Suport/Transcript curs algoritmica grafurilor
XI. Cuplaje în grafuri.

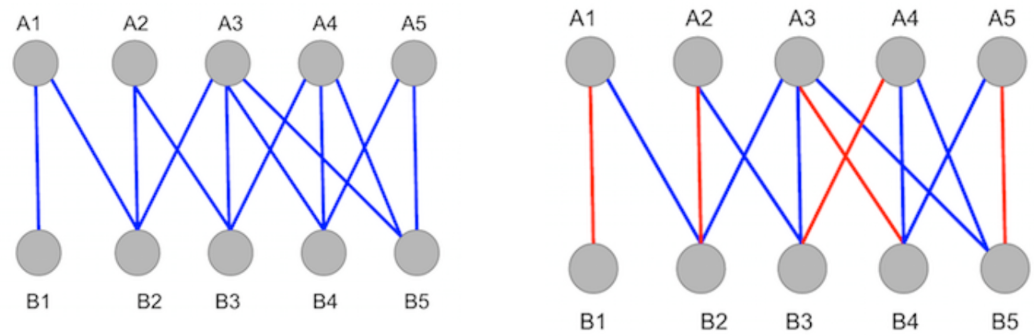
11.1 Cuplaje în grafuri

Recapitulare curs 10 Un cuplaj în G este o mulțime de muchii M în care nici o pereche de muchii nu are un vârf comun. vârfurile adiacente la muchiile din M se numesc vârfuri *saturate de M* (sau *M-saturate*). Celelalte vârfuri se numesc *M-nesaturate*.

Tipuri de cuplaje:

- un cuplaj perfect al lui G este un cuplaj care saturează toate vârfurile lui G,
- un cuplaj maxim al lui G este un cuplaj care are cel mai mare număr posibil de muchii,
- un cuplaj maximal al lui G este un cuplaj care nu poate fi lărgit prin adăugarea unei muchii.

Figura 1a prezintă un graf bipartit și figura 1b un cuplaj aleator în graful bipartit (muchiile care fac parte din cuplaj sunt colorate cu roșu).



(a) (b)

Figura 1: Un graf bipartit şi un cuplaj aleator în graful bipartit.

Un *lanț M-alternant* (*cale M-alternantă*) este un lanț în G în care toate muchiile alternează între muchii din M și muchii ce nu aparțin cuplajului M (figura 2).

Un *M-lanț de creștere* (*M-cale de creștere*) este un lanț M-alternant care are ambele capete M-nesaturate.

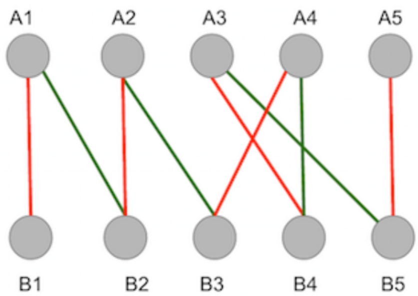
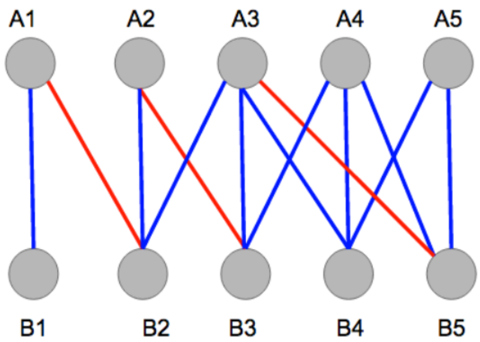


Figura 2: Lanţ M-alternant pentru graful bipartit din figura 1a.

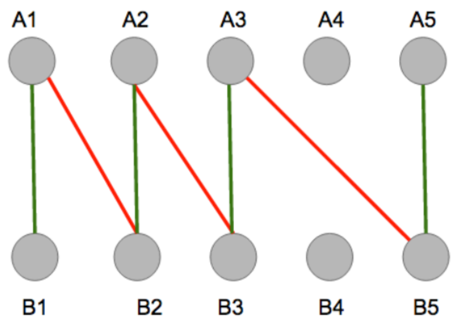
Teorema 9.3 (Berge)
Un cuplaj M al unui graf $G = (V, E)$ este maxim **dacă și numai dacă** G nu conține M-lanțuri de creștere.

Demonstraţie.
(vezi cursul 10) □

De exemplu, cuplajul $M = \{(A1, B2), (A2, B3), (A3, B5)\}$ din figura 3a nu este maxim deoarece conține un M-lanț de creștere $(B1, A1, B2, A2, B3, A3, B5, A5)$ (figura 3b).



(a)



(b)

Figura 3: Un graf bipartit şi un cuplaj aleator în graful bipartit.

Teorema 9.4 (Hall)
Fie G un graf bipartit cu mulțimile partite X și Y. X poate fi cuplat **perfect** în Y dacă și numai dacă $|N(S)| \geq |S|$ pentru toate submulțimile S ale lui X.

11.2 Grafuri bipartite ponderate

Preferințele pot fi exprimate cu ajutorul ponderilor. Fie un graf bipartit ponderat unde:

- muchia $(x, y) \in E$ are asociată ponderea $w(x, y)$
- ponderea cuplajului M este suma ponderilor muchiilor din cuplajul M

$$w(M) = \sum_{(x,y)\in M} w(x, y)$$

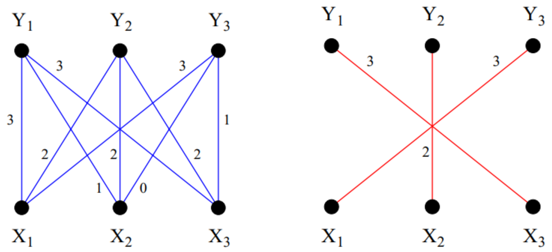


Figura 4: Un graf bipartit complet şi cuplajul de pondere maximă.

Problema: pentru graful bipartit G găsiți un cuplaj de pondere maximă.

Pentru a forma un graf bipartit complet (figura 4 stânga) se adaugă și muchia (X_2, Y_3) care are ponderea 0, $w(X_2, Y_3) = 0$.

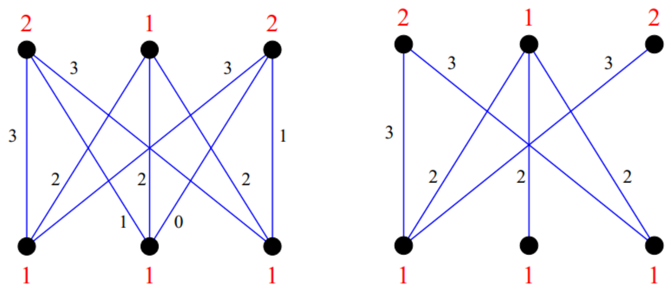
Pentru a găsi cuplajul de pondere maximă se aplică o etichetare a vârfurilor:

- o etichetare a vârfurilor este o funcție $l : V \to \mathbb{R}$,
- o etichetare fezabilă respectă:

$$l(x) + l(y) \geq w(x, y), \forall x \in X, y \in Y,$$

- un graf egal (ținând cont de l) este un graf $G = (V, E_l)$ unde:

$$E_l = \{(x, y)|l(x) + l(y) = w(x, y)\}.$$



(a) etichetare fezabila l (b) Graf egal G_l

Figura 5: O etichetare fezabilă şi graful egal.

Teorema 11.1 (Teorema Kuhn-Munkres)**.**
Dacă l este fezabilă şi M este un cuplaj perfect în E_l atunci M este un cuplaj de pondere maximă.

Demonstraţie.
Fie $e \in E$ şi $e = (e_x, e_y)$, M' un cuplaj perfect în $G = (V, E)$(nu neapărat în E_l).

Deoarece fiecare vârf $v \in V$ e acoperit exact o singură dată de M avem

$$w(M') = \sum_{e \in M'} w(e) \leq \sum_{e \in M'} (l(e_x) + l(e_y)) = \sum_{v \in V} l(v).$$

$\sum_{v \in V} l(v)$ este o limită superioară a oricărui cuplaj.

Fie M un cuplaj perfect în E_l

$$\to w(M) = \sum_{e \in M} w(e) = \sum_{v \in V} l(v)$$

şi atunci
$$w(M') \le w(M)$$
deci M este optimal. □

Teorema KM transformă problema găsirii unui cuplaj de pondere maximă (problemă de optimizare) într-o problemă combinatorială ce presupune găsirea unui cuplaj perfect. Pentru un cuplaj M și o etichetare fezabilă l avem:
$$w(m) \le \sum_{v \in V} l(v),$$
seamănă cu teorema fluxului maxim și a tăieturii minime.

Într-un cuplaj perfect fiecare vârf este acoperit (saturat) o singură dată și
$$w(M) = \sum_{e \in M} w(e) = \sum_{v \in V} l(v);$$
oricare alt cuplaj M' din $G = (V, E)$satisface
$$w(M') = \sum_{e \in M'} w(e) \le \sum_{v \in V} l(v);$$
astfel $w(M') \le w(M)$ și M trebuie să fie optimal.

11.2.1 Metoda maghiară

Un posibil algoritm pentru determinarea cuplajului maxim într-un graf este:

cuplaj(G)
1: start cu o etichetare fezabilă l și un cuplaj M în E_l
2: **while** cuplajul M nu e perfect **do**
3: caută un M-lanț de creștere pentru M în E_l (crește dimensiunea lui M)
4: **if** nu există un M-lanț de creștere **then**
5: îmbunătățește l la l' astfel încât $E_l \subset E_{l'}$

În fiecare pas se crește dimensiunea lui M sau E_l. Conform teoremei Kuhn-Munkres, M va fi un cuplaj de pondere maximă.

Pentru a găsi inițial o etichetare fezabilă se poate folosi:
$$\forall y \in Y, l(y) = 0, \;\; \forall x \in X, l(x) = \max_{y \in Y}\{w(x, y)\}.$$
Astfel este evident
$$\forall x \in X, y \in Y, w(x, y) \le l(x) + l(y).$$
Figura 6 prezintă o astfel de etichetare.

Pentru l o etichetare fezabilă, se definește un vecin al lui $u \in V$ și un set $S \subseteq V$ astfel:
$$N_l(u) = \{v | (u, v) \in E_l\}, \;\; N_l(S) = \cup_{u \in S} N_l(u).$$
Îmbunătățirea etichetării se poate face conform lemei de mai jos.

Lema 11.2 (Îmbunătăţirea unei etichetări)**.** *Fie* $S \subseteq X$ *şi* $T = N_l(S) \ne Y$*. Fie*
$$\alpha_l = \min_{x \in S, y \notin T}\{l(x) + l(y) - w(x, y)\} \tag{1}$$

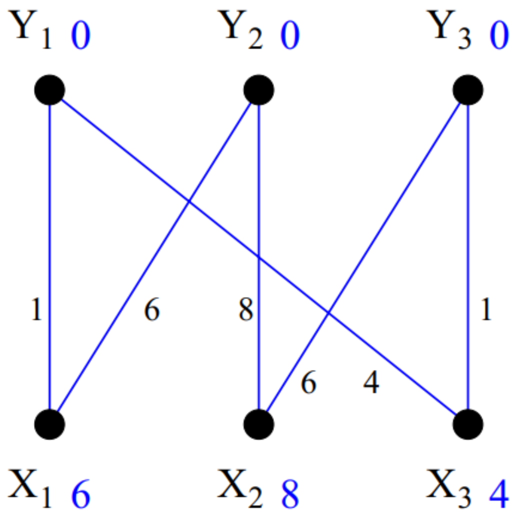


Figura 6: Un graf bipartit complet şi cuplajul de pondere maximă.

$$l'(v) = \begin{cases} l(v) - \alpha_l & \textit{dacă } v \in S, \\ l(v) + \alpha_l & \textit{dacă } v \in T, \\ l(v) & \textit{altfel.} \end{cases} \tag{2}$$

Atunci l' *este o etichetare fezabilă şi*

1. *dacă* $(x, y) \in E_l$ *pentru* $x \in S, y \in T$ *atunci* $(x, y) \in E_{l'}$*;*
2. *dacă* $(x, y) \in E_l$ *pentru* $x \notin S, y \notin T$ *atunci* $(x, y) \in E_{l'}$*;*
3. *există o muchie* $(x, y) \in E_{l'}$ *pentru* $x \in S, y \notin T$*.*

Metoda maghiară - exemplu matriceal Să se rezolve următoarea problemă: Vreau să organizez o petrecere, vreau să angajez un muzician, bucătar și serviciu de curățenie. Am la dispoziție 3 companii, fiecare poate furniza un singur serviciu. Ce companie trebuie să furnizeze fiecare serviciu astfel încât costul total să fie minim?

Companie	Cost muzician	Cost bucătar	Cost curăţenie
A	108	125	150
B	150	135	175
C	122	148	250

Fie matricea asociată tabelului:

108	125	150
150	135	175
122	148	250

Pas 1. Se scade valoarea minimă de pe fiecare rând din fiecare element de pe rând:

0	17	42
15	0	40
0	26	128

Pas 2. Se scade valoarea minimă de pe fiecare coloană din fiecare element de pe coloană:

0	17	2
15	0	0
0	26	88

Pas 3. Se desenează linii pe rândurile și coloanele din matrice ce conțin valoarea 0 astfel încât să se traseze cât mai puține linii:

0	17	2
15	0	0
0	26	88

Au fost trasate doar două linii ($2 < n = 3$), algoritmul continuă.

Se caută cea mai mică valoare care nu este acoperită de nicio linie. Se scade această valoare de pe fiecare rând pe care nu s-au trasat linii și apoi se adaugă la fiecare coloană pe care am trasat linii. Apoi, se revine la Pasul 3.

-2	15	0
15	0	0
-2	24	86

(a) Scade val. minimă

0	15	0
17	0	0
0	24	86

(b) Adaugă pe col.

Se revine la Pasul 3:

0	15	0
17	0	0
0	24	86

Se trasează 3 linii, $n = 3$, algoritmul a terminat. Se alege o alocare prin alegerea unui set de valori 0 astfel încât fiecare rând sau coloană sa aibă o singură valoare selectată.

0	15	0
17	0	0
0	24	86

Ex. compania C trebuie sa furnizeze muzicianul, compania A trebuie să furnizeze serviciul de curățenie → compania B ne dă bucătarul.

Soluția finală:

108	125	150
150	135	175
122	148	250

Se poate verifica soluția și exhaustiv:

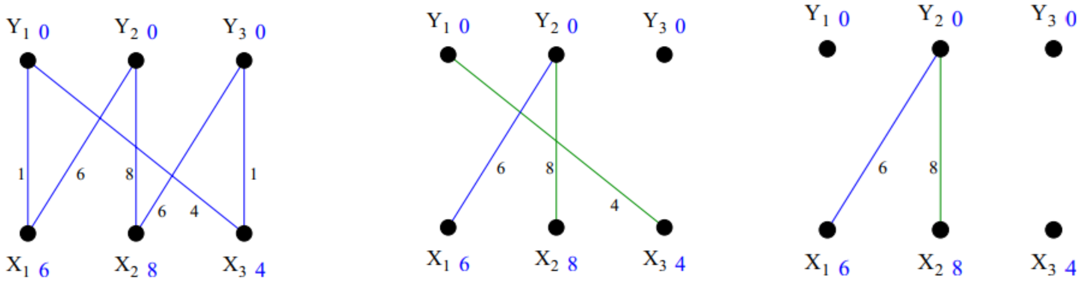
- $108 + 135 + 250 = 493$
- $108 + 148 + 175 = 431$
- $150 + 125 + 250 = 525$
- $150 + 148 + 150 = 448$
- $122 + 125 + 175 = 422$
- $122 + 135 + 150 = 407$

Algoritmul pentru metoda maghiară este prezentat mai jos. Complexitatea algoritmului $O(V^4)$, ulterior redusă la $O(V^3)$.

metoda_maghiară(G)

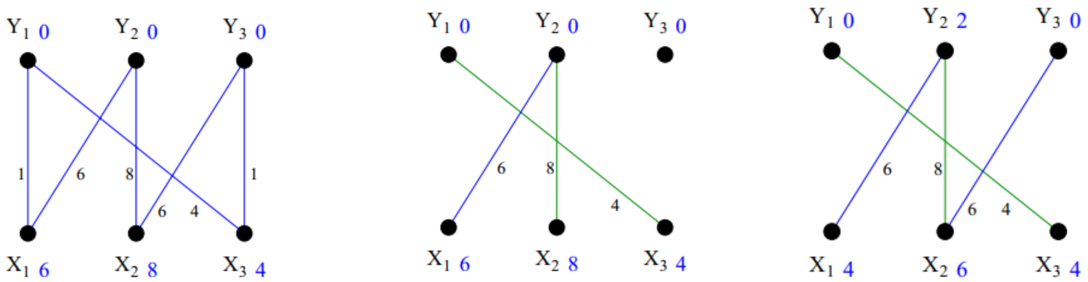
1: generează o etichetare inițială l și un cuplaj M în E_l
2: **if** M nu este un cuplaj perfect **then**
3: alege un vârf liber $x \in X$
4: $S = \{u\}, T = \varnothing$
5: **if** $N_l(S) = T$ **then**
6: actualizează etichetele conform (1) și (2) (forțând $N_l(S) \neq T$)
7: **if** $N_l(S) \neq T$ **then**
8: alege $y \in N_l(S) - T$
9: **if** y e liber **then**
10: $u - y$ este un lanț-M de creștere.
11: îmbunătățește M și sari la linia 2
12: **else**
13: $T = T \cup \{y\}$ și sari la linia 5

Metoda maghiară - exemplu pe un graf Figura de mai jos prezintă graful inițial, etichetarea vârfurilor și graful egal asociat.



(a) graful iniţial (b) E_l şi cuplaj (c) lanţ alternant

- Cuplajul inițial $M = \{(x_3, y_1), (x_2, y_2)\}$, $S = \{x_1\}$, $T = \varnothing$;
- deoarece $N_l(S) \neq T$ se merge la linia 7: alege $y_2 \in N_l(S) - T$;
- y_2 este în cuplaj, se crește lanțul alternant prin adăugarea lui (y_2, x_2), $S = \{x_1, x_2\}$, $T = \{y_2\}$;
- $N_l(S) = T$, se sare la linia 5.

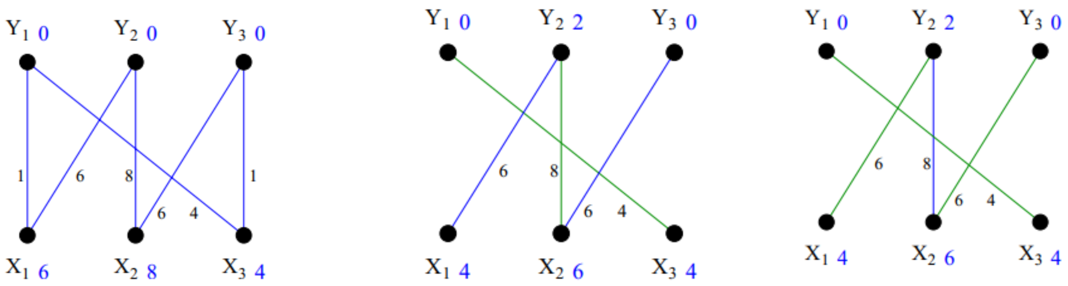


(a) graful iniţial (b) vechiul graf E_l şi M (c) noul graf E_l şi M

- $S = \{x_1, x_2\}$, $T = \{y_2\}$ și $N_l(S) = T$;
- se determină $\alpha_l =$

$$\alpha_l = \min_{x \in S, y \notin T} \begin{cases} 6 + 0 - 1 & (x_1, y_1) \\ 6 + 0 - 0 & (x_1, y_3) \\ 8 + 0 - 0 & (x_2, y_1) \\ 8 + 0 - 6 & (x_2, y_3) \end{cases} = 2$$

- se reduc etichetele lui S cu 2, se măresc etichetele lui T cu 2;
- acum $N_l(S) = \{y_2, y_3\} \neq \{y_2\} = T$, $S = \{x_1, x_2\}$.



(a) graful iniţial (b) noul graf E_l şi M (c) cuplaj nou

- Se alege $y_3 \in N_l(S) - T$ și se adaugă în T;
- y_3 nu e în cuplaj, s-a găsit un lanț de creștere (x_1, y_2, x_2, y_3);
- cuplajul $\{(x_1, y_2), (x_2, y_3), (x_3, y_1)\}$ are costul $6+6+4 = 16$ care este egal cu suma etichetelor grafului final.

11.3 Referințe

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2009. Introduction to Algorithms, Third Edition (3rd ed.). The MIT Press.
2. Geir Agnarsson and Raymond Greenlaw. 2006. Graph Theory: Modeling, Applications, and Algorithms. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
3. Mark Newman. 2010. Networks: An Introduction. Oxford University Press, Inc., New York, NY, USA.
4. Cristian A. Giumale. 2004. Introducere în analiza algoritmilor, teorie și aplicație. Polirom.
5. cursuri Teoria grafurilor: Zoltán Kása, Mircea Marin, Toadere Teodor.
6. `https://www-m9.ma.tum.de/graph-algorithms/matchings-hungarian-method/index_en.html`