Pesymistyczna złożoność obliczeniowa algorytmu faktoryzacji *Fact*

Lech Madeyski[1]**, Zygmunt Mazur**[2]
Politechnika Wrocławska, Wydział Informatyki i Zarządzania, Wydziałowy Zakład Informatyki
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław

Streszczenie. W artykule zaprezentowano algorytm faktoryzacji *Fact* obliczania niezawodności *K*-terminali sieci probabilistycznych reprezentowanych przez graf $G = (V, E)$ z wyróżnionym podzbiorem węzłów K, jak również metodę oceny pesymistycznej złożoności tego algorytmu. Pośród algorytmów faktoryzacji najniższą złożonością pesymistyczną charakteryzuje się algorytm analizowany przez Wooda, którego liczba liści binarnego drzewa obliczeń w przypadkach granicznych, gdy $2 \le |K| \le 5$ oraz $|V| - 2 \le |K| \le |V|$, nie przekracza $(|V| - 2)!$. Dla algorytmu *Fact* wykażemy, że maksymalna liczba liści binarnego drzewa obliczeń $(|V| - 2)!$ jest osiągalna dla dowolnego K ($2 \le |K| \le |V|$), a nie tylko w przypadkach granicznych. Rezultat ten uzyskano przy prostszym niż zaproponowany przez Wooda zbiorze zachowujących niezawodność redukcji grafu i nieskomplikowanej strategii selekcji krawędzi do faktoryzacji.

Słowa kluczowe: analiza algorytmów, algorytmy faktoryzacji, niezawodność *K*-terminali.

1. Algorytm faktoryzacji

W artykule rozpatrywany jest, powszechnie podejmowany przez badaczy, problem analizy niezawodności sieci z wykorzystaniem miar niezawodności bazujących na spójności sieci reprezentowanej poprzez probabilistyczny graf nieskierowany $G = (V, E)$ z wyróżnionym podzbiorem węzłów K (2<=|*K*|<=|*V*|). Krawędzie grafu reprezentują łącza komunikacyjne, które ulegają wzajemnie niezależnie losowym uszkodzeniom ze znanym prawdopodobieństwem. Wśród miar oceny niezawodności sieci, najbardziej uniwersalną i powszechnie stosowaną miarą jest niezawodność *K*-terminali (*K-terminal network reliability*), definiowana jako prawdopodobieństwo, iż wszystkie węzły znajdujące się w zbiorze *K* są połączone za pomocą nie uszkodzonych łączy (mogą się komunikować).

[1] E-mail: madeyski@ci.pwr.wroc.pl.

[2] E-mail: mazur@ci.pwr.wroc.pl.

Spośród algorytmów znajdujących dokładne rozwiązanie problemu niezawodności K-terminali szczególnie dużo uwagi poświęcono algorytmom faktoryzacji i zachowującym niezawodność redukcjom grafu [1–13].

Algorytm faktoryzacji wykorzystuje zdarzenia elementarne sprawności lub niesprawności pojedynczej krawędzi. Stany grafu można podzielić na dwa zbiory ze względu na dwa możliwe stany krawędzi e_i o niezawodności p_i. Stąd niezawodność K-terminali można wyrazić w postaci prostej formuły niezawodności warunkowej:

$$R(G_K) = p_i R(G_K | e_i \text{ funkcjonuje}) + (1 - p_i) R(G_K | e_i \text{ nie funkcjonuje}). \tag{1}$$

Twierdzenie faktoryzacji jest topologiczną interpretacją formuły (1) dla grafów nieskierowanych.

Twierdzenie 1 (Twierdzenie faktoryzacji)

Niezawodność K-terminali sieci probabilistycznej reprezentowanej poprzez graf G z wyróżnionym podzbiorem węzłów K można wyrazić następująco:

$$R(G_K) = p_i R(G_{K'} * e_i) + (1 - p_i) R(G_K - e_i), \tag{2}$$

gdzie:

e_i – dowolna krawędź grafu G_K;

p_i – prawdopodobieństwo, że łącze reprezentowane przez $e_i \in E$ funkcjonuje;

$$G_{K'} * e_i = (V - u - v + w, E - e_i), \quad w = u \cup v;$$

$$K' = \begin{cases} K & \text{jeżeli } u, v \notin K; \\ K - u - v + w & \text{jeżeli } u \in K \text{ lub } v \in K; \end{cases}$$

$$G_K - e_i = (V, E - e_i).$$

Algorytm faktoryzacji uzupełniony o funkcje redukcji grafu można opisać w postaci następującej trójki **($\mathbf{F_0}$, R, S)**, gdzie:

$\mathbf{F_0}$ – szkielet algorytmu obliczania niezawodności wykorzystujący twierdzenie faktoryzacji;

R – zbiór zachowujących niezawodność redukcji i dekompozycji grafu[3];

S – strategia wybierania krawędzi grafu do faktoryzacji (dekompozycji problemu).

[3] Niektóre zachowujące niezawodność redukcje grafu przedstawiono w załączniku 1.

Spośród algorytmów faktoryzacji najniższą pesymistyczną złożonością czasową charakteryzuje się algorytm faktoryzacji opisany przez Wooda [11, 12] (**F₀**, {R1, R2, R3, R4}, **S**), gdzie:

S = $\mathbf{S_2} \cap \mathbf{S_3}$;

$\mathbf{S_2}$ = {Może być wybrana dowolna krawędź $e_i \in E$ za wyjątkiem takiej, że $|K'| = 1$ w $G_K * e_i$ };

$\mathbf{S_3}$ = {Może być wybrana dowolna krawędź $e_i \in E$ jeżeli $G_K * e_i$ i $G_K - e_i$ są dwuspójne i nie mają pętli własnych}.

W przypadku tego algorytmu maksymalna liczba liści binarnego drzewa obliczeń $(|V| - 2)!$ jest osiągalna dla przypadków granicznych, gdy $2 \le |K| \le 5$ oraz $|V| - 2 \le |K| \le |V|$. Znany jest również inny algorytm faktoryzacji [8, 12], dla którego maksymalna liczba liści binarnego drzewa obliczeń jest równa $(|V| - 1)!$ dla dowolnego zbioru K ($2 \le |K| \le |V|$).

Zaprezentowany w tym artykule algorytm faktoryzacji *Fact* ($\mathbf{F_0}$, {R0, R1, R2, R3}, **S”**) jest modyfikacją algorytmu faktoryzacji przedstawionego przez Page’a i Perry [3]. Dzięki zaproponowaniu nowej strategii **S”** selekcji krawędzi do faktoryzacji, jak również nowej metody analizy pesymistycznej złożoności obliczeniowej algorytmu wykażemy, że maksymalna liczba $(|V| - 2)!$ liści binarnego drzewa obliczeń algorytmu faktoryzacji jest osiągalna dla dowolnego zbioru K ($2 \le |K| \le |V|$), a nie tylko dla przypadków granicznych, gdy $2 \le |K| \le 5$ oraz $|V| - 2 \le |K| \le |V|$ (jak to ma miejsce w przypadku algorytmu analizowanego przez Wooda [11, 12]). Rezultat ten uzyskano przy prostszym niż zaproponowany przez Wooda zbiorze zachowujących niezawodność redukcji grafu i nieskomplikowanej strategii selekcji krawędzi do faktoryzacji. Dokładna analiza procesu faktoryzacji w przypadku algorytmu *Fact* wykorzystującego strategię **S”,** jak również sama strategia **S”** selekcji krawędzi do faktoryzacji, przedstawiona jest w następnym rozdziale.

2. Analiza pesymistycznej złożoności obliczeniowej algorytmu faktoryzacji metodą przekształcenia do grafu pełnego

Znane z literatury metody analizy pesymistycznej złożoności algorytmów faktoryzacji wykorzystywały niezmienniki grafów [8, 12]. W przypadku analizy pesymistycznej złożoności algorytmu *Fact* zaproponowano odmienną metodę.

2.1. Idea przekształcenia do grafu pełnego

Mamy dowolną sieć reprezentowaną poprzez graf G_K o n-wierzchołkach. Graf G_K możemy uzupełnić do postaci grafu pełnego krawędziami reprezentującymi łącza o zerowym prawdopodobieństwie uszkodzenia. W wyniku tej operacji niezawodność *K*-terminali takiej sieci nie ulega zmianie. Dokonując faktoryzacji (wykorzystując określoną strategię selekcji **S**) na krawędziach o prawdopodobieństwie uszkodzenia $p_i = 0$, zamiast dwóch podproblemów (jak ma to miejsce przy faktoryzacji na krawędziach o $p_i \neq 0$) otrzymujemy tylko jeden podproblem. W związku z tym liczba liści BDO algorytmu faktoryzacji wykorzystującego strategię **S** przy rozwiązywaniu dowolnych sieci reprezentowanych przez G_K, nie będzie większa niż $L(K_n)$, gdzie K_n jest grafem pełnym o n wierzchołkach. Podobnie głębokość rekurencji przy rozwiązywaniu sieci reprezentowanych przez G_K nie będzie większa niż w przypadku grafu pełnego.

Jeżeli ponadto strategia **S** selekcji krawędzi do faktoryzacji będzie tak dobrana, by miała miejsce dekompozycja problemu obliczenia niezawodności *K*-terminali sieci reprezentowanej przez graf pełny K_n (o n wierzchołkach) na pewną liczbę podproblemów niższego rzędu reprezentowanych poprzez grafy pełne K_{n-1}, to określenie pesymistycznej złożoności obliczeniowej rozważanego algorytmu będzie stosunkowo proste.

2.2. Złożoność czasowa algorytmu faktoryzacji

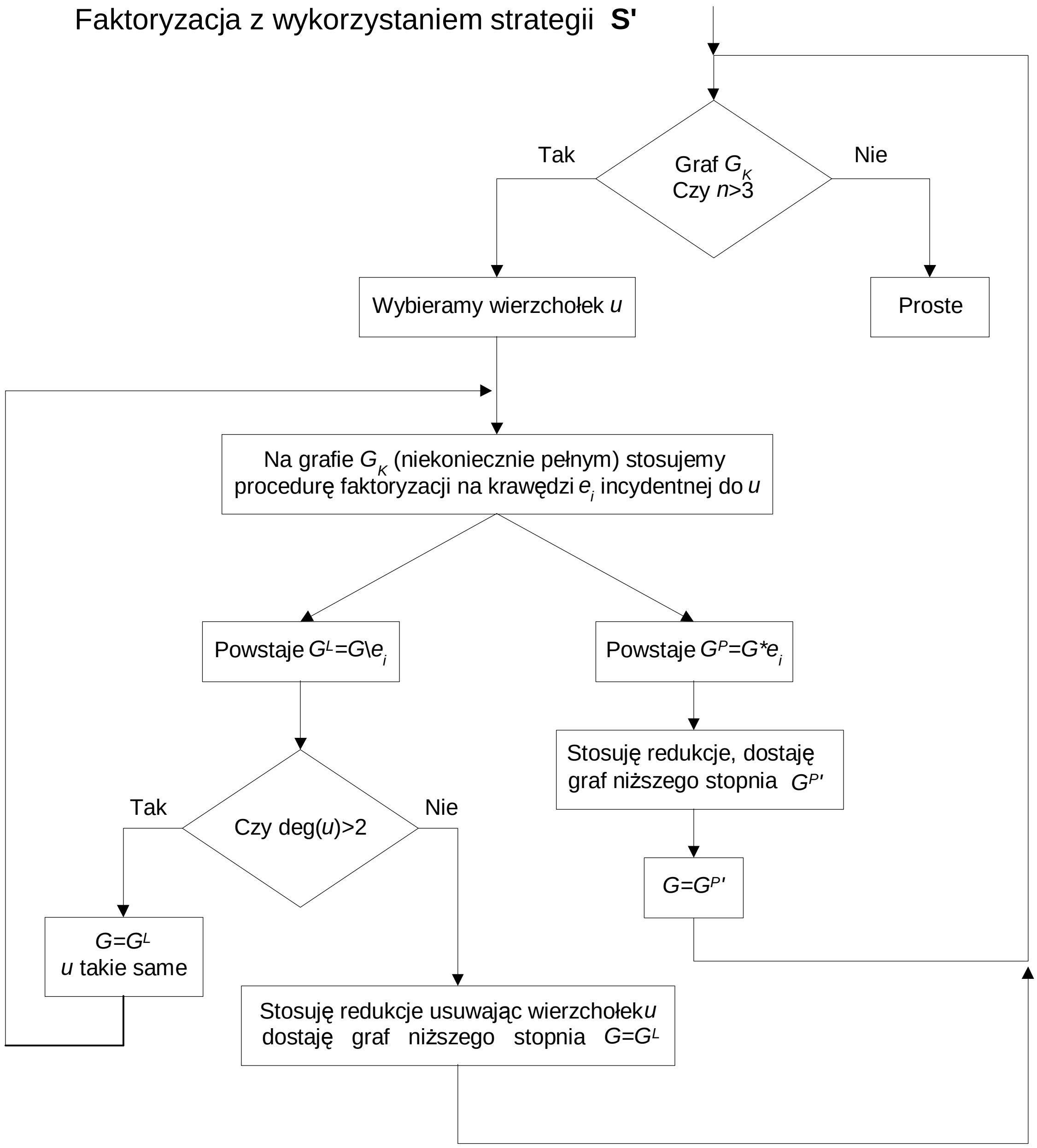
Chcąc oszacować złożoność czasową algorytmu faktoryzacji należy uwzględnić liczbę węzłów (liści) BDO, jak również złożoność czasową operacji dokonywanych w poszczególnych węzłach BDO, którą można oszacować przez $O(b^2)$.

2.2.1. Niezawodność wszystkich terminali

Dla ułatwienia rozpatrzmy najpierw często rozważany przypadek obliczania niezawodności wszystkich terminali (tzn. gdy $|K| = |V|$).

Twierdzenie 2

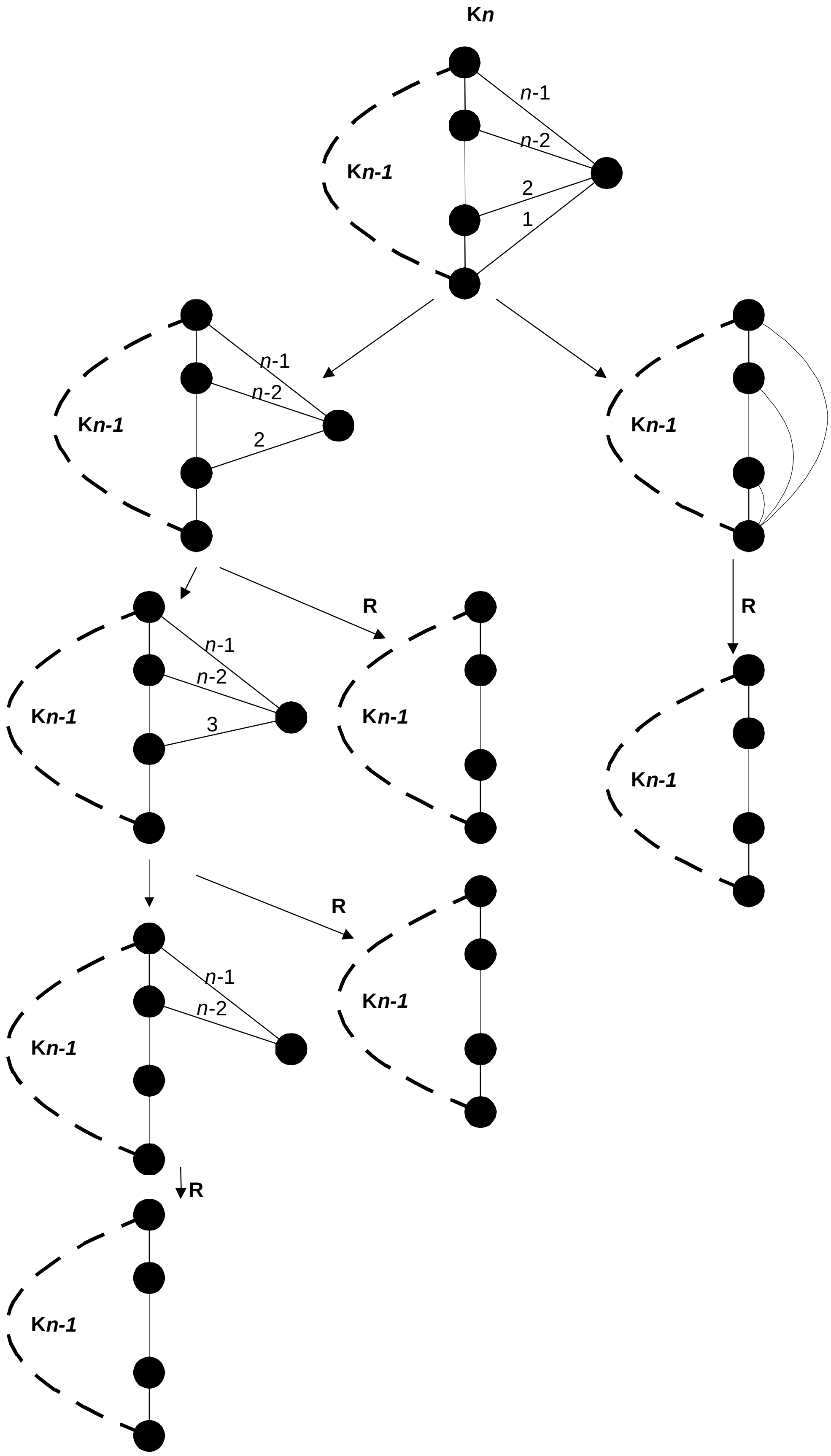
Liczba liści BDO algorytmu faktoryzacji, wykorzystującego zbiór redukcji **R**={R0, R1, R2, R3} oraz strategię **S=S'** (określoną na rysunku poniżej), przy rozwiązywaniu problemu niezawodności wszystkich terminali dowolnej sieci reprezentowanej poprzez graf G_V, spełnia zależność $L(G_V) \le (n-2)!$, gdzie n jest liczbą wierzchołków (przy rozwiązywaniu problemu niezawodności wszystkich terminali redukcje R2 nie są używane).



Rysunek 1. Faktoryzacja z wykorzystaniem strategii S' selekcji krawędzi grafu do faktoryzacji.

Dowód:

W przypadku obliczania niezawodności wszystkich terminali sieci reprezentowanej przez graf pełny, fragment BDO algorytmu faktoryzacji przedstawiony jest na poniższym rysunku.



Rysunek 2. Fragment binarnego drzewa obliczeń algorytmu faktoryzacji (F_0, {R0, R1, R2, R3}**, S') przy rozwiązywaniu problemu niezawodności wszystkich terminali sieci reprezentowanych przez graf pełny.**

Wybieramy wierzchołek u, który ma $n-1$ krawędzi incydentnych. Następnie $n-3$ razy stosujemy formułę faktoryzacji. Otrzymujemy $n-2$ węzłów BDO, w których występuje sieć K_{n-1}. Liczbę liści BDO algorytmu faktoryzacji można więc wyrazić rekurencyjnie:

$$L(K_n) = (n-2)L(K_{n-1})$$
$$L(K_1) = L(K_2) = L(K_3) = 1.$$

W efekcie liczba liści BDO algorytmu faktoryzacji (**F₀**, {R0, R1, R2, R3}, **S'**) w przypadku obliczania niezawodności wszystkich terminali spełnia zależność $L(G_V) \leq (n-2)!$ (redukcje R2 nie są używane).

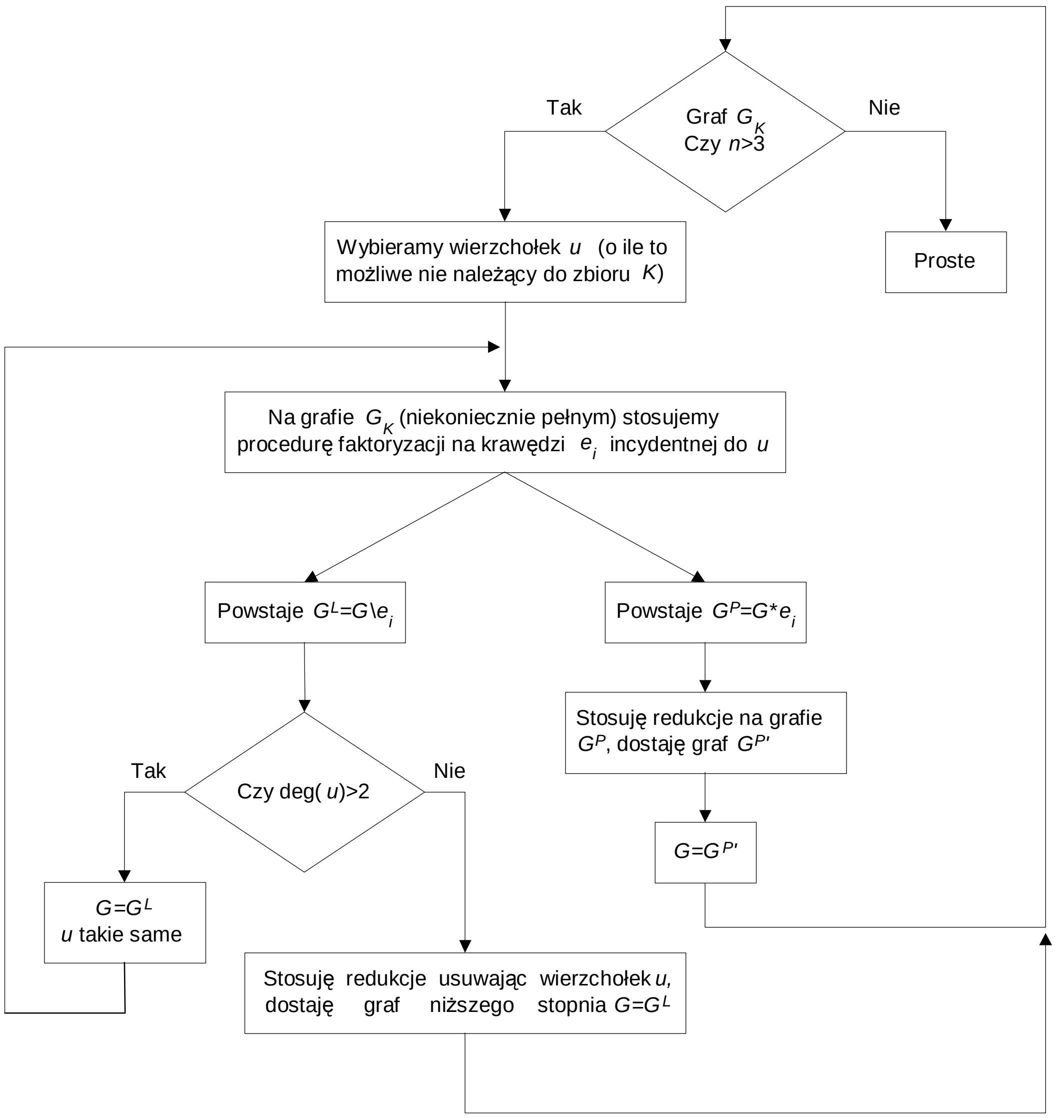
c.n.d.

2.2.2. Niezawodność *K*-terminali

W rozdziale tym uogólnimy rozpatrywany w poprzednim rozdziale przypadek tak, by dotyczył niezawodności *K*-terminali (gdy $2 \leq |K| \leq |V|$), a nie tylko niezawodności wszystkich terminali ($|K| = |V|$).

Modyfikując strategię **S'** można uogólnić twierdzenie 2 na problem niezawodności *K*-terminali. Jeżeli nie wszystkie wierzchołki rozpatrywanego grafu należą do zbioru *K*, wystarczy w ramach stosowanej strategii selekcji krawędzi wybierać taki wierzchołek *u*, że $u \notin K$. Tak zmodyfikowaną strategię selekcji krawędzi **S"** przedstawia poniższy rysunek.

Faktoryzacja z wykorzystaniem strategii **S"**



Rysunek 3. Faktoryzacja z wykorzystaniem strategii S" selekcji krawędzi grafu do faktoryzacji.

Modyfikacja strategii selekcji krawędzi grafu do faktoryzacji umożliwia redukcję wierzchołków stopnia 2:

- za pomocą redukcji R2 w przypadku, gdy nie wszystkie wierzchołki grafu należą do zbioru K ;
- za pomocą redukcji R3 w przypadku, gdy wszystkie wierzchołki grafu należą do zbioru K.

Dzięki temu liczba liści BDO, a co za tym idzie również złożoność algorytmu *Fact* (**F**$_0$, {R0, R1, R2, R3}, **S"**) w przypadku obliczania niezawodności K-terminali nie ulegnie zmianie w porównaniu do przypadku rozpatrywanego w twierdzeniu 2. Dlatego prawdziwe jest poniższe twierdzenie.

Twierdzenie 3

Liczba liści BDO algorytmu faktoryzacji *Fact*, wykorzystującego zbiór redukcji **R**={R0, R1, R2, R3} oraz strategię **S"** (określoną na rysunku powyżej), przy rozwiązywaniu problemu niezawodności *K*-terminali dowolnej sieci reprezentowanej poprzez graf G_K spełnia zależność $L(G_K) \le (n-2)!$, gdzie n jest liczbą wierzchołków.

W rezultacie prawdziwe jest również poniższe twierdzenie.

Twierdzenie 4

Pesymistyczna złożoność czasowa algorytmu faktoryzacji *Fact* (**F₀**, {R0, R1, R2, R3}, **S"**), przy rozwiązywaniu problemu niezawodności *K*-terminali dowolnej sieci reprezentowanej poprzez graf G_K wynosi $O\left(b^2((n-2)!)\right)$.

2.3. Złożoność pamięciowa algorytmu faktoryzacji

Rozmiar wymaganej pamięci można ogólnie oszacować poprzez iloczyn rozmiaru sieci $|G|$ (t.j. zestawu danych, wykorzystanych głównie do reprezentacji sieci w pojedynczej instancji rekurencyjnej funkcji *Prob*) i głębokości rekurencji $d(G_K)$. Stąd zgrubnym pesymistycznym oszacowaniem złożoności pamięciowej algorytmów faktoryzacji jest $O(b|G_K|)$, gdzie *b* jest liczbą krawędzi grafu reprezentującego badaną sieć.

Pesymistyczną złożoność pamięciową algorytmu faktoryzacji można również, podobnie jak w przypadku złożoności czasowej, w prosty sposób analizować wykorzystując grafy pełne. Głębokość rekurencji przy rozwiązywaniu sieci reprezentowanych przez G_K nie będzie większa niż w przypadku grafu pełnego. Uzyskane w ten sposób oszacowania mogą być jednak (szczególnie w przypadku grafów rzadkich) gorsze od oszacowania $O(b|G_K|)$.

Jak można zaobserwować na rysunku 2. głębokość rekurencji algorytmu faktoryzacji, wykorzystującego zbiór redukcji **R**={R0, R1, R2, R3} oraz strategię **S'** lub **S"** w przypadku rozwiązywania problemu niezawodności *K*-terminali sieci reprezentowanej poprzez graf pełny K_n spełnia zależność $d(K_n) = (n-3) + d(K_{n-1})$. Stąd:

$$d(K_n) = \frac{1+(n-3)}{2}(n-3) = \frac{(n-2)(n-3)}{2} = \frac{n^2-5n+6}{2} = \frac{n(n-1)-4n+6}{2} = b-2n+3 .$$

Twierdzenie 5

Głębokość rekurencji algorytmu faktoryzacji *Fact*, wykorzystującego zbiór redukcji **R**={R0, R1, R2, R3} oraz strategię **S"**, przy rozwiązywaniu problemu niezawodności *K*-terminali dowolnej sieci reprezentowanej poprzez graf G_K, spełnia zależność $d(G_K) \le \frac{(n-2)(n-3)}{2}$, gdzie n jest liczbą wierzchołków.

Twierdzenie 6

Pesymistyczna złożoność pamięciowa algorytmu faktoryzacji *Fact* (**F$_0$**, {R0, R1, R2, R3}, **S"**), przy rozwiązywaniu problemu niezawodności *K*-terminali dowolnej sieci reprezentowanej poprzez graf G_K wynosi $O\left(\frac{(n-2)(n-3)}{2}|G_K|\right)$.

3. Podsumowanie

Przedstawiony w tym rozdziale algorytm *Fact* (**F$_0$**, {R0, R1, R2, R3}, **S"**) jest modyfikacją algorytmu faktoryzacji przedstawionego przez Page'a i Perry [3]. Dzięki zaproponowaniu nowej strategii **S"** selekcji krawędzi do faktoryzacji, jak również nowej metody analizy pesymistycznej złożoności obliczeniowej algorytmu wykazano, że maksymalna liczba $(|V|-2)!$ liści binarnego drzewa obliczeń algorytmu faktoryzacji jest osiągalna dla dowolnego zbioru K ($2 \le |K| \le |V|$), a nie tylko dla przypadków granicznych, gdy $2 \le |K| \le 5$ oraz $|V|-2 \le |K| \le |V|$ (jak to ma miejsce w przypadku algorytmu analizowanego przez Wooda [11, 12]). Rezultat ten uzyskano przy prostszym niż zaproponowany przez Wooda zbiorze zachowujących niezawodność redukcji grafu **R**={R0, R1, R2, R3}. Nie ma bowiem konieczności uwzględniania najbardziej skomplikowanego[4] podzbioru redukcji R4 (redukcji wielokąta do łańcucha). Również zaproponowana strategia selekcji krawędzi do faktoryzacji nie wymaga stosowania złożonych procedur testowania własności (np. dwuspójności) sieci występujących w poszczególnych węzłach BDO.

Zaprezentowana w tym rozdziale metoda analizy pesymistycznej złożoności obliczeniowej jest na tyle uniwersalna, że można ją również wykorzystać do analizy odmian algorytmu *Fact*, wykorzystujących różne zbiory zachowujących niezawodność redukcji grafów. Złożoność czasowa tych algorytmów nie była dotąd analizowana a uzyskane rezultaty

[4] Por. załącznik 1.

pomogą określić wpływ zastosowanych redukcji grafu na pesymistyczną złożoność obliczeniową algorytmów faktoryzacji.

Literatura:

[1] M. S. Choi, C. H. Jun, Some variants of polygon-to-chain reductions in evaluation reliability of undirected network, *Microelectron. Reliab.*, 1995(35), 1-11.

[2] M. K. F. Lai, Polygon-to-chain reductions work for networks with imperfect vertices, *Microelectron. Reliab.*, 1994(34), 267-274.

[3] L. B. Page, J. E. Perry, A practical implementation of the factoring theorem for network reliability, *IEEE Trans. Reliability*, 1988 (37) Aug, 259-267.

[4] L. B. Page, J. E. Perry, Reliability of directed networks using the factoring theorem, *IEEE Trans. Reliability*, 1989 (38) Dec, 556-562.

[5] M. G. C. Resende, A program for reliability evaluation of undirected networks via polygon-to-chain reductions, *IEEE Trans. Reliability*, 1986 (R-35) Apr, 24-29.

[6] L. I. P. Resende, Implementation of factoring algorithm for reliability evaluation of undirected networks, *IEEE Trans. Reliability*, 1988 (37) Dec, 462-468.

[7] A. Satyanarayana, R. K. Wood, *Polygon-to-chain reductions and network reliability*, Operations Research Center, University of California, Report ORC 82-4, 1982.

[8] A. Satyanarayana, M. K. Chang, Network Reliability and the Factoring Theorem, *Networks*, 1983 (13), 107-120.

[9] O. R. Theologou, J. G. Carlier, Factoring and Reductions for Networks with Imperfect Vertices, *IEEE Trans. Reliability*, 1991 (40), 210-217.

[10] R. K. Wood, *Polygon-to-chain reductions and extensions for reliability evaluation of undirected networks*, PhD thesis, Dept. of Industrial Engineering and Operations Research, University of California, Berkeley, 1982.

[11] R. K. Wood, A Factoring Algorithm Using Polygon-to-Chain Reductions for Computing *K*-Terminal Network Reliability, *Networks*, 1985 (15), 173-190.

[12] R. K. Wood, Factoring Algorithms for Computing K-Terminal Network Reliability, *IEEE Trans. Reliability*, 1986 (R-35), 269-278.

[13] R. K. Wood, Triconnected decomposition for computing K-terminal network reliability, *Networks*, 1989 (19), 203-220.

Załącznik 1 – Zachowujące niezawodność redukcje grafu

Aby zredukować rozmiar badanych grafów, a co za tym idzie także rozmiar przestrzeni stanów rozpatrywanego problemu niezawodności *K*-terminali, czyli złożoność problemu, można zastosować zachowujące niezawodność redukcje grafu (*reliability preserving reductions*) ze zbioru **R**. Redukcje te wymagają tylko wielomianowego czasu wykonania[5] zmniejszając wykładniczą przestrzeń stanów problemu. Redukcje zmieniają graf G_K topologicznie i probabilistycznie przekształcając go do postaci $G'_{K'}$ tak, że:

$$R(G_K) = \Omega_1 + \Omega_2 R(G'_{K'}). \tag{3}$$

Ω_1 i Ω_2 są stałymi uzyskanymi wyłącznie z oryginalnego grafu G_K. Ta ogólna formuła jest często upraszczana do postaci, która zawiera jedynie stałą multiplikatywną $\Omega = \Omega_2$. Ponieważ większość redukcji wykorzystuje jedynie stałą multiplikatywną równanie (3) przyjmuje następującą postać:

$$R(G_K) = \Omega R(G'_{K'}) \tag{4}$$

Poniżej zostaną przedstawione najczęściej wykorzystywane w praktyce redukcje grafu[6]. Przyjęto następujące oznaczenia wierzchołków na rysunkach opisujących redukcje:

 – wierzchołek grafu ze zbioru *K*

 – wierzchołek grafu spoza zbioru *K*

 – wierzchołek grafu, który może lecz nie musi należeć do zbioru *K*

R0 redukcja stopnia 1 (degree-1 reduction)

Jeśli $e_i = (u,v)$ będzie krawędzią w grafie G_K, a wierzchołek v będzie stopnia jeden tzn. $\deg(v) = 1$, to wtedy:

$$G'_{K'} = G_K * e_i = (V - u - v + w, E - e_i), \quad w = u \cup v,$$

$$K' = \begin{cases} K & \text{jeżeli } u, v \notin K \\ K - u - v + w & \text{jeżeli } u \in K \text{ lub } v \in K. \end{cases}$$

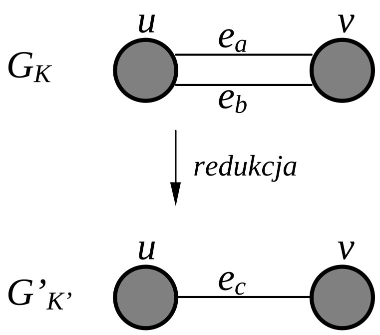
$$\Omega_1 = 0, \Omega_2 = \begin{cases} 1 & \text{jeżeli } v \notin K \\ p_a & \text{jeżeli } v \in K \end{cases}$$

---

[5] Por. [11], s.175 i [8], s.114.

[6] Dokładne opisy redukcji można znaleźć także w pracy [12], s.272-273 oraz [11], s.177-178.

R1 Redukcja równoległa (parallel reduction)



$e_a=(u,v)$ $e_b=(u,v)$

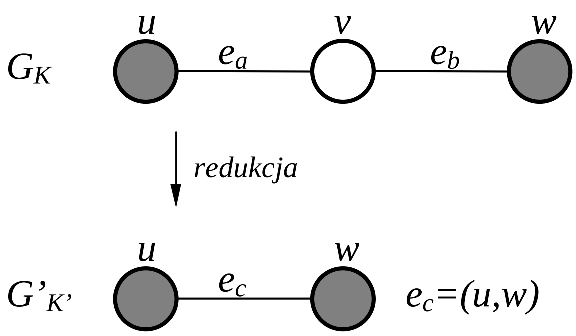
$e_c=(u,v)$

$p_c=1- q_aq_b$

$\Omega_1=0,\ \Omega_2=1$

$K'=K$

R2 Redukcja szeregowa (series reduction)



$e_a=(u,v)$ $e_b=(v,w)$

$v \notin K$

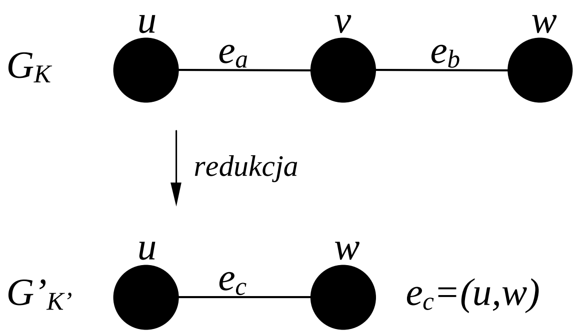
$e_c=(u,w)$

$p_c=p_ap_b$

$\Omega_1=0,\ \Omega_2=1$

$K'=K$

R3 Redukcja stopnia 2 (degree-2 reduction)



$e_a=(u,v)$ $e_b=(v,w)$
$u,v,w \in K$
$deg(v)=2$

$e_c=(u,w)$

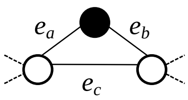
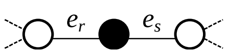
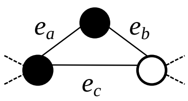
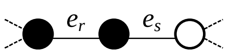
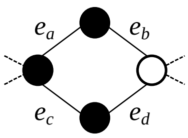
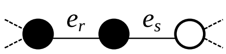
$p_c=p_a p_b /(1-q_a q_b)$

$\Omega_1=0,\ \Omega_2=(1-q_a q_b)$

$K'=K-v$

R4 Redukcje wielokąta do łańcucha (polygon-to-chain reduction)

Satyanarayana i Wood [7, 10, 11] zaproponowali redukcje R4, wielokąta do łańcucha (*polygon-to-chain reductions*). Jeżeli po zastosowaniu redukcji R1, R2, R3 graf zawiera wielokąt, to przybiera on jedną z siedmiu postaci i można go zastąpić przez łańcuch składający się z 1, 2 lub 3 krawędzi. Redukcje te mogą zmniejszyć wysiłek obliczeniowy wymagany do obliczenia niezawodności za pomocą algorytmu faktoryzacji. Poniższa tabela zawiera wszystkie siedem redukcji:

Wielokąt	Łańcuch	Nowe niezawodności łączy łańcucha	gdzie:
 (1)		$p_r=\delta/(\alpha+\delta)$	$\alpha=q_a p_b q_c$ $\beta=p_a q_b q_c$
 (2)		$p_s=\delta/(\beta+\delta)$ $\Omega_2=(\alpha+\delta)(\beta+\delta)/\delta$	$\delta=p_a p_b p_c[1+(q_a/p_a)+(q_b/p_b)+(q_c/p_c)]$
 (3)		$\Omega_1=0$	$\alpha=p_a q_b q_c p_d+q_a p_b p_c q_d+q_a p_b q_c p_d$ $\beta=p_a q_b p_c q_d$ $\delta=p_a p_b p_c p_d[1+(q_a/p_a)+(q_b/p_b)+(q_c/p_c)+(q_d/p_d)]$

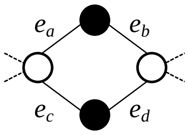
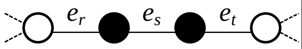
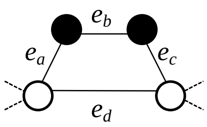
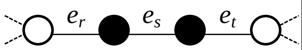
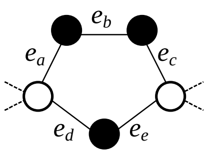
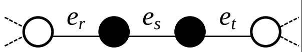
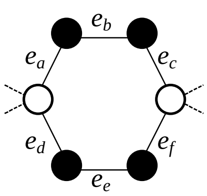
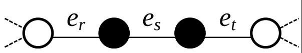
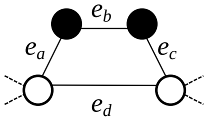
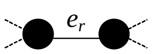
 (4)		$p_r=\gamma/(\alpha+\gamma)$	$\alpha=q_a p_b q_c p_d$ $\beta=p_a q_b q_c p_d+ q_a p_b p_c q_d$ $\delta=p_a q_b p_c q_d$ $\gamma=p_a p_b p_c p_d[1+(q_a/p_a)+(q_b/p_b)+(q_c/p_c)+(q_d/p_d)]$
 (5a)	|K| > 2 	$p_s=\gamma/(\beta+\gamma)$	$\alpha=q_a p_b p_c q_d$ $\beta=p_a q_b p_c q_d$ $\delta=p_a p_b q_c q_d$ $\gamma=p_a p_b p_c p_d[1+(q_a/p_a)+(q_b/p_b)+(q_c/p_c)+(q_d/p_d)]$
 (6)		$p_t=\gamma/(\delta+\gamma)$	$\alpha=q_a p_b p_c q_d p_e$ $\beta=p_a q_b p_c(p_d q_e+q_d p_e)+p_b(q_a p_c p_d q_e+p_a q_c q_d p_e)$ $\delta= p_a p_b q_c p_d q_e$ $\gamma=p_a p_b p_c p_d p_e[1+(q_a/p_a)+(q_b/p_b)+(q_c/p_c)+(q_d/p_d)+ (q_e/p_e)]$
 (7)		$\Omega_2=(\alpha+\gamma)(\beta+\gamma)(\delta+\gamma)/\gamma^2$ $\Omega_1=0$	$\alpha=q_a p_b p_c q_d p_e p_f$ $\beta=p_a q_b p_c(q_d p_e p_f+p_d q_e p_f+p_d p_e q_f) + p_a p_b q_c p_f(p_d q_e+q_d p_e)+ q_a p_b p_c p_d(q_e p_f+p_e q_f)$ $\delta= p_a p_b q_c p_d p_e q_f$ $\gamma=p_a p_b p_c p_d p_e p_f[1+(q_a/p_a)+(q_b/p_b)+(q_c/p_c)+(q_d/p_d)+(q_e/p_e)+(q_f/p_f)]$
 (5b)	|K| = 2 	$p_r=(p_b+p_a q_b p_c p_d)/\Omega_2$ $\Omega_2=p_b+p_a q_b p_c$ $\Omega_1=0$	$\alpha=q_a p_b p_c q_d$ $\beta=p_a q_b p_c q_d$ $\delta=p_a p_b q_c q_d$ $\gamma=p_a p_b p_c p_d[1+(q_a/p_a)+(q_b/p_b)+(q_c/p_c)+(q_d/p_d)]$

Tabela 1 Redukcje wielokąta do łańcucha (polygon-to-chain reductions)