Dumitru Fanache

TEORIA ALGORITMICĂ A GRAFURILOR

NOŢIUNI FUNDAMENTALE

Volumul I

ÎNVĂȚARE DE EXCELENȚĂ®
supersucces

Cuprins

capitolul I

Noţiuni generale despre grafuri

I.1. Scurt istoric al teoriei grafurilor

În problemele care apar în matematică, economie, inginerie, în general, şi în multe alte domenii, apare adeseori necesitatea reprezentării unor relații arbitrare între diferite obiecte, respectiv a interconexiunilor dintre acestea.

Spre exemplu, dându-se traseele aeriene ale unui stat, se cere să se precizeze drumul optim dintre două oraşe. Criteriul de optimalitate poate fi timpul sau prețul, drumul optim putând să difere pentru cele două situații.

Circuitele electrice sunt alte exemple în care interconexiunile dintre obiecte joacă un rol central. Piesele (tranzistoare, rezistențe, condensatoare) sunt interconectate prin fire electrice. Astfel de circuite pot fi reprezentate şi prelucrate de către un sistem de calcul în scopul rezolvării unor probleme simple, cum ar fi: *Sunt toate piesele date conectate în acelaşi circuit?* Sau a unor probleme mai complicate, de tipul: *Este funcțional un anumit circuit electric?*

Un al treilea exemplu îl reprezintă planificarea activităților, în care obiectele sunt *task-uri* (activități, procese), iar interconexiunile precizează care dintre activități trebuie finalizate înaintea altora. Întrebarea la care trebuie să oferim un răspuns este: *Când trebuie planificată fiecare activitate*?

Structurile de date care pot modela în mod natural situații de genul celor de mai sus sunt cele derivate din conceptul matematic cunoscut sub denumirea de *graf*.

Probleme care conduceau implicit la grafuri erau cunoscute de foarte multă vreme, fără să se cunoască însă prezența acestora.

Oficial, actul de naştere al teoriei grafurilor poate fi reprezentat de rezolvarea *problemei celor şapte poduri din Königsberg* de către *L. Euler* în anul 1736: două insule *A* şi *B*, formate de râul Pragel în Königsberg (Prusia orientală, acum oraşul Kaliningrad din vestul Rusiei) sunt conectate între ele şi de malurile *C* şi *D* prin şapte poduri, *fig*.I.1.(*a*), formând, în acea vreme, patru cartiere. În plimbările lor, localnicii încercau să traverseze toate cele şapte poduri, astfel încât, plecând dintr-un anumit cartier, să se înapoieze în acesta trecând o singură dată peste fiecare pod. Nu reuşeau să facă acest lucru.

I.9. Grafuri orientate, mulţimi dominante

Definiția I.37. *Fie V o mulțime finită şi nevidă. Numim graf orientat (digraf) orice pereche $G = (V, E)$ în care $E \subset V \times V$ este o mulțime finită de perechi ordonate cu componente din V (E este o relație binară pe V).*

Elementele mulțimii V se numesc *vârfuri* sau *noduri*, iar elementele lui E se numesc *arce*. Orice arc are forma (a, b), în care a se numeşte *extremitate inițială*, iar b se numeşte *extremitate finală* a arcului (a, b).

Exemplu: Fie graful reprezentat în *fig.* I.24, unde mulțimea vârfurilor este $V = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ şi mulțimea arcelor $E \subset V \times V$: $E = \{(1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (6, 7), (7, 4), (8, 7), (3, 8), (8, 9), (9, 2)\}$.

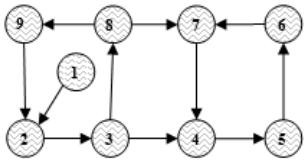


Fig. I.24. *Un graf orientat cu 9 vârfuri*

Un graf orientat se reprezintă printr-o mulțime de puncte corespunzătoare vârfurilor şi printr-o mulțime de segmente orientate (săgeți) corespunzătoare arcelor. O săgeată este orientată de la extremitatea inițială spre extremitatea finală a arcului pe care îl reprezintă. Dacă $u = (x, y) \in E$, spunem că x şi y sunt *adiacente* în G şi că nodurile x şi y sunt *incidente* arcului u sau arcul u este incident nodurilor x şi y. Mai exact, spunem că u este *incident exterior* nodului x (u pleacă sau iese din x) şi că u este *incident interior* nodului y (u ajunge sau intră în y). Arcul $(x, y) \in E$ se mai notează şi prin xy.

Definiția I.38. *Pentru graful $G = (V, E)$ mulțimea $S \subseteq V$ se numeşte mulțime exterior stabilă dacă $(\forall)\ x_j \notin S, \exists\ (x_i, x_j), x_i \in S$. Mulțimea S este una dominantă dacă $S \cup \Gamma^+(S) = V$.*

Definiția I.39. *Mulțimea dominantă S se numeşte minimă dacă nu există o altă mulțime dominantă S' pentru care este îndeplinită condiția $S' \subseteq S$.*

Nu toate mulțimile minime dominante au acelaşi număr de vârfuri $|S|$. Prin urmare, modul de selecție a *celei mai bune mulțimi dominante* va depinde de condițiile inițiale ale problemei studiate.

Definiția I.40. *Fie Q mulțimea tuturor mulțimilor dominante ale grafului $G = (V, E)$. Numărul $\beta_0(G) = \min_{S \in Q} |S|$ se va numi indice de dominanță (număr de stabilitate externă) a grafului G, iar mulțimea S^* pentru care el se obține – mulțime dominantă de putere minimă.*

Legătura dintre mulțimile maxim independente şi mulțimile dominante este evidentă. Algoritmul folosit pentru determinarea mulțimilor dominante va fi organizat după o tehnică similară celui pentru determinarea mulțimilor independente.

Dumitru Fanache

TEORIA ALGORITMICĂ A GRAFURILOR

REȚELE, CUPLAJE, COLORĂRI, PLANARITATE

Volumul II

ÎNVĂȚARE DE EXCELENȚĂ®
supersucces

Cuprins

capitolul VII | Reţele de flux, drumuri critice

VII.1. Reţele de flux

Definiţia VII.1. *O reţea de flux este un graf orientat ponderat şi fără circuite* $G = (V, E)$ *ce îndeplineşte condiţiile următoare:*

- ✓ *există un vârf unic* $s \in V$ *în care nu intră niciun arc (sursa reţelei);*
- ✓ *există un vârf unic* $t \in V$ *din care nu iese niciun arc (destinaţia reţelei);*
- ✓ *G este conex şi există drumuri de la s la t;*
- ✓ *s-a definit o funcţie* $c : E \to \mathbb{R}$ *astfel încât* $c(u) \geq 0$ *pentru* $(\forall)\ u \in E$, *numărul* $c(u)$ *se numeşte capacitatea reţelei.*

Problema determinării fluxului maxim este cea mai simplă problemă legată de reţelele de flux (transport), care cere să se determine *cantitatea maximă de material* care poate fi transportată de la *sursă* la *destinaţie* ţinând cont de restricţiile de capacitate.

Definiţia VII.2. *Un flux într-o reţea de transport* $G = (V, E)$ *este o funcţie* $\varphi : V \times V \to \mathbb{R}^+$, *care respectă următoarele condiţii:*

- ✓ $(\forall)\ u, v \in V,\ c(u, v) \geq \varphi(u, v)$ *(limitarea impusă de capacităţi);*
- ✓ $(\forall)\ u, v \in V,\ \varphi(u, v) = -\varphi(v, u)$ *(antisimetrie);*
- ✓ $(\forall)\ u \in V,\ u \neq s, t,$ *unde* $s, t \in V,\ \sum_{v \in V} \varphi(u, v) = 0$ *(conservarea fluxului).*

Observaţii: Aceste 3 condiţii reies şi intuitiv:

- Fluxul pe fiecare muchie nu trebuie să depăşească capacitatea muchiei;
- Fluxul dintre nodurile u şi v care nu sunt legate prin niciun arc, nu poate fi decât zero. Dacă $(u, v) \notin E$ şi $(v, u) \notin E$, atunci $c(u, v) = c(v, u) = 0$; din cauza restricţiei de capacitate avem $\varphi(u, v) \leq 0$ şi $\varphi(v, u) \leq 0$. Dar, din condiţia de antisimetrie avem $\varphi(u, v) = -\varphi(v, u)$, prin urmare $\varphi(u, v) = \varphi(v, u) = 0$. Rezultă că existenţa fluxului nenul între vârfurile u şi v implică $(u, v) \in E$ sau $(v, u) \in E$ sau ambele;
- Condiţia de conservare a fluxului afirmă că pentru orice vârf x cu $x \neq s$ şi $x \neq t$, suma fluxurilor de pe arcele care intră în x este egală cu suma fluxurilor pe arcele care ies din x.
- Funcţia φ nu este unică.

VII.3.2. Algoritmul Edmonds–Karp

Aşa cum am văzut, *algoritmul Ford–Fulkerson* nu defineşte o metodă de alegere a drumului de ameliorare pe baza căruia se modifică fluxul în graf.

Edmonds[3] şi Karp au sugerat ca metodă de construcţie a fluxului φ^* în reţeaua reziduală *determinarea celui mai scurt drum între sursă şi destinaţie folosind o căutare în lăţime în graful rezidual unde fiecare arc are ponderea 1 şi saturarea acestui drum*.

Se observă intuitiv că fiecare execuţie a acestui proces saturează cel puţin o muchie. După $O(m)$ astfel de execuţii, toate drumurile minime dintre sursă şi destinaţie sunt saturate şi distanţa dintre sursă şi destinaţie trebuie să crească. Aceasta implică faptul că după $O(n \times m)$ astfel de execuţii, distanţa dintre s şi t va deveni mai mare decât n, sau altfel spus, t nu va mai fi accesibil din s. Folosind *algoritmul Edmonds–Karp* pentru a determina fluxul φ^* în reţeaua reziduală, complexitatea algoritmului Ford–Fulkerson devine $O(n \times m^2)$ (vezi codul asociat algoritmului din *fig.*VII.8).

```
Input G=(V,E),s,t
Output φ
while (BFS(Gφ,s,t)==true) do
    P←(s→t)
    find φ*=min{(i,j) | (i,j)∈P}
    for each (i,j)∈P do
        cij ← cij -φ*
        cji ← cji + φ*
φ ←φ+φ*
return φ
```

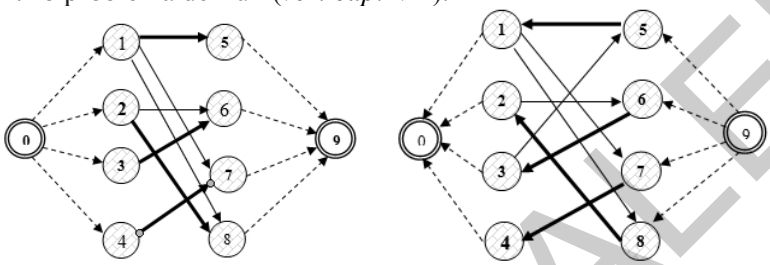
Fig. VII.6. *Algoritmul Edmonds–Karp*

În urma execuţiei *algoritmului Edmonds–Karp*, obţinem pentru reţelele din *fig.*VII.1**,** *fig.*VII.5 rezultatele din *fig.*VII.7(*a, b*).

[3] **Jack Edmonds**, matematician canadian, considerat ca unul din cei mai importanţi specialişti în optimizare combinatorie; în 1985 a primit premiul John von Neumann.

VIII.2. Determinarea unui cuplaj maxim folosind o rețea de transport

Pentru a determina cuplajul maxim putem transforma graful bipartit într-o rețea de transport adăugând un nod sursă (*s*) şi un nod destinație (*t*) şi vom atribui capacitatea 1 pe toate arcele (vezi *fig.* VIII.3). În acest fel am transformat o problemă de afectare într-o problemă de flux (*vezi cap.* VII).



(*a*) (*b*)

Fig.VIII.3. *Transformarea unei probleme de cuplaj într-o problemă de flux*
(*a) rețeaua de transport inițială obținută prin adăugarea a două vârfuri la graful bipartit*;
(*b*) *rețeaua a fost modificată prin inversarea arcelor pe drumurile de creştere*

a) *Cuplaj maxim pentru un graf memorat prin liste de adiacență*

Pentru graful bipartit din *fig.*VIII.3, am adăugat vârfurile 0 şi 9, ca fiind sursa şi destinația rețelei de transport. Listele de adiacență, gradele vârfurilor precum şi drumurile de creştere de la vârfurile sursă(0) şi destinație (9) determinate prin parcurgerea BF sunt date în *fig*.VIII.4.

Liste	$d^-(x)$	$d^+(x)$	Drumuri de creştere
0: 1,2,3,4	0	4	
1: 5,7,8	1	3	
2: 6,8	1	2	I).0,1,5,9
3: 5,6	1	2	II). 0,2,8,9
4: 7	1	1	III). 0,3,6,9
5: 9	2	1	IV). 0,4,7,9
6: 9	2	1	
7: 9	2	1	
8: 9	2	1	
9: 0	4	0	

Fig.VIII.4. *Lista de adiacență a vârfului de start nu este vidă*

După determinarea unui drum de ameliorare, se schimbă sensul arcelor drumului respectiv şi evident că listele de adiacență se vor modifica astfel că, după patru paşi ele vor arăta ca în *fig*. VIII.5. Observăm că lista de adiacență a vârfului de start (0) devine vidă.