Dokumentacja wstępna - Algorytmy Zaawansowane

Szymon Kuś, Piotr Skibiński

2 października 2025

Spis treści

1 Przedstawienie problemu

Problem rozważany w poniższej pracy dotyczy znalezienia skojarzenia doskonałego w nieskierowanym grafie dwudzielnym o najmniejszym koszcie.

Algorytm przyjmuje tylko grafy dwudzielne, tzn. postaci:

$$G(L, P, E, w) \tag{1}$$

gdzie:

$$L = \{l_1, l_2, .., l_n\}, P = \{p_1, p_2, .., p_n\}, E \subseteq L \times P, w : E \to \mathbb{R} \tag{2}$$

L i P to rozłączne zbiory wierzchołków, E to zbiór krawędzi, gdzie każda krawędź łączy wierzchołek z L z wierzchołkiem z P, w jest funkcją wagi.

2 Opis rozwiązania problemu

Będziemy posługiwać się następującymi oznaczeniami:

- M – skojarzenie, zbiór krawędzi.
- nasycona krawędź – krawędź, która należy do skojarzenia.
- nienasycona krawędź – krawędź, która nie należy do skojarzenia.
- nasycony wierzchołek – wierzchołek, który jest częścią dowolnej nasyconej krawędzi.
- nienasycony wierzchołek – wierzchołek, który nie jest częścią żadnej z nasyconych krawędzi.
- $M(L)$ – zbiór nasyconych wierzchołków z L.
- $M(P)$ – zbiór nasyconych wierzchołków z P.
- $N(a)$ – zbiór sąsiadów wierzchołka a.
- $N(A)$ – zbiór sąsiadów wierzchołków należących do A.
- S – pewien podzbiór L.
- T – pewien podzbiór P.
- $w(l, p)$ – waga krawędzi z l do p.
- D – drzewo M-przemienne budowane w trakcie algorytmu.
- ścieżka powiększająca / M-zasilona – ścieżka zwiększająca skojarzenie M.
- skojarzenie nasycające zbór wierzchołków L - takie skojarzenie M, że $|M(L)| = |L|$.

Ważną częścią naszego algorytmu jest twierdzenie Halla, które mówi:

$$\text{Graf dwudzielny } (L, P) \text{ ma skojarzenie nasycające zbiór } L \Leftrightarrow \forall_{S \subset L} |S| \leqslant |N(S)| \tag{3}$$

W przypadku grafu dwudzielnego o równej liczbie wierzchołków w L i P spełnienie warunku Halla dla zbioru L jest równoważne ze znalezieniem skojarzenia doskonałego.

2.1 Znajdowanie skojarzenia doskonałego w grafie ważonym

Implementowany algorytm znajduje skojarzenie o największej wadze, dlatego w grafie wejściowym G ustawiamy wagi na przeciwne. W ten sposób skojarzenie o największej wadze w nowym grafie będzie skojarzeniem o najmniejszej wadze w oryginalnym grafie. Podobnie na wyjściu waga najlepszego skojarzenia będzie miała ponownie zmieniony znak.

$$G = (L, P, E, w) \tag{4}$$

$$G_{przeciwny} = (L, P, E, -w) \tag{5}$$

Następnie iteracyjne szukamy skojarzenia w podgrafach G_I grafu $G_{przeciwny}$. Do stworzenia podgrafu niezbędne jest zdefiniowanie **funkcji potencjału**, która jako argument przyjmuje wierzchołek, a zwraca liczbę rzeczywistą:

$$I : (L \cup P) \to \mathbb{R} \tag{6}$$

takiej, że:

$$\forall_{lp \in E(G_{przeciwny})} I(l) + I(p) \geqslant w(l, p) \tag{7}$$

Graf G_I tworzymy poprzez:

$$G_I = (L, P, E_I) \tag{8}$$

$$E_I = \{lp \in E(G_{przeciwny}) : I(l) + I(p) = w(l, p)\} \tag{9}$$

W kolejnych iteracjach szukamy skojarzeń M^i ($M^0 = \{\}$) w grafach G_I^i które tworzymy poprzez manipulacje potencjałami. Skojarzenie M^{i+1} tworzymy poprzez powiększanie skojarzenia M^i o krawędzie z grafu G_I^{i+1}:

$$(M^{i+1}, S, T) = Algorytm2(G_I^{i+1}, M^i) \tag{10}$$

Możemy wykorzystać poprzednie skojarzenie, bo zaproponowana przez nas modyfikacja krawędzi zapewnia, że $M^i \subset E_I^i$. Dane skojarzenie powiększamy do momentu znalezienia skojarzenia doskonałego lub znalezienia zbiorów S i T, które nie spełniają warunku Halla (3), to znaczy takich, że:

$$|S| > |T|, \text{ gdzie } T = N(S) \tag{11}$$

Algorytm można opisać krokami:

0. Jeśli graf nie posiada skojarzenia doskonałego, zakończ algorytm (sprawdzanie Algorytmem 2)

1. Zamień wagi w grafie na przeciwne i przydziel początkowe wartości potencjałom.

   - $I(l) = max(w(l,p),\ p \in P)$ dla $l \in L$
   - $I(p) = 0$ dla $p \in P$

2. Ustaw początkową wartość skojarzenia $M = \{\}$.

3. Zbuduj graf G_I i znajdź skojarzenie, korzystając z Algorytmu 2, rozpoczynając od M. Algorytm 2 zwraca również zbiory S i T.

4. Jeśli $|M| = |L|$ zakończ algorytm i zwróć skojarzenie.

5. Modyfikujemy potencjały:

   5.1.
   - $q = min\{I(l) + I(p) - w(l,p) : l \in S, p \in P - T\}$
   - $I(l) = I(l) - q$ dla $l \in S$
   - $I(p) = I(p) + q$ dla $p \in T$

   5.2. Wróć do kroku 3.

2.1.1 Pseudokod

Algorytm 1 Znajdowanie skojarzenia doskonałego w grafie ważonym

Utwórz graf G_{bez_wag}
jeśli nie istnieje skojarzenie doskonałe w grafie G_{bez_wag} **wtedy**
  Zakończ algorytm
koniec jeśli
Zamień wagi w grafie G na przeciwne
// Początkowe wartości
dla $l \in L$ **wykonuj**
  $I(l) = max(w(l,p),\ p \in P)$
koniec pętli
dla $p \in P$ **wykonuj**
  $I(p) = 0$
koniec pętli
M = {}
dopóki $|M| < |L|$ **wykonuj**
  Skonstruuj graf $G_I = (L, P, E_I)$
  //Powiększ skojarzenie M w grafie G_I i pobierz wartości S i T
  $(M, S, T) = Algorytm2(G_I, M)$
  jeśli $|M| == |L|$ **wtedy**
    Zakończ program, zwróć M oraz zwróć -w(M)
  w przeciwnym wypadku
    // Modyfikujemy potencjały
    $q = min\{I(l) + I(p) - w(l,p) : l \in S, p \in P - T\}$
    dla $l \in S$ **wykonuj**
      $I(l) = I(l) - q$
    koniec pętli
    dla $p \in T$ **wykonuj**
      $I(p) = I(p) + q$
    koniec pętli
  koniec jeśli
koniec dopóki

2.2 Znajdowanie skojarzenia doskonałego w grafie bez wag

Algorytm polega na iteracyjnym znajdowaniu ścieżki między dwoma dowolnymi wierzchołkami nienasyconymi takiej, że krawędzie na niej na zmianę nie należą i należą do skojarzenia. Po znalezieniu takiej ścieżki dodajemy do skojarzenia M znajdujące się na niej nienasycone krawędzie i usuwamy krawędzie nasycone, które znajdują się na ścieżce.

Algorytm kończy się po znalezieniu skojarzenia doskonałego lub po znalezieni zbiorów $S \subseteq L$ i $T \subseteq P$, które nie spełniają warunku Halla (3).

2.2.1 Pseudokod

Algorytm 2 Znajdowanie skojarzenia doskonałego w grafie bez wag

1: **jeśli** M niezdefiniowane **wtedy**
2: $M = \{\}$
3: **koniec jeśli**
4: // *powieksz* zwraca prawdę, jeśli powiększono
5: // procedura opisana w Algorytmie 3
6: **dopóki** *powieksz*(M) **wykonuj**
7: **jeśli** $|M| == |L|$ **wtedy**
8: Zwróć znalezione skojarzenie M
9: **w przeciwnym wypadku**
10: Nie istnieje doskonałe skojarzenie, zwróć znalezione M oraz zmienne funkcji *powieksz* S, T
11: **koniec jeśli**
12: **koniec dopóki**

Algorytm 3 Znajdowanie ścieżki powiększającej

1: $S = \{\}$
2: $T = \{\}$
3: Weź dowolny wierzchołek $l_{start} \in L - M(L)$ i dodaj go do S jako nieodwiedzony
4: **dopóki** S zawiera nieodwiedzone wierzchołki **wykonuj**
5: Wybierz pierwszy nieodwiedzony wierzchołek l z S i oznacz jako odwiedzony
6: **dla** dla każdego $p \in N(l) - T$ **wykonuj**
7: Dopisz p do T
8: Dodaj p do drzewa D jako dziecko l
9: **jeśli** p nie jest wierzchołkiem nasyconym **wtedy**
10: Powiększ skojarzenie M o ścieżkę z l_{start} do p w drzewie D.
11: Program zakończył się sukcesem, zwróć $true$
12: **w przeciwnym wypadku**
13: znajdź takie l, że $lp \in M$
14: Dopisz l do S i oznacz jako nieodwiedzony
15: **koniec jeśli**
16: **koniec pętli**
17: **koniec dopóki**
18: Graf nie posiada ścieżki powiększającej, zwróć $false$

3 Analiza poprawności

Skupmy się najpierw na Algorytmie 1. Kroki 2–4 stanowią główną pętlę programu. Przyjrzyjmy się bardziej szczegółowo krokowi 4.1. Modyfikuje on wagi

w taki sposób, że funkcja I pozostaje potencjałem, tzn. własność (7) pozostaje prawdziwa. Są następujące przypadki:

1. $l \in S,\ p \in P - T$
   Z definicji $T = N_{G_I}(S)$, więc $lp \notin G_I$. Po zmianie potencjałów otrzymamy $I(l) \leftarrow I(l) - q$ i $I(p) \leftarrow I(p)$. q jest tak zdefiniowane, że nierówność (7) jest nadal spełniona. Dla przypomnienia:

   $$q = min\{I(l) + I(p) - w(l,p) : l \in S, p \in P - T\}$$

   z czego dalej wynika:

   $$\forall_{l \in S, p \in P-T}\ q \leqslant I(l) + I(p) - w(l,p)$$

   po odjęciu q otrzymamy:

   $$\forall_{l \in S, p \in P-T}\ I(l) - q + I(p) - w(l,p) \geqslant 0$$

   Ponadto, istnieje co najmniej jedno takie lp, które pojawi się jako nowa krawędź w G_I.

2. $l \in S,\ p \in T$
   Po zmianie potencjałów otrzymamy $I(l) \leftarrow I(l) - q$ i $I(p) \leftarrow I(p) + q$. $I(l) + I(p)$ się nie zmieni, więc krawędzie, które były w G_I i spełniały te warunki, pozostaną w G_I.

3. $l \in L - S,\ p \in P - T$
   Analogicznie jak wyżej, suma potencjałów się nie zmieni, więc przynależność krawędzi do G_I pozostanie taka sama.

4. $l \in L - S,\ p \in T$
   W tym przypadku $I(l) + I(p)$ zwiększy się o q. Wszystkie takie krawędzie zatem zostaną usunięte z G_I. Nie wpłynie to jednak na budowę drzewa M-przemiennego, gdyż drzewo to zawiera tylko krawędzie między S a T.

Warto zauważyć, że każda krawędź, która jest w M, pasuje do przypadku 2 lub 3. Dla każdego $lp \in M$, albo p zostało napotkane przez drzewo, wtedy $p \in T$, a więc $l \in S$, co wynika ze sposobu konstrukcji drzewa M-przemiennego (linia 14 Algorytmu 3); albo $p \in P - T$, wtedy analogicznym rozumowaniem $l \notin S$, bo gdyby $l \in S$, to wtedy p zostałby dołączony do zbioru T (również linia 14 Algorytmu 3). Patrząc na te przypadki, można stwierdzić, że nigdy nie usuniemy krawędzi z M.

Do stwierdzenia całkowitej poprawności algorytmu, wystarczy pokazać, że ma on własność stopu i własność częściowej poprawności.

3.1 Własność stopu

Jeżeli nie istnieje żadne skojarzenie doskonałe w grafie G, algorytm kończy się w kroku 0.

W przeciwnym wypadku, zauważmy, że każda iteracja głównej pętli algorytmu (kroki 3-5) wpływa na graf w następujący sposób:

1. Nie jest usunięta krawędź z M. Zatem $|M^{i+1}| \geqslant |M^i|$, gdzie M^i oznacza skojarzenie M po i-tym powtórzeniu kroku 2.

2. Albo zwiększy się rozmiar M, albo zwiększy się rozmiar T.

Zbiór T może mieć ograniczoną liczbę elementów, zatem w pewnym momencie, przy odpowiednio dużej liczbie elementów T, znajdziemy choć jedną ścieżkę M-zasiloną i zwiększy się rozmiar M. Algorytm zakończy się, gdy $|M| = |L|$, co wydarzy się w skończonej liczbie kroków.

Żeby zobaczyć, że w przypadku niepowiększenia M zwiększy się rozmiar T, zwróćmy uwagę na następujący fakt. W zbiorze S znajdą się te same (wszystkie) nienasycone wierzchołki z L, jako że M się nie zmieniło. Na podstawie procesu tworzenia drzewa (linia 14 Algorytmu 3) można stwierdzić, że zostaną odwiedzone wszystkie wierzchołki z poprzednich zbiorów S i T, bo żadna krawędź między S a T nie została usunięta. Innymi słowy, jeśli S_i oraz T_i to zbiory na koniec aktualnej, i-tej iteracji, wtedy $S_{i-1} \subset S_i$ oraz $T_{i-1} \subset T_i$. Zbiór $T = N_{G_I}(S)$ wszystkich sąsiadów wierzchołków S w grafie G_I w najgorszym przypadku zwiększy się o 1, gdyż modyfikacja potencjałów o q wprowadzi co najmniej jedną nową krawędź z S do poprzedniego $P - T$, a jednocześnie nie usunie żadnych krawędzi między S a poprzednim T. Wynika z tego, że T_i w każdym takim kroku, gdzie M się nie zmienia, będzie większe od T_{i-1}.

3.2 Własność częściowej poprawności

Wykazaliśmy we wstępie do tej sekcji, że funkcja I przez cały czas trwania algorytmu pozostaje potencjałem. Algorytm dobiega do końca, gdy $|M| = |L|$, lub gdy nie istnieje skojarzenie doskonałe dla grafu. Pozostało więc pokazać, że znalezione na końcu skojarzenie M jest skojarzeniem o największej możliwej wadze (w naszej implementacji zmieniamy znaki wag, by efektywnie rozwiązać problem minimalizacji) oraz, że Algorytm 2, wywoływany w kroku 2, znajdzie skojarzenie doskonałe dla grafu G_I, jeżeli takie istnieje.

3.2.1 Dowód, że M jest skojarzeniem o największej wadze

Oznaczmy jako M^* skojarzenie doskonałe o największej wadze, podczas gdy M jest znalezionym przez nas skojarzeniem doskonałym dla grafu G_I w ostatniej iteracji algorytmu, czyli $|M| = |L|$. Natomiast w to funkcja zwracająca łączną wagę. Wtedy:

$$w(M) = \sum_{lp \in M} w(l,p) = \sum_{lp \in M} (I(l) + I(p)) = \sum_{l \in L} I(l) + \sum_{p \in P} I(p)$$

oraz:

$$w(M^*) = \sum_{lp \in M^*} w(l,p) \leqslant \sum_{lp \in M^*} (I(l) + I(p)) = \sum_{l \in L} I(l) + \sum_{p \in P} I(p)$$

co daje nam:

$$w(M^*) \leqslant w(M)$$

a jednocześnie wiemy, że $w(M^*) \geqslant w(M)$, bo M^* jest skojarzeniem o największej wadze. Otrzymujemy $w(M^*) = w(M)$.

3.2.2 Dowód dla Algorytmu 2

Każde przejście zewnętrznej pętli albo powoduje zwiększenie M, albo stwierdza, że skojarzenie doskonałe nie jest możliwe. Pętla wewnętrzna dodaje stopniowo elementy do S i T w miarę powiększania drzewa M-przemiennego. Zaczynamy od jednego elementu w S, a następnie dla każdego dodanego nowego elementu do T, dodajemy też jego sąsiada w M do S. Cały czas utrzymujemy $|S| > |T|$, dopóki nie znajdziemy ścieżki M-zasilonej. Zawsze zatem, gdy stwierdzamy, że skojarzenie doskonałe nie istnieje, otrzymujemy $T = N_{G_I}(S)$ oraz $|S| > |T|$. Toteż mamy taki podzbiór $S \subset L$ dla którego $|S| > |N_{G_I}(S)|$. Z twierdzenia Halla[1] dzieje się tak wtedy i tylko wtedy, gdy nie istnieje skojarzenie doskonałe w grafie G_I.

4 Analiza złożoności czasowej

Przeanalizujemy złożoność czasową każdego z kroków algorytmu z osobna.

0. Ten krok ma złożoność taką samą, jak krok 2, bo wywoływany jest ten sam algorytm. Przygotowanie grafu G_{bez_wag} zajmie $\mathcal{O}(n^2)$ operacji. Algorytm 2 również zajmie $\mathcal{O}(n^2)$ operacji. Razem daje to nam złożoność $\mathcal{O}(n^2)$.

1. W kroku pierwszym wykonywane jest $\mathcal{O}(n^2)$ operacji (liczba wierzchołków w L razy złożoność czasowa znalezienia maksimum równa $\mathcal{O}(n)$).

2. W drugim kroku znajdujemy skojarzenie dla wybranego grafu G_I. W najgorszym przypadku zajmie to $\mathcal{O}(n^2)$ operacji. Sama budowa grafu G_I również zajmuje $\mathcal{O}(n^2)$ operacji (przyjmując, że za każdym razem budujemy od początku).

3. Trzeci krok jest rzędu $\mathcal{O}(n)$.

4. Czwarty krok zajmie $\mathcal{O}(n^2)$ operacji, żeby znaleźć minimalną krawędź i $\mathcal{O}(n)$, żeby zmodyfikować potencjały. Razem $\mathcal{O}(n^2)$.

Należy pamiętać, że krok 1 wykonywany jest jednokrotnie, a kroki 2–4 mogą się powtarzać. W najgorszym przypadku powtórzą się $\mathcal{O}(n^2)$ razy. W najgorszym przypadku, każde zwiększenie rozmiaru M o jeden zajmie $\mathcal{O}(n)$ operacji. Będzie tak, ponieważ po zwiększeniu M zmienia się kształt drzewa M-przemiennego i następne $T = \{\}$. Każde kolejne wywołanie 4.1 zmieni G_I w taki sposób, że S będzie miał jednego nowego sąsiada, więc $|T_{i+1}| = |T_i| + 1$. T może urosnąć maksymalnie do rozmiaru n, zanim zostanie znaleziona ścieżka M-zasilona i zwiększy się rozmiar M. Podobnie w najgorszym przypadku n razy będzie trzeba zwiększać M. Razem daje nam to $\mathcal{O}(n^2)$ powtórzeń sekwencji 2–4.

Daje to nam łączną złożoność:

$$\mathcal{O}(n^2 + n^2 + n^2 \cdot (n^2 + 1 + n^2)) = \mathcal{O}(n^4)$$

5 Opis wejścia i wyjścia

Na wejściu program przyjmuje ścieżkę do pliku o formacie zdefiniowanym w podsekcji 5.1. Plik wejściowy programu musi być rozszerzeniem `.txt` Wyjściem programu jest plik o nazwie takiej samej, jak plik wejściowy z dopiskiem `-out.txt`, format pliku wyjściowego został zdefiniowany w podsekcji 5.2.

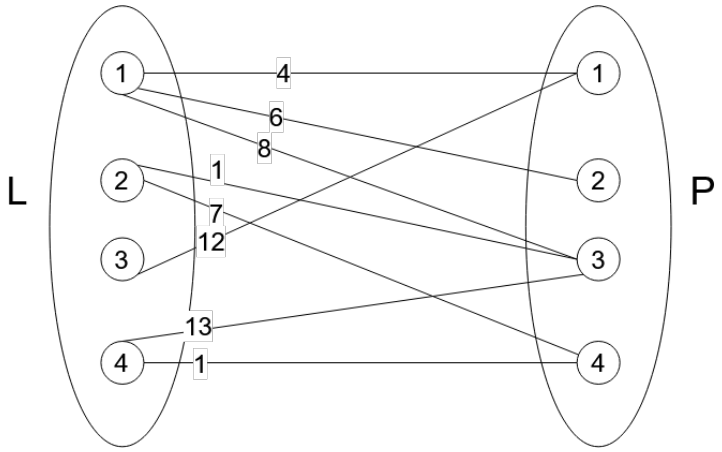
5.1 Format pliku wejściowego

W pierwszej linijce pliku wejściowego będzie się znajdować liczba wierzchołków w każdym rozłącznym zbiorze wierzchołków, zbiory oznaczmy jako **P**, **L**.

Kolejne linijki w pliku będą oznaczały połączenia w grafie w następujący sposób:

- Numer linijki odpowiada numerowi wierzchołka ze zbioru **L**.
- Liczba i-ta w linijce j-tej oznaczają wagę krawędzi między wierzchołkiem v_j ze zbioru **L** a wierzchołkiem v_i ze zbioru **P**.
- Litera "n" na miejscu i-tym w linijce j-tej oznacza, że nie istnieje krawędź między wierzchołkiem v_j ze zbioru **L** a wierzchołkiem v_i ze zbioru **P**.

Graf postaci:



Rysunek 1: Przykładowy graf

W pliku wejściowym zapisany będzie w następujący sposób:

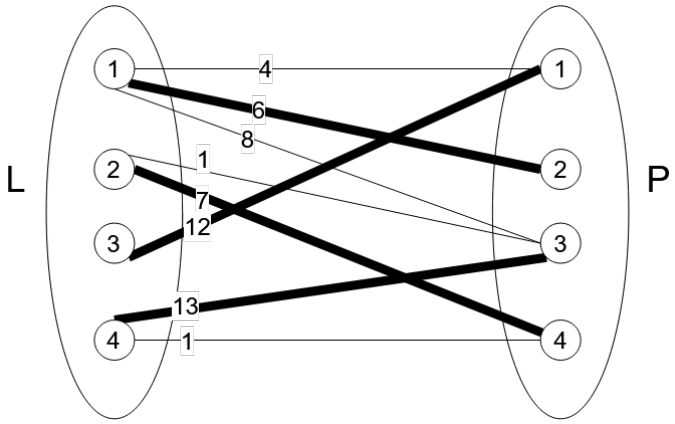
4			
4	6	8	n
n	n	1	7
12	n	n	n
n	n	13	1

Tabela 1: Przykładowy format pliku wejściowego

5.2 Format pliku wyjściowego

Format pliku wyjściowego jest identyczny z formatem pliku wyjściowego, z tą różnicą, że literą "n" oznaczamy brak krawędzi oraz krawędzie które nie należą do skojarzenia doskonałego. Dodatkowo w drugiej linijce pliku znajdować się będzie waga znalezionego skojarzenia.

Dla znalezionego skojarzenia doskonałego oznaczonego pogrubionymi liniami:



Rysunek 2: Skojarzenie doskonałe o minimalnej wadzę

format pliku wyjściowego jest następujący:

4			
38			
n	6	n	n
n	n	n	7
12	n	n	n
n	n	13	n

Tabela 2: Przykładowy format pliku wyjściowego

Gdy graf nie posiada skojarzenia doskonałego plik zwraca aktualne największe skojarzenie, wagę oraz komunikat "nie znaleziono doskonałego":

nie znaleziono doskonałego
4
38
n 6 n n
n n n 7
n n n n
n n n n

Tabela 3: Przykładowy format pliku wyjściowego

Bibliografia

[1] Katarzyna Rybarczyk-Krzywdzińska. *Algorytmy Grafowe Rozdział 8. Skojarzenia w grafach dwudzielnych.* http://kryba.home.amu.edu.pl/2020latoAGR/wyklady/. Materiały z wykładu; Data dostępu 09-04-2024. Maj 2020.