# ARBORI DE LUNGIME MINIMĂ ÎNTR-UN GRAF

# *MINIMUM LENGTH TREES IN A GRAPH*

**Col.(r) prof.univ.dr. Gelu ALEXANDRESCU***

Utilizarea grafurilor identifică şi analizează multilateral ansamblul legăturilor dintre elementele componente ale unui sistem oarecare, ceea ce asigură o formă convenabilă de elaborare mai simplă, precisă şi rapidă a modelului matematic care permite creşterea gradului de eficacitate a procesului de optimizare.

*The use of graphs identifies and multilaterally analyzes the links between the components of an arbitrary system, which provides a convenient manner of a simpler, more accurate and quick design of the mathematical model, thus allowing the improvement of efficiency of the optimization process.*

**Cuvinte-cheie**: graf; arc; vârf; muchie; arbore; algoritm.
***Keywords**: graph; arc; peak; edge; tree; algorithm.*

Graful este constituit dintr-o pereche $G = (X,\Gamma)$, în care $X$ reprezintă mulţimea de elemente numite *vârfuri* (puncte, noduri) $X - \{x_1, x_2, \cdots, x_n\}$, iar $\Gamma$ este o aplicaţie de la $X$ în mulţimea părţilor lui X, care pune în evidenţă tocmai conexiunile dintre vârfurile $x_i$ ale grafului.

Diversitatea problemelor în care apare conceptul de „graf” determină o abstractizare a acestuia, el constituind, de fapt, o mulţime de elemente finite, între care există legături orientate[1].

Un graf neorientat este un cuplu G = (X, U), format dintr-o mulţime X şi o mulţime U de submulţimi nevide ale lui X. Elementele mulţimii X se numesc *nodurile* (vârfurile) grafului neorientat G, iar elementele mulţimii U poartă numele de *muchii*. Astfel, dacă nodurile unui graf sunt legate prin muchii, graful este *neorientat*. Muchia este un ansamblu de două elemente $x_i$ şi $x_j$ astfel încât:

$$(x_i, x_j) \in U \text{ şi/sau } (x_j, x_i) \in U \text{ cu } x_i \neq x_j$$

Cu alte cuvinte, o muchie este o pereche de vârfuri, legate printr-un arc, într-un sens sau în altul, sau prin două arce de sensuri opuse.

Într-un graf neorientat, se poate vorbi despre vârfuri adiacente, muchii adiacente ca şi într-un graf orientat.

Deci, între două vârfuri (noduri) adiacente ale unui graf, există o muchie, dacă cel puţin un arc poate să le lege într-un sens sau în altul.

Dacă $m = (x_1, x_2)$ este o muchie, vârfurile $x_1, x_2$ se numesc extremităţile muchiei $m$.

O muchie de forma $(x_i, x_i)$ se numeşte bucla de extremitate $x_i$. Pentru omogenizarea limbajului, se va spune că bucla $(x_i, x_i)$ are două extremităţi confundate în vârful $x_i$.

Graful neorientat G = (X, U) se numeşte *finit*, dacă mulţimea X este finită. Putem realiza un graf neorientat G, dispunând arbitrar, în plan, vârfurile sale şi unind, printr-o linie continuă, acele vârfuri între care există muchii.

De exemplu, între localităţile $x_1, x_2, x_3, x_4, x_5$, există următoarele drumuri rutiere:

- o autostradă care trece prin localităţile $x_1, x_3, x_4$;
- un drum naţional care trece prin $x_2, x_4, x_5$;
- drumuri judeţene între $x_2$ şi $x_5$; între $x_1$ şi $x_4$; între $x_1$ şi $x_5$.

Construim un graf neorientat G = (X, U), în care:

$X = \{x_1, x_2, x_3, x_4, x_5\}$
$U = \{(x_1, x_3), (x_3, x_4), (x_2, x_4), (x_4, x_5), (x_2, x_5), (x_1, x_4), (x_1, x_5)\}$

În *figura 1* este dată reprezentarea grafică a acestui graf.

***Universitatea Naţională de Apărare „Carol I”***
e-mail: *alexandrescugelu@yahoo.com*

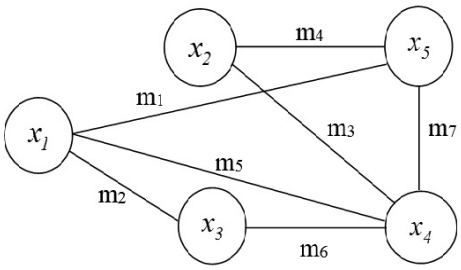


**Fig. 1** *Graf neorientat*

Dintr-un graf orientat G = (X, Γ), se poate obţine un graf neorientat G = (X, U), astfel:

- graful neorientat are aceleaşi vârfuri ca şi cel orientat;
- în graful neorientat, există arcul ($x_i$, $x_j$), arcul ($x_j$, $x_i$) sau amândouă.

Un *lanţ* este o succesiune de muchii:

$$l = (m_1, m_2, ..., m_p),\ p > 3$$

ale grafului neorientat G = (X, U), fiecare muchie $m_k$ fiind legată de muchia $m_{k-1}$ printr-una dintre extremităţile sale şi de $m_{k+1}$ prin cealaltă extremitate.

Un graf se numeşte *conex*, dacă:

$\forall x_i, \forall x_j$ cu $(x_i \neq x_j)$, există un lanţ care merge de la $x_i$ la $x_j$.

Deci, dacă între oricare pereche de vârfuri ale grafului există, cel puţin, un lanţ, graful se numeşte *conex,* iar dacă pentru orice cuplu de vârfuri $x_i$ şi $x_j$, $(x_i \neq x_j)\forall$ există un lanţ de la $x_i$ spre $x_j$, graful se numeşte *tare conex*.

*Lungimea unui lanţ* este dată de *numărul muchiilor componente*.

Se numeşte *grad al unui vârf* $x_i$ numărul de muchii care au o extremitate în $x_i$ (cealaltă extremitate fiind diferită de $x_i$). Astfel, gradul unui vârf[2] dintr-un graf neorientat reprezintă numărul de muchii incidente cu vârful respectiv şi se notează cu *d(x)*. Când *d(x) = 0*, vârful se numeşte *izolat,* iar dacă *d(x) =1*, vârful se numeşte *terminal*.

*Graful se numeşte complet,* dacă oricare dintre cele două vârfuri ale acestuia este adiacent, adică există o muchie între $x_i$ şi $x_j$.

*Numărul de muchii* ale unui graf complet se calculează cu relaţia[3]: $C_n^2 = \frac{n(n-1)}{2}$

Matricea booleană asociată unui *graf de structură neorientat* este *simetrică* faţă de diagonala sa principală, deoarece conţinutul liniei *i* este identic cu cel al coloanei *j*.

Un exemplu de graf cu structura simetrică, exprimat prin muchiile acestuia, se prezintă în figura 2, iar matricea asociată grafului are conţinutul din figura 3.

Folosind matricea booleană, procedeul de marcaj constă în următoarele: coloana lui $x_i$ se consideră marcată; se marchează cu *i* toate coloanele care conţin câte un element 1 pe linia lui $x_i$; dacă presupunem că toate coloanele $x_m$, ..., $x_n$ au fost marcate, atunci în fiecare linie corespunzătoare unui vârf $x_k$, $m \leq k \leq n$, considerăm elementele 1 şi marcăm cu *k* coloanele care le conţin şi n-au fost marcate.

Un concept important al grafurilor neorientate este noţiunea de *arbore*. Un graf se numeşte arbore, dacă nu are cicluri şi are cel puţin două noduri. Un graf parţial şi fără cicluri al unui graf *G = (X, U)* este un arbore, care aparţine grafului *G = (X, U)*.

Un arbore poate fi definit în mai multe moduri. Dacă graful *G = (X, U)* are cel puţin două vârfuri, pentru ca acesta să reprezinte un arbore, este

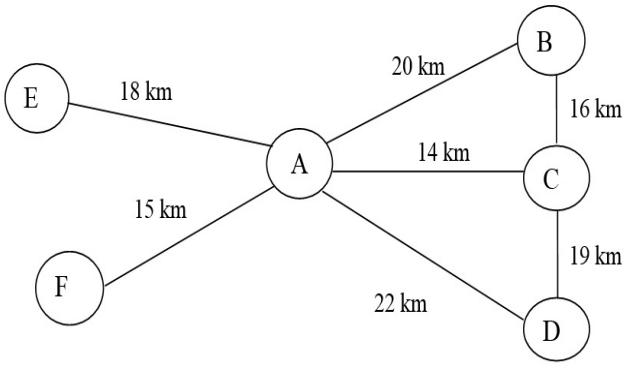


**Fig 2** *Graf cu structura simetrică*

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| A | 0 | 20 | 14 | 22 | 18 | 15 |
| B | 20 | 0 | 16 | 0 | 0 | 0 |
| C | 14 | 16 | 0 | 19 | 0 | 0 |
| D | 22 | 0 | 19 | 0 | 0 | 0 |
| E | 18 | 0 | 0 | 0 | 0 | 0 |
| F | 15 | 0 | 0 | 0 | 0 | 0 |

**Fig. 3** *Reprezentarea matriceală a grafului cu structura simetrică*

suficient să fie îndeplinită una dintre proprietăţi:

1) *G = (X, U)* este conex şi fără cicluri;

2) *G = (X, U)* este fără ciclu şi are *n-1* muchii, *n* fiind numărul de vârfuri;

3) *G = (X, U)* este conex şi are *n-1* muchii, *n* fiind numărul de vârfuri;

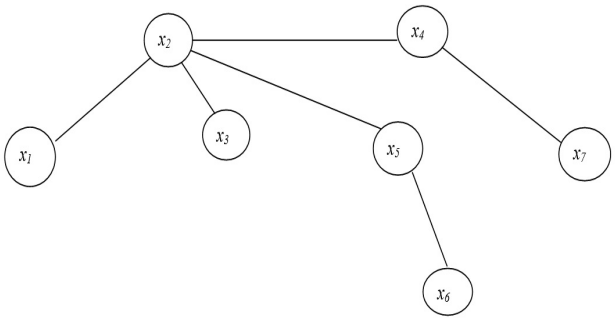


**Fig. 4** *Structura unui arbore*

4) *G = (X, U)* este fără ciclu şi, adăugând o muchie între două vârfuri neadiacente, se creează un ciclu şi numai unul;

5) *G = (X, U)* este conex şi, suprimând o muchie oarecare, acesta nu mai este conex;

6) orice cuplu de vârfuri este legat printr-un singur lanţ şi numai unul.

O proprietate importantă a arborelui, care rezultă din însăşi definiţia lui, este aceea că, suprimând un arc oarecare al acestuia, el îşi pierde proprietatea de a mai fi conex.

Astfel, un graf *G = (X, U)* admite un arbore, având aceeaşi mulţime de vârfuri *X,* dar arce mai puţine, dacă şi numai dacă el este conex.

Un graf *G = (X, U)* poate conţine un graf parţial, care să fie arbore, numai dacă este conex. Un astfel de arbore se numeşte „arbore parţial”.

În figura 4 se prezintă aspectul general al unui arbore.

Într-un arbore, orice pereche de vârfuri este legată printr-un singur lanţ. În mulţimea vârfurilor unui arbore, se întâlnesc vârfuri pentru care nu există decât o muchie incidentă; aceste vârfuri poartă denumirea de *vârfuri terminale*. În figura 4 astfel de vârfuri sunt: $x_1$, $x_3$, $x_6$, $x_7$. Un arbore are cel puţin două vârfuri terminale.

Dacă graful *G = (X, U)* are *n* vârfuri, atunci numărul muchiilor este egal cu *n-1*.

Dacă graful *G = (X, U)* conţine cicluri, există posibilitatea suprimării unor arce, în aşa fel încât graful parţial rămas să fie conex şi fără cicluri, adică să constituie un arbore.

Se poate identifica un algoritm pentru determinarea unui arbore într-un graf conex. Astfel, în graful *G = (X, U)* :

- se pleacă de la un arc arbitrar $u_{i_1} \in U$;
- în graful se alege un arc $u_{i_2} \in U$, care nu formează ciclu cu arcul $u_{i_1} \in U$;
- se alege din nou un arc $u_{i_3} \in U$, care nu formează cicluri cu arcele $u_{i_1}$, $u_{i_2} \in U$;
- se continuă alegerea arcelor $u_{i_k} \in U$, în condiţiile prezentate, cât este posibil.

Presupunem că arcul $u_{i_p} \in U$, este ultimul arc ales. Succesiunea de arce $(u_{i_1}, u_{i_2}, ..., u_{i_k}, ..., u_{i_p})$ formează un arbore în graful dat.

Un caz particular de arbore este *arborescenţa*. Un graf finit *G = (X, U)* se numeşte arborescenţă cu rădăcina $x_0$ , dacă:

- are cel puţin două noduri;
- orice nod $x_j \neq x_0$ este extremitatea terminală a unui singur arc;
- $x_0$ nu este extremitatea terminală a niciunui arc;
- nu conţine circuite.

Orice arborescenţă este un arbore.

Pentru un graf dat *G = (X, U)*, definit de mulţimea vârfurilor şi de mulţimea muchiilor sale, se poate calcula arborele de lungime minimă. Astfel, fiecărei muchii $m_i \in U$ îi corespunde un număr $l(m_i) \geq 0$, numit lungimea muchiei $m_i$. În aceste condiţii, se pune problema determinării arborelui parţial a cărui lungime totală să fie minimă, adică:

$$\sum_{m_i \in M^*} l(m_i) = \text{minimă},$$

unde $M^*$ reprezintă mulţimea muchiilor care intră în compunerea arborelui.

Pentru identificarea arborelui de lungime minimă, au fost identificaţi mai mulţi algoritmi, printre care se pot menţiona algoritmii lui Kruskal şi Sollin.

*Algoritmul Kruskal*

Se pleacă de la următoarea lemă: dacă într-un graf complet *G*=(X, U) lungimile $l(m_i)$ asociate muchiilor $m_i$ sunt diferite, atunci determinarea arborelui de lungime minimă admite o soluţie, şi numai una, notată cu $G = (X, \overline{U})$.

Mulţimea $\overline{U}_{n-1} = \{\overline{m}_1, \overline{m}_2, ..., \overline{m}_{n-1}\}$ se obţine astfel:

- se ia în considerare cea mai scurtă muchie,

notată cu $\overline{m}_1$ ;

- se caută muchia $\overline{m}_2$ minimă, astfel încât $\overline{m}_2 \neq \overline{m}_1$ şi $\overline{U}_2 = \{\overline{m}_1, \overline{m}_2\}$ să nu conţină cicluri;
- se ia în considerare muchia minimă $\overline{m}_3$, astfel încât $\overline{m}_3 \neq \overline{m}_1$, $\overline{m}_3 \neq \overline{m}_2$, iar $\overline{U}_3 = \{\overline{m}_1, \overline{m}_2, \overline{m}_3\}$ să nu conţină cicluri;
- în continuare, se repetă procedeul.

În felul acesta, se obţine o mulţime de $(n-1)$ muchii, notate cu $\overline{U}_{n-1} = \{\overline{m}_1, \overline{m}_2, \ldots, \overline{m}_{n-1}\}$, iar subgraful $G = (X, \overline{U}_{n-1})$ reprezintă un arbore de lungime totală minimă.

Astfel, algoritmul Kruskal constă în alegerea, pe etape, a celei mai mici muchii, astfel încât aceasta să nu formeze cicluri cu muchiile alese anterior.

*Algoritmul Sollin*

Acest algoritm, pentru determinarea arborelui de lungime minimă, foloseşte noţiunea de depărtare de la un vârf la un drum. De exemplu, distanţa de la vârful $x_i$ la drumul $\mu = (u_{j_1}, u_{j_2}, ..., u_{i_k})$ este

$$\min \{(x_i, x_{j_1})(x_i, x_{j_2}) \cdots, (x_i, x_{j_k})$$

Prin intermediul distanţei de la un vârf la un drum, se stabileşte vârful cel mai apropiat de vârful dat, ca fiind vârful care se găseşte faţă de un alt vârf la o distanţă mai mică decât faţă de celelalte vârfuri.

Astfel, algoritmul lui Sollin constă în formarea unor subarbori, prin unirea a câte unui vârf al grafului cu vârful faţă de care muchia are valoarea cea mai mică.

Primul subarbore elimină două vârfuri ale grafului. Apoi, se alege alt vârf, care nu există în subarborele precedent, şi se uneşte cu vârful cel mai apropiat, în sensul în care muchia are valoarea cea mai mică; se obţine un alt subarbore. Procedeul continuă până la obţinerea tuturor subarborilor din graf. Ultima pereche de subarbori conduce la un arbore, care reprezintă arborele de lungime minimă al grafului.

Metodele folosite de teoria grafurilor permit o soluţionare intuitivă, uşoară şi rapidă a unor probleme complexe, care, altfel, ar necesita operaţii complicate[4]. Teoria grafurilor are o largă utilizare în aproape toate structurile, urmărind creşterea eficienţei de rezolvare a diferitelor probleme specifice domeniului de activitate.

**NOTE:**

1 Florentina-Loredana Dragomir, *Eficientizarea acţiunilor militare utilizând teoria grafurilor*, Buletinul Universităţii Naţionale de Apărare „Carol I”, nr. 2, Bucureşti, 2017, p. 49.

2 Adrian Florea, *Elemente de teoria grafurilor,* http// *webspace.ulbsibiu.ro/adrian.florea/ html/.../Planificare_Grafuri.pdf*

3 S. Cataranciuc, *Teoria grafurilor în probleme şi aplicaţii*, 2004, http//www.math. md/studlib/matematica/teoria_graf/capitol_I.pdf

4 Florentina-Loredana Dragomir, *op.cit.*, p. 54.

**BIBLIOGRAFIE**

Dumitru Vasile, Rugină N., Stoian I., *Modelarea matematică a acţiunilor militare*, Editura Diagonal, Bacău, 2002.

Florea Adrian, *Elemente de teoria grafurilor,* http//webspace. ulbsibiu.ro/adrian.florea/html/.../Planificare_Grafuri.pdf

Grad Vasile, Stoian Ion, Kovacs Emil, Dumitru Vasile, *Cercetarea operaţională în domeniul militar*, Editura Sylvi, Bucureşti, 2000.

Grad Vasile ş.a., *Metode de fundamentare a deciziilor militare,* Editura Diagonal, Bacău, 2001.

Macovei Anamaria, *Cercetare operaţională*, Universitatea „Ştefan Cel Mare”, Suceava, 2009.