótszej ścieżki, w którym dany jest skierowany graf $G = (V, E)$ z funkcją wagową $w : E \rightarrow R$, a zadaniem jest znalezienie ścieżki o minimalnej wadze. Rozszerzeniem tego problemu jest wielokryterialny problem wyznaczania najkrótszej ścieżki, w którym danych jest k ($k > 1$) funkcji wagowych. Podczas wyznaczania ścieżki są jednocześnie uwzględniane wszystkie funkcje wagowe. Jeżeli $k = 2$, wówczas mówi się o dwukryterialnym problemie wyznaczania najkrótszej ścieżki. Znanych jest wiele algorytmów rozwiązywania tego problemu, wśród których można wyróżnić: algorytmy korygowania etykiet [1, 4, 7, 24], algorytmy nadawania etykiet [10, 17, 27], algorytmy dwufazowe [19], algorytmy działające na bazie wyznaczenia K najkrótszych ścieżek [2] oraz pozostałe [8, 13, 14, 15, 16, 22]. W wymienionych algorytmach zakłada się stałe wagi łuków grafu.

W niniejszej pracy został omówiony przykładowy dwukryterialny problem wyznaczania najkrótszej ścieżki, jakim jest problem wyznaczania połączeń w sieciach komunikacyjnych. Celem rozważanego problemu jest wyznaczenie połączeń komunikacyjnych między zadaną parą przystanków i ustaloną godziną rozpoczęcia podróży. Podczas wyznaczania połączeń należy dążyć do jednoczesnej minimalizacji czasu i kosztu przejazdu. W przedstawionym algorytmie do wyznaczania połączeń komunikacyjnych została użyta metoda skalaryzacji.

Niniejsza praca stanowi kontynuację prac [31, 32, 34], w których omówiono problem wyznaczania połączeń w sieciach komunikacyjnych oraz przedstawiono algorytmy rozwiązywania tego problemu. W pracy [31] omówiono algorytm *DFSB*, w którym do wyznaczenia połączeń komunikacyjnych jest użyta zmodyfikowana metoda powrotów i przeszukiwanie grafu w głąb. W algorytmie *BFS* przedstawionym w pracy [32] połączenia komunikacyjne są wyznaczone za pomocą zmodyfikowanej metody przeszukiwania grafu wszerz. Algorytm ten został opracowany na podstawie algorytmu przedstawionego w pracy [24]. Z kolei w pracy [34] został przedstawiony algorytm *MKSP*, będący modyfikacją algorytmu wyznaczania K najkrótszych ścieżek [18].

2. Problem wyznaczania połączeń w sieciach komunikacyjnych

2.1. Sformułowanie problemu

Dana jest sieć komunikacyjna składająca się z n przystanków, mających numery z zakresu $1, \dots, n$. W sieci kursują pojazdy M linii komunikacyjnych o numerach z zakresu $1, \dots, M$. Sieć jest podzielona na strefy, co ma wpływ na sposób wyznaczania kosztu przejazdu. Dla

każdej linii komunikacyjnej jest zdefiniowany przystanek początkowy i końcowy, pomiędzy którymi kursuje pojazd danej linii, oraz sekwencja przystanków, przez które następuje przejazd pojazdu. Przejazd pojazdu odbywa się w określonym kierunku, tj. jeśli pojazd danej linii kursuje z przystanku v_i do przystanku v_j , nie oznacza to, że kursuje z przystanku v_j do przystanku v_i . Pojazd danej linii może kursować w obydwu kierunkach, jednak trasy przejazdu mogą się różnić. Trasa przejazdu pojazdu każdej linii komunikacyjnej składa się z sekwencji różnych przystanków, wyjątkiem są przystanki początkowy i końcowy linii. Linia, w której przystanki początkowy i końcowy są takie same jest nazywana linią okrężną.

Pojazdy komunikacyjne kursują pomiędzy przystankiem początkowym i końcowym linii z określoną częstotliwością. Niech trasa przejazdu pojazdu i -tej linii składa się z sekwencji przystanków: $\langle v_0, v_1, \dots, v_{k-1}, v_k \rangle$, gdzie v_0 jest przystankiem początkowym, a v_k jest przystankiem końcowym linii. Pojazd i -tej linii komunikacyjnej wyjeżdża z przystanku v_0 o godzinie T_0 i przejeżdża przez kolejne przystanki v_1, \dots, v_{k-1} wyjeżdżając z nich odpowiednio w godzinach $T_0 + \delta_0, T_0 + \delta_1, \dots, T_0 + \delta_{k-2}$, dojeżdżając do przystanku końcowego v_k o godzinie $T_0 + \delta_{k-1}$. Następny kurs pojazdu i -tej linii rozpoczyna się na przystanku początkowym v_0 o godzinie T_1 ($T_1 = T_0 + \beta_0$), a z przystanków v_1, \dots, v_{k-1} pojazd wyjeżdża w godzinach $T_1 + \delta_0, T_1 + \delta_1, \dots, T_1 + \delta_{k-2}$ i dojeżdża do przystanku v_k o godzinie $T_1 + \delta_{k-1}$. Zakłada się, że godzina wyjazdu pojazdu z przystanku jest równa godzinie przyjazdu, tj. pomija się czas wsiadania i wysiadania pasażerów. Przyjmuje się idealny model sieci, w którym pojazdy kursują zgodnie z rozkładem jazdy. Pojazd i -tej linii komunikacyjnej wykonuje p kursów wyjeżdżając z przystanku v_0 w godzinach T_0, \dots, T_{p-1} ($T_0 < \dots < T_{p-1}$), gdzie $T_j = T_{j-1} + \beta_{j-1}$ ($j = 1, \dots, p-1$). Wartości $T_0, \beta_0, \dots, \beta_{p-2}, \delta_0, \dots, \delta_{k-1}$ są zdefiniowane rozkładem jazdy i -tej linii, przy czym $0 < \beta_0 < \dots < \beta_{p-2}; 0 < \delta_0 < \dots < \delta_{k-1}$.

Dla zadanej sieci komunikacyjnej, rozkładu jazdy i tras przejazdu pojazdów linii komunikacyjnej, dane są przystanki początkowy v_s i końcowy v_e podróży oraz godzina rozpoczęcia podróży T_s na przystanku v_s . Należy wyznaczyć trasę przejazdu z przystanku v_s do przystanku v_e , tj. sekwencję przystanków, przez które następuje przejazd, linie komunikacyjne, pojazdami których następuje przejazd pomiędzy przystankami, godziny wyjazdów z poszczególnych przystanków należących do trasy oraz przystanki ewentualnych przesiadek. Trasę przejazdu należy wyznaczyć minimalizując czas i koszt przejazdu, przy założeniu że wyjazd z przystanku początkowego v_s nie może nastąpić wcześniej niż w godzinie rozpoczęcia podróży T_s .

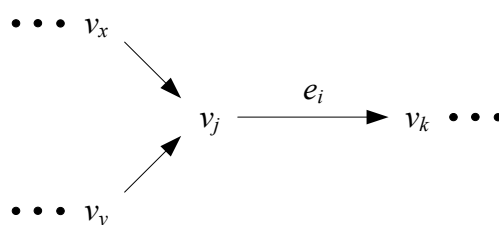
Czas przejazdu wynika z przyjętej trasy przejazdu i jest on równy sumie czasów przejazdu pomiędzy przystankami należącymi do trasy, ewentualnego czasu oczekiwania na przystanku początkowym v_s oraz ewentualnych czasów oczekiwania na połączenie

w przypadku przesiadki. Czasy przejazdu pomiędzy przystankami są zdefiniowane rozkładem jazdy.

Koszt przejazdu jest wyznaczany w następujący sposób. Koszt przejazdu bez przesiadki w granicach jednej strefy wynosi c_1 ($c_1 > 0$) jednostek, w granicach dwóch stref c_2 ($c_2 > c_1$) jednostek, a w granicach trzech lub więcej stref c_3 ($c_3 > c_2$) jednostek. W przypadku wystąpienia przesiadek łączny koszt przejazdu z przystanku v_s do przystanku v_e jest równy sumie kosztów przejazdu pomiędzy przystankami, na których następuje przesiadka. Niektóre linie komunikacyjne są oznaczone jako pospieszne. Przejazd pojazdem takiej linii odbywa się szybciej, co jest uwzględnione w rozkładzie jazdy, natomiast koszt przejazdu jest dwa razy większy od kosztu przejazdu pojazdem linii zwykłej.

2.2. Analiza problemu

Sieć komunikacyjna jest opisana za pomocą skierowanego grafu ważonego $G = (V, E)$, zawierającego n wierzchołków reprezentujących przystanki pojazdów komunikacyjnych [6, 11, 12, 23]¹. Trasy przejazdu pojazdów komunikacyjnych są reprezentowane przez łuki grafu ze zbioru E . Każdy łuk $e_i = (v_j, v_k)$ ($e_i \in E$; $v_j, v_k \in V$) odpowiada jednej linii komunikacyjnej, której pojazdy kursują z przystanku v_j do przystanku v_k . Ponieważ przejazd odbywa się w określonym kierunku, więc obecność w grafie łuku (v_j, v_k) dla i -tej linii nie oznacza obecności w grafie łuku (v_k, v_j) dla tej linii. Między daną parą przystanków mogą kursować pojazdy więcej niż jednej linii komunikacyjnej, dlatego graf może zawierać łuki równoległe.



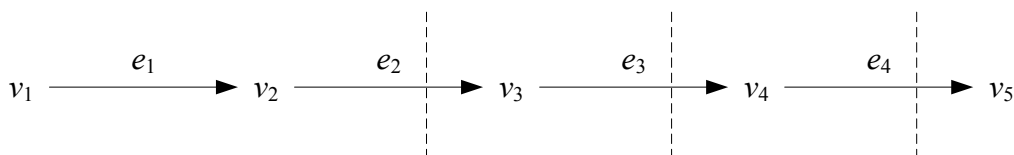
Rys. 1. Fragment grafu reprezentującego sieć komunikacyjną

Fig. 1. A part of graph representing a communication network

Każdy łuk $e_i = (v_j, v_k)$ ma 3 wagi: $l(e_i)$, $t(e_i)$ i $c(e_i)$. Waga $l(e_i)$ opisuje numer linii komunikacyjnej, pojazdem której odbywa się przejazd z v_j do v_k . Waga $t(e_i) = T_k - T_j$ jest równa różnicy między godziną przyjazdu T_k do v_k a godziną przyjazdu T_j do v_j , przyjmuje wartości dodatnie, przy czym jest wagą o zmiennych wartościach. Waga ta jest równa sumie czasu przejazdu z v_j do v_k i czasu oczekiwania na wyjazd w v_j . Czas przejazdu pojazdem danej linii z v_j do v_k jest stały, tak więc o wartości wagi $t(e_i)$ decyduje czas oczekiwania w wierzchołku v_j , który jest czasem zmiennym (rys. 1). Do wierzchołka v_j istnieje możliwość

¹ W dalszej części pracy używając zwrotu wierzchołek, w odniesieniu do grafu reprezentującego sieć komunikacyjną, będziemy mieć na myśli przystanek w sieci komunikacyjnej, który on reprezentuje.

dojazdu z wierzchołków v_x i v_y , przy czym z wierzchołka v_x dojazd następuje w czasie t_j^x , a z wierzchołka v_y w czasie t_j^y . Niech w obydwu przypadkach godzina wyjazdu z wierzchołka v_j jest równa T_j ($T_s + t_j^x \leq T_j$; $T_s + t_j^y \leq T_j$). Jeżeli $t_j^x \neq t_j^y$, to czas oczekiwania w wierzchołku v_j w obydwu przypadkach będzie różny, a co za tym idzie, wartość wagi $t(e_i)$ w obydwu przypadkach także będzie różna.



Rys. 2. Fragment grafu reprezentującego trasę linii komunikacyjnej
Fig. 2. A part of graph representing a route of communication line

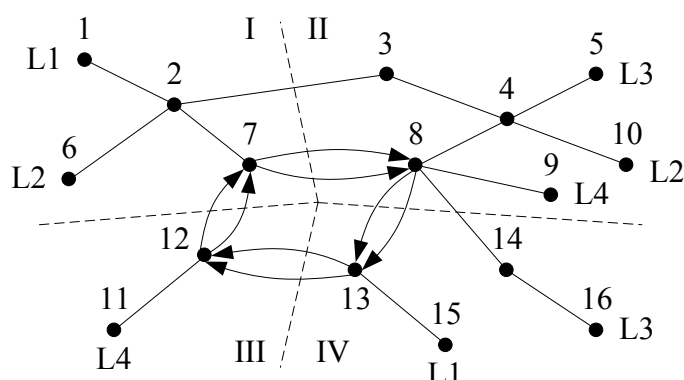
Waga $c(e_i) = c_k - c_j$ jest równa różnicy kosztów przejazdu c_k i c_j , gdzie c_k jest kosztem przejazdu z v_s do v_k , a c_j jest kosztem przejazdu z v_s do v_j . Przyjmuje ona wartości nieujemne i jest także wagą o zmiennych wartościach, a o jej wartości decyduje to, do jakiej strefy należą wierzchołki v_j i v_k oraz czy w wierzchołku v_j jest dokonywana przesiadka. W przypadku dokonania przesiadki w wierzchołku v_j waga jest równa $c(e_i) = c_1$, jeżeli wierzchołki v_j i v_k znajdują się w tej samej strefie lub jest równa $c(e_i) = c_2$, jeżeli wierzchołki należą do różnych stref. Jeżeli w wierzchołku v_j nie jest dokonywana przesiadka i wierzchołki v_j i v_k znajdują się w tej samej strefie, to podczas przejazdu z v_j do v_k nie jest przekraczana strefa, czyli koszt przejazdu nie wzrasta, tak więc waga jest równa $c(e_i) = 0$. W przypadku kiedy w wierzchołku v_j nie jest dokonywana przesiadka i wierzchołki v_j i v_k należą do różnych stref, o wartości wagi decyduje liczba stref, jakie zostały przekroczone od ostatniej przesiadki przy dojeździe do v_j . Jeżeli nie została przekroczona żadna strefa, to waga jest równa $c(e_i) = c_2 - c_1$, w przypadku przekroczenia jednej strefy $c(e_i) = c_3 - c_2$, a przy przekroczeniu dwóch i więcej stref $c(e_i) = 0$.

Tabela 1

Wartości wag c łuków grafu z rys. 2 przy wyznaczaniu kosztu przejazdu do wierzchołka v_5 z różnych wierzchołków początkowych v_s

Wierzchołek v_s	$c(e_1)$	$c(e_2)$	$c(e_3)$	$c(e_4)$
v_1	c_1	$c_2 - c_1$	$c_3 - c_2$	0
v_2	–	c_2	$c_3 - c_2$	0
v_3	–	–	c_2	$c_3 - c_2$
v_4	–	–	–	c_2

Zmienna wartość wagi zostanie zilustrowana na przykładzie dojazdu do wierzchołka v_5 pojazdem linii komunikacyjnej, której trasa została przedstawiona na rys. 2 (linią przerywaną zostały zaznaczone granice stref). Wartość wagi c łuków e_1, \dots, e_4 zależy od wierzchołka początkowego podróży v_s , z którego rozpoczyna się przejazd do wierzchołka v_5 . Wartości te dla wierzchołków początkowych podróży odpowiednio v_1, \dots, v_4 i wierzchołka końcowego podróży $v_e = v_5$ zostały przedstawione w tabeli 1.



Rys. 3. Graf reprezentujący przykładową sieć komunikacyjną
 Fig. 3. A graph representing a sample communication network

Graf reprezentujący przykładową sieć komunikacyjną został przedstawiony na rys. 3, przy czym każdą krawędź należy traktować jako parę łuków przeciwnie skierowanych. Sieć podzielona jest na 4 strefy, kursują w niej pojazdy czterech linii komunikacyjnych, których trasy przejazdu przedstawiono w tabeli 2. Każda z linii kursuje w obydwu kierunkach pomiędzy przystankami początkowym i końcowym (przystanki początkowy i końcowy każdej z linii zaznaczono na rys. 3). W grafie występują łuki równoległe, ponieważ między wybranymi parami przystanków w danym kierunku kursują pojazdy więcej niż jednej linii komunikacyjnej.

Tabela 2
 Trasy przejazdu pojazdów linii komunikacyjnych w sieci komunikacyjnej reprezentowanej przez graf z rys. 3

Linia komunikacyjna	Trasa przejazdu
1	1, 2, 7, 8, 13, 15
	15, 13, 12, 7, 2, 1
2	6, 2, 3, 4, 10
	10, 4, 3, 2, 6
3	5, 4, 8, 14, 16
	16, 14, 8, 4, 5
4	11, 12, 7, 8, 9
	9, 8, 13, 12, 11

Połączenie komunikacyjne z przystanku początkowego v_s do przystanku końcowego v_e jest reprezentowane przez ścieżkę $p_{se} = \langle v_0, (v_0, v_1), v_1, \dots, v_{m-1}, (v_{m-1}, v_m), v_m \rangle$ w grafie G reprezentującym sieć komunikacyjną oraz godziny odjazdów T_0, \dots, T_{m-1} , gdzie T_i jest godziną odjazdu z v_i ($i = 0, \dots, m-1; v_i \in p_{se}$), a $v_0 = v_s$ i $v_m = v_e$. Długość ścieżki p_{se} jest równa liczbie łuków należących do ścieżki i jest oznaczana przez $len(p_{se})$. Ścieżka p_{se} ma dwie wagi $T(p_{se})$ i $C(p_{se})$ będące sumą wag t i c łuków należących do ścieżki, które są określone zależnością (1).

$$T(p_{se}) = \sum_{i=1}^m t(v_{i-1}, v_i), \quad C(p_{se}) = \sum_{i=1}^m c(v_{i-1}, v_i) \quad (1)$$

Wagi $T(p_{se})$ i $C(p_{se})$ reprezentują odpowiednio czas i koszt przejazdu z v_s do v_e . Zgodnie z założeniami przedstawionymi w punkcie 2.1 należy wyznaczyć ścieżkę p_{se} w grafie G minimalizując wagi $T(p_{se})$ i $C(p_{se})$.

Problem wyznaczania połączeń komunikacyjnych jest przykładem zadania optymalizacji wielokryterialnej, w którym danych jest k ($k > 1$) funkcji kryterialnych f_i ($i = 1, \dots, k$), na podstawie których jest dokonywana ocena rozwiązań [3, 21, 26, 28, 29]. Funkcje kryterialne mogą być minimalizowane lub maksymalizowane. W pierwszym przypadku dąży się do wyznaczenia rozwiązania, dla którego funkcja przyjmuje wartość minimalną, a w drugim wartość maksymalną. W większości przypadków nie istnieje jedno rozwiązanie, dla którego wszystkie funkcje kryterialne przyjmują optymalne wartości, stąd rozwiązaniem zadania optymalizacji wielokryterialnej w ogólnym przypadku jest zbiór rozwiązań, zwany zbiorem rozwiązań niezdominowanych (rozwiązań Pareto optymalnych [20]).

Niech będzie danych k minimalizowanych funkcji kryterialnych f_1, \dots, f_k . Rozwiązanie X dominuje rozwiązanie Y , co oznaczamy przez $X > Y$, jeżeli jest spełniona zależność (2).

$$\forall_{i \in \{1, \dots, k\}} f_i(Y) \leq f_i(X) \wedge \exists_{j \in \{1, \dots, k\}} f_j(Y) < f_j(X) \quad (2)$$

Zbiór rozwiązań niezdominowanych S_{ND} jest zbiorem takich rozwiązań, dla których nie istnieją rozwiązania, które je dominują, tj.: $S_{ND} = \{X: \neg \exists Y, Y > X\}$.

Rozwiązanie problemu wyznaczania połączeń komunikacyjnych sprowadza się do rozwiązania dwukryterialnego problemu wyznaczania najkrótszej ścieżki w grafie o zmiennych wagach. Rozwiązaniem problemu jest zbiór ścieżek, reprezentujących połączenia komunikacyjne, stanowiących zbiór rozwiązań niezdominowanych. Wagi T i C ścieżek są funkcjami kryterialnymi minimalizowanymi. Jak wykazano w pracach [9, 10], dwukryterialny problem wyznaczania najkrótszej ścieżki w grafie o stałych wagach, poprzez transformację do 0–1 problemu plecakowego, należy do grupy problemów NP–zupełnych. Liczba rozwiązań w przypadku pesymistycznym rośnie wykładniczo wraz ze wzrostem liczby wierzchołków grafu [24].

3. Metoda skalaryzacji

W przypadku kiedy algorytmy wyznaczenia wszystkich rozwiązań niezdominowanych mają złożoność większą niż wielomianowa, rezygnuje się z ich wyznaczenia i wyznacza się jedynie niektóre z nich. Jedną z metod umożliwiających wyznaczenie rozwiązań niezdomino-

wanych jest skalaryzacja. W metodzie tej poszczególnym funkcjom kryterialnym przyporządkowywane są wagi i jest wprowadzana zastępcza funkcja kryterialna F_Z [20].

Niech będzie danych k funkcji kryterialnych f_1, \dots, f_k oraz niech funkcji f_i ($i = 1, \dots, k$) jest przyporządkowana waga λ_i . Zastępcza funkcja kryterialna F_Z jest określona zależnością (3), przy czym najczęściej wagi dobierane są tak, aby była spełniona zależność (4).

$$F_Z = \sum_{i=1}^k \lambda_i f_i \quad (3)$$

$$\sum_{i=1}^k \lambda_i = 1 \quad (4)$$

Po wprowadzeniu zastępczej funkcji kryterialnej F_Z dąży się do wyznaczenia rozwiązania, dla którego funkcja F_Z przyjmuje wartość minimalną lub maksymalną. Dzięki wprowadzeniu funkcji F_Z zadanie optymalizacji wielokryterialnej jest redukowane do zadania optymalizacji jednokryterialnej i istnieje możliwość zastosowania metod rozwiązywania, które są mniej złożone niż w przypadku zadania optymalizacji wielokryterialnej. Przez odpowiedni dobór wartości wag λ_i jest określany wpływ danej funkcji kryterialnej na wyznaczone rozwiązanie. Wprowadzenie zastępczej funkcji kryterialnej powoduje, że w ogólnym przypadku nie zostaną wyznaczone wszystkie rozwiązania ze zbioru rozwiązań niezdominowanych, a jedynie te, dla których wartość zastępczej funkcji kryterialnej przyjmuje wartość minimalną lub maksymalną [5, 25, 30]. Przeprowadzając wielokrotnie proces optymalizacji dla różnych wartości wag λ_i istnieje możliwość uzyskania kolejnych rozwiązań ze zbioru rozwiązań niezdominowanych.

4. Drzewo ścieżek

Wszystkie ścieżki prowadzące z wierzchołka początkowego v_s do wierzchołka końcowego v_e , które reprezentują połączenia komunikacyjne, można przedstawić w postaci drzewa ścieżek G_T , w którym korzeniem jest wierzchołek v_s , a liśćmi wierzchołek v_e . Dla każdej pary ścieżek $p_i, p_j \in G_T$ określonych zależnościami (5) i (6) jest spełniona zależność (7).

$$p_i = \langle v_0^i = v_s, (v_0^i, v_1^i), v_1^i, \dots, v_{n-1}^i, (v_{n-1}^i, v_n^i), v_n^i = v_e \rangle \quad (5)$$

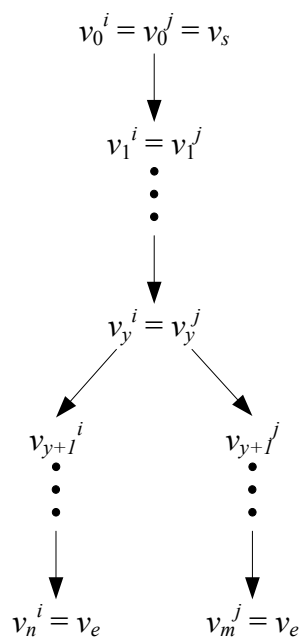
$$p_j = \langle v_0^j = v_s, (v_0^j, v_1^j), v_1^j, \dots, v_{m-1}^j, (v_{m-1}^j, v_m^j), v_m^j = v_e \rangle \quad (6)$$

$$\exists_{0 \leq y < \min(m, n)} \forall_{x \in \{0, \dots, y\}} v_x^i = v_x^j \quad (7)$$

Zależność (7) można zilustrować za pomocą rys. 4, na którym ścieżki p_i i p_j , określone zależnościami (5) i (6), przedstawiono w postaci drzewa ścieżek G_T . Wierzchołki v_y^i oraz v_y^j są ostatnimi wierzchołkami na wspólnej podścieżce. Wierzchołek, który jest ostatnim wierz-

chołkiem na wspólnej podścieżce dwóch ścieżek, jest nazywany wierzchołkiem odchyłającym (ang. *deviation node*) i jest oznaczany przez $d(p)$.

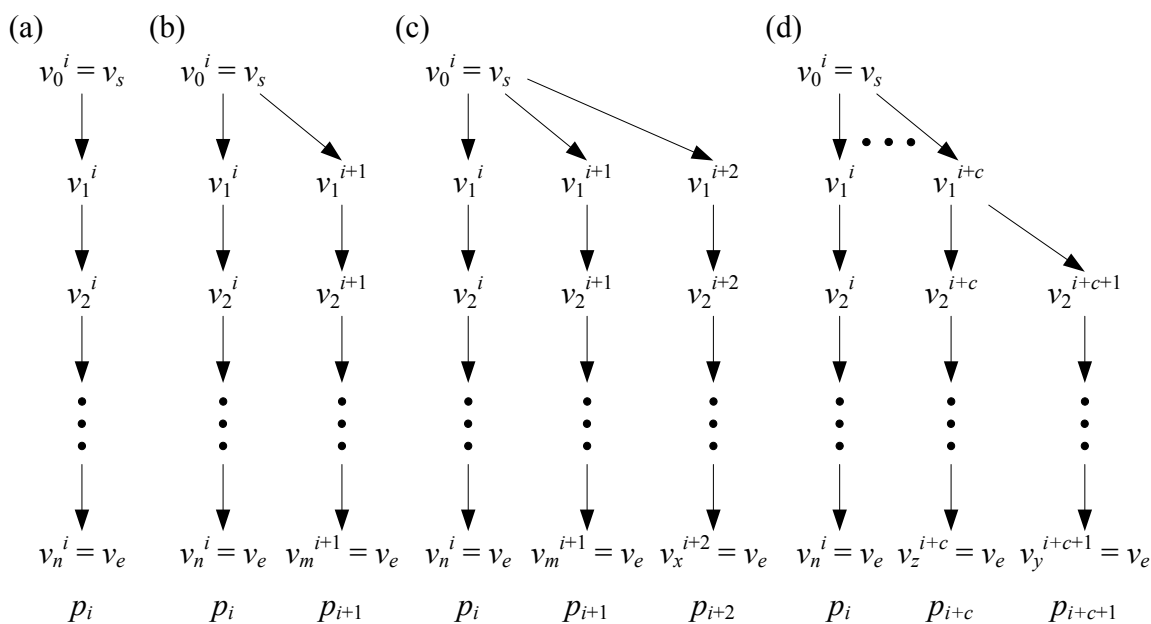
Tworzenie drzewa ścieżek rozpoczyna się od wyznaczenia dowolnej ścieżki p_i z wierzchołka v_s do wierzchołka v_e . Na jej podstawie są wyznaczane pozostałe ścieżki, przy czym w trakcie ich wyznaczania konieczna jest modyfikacja grafu G poprzez usuwanie z niego niektórych wierzchołków oraz łuków. Operacja ta zapobiega wielokrotnemu wyznaczaniu tych samych ścieżek. Na podstawie ścieżki p_i nowe ścieżki powstają przez usunięcie z grafu kolejno łuków $(d(p_i), v_j^i)$, (v_j^i, v_{j+1}^i) , ..., (v_{n-1}^i, v_n^i) , a następnie wyznaczeniu ścieżki odpowiednio z wierzchołków $\{d(p_i), v_j^i, \dots, v_{n-1}^i\}$ do wierzchołka końcowego v_e .



Rys. 4. Przykładowe drzewo ścieżek

Fig. 4. A sample paths tree

Przed wyznaczeniem pierwszej ścieżki są usuwane z grafu wszystkie wierzchołki poprzedzające wierzchołek $d(p_i)$ w ścieżce p_i . Jeżeli nie ma możliwości wyznaczenia ścieżki z danego wierzchołka, to jest on usuwany z grafu i następuje próba wyznaczenia ścieżki z kolejnego wierzchołka. Po wyznaczeniu pierwszej ścieżki p_{i+1} jest przerywana analiza ścieżki p_i i jest podejmowana próba utworzenia nowych ścieżek na podstawie nowo utworzonej ścieżki. Rozpoczynając analizę ścieżki p_{i+1} , nie ma konieczności usuwania z grafu jakichkolwiek wierzchołków i łuków, ponieważ zostały one usunięte w trakcie analizy ścieżek p_1, \dots, p_i . W przypadku braku możliwości utworzenia kolejnych ścieżek na podstawie ścieżki p_{i+1} następuje powrót do analizy ścieżki p_i oraz umieszczenie w grafie wszystkich wierzchołków i łuków, które zostały usunięte podczas analizy ścieżki p_{i+1} . Analiza ścieżki p_i jest kontynuowana od wierzchołka będącego następnikiem w ścieżce wierzchołka, dla którego zakończyła się poprzednia analiza ścieżki p_i .



Rys. 5. Kolejne etapy konstruowania drzewa ścieżek

Fig. 5. A next steps of construction paths tree

Etapy algorytmu tworzenia drzewa ścieżek zostały przedstawione na rys. 5. Niech p_i (rys. 5a) będzie ścieżką, na podstawie której są tworzone kolejne ścieżki, przy czym $d(p_i) = v_0^i$. Tworzenie nowych ścieżek polega na wyznaczeniu ścieżki do wierzchołka v_e z wszystkich kolejnych wierzchołków należących do ścieżki, począwszy od wierzchołka $d(p_i)$. W celu poprawnego wyznaczania ścieżek powinny być usunięte z grafu wszystkie wierzchołki poprzedzające w ścieżce wierzchołek $d(p_i)$. Jest to zapewnione przez kolejność analizy ścieżek, na podstawie których wyznaczane są nowe ścieżki. Ścieżka p_{i+1} (rys. 5b) jest uzyskiwana przez usunięcie z grafu łuku (v_0^i, v_1^i) wychodzącego z wierzchołka $d(p_i)$ i wyznaczenie ścieżki p_{i+1} z wierzchołka $d(p_i) = v_0^i$ do v_e . Dla utworzonej ścieżki $d(p_{i+1}) = v_0^i$. Po wyznaczeniu ścieżki p_{i+1} jest przerywana analiza ścieżki p_i i następuje próba utworzenia nowej ścieżki na podstawie p_{i+1} . Odbывается to w podobny sposób jak w przypadku ścieżki p_i , tj. przez usunięcie z grafu łuku (v_0^i, v_1^{i+1}) wychodzącego z wierzchołka $d(p_{i+1})$ i wyznaczenie ścieżki p_{i+2} z wierzchołka $d(p_i) = v_0^i$ do wierzchołka v_e (rys. 5c). W taki sposób uzyskiwanych jest c ścieżek p_{i+1}, \dots, p_{i+c} , dla których wierzchołkiem odchyłającym jest wierzchołek v_0^i (rys. 5d). Ponieważ na podstawie ścieżki p_{i+c} nie ma możliwości utworzenia nowej ścieżki z wierzchołka $d(p_{i+c}) = v_0^i$ do wierzchołka v_e , więc następuje usunięcie z grafu wierzchołka v_0^i oraz łuku (v_1^{i+c}, v_2^{i+c}) i wyznaczenie ścieżki p_{i+c+1} z kolejnego wierzchołka należącego do ścieżki, a jest nim wierzchołek v_1^{i+c} . W podobny sposób jest podejmowana próba wyznaczenia ścieżki na podstawie ścieżki p_{i+c+1} .

W przypadku braku możliwości wyznaczenia nowej ścieżki na podstawie ścieżki p_{i+1} i każdej z kolejnych ścieżek, następuje powrót do analizy ścieżki p_i . Ponieważ analiza ścieżki p_i po wyznaczeniu ścieżki p_{i+1} zakończyła się na wierzchołku v_0^i , więc będzie ona kontynu-

wana od wierzchołka v_1^i , a wierzchołek v_0^i zostanie usunięty z grafu. Ponadto, w grafie zostaną umieszczone wszystkie wierzchołki i łuki, które zostały usunięte podczas analizy ścieżki p_{i+1} .

Inny algorytm tworzenia drzewa ścieżek został zaproponowany w pracy [18]. Tworzenie drzewa ścieżek także rozpoczyna się od wyznaczenia dowolnej ścieżki p_i z wierzchołka v_s do wierzchołka v_e . Inna jest natomiast kolejność analizy wierzchołków, ponieważ po wyznaczeniu ścieżki p_{i+1} nie jest przerywana analiza ścieżki p_i , tylko podejmowana jest próba wyznaczenia kolejnych ścieżek na podstawie tej ścieżki. Ścieżka p_{i+1} jest poddawana analizie i na jej podstawie jest podejmowana próba utworzenia kolejnych ścieżek dopiero po wyznaczeniu wszystkich możliwych ścieżek, jakie można uzyskać na podstawie ścieżki p_i . Dzięki takiej kolejności analizy ścieżek przed rozpoczęciem analizy ścieżki p_j zachodzi konieczność usunięcia z grafu wszystkich wierzchołków, które poprzedzają w tej ścieżce wierzchołek $d(p_j)$. Z tego powodu wierzchołki są usuwane z grafu większą liczbą razy niż w algorytmie, który został wcześniej opisany.

5. Algorytm wyznaczania połączeń w sieciach komunikacyjnych

W algorytmie została użyta metoda skalaryzacji oraz algorytm tworzenia drzewa ścieżek opisany w rozdziale 4. Zgodnie z zależnością (3) dla wyznaczanych ścieżek reprezentujących jest wprowadzana zastępcza funkcja kryterialna określona zależnością (8). Czas przejazdu T jest uwzględniany z wagą λ_T , a koszt przejazdu C z wagą λ_C . Wartości wag λ_T i λ_C są dobrane tak, aby wartości funkcji kryterialnej F_Z ścieżek o minimalnym czasie i koszcie przejazdu były sobie równe oraz była spełniona zależność (4), co określa układ równań (9).

$$F_Z = \lambda_T T + \lambda_C C. \quad (8)$$

$$\begin{cases} \lambda_T T_{\max} + \lambda_C C_{\min} = \lambda_T T_{\min} + \lambda_C C_{\max} \\ \lambda_T + \lambda_C = 1 \end{cases} \quad (9)$$

W układzie równań (9) wartości (T_{\max}, C_{\min}) są wagami ścieżki o minimalnym koszcie przejazdu, natomiast (T_{\min}, C_{\max}) są wagami ścieżki o minimalnym czasie przejazdu. Warto zwrócić uwagę na to, że waga C_{\max} ścieżki o minimalnym czasie przejazdu określa maksymalny koszt ścieżki będącej rozwiązaniem niezdominowanym. Podobnie w przypadku ścieżki o minimalnym koszcie przejazdu, jej waga T_{\max} określa maksymalny czas przejazdu dla ścieżki należącej do zbioru rozwiązań niezdominowanych. Powyższe zależności wynikają z definicji rozwiązania niezdominowanego. Z układu równań (9) są otrzymywane wartości wag λ_T i λ_C , określone zależnościami (10) i (11). Zależności te obowiązują wyłącznie

w przypadku, kiedy wagi ścieżek o minimalnym czasie i koszcie przejazdu są różne. Jeżeli są one równe, to wagi λ_T i λ_C przyjmują wartość 0.5.

$$\lambda_T = \frac{C_{\max} - C_{\min}}{C_{\max} - C_{\min} + T_{\max} - T_{\min}} \quad (10)$$

$$\lambda_C = \frac{T_{\max} - T_{\min}}{C_{\max} - C_{\min} + T_{\max} - T_{\min}} \quad (11)$$

Algorytm wyznaczania połączeń komunikacyjnych z użyciem metody skalaryzacji oraz drzewa ścieżek został przedstawiony w postaci procedury *SSP*.

Wejście: $G = (V, E)$ – graf skierowany reprezentujący sieć komunikacyjną
 v_s – wierzchołek reprezentujący przystanek początkowy
 v_e – wierzchołek reprezentujący przystanek końcowy
 T_s – godzina rozpoczęcia podróży na przystanku początkowym
Wyjście: R – zbiór rozwiązań, tj. ścieżek reprezentujących połączenia komunikacyjne

```

1: procedure SSP( $G, v_s, v_e, T_s, R$ )
2: begin
3:    $R := \emptyset;$        $S \leftarrow \emptyset;$    {pusty zbiór rozwiązań i stos zawierający ścieżki}
4:    $p := \text{MinKoszt}(G, v_s, v_e, T_s);$     $R := R \cup \{p\};$ 
5:    $p := \text{MinCzas}(G, v_s, v_e, T_s);$     $R := R \cup \{p\};$ 
6:   wyznacz wartości  $\lambda_T, \lambda_C$  i  $F_{\max}$ ;
7:    $p_{\text{akt}} := p;$            {aktualnie analizowana ścieżka}
8:    $i := 0;$                  {indeks analizowanego wierzchołka w ścieżce  $p_{\text{akt}}$ }
9:   koniec := false;       {znacznik określający, czy zakończyć obliczenia}
10:  while not koniec do
11:     $op := \text{WYZNACZ\_ŚCIEŻKĘ};$    {aktualnie wykonywana operacja}
12:    while  $i < \text{len}(p_{\text{akt}})$  and  $op \neq \text{NOWA\_ŚCIEŻKA}$  do
13:      usuń z grafu  $G$  łuk  $(v_i, v_{i+1})$  należący do ścieżki  $p_{\text{akt}}$ ;
14:       $(p, op) := \text{NowaŚcieżka}(G, v_i, v_e, T_s, \lambda_T, \lambda_C, F_{\max}, p_{\text{akt}});$ 
15:      if  $op = \text{BRAK\_ŚCIEŻKI}$  then
16:        usuń z grafu  $G$  wierzchołek  $v_i;$     $i := i + 1;$ 
17:      end if
18:    end while
19:    if  $op = \text{NOWA\_ŚCIEŻKA}$  then   {znaleziono nową ścieżkę}
20:       $R := R \cup \{p\};$ 
21:       $S \leftarrow (p_{\text{akt}}, i);$    {umieszczenie ścieżki  $p_{\text{akt}}$  na stosie}
22:       $p_{\text{akt}} := p;$     $i := d(p_{\text{akt}});$ 
23:    else   { $op = \text{BRAK\_ŚCIEŻKI}$ }
24:      umieść w grafie  $G$  usunięte wierzchołki i łuki dla ścieżki  $p_{\text{akt}}$ ;
25:      if  $S \neq \emptyset$  then   {stos nie jest pusty}
26:         $(p_{\text{akt}}, i) \leftarrow S;$    {pobranie ścieżki ze stosu}
27:        usuń z grafu  $G$  wierzchołek  $v_i;$     $i := i + 1;$ 
28:      else
29:        koniec := true;   {zakończenie obliczeń}
30:      end if
31:    end if
32:  end while
33:  usuń rozwiązania zdominowane ze zbioru  $R;$ 
34:  return  $R;$ 
35: end

```

Wyznaczone połączenia komunikacyjne są zapamiętywane w zbiorze rozwiązań R . Dodatkowo użyty jest stos S zawierający analizowane ścieżki, na podstawie których są wyznaczone nowe ścieżki. W początkowych krokach algorytmu są wyznaczane i umieszczane

w zbiorze rozwiązań R ścieżki o minimalnym koszcie i czasie przejazdu (wiersze 4–5). Do ich wyznaczenia jest stosowany zmodyfikowany algorytm Dijkstry [80]. Na podstawie wyznaczonych ścieżek wyznaczane są, zgodnie z zależnościami (10) i (11), wartości wag λ_T i λ_C , a także wartość F_{\max} (wiersz 6). Wartość F_{\max} jest równa wartości zastępczej funkcji kryterialnej F_Z dla ścieżki o minimalnym czasie przejazdu i jest określona zależnością (12)². W zbiorze rozwiązań R będą zapamiętywane wyłącznie rozwiązania, dla których wartość zastępczej funkcji kryterialnej F_Z nie przekracza wartości F_{\max} .

$$F_{\max} = \lambda_T T_{\min} + \lambda_C C_{\max}. \quad (12)$$

Kolejne rozwiązania są wyznaczane na podstawie aktualnie analizowanej ścieżki $p_{\text{akt}} = \langle v_0 = v_s, (v_0, v_1), v_1, \dots, v_{n-1}, (v_{n-1}, v_n), v_n = v_e \rangle$ (wiersze 10–32), przy czym początkowo jest nią ścieżka o minimalnym czasie przejazdu (wiersz 7). Tworzenie nowych ścieżek odbywa się w iteracji **while** (wiersze 12–18) przez próbę wyznaczenia nowej ścieżki (wiersz 14) z każdego wierzchołka $\{d(p_{\text{akt}}), \dots, v_{n-1}\}$, należącego do ścieżki p_{akt} , do wierzchołka końcowego v_e . W trakcie wyznaczania ścieżek jest użyty znacznik *op* zawierający informację o wykonywanej operacji i przyjmujący jedną z wartości:

- WYZNACZ_ŚCIEŻKĘ – oznacza, że zostanie podjęta próba wyznaczenia nowej ścieżki z analizowanego wierzchołka v_i do wierzchołka v_e ,
- NOWA_ŚCIEŻKA – oznacza, że została wyznaczona nowa ścieżka z wierzchołka v_i do wierzchołka v_e ,
- BRAK_ŚCIEŻKI – oznacza, że nie udało się wyznaczyć nowej ścieżki z wierzchołka v_i do wierzchołka v_e .

Przed wyznaczeniem nowej ścieżki jest usuwany z grafu G łuk (v_i, v_{i+1}) , należący do ścieżki p_{akt} , wychodzący z wierzchołka o indeksie i (wiersz 13). Operacja ta zapobiega ponownemu wyznaczeniu tej samej ścieżki i zapewnia skończoność algorytmu. W przypadku kiedy nie powiodła się operacja wyznaczenia nowej ścieżki z wierzchołka v_i , następuje usunięcie z grafu wierzchołka v_i i przejście do analizy następnego wierzchołka należącego do ścieżki p_{akt} (wiersze 15–17).

Po wyznaczeniu nowej ścieżki p jest ona dodawana do zbioru rozwiązań R (wiersz 20), na stosie S umieszczana jest aktualnie analizowana ścieżka p_{akt} wraz z indeksem i wierzchołka, na którym zakończyła się analiza ścieżki p_{akt} (wiersz 21). Wyznaczona ścieżka p staje się ścieżką p_{akt} (wiersz 22), która zostanie poddana analizie i na podstawie której zostanie podjęta próba wyznaczenia nowych ścieżek. Analiza ścieżki rozpocznie się od wierzchołka odchylającego $d(p_{\text{akt}})$ (wiersz 22).

² Wartość F_{\max} jest równa także wartości zastępczej funkcji kryterialnej F_Z dla ścieżki o minimalnym koszcie przejazdu.

W przypadku braku możliwości wyznaczenia nowej ścieżki na podstawie ścieżki p_{akt} , w grafie G są umieszczane wszystkie wierzchołki i łuki, które zostały usunięte podczas analizy bieżącej ścieżki p_{akt} (wiersz 24). Jeżeli stos S nie jest pusty, to pobierana jest z niego ścieżka, która zostanie poddana analizie, oraz indeks i wierzchołka, na którym zakończyła się poprzednia analiza pobranej ścieżki (wiersz 26). Ponieważ dokonana już została próba wyznaczenia nowych ścieżek z tego wierzchołka, więc jest on usuwany z grafu i następuje przejście do analizy kolejnego wierzchołka należącego do pobranej ścieżki p_{akt} (wiersz 27). W przypadku kiedy stos S jest pusty, następuje zakończenie wyszukiwania rozwiązań przez nadanie znacznikowi *koniec* wartości **true** (wiersz 29). W ostatnim kroku algorytmu usuwane są rozwiązania zdominowane ze zbioru rozwiązań R (wiersz 33).

Próba wyznaczenia nowej ścieżki w wierszu 14 odbywa się z użyciem procedury *NowaŚcieżka*.

Wejście: $G = (V, E)$ – skierowany graf reprezentujący sieć komunikacyjną
 v_i – wierzchołek, z którego jest wyznaczana ścieżka
 v_e – wierzchołek końcowy
 T_s – godzina rozpoczęcia podróży na przystanku początkowym
 λ_T, λ_C – wagi, z jakimi uwzględniany jest czas i koszt przejazdu w zastępczej funkcji kryterialnej F_Z
 F_{max} – maksymalna wartość zastępczej funkcji kryterialnej F_Z dla akceptowalnego rozwiązania
 p_{akt} – ścieżka, na podstawie której jest podejmowana próba utworzenia nowej ścieżki

Wyjście: (p, op) – wyznaczona ścieżka p , informacja o wykonanej operacji

```

1: procedure NowaŚcieżka( $G, v_i, v_e, T_s, \lambda_T, \lambda_C, F_{max}, p_{akt}$ )
2: begin
3:    $T_i :=$  czas przejazdu z wierzchołka  $v_s$  do wierzchołka  $v_i$  w ścieżce  $p_{akt}$ ;
4:    $C_i :=$  koszt przejazdu z wierzchołka  $v_s$  do wierzchołka  $v_i$  w ścieżce  $p_{akt}$ ;
5:   for all  $v \in V$  do
6:      $F_Z[v] := \infty$ ;  $T[v_i] := \infty$ ;  $C[v_i] := \infty$ ;  $\Delta C[v_i] := \infty$ ;
7:   end for
8:    $F_Z[v_i] := \lambda_T T_i + \lambda_C C_i$ ; {wartość funkcji  $F_Z$  dla wierzchołka  $v_i$ }
9:    $T[v_i] := T_i$ ;  $C[v_i] := C_i$ ;  $\Delta C[v_i] := 0$ ;
10:   $Q \leftarrow V$ ; {wstaw wszystkie wierzchołki do kolejki}
11:  koniec := false; {znacznik określający czy zakończyć obliczenia}
12:  while  $Q \neq \emptyset$  and not koniec do
13:     $v_{min} \leftarrow \min(Q)$ ; {pobranie z kolejki wierzchołka o minimalnej wartości  $F_Z$ }
14:    if  $v_{min} = v_e$  or  $F_Z[v_{min}] = \infty$  then
15:      koniec := true;
16:    else
17:      for all  $(v_{min}, v_j) \in E$  and  $v_j \in Q$  then
18:         $T_j :=$  czas dojazdu do wierzchołka  $v_j$ ;
19:         $C_j :=$  koszt dojazdu do wierzchołka  $v_j$ ;
20:         $\Delta C_j :=$  przyrost kosztu dojazdu do wierzchołka  $v_j$ ;
21:         $F_{Zj} := \lambda_T T_j + \lambda_C (C_j - \Delta C_j)$ ; {wartość funkcji  $F_Z$  dla wierzchołka  $v_j$ }
22:        if  $F_{Zj} < F_Z[v_j]$  and  $F_{Zj} \leq F_{max}$  then
23:           $arc[v_j] := (v_{min}, v_j)$ ; {łuk wchodzący do wierzchołka  $v_j$ }
24:           $T[v_j] := T_j$ ;  $C[v_j] := C_j$ ;  $\Delta C[v_j] := \Delta C_j$ ;  $F_Z[v_j] := F_{Zj}$ ;
25:        end if
26:      end for
27:    end if
28:  end while
29:  if  $F_Z[v_e] = \infty$  then {nie wyznaczono rozwiązania (nowej ścieżki)}
```

```

30:   return (∅, BRAK_ŚCIEŻKI);
31:   else                               {wyznaczono nową ścieżkę}
32:     p := ścieżka z vs do ve utworzona na podstawie pakt i arc;
33:     return (p, NOWA_ŚCIEŻKA);
34:   end if
35: end

```

W procedurze *NowaŚcieżka* jest wyznaczana ścieżka z wierzchołka v_i do wierzchołka v_e o minimalnej wartości zastępczej funkcji F_Z . Do wyznaczenia ścieżki jest stosowany zmodyfikowany algorytm Dijkstry. Dla każdego wierzchołka $v_i \in V$ są pamiętane następujące wartości:

- $F_Z[v_i]$ – wartość zastępczej funkcji kryterialnej,
- $T[v_i]$, $C[v_i]$ – czas i koszt dojazdu z wierzchołka początkowego v_s do wierzchołka v_i ,
- $\Delta C[v_i]$ – wartość przyrostu kosztu, o jaki jest pomniejszany koszt dojazdu przy wyznaczaniu wartości zastępczej funkcji kryterialnej F_Z .

$$\Delta C[v_i] = \sum_{j=0}^{i-1} C_{j,j+1} \quad (13)$$

Przy wyznaczaniu wartości zastępczej funkcji kryterialnej F_Z dla wierzchołka v_i , koszt przejazdu jest pomniejszany o wartość przyrostu kosztu $\Delta C[v_i]$. Wartość ta pomniejsza koszt przejazdu w przypadku przesiadek oraz przejazdu pojazdem linii pospiesznej, w sytuacji kiedy przesiadka nie jest konieczna lub istnieje możliwość przejazdu pojazdem linii zwykłej, zamiast pospiesznej. Przyrost kosztu nie przekracza minimalnego kosztu dojazdu do wierzchołka v_i i jest określony zależnością (13), gdzie $C_{j,j+1}$ jest równe:

- 0 – jeżeli nie jest dokonywana przesiadka na przystanku reprezentowanym przez wierzchołek v_j ,
- kosztowi przejazdu z v_j do v_{j+1} , jeżeli w v_j jest dokonywana przesiadka i nie jest on przystankiem końcowym w trasie linii, pojazdem której dojechano do v_j ,
- połowie kosztu przejazdu z v_j do v_{j+1} , jeżeli w v_j jest dokonywana przesiadka i jest on przystankiem końcowym w trasie linii, pojazdem której dojechano do v_j , a do v_{j+1} przejazd odbywa się pojazdem linii pospiesznej,
- połowie kosztu przejazdu z v_j do v_{j+1} , jeżeli v_j jest wierzchołkiem reprezentującym przystanek początkowy podróży v_s i do v_{j+1} przejazd odbywa się pojazdem linii pospiesznej.

W części inicjalizacyjnej procedury *NowaŚcieżka* następuje wyznaczenie czasu T_i i kosztu dojazdu C_i z wierzchołka początkowego v_s do wierzchołka v_i w ścieżce p_{akt} , na podstawie której jest podejmowana próba utworzenia nowej ścieżki (wiersze 3–4). Ponadto, dla każdego wierzchołka grafu wartości $F_Z[v_i]$, $T[v_i]$, $C[v_i]$, $\Delta C[v_i]$ są inicjowane wartościami początkowymi (wiersze 5–9), a wszystkie wierzchołki są wstawiane do kolejki priorytetowej Q

(wiersz 10). Wierzchołki w kolejce są uporządkowane niemalejąco według wartości zastępczej funkcji kryterialnej F_Z .

Wyznaczenie nowej ścieżki z wierzchołka v_i do wierzchołka końcowego v_e , o minimalnej wartości F_Z i nieprzekraczającej wartości F_{\max} , odbywa się w iteracji **while** (wiersze 12–28). W każdej iteracji jest pobierany z kolejki wierzchołek v_{\min} o minimalnej wartości F_Z (wiersz 13)³. Jeżeli pobranym wierzchołkiem jest wierzchołek końcowy v_e , to następuje zakończenie obliczeń przez nadanie znacznikowi *koniec* wartości **true** (wiersz 15). Zakończenie obliczeń następuje także w sytuacji, kiedy nie istnieje wierzchołek, dla którego wyznaczono ścieżkę, co jest sygnalizowane wartością $F_Z[v_{\min}] = \infty$.

Jeżeli istnieje wierzchołek, dla którego wyznaczono ścieżkę i nie jest nim wierzchołek końcowy v_e , to następuje próba poprawienia wartości F_Z dla wszystkich wierzchołków v_j sąsiadujących z v_{\min} i znajdujących się w kolejce Q , co się odbywa przez analizę wszystkich łuków wychodzących z wierzchołka v_{\min} (wiersze 17–26). W tym celu jest wyznaczany czas T_j i koszt dojazdu C_j do wierzchołka v_j przez wierzchołek v_{\min} (wiersze 18–19). Wyznaczany jest także zgodnie z zależnością (13) przyrost kosztu ΔC_j (wiersz 20) oraz wartość zastępczej funkcji kryterialnej F_{Z_j} (wiersz 21). Przy wyznaczaniu wartości zastępczej funkcji kryterialnej F_{Z_j} koszt przejazdu C_j jest pomniejszany o wartość ΔC_j . Jeżeli wyznaczona wartość zastępczej funkcji kryterialnej F_{Z_j} jest mniejsza od dotychczas wyznaczonej wartości $F_Z[v_j]$ dla wierzchołka v_j i nie przekracza wartości F_{\max} , to następuje zapamiętanie wyznaczonej ścieżki do wierzchołka v_j przez wierzchołek v_{\min} (wiersze 23–24). Zapamiętywany jest łuk (v_{\min}, v_j) wchodzący do wierzchołka v_j należący do ścieżki (wiersz 23) oraz czas T_j i koszt dojazdu C_j , przyrost kosztu ΔC_j , a także wartość zastępczej funkcji kryterialnej F_{Z_j} (wiersz 24).

Procedura zwraca jako wynik parę wartości: wyznaczoną ścieżkę oraz kod wykonanej operacji równy `NOWA_ŚCIEŻKA` (wiersze 32–33). W przypadku kiedy nie jest możliwe wyznaczenie nowej ścieżki, zwracana jest pusta ścieżka i kod operacji `BRAK_ŚCIEŻKI` (wiersz 30). Brak możliwości wyznaczenia nowej ścieżki oznacza, że zostały już wcześniej wyznaczone wszystkie ścieżki z wierzchołka v_i do wierzchołka końcowego v_e lub nie istnieje ścieżka, dla której wartość zastępczej funkcji kryterialnej F_Z nie przekracza wartości F_{\max} .

6. Wyniki badań eksperymentalnych

Badania eksperymentalne zostały przeprowadzone z użyciem komputera PC z procesorem Pentium IV 2.4 GHz dla sieci komunikacyjnej składającej się z 1211 przystanków. Sieć jest

³ Jeżeli istnieje wiele takich wierzchołków, to pobierany jest dowolny z nich.

podzielona na 26 stref i kursują w niej pojazdy 500 linii komunikacyjnych. Maksymalna długość trasy linii komunikacyjnej jest równa 29, a minimalna długość jest równa 6⁴. Graf G opisujący sieć komunikacyjną zawiera 1211 wierzchołków, 5549 łuków i ma następujące parametry:

- maksymalny stopień wejściowy i wyjściowy wierzchołka są równe 13,
- maksymalny stopień wierzchołka jest równy 26,
- minimalny stopień wejściowy i wyjściowy są równe 1,
- minimalny stopień wierzchołka jest równy 2.

Celem przeprowadzonych badań był pomiar czasu wyznaczania połączeń komunikacyjnych z użyciem opracowanego algorytmu SSP. Na czas wyznaczania połączeń ma wpływ wielkość przeszukiwanego obszaru przestrzeni rozwiązań, a co za tym idzie liczba wszystkich wyznaczonych połączeń, także będących rozwiązaniami zdominowanymi. Stąd celem badań był także pomiar liczby wyznaczonych połączeń komunikacyjnych. W celu porównania uzyskanych wyników dodatkowo przeprowadzono badania z użyciem algorytmów DFSB, BFS i MKSP, które zostały omówione w pracach [31, 32, 34].

Tabela 3

Wyniki badań eksperymentalnych z użyciem algorytmu SSP

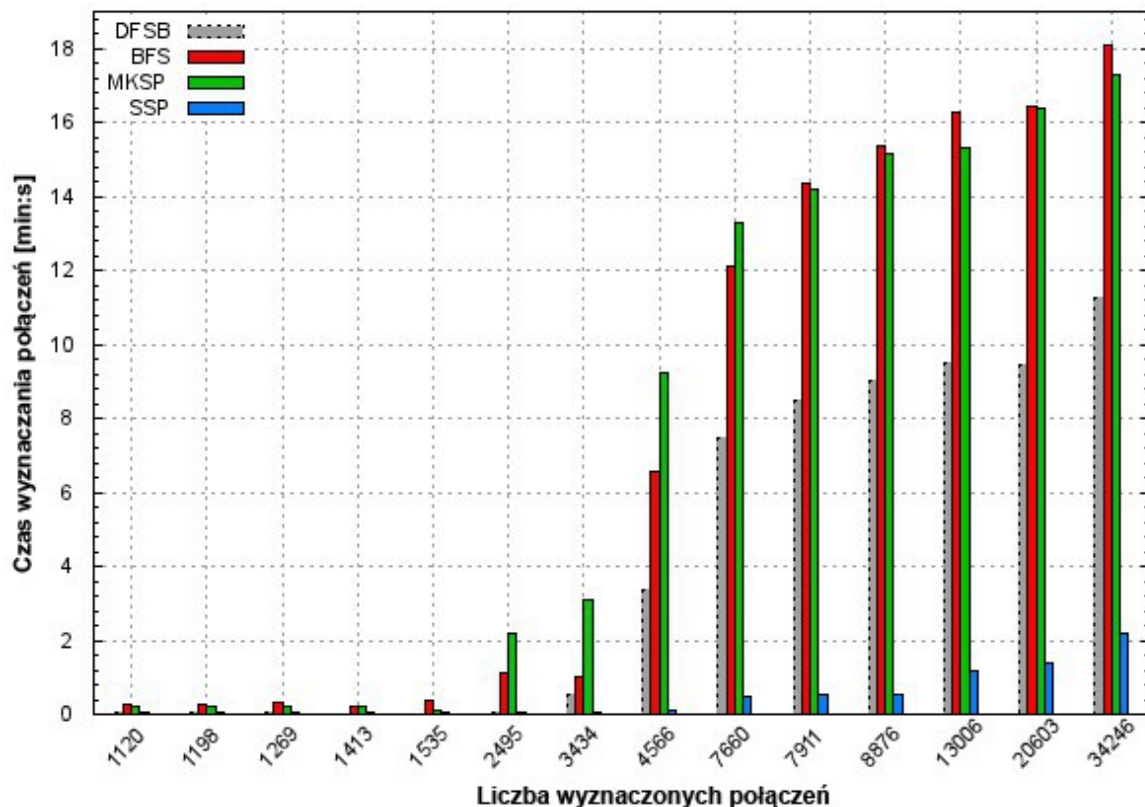
Przystanek		Długość trasy		Liczba wyznaczonych połączeń		Czas wyznaczania połączeń [min:s]
początkowy	końcowy	min.	maks.	niezdominowane	wszystkie	
270	398	32	49	60	8876	0:55
360	485	25	40	49	1413	0:04
364	219	22	34	40	1269	0:04
394	278	33	58	33	13006	1:19
415	475	27	34	13	3434	0:07
445	480	22	28	25	4566	0:12
440	475	23	36	11	1535	0:03
475	570	21	30	19	2495	0:07
545	685	34	45	8	20603	1:38
650	780	34	58	8	34246	2:18
925	1080	29	41	41	7660	0:50
930	1055	32	40	16	7911	0:52
945	1055	26	36	11	1122	0:04
1209	1104	19	24	15	1198	0:03

⁴ Długość trasy linii komunikacyjnej jest mierzona liczbą przystanków należących do trasy.

Tabela 4

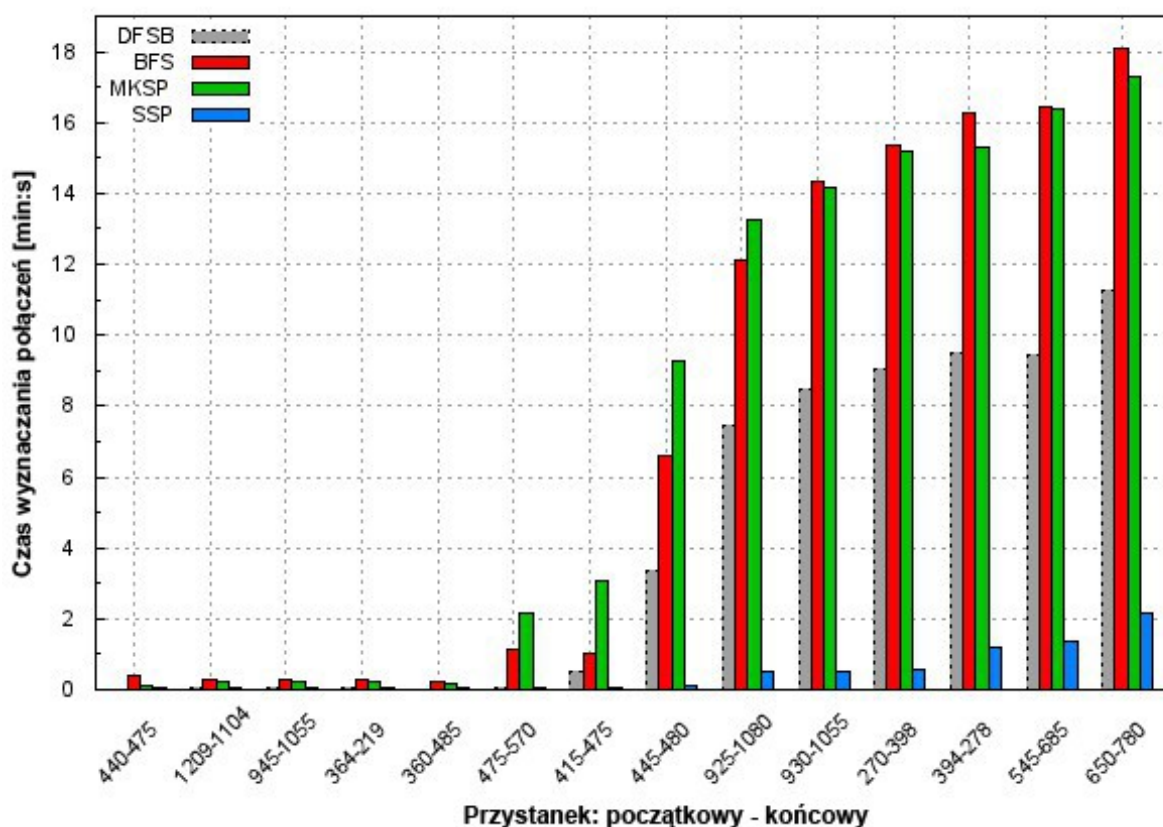
Wyniki badań eksperymentalnych z użyciem algorytmów *DFSB*, *BFS*, *MKSP*

Przystanek		Liczba wszystkich wyznaczonych połączeń			Czas wyznaczania połączeń [min:s]		
początkowy	końcowy	<i>DFSB</i>	<i>BFS</i>	<i>MKSP</i>	<i>DFSB</i>	<i>BFS</i>	<i>MKSP</i>
270	398	94638	150911	150673	9:03	15:37	15:18
360	485	1172	5229	4893	0:02	0:24	0:19
364	219	1242	5472	4963	0:04	0:30	0:21
394	278	95008	151437	150813	9:49	16:29	15:32
415	475	12984	12974	31836	0:51	1:00	3:10
445	480	47201	62623	96421	3:35	6:59	9:25
440	475	1042	5611	4528	0:01	0:38	0:13
475	570	1405	9736	18673	0:05	1:11	2:19
545	685	96497	151694	151613	9:46	16:42	16:38
650	780	130482	155015	153007	11:28	18:08	17:29
925	1080	85935	139935	141735	7:47	12:14	13:28
930	1055	93692	143173	142677	8:47	14:35	14:18
945	1055	1592	5286	4994	0:06	0:28	0:24
1209	1104	1207	5329	4947	0:03	0:29	0:21

Rys. 6. Zależność czasu wyznaczania połączeń od liczby wyznaczonych połączeń za pomocą algorytmu *SSP*Fig. 6. A computation time of routes depending on a number of computed routes using the *SSP* algorithm

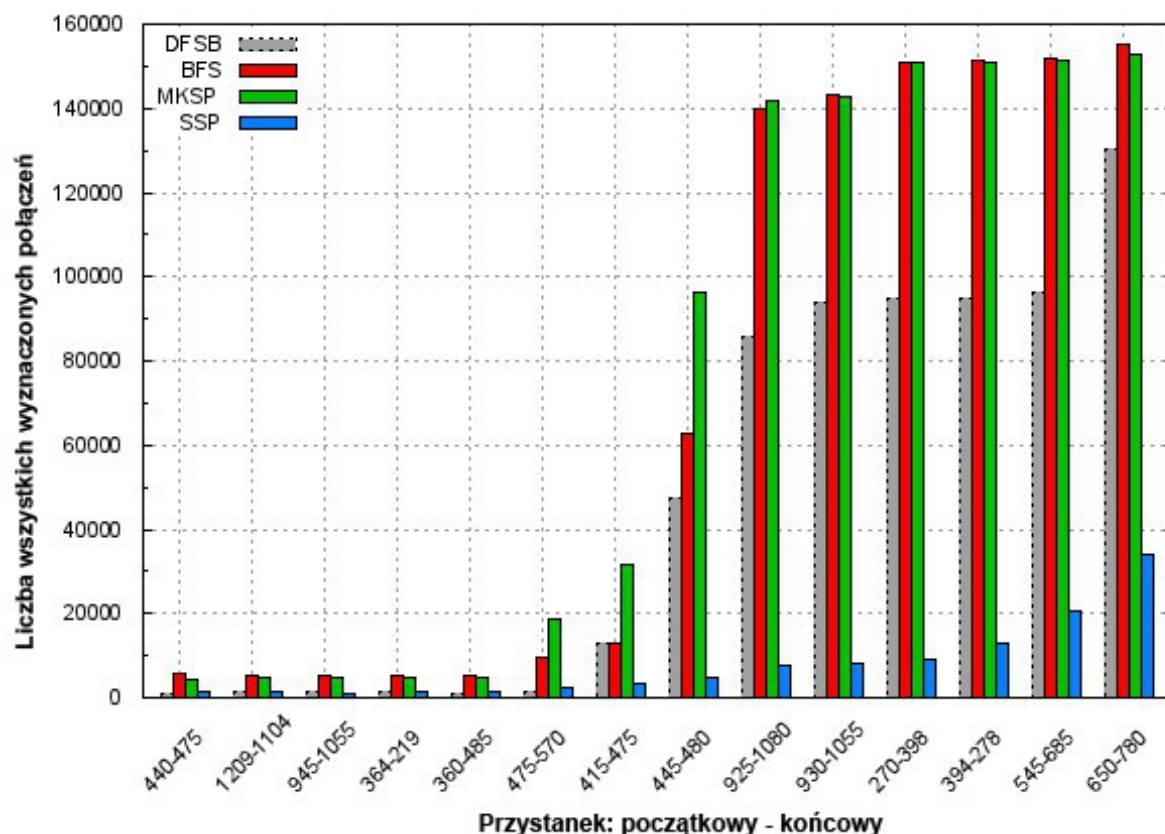
W tabeli 3 przedstawiono wyniki przeprowadzonych badań dla algorytmu *SSP*, który został opisany w rozdziale 5. Dla danych par przystanków początkowego i końcowego przedstawiono liczbę wszystkich wyznaczonych połączeń komunikacyjnych, w tym połączeń będących rozwiązaniami niezdominowanymi. Przedstawiono także czas wyznaczania połączeń oraz maksymalną i minimalną długość trasy wyznaczonego połączenia będącego rozwiązaniem niezdominowanym. Liczba wszystkich wyznaczonych połączeń uwzględnia wyznaczone połączenia końcowe, tj. połączenia z przystanku początkowego do przystanku końcowego, a także wyznaczone połączenia częściowe. Przez połączenie częściowe rozumie się każde wyznaczone rozwiązanie w procedurze *NowaŚcieżka* (wiersze 23–24).

Wyniki przeprowadzonych badań z użyciem algorytmów *DFSB*, *BFS* i *MKSP* zostały przedstawione w tabeli 4, gdzie przedstawiono czas wyznaczania połączeń oraz liczbę wszystkich wyznaczonych połączeń. Dodatkowo wyniki przeprowadzonych badań zostały przedstawione na wykresach (rys. 6–8). Czas wyznaczania połączeń z użyciem algorytmów *DFSB*, *BFS*, *MKSP* i *SSP* przedstawia wykres na rys. 7, a wykres z rys. 8 przedstawia liczbę wszystkich wyznaczonych połączeń. Dodatkowo na rys. 6 przedstawiono wykres zależności czasu wyznaczania połączeń od liczby wszystkich wyznaczonych połączeń z użyciem algorytmu *SSP*. W celu porównania algorytmów zamieszczono także na wykresie czas wyznaczania połączeń za pomocą pozostałych badanych algorytmów.



Rys. 7. Czas wyznaczania połączeń z użyciem algorytmów *DFSB*, *BFS*, *MKSP*, *SSP*
 Fig. 7. A computation time of routes using the *DFSB*, *BFS*, *MKSP*, *SSP* algorithms

Najkrótszy czas wyznaczania rozwiązań, tj. połączeń komunikacyjnych, został uzyskany dla algorytmu *SSP*, co wynika ze sposobu wyznaczania rozwiązań i przeszukiwania przestrzeni rozwiązań. Jak wspomniano w rozdziale 3, zastosowanie metody skalaryzacji nie gwarantuje wyznaczenia wszystkich rozwiązań niezdominowanych. Jak wykazano w pracach [5, 25, 30], niektóre rozwiązania nie mogą być wyznaczone z użyciem tej metody. Z tego powodu w algorytmie *SSP* jest przeszukiwany najmniejszy obszar przestrzeni rozwiązań, dzięki czemu osiągnięto najkrótszy czas ich wyznaczania. W najdłużej trwającym eksperymencie, jakim było wyznaczenie rozwiązań między przystankami 650 i 780, czas ten jest ok. 5-krotnie mniejszy niż dla algorytmu *DFSB* i ok. 9-krotnie mniejszy niż dla algorytmów *MKSP* i *BFS*. Przeszukanie najmniejszego obszaru przestrzeni rozwiązań potwierdza liczba wszystkich wyznaczonych rozwiązań, która jest najmniejsza dla algorytmu *SSP*. Najwięcej rozwiązań wyznaczono dla pary przystanków 650 i 780. Liczba wyznaczonych rozwiązań z użyciem algorytmu *SSP* jest 4-krotnie mniejsza niż w przypadku algorytmu *DFSB* i 5-krotnie mniejsza niż w przypadku algorytmów *MKSP* i *BFS*.



Rys. 8. Liczba wszystkich wyznaczonych połączeń z użyciem algorytmów *DFSB*, *BFS*, *MKSP*, *SSP*
 Fig. 8. A number of all computed routes using the *DFSB*, *BFS*, *MKSP*, *SSP* algorithms

7. Podsumowanie

W niniejszej pracy został przedstawiony problem wyznaczania połączeń w sieciach komunikacyjnych. Celem badanego problemu jest, dla zadanej pary przystanków początkowego i końcowego oraz godziny rozpoczęcia podróży, wyznaczenie połączeń komunikacyjnych minimalizując jednocześnie czas i koszt przejazdu. Rozwiązanie niniejszego problemu sprowadza się do rozwiązania dwukryterialnego problemu wyznaczania najkrótszej ścieżki w grafie o zmiennych wagach. Problem ten należy do grupy zadań optymalizacji wielokryterialnej, a jego rozwiązaniem w ogólnym przypadku jest zbiór rozwiązań, zwany zbiorem rozwiązań niezdominowanych. Kryteriami jakości, na podstawie których dokonywana jest ocena rozwiązań, tj. połączeń komunikacyjnych, są czas i koszt przejazdu.

W pracy zaproponowano algorytm *SSP*, w którym połączenia komunikacyjne należące do zbioru rozwiązań niezdominowanych są wyznaczone z użyciem metody skalaryzacji. W metodzie tej poszczególnym kryteriom jakości są przypisywane wagi określające wpływ danego kryterium na wyznaczone rozwiązanie i jest wprowadzana zastępcza funkcja kryterialna F_Z , która następnie jest poddawana optymalizacji. Dzięki wprowadzeniu zastępczej funkcji kryterialnej zadanie jest zredukowane do zadania optymalizacji jednokryterialnej i jest możliwość zastosowania mniej złożonych metod rozwiązywania problemu. Jednak w ogólnym przypadku nie ma gwarancji wyznaczenia wszystkich rozwiązań niezdominowanych. Przeprowadzając wielokrotnie proces optymalizacji dla różnych wag, istnieje możliwość uzyskania różnych rozwiązań niezdominowanych.

W algorytmie *SSP* jest stosowana zastępcza funkcja kryterialna F_Z określona zależnością (8), w której czas i koszt przejazdu są odpowiednio uwzględniane z wagami λ_T i λ_C określonymi zależnościami (10) i (11). Ich wartości są wyznaczone na podstawie połączeń o minimalnym czasie i koszcie przejazdu według równań (9) i nie są zmieniane podczas wyznaczania rozwiązań dla danej pary przystanków. Nowe rozwiązania są wyznaczone na podstawie dotychczas wyznaczonych rozwiązań poprzez analizę kolejnych wierzchołków należących do aktualnie analizowanego rozwiązania i wyznaczenie ścieżki z analizowanego wierzchołka do wierzchołka reprezentującego przystanek końcowy. Podczas wyznaczania ścieżek jest modyfikowany graf reprezentujący sieć komunikacyjną, co zapobiega wielokrotnemu wyznaczaniu tych samych rozwiązań. Rozwiązaniem startowym, od którego rozpoczyna się wyznaczanie kolejnych rozwiązań, jest rozwiązanie o minimalnym czasie przejazdu, a do wyznaczania ścieżek został użyty zmodyfikowany algorytm Dijkstry.

Dla opracowanego algorytmu *SSP* zostały przeprowadzone badania eksperymentalne. W celu porównania czasu wyznaczania rozwiązań z użyciem algorytmu *SSP* przeprowadzono dodatkowe badania z użyciem algorytmów *DFSB*, *BFS*, *MKSP*, które zostały opisane

w pracach [31, 32, 34]. Najmniejszy czas wyznaczenia rozwiązań został uzyskany dla algorytmu *SSP*, co wynika ze sposobu przeszukiwania przestrzeni rozwiązań. Metoda skalaryzacji nie gwarantuje wyznaczenia wszystkich rozwiązań niezdominowanych, dlatego niektóre obszary przestrzeni rozwiązań nie są przeszukiwane. W przypadku pozostałych trzech algorytmów są wyznaczane wszystkie rozwiązanie niezdominowane, a co za tym idzie jest przeszukiwany większy obszar przestrzeni rozwiązań niż w algorytmie *SSP*. Warto jednak podkreślić, że w algorytmie *SSP* sposób przeszukiwania przestrzeni rozwiązań gwarantuje wyznaczenie rozwiązań optymalnych ze względu na każde z kryteriów, tj. rozwiązań o minimalnym czasie i koszcie przejazdu.

BIBLIOGRAFIA

1. Brumbaugh–Smith J., Shier D.: An empirical investigation of some bicriterion shortest path algorithms. *European Journal of Operational Research*, Vol. 43, No. 2, 1989, s. 216÷224.
2. Climaco J. C., Martins E. Q. V.: A bicriterion shortest path algorithm. *European Journal of Operational Research*, Vol. 11, No. 4, 1982, s. 399÷404.
3. Coello Coello C. A., van Veldhuizen D. A., Lamont G. B.: *Evolutionary algorithms for solving multi-objective problems*. Kluwer Academic Publishers, New York 2002.
4. Corley H.W., Moon I. D.: Shortest paths in networks with vector weights. *Journal of Optimization Theory and Application*, Vol. 46, No. 1, 1985, s. 79÷86.
5. Coutinho–Rodrigues J. M., Climaco J. C., Current J. R.: An interactive bi-objective shortest path approach: searching for unsupported nondominated solutions. *Computers & Operations Research*, Vol. 26, No. 8, 1999, s. 789÷798.
6. Cormen T. H., Leiserson Ch. E., Rivest R. L.: *Wprowadzenie do algorytmów*. WNT, Warszawa 2000.
7. Daellenbach H. G., De Kluyver C. A.: Note on multiple objective dynamic programming. *Journal of the Operational Research Society*, Vol. 31, No. 7, 1980, s. 591–594.
8. Dell'Olmo P., Gentili M., Scozzari A.: On finding dissimilar Pareto-optimal paths. *European Journal of Operational Research*, Vol. 162, No. 1, 2005, s. 70÷82.
9. Garey M., Johnson D.: *Computers and intractability: a guide to the theory of NP-completeness*. W. H. Freeman & Co., New York, USA, 1990.
10. Hansen P.: *Bicriterion path problems. Multiple Criteria Decision Making: Theory and Application*, Springer–Verlag, Berlin, Germany, 1980, s. 109÷127.
11. Jungnickel D.: *Graphs, networks and algorithms*. Springer, Berlin 1999.
12. Lipski W.: *Kombinatoryka dla programistów*. WNT, Warszawa 1989.

13. Machuca E., Mandow L., Pérez de la Cruz J.L.: An Evaluation of Heuristic Functions for Bicriterion Shortest Path Problems. Proceedings of the XIV Portuguese Conference on Artificial Intelligence (EPIA 2009), Universidade de Aveiro, 2009, s. 205÷216.
14. Mandow L., Pérez de la Cruz J. L.: Frontier search for Bicriterion Shortest Path Problems. Proceeding of the 2008 conference on ECAI 2008 18th European Conference on Artificial Intelligence, IOS Press, 2008, s. 480÷484.
15. Mandow L., Pérez de la Cruz J. L.: Path recovery in frontier search for multiobjective shortest path problems. Journal of Intelligent Manufacturing, Vol. 21, No. 1, 2008, s. 89÷99.
16. Martí R., González Velarde J. L., Duarte A.: Heuristics for the bi-objective path dissimilarity problem. Computers & Operations Research, Vol. 36, No. 11, 2009, s. 2905÷2912.
17. Martins E. Q. V.: On a multicriteria shortest path problem. European Journal of Operational Research, Vol. 16, No. 2, 1982, s. 236÷245.
18. 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).
19. Mote J., Murthy I., Olson D. L.: A parametric approach to solving bicriterion shortest path problems. European Journal of Operational Research, Vol. 53, No. 1, 1991, s. 81÷92.
20. Pareto V.: Course d'Economie Politique, F. Rouge, Lausanne, 1896.
21. Peschel M., Riedel C.: Polioptymalizacja. Metody podejmowania decyzji kompromisowych w zagadnieniach inżynierijno-technicznych. WNT, Warszawa 1979.
22. Raith A., Ehrgott M.: A comparison of solution strategies for biobjective shortest path problems. Journal Computers and Operations Research, Vol. 36, No. 4, 2009, s. 1299÷1331.
23. Reingold E. M., Nievergelt J., Deo N.: Algorytmy kombinatoryczne. PWN, Warszawa 1985.
24. 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.
25. Skriver A. J. V., Andersen K. A.: A classification of bicriteria shortest path (BSP) algorithms. Asia-Pacific Journal of Operational Research, Vol. 17, 2000, s. 199÷212.
26. Stadler W.: A survey of Multicriteria Optimization or the Vector Maximum Problem. Part I: 1776–1960, Journal of Optimization Theory & Application, Vol. 29, No. 1, 1979, s. 1÷52.
27. Tung C.T., Chew K.L.: A bicriterion Pareto-optimal path algorithm. Asia-Pacific Journal of Operational Research, Vol. 5, 1988, s. 166÷172.

28. Ulungu E. L., Teghem J.: Multi-objective combinatorial optimization problems: a survey. *Journal of Multi-Criteria Decision Analysis*, Vol. 3, No. 2, 1994, s. 83÷104.
29. Voorneveld M.: Characterization of Pareto dominance. *Operations Research Letters*, Vol. 31, No. 1, 2003, s. 7÷11.
30. White D. J.: The set of efficient solutions for multiple objective shortest path problems. *Computers & Operations Research*, Vol. 9, No. 2, 1982, s. 101÷107.
31. Widuch J.: Problem wyznaczania połączeń w sieciach komunikacyjnych. *Studia Informatica*, Vol. 22, No. 4(46), Gliwice 2001, s. 117÷134.
32. Widuch J.: Wyznaczanie połączeń w sieciach komunikacyjnych o zmiennych wagach. *Studia Informatica*, Vol. 23, No. 4(51), Gliwice 2002, s. 85÷104.
33. Widuch J.: Algorytmy optymalizacji wielokryterialnej w problemach komunikacyjnych. *Rozprawa doktorska, Politechnika Śląska, Gliwice 2008.*
34. Widuch J.: Rozwiązanie problemu wyznaczania połączeń w sieciach komunikacyjnych za pomocą zmodyfikowanego algorytmu wyznaczania K najkrótszych ścieżek. *Studia Informatica*, Vol. 31, No. 1(88), Gliwice 2010, s. 55÷70.

Recenzent: Dr hab. inż. Mariusz Boryczka, prof. UŚ

Wpłynęło do Redakcji 3 października 2011 r.

Abstract

The communication networks routing problem (CNRP) which is a sample of the bicriterion shortest path problem (BSP) with variable weights have been presented. Several algorithms have been proposed in the literature for solving the BSP, but these algorithms assume constant weights [1, 2, 4, 7, 8, 10, 13, 14, 15, 16, 17, 19, 22, 24, 27]. The CNRP is an example of multicriteria optimization problem (MOP), too. In the paper we present fundamental information about MOP and we present the CNRP description and analysis of the problem. 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 *SSP* 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 find the solutions using scalarization method, where the criterion functions are replaced with single weighted criterion function F_Z defined by formula (8).

The algorithm was implemented and tested for a random generated bus network. Additionally, tests were conducted for the algorithms *DFSB*, *BFS* and *MKSP* presented in [31, 32, 34]. The results of tests are presented in Tab. 3–4 and additionally, they are presented in Fig. 6–8. The shortest computation time was obtained for presented the *SSP* algorithm. On the basis of the test results we found that the *SSP* algorithm exhibits a reasonable execution time for the network of the size of 1200 stops. We take into consideration only two criterion functions which are the time and the cost of travel but it is possible to add other criterion functions.

Adres

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