COMBINATORICĂ ŞI TEORIA GRAFURILOR. Partea a doua: Elemente de teoria grafurilor

Prof. dr. Ioan Tomescu
Facultatea de Matematică şi Informatică,
Universitatea din Bucureşti

NOŢIUNI ŞI DEFINIŢII DE BAZĂ

Prima lucrare de teoria grafurilor a fost scrisă de Euler in anul 1736, fiind consacrată "problemei celor 7 poduri" din Königsberg, in care a demonstrat imposibilitatea realizării unui traseu care să treacă o singură dată peste fiecare pod. Un graf neorientat este compus din două mulţimi: o mulţime de vârfuri şi o mulţime de muchii care sunt perechi de vârfuri. Aceste muchii pot reprezenta legăturile chimice dintr-o moleculă, rezistenţele sau tensiunile electromotoare dintr-o reţea electrică, legăturile rutiere sau feroviare dintre localităţi, muchiile unui poliedru, legăturile dintre nodurile unei reţele de calculatoare, ierarhiile dintr-o structură administrativă, meciurile directe dintre echipele participante la un turneu sportiv, graniţele dintre statele reprezentate pe o hartă. In secolul al XIX-lea dezvoltarea teoriei grafurilor a fost impulsionată de aplicaţiile in fizică (teoria reţelelor electrice), chimie (studiul fenomenelor de izomerism, descoperite de Crum Brown), studiul poliedrelor convexe şi a legăturii lor cu grafurile planare şi problema celor patru culori. In prezent teoria grafurilor este o ramură a matematicii, dar modelul de graf este utilizat in cercetarea operaţională pentru rezolvarea unor probleme de optimizare, in informatică pentru studiul reţelelor de calculatoare, al algoritmilor de sortare şi regăsirea informaţiei, al reţelelor de interconexiune din calculul paralel, in chimie, precum şi in multe alte domenii.

Mai intâi vom defini noţiunile de bază relative la grafurile neorientate, pe care le vom numi pe scurt grafuri. Un *graf* este o pereche $(V(G), E(G))$,

unde $V(G)$ este o mulţime finită şi nevidă de elemente numite *vârfuri* şi $E(G)$ este o mulţime de perechi neordonate de elemente distincte din $V(G)$ numite *muchii*. $V(G)$ se numeşte *mulţimea vârfurilor* şi $E(G)$ *mulţimea muchiilor* grafului G. O muchie $\{x, y\}$ se va nota mai simplu xy şi vom spune că ea uneşte vârfurile x şi y. Orice graf G poate fi desenat in plan reprezentând vârfurile sale prin puncte şi muchiile prin linii care unesc anumite perechi de vârfuri. Intr-o astfel de reprezentare nu contează distanţele dintre vârfuri sau unghiurile dintre muchii. Două vârfuri unite printr-o muchie se numesc *adiacente*. Cele două vârfuri care compun o muchie se numesc *extremităţile* muchiei. *Gradul* unui vârf x este egal cu numărul muchiilor care au o extremitate in vârful x şi se notează cu $d(x)$. Un vârf de gradul zero se numeşte *vârf izolat*, iar un vârf de gradul unu este numit *vârf terminal*.

PROPOZIŢIA 1. Pentru orice graf suma gradelor este egală cu dublul numărului de muchii, adică $\sum_{x \in V(G)} d(x) = 2|E(G)|$.

Demonstraţie. Această proprietate rezultă din aceea că fiecare muchie xy a unui graf are două extremităţi x şi y, ea contribuind cu o unitate atât la $d(x)$ cât şi la $d(y)$. □

De aici rezultă că pentru orice graf numărul vârfurilor de grad impar este par.

Un *graf parţial* al unui graf G este un graf H care are aceeaşi mulţime de vârfuri cu G, deci $V(H) = V(G)$ şi $E(H) \subseteq E(G)$. Deci un graf parţial al lui G este G sau se obţine din G prin suprimarea anumitor muchii din G. Un *subgraf* al unui graf G este un graf H astfel incât $V(H) \subseteq V(G)$, iar muchiile lui H sunt toate muchiile lui G care au ambele extremităţi in mulţimea de vârfuri $V(H)$. Deci un subgraf H al unui graf G este graful G insuşi sau se obţine din G prin suprimarea anumitor vârfuri şi a tuturor muchiilor care au cel puţin o extremitate in mulţimea vârfurilor care au fost suprimate. Se spune că subgraful H este *indus* sau *generat* de mulţimea de vârfuri $V(H)$.

Un *lanţ* intr-un graf G este o succesiune finită de vârfuri ale lui G care se notează $[x_0, x_1, \ldots, x_r]$, cu proprietatea că oricare două vârfuri vecine sunt adiacente, adică $x_i x_{i+1} \in E(G)$ pentru orice $0 \leq i \leq r$. Vârfurile x_0 şi x_r se numesc extremităţile lanţului, iar numărul r care reprezintă numărul de muchii ale lanţului se numeşte *lungimea* acestui lanţ. Dacă vârfurile $x_0, x_1, \ldots, x_r$ sunt distincte, lanţul se numeşte *elementar*. Un lanţ poate fi interpretat ca traseul unei deplasări pe muchiile grafului. De aceea, un lanţ de extremităţi x_0 şi x_r se mai spune că este un lanţ de la x_0 la x_r. O altă noţiune fundamentală este aceea de ciclu. Un *ciclu* intr-un graf G este o succesiune finită de vârfuri ale lui G, notată $[x_0, x_1, \ldots, x_r]$, cu următoarele proprietăţi:

a) oricare două vârfuri vecine sunt adiacente, adică $x_i x_{i+1} \in E(G)$ pentru orice $0 \leq i \leq r-1$;

b) $x_0 = x_r$;

c) toate muchiile x_0x_1, x_1x_2,...,$x_{r-1}x_r$ sunt distincte.

Numărul r, care reprezintă numărul de muchii ale ciclului, se numeşte lungimea acestui ciclu. Dacă toate vârfurile $x_0, x_1, \ldots, x_{r-1}$ sunt distincte, ciclul se numeşte elementar. Deci un ciclu este un lanţ care se intoarce in punctul de plecare şi pentru care toate muchiile parcurse sunt distincte. Dacă nu am impune condiţia c), atunci orice graf care are o muchie ab ar conţine şi un ciclu şi anume $[a, b, a]$.

Un graf G se numeşte *conex* dacă pentru orice pereche de vârfuri distincte x, y ale lui G, există un lanţ de extremităţi x şi y in G. O *componentă conexă* C a unui graf G este un subgraf conex maximal al lui G (maximal in sensul că nu există nici un lanţ care să unească un vârf din C cu orice vârf care nu aparţine lui C). Grafurile conexe au o singură componentă conexă.
Graful *complementar* grafului G se notează prin $\overline{G}$ şi el are $V(\overline{G}) = V(G)$ şi $E(\overline{G}) = \overline{E(G)}$, adică complementara mulţimii muchiilor lui G, două vârfuri distincte fiind adiacente in $\overline{G}$ dacă şi numai dacă ele nu sunt adiacente in G.
Clase speciale de grafuri.

Un graf pentru care oricare două vârfuri sunt adiacente se numeşte *graf complet*. Graful complet cu n vârfuri se notează K_n şi el are $\binom{n}{2}$ muchii. Toate vârfurile grafului K_n au gradul $n-1$. Un graf se numeşte *regulat* dacă toate vârfurile sale au un acelaşi grad. Dacă acest grad comun al vârfurilor este egal cu k, graful se numeşte k-regulat. Graful complet K_n este $(n-1)$-regulat. Un interes special in clasa grafurilor regulate il reprezintă grafurile formate cu vârfurile şi muchiile celor 5 poliedre regulate. Astfel graful tetraedrului regulat este graful complet K_4; graful cubului este 3-regulat (astfel de grafuri se mai numesc *cubice*); el are 8 vârfuri. Graful octaedrului este 4-regulat şi are 6 vârfuri; graful dodecaedrului este cubic şi are 20 vârfuri, iar graful icosaedrului este 5-regulat şi are 12 vârfuri.

Ciclurile elementare cu p vârfuri se notează C_p. Dacă mulţimea vârfurilor unui graf G poate fi partiţionată in două mulţimi disjuncte de vârfuri, $V(G) = V_1 \cup V_2$, astfel incât fiecare muchie uneşte un vârf din V_1 cu un vârf din V_2, graful G se numeşte *bipartit*. Intr-un graf bipartit orice ciclu conţine un număr par de vârfuri, deoarece el trece alternativ din V_1 in V_2 şi se intoarce in punctul de plecare. De exemplu ciclurile impare C_{2k+1}, unde $k \in \mathbb{N}$ nu sunt bipartite şi vom vedea că inexistenţa ciclurilor impare caracterizează grafurile bipartite. Dacă orice vârf din V_1 este adiacent cu orice vârf din V_2, graful se numeşte *bipartit complet*. Un graf bipartit complet pentru care $|V_1| = m$ şi $|V_2| = n$ se notează $K_{m,n}$. Un graf bipartit complet de forma $K_{1,n}$, format dintr-un vârf central adiacent cu alte n vârfuri, neadiacente intre ele se numeşte *graf-stea*.

TEOREMA 1 (König). Un graf G este bipartit dacă şi numai dacă nu

conţine cicluri impare.

Demonstraţie. Am văzut că un graf bipartit nu conţine cicluri impare. Pentru a demonstra suficienţa, fie G un graf care nu conţine cicluri impare. Trebuie să demonstrăm că G este bipartit. Pentru aceasta vom defini o metrică pe multimea vârfurilor unui graf conex H in felul următor: distanţa dintre vârfurile $x, y \in V(H)$, notată $d(x, y)$, este egală cu 0 dacă $x = y$ şi este egală cu lungimea celui mai scurt lanţ dintre x şi y in caz contrar. Distanţa astfel definită este o metrică, deoarece ea este: a) nenegativă: $d(x, y) \geq 0$ şi $d(x, y) = 0$ dacă şi numai dacă $x = y$; b) simetrică: $d(x, y) = d(y, x)$ deoarece graful este neorientat; c) verifică inegalitatea triunghiului: $d(x, y) \leq d(x, z) + d(z, y)$ oricare ar fi $x, y, z \in V(H)$. Pentru a justifica ultima inegalitate, să considerăm un lanţ l_1 de lungime minimă intre x şi z şi un lant l_2 de lungime minimă intre z şi y. Prin concatenarea celor două lanţuri se obţine un lanţ intre x şi y, de lungime $d(x, z) + d(z, y)$. Cum $d(x, y)$ este lungimea minimă a lanţurilor dintre x şi y, rezultă inegalitatea c). Să trecem acum la demonstrarea suficienţei teoremei lui König şi să considerăm mai intâi graful G conex. Fie $x \in V(G)$ un vârf fixat al grafului G şi să notăm $V_i(x) = \{y | y \in V(G) \text{ şi } d(x, y) = i\}$. Este clar că $V_0(x) = \{x\}$ şi vom defini $A = \bigcup_{i \geq 0} V_{2i}(x)$ şi $B = \bigcup_{i \geq 0} V_{2i+1}(x)$. Vom arăta că orice muchie are o extremitate in A şi cealaltă extremitate in B. Să presupunem, prin reducere la absurd, că există o muchie uv având ambele extremităţi in A, deci există numerele $p, q \in \mathbb{N}$ astfel incât $u \in V_{2p}(x)$ şi $v \in V_{2q}(x)$. Să considerăm două lanţuri minime de la x la u şi respectiv de la x la v şi să presupunem că aceste lanţuri au un ultim vârf comun in w (in sensul deplasării de la x către u, respectiv v). Deoarece orice sublanţ al unui lanţ minim este de asemenea minim, rezultă că notând $d(x, w) = r$, avem $d(w, u) = 2p - r$, $d(w, v) = 2q - r$. Rezultă că sublanţul de la w la u impreună cu sublanţul de la w la v şi cu muchia uv induc un ciclu elementar impar, de lungime $2p + 2q - 2r + 1$, ceea ce contrazice ipoteza că graful G nu are cicluri impare. O concluzie identică se deduce când $u, v \in B$.

Dacă graful G nu este conex, el are m componente conexe, $m \geq 2$. Vom aplica construcţia anterioară fiecărei componente care nu are cicluri impare, deci este bipartită, găsind părţile A_1 şi $B_1, \ldots, A_m$ şi B_m astfel incât orice muchie din componenta i are o extremitate in A_i şi cealaltă extremitate in B_i. Deoarece oricare două componente conexe nu au vârfuri comune, rezultă că notând $A = \bigcup_{i=1}^{m} A_i$ şi $B = \bigcup_{i=1}^{m} B_i$, orice muchie a grafului G are o extremitate in A iar cealaltă in B, ceea ce incheie demonstraţia. □

Două grafuri G şi H se numesc izomorfe dacă există o bijecţie $f : V(G) \to V(H)$ astfel incât: a) dacă $xy \in E(G)$ atunci $f(x)f(y) \in E(H)$; b) dacă $xy \notin E(G)$ atunci $f(x)f(y) \notin E(H)$. Cu alte cuvinte, grafurile G şi H diferă eventual printr-o renumerotare a vârfurilor. Deoarece f este o bijecţie intre

mulţimile de vârfuri ale celor două grafuri, rezultă că oricare două grafuri izomorfe au acelaşi număr de vârfuri. De asemenea, grafurile izomorfe au acelaşi număr de muchii, aceleaşi şiruri ale gradelor vârfurilor, acelaşi număr de cicluri de o anumită lungime etc. Noţiunea de izomorfism al grafurilor explică fenomenul de izomerism din chimie. De exemplu, moleculele de hidrocarburi constau din atomi de carbon de valenţă 4 şi de hidrogen de valenţă 1. Astfel moleculele de butan şi de 2-metilpropan (izobutan) conţin fiecare patru atomi de carbon şi zece atomi de hidrogen, având formula C_4H_{10}. Grafurile asociate au fiecare câte 14 vârfuri, dar nu sunt izomorfe. Astel de structuri neizomorfe care au aceeaşi formulă chimică (in cazul de faţă C_4H_{10}) se numesc *izomeri*. Recunoaşterea izomorfismului a două grafuri este importantă in chimie, deoarece o substanţă nou descoperită trebuie să aibă graful neizomorf cu graful substanţelor deja cunoscute.

Există situaţii când modelul de graf neorientat nu este suficient pentru a reprezenta o situaţie dată, fiind necesar să definim o anumită orientare a muchiilor, care devin *arce*. Un *graf orientat* G este o pereche $(V(G), E(G))$, unde $V(G)$ este o mulţime finită şi nevidă de elemente numite vârfuri şi $E(G)$ este o mulţime de perechi ordonate de elemente distincte din $V(G)$, numite arce. $V(G)$ se numeşte *mulţimea vârfurilor* şi $E(G)$ *mulţimea arcelor* grafului G. Deci in cazul grafurilor orientate mulţimea arcelor este o submulţime a produsului cartezian $V(G)\times V(G)$ care conţine numai perechi (x, y) de vârfuri distincte. Se spune că x este *extremitatea iniţială* şi y este *extremitatea finală* a arcului (x, y). Ca şi in cazul grafurilor neorientate, un graf orientat poate fi desenat in plan reprezentând vârfurile prin puncte şi arcele prin săgeţi orientate de la extremitatea iniţială către extremitatea finală a oricărui arc. Două vârfuri unite printr-un arc se numesc *adiacente*. Spre deosebire de cazul neorientat, orice vârf x are două grade: *gradul exterior*, notat $d^+(x)$, este numărul arcelor care il au pe x extremitate iniţială, deci care pleacă din vârful x. *Gradul interior*, notat $d^-(x)$, este numărul arcelor care il au pe x extremitate finală, deci care intră in vârful x. Următorul rezultat este analogul propoziţiei 1 pentru grafuri orientate.

PROPOZIŢIA 2. Pentru orice graf orientat G suma gradelor de intrare este egală cu suma gradelor de ieşire ale vârfurilor lui G, care este egală cu numărul de arce ale grafului G.

Demonstraţie. Obţinem $\sum_{x\in V(G)} d^+(x) = \sum_{y\in V(G)} d^-(y) = |E(G)|$ deoarece fiecare arc (x, y) al grafului G contribuie cu o unitate atât la $d^+(x)$ cât şi la $d^-(y)$. □

Noţiunile de *graf parţial* şi de *subgraf* al unui graf orientat se definesc la fel ca şi in cazul neorientat, inlocuind peste tot cuvântul muchie prin cuvântul arc.

Un *lanţ* intr-un graf orientat G este o succesiune finită de vârfuri ale lui

G care se notează $[x_0, \ldots, x_r]$, cu proprietatea că oricare două vârfuri vecine sunt adiacente, adică $(x_i, x_{i+1}) \in E(G)$ sau $(x_{i+1}, x_i) \in E(G)$ pentru orice $0 \leq i \leq r-1$. Ca şi in cazul neorientat, vârfurile x_0 şi x_r se numesc *extremitatea iniţială*, respectiv *extremitatea finală* ale lanţului, iar numărul de arce r este *lungimea* lanţului. Un *drum* intr-un graf orientat G este un caz particular de lanţ, care se obţine când toate arcele au o aceeaşi orientare, de la extremitatea iniţială către extremitatea finală. Deci un drum este o succesiune finită de vârfuri ale lui G care se notează $(x_0, \ldots, x_r)$, cu proprietatea că $(x_i, x_{i+1}) \in E(G)$ pentru orice $0 \leq i \leq r-1$.
Un lanţ inchis pentru care arcele nu se repetă se numeşte *ciclu*, iar un drum inchis pentru care arcele nu se repetă se numeşte *circuit*. Deci un ciclu intr-un graf orientat G este o succesiune finită de vârfuri ale lui G, notată $[x_0, \ldots, x_r]$ cu următoarele proprietăţi: a) $(x_i, x_{i+1}) \in E(G)$ sau $(x_{i+1}, x_i) \in E(G)$ pentru orice $0 \leq i \leq r-1$; b) $x_0 = x_r$; c) toate arcele lanţului sunt distincte. Un circuit este o succesiune finită de vârfuri ale lui G, notată $(x_0, \ldots, x_r)$ cu următoarele proprietăţi: a) $(x_i, x_{i+1}) \in E(G)$ pentru orice $0 \leq i \leq r-1$; b) $x_0 = x_r$; c) toate arcele $(x_0, x_1), (x_1, x_2), \ldots, (x_{r-1}, x_r)$ sunt distincte. Numărul arcelor unui ciclu, respectiv circuit, se numeşte *lungimea* sa. Dacă toate vârfurile $x_0, \ldots, x_{r-1}$ sunt distincte, ciclul, respectiv circuitul se numeşte *elementar*. Un graf orientat G cu proprietatea că pentru orice arc $(x, y) \in E(G)$ arcul $(y, x) \in E(G)$ se numeşte *simetric*. Astfel putem considera grafurile neorientate cazuri particulare de grafuri orientate şi anume grafuri orientate simetrice. Ca şi in cazul neorientat, un graf orientat se numeşte *conex* dacă intre oricare două vârfuri distincte există un lanţ care le uneşte. In cazul grafurilor orientate mai putem defini şi o altă noţiune şi anume conexitatea tare: un graf orientat G se numeşte *tare conex* dacă oricare două vârfuri distincte ale lui G sunt unite printr-un drum. Deoarece putem schimba rolurile celor două vârfuri, uneori definiţia conexităţii tare se mai formulează astfel: un graf orientat G se numeşte tare conex dacă oricare ar fi vârfurile $x, y \in V(G)$, $x \neq y$, există un drum de la x la y şi un drum de la y la x.

MATRICI ASOCIATE UNUI GRAF ORIENTAT

Un graf orientat G cu n vârfuri poate fi caracterizat de o matrice pătrată cu n linii şi n coloane, cu elemente 0 şi 1 şi numită *matricea de adiacenţă* a grafului. Dacă vârfurile grafului sunt $x_1, \ldots, x_n$, matricea de adiacenţă, notată $A = (a_{i,j})_{i,j=1,\ldots,n}$ se defineşte astfel: $a_{i,j} = 1$ dacă există arcul $(x_i, x_j) \in E(G)$ şi $a_{i,j} = 0$ in caz contrar. Dacă graful G este neorientat, matricea sa de adiacenţă are proprietatea că $a_{i,j} = 1$ dacă există muchia $x_i x_j$ intre vârfurile x_i şi x_j şi $a_{i,j} = 0$ in caz contrar. In acest caz matricea de adiacenţă este simetrică, adică $a_{i,j} = a_{j,i}$ pentru orice $i, j = 1, \ldots, n$. O altă matrice utilă in studiul conexităţii unui graf este matricea drumurilor (sau

matricea inchiderii tranzitive a relaţiei binare asociate grafului orientat), care se notează $A^* = (a^*_{i,j})_{i,j=1,\ldots,n}$ şi care se defineşte astfel: $a^*_{i,j} = 1$ dacă există in graful G un drum care pleacă din vârful x_i şi ajunge in vârful x_j şi $a^*_{i,j} = 0$ in caz contrar. Vom avea $a^*_{i,i} = 1$ dacă există un circuit care trece prin x_i şi $a^*_{i,i} = 0$ in caz contrar.

O altă matrice care caracterizează un graf orientat şi este utilă in studiul circuitelor electrice este matricea de incidenţă nod-arc. Dacă graful orientat G are n vârfuri $x_1, \ldots, x_n$ şi m arce $e_1, \ldots, e_m$, matricea de incidenţă nod-arc este o matrice dreptunghiulară cu n linii şi m coloane, pe care o notăm $B = (b_{i,j})_{\substack{i=1,\ldots,n \\ j=1,\ldots,m}}$ şi o definim astfel: $b_{i,j} = 1$ sau -1 după cum arcul e_j pleacă din vârful x_i sau intră in vârful x_i şi $b_{i,j} = 0$ in caz contrar, deci dacă x_i nu este extremitate a arcului e_j. Să observăm că fiecare coloană a matricii de incidenţă B conţine un 1, un -1 şi $n-2$ zerouri. Fixând un sens al curentului pe fiecare latură a unei reţele electrice, care devine astfel un graf orientat şi notând cu $\mathbf{i} = (i_1, \ldots, i_m)$ vectorul intensităţilor curentului pe arcele reţelei electrice, prima lege a lui Kirchhoff se scrie sub forma: $B\mathbf{i}^t = \mathbf{0}^t$, unde t reprezintă transpusul unui vector, iar $\mathbf{0}$ reprezintă vectorul linie cu n componente nule.

In cazul grafurilor neorientate, matricea de incidenţă nod-muchie va conţine doi de 1 pe fiecare coloană. Suma oricăror r linii ale matricii B pentru un graf G conex cu n vârfuri conţine cel puţin o componentă nenulă pentru $r < n$. Intr-adevăr, in caz contrar ar rezulta că din mulţimea de vârfuri corespunzătoare celor r linii nu mai pleacă nici un arc in exterior şi nu mai soseşte nici un arc din exterior, ceea ce contrazice faptul că G este conex. Deci oricare r linii ($r < n$) ale matricii B sunt linear independente. Dar suma celor n linii ale matricii B este vectorul nul, deci ele sunt linear dependente. Am obţinut următorul rezultat:

PROPOZIŢIA 3 (Kirchhoff). Dacă graful G are n vârfuri şi este conex, rangul matricii sale de incidenţă nod-arc este $n-1$.

Aplicând acest rezultat submatricilor corespunzătoare componentelor conexe ale lui G, se obţine că rangul matricii B este $n-p$ dacă graful are p componente conexe.

Algoritmul lui Roy-Warshall

Acest algoritm serveşte la obţinerea matricii drumurilor A^* plecând de la matricea de adiacenţă A a unui graf G cu n vârfuri şi constă in următoarele:

1. Se face $k = 1$.

2. Pentru $i = 1, \ldots, n$ şi $j = 1, \ldots, n$ şi $i, j \neq k$ se inlocuiesc elementele $a_{i,j} = 0$ prin $\min(a_{i,k}, a_{k,j})$.

3. Se repetă pasul 2 pentru $k = 2, \ldots, n$.

Vom arăta că la sfârşitul aplicării algoritmului matricea obţinută este tocmai matricea drumurilor A^*. Acest algoritm poate fi exprimat şi in limbaj

operatorial, definind operatorul T_k ce acţionează asupra unei matrici pătrate de ordinul n cu 0 şi 1 in felul următor: $T_k(A) = B$, unde $B = (b_{i,j})_{i,j=1,\ldots,n}$ este o matrice pătrată de ordinul n, definită astfel:

$$b_{i,j} = \max(a_{i,j}, \min(a_{i,k}, a_{k,j})) = a_{i,j} \vee a_{i,k}a_{k,j}$$

pentru $i, j = 1, \ldots, n$. Pentru simplificarea scrierii am notat operaţia maximum prin $\vee$, iar operaţia de minimum este tocmai operaţia produs in algebra booleană $\{0, 1\}$. Din cauza proprietăţii de absorbţie mai rezultă că $b_{i,j} = a_{i,j}$ dacă i sau j este egal cu k. La pasul 2 al algoritmului se calculează tocmai matricea $T_k(A)$, deoarece elementele $a_{i,j} = 1$ rămân invariante din definiţia operatorilor T_k.

PROPOZIŢIA 4. Transformările T_k sunt idempotente, iar oricare două transformări T_k, T_h comută intre ele.

Demonstraţie. Idempotenţa operatorilor T_k, adică $T_k^2 = T_k$ este imediată, deoarece $(T_k^2(A))_{i,j} = a_{i,j} \vee a_{i,k}a_{k,j} \vee a_{i,k}a_{k,j} = a_{i,j} \vee a_{i,k}a_{k,j} = (T_k(A))_{i,j}$ pentru orice $i, j = 1, \ldots, n$. Pentru a arăta că $T_hT_k = T_kT_h$ vom calcula $(T_hT_k(A))_{i,j} = b_{i,j} \vee b_{i,h}b_{h,j}$, unde $B = T_k(A)$. Rezultă că $(T_hT_k(A))_{i,j} = a_{i,j} \vee a_{i,k}a_{k,j} \vee (a_{i,h} \vee a_{i,k}a_{k,h})(a_{h,j} \vee a_{h,k}a_{k,j})$. Aplicând proprietatea de distributivitate, această ultimă expresie este egală cu $a_{i,j} \vee a_{i,k}a_{k,j} \vee a_{i,h}a_{h,j} \vee a_{i,h}a_{h,k}a_{k,j} \vee a_{i,k}a_{k,h}a_{h,j} \vee a_{i,k}a_{k,h}a_{h,k}a_{k,j}$. Deoarece ultimul termen este absorbit de al doilea termen, expresia obţinută este simetrică in h şi k, ceea ce demonstrează că $T_hT_k(A) = T_kT_h(A)$. □

TEOREMA 2. Există egalitatea

$$\prod_{k=1}^{n} T_k(A) = A^*$$

pentru orice matrice de adiacenţă A de ordinul n.

Demonstraţie. Vom arăta mai intâi că dacă o matrice A de ordinul n cu 0 şi 1 este invariată de orice transformare $T_k(k = 1, \ldots, n)$, atunci $A = A^*$. Intr-adevăr, să presupunem prin reducere la absurd că $A \neq A^*$. Aceasta inseamnă că există doi indici, fie 1 şi p, astfel incât $a_{1,p} = 0$ dar $a^*_{1,p} = 1$. Deci in graful orientat care are pe A matrice de adiacenţă există un drum, fie $(x_1, x_2, \ldots, x_p)$ astfel incât $(x_i, x_{i+1}) \in E(G)$ pentru $i = 1, \ldots, p-1$ dar $(x_1, x_p) \notin E(G)$. Să notăm cu k prima valoare din şirul $2, 3, \ldots, p$ cu proprietatea că $(x_1, x_k) \notin E(G)$. In acest caz transformarea T_{k-1} nu poate lăsa invariantă matricea A, deoarece $a_{1,k-1} = 1$, $a_{k-1,k} = 1$, deci prin aplicarea operatorului T_{k-1} elementul 0 situat in linia 1 şi coloana k va fi inlocuit cu 1, ceea ce contrazice ipoteza, deci $A = A^*$. Acum să notăm $\prod_{k=1}^{n} T_k(A) = B$. Este clar că $(T_k(A))^* = A^*$ pentru orice $k = 1, \ldots, n$, deoarece operatorul T_k nu face decât să introducă un arc intre toate perechile de vârfuri x_i şi x_j din graful asociat care nu erau adiacente, dar care erau legate printr-un drum

de lungime egală cu 2 de la x_i la x_j, care trecea prin x_k, deci $T_k(A)$ şi A au aceeaşi matrice a drumurilor. Prin aplicarea repetată a acestei proprietăţi se obţine că $B^* = A^*$. Insă conform propoziţiei precedente matricea B este invariantă de toţi operatorii T_k pentru $k = 1, \ldots, n$, deci $B = B^*$ conform celor demonstrate anterior. Rezultă că $B = A^*$ şi algoritmul Roy-Warshall este justificat. □

Să observăm că transformarea T_k constă in inlocuirea tuturor elementelor 0 care nu se găsesc pe linia sau coloana k prin 1, dacă ambele lor proiecţii pe linia şi coloana k sunt egale cu 1. Deoarece sunt $(n-1)^2$ elemente care nu se găsesc pe linia şi coloana k, rezultă că atât numărul de operaţii minimum cât şi numărul de operaţii maximum necesitate de aplicarea algoritmului este egal fiecare cu $n(n-1)^2$, care este complexitatea acestui algoritm.

Să notăm cu $S(x_i)$ mulţimea vârfurilor care sunt extremităţi terminale ale unor drumuri care pleacă din x_i şi cu $P(x_i)$ mulţimea vârfurilor care sunt extremităţi iniţiale ale unor drumuri care se termină in x_i. Componenta tare conexă care conţine vârful x_i este tocmai mulţimea $S(x_i) \cap P(x_i) \cup \{x_i\}$. Dar mulţimile $S(x_i)$, respectiv $P(x_i)$ se obţin imediat din matricea drumurilor A^*: $S(x_i)$ este compusă din acele vârfuri x_j pentru care indicele j verifică $a^*_{i,j} = 1$, deci corespund coloanelor care conţin un 1 in linia i, iar $P(x_i)$ este compusă din acele vârfuri x_j pentru care $a^*_{j,i} = 1$, deci care corespund liniilor care conţin un 1 in coloana i. De aici rezultă un algoritm pentru găsirea componentelor tare conexe ale unui graf orientat G plecând de la matricea drumurilor A^*, tinând seama că aceste componente tare conexe constituie o partiţie a mulţimii de vârfuri $V(G)$.

ARBORI ŞI ALGORITMUL LUI KRUSKAL

Un graf conex şi fără cicluri se numeşte *arbore*. Terminologia ţine seama de asemănarea cu arborii din natură. Am văzut că vârfurile de gradul 1 ale unui graf se numesc *vârfuri terminale*, care in cazul unui arbore se mai numesc şi *frunze*. Astfel unei expresii aritmetice i se asociază in mod natural un arbore, in vârfurile căruia se reprezintă atât operatorii cât şi operanzii. Frunzele au asociaţi operanzi, iar celelalte vârfuri conţin operatori. Deoarece unele operaţii (scăderea, ridicarea la putere, impărţirea) nu sunt comutative, avem de a face cu un tip special de arbori, numiţi *arbori binari*. Astfel vârful in care este reprezentată ultima operaţie se numeşte *rădăcina* arborelui binar. Rădăcina are un *subarbore stâng* şi un *subarbore drept*. Subarborii stâng şi drept ai rădăcinii sunt, la rândul lor, arbori binari. In cazul arborilor binari se face distincţie intre stânga şi dreapta. Aceşti arbori sunt utilizaţi in multe capitole ale informaticii. Se observă din desenul unui arbore că, oricum am suprima o muchie a unui arbore se obţine un graf neconex, cu două componente conexe. De asemenea, oricum am uni printr-o nouă muchie două vârfuri neadiacente ale unui arbore, in graful astfel obţinut apare un

ciclu. Aceste proprietăţi au loc pentru orice arbore.

TEOREMA 3. Următoarele afirmaţii sunt echivalente pentru un graf G:

a) G este conex şi fără cicluri;

b) G este un graf conex minimal, adică G este conex, dar devine neconex prin suprimarea unei muchii oarecare;

c) G este un graf fără cicluri maximal, adică G nu conţine cicluri, dar prin unirea printr-o muchie a oricăror două vârfuri neadiacente ale lui G, graful obţinut conţine cicluri.

Demonstraţie. Vom arăta că implicaţiile a) $\rightarrow$ b) $\rightarrow$ c) $\rightarrow$ a) sunt adevărate.

a) $\rightarrow$ b): Să presupunem că G este conex şi fără cicluri şi că graful $G-xy$ obţinut din G prin suprimarea unei muchii xy este conex. Rezultă că graful $G-xy$ conţine un lanţ de extremităţi x şi y, deci şi un lanţ elementar L de extremităţi x şi y. Dar lanţul L, impreună cu muchia xy, formează un ciclu elementar in graful G, ceea ce contrazice a). Deci are loc b).

b) $\rightarrow$ c): Fie G un graf conex minimal. Dacă presupunem că G are un ciclu C, fie xy o muchie a ciclului. Dacă suprimăm muchia xy din graful G, din ciclul C rămâne un lanţ L care uneşte extremităţile muchiei xy. Deci orice lanţ care uneşte două vârfuri distincte ale lui G se transformă intr-un lanţ de aceleaşi extremităţi in graful $G-xy$ inlocuind muchia xy prin lanţul L. Rezultă că $G-xy$ este un graf conex, ceea ce contrazice ipoteza că G este conex minimal. In consecinţă, G nu conţine cicluri. Fie acum x, y două vârfuri neadiacente ale lui G. Deoarece G este conex, există un lanţ elementar L de extremităţi x şi y. Deci $G+xy$, graful obţinut din G prin adăugarea muchiei xy conţine un ciclu elementar obţinut din L şi muchia xy. Rezultă că G este un graf fără cicluri maximal.

c) $\rightarrow$ a): Fie G un graf fără cicluri maximal. Trebuie să arătăm că G este conex. Presupunând că G nu este conex, rezultă existenţa a două vârfuri x şi y care aparţin unor componente conexe diferite ale lui G. Conform ipotezei, $G+xy$ are un ciclu C care conţine muchia xy, deoarece graful G nu are cicluri. Eliminând muchia xy din ciclul C, rezultă un lanţ L care uneşte vârfurile x şi y in graful G. Dar aceasta contrazice presupunerea că x şi y aparţin unor componente conexe diferite in G. Deci G trebuie să fie conex, iar proprietatea de a nu avea cicluri este conţinută in ipoteză. □

Am văzut că un graf parţial H al grafului G, este graful G sau se obţine din G prin suprimarea anumitor muchii. Dacă graful parţial H este arbore, el se numeşte *arbore parţial* al lui G.

COROLARUL 1. Un graf G are un arbore parţial dacă şi numai dacă G este conex.

Demonstraţie. Necesitatea este imediată, deoarece orice arbore este un graf conex. Să arătăm acum că orice graf conex G conţine un arbore parţial

H. Dacă G este un graf conex minimal, din teorema precedentă rezultă că G este arbore şi definim $H = G$. In caz contrar, există o muchie $xy \in E(G)$ astfel incât graful $G_1 = G - xy$ este conex. Dacă G_1 este un graf conex minimal, definim $H = G_1$. In caz contrar vom repeta pentru G_1 procedeul de suprimare a unei muchii aplicat lui G, până la obţinerea unui graf conex minimal, care va fi un arbore parţial al lui G. Procedeul descris are un număr finit de paşi, deoarece graful G de la care am plecat are un număr finit de muchii, iar la fiecare pas se suprimă câte o muchie a grafului parţial obţinut in acel moment. □

PROPOZIŢIA 5. Orice arbore cu $n \geq 2$ vârfuri are cel puţin două vârfuri terminale.

Demonstraţie. Să presupunem, prin reducere la absurd, că există un arbore A cu $n \geq 2$ vârfuri care are cel mult un vârf terminal. Fie $L = [x, z_1, \ldots, z_k, y]$ un lanţ elementar de lungime maximă al lui A. Cel puţin una dintre extremităţile lui L, fie aceasta y, are gradul $d(y) \geq 2$. Deoarece y este adiacent cu z_k, rezultă că y mai este adiacent cu cel puţin un vârf al lui A. Deoarece lanţul L are o lungime maximă, rezultă că y este adiacent cu unul dintre vârfurile $x, z_1, \ldots, z_{k-1}$, ceea ce produce un ciclu in A. Deoarece A este arbore am ajuns la o contradicţie şi proprietatea este demonstrată. □

Arborii au o proprietate remarcabilă, dată de teorema următoare.

TEOREMA 4. Orice arbore cu n vârfuri are $n - 1$ muchii.

Demonstraţie. Demonstraţia se face prin inducţie după n. Pentru $n = 1$ există un singur arbore cu un vârf şi fără nici o muchie, deci proprietatea se verifică. Fie $n \geq 1$, să presupunem că proprietatea este adevărată pentru orice arbore cu n vârfuri şi fie A un arbore cu $n + 1$ vârfuri. Conform propoziţiei precedente, arborele A are cel puţin două vârfuri de gradul 1. Fie x unul dintre acestea şi A_1 subgraful obţinut din A prin suprimarea vârfului x şi a unicei muchii care are o extremitate in x. Deoarece A nu conţine cicluri rezultă că nici A_1 nu conţine cicluri. Se observă uşor că A_1 este şi conex, deci este un arbore cu n vârfuri. Aplicând ipoteza de inducţie pentru A_1 se deduce că A_1 are $n - 1$ muchii, deci A are n muchii şi proprietatea este demonstrată. □

Proprietatea unui arbore de a fi un graf conex minimal face ca arborii să apară intr-o serie de probleme de optimizare, cum este următoarea:

Problema arborelui parţial minim

Fie G un graf conex şi c o funcţie $c : E(G) \to (0, \infty)$ care asociază fiecărei muchii a grafului G un număr real pozitiv numit costul acelei muchii. Costul unui graf parţial al lui G este egal prin definiţie cu suma costurilor asociate muchiilor lui H, ceea ce vom scrie: $c(H) = \sum_{u \in E(H)} c(u)$. Se pune problema determinării unui graf parţial H al lui G care trebuie să fie conex şi să aibă un cost minim. Un astfel de graf parţial conex de cost minim trebuie să fie

un arbore parţial, deoarece arborii sunt singurele grafuri conexe minimale. Deci orice graf parţial de cost minim care este conex este un arbore parţial al lui G. Acest arbore parţial există conform corolarului 1. Pentru găsirea unui arbore parţial de cost minim al unui graf conex, pe care il vom numi, pe scurt, *arbore parţial minim*, descriem in continuare algoritmul lui Kruskal.

Algoritmul lui Kruskal

Fie G un graf conex cu $n \geq 2$ vârfuri şi o funcţie cost $c : E(G) \to (0, \infty)$. Muchiile unui arbore parţial minim sunt selectate una câte una, după cum urmează:

Pasul 1. Dintre muchiile nealese ale lui $E(G)$ se selectează o muchie de cost minim cu condiţia să nu formeze cicluri cu muchiile deja alese.

Pasul 2. Au fost selectate $n-1$ muchii? Dacă da, ne oprim. Muchiile selectate sunt muchiile unui arbore parţial minim al lui G. Dacă nu, se repetă pasul 1.

TEOREMA 5. Algoritmul lui Kruskal produce un arbore parţial de cost minim.

Demonstraţie. Să arătăm mai intâi că cele $n-1$ muchii selectate sunt muchiile unui arbore parţial al lui G. Pentru aceasta trebuie să demonstrăm că orice graf H cu n vârfuri, $n-1$ muchii şi fără cicluri este un arbore, deci mai trebuie să arătăm că H este conex. Să presupunem, prin reducere la absurd, că H nu este conex şi are $p \geq 2$ componente conexe. Atunci fiecare componentă conexă este un arbore. Dacă notăm cu n_i $(1 \leq i \leq p)$ numărul de vârfuri din componenta conexă numărul i a lui H, numărul de muchii din această componentă este egal cu $n_i - 1$. Deci numărul total de muchii ale lui H este egal cu $\sum_{i=1}^{p}(n_i - 1) = n - p$. Deoarece $p \geq 2$ obţinem $n - p < n - 1$, ceea ce contrazice ipoteza. Rezultă că $p = 1$, deci H este conex şi fără cicluri, deci este un arbore.

Să arătăm că arborele parţial A obţinut cu algoritmul lui Kruskal este un arbore parţial minim al grafului G. Să presupunem că muchiile lui A sunt $e_1, \ldots, e_{n-1}$, fiind obţinute in această ordine cu algoritmul lui Kruskal. Să presupunem că A nu este de cost minim, deci există arbori parţiali ai lui G de cost strict mai mic decât A. In mulţimea acestor arbori parţiali să alegem unul pe care il notăm B, care are un număr maxim de muchii in comun cu A. Deci $c(B) < c(A)$ şi din această cauză mulţimile de muchii ale arborilor A şi B sunt diferite. Fie e_k prima muchie din şirul ordonat $e_1, \ldots, e_{n-1}$ care nu este muchie a lui B. Adăugând muchia e_k la arborele B apare un ciclu C care conţine această muchie. Ciclul C mai conţine in afara muchiei e_k, cel puţin alte două muchii ale lui B. Dintre aceste muchii, există cel puţin o muchie e care este muchie a lui B dar nu este muchie a lui A, deoarece in caz contrar arborele A ar conţine ciclul C. Dacă suprimăm muchia e din arborele B la care am adăugat muchia e_k, obţinem un nou arbore parţial B'. Intr-adevăr,

graful B' este conex, deoarece intre extremităţile muchiei suprimate e există un lanţ indus de ciclul C. B' are $n-1$ muchii. Considerând un arbore parţial al lui B' acesta are $n-1$ muchii, deci coincide cu B'. Deci B' este un arbore parţial al lui G. Costul lui este egal cu

$$c(B') = c(B) + c(e_k) - c(e)$$

Insă din descrierea algoritmului lui Kruskal, e_k este o muchie de cost minim care nu formează cicluri cu muchiile deja alese $e_1, \ldots, e_{k-1}$. Insă nici e nu formează cicluri cu aceste muchii, care sunt muchii şi ale arborelui B, iar $e \in E(B)$. Rezultă $c(e_k) \leq c(e)$, deci $c(B') \leq c(B) < c(A)$. Am ajuns la o contradicţie, deoarece arborele parţial B' are costul strict mai mic ca A şi are mai multe muchii in comun cu A decât B, ceea ce contrazice ipoteza făcută despre B. Deci A este un arbore parţial de cost minim al grafului G. □

Se observă că la pasul curent al aplicării algoritmului muchiile selectate, in ordinea crescătoare a costurilor, formează un număr de componente conexe care toate sunt arbori. Aceştia se unesc in final pentru a forma un singur arbore, care este un arbore parţial de cost minim. Dacă atribuim o aceeaşi etichetă (de exemplu un număr natural cuprins intre 1 şi n) tuturor vârfurilor dintr-o aceeaşi componentă, având grijă ca componentele diferite să aibă etichete diferite, selectarea unei noi muchii se poate face acum uşor, verificând numai ca etichetele extremităţilor ei să fie diferite; apoi cele două componente in care se găsesc extremităţile ei se unifică intr-o singură componentă, deci toate etichetele celor două componente trebuie unificate (de exemplu cu eticheta cea mai mică). Folosind un vector cu n componente V se iniţializează $V(i) = i$ pentru $i = 1, \ldots, n$, apoi se aplică algoritmul descris de unificare a etichetelor la alegerea unei noi muchii pq astfel incât $V(p) \neq V(q)$. Dacă G are n vârfuri şi m muchii, sortarea prealabilă a muchiilor in ordinea crescătoare a costurilor necesită o complexitate de calcul de ordinul $O(m\log m)$, iar parcurgerea de $n-1$ ori a vectorului V al etichetelor şi unificarea etichetelor vârfurilor din componentele care se unesc la fiecare pas mai necesită o complexitate de $O(n^2)$. Rezultă in total o complexitate timp de ordinul $O(m\log m + n^2)$, al doilea termen fiind absorbit de primul dacă graful are multe muchii.

DISTANŢE ŞI DRUMURI MINIME IN GRAFURI ORIENTATE

Pentru un graf orientat G să presupunem că pe mulţimea arcelor sale $E(G)$ s-a definit o *funcţie distanţă* $d : E(G) \to (0, \infty)$ care asociază fiecărui arc al grafului G lungimea sa $d(i,j) > 0$, care va fi notată in continuare cu $d_{i,j}$. In cazul unui astfel de graf ponderat vom defini *lungimea* unui drum D ca fiind egală cu suma lungimilor arcelor drumului D. Să observăm că in cazul unui graf neponderat lungimea unui drum (respectiv lanţ in cazul neorientat) s-a definit analog dacă se asociază fiecărui arc (respectiv muchii)

o lungime egală cu 1. Pentru două vârfuri $x, y \in V(G)$ pot exista in graful G mai multe drumuri care pleacă din x şi sosesc in y. Vom defini *distanţa minimă* de la vârful x la vârful y in graful G ($x \neq y$) şi o vom nota cu $d^*(x, y)$, minimul lungimilor drumurilor din graful G care au extremitatea iniţială in x şi extremitatea finală in y. Dacă nu există nici un drum de la x la y in G vom defini $d^*(x, y) = \infty$.

Lungimile $d_{i,j}$ ale arcelor grafului se mai numesc şi *distanţele directe* dintre vârfurile grafului şi formează *matricea distanţelor directe* care se defineşte in felul următor pentru un graf G cu n vârfuri notate $1, \ldots, n$: $D = (d_{i,j})_{i,j=1,\ldots,n}$, unde $d_{i,i} = 0$ pentru orice $i = 1, \ldots, n$; $d_{i,j}$ reprezintă lungimea arcului (i, j) dacă $(i, j) \in E(G)$ şi $d_{i,j} = \infty$ dacă $(i, j) \notin E(G)$. *Matricea distanţelor minime*, notată $D^* = (d^*_{i,j})_{i,j=1,\ldots,n}$ se defineşte in modul următor: $d^*_{i,i} = 0$ pentru orice $i = 1, \ldots, n$; $d^*_{i,j}$ reprezintă distanţa minimă de la vârful i la vârful j dacă există cel puţin un drum de la i la j şi $d^*_{i,j} = \infty$ dacă nu există nici un drum de la i la j in graful G. Algoritmul lui Floyd permite calcularea matricii D^* plecând de la matricea D a distanţelor directe şi este analog algoritmului lui Roy-Warshall pentru obţinerea matricii drumurilor plecând de la matricea de adiacenţă a unui graf. Se schimbă numai operaţiile de maximum şi minimum prin minimum şi respectiv +, definite pe mulţimea numerelor reale pozitive, iar elementele de pe diagonala principală nu se mai calculează, ele rămânând egale cu 0.

Algoritmul lui Floyd are următorii paşi:

1. Se face $k = 1$.

2. Pentru $i = 1, \ldots, n$ şi $j = 1, \ldots, n$, $i \neq j$, $i \neq k$, $j \neq k$, se inlocuieşte elementul $d_{i,j}$ prin $\min(d_{i,j}, d_{i,k} + d_{k,j})$.

3. Se repetă pasul 2 pentru $k = 2, \ldots, n$.

La sfârşitul aplicării algoritmului, in locul matricii D apare matricea D^*. La fel ca in cazul găsirii matricii drumurilor, acest algoritm se poate exprima in limbaj operatorial şi se poate justifica in mod asemănător.

Deoarece elementele de pe diagonala principală a matricii D rămân neschimbate, numărul de adunări cât şi numărul de comparaţii pentru algoritmul lui Floyd este egal cu $n(n-1)(n-2)$.

In continuare vom prezenta un algoritm pentru determinarea atât a distanţelor cât şi a drumurilor minime de la un vârf fixat s al unui graf orientat la toate celelalte vârfuri, având complexitatea timp de ordinul $O(n^2)$ dacă graful are n vârfuri. Pentru a descrie modul in care se obţin drumurile minime, vom defini mai intâi noţiunea de *arborescenţă*. O arborescenţă se obţine dintr-un arbore A in modul următor: se alege un vârf r al arborelui A şi se orientează toate muchiile arborelui (care astfel devin arce) in mod unic astfel incât pentru orice vârf $x \neq r$ al arborelui să existe un drum de la r la x. Vârful r se numeşte *rădăcina* arborescenţei. Algoritmul care va fi descris

calculează distanţele şi drumurile minime de la un vârf s la toate celelalte vârfuri ale unui graf orientat G cu lungimile arcelor numere nenegative. La sfârşitul aplicării algoritmului se obţine o arborescenţă A cu rădăcina s şi mulţimea vârfurilor $V(G)$ cu proprietatea că pentru orice vârf $i \neq s$ drumul unic de la s la i in A este un drum de lungime minimă in G. Arborescenţa A este definită folosind funcţia predecesor, notată *pred*, astfel incât dacă $(i, j) \in E(A)$ atunci $pred(j) = i$. Algoritmul etichetează fiecare vârf i al grafului cu o etichetă notată $l(i)$ care este un majorant al distanţei minime de la vârful s la vârful i, iar in final va fi egală chiar cu distanţa minimă de la s la i. La fiecare pas intermediar algoritmul partiţionează vârfurile in două mulţimi: mulţimea S a vârfurilor cu etichete permanente şi complementara sa $\overline{S}$, a vârfurilor cu etichete temporare. Etichetele permanente ale vârfurilor din S reprezintă distanţele minime de la s la vârfurile respective, in timp ce etichetele temporare ale vârfurilor din $\overline{S}$ reprezintă nişte majorări ale distanţelor minime. La fiecare pas un vârf din $\overline{S}$ cu cea mai mică etichetă temporară este trecut din $\overline{S}$ in S. La sfârşitul aplicării algoritmului $\overline{S} = \emptyset$, iar $l(i)$ va reprezenta pentru orice vârf i distanţa minimă de la s la i.

Algoritmul lui Dijkstra

Se consideră un graf orientat cu n vârfuri notate $1, \ldots, n$ pentru care lungimea fiecărui arc (i, j) se notează cu $d_{i,j} \geq 0$. Se fixează un vârf s din G. La sfârşitul aplicării algoritmului $l(i)$ va fi distanţa minimă de la vârful s la vârful i pentru orice $1 \leq i \leq n$. Prin definiţie, $l(s) = 0$. Dacă la sfârşitul aplicării algoritmului pentru un vârf k se obţine $l(k) = \infty$, aceasta semnifică inexistenţa drumurilor de la vârful s la vârful k in graf. In acest caz arborescenţa drumurilor minime nu va conţine vârful k. Arborescenţa A a drumurilor minime de la s la celelalte vârfuri are $n-1$ arce, care sunt perechile ordonate $(pred(i), i)$ pentru orice $i = 1, \ldots, n$ şi $i \neq s$. Paşii algoritmului sunt următorii:

1. $S \leftarrow \emptyset$, $\overline{S} \leftarrow V(G)$.
2. $l(s) \leftarrow 0$ şi $l(i) \leftarrow \infty$ pentru orice $i \neq s$.
3. Cât timp $|S| < n$ se execută următoarele operaţii:

a) se selectează un vârf $i \in \overline{S}$ pentru care

$$l(i) = \min\{l(j) | j \in \overline{S}\}$$

;

b) $S \leftarrow S \cup \{i\}$; $\overline{S} \leftarrow \overline{S} \backslash \{i\}$:

c) pentru fiecare arc (i, j) care are extremitatea iniţială in i şi extremitatea finală in $j \in \overline{S}$, dacă

$$l(j) > l(i) + d_{i,j},$$

atunci $l(j) \leftarrow l(i) + d_{i,j}$ şi $pred(j) \leftarrow i$.