

Wojciech Sikora

Michał Urbaniak

Uniwersytet Ekonomiczny w Poznaniu

SPRAWNOŚĆ METOD OGÓLNYCH I WYSPECJALIZOWANYCH W ANALIZIE CZASOWO-KOSZTOWEJ PRZEDSIĘWZIĘĆ

Wprowadzenie

W pracy rozpatrujemy klasyczny problem analizy czasowo-kosztowej przedsięwzięć polegający na minimalizacji kosztów przy zadanym (dyrektywnym) terminie jego zakończenia. O ile w analizie czasowej przedsięwzięć czas trwania poszczególnych czynności jest dany, to w analizie czasowo-kosztowej czas ten jest zmienną decyzyjną, mieszczącą się w określonym przedziale czasowym i generującą różne koszty wykonania.

Zagadnienia optymalizacyjne związane z analizą czasowo-kosztową można rozwiązywać dwojako:

- 1) sformułować ich modele w postaci zadań programowania liniowego (PL) lub programowania dyskretnego, liniowego (PDL), które rozwiązujemy korzystając z ogólnych metod,
- 2) sformułować specjalne, efektywne algorytmy wykorzystujące sieciowy opis przedsięwzięcia.

W drugim przypadku algorytmy specjalne dzielimy na dwie klasy¹:

- 1) dokładne, dosyć skomplikowane i czasochłonne, gwarantujące uzyskanie rozwiązania optymalnego,
- 2) przybliżone, o prostej postaci, dające szybko rozwiązanie suboptymalne.

W artykule przedstawimy dwa modele optymalizacyjne rozważanego problemu dla liniowej i punktowej reprezentacji krzywej kosztów. Model w postaci zadania PL wraz z algorytmem simpleks tworzą pierwszą, ogólną metodę roz-

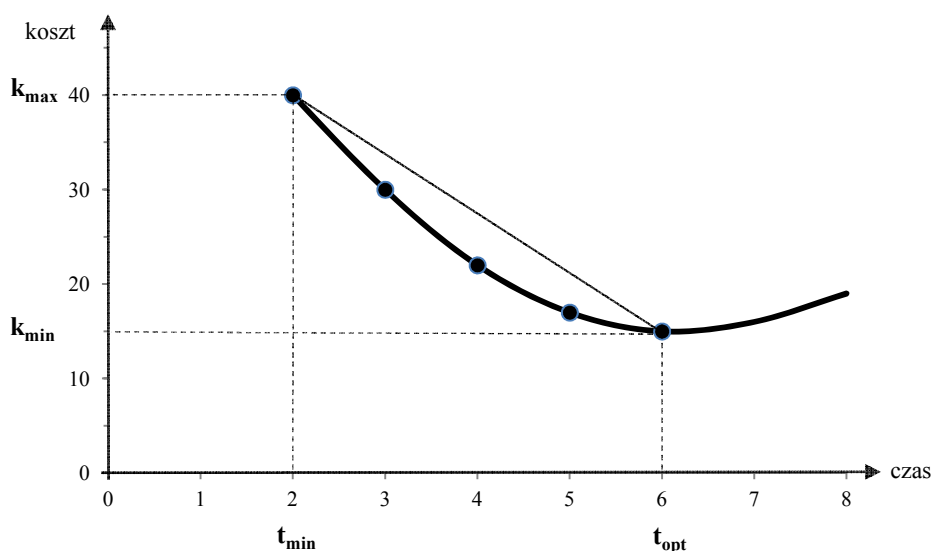
¹ H. Gaspar: Propozycja nowego algorytmu w analizie czasowo-kosztowej przedsięwzięć. „Badania Operacyjne i Decyzje” 2006, nr 3-4.

wiązywania problemu. W zależności, czy punktem wyjścia jest najkrótszy bądź najdłuższy czas wykonywania czynności otrzymujemy dwa podejścia polegające na wydłużaniu bądź skracaniu poszczególnych czynności. Reprezentantem pierwszego podejścia jest nowy algorytm wydłużania W1, sformułowany w artykule Sikory², przedstawicielem natomiast drugiego podejścia jest zmodyfikowany algorytm skracania czynności krytycznej Kaufmanna i Desbazeille'a³.

W pracy przedstawiono obydwa algorytmy oraz porównano sprawność wszystkich trzech metod.

1. Analiza czasowo-kosztowa

W analizie czasowo-kosztowej czas trwania czynności nie jest parametrem, ale zmienną decyzyjną przyjmującą wartości z zadanego przedziału czasu. Zależność kosztu wykonania czynności od czasu jej realizacji można przedstawić w postaci tzw. krzywej kosztu – na rys. 1.



Rys. 1. Krzywa kosztu

² W. Sikora: Metoda wydłużania czynności w analizie czasowo-kosztowej przedsięwzięć. W: Z prac katedry badań operacyjnych. Red. W. Sikora. UEP, Poznań 2012.

³ A. Kaufmann, G. Desbazeille, E. Ventura: La méthode du chemin critique. Dunod, Paris 1964

Zależność ta jest funkcją wypukłą. Interesuje nas czas od $t_{\min}$ do t_{opt} . Przedziałowi temu odpowiada przedział kosztu $[k_{\max}, k_{\min}]$. Im krótszy czas realizacji, tym większy koszt. Im dłuższy czas, tym niższy koszt, który osiąga minimum dla czasu t_{opt} , a potem wzrasta.

Krzywa kosztu w postaci ciągłej funkcji wypukłej to oczywiście pewna idealizacja zjawiska. W praktyce nie jesteśmy w stanie oszacować jej postaci analitycznej. Zakładamy, że dokonujemy na przedziale $[t_{\min}, t_{\text{opt}}]$:

- jej aproksymacji liniowej,
- aproksymacji odcinkowo-liniowej,
- ustalenia jej wartości dla wybranych czasów realizacji.

W dalszych analizach przyjmujemy dwa warianty reprezentacji krzywej kosztu:

- liniowa – znamy $k_{\max}$, $k_{\min}$, $t_{\min}$, t_{opt} i określamy średnie tempo spadku kosztu Δk ,
- punktowa – znamy wartości krzywej kosztu dla wszystkich czasów wykonania wyrażonych w liczbach całkowitych, ustalamy wówczas kolejne spadki kosztu $\Delta k_1, \Delta k_2, \Delta k_3, \dots$

Zakładamy, że wszystkie możliwe czasy realizacji czynności są liczbami całkowitymi z przedziału $[t_{\min}, t_{\text{opt}}]$.

Realizując przedsięwzięcie często mamy zadany dyrektywny czas jego ukończenia T^d bądź dyrektywny koszt jego wykonania K^d , które nie mogą być przekroczone. Powstają wówczas dwa podstawowe problemy optymalizacyjne:

1) problem I – ustal taki plan realizacji przedsięwzięcia, aby koszt realizacji był minimalny, a czas realizacji nie przekraczał czasu dyrektywnego T^d ,

2) problem II – ustal taki plan realizacji przedsięwzięcia, aby czas realizacji był minimalny, a koszt realizacji nie przekraczał kosztu dyrektywnego K^d .

Problem I symbolicznie można zapisać następująco: znajdź taki plan realizacji X , aby

$$k(X) \Rightarrow \min \quad (1)$$

$$t(X) \leq T^d \quad (2)$$

gdzie:

$k(X)$ – koszt realizacji przedsięwzięcia dla planu X ,

$t(X)$ – czas realizacji przedsięwzięcia dla planu X .

Problem II zapisujemy wówczas następująco: znajdź taki plan realizacji X , aby

$$t(X) \Rightarrow \min \quad (3)$$

$$k(X) \leq K^d \quad (4)$$

Każdy z tych problemów można sformułować w postaci zadania programowania liniowego (PL), jeżeli zakładamy liniową reprezentację krzywej kosztów bądź w postaci zadania programowania dyskretnego, liniowego (PDL), jeżeli zakładamy punktową reprezentację krzywej kosztów. Dalej zajmujemy się tylko problemem I.

2. Modele optymalizacyjne

Strukturę przedsięwzięcia możemy przedstawić w postaci grafu $G = \langle P, U \rangle$, gdzie P – zbiór węzłów reprezentujących zdarzenia, a U – zbiór łuków reprezentujących czynności. Graf ten powinien być grafem prostym, zorientowanym i acyklicznym.

Oznaczmy:

t'_{ij} – najkrótszy czas realizacji czynności $\langle i, j \rangle$,

t''_{ij} – najdłuższy czas realizacji czynności $\langle i, j \rangle$,

Δt_{ij} – maksymalnie dopuszczalne wydłużenie czasu realizacji czynności $\langle i, j \rangle$,

w_{ij} – wydłużenie czasu realizacji czynności $\langle i, j \rangle$,

t_i – najwcześniejszy moment zajścia zdarzenia i -tego,

k'_{ij} – maksymalny koszt wykonania czynności $\langle i, j \rangle$,

k''_{ij} – minimalny koszt wykonania czynności $\langle i, j \rangle$,

Δk_{ij} – średni spadek kosztu, o ile wydłużymy czynność $\langle i, j \rangle$ o jednostkę czasu,

K – łączny koszt realizacji przedsięwzięcia,

$K_{\max}$ – maksymalny łączny koszt realizacji przedsięwzięcia,

n – liczba węzłów w sieci czynności (numer węzła końcowego),

m – liczba czynności tworzących przedsięwzięcie.

Między tymi wielkościami zachodzą następujące zależności

$$\Delta t_{ij} = t''_{ij} - t'_{ij} \quad (5)$$

$$\Delta k_{ij} = \frac{k'_{ij} - k''_{ij}}{\Delta t_{ij}} \quad (6)$$

$$K = K_{max} - \sum_{\langle i,j \rangle \in U} \Delta k_{ij} w_{ij} \quad (7)$$

Problem I przy reprezentacji liniowej krzywej kosztu można zapisać następująco: znajdź takie wartości zmiennych decyzyjnych t_i oraz w_{ij} , aby

$$\sum_{\langle i,j \rangle \in U} \Delta k_{ij} w_{ij} \Rightarrow \max \quad (8)$$

$$t_1 = 0 \quad (9)$$

$$t_j - t_i \geq t'_{ij} + w_{ij}, \text{ dla } \langle i, j \rangle \in U \quad (10)$$

$$t_n \leq T^d \quad (11)$$

$$0 \leq w_{ij} \leq \Delta t_{ij}, \text{ dla } \langle i, j \rangle \in U \quad (12)$$

Zadanie (8)-(12) jest zadaniem PL. Oznaczmy je jako model MLI. Liczba zmiennych $L(z) = n + m$, a liczba warunków $L(w) = 2 + 2m$. Maksymalizacja łącznego spadku kosztu funkcji celu (8) jest równoważna minimalizacji kosztu realizacji przedsięwzięcia.

Przy reprezentacji punktowej krzywej kosztu sformułowanie modelu jest trudniejsze. Oznaczmy:

k^r_{ij} – spadek kosztu, o ile czynność $\langle i, j \rangle$ jest wydłużana o r -tą z kolei jednostkę czasu,

$$w^r_{ij} = \begin{cases} 1, & \text{jeżeli czynność } \langle i, j \rangle \text{ jest wydłużana o } r\text{-tą jednostkę czasu,} \\ 0, & \text{w przypadku przeciwnym.} \end{cases}$$

Z wypukłości krzywej kosztu wynika, że

$$k^1_{ij} > k^2_{ij} > \dots > k^{r-1}_{ij} > k^r_{ij} > \dots$$

Zakładamy także, że wszystkie parametry Δt_{ij} i t'_{ij} są liczbami całkowitymi.

Problem PI przy reprezentacji punktowej krzywej kosztu można zapisać następująco: znaleźć takie wartości zmiennych t_i oraz w_{ij}^r , aby

$$\sum_{\langle i,j \rangle \in U} \sum_{r=1}^{\Delta t_{ij}} k_{ij}^r w_{ij}^r = \max \quad (13)$$

$$t_1 = 0 \quad (14)$$

$$t_j - t_i \geq t'_{ij} + \sum_{r=1}^{\Delta t_{ij}} w_{ij}^r, \text{ dla } \langle i,j \rangle \in U \quad (15)$$

$$t_n \leq T^d \quad (16)$$

$$w_{ij}^r \in \{0, 1\}, \text{ dla } \langle i,j \rangle \in U, r = 1, \dots, \Delta t_{ij} \quad (17)$$

W zadaniu (13)-(17) zmienna w_{ij}^r może przyjąć wartość 1 tylko wówczas, gdy $w_{ij}^g = 1$ dla $g < r$. Wprowadzenie tego typu ograniczenia do modelu nie jest jednak konieczne, gdyż spełnienie tej relacji wymusza wypukłość funkcji kosztu.

Zadanie (13)-(17) jest zadaniem programowania dyskretnego, liniowego (PDL). Oznaczmy je jako model MDI. Jeżeli przyjmiemy, że średnio maksymalny czas wydłużenia czynności wynosi s , to liczba zmiennych $L(z) = n + ms$, a liczba warunków $L(w) = 2 + m$. Pomijamy w tym rachunku warunek (17) dotyczący binarności zmiennych w_{ij}^r .

Jeżeli w zadaniu (13)-(17) warunek (17) zastąpimy warunkiem słabszym

$$0 \leq w_{ij}^r \leq 1, \text{ dla } \langle i,j \rangle \in U, r = 1, \dots, \Delta t_{ij} \quad (18)$$

to dostajemy zadanie PL. Oznaczmy je jako MDLI. Liczba zmiennych $L(z) = n + ms$, a liczba warunków $L(w) = 2 + m + ms$. Zadanie MDLI łatwiej rozwiązywać niż zadanie MDI. Jego rozwiązywanie byłoby bardziej efektywne, gdybyśmy zastosowali specjalne algorytmy uwzględniające specyficzną strukturę zadania⁴. W pracy Sikory⁵ podano twierdzenie o równoważności obydwu zadań.

⁴ W. Sikora: Metoda stopniowej analizy zmiennych dla zadania programowania rozłącznego. „Przegląd Statystyczny” 1991, nr 2.

⁵ Idem: Metoda wydłużania..., op. cit.

Rozwiązywanie zadań sformułowanych w tym rozdziale algorytmem sympleks daje nam ogólną metodę analizy czasowo-kosztowej. Omawiając metody specjalne najpierw przedstawimy metodę wydłużania czynności, zapewniającą uzyskanie rozwiązań przybliżonych (suboptymalnych), którą zaproponowano w pracy Sikory⁶.

3. Idea metody wydłużania czynności

Ogólna idea tej metody sprowadza się do stopniowego wydłużania czynności tak, aby prowadziło to do jak największego spadku kosztu realizacji całego przedsięwzięcia $k(X)$.

Zakładamy, że strukturę każdego przedsięwzięcia opisuje graf $G = \langle P, U \rangle$. Dane są: czasy minimalne trwania czynności – t'_{ij} , maksymalne koszty realizacji czynności – k'_{ij} , kolejne spadki kosztu realizacji czynności Δk^r_{ij} (dla $r = 1, \dots, \Delta t_{ij}$) oraz czas dyrektywny wykonania przedsięwzięcia – T^d .

Zdefiniujemy kilka podstawowych pojęć dla metody wydłużania czynności.

Definicja 1. Siecią czasową nazywamy sieć czynności, w której poszczególnym czynnościom (łukom) przyporządkowane są ich bieżące czasy trwania.

Definicja 2. Siecią kosztową nazywamy sieć czynności, w której poszczególnym czynnościom (łukom) przyporządkowane są aktualne spadki kosztu w wyniku ich wydłużenia o jednostkę czasu.

Definicja 3. Frontem sieci kosztowej nazywamy każdy zbiór czynności (łuków) zawierający dokładnie po jednym łuku z każdej ścieżki pełnej z tej sieci.

Frontem w sieci kosztowej jest więc każdy maksymalny zbiór czynności równoczesnych, czyli takich, że żadne dwie czynności frontu nie leżą na tej samej ścieżce⁷.

Definicja 4. Kosztem frontu nazywamy aktualny spadek kosztu realizacji przedsięwzięcia w wyniku wydłużenia czynności frontu o jednostkę czasu.

Jeżeli czynności przyporządkowany jest koszt zerowy, to nie jest ona wydłużana o jednostkę czasu.

Definicja 5. Frontem o maksymalnym spadku nazywamy front o największym koszcie spośród wszystkich frontów, które można utworzyć w danej sieci.

⁶ Więcej o metodach specjalnych zobacz: H. Gaspars-Wieloch: Analiza sieciowa przedsiębiorstw. W: Badania operacyjne. Red. W. Sikora. PWE, Warszawa 2008; a o niektórych niedostatkach tych algorytmów – zobacz Eadem: Propozycja..., op. cit.

⁷ H. Gaspars-Wieloch taki zbiór nazywa przekrojem dokładnym. Nieco inną definicję frontu podano w pracy: E. Ignasiak: Programowanie sieciowe. PWE, Warszawa 1972, s. 50.

4. Algorytm W1 dla zadania MDI

W zadaniu MDI dany jest czas dyrektywny T^d realizacji przedsięwzięcia, który nie może być przekroczony. W algorytmie W1, w każdej iteracji generujemy bieżącą sieć czasową (ST^p) oraz bieżącą sieć kosztową (SK^p).

W sieci czasowej, na podstawie aktualnych czasów wykonania t_{ij} , ustalamy: najwcześniejsze momenty t_j , najpóźniejsze momenty t'_j oraz całkowite zapasy czasu z_{ij}^c .

Najwcześniejsze momenty liczymy zgodnie z wzorem

$$t_j = \begin{cases} 0, & \text{dla } j = 1 \\ \max_{i \in P_j} \{t_i + t_{ij}\}, & \text{dla } j = 2, \dots, n \end{cases} \quad (19)$$

a najpóźniejsze momenty

$$t'_i = \begin{cases} T^d, & \text{dla } i = n \\ \min_{j \in N_i} \{t'_j - t_{ij}\}, & \text{dla } i = 1, \dots, n-1 \end{cases} \quad (20)$$

gdzie:

- t_j – najwcześniejszy moment zajścia zdarzenia j-tego,
- t'_i – najpóźniejszy moment zajścia zdarzenia i-tego,
- t_{ij} – aktualny czas trwania czynności $\langle i, j \rangle$,
- P_j – zbiór poprzedników (węzłów) zdarzenia j-tego,
- N_i – zbiór następników (węzłów) zdarzenia i-tego.

Mając ustalone momenty najwcześniejsze i najpóźniejsze możemy obliczyć dla poszczególnych czynności całkowite zapasy czasu

$$z_{ij}^c = t'_j - t_i - t_{ij}, \text{ dla } \langle i, j \rangle \in U \quad (21)$$

Czynności, dla których zapasy całkowite są zerowe nazywamy czynnościami krytycznymi, a ścieżkę składającą się z nich – ścieżką krytyczną. Pozostałe czynności o dodatnich zapasach całkowitych nazywamy niekrytycznymi.

Jeżeli $t_n < T^d$, to aktualna sieć czasowa nie ma żadnej czynności krytycznej, czyli takiej, dla której zapas całkowity jest zerowy.

W sieci kosztowej koszt bieżący dla czynności $\langle i, j \rangle$ wynosi

$$k_{ij} = \begin{cases} 0, \text{ jeżeli czynność } \langle i, j \rangle \text{ jest krytyczna,} \\ 0, \text{ jeżeli czynność } \langle i, j \rangle \text{ jest niekrytyczna, niewydłużalna,} \\ k_{ij}^r, \text{ jeżeli czynność } \langle i, j \rangle \text{ jest niekrytyczna, wydłużalna,} \\ \text{o aktualnym czasie trwania } t_{ij}' + r - 1 \end{cases} \quad (22)$$

W sieci czasowej ST^{p+1} wydłużamy o jednostkę czasu realizacji tych czynności, które należą do frontu o maksymalnym spadku i mają dodatnie koszty.

Kończymy procedurę wydłużenia czynności, jeżeli bieżąca sieć kosztów ma tylko zerowe koszty wydłużenia czynności.

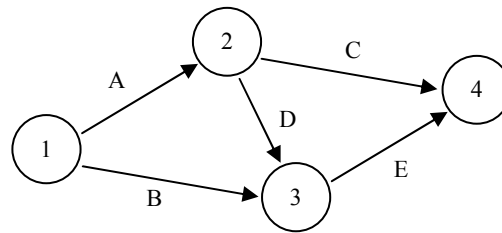
Jeżeli dla pierwszej sieci czasowej ST^1 czas realizacji przedsięwzięcia $T^* = t_n' > T^d$, to przedsięwzięcie jest niewykonalne w czasie dyrektywnym T^d .

Algorytm W1 sprowadza się do następujących kroków:

- K1. Utwórz początkową sieć czasową ST^1 , w której $t_{ij} = t_{ij}'$, oblicz momenty najwcześniejsze t_j , najpóźniejsze t_j' , całkowite zapasy czasu z_{ij}^c . Utwórz początkową sieć kosztową SK^1 , zgodnie z wzorem (22), podaj aktualny czas i koszt realizacji przedsięwzięcia.
- K2. Sprawdź, czy aktualna sieć kosztowa SK^p jest zerowa. Jeżeli tak – idź do kroku K5. Jeżeli nie – idź do kroku K3.
- K3. W sieci kosztowej SK^p ustal front o maksymalnym spadku.
- K4. W następnej sieci czasowej ST^{p+1} wydłuż o jednostkę czas wykonania czynności, które miały w ostatniej sieci SK^p dodatnie spadki kosztu. Utwórz nową macierz kosztów SK^{p+1} . Ustal bieżący czas i koszt realizacji przedsięwzięcia i przejdź do kroku K2.
- K5. Aktualny plan realizacji jest planem końcowym (koniec procedury wydłużania czynności).

Przykład 1. Rozważmy przedsięwzięcie składające się z pięciu czynności A, B, C, D, E. Strukturę przedsięwzięcia przedstawia rys. 2. W tabeli 1 podano charakterystykę czasowo-kosztową wszystkich czynności składających się na przedsięwzięcie. Dyrektywny czas realizacji $T^d = 12$.

Należy ustalić taki plan realizacji przedsięwzięcia, aby wykonać je w czasie nie dłuższym od czasu dyrektywnego, a koszt realizacji był minimalny.



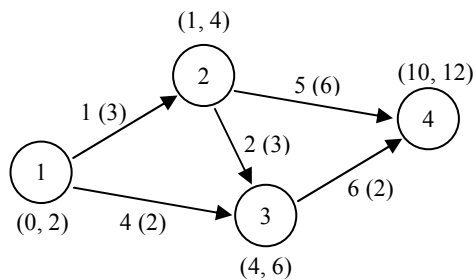
Rys. 2. Struktura przedsięwzięcia

Tabela 1

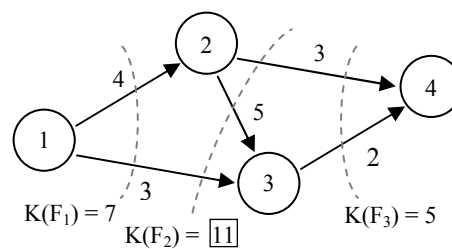
Charakterystyka czasowo-kosztowa przedsięwzięcia

Czynności	$\langle 1, 2 \rangle$	$\langle 1, 3 \rangle$	$\langle 2, 4 \rangle$	$\langle 2, 3 \rangle$	$\langle 3, 4 \rangle$
t'_{ij}	1	4	5	2	6
k'_{ij}	18	22	20	19	11
Δk^1_{ij}	4	3	3	5	2
Δk^2_{ij}	3	2	2	4	–

Do rozwiązania tego zadania stosujemy algorytm W1. Początkową sieć czasową zawiera rys. 3. Dla planu realizacji P1 minimalny czas realizacji $T1 = 10$, a maksymalny koszt realizacji $K1 = 90$. W sieci kosztowej SK1 podane są spadki kosztów, jeżeli wydłużamy czas realizacji poszczególnych czynności o 1 jednostkę. W sieci tej mamy 3 fronty: $F_1 = \{\langle 1, 2 \rangle, \langle 1, 3 \rangle\}$, $F_2 = \{\langle 2, 4 \rangle, \langle 2, 3 \rangle, \langle 1, 3 \rangle\}$, $F_3 = \{\langle 2, 4 \rangle, \langle 3, 4 \rangle\}$ ze spadkami kosztu odpowiednio: 7, 11, 5.



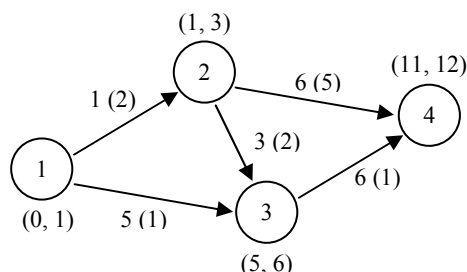
Rys. 3. Sieć czasowa ST1



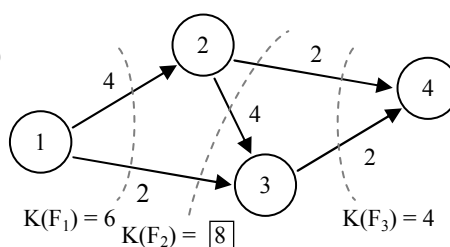
Rys. 4. Sieć kosztowa SK1

P1: $T1 = 10, K1 = 90$

Ponieważ front F_2 jest frontem o maksymalnym spadku, więc czynności $\langle 2, 4 \rangle$, $\langle 2, 3 \rangle$ i $\langle 1, 3 \rangle$ wydłużamy o jednostkę. Nowy plan realizacji zawiera rys. 5.



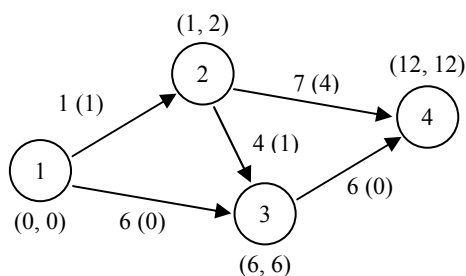
Rys. 5. Sieć czasowa ST2



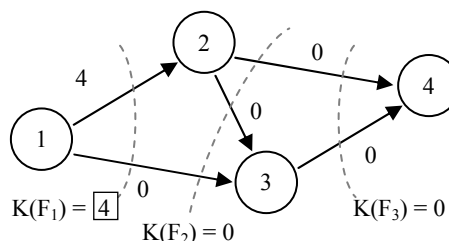
Rys. 6. Sieć kosztowa SK2

$$P2: T2 = 11, K2 = 90 - 11 = 79$$

Plan przedstawiony na rys. 5 ma czas realizacji $T2 = 11$, koszt realizacji 79. W sieci SK2 szukamy ponownie frontu o maksymalnym spadku, jest nim front F_2 . Wydłużenie czynności go tworzących obniża aktualny koszt realizacji o 8.



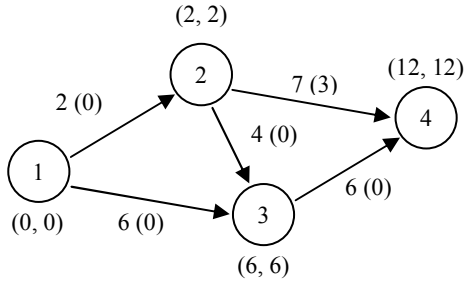
Rys. 7. Sieć czasowa ST3



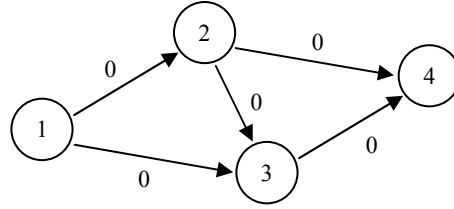
Rys. 8. Sieć kosztowa SK3

$$P3: T3 = 12, K3 = 79 - 8 = 71$$

Plan $P3$ przedstawiony na rys. 7 ma czas realizacji $T3 = 12$ równy czasowi dyrektywnemu T^d . Czynności $\langle 1, 3 \rangle$ i $\langle 3, 4 \rangle$ są krytycznymi, a czynności $\langle 2, 3 \rangle$ i $\langle 2, 4 \rangle$ są już niewydłużalne, a więc zgodnie ze wzorem (22) jedyny dodatni spadek kosztu ma czynność niekrytyczna $\langle 1, 2 \rangle$ ze spadkiem 4. Ta czynność zostanie wydłużona o 1 jednostkę.



Rys. 9. Sieć czasowa ST4



Rys. 10. Sieć kosztowa SK4

$$P4: T4 = 12, K2 = 71 - 4 = 67$$

Plan P4 jest planem końcowym, gdyż odpowiadająca mu na rys. 10 sieć kosztowa jest zerowa. Nie ma już żadnej czynności, którą można wydłużyć nie przekraczając czasu dyrektywnego $T^d = 12$.

W pracy Sikora⁸ do szukania frontu o maksymalnym spadku zaproponowany został specjalny algorytm. Jest on analogiem algorytmu Forda-Fulkersona szukania maksymalnego przepływu w sieci⁹. W naszym przypadku, aby wyznaczyć front o maksymalnym spadku, szukamy minimalnego przepływu w sieci z dolnymi ograniczeniami¹⁰.

Algorytm W1 można stosować także do rozwiązywania problemu z liniową reprezentacją krzywej kosztów, w tym przypadku zamiast wydłużać czas trwania czynności tworzących front o jednostkę, można je wydłużać o więcej jednostek. Aby to ustalić, dla każdej czynności tworzącej front o dodatnim koszcie bieżącym wyznaczamy maksymalne aktualne jej wydłużenie,

$$w_{ij} = \min\{t_{ij}'' - t_{ij}, z_{ij}^c\} \quad (23)$$

a następnie obliczamy wielkość wydłużenia czynności frontu, zgodnie z wzorem

$$w = \min\{w_{ij} \mid \langle i, j \rangle \in \bar{U}^p\} \quad (24)$$

gdzie:

$\bar{U}^p$ – zbiór czynności tworzących front, o dodatnich kosztach bieżących.

⁸ W. Sikora: Metoda wydłużania..., op. cit.

⁹ L.R. Ford, D.R. Fulkerson: Przepływy w sieciach. PWN, Warszawa 1969; M. Anholcer: Przepływy w sieciach. W: Badania operacyjne. Op. cit.

¹⁰ W. Sikora: Metoda wydłużania..., op. cit., s. 125-128.

5. Zmodyfikowany algorytm Kaufmanna i Desbazeille'a (ZKD)

Klasyczny algorytm Kaufmanna i Desbazeille'a (KD) polega na stopniowym skracaniu czynności krytycznych. Dla danej sieci $S = \langle P, U \rangle$, dla każdej czynności $\langle i, j \rangle$ znamy: t_{ij}'' – jej najdłuższy czas realizacji oraz k_{ij}^r – wzrost kosztu realizacji, jeżeli skracamy ją o r -tą z kolei jednostkę czasu.

W każdej iteracji, podobnie jak w algorytmie W1, generujemy bieżącą sieć czasową (ST^P) oraz bieżącą sieć kosztową (SK^P).

W sieci czasowej, na podstawie aktualnych czasów realizacji, ustalamy najwcześniejsze momenty zajścia zdarzeń t_j , najpóźniejsze momenty zajścia zdarzeń t_j' oraz całkowite zapasy czasu z_{ij}^c . Przyjmujemy przy tym, że: $t_1 = 0$, $t_n' = t_n$.

Sieć kosztowa składa się tylko z czynności krytycznych, dla których aktualny, całkowity zapas czasu $z_{ij}^c = 0$. W krytycznej sieci kosztowej, koszt bieżący dla czynności $\langle i, j \rangle$ wynosi:

$$k_{ij} = \begin{cases} k_{ij}^r, & \text{jeżeli czynności } \langle i, j \rangle \text{ ma aktualny czas trwania } t_{ij}'' - r + 1 \\ \infty, & \text{jeżeli czynność } \langle i, j \rangle \text{ nie może być dalej skracana, gdyż } t_{ij} = t_{ij}' \end{cases} \quad (25)$$

W sieci kosztowej szukamy frontu o minimalnym wzroście kosztu. Wszystkie czynności tworzące taki front skracamy o jednostkę, a czas realizacji całego przedsięwzięcia skraca się także o jednostkę.

Przy komputerowej implementacji algorytmu wygodniej w sieci kosztowej szukać nie frontu o minimalnym koszcie, ale przekroju o minimalnym koszcie.

Definicja 6. Jeżeli zbiór węzłów sieci kosztowej podzielimy na dwa takie podzbiory $P1$ oraz $P2$, że węzeł „1” $\in P1$, a węzeł „n” $\in P2$, to przekrojem nazywamy każdy maksymalny zbiór łuków $\langle i, j \rangle$ takich, iż $i \in P1$, a $j \in P2$.

Przekrój może zawierać więcej niż jedną czynność z każdej ścieżki pełnej w sieci, natomiast front – tylko jedną.

Do szukania minimalnego przekroju w sieci, można wykorzystać znany algorytm Forda, Fulkersona, Edmondsa i Karpa (FFEK). Zastosowanie algorytmu FFEK powinno poprawić działanie algorytmu KD zwiększając szybkość obliczeń oraz polepszając jego dokładność¹¹.

Dalej algorytm KD z podaną poprawką nazywać będziemy zmodyfikowanym algorytmem Kaufmanna i Desbazeille'a (ZKD).

¹¹ M. Anholcer: Op. cit.

Algorytm ZKD sprowadza się do następujących kroków:

- K1. Utwórz początkową sieć czasową ST^1 , w której $t_{ij} = t_{ij}''$. Oblicz najwcześniejsze momenty t_j , najpóźniejsze momenty t_j' oraz całkowite zapasy czasu z_{ij}^c . Przyjmij $p := 1$.
- K2. Sprawdź, czy aktualny czas realizacji projektu $T^p \leq T^d$. Jeżeli tak, idź do kroku K6. Jeżeli nie, idź do kroku K3.
- K3. W sieci kosztowej SK^p ustal przekrój o minimalnym koszcie $\underline{U}^p$.
- K4. Sprawdź, czy koszt minimalnego przekroju $K(\underline{U}^p) < \infty$. Jeżeli tak, przejdź do kroku K5. Jeżeli nie, przejdź do kroku K7.
- K5. W kolejnej sieci czasowej SK^{p+1} , skróć o jednostkę czasu trwania czynności, które tworzą minimalny przekrój. Ustal dla tej sieci czas realizacji projektu T^{p+1} oraz zapasy czasu z_{ij}^c . Przyjmij $p := p + 1$ i wróć do kroku K2.
- K6. Znalezione zostały końcowy plan realizacji projektu.
- K7. Analizowane zadanie jest sprzeczne (brak dopuszczalnego planu realizacji).

Algorytm ZKD można stosować także do rozwiązywania problemu z liniową reprezentacją krzywej kosztów. Efektywnie wówczas skracać czynności nie o jedną jednostkę, ale o więcej niż jedną jednostkę. Aby tę wielkość ustalić, dla czynności należących do minimalnego przekroju obliczamy aktualne możliwe skrócenie

$$s_{ij} = t_{ij} - t_{ij}', \quad \langle i, j \rangle \in \underline{U}^p \quad (26)$$

Niech N będzie aktualnym zbiorem czynności niekrytycznych. Ustalamy trzy parametry

$$a = \min\{s_{ij} \mid \langle i, j \rangle \in \underline{U}^p\} \quad (27)$$

$$z = \min\{z_{ij}^c \mid \langle i, j \rangle \in N\} \quad (28)$$

$$v = T^p - T^d \quad (29)$$

i wyznaczamy maksymalne skrócenie czynności minimalnego przekroju

$$s = \min\{a, z, v\} \quad (30)$$

6. Analiza sprawności metod ogólnych i wyspecjalizowanych

W analizie wzięto pod uwagę trzy metody:

- 1) metodę ogólną – zadanie PL + odpowiedni algorytm jego rozwiązywania, czyli algorytm simpleks,
- 2) algorytm wydłużania czynności W1,
- 3) algorytm skracania czynności ZKD.

Wszystkie trzy metody oprogramowano w języku Java tworząc również generator przypadków testowych sparametryzowanych względem: liczby wierzchołków, stopni wierzchołków, liczby spadku kosztów. Zbadano czas obliczeń i dokładność wyników. Uzyskane wyniki przedstawiono w tabelach 2 i 3.

Tabela 2

Czas obliczeń (w sekundach) dla trzech metod

Liczba wierzchołków	Stopień wierzchołka	PL	W1	ZKD
5	2	0,0020	0,0019	0,0023
5	3	0,0022	0,0019	0,0013
10	2	0,0037	0,0002	0,0010
10	3	0,0041	0,0004	0,0003
10	5	0,0092	0,0018	0,0004
20	2	0,0083	0,0015	0,0009
20	3	0,0128	0,0016	0,0018
20	5	0,0229	0,0030	0,0020
20	10	0,0611	0,0061	0,0026
50	3	0,0694	0,0059	0,0041
50	5	0,1652	0,0144	0,0115
50	10	0,5633	0,0418	0,0354
50	20	1,5989	0,1481	0,1052
100	3	0,2775	0,0519	0,0124
100	5	0,7020	0,0735	0,0314
100	10	2,3987	0,2331	0,1534
100	20	8,6875	0,8679	0,4128
100	50	41,1588	3,4239	2,3166

Średni czas obliczeń zależy od rozmiarów sieci. Im więcej węzłów i łączących je łuków, tym czas obliczeń jest większy. Dla metody simpleks czas wzrasta od 0,002 s w najmniejszym przypadku, do 42,16 s – w największej sieci. Dla algorytmu W1 czas wzrasta odpowiednio od 0,0019 s do 3,4239 s, a dla algorytmu ZKD – od 0,0023 s do 2,3166 s.

Średnio czas obliczeń dla metody simpleks jest około 10 razy dłuższy niż dla algorytmu W1 i około 15 razy dłuższy dla algorytmu ZKD. Jednak dla żadnego przypadku czas obliczeń nie przekracza 1 minuty.

Pod względem dokładności algorytm ZKD jest także lepszy od algorytmu W1. Dokładność obu algorytmów nie zależy w tym przypadku od rozmiarów analizowanej sieci.

Tabela 3

Dokładność trzech metod (w %)

Liczba wierzchołków	Stopień wierzchołka	PL	W1	ZKD
5	2	---	0,18	0,00
5	3	---	0,47	0,00
10	2	---	0,92	0,00
10	3	---	1,38	0,02
10	5	---	0,62	0,00
20	2	---	0,74	0,00
20	3	---	1,33	0,01
20	5	---	1,65	0,00
20	10	---	0,63	0,01
50	3	---	1,05	0,00
50	5	---	1,10	0,01
50	10	---	1,03	0,00
50	20	---	0,31	0,00
100	3	---	0,39	0,02
100	5	---	0,65	0,00
100	10	---	0,79	0,01
100	20	---	0,82	0,00
100	50	---	0,60	0,02

Większa skuteczność algorytmu ZKD niż algorytmu W1 wynika prawdopodobnie z dwóch faktów:

1) szukamy minimalnego przekroju, a nie frontu, czyli minimum na szerszym zbiorze czynności, co generalnie daje większą dokładność wyniku,

2) szukamy minimalnego przekroju sieci krytycznej, a nie w całej analizowanej sieci, co pozwala na szybsze działanie algorytmu.

Gaspars-Wieloch¹² analizowała także sprawność metody ogólnej (zadanie PL + algorytm simpleks), algorytmu ZKD i własnej propozycji – algorytmu SCTPPN. Uzyskane rezultaty są zbliżone do wyników z tabeli 2 i 3. Algorytm SCTPPN działał wolniej niż ZKD, ale za to dawał wyniki bardziej dokładne niż procedura ZKD.

Podsumowanie

Analiza sprawności trzech metod wykazuje, że:

1. Zmodyfikowany algorytm skracania Kaufmanna i Desbazeille'a daje lepszą średnią dokładność oraz krótszy czas obliczeń niż algorytm wydłużania W1.
2. Metoda ogólna oparta na zadaniu PL oraz algorytmie simpleks ma czas liczenia w porównaniu z algorytmami specjalnymi kilka lub kilkanaście razy dłuższy, ale i tak czas obliczeń nie przekracza 1 minuty.
3. Ciekawe byłoby porównanie sprawności algorytmu dokładnego, ale specjalnego np. zmodyfikowanego algorytmu Philips'a i Dessouky'ego¹³.

Literatura

- Anholcer M.: Przepływy w sieciach. W: Badania operacyjne. Red. W. Sikora. PWE, Warszawa 2008.
- Anholcer M., Gaspars-Wieloch H.: The Efficiency Analysis of the Kaufmann and Desbazeille Algorithm for the Deadline Problem. „Operations Research and Decisions” 2011, Vol. 2.
- Ford L.R., Fulkerson D.R.: Przepływy w sieciach. PWN, Warszawa 1969.
- Gaspars H.: Analiza czasowo-kosztowa (CPM-COST). Algorytm a model optymalizacyjny. „Badania Operacyjne i Decyzje” 2006, nr 1.
- Gaspars H.: Propozycja nowego algorytmu w analizie czasowo-kosztowej przedsięwzięć. „Badania Operacyjne i Decyzje” 2006, nr 3-4.
- Gaspars-Wieloch H.: Analiza sieciowa przedsiębiorstw. W: Badania operacyjne. Red. W. Sikora. PWE, Warszawa 2008.

¹² H. Gaspars-Wieloch: Metody optymalizacji czasowo-kosztowej przedsięwzięcia. AE, Poznań 2008.

¹³ S.J. Philips, M.I. Dessouky: Solving the Time/Cost Tradeoff Problem using the Minimum Cut Concept. „Management Science” 1977, Vol. 24(4).

- Gaspars-Wieloch H.: Metody optymalizacji czasowo-kosztowej przedsięwzięcia. AE, Poznań 2008.
- Ignasiak E.: Programowanie sieciowe. PWE, Warszawa 1972.
- Kaufmann A., Desbazeille G., Ventura E.: La méthode du chemin critique. Dunod, Paris 1964.
- Philips S.J., Dessouky M.I.: Solving the Time/Cost Tradeoff Problem using the Minimum Cut Concept. „Management Science” 1977, Vol. 24(4).
- Sikora W.: Metoda stopniowej analizy zmiennych dla zadania programowania rozłącznego. „Przegląd Statystyczny” 1991, nr 2.
- Sikora W.: Metoda wydłużania czynności w analizie czasowo-kosztowej przedsięwzięć. W: Z prac katedry badań operacyjnych. Red. W. Sikora. UEP, Poznań 2012.

THE EFFICIENCY OF THE GENERALISED AND SPECIALISED METHODS IN THE TIME-COST ANALYSIS

Summary

The paper considers the classical problem of time-cost analysis of projects involving the minimization of the cost in the given date of completion. Two models for linear and point representation of costs were formulated. We compared the efficiency (accuracy and computation time) of three methods of solving the problem: general method, a new algorithm for extending the activities W1 and the modified Kaufmann and Desbazeille's algorithm.