Badania Operacyjne
wykład 3

Badania Operacyjne – I st. II rok AiR
Katedra Automatyki i Robotyki
Laboratorium Badań Operacyjnych i Systemowych

Problemy grafowe

Akademia Górniczo-Hutnicza w Krakowie
Wydział Elektrotechniki, Automatyki, Informatyki i Elektrotechniki

Zagadnienia

- Problemy grafowe – definicje modeli
  - Zagadnienie komiwojażera (ang. Traveling Salesman)
  - Najdłuższa ścieżka (ang. Longest Path)
  - Pokrycie wierzchołkowe (ang. Vertex Cover)
  - Zbiór dominujący (ang. Dominating Set)
  - Skojarzenie (ang. Matching)
  - Podział grafu (ang. Graph Partitioning)
  - Kolorowanie grafu (ang. Graph Coloring)
  - Izomorfizm grafu (ang. Graph Isomorphism)
  - Zagadnienie przydziału (ang. Assigment Problem)
- Algorytmy dedykowane
  - Algorytmy zachłanne dla TSP

Model matematyczny

Model matematyczny procesu (którym może być zjawisko fizyczne, proces technologiczny, system ekonomiczny, system produkcji, transportu itd.) składa się z:

zbioru zmiennych decyzyjnych (zmiennych projektowych, zmiennych „manipulacyjnych") oraz pozostałych parametrów opisujących problem,

funkcji celu (lub zbioru funkcji celów), będącej matematycznym zapisem kryterium optymalizacyjnego,

zbioru ograniczeń (warunków ograniczających).

Klasyfikacja modeli matematycznych

Model deterministyczny – analityczne przedstawienie pojęcia, systemu lub działań, w którym dla danych wielkości wejściowych wyniki są określone jednoznacznie.

Model niedeterministyczny – model, w którym powiązania funkcyjne zależą od wielkości losowych. Dla danych wielkości wejściowych wyniki mogą być jedynie przewidziane zgodnie z zasadami probabilistyki.

Model wartości oczekiwanych – model, w którym wielkościom losowym zostały nadane ich wartości oczekiwane.

Klasyfikacja modeli matematycznych

Ze względu na charakter zbioru zmiennych decyzyjnych:

Model optymalizacji dyskretnej, gdy zbiór zmiennych decyzyjnych jest skończonym zbiorem wartości dyskretnych, np. zgodnych z normami.

Model optymalizacji ciągłej, bez ograniczenia zakresu zmiennych.

Klasyfikacja modeli matematycznych

Ze względu na liczbę funkcji celów (kryteriów optymalizacyjnych):

Model optymalizacji skalarnej, gdy zadanie wykorzystuje tylko jedną funkcję celu.

Model optymalizacji wielokryterialnej (wektorowej), z kilkoma funkcjami celu.

Klasyfikacja modeli matematycznych

Ze względu na rodzaj funkcji celu oraz ograniczeń:

Model liniowy, gdy zarówno funkcja celu, jak i wszystkie ograniczenia są funkcjami liniowymi.

Model nieliniowy, gdy funkcja celu lub chociaż jedno z ograniczeń ma charakter nieliniowy.

Dyskretny – zmienne decyzyjne są dyskretne

Permutacyjny - zmienne decyzyjne są permutacją pewnego zbioru

Model matematyczny

Tworząc model, należy wykorzystywać podejście systemowe.

Budowa modelu obejmuje:

- określenie zmiennych decyzyjnych,
- określenie funkcji celu,
- określenie obszaru dopuszczalnego (obszaru rozwiązań dopuszczalnych).

Zagadnienie komiwojażera - TSP

polega na znalezieniu minimalnego cyklu Hamiltona w grafie ważonym.

Nazwa pochodzi od wędrownego sprzedawcy (komiwojażera), który ma odwiedzić dokładnie jeden raz każde z *n* miast (gdzie znana jest odległość pomiędzy miastami) i wrócić do miasta początkowego.

Dany jest

- zbiór wierzchołków $N=\{1,\ldots,n\}$
- macierz odległości $(n \times n)$-wymiarowa $A=[a_{i,k}]$

Należy znaleźć permutację $\pi = (\pi(1),\ldots,\pi(n))$ elementów zbioru N, która minimalizuje funkcję celu $\phi(\pi)$:

$$\phi(\pi) = \sum_{i=1}^{n-1} a_{\pi(i)\pi(i+1)} + a_{\pi(n)\pi(1)}$$

Cykl Hamiltona

- to taki cykl w grafie, w którym każdy wierzchołek grafu występuje jeden raz. Znalezienie cyklu Hamiltona o minimalnej sumie wag krawędzi jest równoważne rozwiązaniu zagadnienia komiwojażera.
- Grafy zawierające cykl Hamiltona nazywamy **hamiltonowskimi**.
- **Cykl Eulera** to taka zamknięta droga w grafie, która przechodzi przez każdą jego krawędź dokładnie jeden raz.
- Grafy zawierające cykl Eulera nazywamy eulerowskimi.
- **Symetryczne zagadnienie komiwojażera** (STSP) polega na tym, że odległość z miasta A do B oraz z miasta B do A jest zawsze taka sama.
- **Asymetryczne zagadnienie komiwojażera** (ATSP) odległość z miasta A do B może być inna, niż odległość z miasta B do A.
- **Otwarte zagadnienie komiwojażera** (OTSP) – nie tworzy cyklu, wierzchołek początkowy i końcowy nie jest zadany

Zagadnienie komiwojażera - TSP

Dany jest

- zbiór wierzchołków $N=\{1,\ldots,n\}$
- macierz odległości ($n \times n$) - wymiarowa $C=[c_{i,j}]$

Należy znaleźć rozwiązanie X, które minimalizuje funkcję celu:

$$\min \quad f(X) = \sum_{i=1}^{m} \sum_{j=1}^{m} c_{ij} x_{ij}$$

$$\text{ogr.} \quad \sum_{i=1}^{n} x_{ij} = 1 \quad \text{dla } j = 1, \ldots, n$$

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \text{dla } i = 1, \ldots, n$$

cykl Hamiltona

$$\text{gdzie} \quad x_{ij} = \{0, 1\}$$

Zagadnienie komiwojażera - TSP

Definicja zadania programowania liniowego całkowitoliczbowego - formalizacja Miller-Tucker-Zemlin (MTZ):

Należy znaleźć rozwiązanie X:

$$x_{ij} = \begin{cases} 1 & \text{gdy droga zawiera odcinek z } i \text{ do } j \\ 0 & \text{w przeciwnym przypadku (wpp)} \end{cases}$$

$$\min \sum_{i=1}^{n} \sum_{j \neq i, j=1}^{n} c_{ij} x_{ij}:$$

$$x_{ij} \in \{0,1\} \qquad i,j = 1,\ldots,n;$$

$$u_i \in \mathbf{Z} \qquad i = 2,\ldots,n;$$

$$\sum_{i=1, i \neq j}^{n} x_{ij} = 1 \qquad j = 1,\ldots,n;$$

$$\sum_{j=1, j \neq i}^{n} x_{ij} = 1 \qquad i = 1,\ldots,n;$$

$$u_i - u_j + n x_{ij} \leq n - 1 \qquad 2 \leq i \neq j \leq n;$$

$$0 \leq u_i \leq n - 1 \qquad 2 \leq i \leq n.$$

Zagadnienie komiwojażera - TSP

Definicja zadania programowania liniowego całkowitoliczbowego - formalizacja Dantzig-Fulkerson-Johnson (DFJ):

Należy znaleźć rozwiązanie X:

$$x_{ij} = \begin{cases} 1 & \text{gdy droga zawiera odcinek z } i \text{ do } j \\ 0 & \text{w przeciwnym przypadku (wpp)} \end{cases}$$

$$\min \sum_{i=1}^{n} \sum_{j \neq i, j=1}^{n} c_{ij} x_{ij} :$$

$$0 \leq x_{ij} \leq 1 \qquad i, j = 1, \ldots, n;$$

$$\sum_{i=1, i \neq j}^{n} x_{ij} = 1 \qquad j = 1, \ldots, n;$$

$$\sum_{j=1, j \neq i}^{n} x_{ij} = 1 \qquad i = 1, \ldots, n;$$

$$\sum_{i \in Q} \sum_{j \in Q} x_{ij} \leq |Q| - 1 \qquad \forall Q \subsetneq \{1, \ldots, n\}, |Q| \geq 2$$

Algorytmy dla TSP

Algorytmy dokładne – tylko dla instancji o małym rozmiarze:

- Przegląd zupełny
- Programowanie dynamiczne,
- Metoda podziału i ograniczeń – alg. Little'a
- Alg. Eastmana - relaksacja problemu TSP do AP

Algorytmy przybliżone – rozwiązanie suboptymalne w akceptowanym czasie:

- Alg. najbliższego sąsiada (NN)
- Alg. GTSP, FARIN, MEARIN, …
- Alg. przeszukiwania losowego
- Alg. Ewolucyjne
- ….

TSP – algorytm najbliższego sąsiada (NN)

- Algorytm zachłanny
- Złożoność $o(n^2)$ – rozwiązanie nieoptymalny.
- Dolne ograniczenie dla instancji.

Etapy algorytmu:

1. Start z dowolnego wierzchołka (wierzchołek aktualny), który markujemy jako odwiedzony (wstawienie do rozwiązania).
2. Znajdź krawędź o najmniejszej wadze łączącą wierzchołek aktualny z nieodwiedzonymi wierzchołkami - v.
3. Przejdź do v (wierzchołek aktualny).
4. Oznacz v jako odwiedzony (wstawienie do rozwiązania).
5. Jeżeli wszystkie wierzchołki V są odwiedzane: STOP – zwróć sekwencję odwiedzonych wierzchołków.
6. Idź do kroku 2 .

TSP – algorytm NN

Macierz A	1	2	3	4	5
1	∞	5	4	6	6
2	8	∞	5	3	4
3	4	3	∞	3	1
4	8	2	5	∞	6
5	2	2	7	0	∞

Wybrany wierzchołek początkowy - 1

Kolejność odwiedzonych wierzchołków: <1-3-5-4-2-1>

Funkcja celu: 4+1+0+2+8=15.

- Repetitive Nearest Neighbour Algorithm – algorytm najbliższego sąsiada uruchamia się dla każdego wierzchołka początkowego i wybiera najlepsze z uzyskanych rozwiązań.

Algorytm zachłanny – G_TSP

G_TSP – ang. *greed TSP*

Krok 1: Uporządkuj łuki (krawędzie) wg wag w ciąg niemalejący:

$$a_{e_1} \leq a_{e_2} \leq \ldots \leq a_{e_n}$$

gdzie : $e_j \in E$
m - liczba łuków, tzn. $m = |E|$

Krok 2: Dla j=1 do m wykonaj:

dołącz e_j do rozwiązania, jeżeli nie powoduje to powstania podcyklu (podkonturu)

- Złożoność obliczeniowa *kroku 1* – sortowanie: *O(m*log m)*
- Złożoność obliczeniowa *kroku 2* – *O(m)*
- Dla zagadnienia asymetrycznego A_TSP: $m \leq (n-1)*n$
- Dla zagadnienia symetrycznego S_TSP: $m \leq 1/2*(n-1)*n$

TSP – FARIN

- *Farthest Insertion Heuristik*
- Algorytm zachłanny – wstawienia najdalszego wierzchołka
- Złożoność $o(n^2)$ – rozwiązanie nieoptymalny.
- Dolne ograniczenie dla instancji.

1. Algorytm zaczyna się od rozwiązania (drogi komiwojażera), które składa się z jednego, losowo wybranego węzła.
2. Wybierz "najdroższy" wierzchołek v (wierzchołek nieoznaczony najdalszy od aktualnej trasy)
3. Wstaw v do rozwiązania w "najtańszym" miejscu sekwencji
4. Jeżeli wszystkie wierzchołki V są odwiedzane: STOP – zwróć sekwencję odwiedzonych wierzchołków.
5. Idź do kroku 2.

TSP – algorytm FARIN

A	1	2	3	4	5
1	∞	5	4	6	6
2	8	∞	5	3	4
3	4	3	∞	3	1
4	8	2	5	∞	6
5	2	2	7	0	∞

Wybrany wierzchołek początkowy – 1

Wierzchołek – 4: <1-4-1>=6+8=14

Wierzchołek – 5: <1-4-5-1>=6+6+2=14
<1-5-4-1>=6+0+8=14

Wierzchołek – 3: <1-3-4-5-1>=4+3+6+2=15
<1-4-3-5-1>=6+5+1+2=14
<1-4-5-3-1>=6+6+7+4=23

Wierzchołek – 2: <1-2-4-3-5-1>=5+5+3+6+2=23
<1-4-2-3-5-1>=6+2+5+1+2=16
<1-4-3-2-5-1>=6+5+3+4+2=20
<1-4-3-5-2-1>=6+5+1+2+8=22

Rozwiązanie: <1-4-2-3-5-1>

Funkcja celu: 16

TSP – NEARIN

- *Nearest-Insertion-Heuristik*
- Algorytm zachłanny – wstawienia najbliższego wierzchołka
- Złożoność $o(n^2)$ – rozwiązanie nieoptymalny.
- Dolne ograniczenie dla instancji.

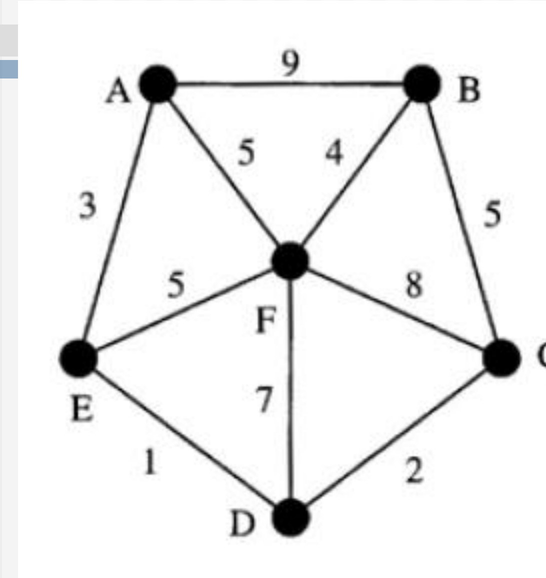
1. Algorytm zaczyna się od rozwiązania (drogi komiwojażera), które składa się z jednego, losowo wybranego węzła.
2. Wybierz najbliższy wierzchołek v (wierzchołek nieoznaczony najbliższy do wierzchołków rozwiązania)
3. Wstaw v do rozwiązania w "najtańszym" miejscu sekwencji
4. Jeżeli wszystkie wierzchołki V są odwiedzane: STOP – zwróć sekwencję odwiedzonych wierzchołków.
5. Idź do kroku 2.

- RANDIN - *Random Insertion*

Heurystyki dla TSP

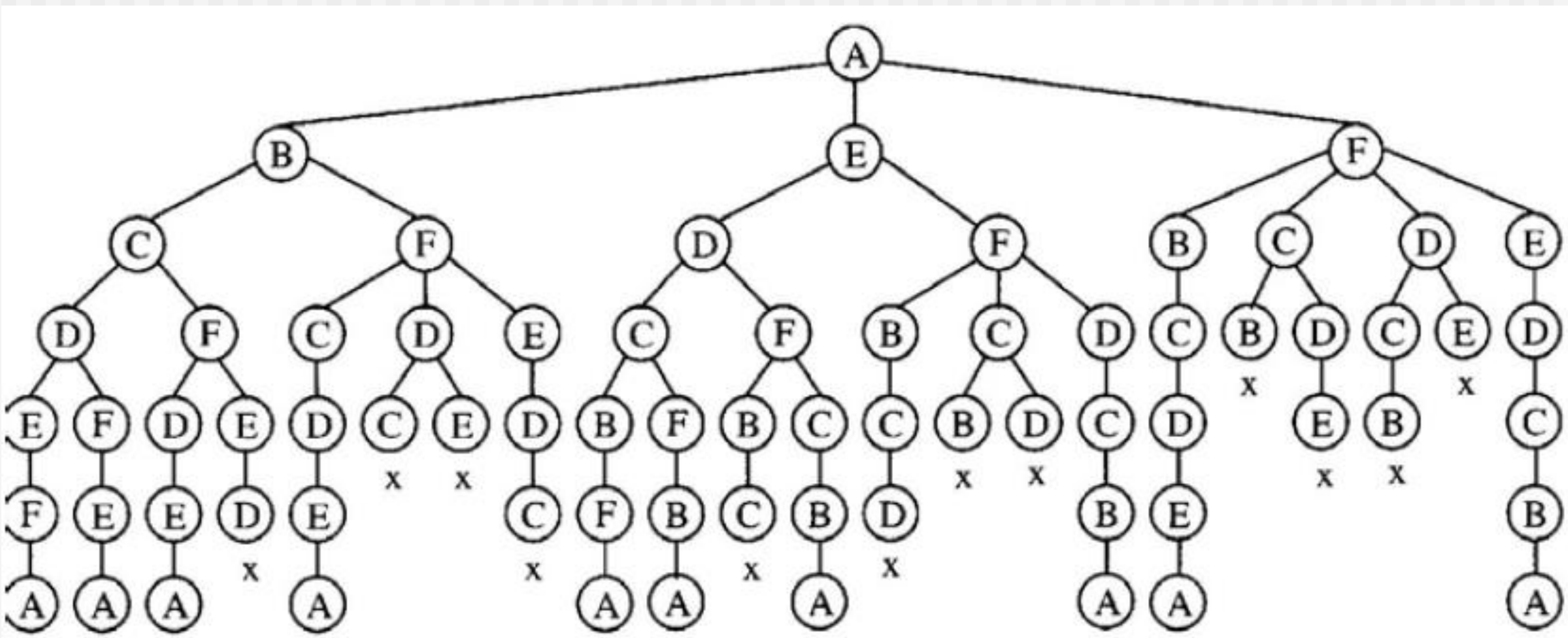
Algorytm (heurystyka)	**Typ**	**Złożoność obliczeniowa**
Nearest-/ Farthest- Neighbor (NN, FN)	Konstrukcyjny	$O(n^2)$
Farthest-Insertion (FARIN)	Konstrukcyjny	$O(n^2)$
Nearest-Insertion (NEARIN)	Konstrukcyjny	$O(n^2)$
Minimum Spanning Tree	Konstrukcyjny	$O(n^2 log(n))$
Heurystyka Christofidesa	Konstrukcyjny	$O(n^3)$
K-opt	Poprawy	$O(k!)$ – każdy krok
Suma *n* najkrótszych krawędzi (LB)	Heurystyki podwójne	$O(n^2 log(n))$
Długość minimalnego drzewa rozpinającego (MST + 2-Matching)	Heurystyki podwójne	$O(n^2 log(n))$

Przeszukiwanie zupełne



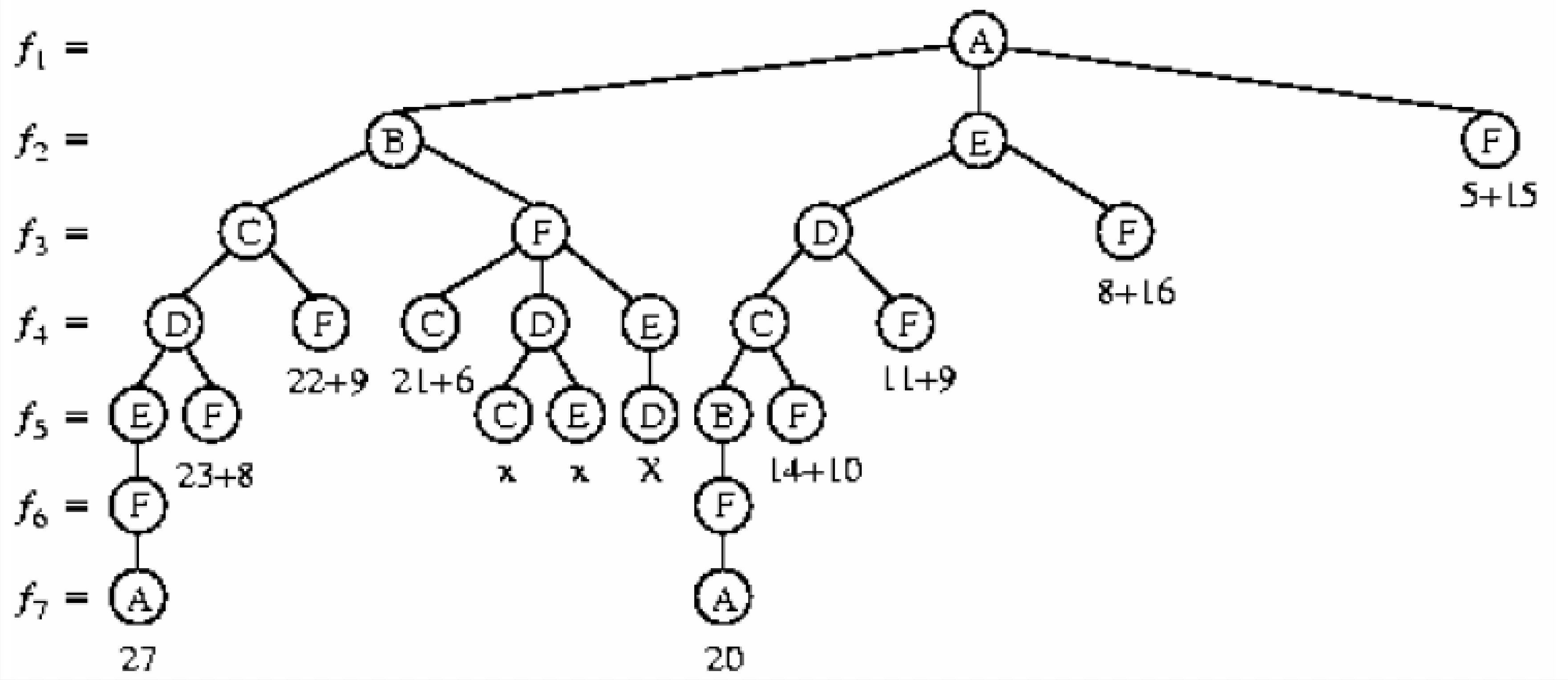
Pełny przegląd przestrzeni rozwiązań:

- Zagadnienie komiwojażera: TSP
- Zadanie testowe: graf
- Drzewo utworzonych rozwiązań
- cc.ee.ntu.edu.tw/~eda/Course/VLSIDesignAuto/LN/complexityOptimize.pdf



Metoda podziału i ograniczeń

- Porównanie drzewa rozwiązań (podproblemów) dla tego samego zadania testowego zagadnienia TSP
- Znalezione rozwiązanie optymalne: **AEDCBFA**
- Wartość funkcji celu: **20**



Złożoność obliczeniowa

Zagadnienie permutacyjne - metoda pełnego przeglądu – 10^{12} rozwiązań przeglądanych w czasie 1 sekundy

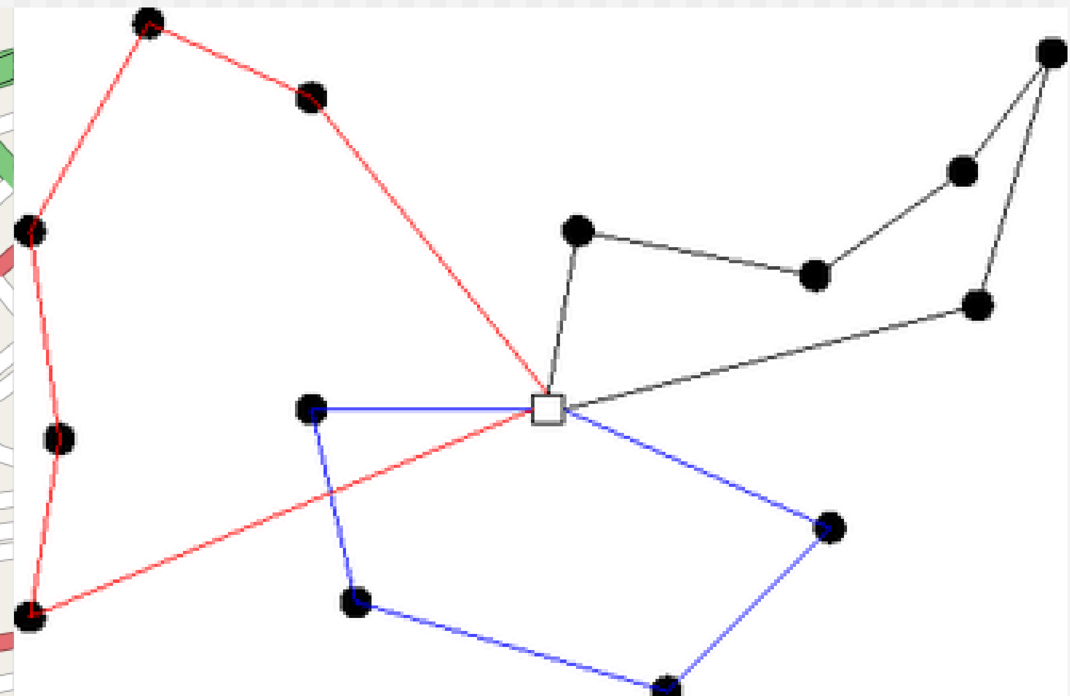
N	
6	**$72*10^{-11}$ s**
10	**$36*10^{-7}$ s**
12	**$15*10^{-5}$ s**
20	**$24*10^{5}$ s (17 dni)**
25	**$16*10^{12}$ s (500 000 lat)**
30	**$27*10^{19}$ s (~$8.5*10^{12}$ lat - 700 *obecny wiek wszechświata)**

Dla problem TSP (NP-trudny) algorytm dokładny ma złożoność $O(n!)$, co praktycznie nie pozwala rozwiązywać instancji o rozmiarze n>20 wierzchołków.

Rozszerzenie TSP – problem VRP

Problem marszrutyzacji:

- Vehicle Routing Problem (VRP)
- wyznaczeniu optymalnych tras przewozowych dla pewnej ściśle określonej liczby środków transportu



Model matematyczny problemu VRP

Funkcja celu:

$$\min C = \sum_{r \in R} \sum_{f \in \Psi} \sum_{g \in \Psi} c_{fg} x_{fgr},$$

gdzie:

r – pojazd należący do zbioru jednorodnych (identycznych) pojazdów R,

f, g – wierzchołki pomiędzy, którymi odbywa się przewóz,

c_{fg} – koszt przewozu pomiędzy wierzchołkami f i g,

x_{fgr} – zmienna binarna określająca, czy pomiędzy wierzchołkami f i g pojazd r wykonuje przewóz.

Ograniczenia – jedna baza początkowa i końcowa:

$\forall_{r \in R} \sum_{g \in \epsilon} x_{0,g,r} = 1$ – dla bazy początkowej,

$\forall_{r \in R} \sum_{f \in \epsilon} x_{f,n+1,r} = 1$ – dla bazy końcowej,

$\forall_{r \in R} \wedge \forall_{f \in \Psi} \sum_{f \in \epsilon} x_{f,z,r} - \sum_{g \in \epsilon} x_{z,g,r} = 0$ – dla wierzchołków pośrednich.

Model matematyczny problemu VRP

Ograniczenia – jeden pojazd dla klienta, dostawy nie są dzielone:

$\forall_{f\in\Psi} \sum_{g\in\epsilon} \sum_{r\in R} x_{fgr} = 1$ – warunek przypisania dokładnie jednego pojazdu,

$\forall_{f\in\epsilon} \wedge \forall_{g\in\epsilon} \wedge \forall_{r\in R}\ x_{fgr} \in \{0,1\}$ – warunek niedzielonych dostaw.

Ograniczenia – pojemności poszczególnych środków transportu (problem CVRP):

$$\forall_{r\in R} \sum_{f\in\Psi} d_f \sum_{g\in\Psi} x_{fgr} \leqslant m_r,$$

gdzie:

d_f – popyt przypisany do danego klienta,

m_r – pojemność pojazdów.

Model matematyczny problemu VRP

Ograniczenia - w problemach z oknami czasowymi:

$$\forall_{r \in R} \wedge \forall_{f \in \Psi} \wedge \forall_{g \in \Psi} \; x_{fgr}(t_{fr} + t_{fg} - t_{gr}) \leqslant 0,$$

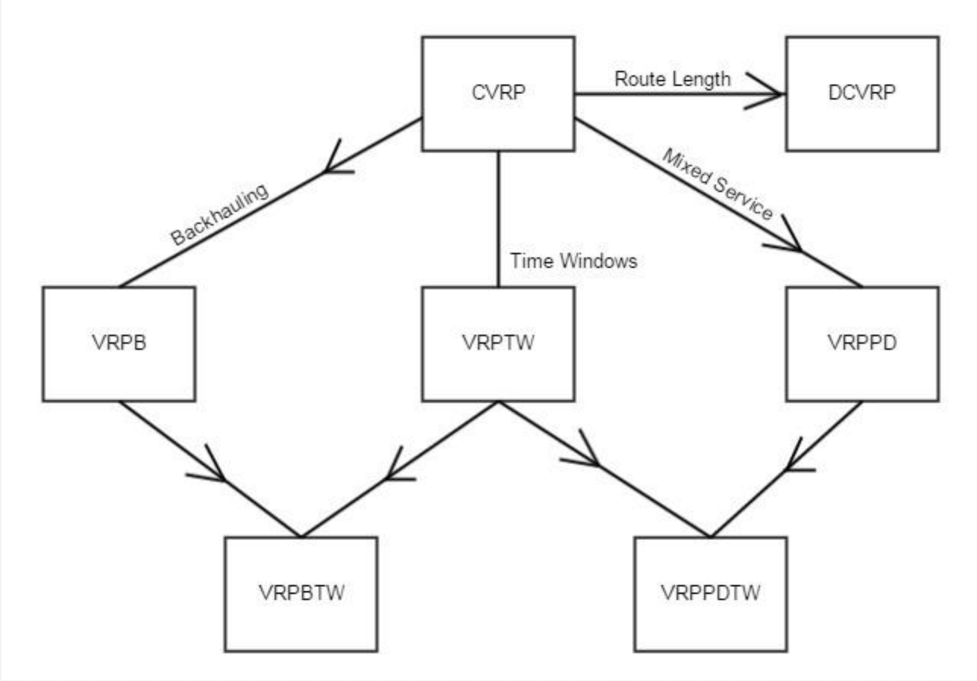
gdzie:

t_{fr} – czas rozpoczęcia obsługi klienta f,

t_{fg} – czas przejazdu pomiędzy f a g,

t_{gr} – czas rozpoczęcia obsługi klienta g.

Ograniczenia generują klasy problemów VRP:



Rozszerzenia problemu VRP

- problemy uwzględniające niesymetryczność kosztów przewozu pomiędzy wierzchołkami,
- problemy uwzględniające niehomogeniczność taboru,
- problemy uwzględniające przejazdy drobnicowe (Less Than Truckload),
- problemy uwzględniające ograniczenie maksymalnej długości trasy,
- problemy umożliwiające ustalenie baz (jednej lub kilku), w których pojazdy zaczynają i kończą podróż (Multiple Depot VRP),
- problemy umożliwiające dodanie baz pomocniczych (VRP with Satellite Facilities),
- problemy umożliwiające ustalenie częstotliwości odbioru/dostawy ładunku,
- problemy umożliwiające uwzględnienie okien czasowych (VRP with Time Windows) odbioru/wysłania towaru,

Rozszerzenia problemu VRP

- problemy wiążące problem marszrutyzacji z problemem kontroli zapasów u klientów,
- problemy uwzględniające możliwość obsługi jednego klienta przez kilka pojazdów (Split Delivery VRP),
- problemy w których kosztowa funkcja celu zastąpiona została innymi parametrami (np. czas wykonania zleceń, długość tras, ilość przewiezionego ładunku),
- problemy umożliwiające zdefiniowanie kolejności odwiedzania poszczególnych miejsc oraz opcjonalnego odwiedzania niektórych punktów,
- problemy uwzględniające możliwości zwrotów i wysyłki towarów przez klientów (VRP with Backhauls oraz VRP with Pick-Up and Delivery – problem rozwózkowo-zwózkowy),
- problemy, w których warunki zostały ujęte stochastycznie (Stochastic VRP).

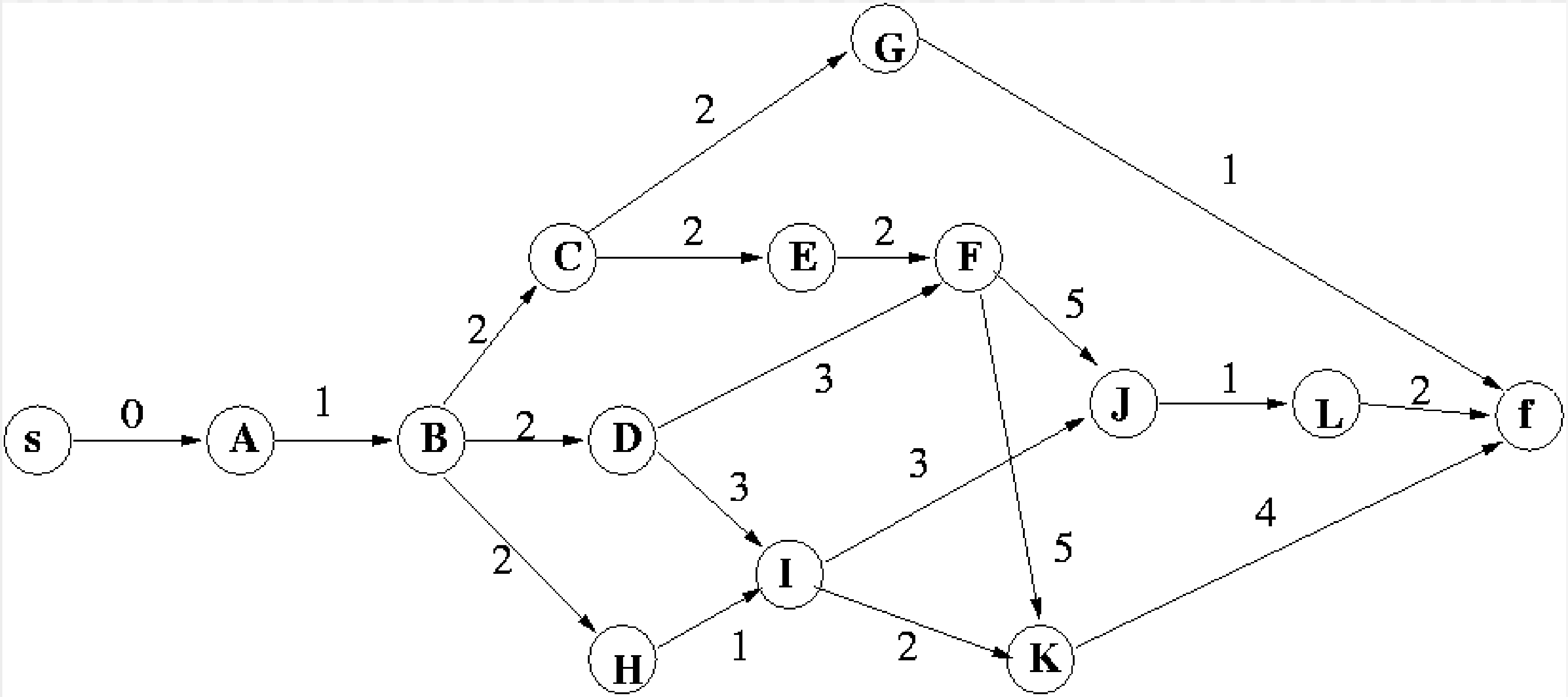
Poszukiwanie najdłuższej ścieżki

- Zagadnienie jest wykorzystywane w metodach programowania sieciowego (CPM, PERT).
- W zarządzaniu projektem problem znalezienia ścieżki krytycznej określa ciąg czynności, których opóźnienie powoduje wydłużenie czasu realizacji całości projektu.
- Przykład: poszukujemy jaki jest minimalny czas produkcji samochodu, gdzie określono zbiór wymaganych czynności, ich następstwo oraz czasy realizacji.

Poszukiwanie najdłuższej ścieżki - cd

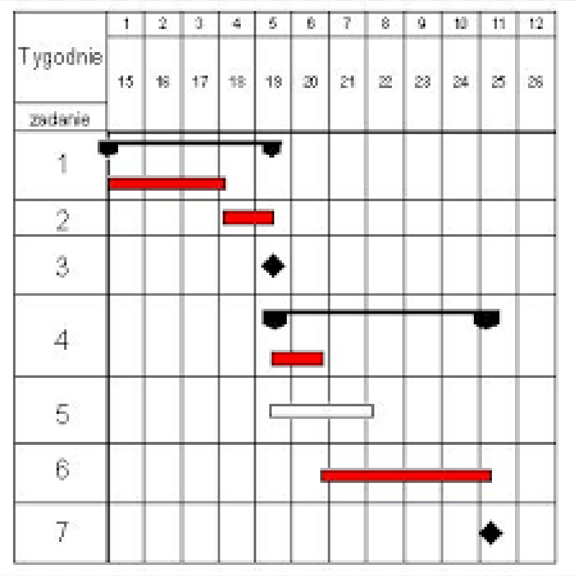
Zadania	*Oznaczenie*	*Czas [h]*	*Poprzednik*
Karoseria	A	1	-
Zawieszenie	B	2	A
Silnik	C	2	B
Skrzynia biegów	D	3	B
Układ kierowniczy	E	2	C
Instalacja elektr.	F	5	D
Koła	G	1	C
Drzwi	H	1	B
Malowanie	I	2	D
Okna	J	2	F
Wyposażenie w.	K	4	F
Test	L	2	J

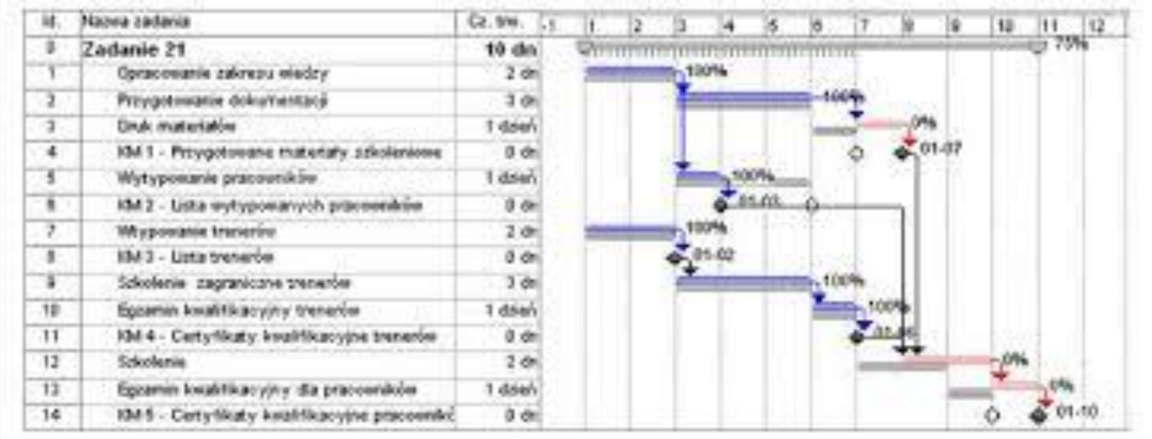
Poszukiwanie najdłuższej ścieżki - cd

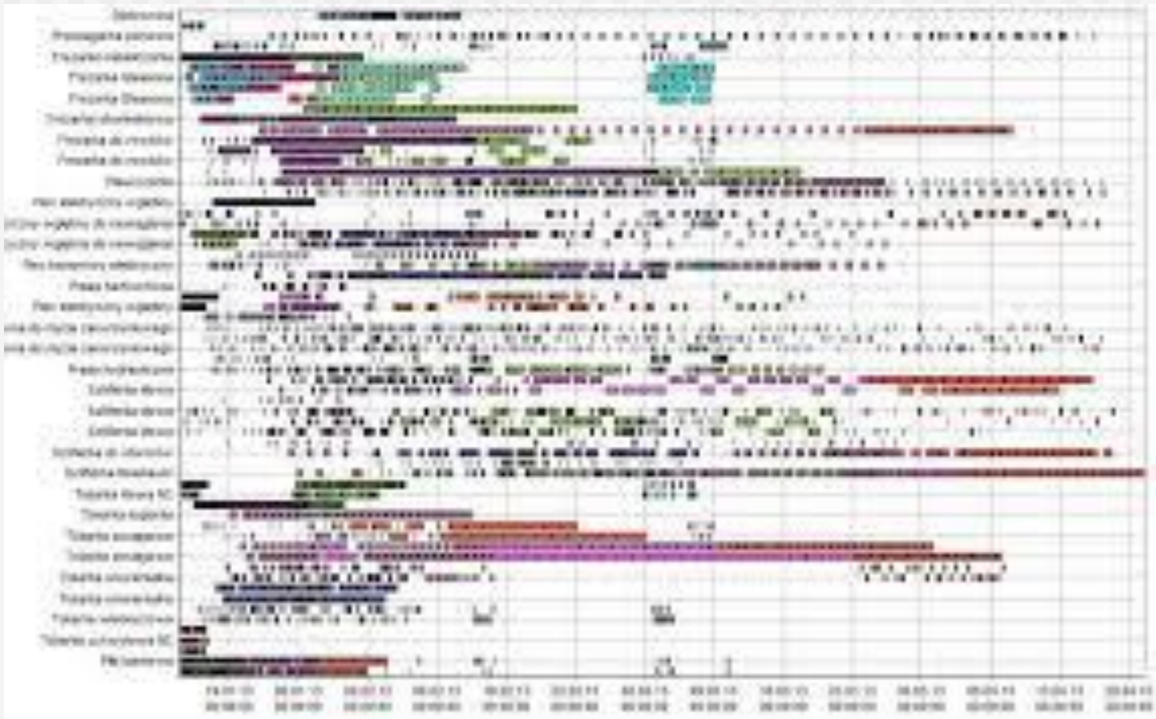


- Wierzchołki grafu skierowanego oznaczają zadania.
- Wprowadzono dwa dodatkowe wierzchołki *s*, *f* – początkowy i końcowy.
- Wierzchołki *u* i *v* są połączone łukiem jeśli zadanie *u* musi być ukończone przed rozpoczęciem *v.*
- Wagi łuków wychodzących z wierzchołka odpowiadają czasowi realizacji zadania.

Harmonogram Gantt'a





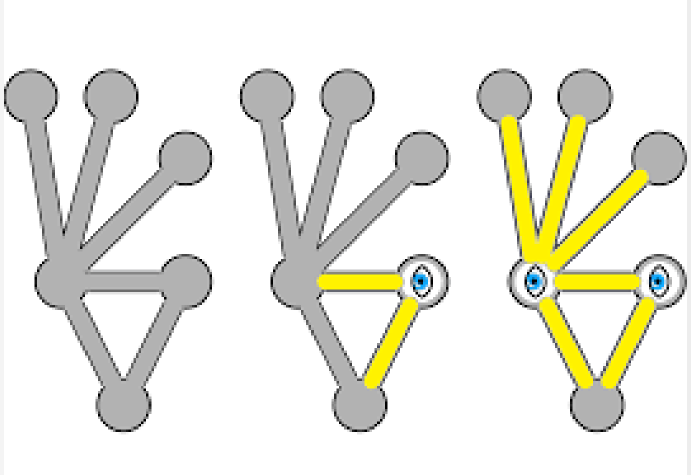


Pokrycie wierzchołkowe

- **Pokrycie wierzchołkowe grafu** G to taki podzbiór jego wierzchołków, że każda krawędź G jest incydentna do jakiegoś wierzchołka z tego podzbioru.
- Pokryciem wierzchołkowym grafu $G = (V,E)$ nazywamy taki zbiór V', że: $V' \subseteq V \wedge (\forall e \in E, \exists v \in V' : v \in e)$
- Problem optymalizacyjny – poszukiwanie dla danego grafu pokrycia wierchołkowego o najmniejszy rozmiarze (liczbie wierzchołków)
- Problem decyzyjny – czy istnieje w danym grafie pokrycie wierchołkowego o zadanym rozmiarze
- Przykład: CBA ma listę zamieszanych w aferę *n* polityków. Zna powiązania wzajemne polityków (kto się z kim komunikuje) i chce podsłuchiwać wszystkie ich rozmowy telefoniczne. Jaką minimalną liczbę podsłuchów musi zainstalować?

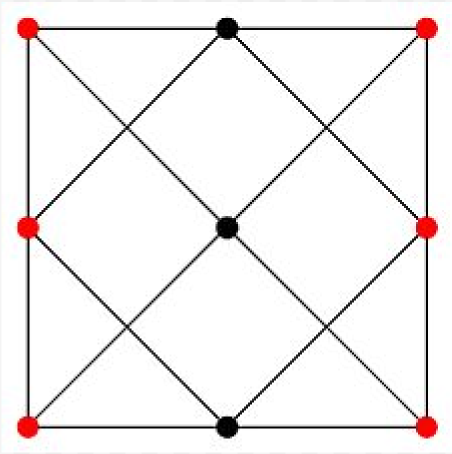
Pokrycie wierzchołkowe - cd

- Przyporządkowujemy wierzchołek do każdego polityka.
- Dwa wierzchołki łączymy krawędzią, jeśli politycy się znają.
- Krawędź *(u,v)* jest incydentna z wierzchołkami *u* oraz *v*.
- Wierzchołek należy do pokrycia krawędzi, jeśli jest z nią incydentny.
- Zbiór wierzchołków incydentnych z wszystkimi krawędziami nazywany jest pokryciem wierzchołkowym.
- Rozwiązywany problem polega na znalezieniu optymalnego pokrycia wierzchołkowego tzn. o minimalnej wielkości.

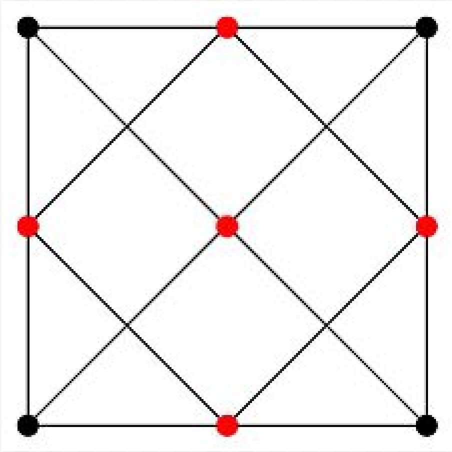


Pokrycie wierzchołkowe - cd

- Pokrycie wierzchołkowe zabioru:



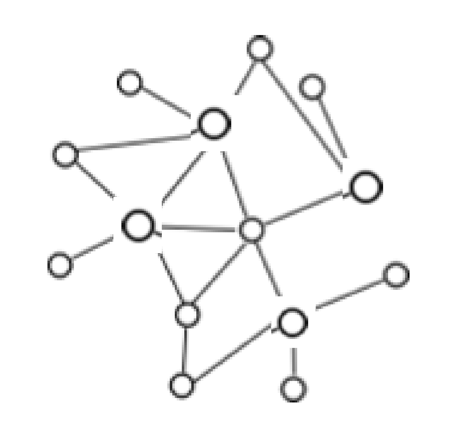
- Najmniejsze pokrycie wierzchołkowe w grafie:



Zbiór dominujący

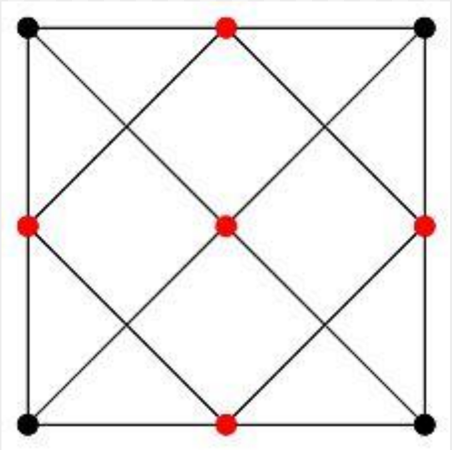
- **Zbiorem dominującym** grafu $G = (V,E)$ nazywamy taki podzbiór V' zbioru wierzchołków V, że każdy wierzchołek, który nie należy do V' ma w tym zbiorze co najmniej jednego sąsiada (jest połączony krawędzią z przynajmniej jednym wierzchołkiem z V').
- Zwyczajowo przez $\gamma(G)$ oznaczamy liczbę wierzchołków w najmniejszym zbiorze dominującym grafu G.

- Przykład: Należy ostrzec grupę zamieszanych w aferę n osób przed planowana akcją CBA. Znane są powiązania między osobami (kto się z kim komunikuje). Które osoby (możliwie najmniej!) trzeba poinformować, aby za ich pośrednictwem dotrzeć do wszystkich zainteresowanych?



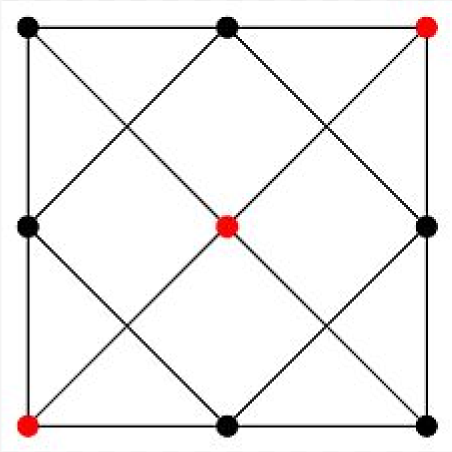
Zbiór dominujący - cd

- Zbiór dominujący (nie totalnie) w grafie:



- Zbiór totalnie dominujący - taki zbiór dominujący V', w którym każdy wierzchołek z V' ma co najmniej jednego sąsiada V'

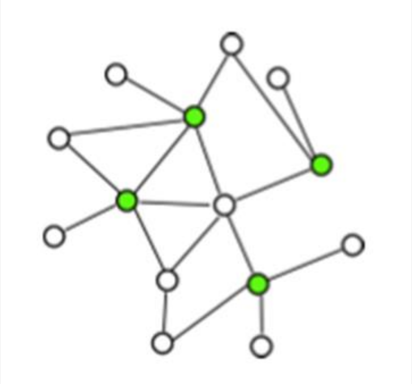
- Najmniejszy zbiór totalnie dominujący, liczba totalnego dominowania – 3:



Zbiór dominujący - cd

- Przyporządkowujemy wierzchołek do każdej z *n* osób.
- Dwa wierzchołki łączymy krawędzią, jeśli osoby się znają.
- Wierzchołek *u* jest przyległy (sąsiedni) do wierzchołka *v* jeśli są połaczone krawędzią *(u,v)*.
- Wierzchołek jest dominujący do siebie i wierzchołków przyległych.
- Zbiór wierzchołków dominujących - S jest podzbiorem zbioru wierzchołków takim, że każdy wierzchołek grafu jest dominowany przez wierzchołek z S.
- Rozwiązywany problem polega na znalezieniu optymalnego zbioru dominującego tzn. o minimalnej wielkości.

- Przykład rozwiązania:

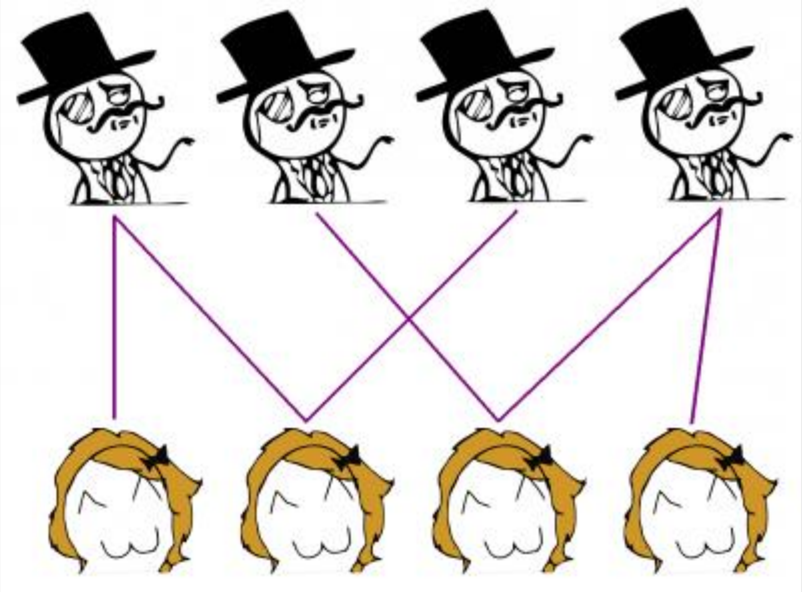


Skojarzenie

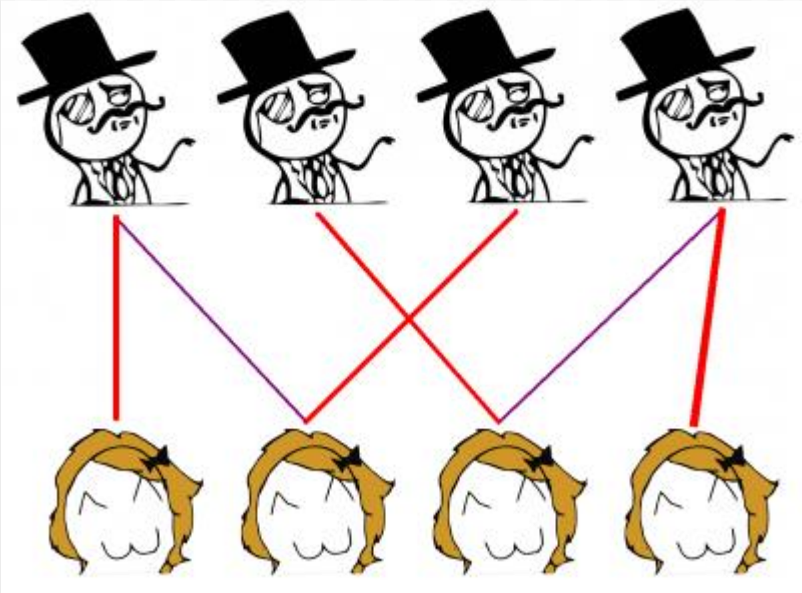
- **Skojarzeniem grafu** nazywa się nie zawierający pętli podzbiór *M* krawędzi grafu *E* taki, że żadne dwie krawędzie w *M* nie są sąsiednie, tj. nie spotykają się w jednym wierzchołku.
- Wierzchołki będące końcami krawędzi należących do *M* są ***M-nasycone***. Wierzchołki nie będące końcami krawędzi należących do *M* są ***M-nienasycone***.
- **Skojarzenie doskonałe** to podzbiór *M* krawędzi grafu *G*, taki, że każdy wierzchołek *G* jest *M-nasycony*.
- Skojarzenie doskonałe jest zawsze skojarzeniem **największym**, tj. takim, że nie istnieje skojarzenie grafu *G* o większej liczbie krawędzi.
- Pary wierzchołków połączone bezpośrednio krawędzią należącą do *M* są skojarzone przez *M*.
- **M-przemienna ścieżka** to ścieżka ułożona naprzemiennie z krawędzi należących i nie należących do *M*.

Skojarzenie - 1

- Krawędź pokazuje, którego z kawalerów zna panna.

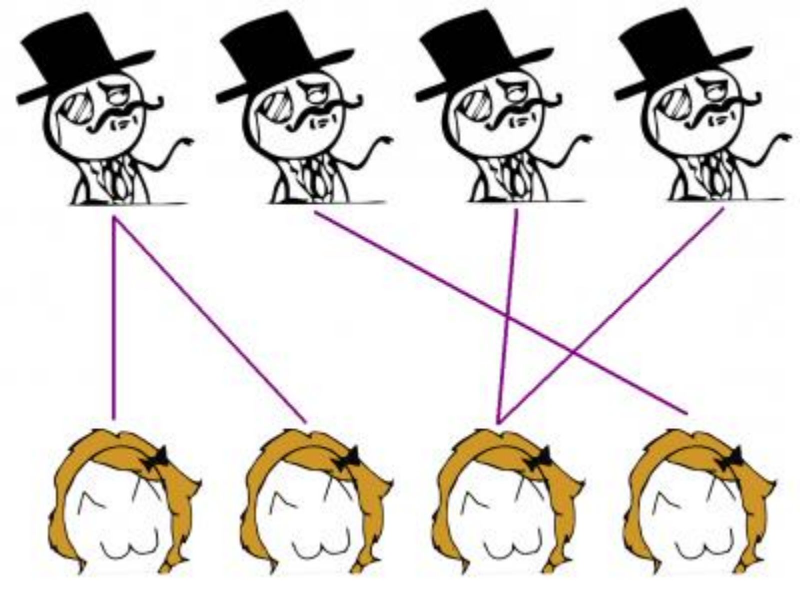


- W tej grupie istnieje skojarzenie doskonałe

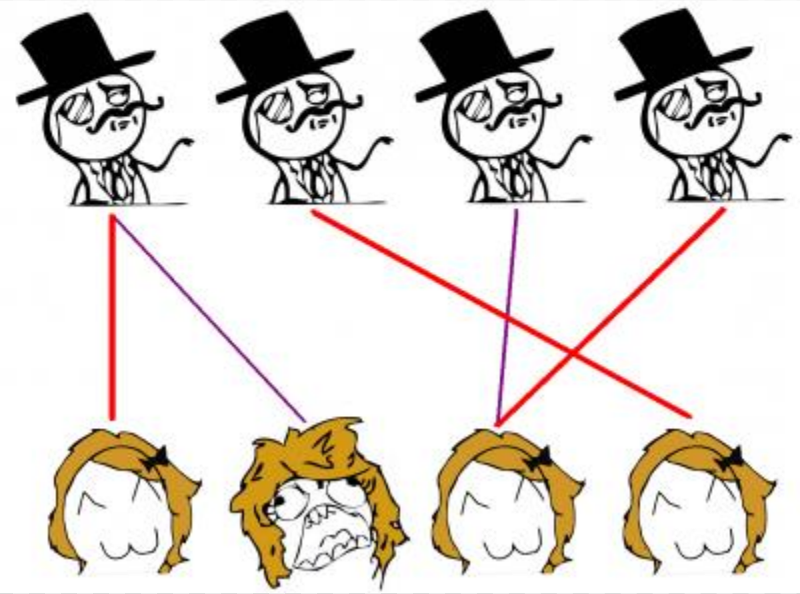


Skojarzenie - 2

- Krawędź pokazuje, którego z kawalerów zna panna.



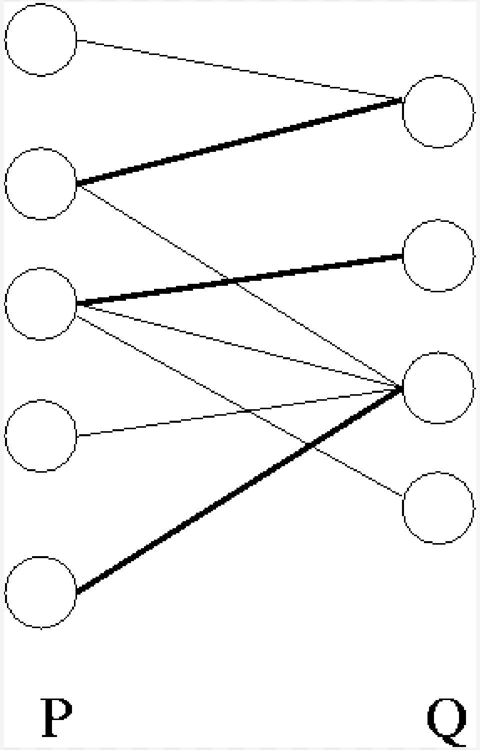
- W tej grupie nie istnieje skojarzenie doskonałe



Skojarzenie - cd

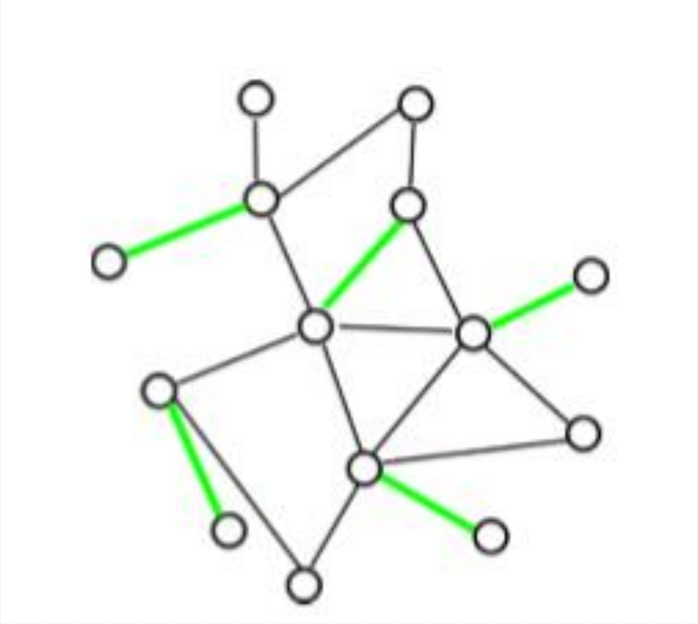
Przykład: zagadnienie kojarzenia małżeństw.

- **Twierdzenie o kojarzeniu małżeństw** (Philip Hall)- dotyczące istnienia pełnego skojarzenia grafu dwudzielnego
- Mamy dwie grupy - dziewcząt **P** i chłopców **Q**, oraz pewną sieć znajomości, to znaczy wiemy, których chłopców z tej grupy zna każda z dziewcząt. Kiedy zachodzi sytuacja, w której każdej dziewczynie można przyporządkować jednego kandydata na męża? (kandydaci nie mogą się powtarzać)
- Warunkiem koniecznym i wystarczającym na to, by istniało takie skojarzenie par, jest to, by każda podgrupa dziewcząt, licząca k-osób, znała co najmniej k-chłopców.



Skojarzenie - cd

Przykład zastosowania:



- Zagadnienia maksymalnego i największego skojarzenia znajdują zastosowanie w wielu innych praktycznych problemach informatycznych, gdzie jednym z przykładów może być optymalny przydział zadań.

- Skojarzenie M nazywamy **maksymalnym** skojarzeniem, jeżeli nie zawiera się w żadnym innym skojarzeniu.
- Skojarzenie M o największej możliwej mocy nazywamy **największym** skojarzeniem i oznaczamy jako M_*.

Zagadnienie podziału grafu

Dany jest
- nieskierowany, ważony graf *G=(N,E)*
- zbiór liczb $\{b_1, b_2, \ldots, b_m\}$, gdzie $b_i > 0$ dla *i=1,...,m*

Należy znaleźć podział zbioru *N*

$$X = (X_1, X_2, \ldots, X_i, \ldots, X_m) \quad , \quad X_i \subset N \qquad (i=1,\ldots,m),$$

który maksymalizuje funkcję celu:

$$f(X) = \sum_{i=1}^{m} \sum_{j,k \in X_i} c_{jk} \qquad \text{(I)}$$

przy ograniczeniach:

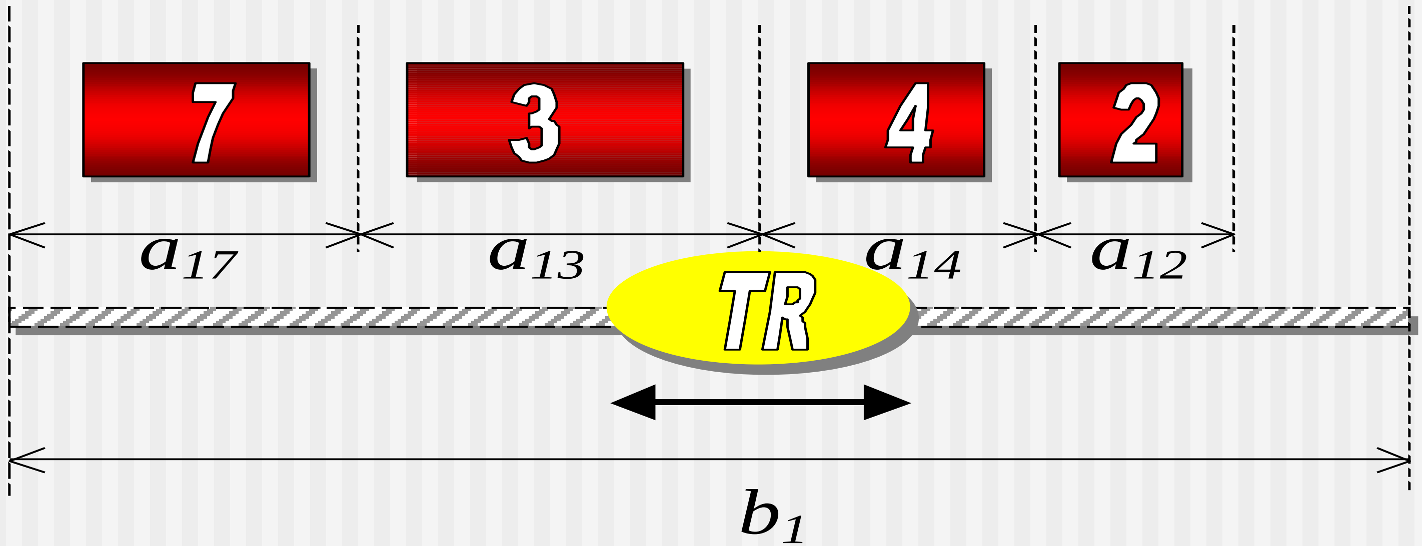
$$\bigcup_{i=1}^{m} X_i = N \quad , \quad X_i \cap X_k = \emptyset \text{ dla } i \neq k \qquad \text{(II)}$$

$$\sum_{j \in X_i} a_j \leq b_i \qquad i=1,\ldots,m \qquad \text{(III)}$$

gdzie:
- $a_j = (a_{1j}, a_{2j}, \ldots, a_{ij}, \ldots, a_{mj})$ - waga wierzchołka $j \in N$
- c_{jk} - waga krawędzi $jk \in E$
- b_i - rozmiar punktu skupienia *i*
- m - liczba punktów skupienia

Zagadnienie podziału grafu -cd

Przykład: grupowanie wyrobów i maszyn w elastycznych systemach wytwarzania



Liniowe rozmieszczenie maszyn w komórce *i=1*

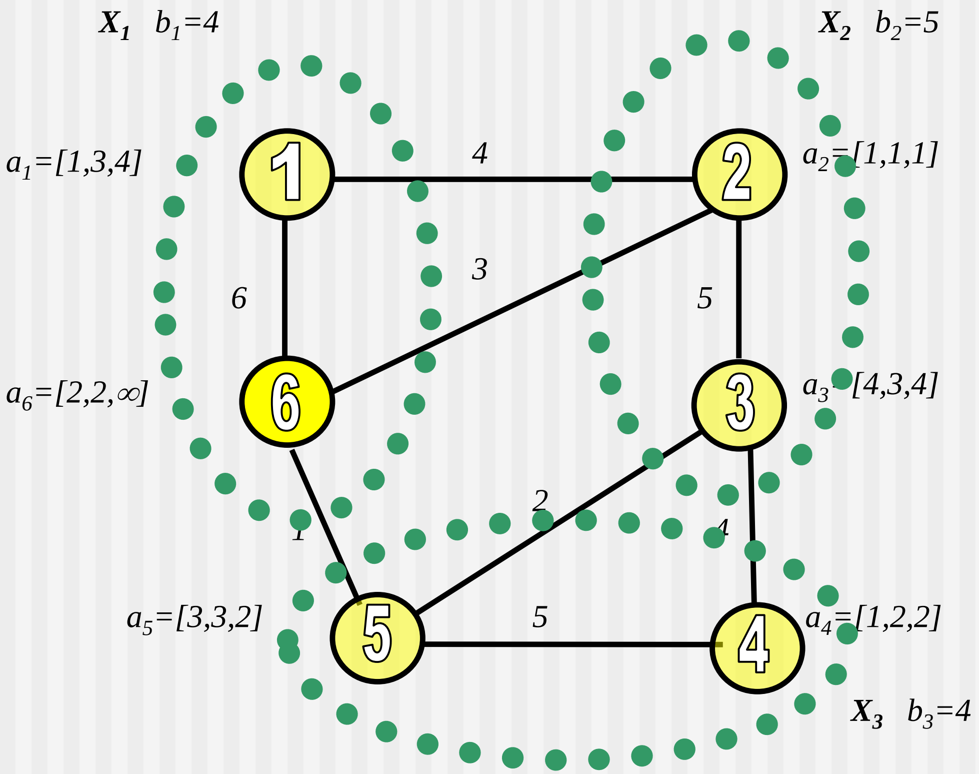
b_i – zasób - długość liniowego robota transportowego *TR*
a_{ij} – zapotrzebowanie na zasób w komórce *i* maszyny *j*
c_{jk} – wielkość transportu pomiędzy maszynami *j* oraz *k*

Celem jest taki podział maszyn między komórkami, który:

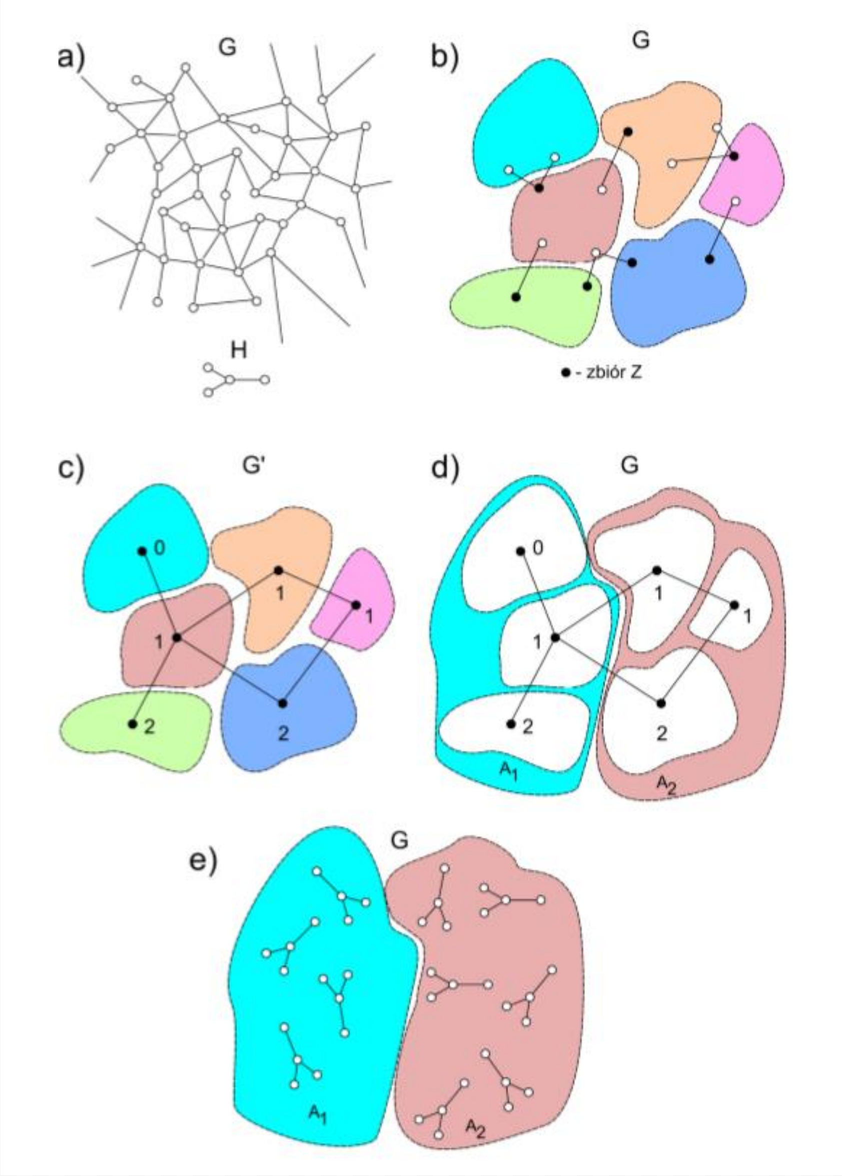
- ☑ maksymalizuje tańszy transport wewnętrzny
- ☒ lub minimalizuje transport między komórkami

Zagadnienie podziału grafu -cd

Realizacja podziału grafu n=6/m=3



Algorytm klastrowania grafu



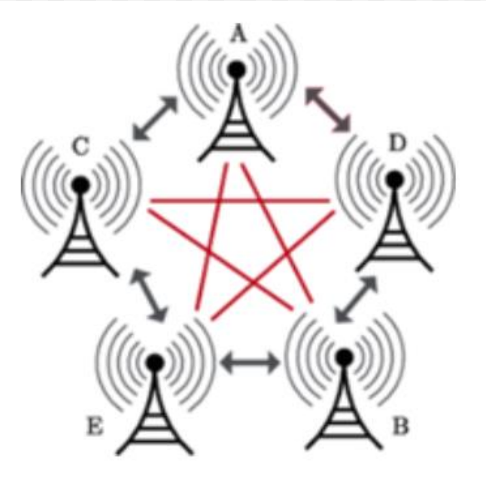
Zagadnienie kolorowania grafu

- Klasyczne - wierzchołkowe zagadnienie **kolorowanie grafu** jest przypisaniem wszystkim wierzchołkom grafu – jednego z kolorów tak, aby żadne z sąsiednich wierzchołków nie miały tego samego koloru.
- Pokolorowanie wierzchołkowe jest poprawne (legalne, dozwolone) wtedy, gdy końcom żadnej krawędzi nie przypisano tego samego koloru.
- **Optymalnym pokolorowaniem** danego grafu nazywamy legalne pokolorowanie zawierające najmniejszą możliwą liczbę kolorów.
- **Liczbą chromatyczną grafu** *G* nazywamy liczbę równą najmniejszej możliwej liczbie kolorów potrzebnych do legalnego pokolorowania wierzchołków grafu *G*.
- Kolorowanie krawędziowe
- Twierdzenie o 4 barwach dla grafów planarnych (hipoteza - 1852 r. , dowód z komputerowym sprawdzeniem 1936-ciu przypadków szczególnych - 1976 r.)

Zagadnienie kolorowania grafu

Twierdzenie o czterech barwach – dla każdego skończonego grafu planarnego (V, E) istnieje funkcja $k\colon V \to \{k_1, k_2, k_3, k_4\}$, taka że $\forall_{\{v_1, v_2\} \in E} \, (k(v_1) \neq k(v_2))$, czyli możliwe jest przypisanie każdemu z jego wierzchołków jednej z czterech liczb 1, 2, 3 i 4 w taki sposób, aby żadne sąsiednie wierzchołki nie miały przyporządkowanej tej samej liczby. Jest to jeden z najsłynniejszych problemów matematycznych.

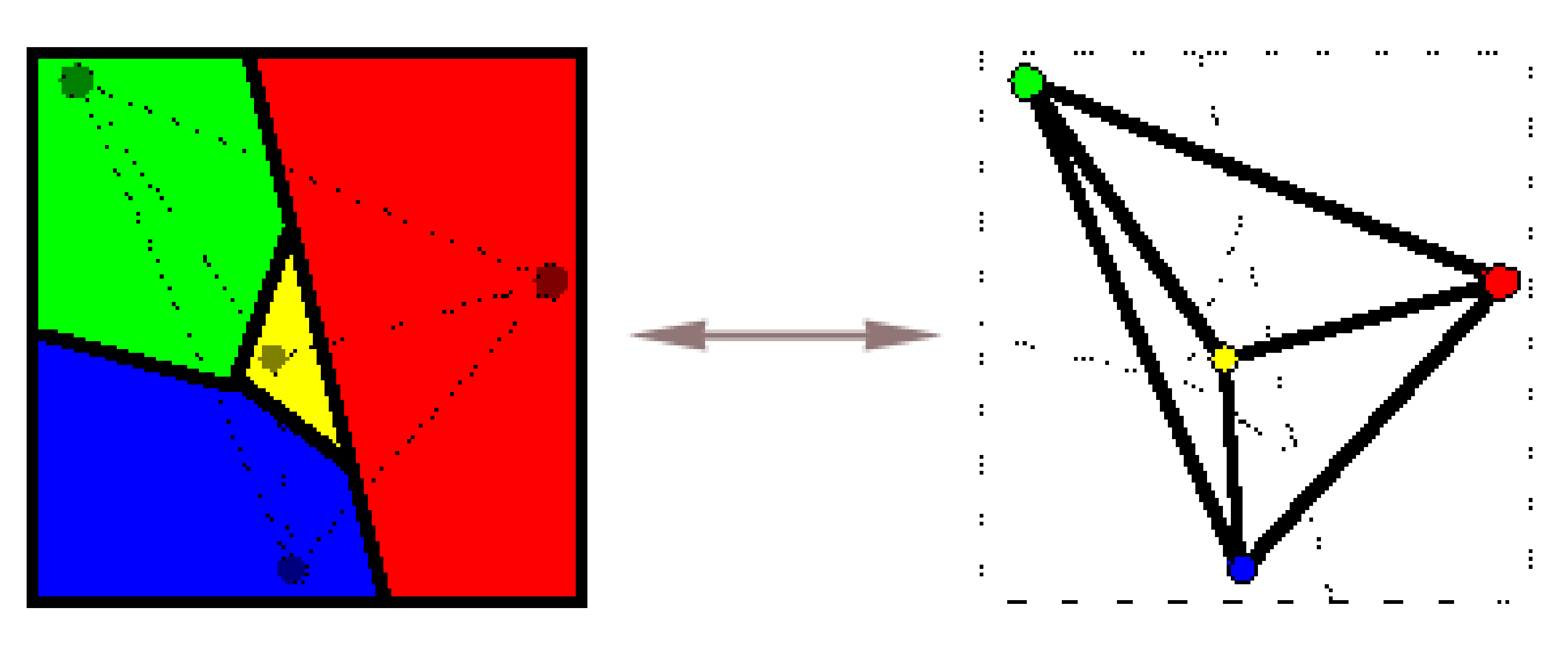
przydziału częstotliwości
za pomocą kolorowania
grafów



Zagadnienie kolorowania grafu - cd

- Przykład: należy pokolorować mapę minimalną liczbą kolorów, tak aby każde dwa sąsiednie państwa miały różny kolor.

- Równoważność zagadnienia dla mapy i dla grafu



Zagadnienie kolorowania grafu - cd

- Problem NP-trudny – algorytmy przybliżone

Algorytm LF (largest first):

- Uporządkuj wierzchołki grafu malejąco według ich stopni (liczby krawędzi z nich wychodzących).
- Koloruj wierzchołki zachłannie, zgodnie z ustaloną wcześniej kolejnością (zaczynając od wierzchołka o największym stopniu).

Algorytm LF jest algorytmem statycznym, gdyż raz ustalona kolejność wierzchołków nie zmienia się w trakcie jego działania. Najmniejszym dość trudnym grafem jest ścieżka P_6

Zagadnienie kolorowania grafu - cd

Algorytm SL (smallest last):

- Znajdź wierzchołek o minimalnym stopniu i usuń go z grafu.
- Powtarzaj krok pierwszy tak długo, aż graf będzie pusty (zapamiętaj kolejność usuwanych wierzchołków).
- Koloruj wierzchołki zachłannie, zgodnie z ustaloną wcześniej kolejnością (zaczynając od wierzchołków usuniętych później)

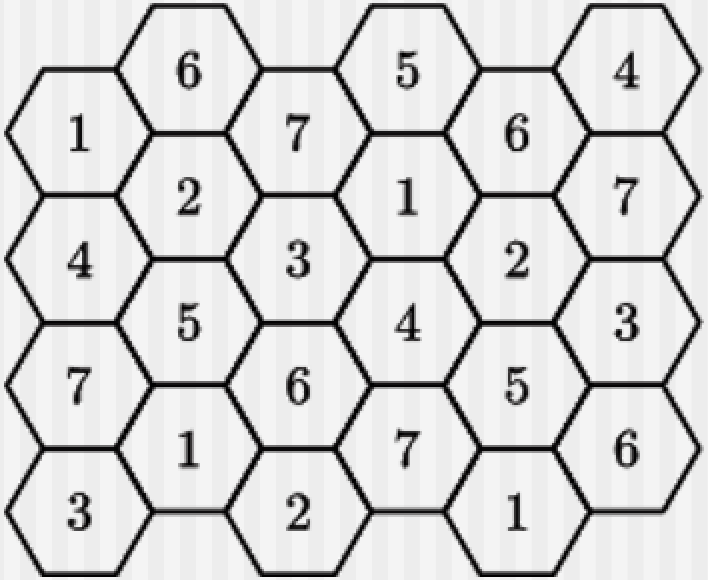
Alg. SL jest statyczny, jego złożoność wynosi O(n+m) ,

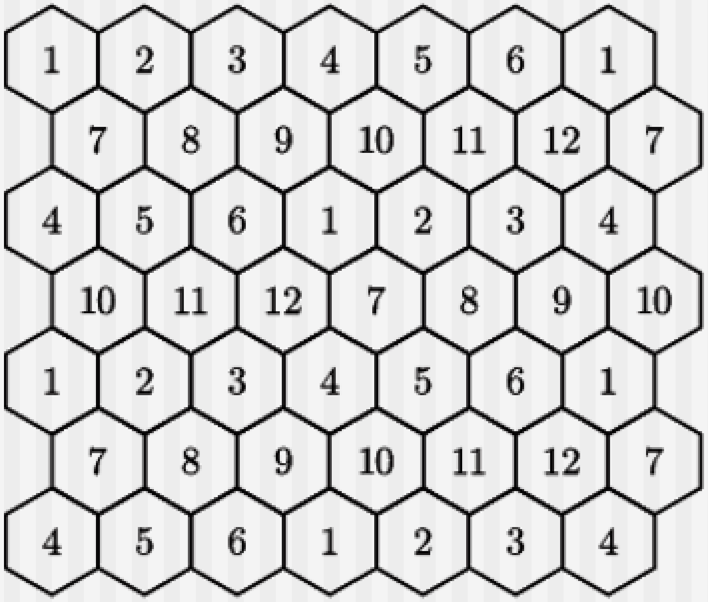
Zagadnienie kolorowania grafu - cd

przydziału częstotliwości za pomocą kolorowania grafów

=

Ile kolorów potrzeba, aby pokolorować (nieskończony) graf którego wierzchołkami są wszystkie punkty płaszczyzny euklidesowej a dwa punkty/wierzchołki są połączone krawędzią, jeśli znajdują się w odległości dokładnie 1?

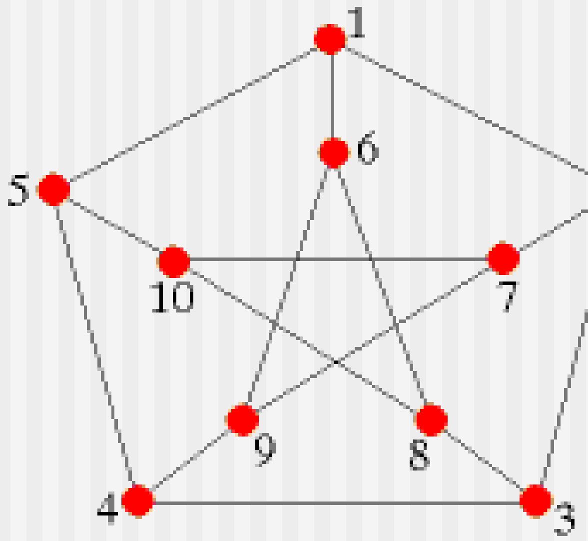


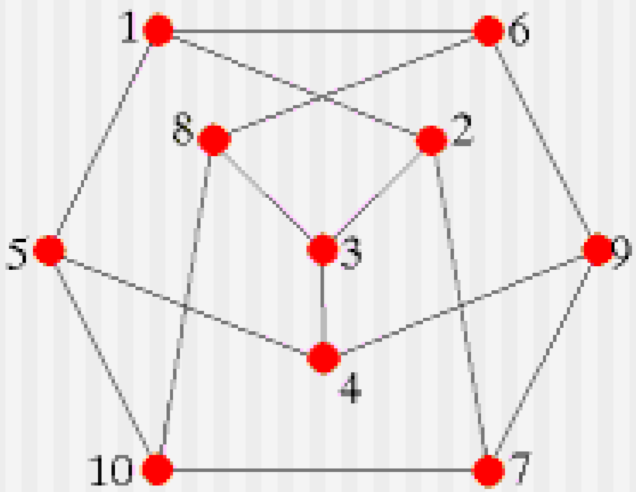


http://www.deltami.edu.pl/temat/matematyka/zastosowania/2014/10/30/O_modelowaniu_przydzialu_czestot/

Izomorfizm grafu

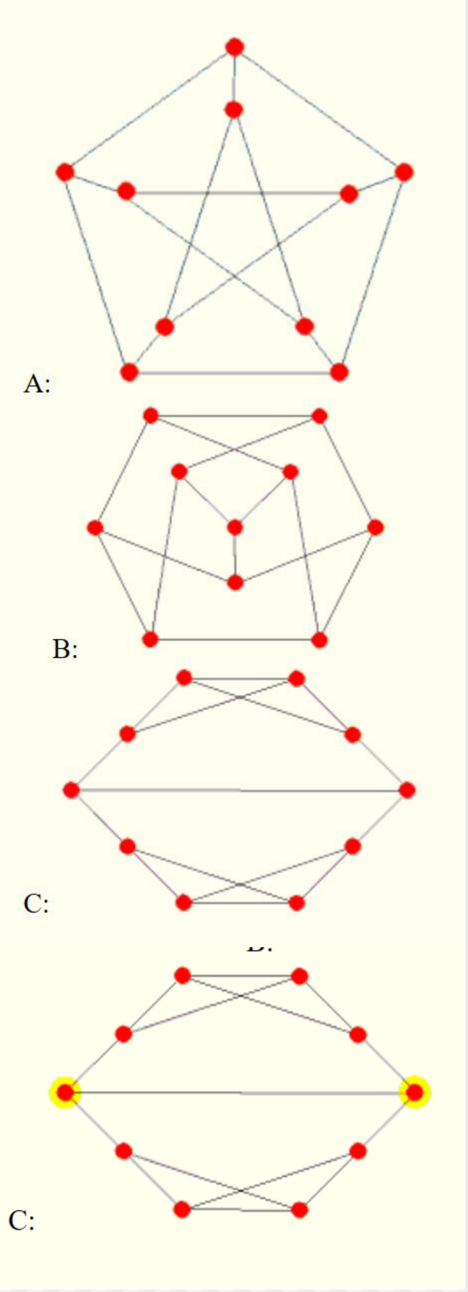
- Grafy G i F nazywamy izomorficznymi, jeżeli istnieje bijekcja zbioru wierzchołków grafu G na zbiór wierzchołków grafu F, która zachowuje strukturę grafu (krawędzie).
- Rozstrzygnięcie, czy grafy G i F są tym samym grafem? (permutacja wierzchołków)





Izomorfizm grafu

- Izomorfizm grafów zachowuje właściwie wszystkie interesujące własności)
- Graf B jest izomorficzny z grafem A. Na rysunku podano bijekcję między wierzchołkami obu grafów.
- Graf C nie jest izomorficzny z grafem A, gdyż usunięcie dwóch zaznaczonych wierzchołków spowoduje, że graf przestanie być spójny. W grafie A takie wierzchołki nie istnieją.



Zagadnienie przydziału - AP

Dany jest
- zbiór zadań $N=\{1,...,n\}$
- zbiór maszyn $M=\{1,...,n\}$
- macierz kosztów przydziału $(n \times n)$ - $A=[a_{i,j}]$

Należy znaleźć rozwiązanie X, które minimalizuje funkcję celu:

$$\min \quad f(\mathrm{X}) = \sum_{i=1}^{m} \sum_{j=1}^{m} a_{ij} x_{ij}$$

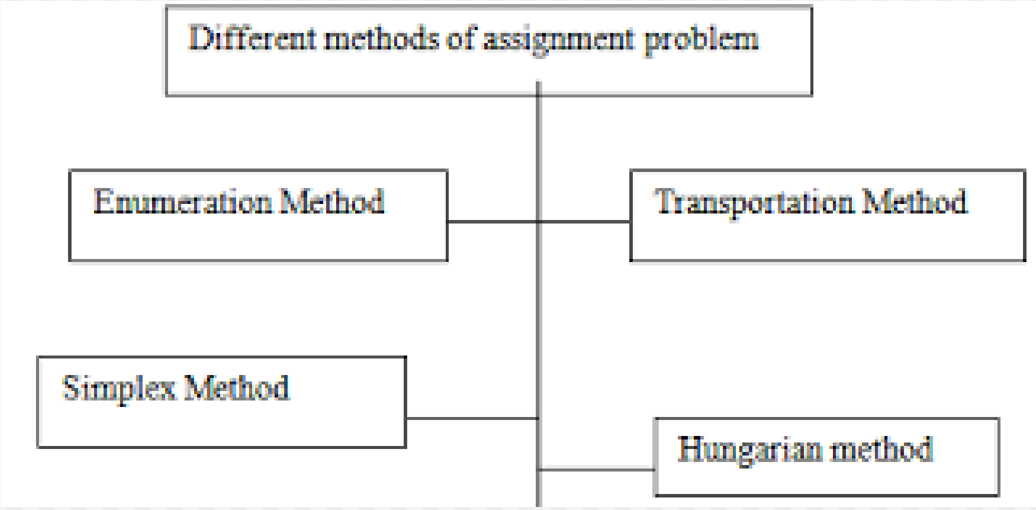
$$\text{ogr.} \quad \sum_{i=1}^{n} x_{ij} = 1 \quad \text{dla } j = 1, \ldots, n$$

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \text{dla } i = 1, \ldots, n$$

$$\text{gdzie} \quad x_{ij} = \{0, 1\}$$

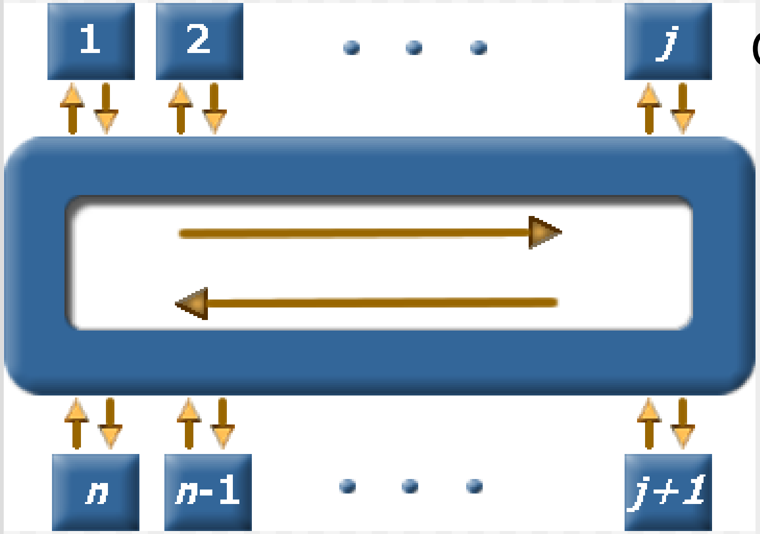
Algorytmy dla AP

- Sprawdzenie wszystkich przydziałów - *n!*
- Efektywne algorytmy wielomianowe
- Problem przypisania jest szczególnym przypadkiem problemu transportowego, który jest szczególnym przypadkiem problemu minimalnego przepływu, który z kolei jest szczególnym przypadkiem programowania liniowego.
- Algorytm simpleks
- Metoda węgierska



Przykład zagadnienia

Elastyczny system produkcyjny składający się z *n* stanowisk roboczych z maszynami rozmieszczonych wokół przenośnika o ruchu okrężnym



Odległość pomiędzy stanowiskami $[d_{i,j}]_{n\times n}$:

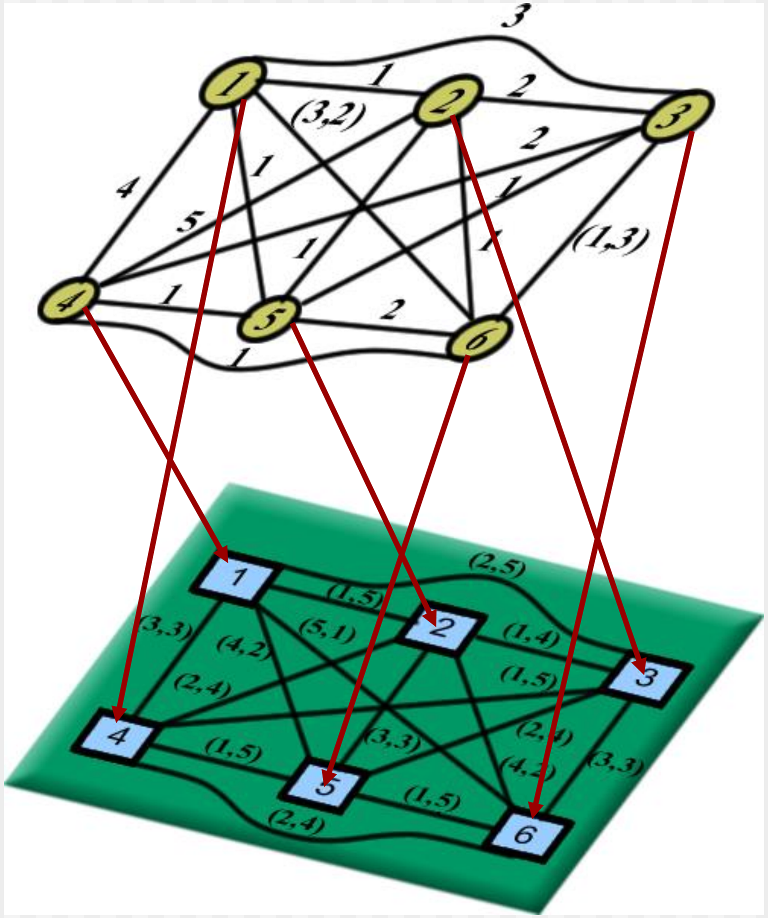
$$d_{ij} = \begin{cases} j-i & \text{dla } i \le j \\ n-i+j & \text{dla } i > j \end{cases}$$

$$\text{dla } i, j = 1, \ldots, n.$$

Liczba detali, na jednostkę czasu, które po obróbce na maszynie *k* są poddane dalszemu procesowi technologicznemu – na maszynie *l* : $[f_{kl}]_{n\times n}$.

Cel - zminimalizowanie sumarycznego czasu transportu detali, poprzez odpowiednie przydzielenie maszyn do stanowisk roboczych.

Przykład zagadnienia



$$D = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 \\ 5 & 0 & 1 & 2 & 3 & 4 \\ 4 & 5 & 0 & 1 & 2 & 3 \\ 3 & 4 & 5 & 0 & 1 & 2 \\ 2 & 3 & 4 & 5 & 0 & 1 \\ 1 & 2 & 3 & 4 & 5 & 0 \end{bmatrix} \qquad F = \begin{bmatrix} 0 & 1 & 3 & 4 & 1 & 3 \\ 1 & 0 & 1 & 5 & 1 & 1 \\ 3 & 1 & 0 & 2 & 1 & 1 \\ 4 & 5 & 2 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 2 \\ 3 & 2 & 3 & 1 & 2 & 0 \end{bmatrix}$$

Liczba możliwych przydziałów: 6!=720

4 → 1

1 → 4

2 → 3

5 → 2

6 → 5

3 → 6

$\pi = (4, 5, 2, 1, 6, 3)$

$$\sum_{i=1}^{6}\sum_{j=1}^{6} d_{ij} f_{\pi(i)\pi(j)} = 180$$

Model matematyczny zagadnienia

Dane są trzy $n \times n$ wymiarowe rzeczywiste macierze $F=(f_{ij})$, $D=(d_{kl})$ oraz $B=(b_{ik})$

f_{ij} przepływ (liczba połączeń) pomiędzy zadaniami i oraz j,

d_{kl} odległość pomiędzy stanowiskiem k oraz l

~~b_{ik} koszt związany z przydziałem zadania i do maszyny k~~

n liczba maszyn oraz zadań, które mają zostać do siebie przyporządkowane

Do każdej maszyny przydzielamy jedno i tylko jedno zadanie oraz każde zadanie jest przydzielone do jednej maszyny

Chcemy zminimalizować całkowity koszt funkcjonowania systemu, związanego np. z kosztami transportu technologicznego (*Quadratic Assignmnent Problem* postać Koopmansa-Beckmana).

$$\min_{\pi \in S_n} \left\{ \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} d_{\pi(i)\pi(j)} + \cancel{\sum_{i=1}^{n} b_{i\pi(i)}} \right\}$$

$\pi = \{\pi(1), \ldots, \pi(i), \ldots, \pi(j), \ldots, \pi(n)\}$ permutacja zbioru $N = \{1, 2, \ldots, n\}$

S_n - jest zbiorem wszystkich n-elementowych permutacji

Literatura:

[1] Wojciech Wawrzyniak - Rozproszone algorytmy aproksymacyjne w analizie własności grafowych.
[2] Wikipedia
[3] Tw. Halla - https://eszkola.pl/matematyka/problem-kojarzenia-malzenstw-5604.html

Pytania?
Dziękuję za uwagę!