

ELEMENTE DE TEORIA
GRAFURILOR

PROBLEME DE
OPTIMIZARE



MAZILESCU GEORGIAN

ELEMENTE DE TEORIA GRAFURILOR

PROBLEME DE OPTIMIZARE

Copyright © 2020
Autor: Mazilescu Georgian

Toate drepturile rezervate.

ISBN 978-606-94763-2-1

Editura Evomind, 2020

<https://evomind.org/>

CUPRINS

INTRODUCERE	4
CAPITOLUL 1 GRAFURI ORIENTATE ȘI NEORIENTATE.....	6
1.1. Noțiuni generale despre grafuri	6
1.2. Tipuri de conexiune a grafurilor	10
1.3. Matrici atașate unui graf orientat	12
1.4. Metode pentru determinarea componentelor tare conexe ale unui graf orientat	19
1.5. Aplicații	24
CAPITOLUL 2 DRUMURI MINIME-MAXIME ÎN GRAFURI	28
2.1. Transport pe drumuri minime-maxime.....	28
2.2. Algoritmi pentru determinarea drumurilor minime	33
2.2.1. Algoritmul Bellman-Kalaba.....	33
2.2.2. Algoritmul lui Yen.	36
2.2.3. Algoritmul Roy - Floyd.....	39
2.2.4. Algoritmul lui Dantzig.	41
2.3. Arbori.....	44
2.3.1. Definiții și elemente de bază	44
2.3.2. Arbore parțial de cost minim. Algoritmi	47
2.3.2.1. Algoritmul lui Kruskal	48
2.3.3. Arbori binari de căutare optimali	52
2.4. Aplicații	56
CAPITOLUL 3 DRUMURI HAMILTONIENE ÎN GRAFURI.....	61
3.1. Transport pe drumuri hamiltoniene	61
3.2. Algoritmi pentru determinarea drumurilor hamiltoniene	64
3.2.1. Algoritmul ce folosește graful condensat	64
3.2.2. Algoritmul lui Kaufmann (produs latin)	66
3.2.3. Algoritmul lui Little	72
3.2.4. Algoritm euristic bazat pe arborele economic.....	75
3.3. Aplicații	81
BIBLIOGRAFIE.....	90

INTRODUCERE

Teoria grafurilor s-a dezvoltat la început în paralel cu algebra și a avut ca punct de plecare jocurile matematice. Această formă a științei a căpătat în timp atât formă cât și conținut propriu, devenind un tot unitar bine conturat și bine fundamentat teoretic, cu largă aplicare practică.

Originile teoriei grafurilor se găsesc în rezolvarea unor probleme de jocuri și amuzamente matematice, care au atras atenția unor matematicieni de seamă, cum ar fi: Euler, Hamilton, Cayley, Sylvester, Birkoff.

„Data nașterii” teoriei grafurilor poate fi considerată anul 1736, când matematicianul elevățian **Leonhard Euler** a publicat în revista *Comentarii Academiae Scientiarum Imperialis Petropolitanae* un articol în limba latină (*Solutio problematis ad geometriam situs pertinentis – Soluția unei probleme legate de geometria propoziției*) în care a clarificat „problema celor șapte poduri”, stabilind astfel o metoda pentru rezolvarea unei întregi clase de probleme.

Numeroase situații din viața cotidiană pot fi modelate cu ajutorul teoriei grafurilor.

Numeroase probleme practice, cu aplicații în fizică, economie, chimie, sociologie etc., își găsesc soluția utilizând algoritmi din teoria grafurilor.

Cu 200 de ani mai târziu, în 1936, apărea la Leipzig prima carte de teoria grafurilor, al cărui autor este matematicianul maghiar Dénes König. În amintirea contribuției lui Euler, unele noțiuni și tipuri de grafuri de care acesta s-a ocupat sunt denumite de către König lanț (ciclu) eulerian, graf eulerian, etc.

Un alt matematician care s-a ocupat de aceleași probleme ca și Euler (se pare, fără a cunoaște articolul) dar care și-a publicat rezultatele cercetărilor sale în anul 1873, a fost Carl Hierholzer. Acesta a demonstrat în plus unele rezultate care lui Euler i se păruseră evidente.

În 1851 articolul lui Euler a fost tradus și publicat în revista *Nouvelles Annales de Mathematiques*, iar rezultatele sale au fost îmbogățite, fiind studiate în clase speciale de grafuri.

Alte izvoare ale teoriei grafurilor sunt: studiul rețelelor electrice, problema celor patru culori, aplicațiile teoriei grafurilor în chimie (inițiate de Cayley), probleme hamiltoniene, grafuri planare, etc.

Fizicianul Kirchhoff a studiat la mijlocul secolului trecut rețelele electrice cu metode care aparțin astăzi teoriei grafurilor, contribuind la dezvoltarea acestei teorii (în anul 1845 a formulat legile care guvernează circulația curentului într-o rețea electrică iar în anul 1847 tot el a arătat cum poate fi construită într-un graf o mulțime fundamentală de cicluri, demonstrând că, pentru orice graf conex cu n vârfuri și m muchii, o mulțime fundamentală de cicluri conține întotdeauna $m-n+1$ cicluri).

Lucrarea de față are drept scop să-i introducă pe cititorii ei, cu diferite niveluri de pregătire, în aplicațiile de **Teoria Grafurilor**. Ideea de bază, în jurul căreia gravitează întregul material, este de a găsi decizii pe termen scurt la problemele practice prin intermediul unor modele matematice aparținând teoriei grafurilor.

Punctul de vedere și tematica lucrării reflectă preocuparea de a găsi posibilitatea înțelegerii avantajelor pe care le oferă matematica unui cerc larg de cititori, începând chiar cu elevii de liceu. Pentru elevii de liceu, materialul prezentat constituie un stimulent în învățarea algoritmică și logică a problemelor din domeniul fizicii, chimiei, informaticii etc.

În scopul atingerii obiectivelor amintite, lucrarea, reluând teoria matricilor, vectorilor și sistemelor liniare și prezentând-o într-o manieră accesibilă, construiește o gamă largă de procedee și metode matematice pe care le trimite la probleme de interes practic. Noțiunile

sunt prezentate treptat în funcție de gradul de dificultate, iar necesitatea introducerii acestora este justificată cu numeroase exemple practice.

Lucrarea este structurată în patru capitole și anume:

- Capitolul I - *Elemente de teoria grafurilor*, care cuprinde noțiuni generale despre grafuri orientate și neorientate, tipuri de conexitate în grafuri, matrici atașate unui graf orientat, metode pentru determinarea componentelor tare conexe ale unui graf și aplicații practice ale grafurilor;

- Capitolul II - *Drumuri minime-maxime în grafuri*, care cuprinde transport pe drumuri minime-maxime în grafuri, algoritmi pentru determinarea drumurilor minime și aplicații;

- Capitolul III, *Drumuri hamiltoniene în grafuri*, care cuprinde transport pe drumuri hamiltoniene, algoritmi pentru determinarea drumurilor hamiltoniene și aplicații la transporturi pe drumuri hamiltoniene;

- Capitolul IV, *Aspecte metodice*, care cuprinde aspecte metodice și aplicații.

Fiecare capitol cuprinde o prezentare teoretică a informației, iar la final aplicații practice la noțiunile prezentate.

Am ferma convingere că maniera în care este concepută lucrarea va duce la creșterea interesului pentru informatică și mai ales pentru ***Teoria Grafurilor*** cu aplicabilitate mare în toate domeniile de activitate, arătând mai clar universalitatea aplicațiilor ei.

CAPITOLUL 1

GRAFURI ORIENTATE ȘI NEORIENTATE

1.1. Noțiuni generale despre grafuri

Definiția 1.1.1. Fie V o mulțime finită și nevidă de elemente (pe care le numim *vârfuri*, *noduri* sau *puncte*) și fie E o mulțime de perechi ordonate de vârfuri din V (pe care le numim *arce* sau *linii*).

Cuplul format din mulțimile V și E se numește **graf orientat** și se notează prin $G = (V, E)$.

Observația 1.1.1. Mulțimea V , a vârfurilor grafului, poate fi și infinită, dar numărabilă, caz în care graful este numit graf *infinit*. De asemenea, pot fi luate în considerație grafuri la care între doua vârfuri pot exista mai multe arce cu același sens (așa-numitele *multigrafuri*). Nu voi folosi astfel de grafuri în lucrarea de față.

Observația 1.1.2. Un arc de forma (x_i, x_j) cu $x_i \in V$, se numește *bucură*. În lucrarea de față, nu voi lucra cu grafuri ce conțin bucle.

Exemplul 1.1.1. Fie $V = [x_1, x_2, x_3, x_4, x_5]$ mulțimea operațiilor de prelucrare, ce trebuie aplicate pentru fabricarea unei piese P , aceste operații reprezentând vârfurile unui graf orientat G . Vom considera că graful G conține arcul (x_i, x_j) , cu $x_i, x_j \in V$, dacă operația notată cu x_j nu poate fi efectuată dacă nu s-a efectuat anterior (indiferent când) operația x_i . Conform acestei convenții, să presupunem că mulțimea arcelor grafului G este următoarea:
 $E = [(x_1, x_2), (x_1, x_3), (x_1, x_4), (x_1, x_5), (x_2, x_4), (x_2, x_5), (x_3, x_4), (x_3, x_5), (x_4, x_5)]$

Fiind cunoscute mulțimile V și E , graful $G = (V, E)$, care exprimă matematic, fenomenul de realizare a piesei P , este perfect determinat.

Deosebit de utilă este adesea *reprezentarea geometrică* a unui graf orientat, care se obține în modul următor:

- se reprezintă vârfurile grafului ca puncte în plan;
- fiecare arc $(x_i, x_j) \in E$ se reprezintă printr-o linie ce unește punctul corespunzător vârfului x_i cu punctul corespunzător vârfului x_j , linie pe care se găsește o săgeată, cu sensul de la x_i către x_j .

Dăm, ca exemplu, reprezentarea geometrică a grafului $G=(V, E)$, definit în exemplul precedent, aceasta având forma din figura 1.1.1.

În unele aplicații practice, la folosirea conceptului de graf nu este necesară orientarea liniilor grafului. În astfel de situații, în locul mulțimii de arce, se consideră o mulțime analoagă numită *mulțimea muchiilor*. O *muchie*, ce unește vârfurile x_i și x_j , este perechea neordonată a acestor vârfuri (nu contează ordinea lor); ea se reprezintă simbolic prin $[x_i, x_j]$ și este evident identică cu $[x_j, x_i]$. Putem astfel să vorbim despre graful neorientat $G=(V, E)$.

Observația 1.1.3. Un graf neorientat $G=(V, E)$ poate fi înlocuit în aplicații printr-unul orientat, cu aceeași mulțime de vârfuri, dar corespunzător fiecărei muchii $[x_i, x_j] \in E$, introducându-se în noul graf două arce de sens contrar, (x_i, x_j) și (x_j, x_i) . Pot fi folosite, de asemenea, și *grafuri mixte*, care conțin atât arce, cât și muchii.

Observația 1.1.4. Reprezentarea geometrică a unui graf neorientat se face asemănător cu aceea a unui graf orientat, cu deosebirea că nu mai punem săgeți pe linii.

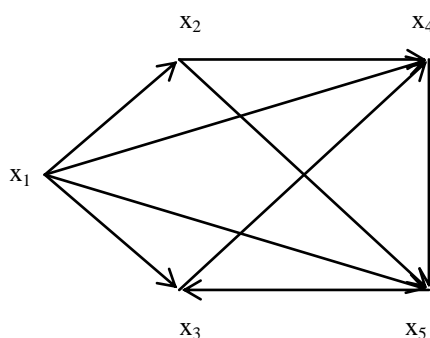


Fig.1.1.1

Exemplul 1.1.2. Să interpretăm punctele importante ale unei rețele de curent alternativ ca vârfuri, iar liniile electrice care le unesc, ca niște muchii (evident, nu contează sensul acestora). Se definește astfel un graf neorientat, ce poate reprezenta matematic rețeaua electrică respectivă.

Considerând un arc (x_i, x_j) , al unui graf orientat, spunem că x_i este *extremitatea inițială*, iar x_j *extremitatea finală* a arcului. Evident, în cazul unei muchii, distincția dintre cele două extremități nu se mai face.

Definiția 1.1.2. Două vârfuri ale unui graf, $G=(V, E)$, se numesc **vârfuri adiacente**, dacă ele sunt unite în graf prin cel puțin un arc (o muchie). Spunem, de asemenea că un arc (muchie) este **incident** unui vârf, dacă vârful respectiv reprezintă una din extremitățile arcului (muchiei). Mai precis, în cazul grafului orientat, dacă este **incident exterior** vârfului, iar dacă vârful este extremitatea finală a arcului, spunem că arcul este **incident interior** vârfului.

Ca exemplu, să luăm arcul (x_2, x_5) din figura 1.1.1 el este incident exterior vârfului x_2 și incident interior vârfului x_5 .

Definiția 1.1.3. Numărul de arce incidente exterior unui vârf $x_i \in V$, poartă numele de **grad exterior de incidență** corespunzător vârfului x_i și se notează cu $g^+(x_i)$, sau, mai simplu, cu g_i^+ . Analog, numărul de arce incident interior vârfului x_i este **gradul interior de incidență** al vârfului x_i și se notează cu $g^-(x_i)$ sau cu g_i^- . **Gradul de incidență** al unui vârf $x_i \in V$ (fără specificarea de grad exterior sau interior) este dat de numărul de arce incidente vârfului x_i ; el se notează cu $g(x_i)$, sau mai simplu cu g_i . În cazul unui graf orientat și fără bucle, avem evident: $g_i = g_i^+ + g_i^-$.

Pentru aplicarea grafurilor la problemele puse de practică, unui graf $G=(V,E)$ i se mai atașează adesea o funcție reală $w : E \rightarrow \mathbb{R}$, numită *funcție valoare*. Această funcție face ca fiecărui arc $(x_i, x_j) \in E$ să-i corespundă o valoare reală notată cu $w(x_i, x_j)$ și numită *valoarea arcului*. În acest fel, graful devine un **graf ponderat** și se notează, simbolic prin $G=(V,E,w)$. Desigur, asemănător se pot atașa valori și muchiiilor unui graf neorientat.

În ceea ce privește interpretarea valorilor ce se atașează, arcelor (muchiiilor) unui graf, acesta depinde de problema practică ce se rezolvă cu ajutorul modelului graf corespunzător.

Exemplul 1.1.3 Se dă graful $G = (V, E, w)$, în care:

- vârfurile reprezintă punctele ce trebuie aprovizionate de un depozit,
- arcele reprezintă posibilitățile de transport între diferitele puncte ale sistemului,
- valoarea $w(x_i, x_j)$, atașată arcului $(x_i, x_j) \in E$, reprezintă cheltuielile de transport al unei unități de produs între punctele x_i și x_j (în această ordine).

Definiția 1.1.4. Se numește **drum** între vârfurile x_i și x_j , într-un graf orientat, o succesiune de arce, astfel încât extremitatea inițială a fiecăruia coincide cu extremitatea

finală a celui precedent, excepție făcând extremitatea inițială a primului arc, care este x_i și extremitatea finală a ultimului arc, care este x_j . Drumul descris se reprezintă pe scurt prin $d(x_i, x_j)$, iar desfășurat se scrie astfel:

$$d(x_i, x_j) = (x_i, x_k, x_l, x_m, \dots, x_p, x_j),$$

cu convenția că nu se scriu de două ori extremitățile ce coincid, pentru fiecare pereche de arce alăturate. **Lungimea unui drum** este, prin convenție, numărul de arce din care este compus drumul. Evident, un drum de lungime egală cu unu nu este altceva decât un arc. Valoarea unui drum este egală cu suma valorilor arcelor ce formează drumul. Astfel, fiind dată funcția valoare $w: E \rightarrow R$, valoarea drumului $d(x_i, x_j)$ este dată prin:

$$w[d(x_i, x_j)] = \sum_{(x_s, x_t) \in d} w(x_s, x_t)$$

Definiția 1.1.5. Se numește **drum elementar** un drum care nu trece de două ori prin același vârf.

Definiția 1.1.6. Se numește **drum hamiltonian** - notat prescurtat cu **DH** - un drum care trece prin toate vârfurile grafului, câte o singură dată.

Dacă un graf orientat conține cel puțin un drum de la x_i la x_j , spunem că x_i atinge pe x_j , sau că x_j este atins de la x_i . Numărul de vârfuri, ce pot fi atinse de la vârfuri x_i , poartă numele de putere de atingere a vârfului x_i și se notează cu $p(x_i)$, sau mai scurt cu P_i . Două vârfuri, x_i și x_j , astfel încât în graful G , x_i atinge pe x_j , iar x_j atinge pe x_i , se zic vârfuri *mutual atinse*; aceasta înseamnă de fapt că graful G conține cel puțin un drum de la x_i la x_j și cel puțin un drum de la x_j la x_i .

Definiția 1.1.7. Se numește **circuit** într-un graf, un drum închis, în sensul că extremitatea finală a drumului coincide cu extremitatea inițială a sa.

Definiția 1.1.8. Se numește **circuit Hamiltonian**, notat pe scurt prin **CH**, un circuit care trece prin toate vârfurile grafului câte o singură dată, cu excepția extremităților, ce coincid.

Toate conceptele, definite mai sus, ce exprimă legături între diferitele vârfuri ale unui graf, se numesc *concepte orientate*, deoarece în definirea lor se ține seama de orientarea arcelor. La baza lor stă noțiunea de drum, acesta având un sens ce corespunde cu sensurile tuturor arcelor ce îl compun.

Exemplul 1.1.4. Fie graful orientat din figura 1.1.2. Unul dintre drumurile de la x_1 la x_5 este format din arcele (x_1, x_4) și (x_4, x_5) , puse cap la cap. El se scrie astfel: $d(x_1, x_5) = (x_1, x_4, x_5)$ și are o lungime egală cu doi (este format din două arce). Un alt drum de la x_1 , la x_5 , de asemenea, cu o lungime egală cu doi, este drumul $d'(x_1, x_5) = (x_1, x_3, x_5)$. Există în graful dat și alte drumuri între vârfurile x_1 și x_5 . Dintre acestea, iată unul care nu este elementar, trecând de două ori prin vârful x_3 : $d''(x_1, x_5) = (x_1, x_3, x_6, x_2, x_3, x_5)$.

De asemenea, prezentăm două drumuri hamiltoniene ale grafului:

$$DH1 = (x_1, x_2, x_3, x_6, x_4, x_5)$$

$$DH2 = (x_3, x_6, x_4, x_5, x_1, x_2).$$

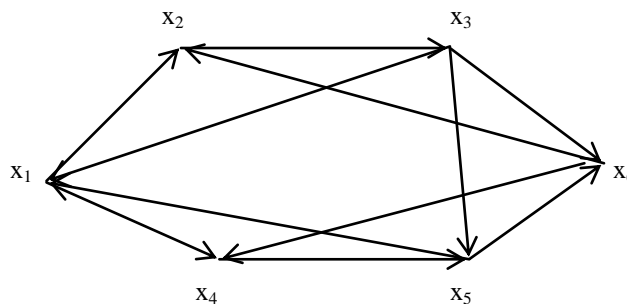


Fig. 1.1.2

Iată și un circuit hamiltonian: $CH = (x_1, x_2, x_3, x_6, x_4, x_5, x_1)$. În sfârșit, dăm un exemplu de circuit, al grafului, care nu este hamiltonian: $d(x_2, x_2) = (x_2, x_3, x_5, x_2)$.

Conceptele neorientate, corespunzătoare celor prezentate mai sus, se definesc într-un graf neorientat.

Definiția 1.1.9. Într-un graf neorientat o succesiune de muchii puse cap la cap formează un **lanț**. Dacă lanțul trece prin fiecare vârf al grafului, câte o singură dată, el se numește **lanț hamiltonian**, prescurtat LH.

Definiția 1.1.10. Un lanț închis (care se termină cu același vârf cu care începe) se numește **ciclu**. Un ciclu este **hamiltonian** dacă trece prin toate vârfurile grafului câte o singură dată, cu excepția primului și ultimului vârf, care coincid; el se notează prescurtat prin CH.

Observația 1.1.5. În lucrarea de față, vom scrie lanțurile ca și pe drumuri, dar cu paranteze drepte.

Conceptele neorientate se pot defini și într-un graf orientat, dar ele se construiesc fără a ține seama de orientările arcelor, ci considerându-le ca pe muchii. Astfel, pentru două arce alăturate, ce intră în compunerea unui lanț definit pe un graf orientat, pot coincide două extremități de același fel (fie extremitățile inițiale, fie cele finale). În acest fel, unele arce ale lanțului au un sens ce coincide cu sensul de citire a lanțului, iar altele au sens contrar sensului: de citire a lanțului. Exemplele de mai jos vor lămurii mai bine aceste afirmații.

Exemplul 1.1.5. Fie graful neorientat din figura 1.1.3, în care există următoarele lanțuri între vârfurile x_2 și x_4 : $I1(x_2, x_4) = [x_2, x_3, x_4]$, $I2(x_2, x_4) = [x_2, x_5, x_4]$, $I3(x_2, x_4) = [x_2, x_5, x_3, x_4]$ etc.

Un lanț hamiltonian este următorul: $LH1 = [x_5, x_1, x_2, x_3, x_4]$, iar un ciclu hamiltonian este $CH1 = [x_1, x_2, x_3, x_4, x_5, x_1]$. Exemple de cicluri: $[x_1, x_2, x_5]$, $[x_2, x_5, x_4, x_3]$ etc.

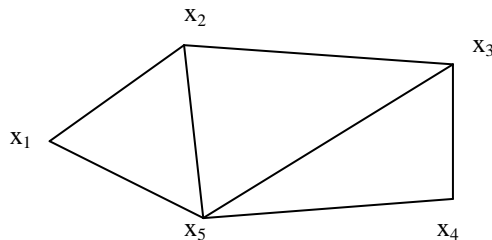


Fig.1.1.3

Exemplul 1.1.5. Fie acum graful orientat din figura 1.1.4, în care considerăm următorul lanț: $l(x_1, x_2) = [x_1, x_5, x_3, x_2]$. El este format cu ajutorul arcelor (x_1, x_5) , (x_3, x_5) și (x_2, x_3) , dintre care numai sensul primului coincide cu sensul de citire a lanțului, celelalte două având sensuri contrare și anume:

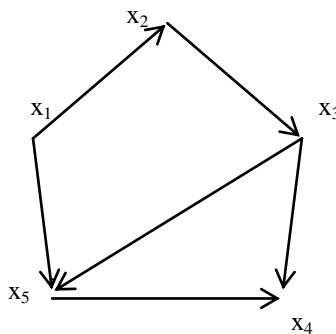


Fig. 1.1.4

- arcul (x_3, x_5) apare în lanț cu sensul de citire de la x_5 , către x_3

- arcul (x_2, x_3) apare în lanț cu sensul de citire de la x_3 către x_2

În aceeași figură 1.1.4, putem considera și ciclul $[x_1, x_2, x_3, x_4, x_5, x_1]$, în care ultimele două arce sunt parcurse în sens contrar sensului lor.

1.2. Tipuri de conexiune a grafurilor

Definiția 1.2.1. Un graf (orientat sau neorientat) se numește **graf conex**, dacă pentru orice vârfuri, x_i și x_j ale sale, există în graf cel puțin un lanț care le leagă. În cazul contrar, când graful conține cel puțin două vârfuri nelegate prin nici un lanț, el este **graf neconex**. Ca exemple de grafuri conexe, putem da pe cele din figurile 1.1.1 - 1.1.4, iar ca exemplu de graf neconex dăm graful din figura 1.2.1, care are șase vârfuri. Într-adevăr, în acest din urmă graf, vârfurile x_1 și x_4 (de exemplu), nu sunt unite prin nici un lanț.

Definiția 1.2.2. Un graf orientat este **tare conex**, dacă orice două vârfuri ale sale sunt mutual atinse. Aceasta înseamnă că, dacă luăm spre cercetare două vârfuri ale grafului, x_i și x_j , alese la întâmplare, graful conține cel puțin un drum de la x_i la x_j și cel puțin un drum de la x_j la x_i .

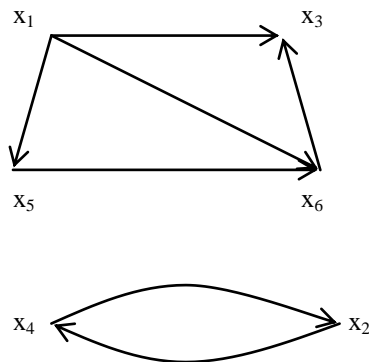


Fig. 1.1.1

Observația 1.2.1. Proprietatea de tare conexiune este foarte importantă, de exemplu, în rețelele de comunicații, în cadrul cărora trebuie să se poată ajunge oricând de la un punct oarecare x_i al rețelei, la un alt punct x_j al aceleiași rețele. Dăm, ca exemplu de graf tare conex, pe cel din figura 1.2.2, verificarea proprietății de tare conexiune rămânând pe seama cititorului. Desigur, dacă graful ar avea un număr mare de vârfuri această verificare nu se mai poate face cu ușurință în mod direct, fiind necesară o metodă matematică.

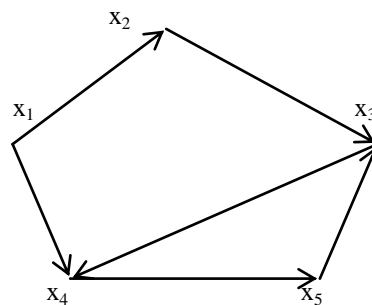


Fig. 1.2.2

Definiția 1.2.3. Dacă dintr-un graf G eliminăm o parte din arce (sau muchii), dar păstrăm toate vârfurile, obținem un alt graf, care se numește **graf parțial** al lui G .

Definiția 1.2.3. Dacă dintr-un graf G eliminăm o parte din vârfuri, împreună cu toate arcele incidente cu vârfurile eliminate, se obține un **subgraf** al lui G .

Definiția 1.2.4. Un subgraf maximal tare conex, al unui graf orientat G , se numește **componentă tare conexă** a lui G . Aici, cuvintul „maximal” are sensul că dacă subgraful menționat are măcar un vârf în plus, atunci își pierde proprietatea de tare conexiune. Analog, un subgraf maximal conex, al unui graf G (orientat sau neorientat), se numește **componentă conexă** a grafului G .

Observația 1.2.2. În lucrarea de față vom lucra numai cu grafuri conexe, deci cu grafuri ce au câte o singură componentă conexă.

Exemplul 1.2.1. Graful din figura 1.2.1. are două componente conexe și anume subgrafurile cu mulțimile de vârfuri $\{x_1, x_3, x_5, x_6\}$ și $\{x_2, x_4\}$. Într-adevăr, dacă la prima submulțime a vârfurilor am adăuga fie vârful x_2 , fie vârful x_4 , am obține subgrafuri neconexe ale celui dat inițial. Pe de altă parte, să cercetăm graful G , din figura 1.2.3. Subgraful lui G , care conține numai vârfurile x_1, x_2 și x_5 , este o componentă tare conexă, deoarece:

- între aceste vârfuri există drumuri oricum le-am grupa, ceea ce se observă ușor, în mod direct,

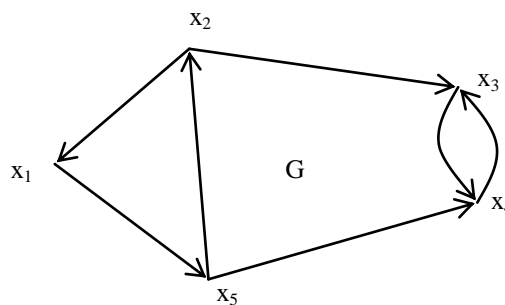


Fig. 1.2.3

- dacă adăugăm la aceste vârfuri, fie vârful x_3 , fie vârful x_4 , fie pe amândouă, în subgraful obținut există grupe de vârfuri între care nu există nici un drum.

Cea de a doua componentă tare conexă, a grafului G , este reprezentată de subgraful cu vârfurile x_3 și x_4 , ceea ce se verifică, de asemenea, ușor, în mod direct.

Definiția 1.2.5. Graful condensat, corespunzător unui graf orientat G , este un graf $G^* = (V^*, E^*)$, definit astfel:

- vârfurile lui sunt componentele tare conexe ale grafului G , notate, de exemplu, prin C_1, C_2, \dots, C_k ,

- mulțimea E^* conține arcul (C_r, C_s) , dacă și numai dacă graful inițial G conține cel puțin un arc cu extremitatea inițială într-un vârf din C_r și cu extremitatea finală într-un vârf din C_s . Ca exemplu, dăm graficul condensat al grafului G reprezentat în figura 1.2.3; el are numai două vârfuri (corespunzătoare celor două componente tare conexe indicate mai sus) și un singur arc, așa cum apare în figura 1.2.4. Într-adevăr, dacă notăm cu C_1 componenta tare conexă a lui G , care conține vârfurile x_1, x_2 și x_5 , iar cu C_2 cealaltă componentă tare conexă, graful G^* va conține arcul (C_1, C_2) , deoarece G are arcele (x_2, x_3) și (x_5, x_4) , dar nu va conține arcul (C_2, C_1) , neexistând în G nici un arc de la x_3 sau x_4 spre x_1, x_2 sau x_5 .

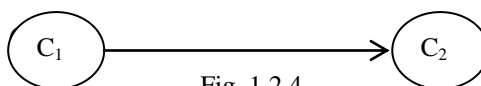


Fig. 1.2.4

1.3. Matrici atașate unui graf orientat

Pentru rezolvarea diferitelor probleme practice, cu ajutorul modelelor-grafuri, mai ales în cazul unor grafuri de dimensiuni mari, este necesar ca acestea să fie reprezentate sub forma unor matrici, pe care calculele se efectuează ușor.

Definiția 1.3.1. Fiind dat un graf orientat G , cu n vârfuri, se numește **matricea arcelor** a acestui graf o matrice pătrată A , de ordinul n , ale cărei elemente sunt egale cu zero și cu unu, construită după regula următoare: elementul ce se află la intersecția liniei de ordin i , cu coloana de ordin j , este egal cu unu dacă graful G conține arcul (x_i, x_j) și este egal cu zero dacă graful nu conține acest arc. Putem reprezenta matricea arcelor grafului $G=(V,E)$ simbolic, sub forma următoare:

$$A=(a_{ij}), i=1, 2, \dots, n; j= 1, 2, \dots, n,$$

Unde:

$$a_{ij} = \begin{cases} 1, & \text{dac } \ddot{a} (x_i, x_j) \in A \\ 0, & \text{în caz contrar} \end{cases}$$

Observația 1.3.1. Elementele matricei arcelor grafului G le vom nota totdeauna cu a_{ij} , $i=1, 2, \dots, n; j= 1, 2, \dots, n$.

Deoarece toate elementele matricei A sunt egale cu zero și unu, se spune că aceasta este o *matrice booleană*. Accentuăm încă odată, că existența unui arc, între două vârfuri fixate ale grafului, este pusă în evidență de un element egal cu unu din matricea A , după regula: extremitatea inițială a arcului corespunde la linie, iar extremitatea finală la coloana matricei, în care se găsește elementul egal cu unu. De aceea, putem spune că liniile consecutive ale matricei A corespund la vârfurile grafului, așezate într-o ordine stabilită (de exemplu, în ordinea crescătoare a indicilor, x_1, x_2, \dots, x_n), iar coloanele consecutive ale matricei A corespund, de asemenea, vârfurilor grafului, așezate în aceeași ordine. Pentru a evidenția această corespondență, ce va ajuta în urmărirea calculelor ce vor fi descrise mai departe, vârfurile în cauză se pot trece pe marginile matricei, în partea stângă (pentru linii) și în partea de sus (pentru coloane), așa cum rezultă și din exemplul ce urmează.

Exemplul 1.3.1. Să considerăm graful orientat G , din figura 1.3.1. Matricea arcelor acestui graf este următoarea:

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

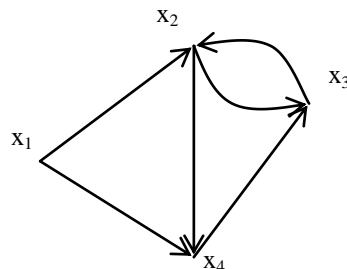


Fig.1.3.1

Pentru o urmărire mai ușoară a modului în care s-a completat această matrice, ca și pentru folosirea ei în eventualele calcule, vom obișnui să scriem matricea arcelor ca în tabelul 1.3.1, notând în evidență vârfurile, ce corespund la diferitele linii și coloane ale ei.

Astfel, de exemplu, elementul egal cu unu de pe linia x_1 , și coloana x_2 , arată existența în graf a arcului (x_1, x_2) . De asemenea, elementul egal cu zero, de pe linia x_1 și coloana x_3 , arată că în grafurile G nu există arcul (x_1, x_3) .

Observația 1.3.2. Conform convenției făcute, în partea stângă a matricei A sunt scrise totdeauna plecările arcelor, iar în partea de sus a matricei sosirile acestor arce (arce corespunzătoare elementelor egale cu unu).

Observația 1.3.3. Faptul că un graf G nu conține bucle se poate deduce din inexistența elementelor egale cu unu, de pe diagonala principală a matricei arcelor acestui graf.

Tabelul 1.3.1

	x_1	x_2	x_3	x_4
x_1	0	1	0	1
x_1	0	0	1	1
x_1	0	1	0	0
x_1	0	0	1	0

Observația 1.3.4. Deoarece elementele egale cu unu, de pe linia x_i , a matricei arcelor unui graf G , reprezintă singurele arce ce au ca punct de plecare vârful x_i , suma elementelor liniei x_i este egală cu gradul exterior de incidență al acestui vârf. Analog, suma elementelor de pe coloana x_j , din matricea arcelor unui graf G , este egală cu gradul interior de incidență al vârfului x_j .

Definiția 1.3.2. O altă matrice, ce se atașează unui graf orientat G cu n vârfuri și pe care o folosim în mod deosebit, este **matricea drumurilor**, pe care o notăm cu D . Ea este, de asemenea, o matrice booleană, pătratică de ordinul n . O vom reprezenta simbolic prin:

$$D = (d_{ij}), i = 1, 2, \dots, n; j = 1, 2, \dots, n,$$

unde:

$$d_{ij} = \begin{cases} 1, & \text{daca } G \text{ contine cel puțin un drum } d(x_i, x_j), \\ 0, & \text{în caz contrar} \end{cases}$$

Pentru o utilizare comodă a matricei drumurilor, aceasta va fi reprezentată sub forma unui tabel asemănător cu cel folosit la matricea arcelor, adică în care atât liniile cât și coloanele corespund la vârfurile grafului, puse într-o anumită ordine. Convenția este aceeași, liniile corespunzând la plecările de drumuri, iar coloanele la sosirile acestor drumuri.

Observația 1.3.5. Existența unor elemente egale cu unu, pe diagonala principală a matricei drumurilor, arată existența unor circuite în grafurile G . Evident, dacă matricea drumurilor unui graf are toate elementele de pe diagonala principală egale cu zero, grafurile corespunzător nu are circuite și reciproc.

Observația 1.3.6. Grafurile G este tare conex dacă și numai dacă toate elementele matricei drumurilor sunt egale cu unu. Această observație poate folosi ca metodă de verificare a faptului că un graf este sau nu tare conex.

Observația 1.3.7. Elementele egale cu unu, de pe linia x_i , a matricei drumurilor unui graf G , arată spre ce vârfuri pornesc drumuri din x_i , adică ce vârfuri sunt atinse de la x_i . În acest sens, suma elementelor de pe linia x_i a matricei drumurilor este egală cu puterea de atingere a vârfului x_i . În plus, putem preciza că elementele egale cu unu, de pe coloana x_i a matricei drumurilor, arată de la ce vârfuri este atins x_i în grafurile considerate

Exemplul 1.3.2. Metodă de calcul a matricei drumurilor. În cele ce urmează vom arăta, pe baza unui exemplu, cum se poate calcula matricea drumurilor unui graf orientat, pornind de la matricea arcelor a sa. Fie grafurile G , din figura 1.3.2, a cărui matrice a arcelor A este reprezentată în tabelul 1.3.2. Să calculăm, mai întâi, prima linie din matricea drumurilor D , linie ce corespunde vârfului x_1 : ea va conține ca elemente egale cu unu, pe cele ce indică existența drumurilor din graf, ce pornesc din x_1 . Există, evident, două drumuri cu lungimea

egală cu unu, de acest fel, ele fiind formate din arcele (x_1, x_2) și (x_1, x_3) , deci $d_{12}=1$ și $d_{13}=1$. Aceasta înseamnă, în general, că linia x_1 , din matricea drumurilor conține, pe lângă alte eventuale elemente egale cu unu: pe cele de pe linia corespunzătoare a matricei arcelor. Completând, într-o primă etapă, linia x_1 , din D cu elementele egale, cu unu ce corespund la drumuri de lungime egală cu unu, aceasta are forma următoare (formă intermediară):

(1)

	x_1	x_2	x_3	x_4	x_5	x_6
x_1		1v	1			

Tabelul 1.3.2

	x_1	x_2	x_3	x_4	x_5	x_6
x_1	0	1	1	0	0	0
x_2	0	0	0	1	1	0
x_3	1	0	0	0	1	0
x_4	0	0	0	0	0	1
x_5	0	0	0	1	0	1
x_6	0	0	0	0	0	0

Aici, pe locurile rămase libere considerăm zerouri, pe care nu le scriem, deocamdată, deoarece se pot transforma în elemente egale cu unu (dacă există drumuri corespunzătoare, de lungimi mai mari ca unu).

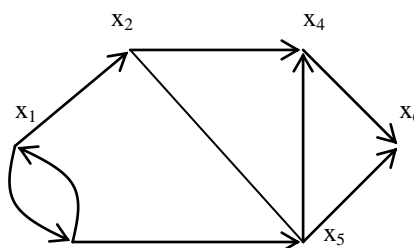


Fig.1.3.2

Să cercetăm acum existența sau inexistența, pentru locurile rămase libere, a unor drumuri de câte două arce. Deoarece aceste drumuri trebuie să plece din x_1 , ele încep, fie cu arcul (x_1, x_2) , fie, cu (x_1, x_3) și se continuă cu încă câte un arc, ce pornește respectiv din x_2 sau x_3 . Să considerăm aceste posibilități pe rând. Astfel, luând spre cercetare arcul (x_1, x_2) indicat prin primul element egal cu unu de pe linia incompletă de mai sus), el trebuie prelungit cu un alt arc, ce pornește din x_2 . Astfel de arce sunt indicate însă de elementele egale cu unu, de pe linia x_2 din matricea arcelor; există două astfel de elemente, unul în dreptul coloanei x_4 și altul în dreptul coloanei x_5 , ele indicând deci existența în graf a arcelor (x_2, x_4) și (x_2, x_5) . Cuplând arcele în cauză, rezultă că graful dat conține drumurile (x_1, x_2, x_4) și (x_1, x_2, x_5) , adică x_1 atinge atât pe x_4 cât și pe x_5 . Consemnăm cele de mai sus, prin completarea liniei x_1 , din matricea drumurilor, cu încă două elemente, ea căpătând forma intermediară următoare:

(2)

	x_1	x_2	x_3	x_4	x_5	x_6
x_1		1v	1v	1	1	

Și de această dată, locurile rămase libere sunt considerate ca zerouri, dar nu au fost efectiv completate, deoarece unele dintre ele se pot transforma în elemente unu (dacă există în graf unele drumuri de câte trei sau mai multe arce).

A fost folosit elementul egal cu unu, din dreptul coloanei x_2 , al schemei (1), fapt indicat pe această schemă printr-o bifă. Continuăm raționamentul folosind elementul egal cu unu din dreptul coloanei x_3 , bifat și el în schema (2). Arcul (x_1, x_3) poate fi prelungit cu un arc ce pleacă din x_3 : există două astfel de arce (x_3, x_1) și (x_3, x_5) , indicate de elementele egale cu unu de pe linia x_3 a matricei A. Aceasta înseamnă că graful G conține drumurile (x_1, x_3, x_1) și (x_1, x_3, x_5) . Existența primului, dintre aceste două drumuri, ne determină să punem $d_{11}=1$ (primul element al liniei x_1 din D este egal cu unu). Cel de al doilea, drum nu mai modifică linia x_1 din schema (2), deoarece în această schemă este marcat anterior $d_{15}=1$. Aceasta se explică prin faptul că de la x_1 la x_5 , există mai mult decât un drum (unul prin x_2 , altul prin x_3), iar noi marcăm cu unu existența a cel puțin unui drum între vârfurile respective. Consemnarea raționamentului de mai sus conduce la următoarea formă intermediară a liniei x_1 , din matricea D:

$$(3) \quad \begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_1 & 1v & 1v & 1v & 1v & 1 & \end{array}$$

în care a rămas necompletat un singur loc. Pentru eventuala completare cu unu, a locului ramas liber, să cercetăm existența în graf a drumurilor de către trei arce. Un astfel de drum este compus dintr-un drum de două arce, prelungit cu încă un arc. Drumurile de câte două arce sunt reprezentate însă sub forma elementelor egale cu unu, apărute în etapa precedentă există drumuri de acest fel de la x_1 la x_1 , x_4 și x_5 , deoarece pe linia x_1 din schema (3) au apărut elemente egale cu unu în dreptul coloanelor x_1 , x_4 și x_5 . Cercetarea posibilităților de prelungire a lor cu câte un arc decurge în felul următor :

a) Drumul de la x_1 la x_1 se poate prelungi cu arcul (x_1, x_2) sau cu arcul (x_1, x_3) , dând drumuri de la x_1 la x_2 sau de la x_1 la x_3 . Existența unor asemenea drumuri este indicată însă în schema (3), prin elemente egale cu unu în dreptul coloanelor x_2 , și x_3 (au existat drumuri cu lungimi mai mici, între vârfurile respective).

b) Drumul de la x_1 , la x_4 se poate prelungi cu arcul (x_4, x_6) indicat de elementul $a_{46}=1$ de pe linia x_4 a matricei A. Există, deci un drum de trei arce, de la x_1 , la x_6 , ceea ce se va marca printr-un element egal cu unu in locul rămas liber în schema (3). Obținem astfel schema următoare:

$$(4) \quad \begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}$$

în care nu mai are rost să continuăm operațiile, deoarece nu mai poate apărea nici un element egal cu unu. Prima linie, a matricei drumurilor grafului G, are numai elemente egale cu unu, ceea ce înseamnă că x_1 atinge toate vârfurile grafului.

Să sintetizăm calculele efectuate până acum, pentru a da o regulă generală, de determinare a liniei x_1 , din matricea D, metodă în care să nu mai fie nevoie de a interpreta de fiecare dată unele sau altele dintre drumurile existente în graf. Pentru aceasta, vom folosi operația de *adunare booleană* a doua numere ce se execută numai în cadrul mulțimii $\{0,1\}$. Semnul acestei operații este semnul obișnuit de adunare a numerelor (semnul plus), cu un punct deasupra. Cele patru posibilități de adunare booleană, a două, numere, se definesc astfel:

$$0 + 0 = 0; \quad 0 + 1 = 1; \quad 1 + 1 = 1; \quad 1 + 1 = 1.$$

Se observă că adunarea booleană a unui termen egal cu unu, face totdeauna ca rezultatul să fie egal cu unu (indiferent de termenul la care se adună el).

Revenind la modul de, calcul al liniei x_1 , din matricea D, observăm că ea începe cu linia corespunzătoare (linia x_1 .) din matricea A (vezi schema (1)). Cele două elemente egale

cu unu, de pe această linie, se găsesc în dreptul coloanelor x_2 și x_3 , de aceea, adunăm boolean, la linia din schema (1), liniile x_2 și x_3 din matricea arcelor (evident, pe rând, bifând elementele folosite, obținând schemele (2) și (3)). În acest fel, au apărut noi elemente egale cu unu, ce se găsește în dreptul coloanelor x_1 , x_4 și x_5 . Conform acestor noi elemente apărute, adunăm boolean la linia din schema (3), liniile x_1 , x_4 și x_5 din matricea arcelor. Astfel de operații ar fi continuat cât era posibil dacă nu s-ar fi completat toată linia din, schema (4) cu elemente egale cu unu.

Din modul în care am raționat pentru calculul liniei x_1 , a matricei D, putem deduce cu ușurință că deducerea oricărei linii x_i , $i= 1, 2, \dots, n$, a matricei D, se poate face în mod asemănător, dar pornind inițial de la linia corespunzătoare x_i , a matricei arcelor. Astfel, pentru a calcula linia x_2 , a matricei drumurilor, pornim de la schema următoare, în care am încărcat numai elementele egale cu unu de pe linia x_2 a matricei A.

$$(5) \quad \begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_2 & & & & 1v & 1 & \end{array}$$

În schema (5) am bifat elementul egal cu unu din dreptul coloanei x_4 , urmând să adunăm boolean la linia x_2 a schemei, linia x_4 , din matricea A. Această adunare booleană mai aduce un element egal cu unu, în dreptul coloanei x_6 , ajungându-se la schema următoare:

$$(6) \quad \begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_2 & & & & 1v & 1v & 1v \end{array}$$

În schema (6) am bifat elementul egal cu unu, din dreptul coloanei x_5 , urmând să adunăm boolean, la linia din schema respectivă linia x_5 din matricea arcelor; această operație nu mai aduce nimic nou la schema (6), existând elemente egale cu unu în dreptul coloanelor x_4 și x_6 . Vom bifa deci și elementul egal cu unu, din dreptul coloanei x_6 , urmând să adunăm boolean linia x_6 din matricea arcelor; nici această operație nu aduce nimic nou, linia în cauză fiind plină cu zerouri. Deoarece procesul de mai sus nu mai poate continua (nemaexistând elemente egale cu unu nebifate), completăm efectiv locurile libere cu zerouri, rezultând astfel linia x_2 din matricea drumurilor:

$$(7) \quad \begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_2 & 0 & 0 & 0 & 1 & 1 & 1 \end{array}$$

Procedând în mod asemănător cu toate vârfurile grafului G, se obține matricea drumurilor acestui graf, ce se găsește în tabelul 1.3.3

Tabelul 1.3.3

	x_1	x_2	x_3	x_4	x_5	x_6
x_1	1	1	1	1	1	1
x_2	0	0	0	1	1	1
x_3	0	1	1	1	1	1
x_4	1	0	0	0	0	1
x_5	0	0	0	1	0	1
x_6	0	0	0	0	0	0

Observația 1.3.8. Ordinea în care se bifează elementele egale cu unu, pentru a face adunările booleene corespunzătoare, este indiferentă (chiar dacă aceste elemente au apărut

mai devreme sau mai târziu). De asemenea, dacă una din liniile matricei D' a fost calculată, fie ea linia x_i , iar la un moment dat urmează să adunăm boolean, la una din linii, linia x_i a matricei A , putem înlocui această adunare cu adunarea booleană a liniei x_i din matricea D , ceea ce poate scurta calculul.

Proprietatea 1.3.1. Deoarece $d_{ij} = 1$, dacă și numai dacă x_i , atinge pe x_j rezultă că suma elementelor de pe linia x_i , $i = 1, 2, \dots, n$, a matricei drumurilor, reprezintă puterea de atingere a vârfului x_i .

Proprietatea 1.3.2. Elementele egale cu unu, de pe diagonala principală a matricei drumurilor, arată existența în graf a unor circuite. De aici rezultă și o metodă de a testa dacă un graf este fără circuite: se calculează matricea drumurilor sale, ea trebuind să aibă pe diagonala principală numai elemente egale cu zero.

Teorema 1.3.1. Dacă în matricea drumurilor unui graf fără circuite, vârfurile sunt scrise într-o ordine descrescătoare a puterilor de atingere, această matrice este triunghiulară superior, adică toate elementele egale cu unu din ea se găsesc deasupra diagonalei principale.

Pentru justificarea acestei teoreme, să notăm vârfurile grafului, scrise într-o ordine descrescătoare a puterilor de atingere, prin $x_1, x_2, x_3, \dots, x_n$, deci:

$$p(x_1) \geq p(x_2) \geq p(x_3) \geq \dots \geq p(x_n)$$

Evident, aceste vârfuri sunt tot vârfurile $x_1, x_2, x_3, \dots, x_n$, dar scrise eventual într-o altă ordine. Menționăm în plus că atunci când unele vârfuri ale grafului au puteri de atingere egale, aceste vârfuri se așează între ele într-o ordine indiferentă, ceea ce justifică semnele de egalitate, din înșiruirea de mai sus a puterilor de atingere.

Matricea drumurilor grafului dat, în care vârfurile sunt scrise în ordinea $x_1', x_2', x_3', \dots, x_n'$, are forma din tabelul 1.3.4, ea fiind notată cu D' .

Tabelul 1.3.4

	$x_1', \dots, x_2', \dots, x_i', \dots, x_j', \dots, x_n'$
x_1'	D'_{ij}
\dots'	
x_2'	
\dots'	
x_i'	
\dots'	
x_j'	
\dots'	
x_n'	

Să notăm elementele ei cu d'_{ij} , $i=1, 2, \dots, n$; $j=1, 2, \dots, n$, elementul d'_{ij} fiind cel ce se află la intersecția liniei x_i' cu coloana x_j' . În tabelul 1.3.4, aceste elemente nu apar, ele fiind cercetate în cele ce urmează. Trebuie să arătăm că un element oarecare al matricei, care este egal cu unu, nu se poate afla decât deasupra diagonalei principale, marcată printr-o linie de tabel. Fie din elementul $d'_{ij} = 1$ și să arătăm că $i < j$, adică linia x_i' (respectiv coloana x_j') se află înaintea liniei x_j' (respectiv coloanei x_j'), ca în tabelul 1.3.4. aceasta înseamnă că trebuie să arătăm că vârful x_i' are o putere de atingere cel puțin egală cu aceea a vârfului x_j' . Pentru aceasta, să folosim definiția matricei drumurilor, din care deducem că graful dat conține cel puțin un drum de la x_i' la x_j' (avem $d'_{ij} = 1$). Privind figura 1.3.3, observăm că, în condițiile menționate, orice vârf atins de la x_j' , este atins și de la x_i' ; în plus x_i' atinge încă cel puțin un vârf în plus, pe x_j' , ceea ce înseamnă că $p(x_i') > p(x_j')$. Cu aceasta, justificarea teoremei este completă.

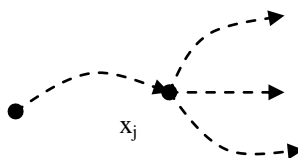


Figura 1.3.3

Definiția 1.3.3. Vom numi matricea drumurilor, în care vârfurile sunt scrise într-o ordine descrescătoare a puterilor de atingere, **matricea drumurilor triangularizată**, ea fiind notată cu D' . În această matrice, toate elementele de pe diagonala principală și de sub această diagonală, sunt egale cu zero. Așa cum s-a arătat, toate elementele diferite de zero (egale cu unu) ale matricei se află deasupra diagonalei principale, putând însă exista deasupra acestei diagonale și elemente nule.

Observația 1.3.9. Pentru a scrie matricea D' , corespunzătoare unui graf fără circuite, calculăm mai întâi matricea drumurilor grafului, din care se determină puterile de atingere ale vârfurilor. Apoi, în matricea D schimbăm ordinea liniilor, apoi a coloanelor, conform unei succesiuni a vârfurilor, având puteri de atingere descrescătoare. Exemplul ce urmează va lămurii pe deplin metoda.

Exemplul 1.3.3. Fie graful G , din figura 1.3.4, a cărui matrice a drumurilor este conținută de tabelul 1.3.5. În partea dreaptă a acestui tabel sunt calculate puterile de atingere ale vârfurilor, ca sume ale elementelor de pe liniile matricei D . Cea mai mare putere de atingere o are vârful $x_4(p_4)$, ceea ce înseamnă că în noua înșiruire a vârfurilor, vom începe cu $x_4(x'_1=x_4)$. Urmează vârfurile x_3 și x_5 , cu puterile de atingere egale ($p_3=p_5=2$), pe care le vom așeza într-o ordine arbitrară, de exemplu $x_2=x_3$ și $x_3=x_5$. Vor urma apoi în ordine, vârful $x'_4=x_2$ (cu puterea de atingere egală cu unu) și vârful $x_5=x_1$ (cu putere de atingere nulă). În cazul nostru, succesiunea teoretică a vârfurilor, x_1, x_2, x_3, x_4, x_5 , este următoarea:

x_4, x_3, x_5, x_2, x_1 .

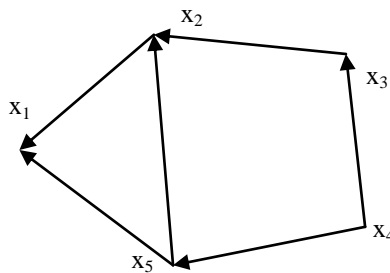


Figura. 1.3.4

Tabelul 1.3.5

	x_1	x_2	x_3	x_4	x_5	p_i
x_1	0	0	0	0	0	0
x_2	1	0	0	0	0	1
x_3	1	1	0	0	0	2
x_4	1	1	1	0	1	4
x_5	1	1	0	0	0	2

Să schimbăm, mai întâi, ordinea liniilor matricei D , conform acestei noi succesiuni a vârfurilor, păstrând ordinea coloanelor neschimbată; obținem astfel matricea intermediară \tilde{D} , din tabelul 1.3.6.

Tabelul 1.3.6

	x_1	x_2	x_3	x_4	x_5
x_4	1	1	1	0	1
x_3	1	1	0	0	0
x_5	1	1	0	0	0
x_2	1	0	0	0	0
x_1	0	0	0	0	0

În această matrice intermediară, schimbăm ordinea coloanelor, astfel încât să coincidă cu ordinea nouă fixată mai sus, deci cu ordinea liniilor. Se obține astfel matricea drumurilor triangularizată D' din tabelul 1.3.7.

Tabelul 1.3.7

(D')

	x_4	x_3	x_5	x_2	x_1
x_4	0	1	1	1	1
x_3	0	0	0	1	1
x_5	0	0	0	1	1
x_2	0	0	0	0	1
x_1	0	0	0	0	0

Observația 1.3.9. Un rezultat asemănător s-ar fi obținut, dacă în ordinea descrescătoare a puterilor de atingere, vârfurile s-ar fi așezat în succesiunea x_4, x_5, x_3, x_2, x_1 .

Teorema 1.3.2. În matricea drumurilor triangularizată, corespunzătoare unui graf fără circuite, primul element egal cu unu de pe fiecare linie și ultimul element egal cu unu de pe fiecare coloană, corespund la drumuri de câte un singur arc în vârf.

Pentru justificare, să considerăm $d_{ij}=1$, ca prim element egal cu unu, de pe linia x_i a matricei D' . Pe de o parte, acest lucru înseamnă că graful conține cel puțin un drum de la x_i la x_j . Să presupunem prin absurd că ar exista un astfel de drum cu mai mult de un arc, deci care ar trece printr-un vârf intermediar $x_m \neq x_i$. Reprezentăm acest drum prin $d = (x_i, \dots, x_m, \dots, x_j)$. Este clar de aici, că graful dat conține cel puțin un drum de la x_i la x_m , adică $d_{im}=1$; în plus, puterea de atingere a lui x_m este mai mare decât a lui x_j (orice vârf atins de la x_j este atins și de la x_m). În consecință, coloana x_m se găsește înaintea coloanei x_j în matricea D' , ceea ce înseamnă că elementul $d_{im}=1$ se află pe linia x_i , înaintea elementului $d_{ij}=1$. Această concluzie este însă absurdă prin ipoteză fiind precizat că $d_{ij}=1$ este primul element egal cu unu, de pe linia x_i . Ca urmare, un vârf intermediar, de forma x_m , nu poate exista în drumul de la x_i la x_j : drumul $d(x_i, x_j)$ - (care sigur există) nu poate fi format decât dintr-un singur arc, (x_i, x_j) . În mod asemănător se poate arăta că ultimul element egal cu unu, de pe o coloană x_j , a matricei D' , corespunde la un arc în graf.

1.4. Metode pentru determinarea componentelor tare conexe ale unui graf orientat

O componentă tare conexă, a unui graf G , este determinată atunci când se cunoaște mulțimea M , a vârfurilor sale. Într-adevăr, componenta reprezentând un subgraf al grafului

G , are drept arce toate arcele lui G , care au ambele extremități în mulțimea M , aceste arce putându-se deci determina cu ușurință dacă se cunoaște M . În cele ce urmează, căutăm o metodă pentru determinarea mulțimii M , a vârfurilor unei componente tare conexe, din care face parte un vârf fixat, $x_i \in X$. Notăm această mulțime prin $M(x_i)$, componenta tare conexă corespunzătoare fiind $C(x_i)$. Este evident că, utilizând aceste notații, dacă vârfurile x_i și x_j fac parte din aceeași componentă tare conexă, avem $C(x_i) = C(x_j)$, respectiv $M(x_i) = M(x_j)$. Vom folosi în mod deosebit proprietatea aproape evidentă, că orice două vârfuri din G , care sunt mutual atinse, fac parte din aceeași componentă tare conexă (o justificare riguroasă a afirmației este ușor de făcut).

În cele de mai jos, facem raționamentul de deducere a metodei cerute, pe baza unui exemplu, anume vom căuta componenta tare conexă ce conține vârfurile x_1 , pe graful G din figura 1.4.1. Această componentă, notată cu $C(x_1)$, va conține toate vârfurile grafului G , mutual atinse cu x_1 (conform proprietății menționate anterior) și numai pe acestea (conform definiției conexiunii tari). De aceea, vom determina separat următoarele două submulțimi de vârfuri ale grafului G :

- mulțimea vârfurilor atinse de la x_1 , pe care o vom nota cu V_1 .
- mulțimea vârfurilor ce ating pe x_1 , pe care o vom nota cu V'_1 .

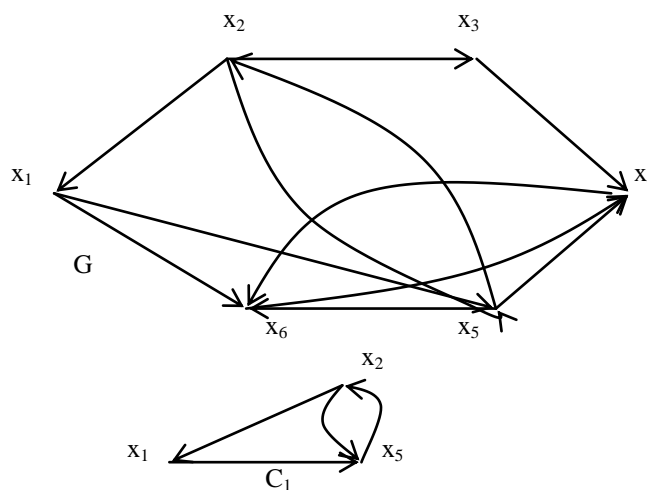


Fig. 1.4.1

Cu notațiile indicate, este clar că mulțimea vârfurilor mutual atinse cu x_1 va fi dată de intersecția $V_1 \cap V'_1$. Deoarece există și posibilitatea ca această intersecție să fie vidă, caz în care componenta tare conexă căutată este formată numai din vârfurile x_1 , vom scrie, în general, următoarea formulă de calculul a mulțimii vârfurilor componente $C(x_1)$:

$$(i) M(x_1) = V_1 \cap V'_1 \cup \{x_1\}$$

Arătăm acum în ce mod se calculează submulțimile de vârfuri V_1 și V'_1 . Este cunoscut faptul că vârfurile atinse de la x_1 , în graful G , sunt indicate prin elementele egale cu unu, de pe linia x_1 , a matricei drumurilor. În tabelul 1.4.1 este scrisă matricea arcelor grafului G , sub care este calculată linia x_1 , din matricea drumurilor (cu ajutorul metodei prezentate anterior).

Tabelul 1.4.1

	x_1	x_2	x_3	x_4	x_5	x_6	V_1
x_1	0	0	0	0	1	1	1
x_2	1	0	1	0	1	0	1
x_3	0	0	0	1	0	0	0
x_4	0	0	0	0	0	1	0
x_5	0	1	0	1	0	1	1
x_6	0	0	0	1	0	0	0
V_1	1	1	1	1	1	1	1

Această linie este notată cu V_1 , deoarece elementele egale cu unu, de pe ea, arată că vârfurile sunt atinse de la x_1 . Mai precis, deoarece există elemente egale cu unu, în dreptul tuturor coloanelor tabelului, putem scrie:

$$V_1 = \{x_1, x_2, x_3, x_4, x_5, x_6\}.$$

Pornind de la coloana x_1 a matricei A și efectuând operații anologice, doar pe coloane, obținem în dreapta tabelului o coloană suplimentară V_1 , ale cărei elemente egale cu unu indică vârfurile ce ating pe x_1 , în graful G . Aceste calcule sunt efectuate, de asemenea, în tabelul 1.4.8, decurgând conform următoarei succesiuni de operații:

- mai întâi s-a scris, pe coloana V_1 , elementul egal cu unu, din dreptul liniei x_2 , un astfel de element aflându-se inițial pe coloana x_1 ,
- datorită elementului egal cu unu menționat, s-a adunat boolean la coloana V_1 , coloana x_2 , această adunare aducând un nou element egal cu unu, în dreptul liniei x_5 .
- datorită noului element egal cu unu, apărut anterior, s-a adunat boolean la coloana V_1 , coloana x_2 , ceea ce a adus un nou element egal cu unu, în dreptul liniei x_1 (în dreptul liniei x_2 exista element egal cu unu, deci aici nu se modifică nimic),
- adunarea booleană a coloanei x_1 , la coloana V_1 , nu mai aduce modificări, ceea ce înseamnă că procesul descris mai sus a luat sfârșit,
- locurile rămase libere au fost completate cu zerouri.

Coloana V_1 , calculată cu ajutorul operațiilor descrise mai sus, este egală cu prima coloană a matricei drumurilor, ceea ce justifică afirmația că elementele egale cu unu, de pe ea, indică vârfurile ce ating pe x_1 . Într-adevăr, dacă în graful G s-ar schimba sensurile tuturor arcelor, s-ar obține un alt graf G' , a cărui matrice a arcelor este transpusă matricei A din tabelul 1.4.8. Datorită schimbării sensurilor, de pe arcele arborelui G , vârfurile atinse de la x_1 , în graful G' , vor fi chiar vârfurile ce ating pe x_1 , în graful G , deci acestea se determină calculând prima linie a matricei drumurilor grafului G' (cu operațiile cunoscute, dar aplicate pe transpusa matricei A).

Din cele de mai sus, rezultă că $V_1 = \{x_1, x_2, x_5\}$ și prin urmare:

$$M(x_1) = (V_1 \cap V_1) \cup \{x_1\} = \{x_1, x_2, x_5\} \cup \{x_1\} = \{x_1, x_2, x_5\}$$

Componenta $C(x_1)$, pe care o vom nota pe scurt cu C_1 (prima componentă tare conexă găsită), este reprezentată în figura 1.4.2. Această componentă a fost construită astfel:

- vârfurile ei sunt x_1, x_2 , și x_5 , cele ce formează mulțimea $M(x_1)$,
- arcele ei sunt toate arcele grafului G , ce au ambele extremități în mulțimea $M(x_1)$.

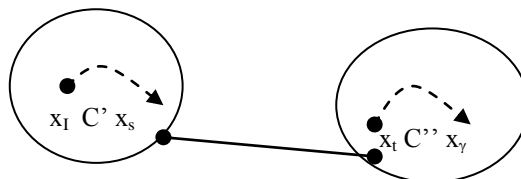


Fig.1.4.2

Pentru a determina următoarea componentă tare conexă a grafului G, vom repeta procedeul pe subgraful lui G, cu vârfurile rămase în discuție, adică cu mulțimea de vârfuri: $X - M(x_1) = \{x_3, x_4, x_6\}$. Acest lucru se poate realiza pe același tabel, al matricei arcelor grafului G, adăugând linia suplimentară V_2 și coloana suplimentară V_2 (pentru găsirea celei de a doua componente, pe care, o vom nota cu C_2), din tabel fiind șterse liniile și coloanele corespunzătoare vârfurilor x_1, x_2 și x_5 . Se ajunge astfel la tabelul 1.4.2, se preiau calculele din vechiul tabel 1.4.1 și le prelungește în sensul indicat.

Tabelul 1.4.2

	x_1	x_2	x_3	x_4	x_5	x_6	V_1	V_2	V_3
x_1	0	0	0	0	1	1	1	////	////
x_2	1	0	1	0	1	0	1	////	////
x_3	0	0	0	1	0	0	0	0	////
x_4	0	0	0	0	0	1	0	0	1
x_5	0	1	0	1	0	1	1	////	////
x_6	0	0	0	1	0	0	0	0	1
V_1	1	1	1	1	1	1	-	-	-
V_2	////	////	0	1	////	1	-	-	-
V_3	////	////	////	1	////	1	-	-	-

Pentru a nu strica claritatea tabelului, nu au fost șterse efectiv liniile și coloanele x_1, x_2 , și x_5 , ci pe linia V_2 și coloana V_2 au fost hașurate locurile corespunzătoare lor. Pentru calculul liniei V_2 și al coloanei V_2 , s-a pornit cu vârful x_3 al grafului G (x_1 și x_2 au intrat în componența C_1). Astfel, pentru linia V_2 , s-au făcut următoarele calcule:

- linia x_3 a adus un singur element egal cu unu, în dreptul coloanei x_4 ,
- adunarea booleană a liniei x_4 a adus un nou element egal cu unu, în dreptul coloanei x_6 ,
- adunarea booleană a liniei x_6 , nu a mai adus nici o modificare, locul rămas liber completându-se cu un zero.

Observația 1.4.1. Dacă prin adunarea booleană a unei linii ar trebui să fie adus un element egal cu unu pe un loc hașurat, nu ținem seama de el, coloana corespunzătoare din tabel fiind presupusă ștergută.

Din calculele efectuate, rezultă că: $V_2 = \{x_4, x_6\}$. Pe de altă parte, pentru calculul coloanei V_2 , de la început coloana x_3 nu aduce nici un element egal cu unu (cel din dreptul liniei x_2 cade într-o regiune hașurată), deci locurile libere se completează cu zerouri și $V_2=0$. Aplicând o formulă asemănătoare cu formula (i), avem:

$$(ii) M(x_3) = (V_2 \cap V_2) \cup \{x_3\} = \{x_3\}$$

În consecință, C_2 , cea de a doua componentă tare conexă a grafului G este formată dintr-un singur vârf, x_3 . În tabelul 1.4.2 este calculată și cea de a treia componentă tare conexă a lui G, prin eliminarea liniei și coloanei x_3 (pe lângă cele eliminate anterior). Această componentă, notată cu C_3 , are următoarea mulțime de vârfuri:

$$(iii) M(x_4) = (V_3 \cap V_3) \cup \{x_4\} = \{x_4, x_6\}$$

Observația 1.4.1. Într-un graf fără circuite, toate componentele tare conexe sunt formate din câte un singur vârf.

Calculul matricei arcelor grafului condensat. Să considerăm că graful G are k componente tare conexe, C_1, C_2, \dots, C_k . Matricea A^* , a grafului condensat G^* , este o matrice pătrată de ordinul k, ale cărei linii și coloane corespund la componentele C_1, C_2, \dots, C_k ,

considerate ca vârfuri. Această matrice se deduce din matricea A , a arcelor grafului G , conform următoarelor proprietăți ușor de dedus (cititorul le poate intui sau justifica, imediat):

- linia C_s , $s = 1, 2, \dots, k$, din matricea A^* , este suma booleană a liniilor din A , ce, corespund la vârfurile ce intră în componența componenteii tare conexe C_s ,

- coloana C_t , $t=1, 2, \dots, k$, din matricea A^* , este suma booleană, a colonelor din A , ce, corespund la vârfurile ce intră în componența componenteii tare conexe C_t ,

- matricea A^* are în mod obligatoriu, toate elementele de pe diagonala principală egale, cu zero (dacă apare vreun element egal cu unu, pe această diagonală el se înlocuiește cu unu).

Evident, în calculul practic al matricei A^* , matricea A se condensează - în modul indicat - mai întâi pe linii, apoi pe coloane (sau invers). Exemplul ce urmează va lămurii complet cele descrise mai sus.

Exemplul 1.4.1. Pentru graful din figura 1.4.1, a cărui matrice a arcelor, este cea din tabelul 1.4.1 și ale cărui componente au fost calculate în exemplul precedent, calculul matricei A se face în trei etape, după cum urmează. Mai întâi, se calculează, matricea dreptunghiulară A , din tabelul 1.4.3, ale cărei linii corespund la componentele tare conexe C_1, C_2 și C_3 , iar ale cărei coloane corespund la vârfurile grafului G . Linia C_1 a fost calculată ca suma booleană a liniilor x_1, x_2 și x_5 din matricea A (x_1, x_2 și x_5 fiind vârfurile ce intră în componența C_1). Linia C_2 coincide cu linia x_3 din matricea A , x_3 fiind singurul vârf din componența C_2 . În sfârșit, linia C_3 a fost calculată ca suma booleană a liniilor x_4 și x_6 din matricea A .

Tabelul 1.4.3.

	x_1	x_2	x_3	x_4	x_5	x_6
(\tilde{A}) C_1	1	0	1	1	0	1
C_2	0	0	0	1	0	0
C_3	0	0	0	0	1	1

În cea de a doua etapă, se procedează asemănător pe coloane, dar asupra matricei \tilde{A} , ceea ce conduce la matricea $\tilde{\tilde{A}}$, din tabelul 1.4.4., de data aceasta o matrice pătrată, ale cărei linii și coloane corespund la componentele C_1, C_2 și C_3 .

Tabelul 1.4.4.

	C_1	C_2	C_3
C_1	1	1	1
C_2	0	0	1
C_3	0	0	1

Astfel, coloana C_1 din matricea $\tilde{\tilde{A}}$ este suma booleană a coloanelor x_1, x_2 și x_5 , din matricea \tilde{A} , coloana C_2 din $\tilde{\tilde{A}}$ coincide cu coloana x_3 a matricei \tilde{A} , iar coloana C_3 din $\tilde{\tilde{A}}$ este suma booleană a coloanelor x_4 și x_6 din \tilde{A} .

În final, cea de a treia etapă constă în înlocuirea celor două elemente egale cu unu, de pe diagonala principală a matricei $\tilde{\tilde{A}}$, prin zero, ceea ce conduce la matricea arcelor grafului condensat G , din tabelul 1.4.5.

Tabelul 1.4.5

	C_1	C_2	C_3
C_1	0	1	1
C_2	0	0	1
C_3	0	0	0

Încheiem prezentarea noțiunilor generale despre grafuri, cu o proprietate a grafului condensat.

Teorema 1.4.1. Graful condensat, corespunzător unui graf orientat oarecare, este un graf fără circuite.

Justificarea acestei proprietăți rezultă din următoarea observație simplă: dacă în G^* , există arcul (C', C'') atunci în G există drum de la orice vârf $x_i \in C'$, spre orice vârf $x_j \in C''$. Într-adevăr, există în G^* , a arcului (C', C'') , arată, conform definiției grafului condensat, că G conține un arc (x_s, x_t) , cu x_s în componenta C' și x_t în componenta C'' . Dar în C' și C'' existând legături prin drumuri între orice vârfuri interne lor, există drumuri de forma $d(x_i, x_s)$ și $d(x_t, x_j)$, care se pot lega între ele prin intermediul arcului (x_s, x_t) , conducând la un drum $d(x_i, x_j)$ figura I.4.2.

Conform observației de mai sus, existența unui circuit în graful condensat ar însemna că două vârfuri, x_i și x_j , ce fac parte din două vârfuri distincte ale lui G^* , de pe acel circuit, sunt mutual atinse, ceea ce nu se poate (cele două componente, presupuse vârfuri distincte ale lui G^* , s-ar contopi într-una singură).

1.5. Aplicații

Aplicația 1.5.1. Cele cinci puncte de lucru importante dintr-o secție sunt legate printr-un sistem de benzi transportoare. Asociem situației din secție un model-graf $G=(V, E)$ în care vârfurile x_1, x_2, x_3, x_4 și x_5 reprezintă cele cinci puncte de lucru, iar arcul $(x_i, x_j) \in E$ reprezintă banda transportoare care duce piesele din x_i către x_j . Fie G graful din figura 1.5.1. Să se analizeze situația din secție cu ajutorul matricei arcelor A , indicând cum poate fi folosit un astfel de sistem de benzi, pentru un proces de fabricație ce trece prin toate cele cinci puncte.

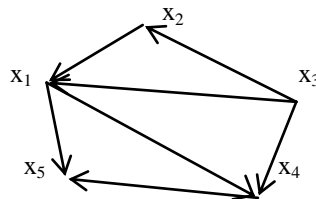


Fig.1.5.1

Rezolvare. Matricea arcelor grafului G este prezentată în tabelul 1.5.1. La această matrice s-au atașat o linie și o coloană suplimentare, în care s-au calculat respectiv sumele pe coloane și sumele pe linii.

Tabelul 1.5.1

	x_1	x_2	x_3	x_4	x_5	g_i^+
x_1	0	0	0	1	1	2
x_2	1	0	0	0	0	1
x_3	1	1	0	1	0	3
x_4	0	0	0	0	1	1
x_5	0	0	0	0	0	0
g_i^-	2	1	0	2	2	-

(A) În partea dreapta a matricei, se găsesc deci gradele exterioare de incidență ale vârfurilor, iar în partea de jos a aceleiași matrice se găsesc gradele interioare ale acestor vârfuri. Se observă că gradul interior de incidență a vârfului x_3 este nul ($g_3^- = 0$), ceea ce înseamnă că procesul de fabricație trebuie să pornească din punctul de lucru x_3 , (aici nu sosește nici o bandă transportoare). Pe de altă parte, gradul exterior de incidență a vârfului x_5 este nul ($g_5^+ = 0$) ceea ce înseamnă că procesul de fabricație trebuie să se termine în punctul de lucru x_5 . În plus, prin observații directe, se poate găsi că procesul de fabricație trebuie să străbată succesiv punctele x_3, x_2, x_1, x_4, x_5 .

Aplicația 1.5.2. Un anumit produs poate fi fabricat într-o secție a unei întreprinderi, prin mai multe procedee tehnologice, folosind unele materii prime și trecând prin diverse tipuri de produse intermediare (semifabricate). Fie $G=(V, E)$ un graf, în care $X=\{x_1, x_2, \dots, x_8\}$ este mulțimea materiilor prime ce se pot folosi și a produselor intermediare respective (numite împreună mai scurt stări de prelucrare), iar E mulțimea posibilităților de trecere de la o stare de prelucrare la alta. Mai precis, $(x_i, x_j) \in E$, dacă starea de prelucrare x_i (eventual împreună cu alte stări din mulțimea V) poate conduce, printr-o operație de prelucrare, la starea x_j . Fie graful G , cel din figura 1.5.2.

a) Să se cerceteze posibilitățile de trecere între diferitele stări de prelucrare indicate în problemă.

b) Să se arate la ce stări de prelucrare se poate ajunge, pornind de la fiecare dintre stările de prelucrare presupuse în problemă.

c) x_8 fiind starea finală (produsul finit), să se arate de la ce stări se poate ajunge la aceasta, conform condițiilor date de problemă.

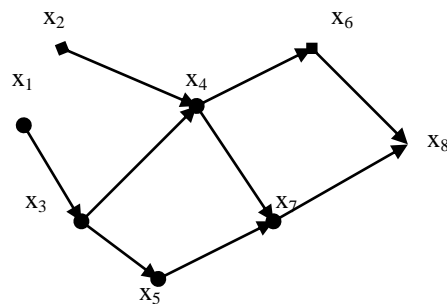


Fig.1.5.2

Rezolvare. a) Matricea arcelor grafului G este calculată în tabelul 1.5.2.

Tabelul 1.5.2

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
(A) x_1	0	0	1	0	0	0	0	0
x_2	0	0	0	1	0	0	0	0
x_3	0	0	0	1	1	0	0	0
x_4	0	0	0	0	0	1	1	0
x_5	0	0	0	0	0	0	1	0
x_6	0	0	0	0	0	0	0	1
x_7	0	0	0	0	0	0	0	1
x_8	0	0	0	0	0	0	0	0

Pornind de la această matrice și prin aplicarea metodei descrise în acest paragraf, s-a calculat matricea drumurilor din tabelul 1.5.3. Ea arată toate posibilitățile de trecere, între diferitele stări de prelucrare, indicate în problemă. Astfel, dacă elementul $d_{ij}, i=1, 2, \dots, 8; j=1, 2, \dots, 8$, este egal cu unu, atunci există cel puțin o posibilitate de trecere de la starea x_i la starea x_j . În caz contrar, dacă $d_{ij}=0$, nu se poate trece de la starea x_i la starea x_j . Astfel, de pildă, deoarece $d_{28}=1$, există posibilitatea trecerii de la starea x_2 la starea x_8 . Prin observații directe, cititorul poate găsi următoarele trei drumuri de la x_3 la x_8 ; $d_1=(x_3, x_5, x_7, x_8)$, $d_2=(x_3, x_4, x_6, x_8)$, $d_3=(x_3, x_4, x_7, x_8)$.

Tabelul 1.5.3

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	p_i
(D) x_1	0	0	1	1	1	1	1	1	6
x_2	0	0	0	1	0	1	1	1	4
x_3	0	0	0	1	1	1	1	1	5
x_4	0	0	0	0	0	1	1	1	3
x_5	0	0	0	0	0	0	1	1	2
x_6	0	0	0	0	0	0	0	1	1
x_7	0	0	0	0	0	0	0	1	1
x_8	0	0	0	0	0	0	0	0	0

b) Elementele egale cu unu din matricea D, ce se află pe o linie x_i , arată la ce stări se poate ajunge pornind de la starea x_i . În plus, puterile de atingere, calculate în partea dreaptă a matricei D, ca sume, ale elementelor de pe liniile respective, arată la câte stări se poate ajunge, pornind de la fiecare stare în parte.

c) Elementele egale cu unu, de pe coloana x_8 a matricei drumurilor, arată de la ce stări se poate ajunge la starea finală x_8 . După cum se observa din tabelul 1.5.3, la starea finală se poate ajunge de la orice stare de prelucrare, menționată în problemă.

Aplicația 1.5.3. Șapte puncte, de lucru ale unei secții, ce folosea curent electric și alternativ, pot fi alimentate prin racordarea la rețea, folosind unele din legăturile posibile, reprezentate prin muchii în graful neorientat G, din figura 1.5.3. Să se determine una dintre posibilitățile de legare a punctelor de lucru, astfel încât toate să poată funcționa și nefolosind legături de prisos. În condițiile puse de problemă, există tipuri de legare a punctelor de lucru, care să folosească mai puține sau mai multe legături între perechile de puncte?

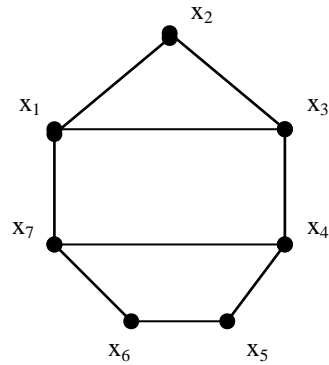


Fig.1.5.3

Rezolvare. Trebuie construit un arbore-acoperire al grafului G . Pentru aceasta, vom considera toate cele șapte vârfuri, x_1, x_2, \dots, x_7 , drept vârfuri ale arborelui căutat, dar dintre muchii selectăm numai $n - 1 = 7 - 1 = 6$, muchii, ca în exemplul ce urmează.

Să pornim, de pildă, cu muchia $[x_1, x_2]$. Următoarea muchie, pe care o solicităm, va avea una din extremități în x_1 sau în x_2 (pentru conexiune): fie să selectăm pe $[x_2, x_3]$. La a treia selecție, vom evita alegerea muchiei $[x_1, x_3]$, care ar închide un ciclu, să alegem, spre exemplu, muchia $[x_3, x_4]$.

Evitând, de fiecare dată, alegerea unei muchii ce închide un ciclu, să presupunem că au fost alese succesiv următoarele muchii, de la a patra selecție în sus: $[x_4, x_5]$, $[x_3, x_6]$, $[x_4, x_7]$. Se observă că, la subgraful obținut, nu se mai poate adăuga nici o muchie, fără, să se închidă un ciclu, deci că s-a obținut un arbore-acoperire (vezi figura 1.5.4). Aceasta este una dintre soluțiile posibile ale problemei. În practică, alegerea diferitelor muchii ce se adaugă, se face conform unor criterii economice, ce vor fi prezentate.

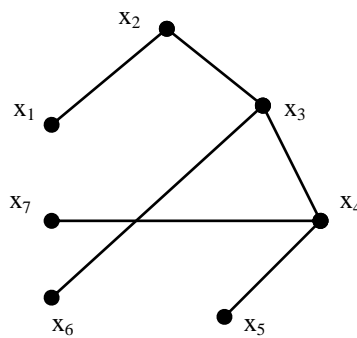


Fig.1.5.4

În ceea ce privește numărul de muchii, ce trebuie selectate, pentru a forma un arbore-acoperire al lui G , acesta este mereu același, anume egal cu $n - 1 = 6$.

CAPITOLUL 2

DRUMURI MINIME-MAXIME ÎN GRAFURI

2.1. Transport pe drumuri minime-maxime

În practica economică apar multe probleme ce se pot rezolva folosind modele grafuri în care urmează să determinăm anumite drumuri de valori minime (sau maxime). În acest sens, dăm, în cele ce urmează, un exemplu.

Exemplul 2.1.1. Să presupunem că pentru transportul unui produs, între două localități, se poate utiliza o rețea de șosele, în cadrul căreia există mai multe posibilități de alegere a rutei pe care urmează să se efectueze acest transport. Rețeaua în cauză este reprezentată printr-un graf ponderat, ale cărui vârfuri sunt localități intermediare prin care se poate trece, iar ale cărui arce (muchii) sunt șoselele ce leagă localitățile respective, Ponderile (valorile) de pe arce pot reprezenta costurile unitare de transport de-a lungul acestor șosele sau alți indicatori legați de folosirea șoselelor. Evident, probleme asemănătoare de transport se pot pune și rezolva în cadrul unor sisteme de dimensiuni mai mici, ca spre exemplu în rezolvarea unor probleme de transport intern al unei întreprinderi, secții etc.

Observația 2.1.1. Și de această dată putem avea situații când rezolvăm prin metode asemănătoare (cu ajutorul determinării drumurilor de valori minime în grafuri) probleme ce nu sunt de transport, dar ale căror modele matematice coincid cu ale acestora.

În continuare vom arăta în ce constă problema determinării drumurilor de valori minime (respectiv maxime) într-un graf. Astfel, fiind dat un graf ponderat, $G=(V, E, w)$, având vârfurile x_1, x_2, \dots, x_n , să considerăm două vârfuri fixate ale sale, de exemplu x_1 și x_n , (x_1 este localitatea de plecare, iar x_n , este localitatea de destinație a transportului). Graful G conține mai multe drumuri de la x_1 la x_n , (cazul când există un singur drum de acest fel este banal). Să notăm aceste drumuri prin: $d_1(x_1, x_n), d_2(x_1, x_n), d_3(x_1, x_n), \dots, d_k(x_1, x_n)$. Fiecărui drum, dintre acestea, îi corespunde câte o valoare, egală cu suma valorilor arcelor ce îl compun (conform definiției date în paragraful precedent). Problema cercetată, cere ca, dintre drumurile menționate, să ne oprim la acela (sau acelea) cu valoarea cea mai mică. Notând un astfel de drum prin $d_s(x_1, x_n)$, putem scrie deci definiția lui astfel:

$$w[d_s(x_1, x_n)] = \min_{i=1,2,\dots,k} \{w[d_i(x_1, x_n)]\}.$$

Analog, dintre cele k drumuri de la x_1 la x_n , unul (sau mai multe) are valoarea cea mai mare. Notând un drum de valoare maximă de la x_1 la x_n , prin $d_t(x_1, x_n)$, definiția lui poate fi scrisă astfel:

$$w[d_t(x_1, x_n)] = \max_{i=1,2,\dots,k} \{w[d_i(x_1, x_n)]\}.$$

Observația 2.1.2. Vom nota, prin convenție, valoarea minimă drumurilor de la x_1 la x_n , cu m_{1n} , iar valoarea maximă a acestor drumuri cu M_{1n} . În general, considerând două vârfuri oarecare ale grafului, $x_i \in X$ și $x_j \in X$, convenim să notăm:

m_{ij} = valoarea minimă a drumurilor de la x_i la x_j

M_{ij} = valoarea maxima a drumurilor de la x_i la x_j .

Observația 2.1.3. Uneori, vom mai numi valoarea minimă (maximă) a drumurilor dintre x_i și x_j , distanța minimă și maximă între cele două vârfuri.

Exemplul 2.1.2. Fie graful ponderat din figura 2.1.1, valorile arcelor fiind scrise pe arce. Ne propunem să găsim, prin enumerare (cercetarea directă a tuturor posibilităților), drumul (drumurile) de valoare minimă, respectiv maximă, între vârfurile x_1 și x_6 .

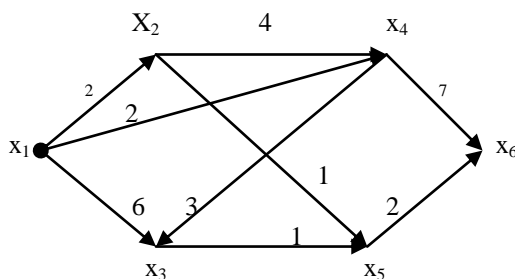


Fig. 2.1.1

Se poate observa cu ușurință, cercetând graful din figura 2.1.1, că există în acest graf șase (drumuri de la x_1 la x_6 , și anume următoarele:

$$d_1(x_1, x_6) = (x_1, x_2, x_4, x_6); \quad d_2(x_1, x_6) = (x_1, x_2, x_4, x_3, x_5, x_6);$$

$$d_3(x_1, x_6) = (x_1, x_2, x_5, x_6); \quad d_4(x_1, x_6) = (x_1, x_3, x_5, x_6);$$

$$d_5(x_1, x_6) = (x_1, x_4, x_6); \quad d_6(x_1, x_6) = (x_1, x_4, x_3, x_5, x_6).$$

Valorile acestor șase drumuri se calculează după cum urmează:

$$w[d_1(x_1, x_6)] = w(x_1, x_2) + w(x_2, x_4) + w(x_4, x_6) = 2 + 4 + 7 = 13,$$

$$w[d_2(x_1, x_6)] = w(x_1, x_2) + w(x_2, x_4) + w(x_4, x_3) + w(x_3, x_5) + w(x_5, x_6) = 2 + 4 + 3 + 1 + 2 = 12,$$

$$w[d_3(x_1, x_6)] = w(x_1, x_2) + w(x_2, x_5) + w(x_5, x_6) = 2 + 4 + 2 = 8,$$

$$w[d_4(x_1, x_6)] = w(x_1, x_3) + w(x_3, x_5) + w(x_5, x_6) = 6 + 1 + 2 = 9,$$

$$w[d_5(x_1, x_6)] = w(x_1, x_4) + w(x_4, x_6) = 2 + 7 = 9,$$

$$w[d_6(x_1, x_6)] = w(x_1, x_4) + w(x_4, x_3) + w(x_3, x_5) + w(x_5, x_6) = 2 + 3 + 1 + 2 = 8.$$

Valoarea minimă a drumurilor de la x_1 la x_6 , este deci:

$$m_{16} = \min\{13; 12; 8; 9; 9; 8\} = 8,$$

această valoare fiind atinsă pentru două drumuri dintre cele șase. Există deci două drumuri de valoare minimă (ambele cu valoarea egală cu opt) și anume $d_3(x_1, x_6)$ și $d_6(x_1, x_6)$. Dacă nu am fi scris și numerotat toate drumurile de la x_1 , la x_6 , aceste două drumuri ar fi putut fi scrise - prin convenție și astfel:

$$d^{\text{p}}_{\min}(x_1, x_6) = (x_1, x_2, x_5, x_6); \quad d^{\text{n}}_{\min}(x_1, x_6) = (x_1, x_4, x_3, x_5, x_6).$$

Analog, valoarea maximă a drumurilor de la x_1 la x_6 este:

$$M_{16} = \max\{13; 12; 8; 9; 9; 8\} = 13,$$

ea fiind atinsă numai pentru drumul $d_1(x_1, x_6)$, care este deci singurul drum de valoare maximă de la x_1 la x_6 . Putem scrie:

$$d_{\max}(x_1, x_6) = (x_1, x_2, x_4, x_6)$$

Observația 2.1.4. Drumurile de valoare minimă (respectiv maximă) dintr-un graf vor fi denumite pe scurt **drumuri minime respectiv maxime** în graf. Se mai folosește o denumire unitară, de **drumuri optime**, natura optimului (maxim sau minim) rezultând din contextul economic al problemei rezolvate. În același sens, vom folosi și expresia de **valoare optimă a drumurilor** dintre două vârfuri ale unui graf.

Observația 2.1.5. Trebuie să facem distincție între două noțiuni și anume:

- valoarea optimă a drumurilor dintre două vârfuri ale unui graf, care este un număr;
- drumul (drumurile) de valoare optimă între cele două vârfuri, care reprezintă succesiunea (succesiunile) corespunzătoare de vârfuri.

În sensul acestor două aspecte ale problemei, va trebui să dăm metode de rezolvare pentru fiecare în parte.

Înainte de a începe rezolvarea problemelor puse, vom accentua asupra a două proprietăți importante, pe care le prezentăm sub forma a două observații, ținând seama de faptul că cele mai multe dintre grafurile ponderate cu care lucrăm au valori pozitive și finite pe arce.

Proprietatea 2.1.1 Într-un graf cu valori finite și pozitive pe arce, există drumuri de valori minime finite între oricare două vârfuri ale grafului, ce sunt legate prin cel puțin un drum. În plus, orice drum de valoare minimă între două vârfuri, ale unui astfel de graf, este elementar.

Prima parte a acestei proprietăți este evidentă, din moment ce între cele două vârfuri există cel puțin un drum, a cărui valoare este sigur finită și pozitivă. Partea a doua a afirmației făcute va fi justificată pe baza unui exemplu simplificat și anume asupra grafului cu circuite din figura 2.1.2, (pe un graf fără circuite afirmația este evidentă, în astfel de grafuri existând numai drumuri elementare). Astfel, fie drumul neelementar $d_1(x_1, x_5) = (x_1, x_2, x_3, x_4, x_2, x_5)$, de valoare egală cu 16; el este neelementar, trecând de două ori prin vârful x_2 . Se observă cu ușurință că un astfel de drum (neelementar) trebuie să conțină un subdrum (o parte din drum), sub forma unui circuit, în cazul nostru (x_2, x_3, x_4, x_2) . Eliminând din drumul $d_1(x_1, x_5)$ circuitul menționat se obține cu ușurință un alt drum de la x_1 la x_5 , drum de forma $d_2(x_1, x_5) = (x_1, x_2, x_5)$ a cărui valoare este mai mică decât valoarea lui $d_1(x_1, x_5)$, cu cinci unități (valoarea circuitului eliminat din drumul inițial). Modificările indicate ajută cu atât mai mult când drumul de la care se pornește trece de mai multe ori printr-un vârf, caz în care repetăm de câte ori se poate eliminarea unui circuit din drumul respectiv. Raționamentul descris ne convinge că un drum neelementar nu poate avea valoare minimă, existând totdeauna un alt drum, de valoare mai mică decât el, între aceleași vârfuri ale grafului.

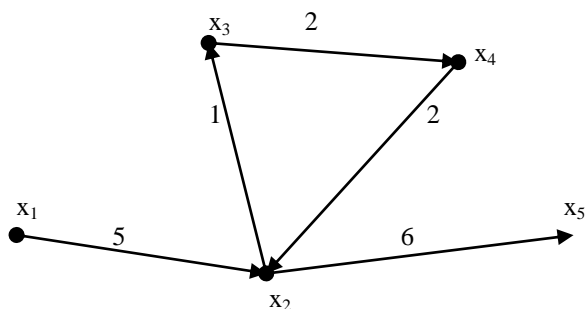


Fig. 2.1.2

Reținem, din justificarea făcută mai sus, o proprietate mai generală, ce rezultă cu ușurință și anume că:

Observația 2.1.6. Orice metodă de căutare a drumurilor de valori minime, între vârfurile unui graf cu circuite evită de la sine introducerea în aceste drumuri a circuitelor cu valori pozitive.

Proprietatea 2.1.2. Într-un graf cu circuite, în care există circuite cu valori pozitive, există vârfuri între care putem obține drumuri cu valori oricât de mari. În acest sens, problema determinării drumurilor de valoare maximă, între vârfurile respective, își pierde sensul. Dăm ca exemplu graful din figura 2.1.2, ce conține circuitul (x_2, x_3, x_4, x_2) , de valoare egală cu cinci unități și în care putem scrie următorul șir de drumuri de la x_1 la x_5 cu valori din ce în ce mai mari:

$$d_1(x_1, x_5) = (x_1, x_2, x_5), \text{ de valoare egală cu } 11,$$

$$d_2(x_1, x_5) = (x_1, x_2, x_3, x_4, x_2, x_5), \text{ de valoare egală cu } 16,$$

$$d_3(x_1, x_5) = (x_1, x_2, x_3, x_4, x_2, x_3, x_4, x_2, x_5), \text{ de valoare egală cu } 21,$$

.....
 Este clar că putem repeta circuitul (x_2, x_3, x_4, x_2) , în drumul cerut, de câte ori este nevoie, pentru a obține un drum de valoare oricât de mare; acesta este echivalent cu a spune că, deși vârfurile x_1 și x_5 , sunt legate prin drumuri, nu există un drum cu valoarea maximă între aceste vârfuri.

Observația 2.1.7. Într-un graf cu circuite pozitive nu are rost să căutăm drumurile de valori maxime între diferitele vârfuri ale grafului.

Observația 2.1.8. Dacă valorile atașate arcelor grafului pot fi și negative, probleme analoge se pot pune în cazul când graful conține circuite de valori negative, dar cu următoarele rezultate:

- drumurile de valori maxime, între diferitele vârfuri ale grafului, există întotdeauna (evident dacă vârfurile respective sunt legate prin drumuri și nu există circuite cu valori pozitive); aceste drumuri de valori maxime evită includerea în ele a circuitelor negative, fiind deci elementare,

- drumurile de valori minime nu există pentru anumite perechi de vârfuri, deși vârfurile respective sunt legate prin perechi de drumuri, adică problema pusă își pierde sensul (există drumuri cu valori oricât de mici, evident negative).

În paragraful de față, ne vom referi numai la grafuri ponderate, ale căror valori de pe arce sunt pozitive. De asemenea, vom trata numai cazul drumurilor cu valori minime, indicând de fiecare dată cum se adaptează metodele descrise, în situația când se cer drumurile cu valori maxime.

Problemele practice, care conduc la modelele-grafuri pe care le folosim, pot duce la două tipuri de probleme de determinare a drumurilor cu valori minime, dintre care, prima reprezintă un caz particular al celeilalte și anume:

(i) Determinarea valorii minime, respectiv a drumului (drumurilor) cu valoarea minimă, între două vârfuri fixate ale grafului.

(ii) Determinarea valorilor minime, respectiv a drumurilor cu valori minime, între oricare două vârfuri ale grafului.

Rezolvarea uneia, sau celeilalte probleme, dintre cele două de mai sus, depinde evident de problema practică ce se rezolvă, de rezultatele ce se cer. Evident, problema (i) se va rezolva cu calcule mai puține, dar problema (ii) dă rezultate mai cuprinzătoare.

În ceea ce privește rezolvarea problemei (ii), ea se realizează cu ajutorul unei matrici pătrate de ordinul n ($n =$ numărul vârfurilor din graf), pe care o notăm de obicei cu M și o numim **matricea valorilor minime** ale drumurilor dintre oricare două vârfuri ale grafului. Mai precis, notând cu m_{ij} elementul matricii M , ce se află la intersecția liniei x_i cu valoarea x_j , avem:

$$m_{ij} = \begin{cases} 0, & \text{daca } i = j \text{ (pentru elementele de pe diagonala principala)} \\ \infty, & \text{daca } i \neq j \text{ și nu exista nici un drum în graf, de la } x_i \text{ la } x_j \\ \text{valoarea minima a drumurilor de la } x_i \text{ la } x_j, & \text{daca există astfel de drumuri} \end{cases}$$

Observația 2.1.9. Pentru elementul m_{ij} , linia în care se găsește el arată plecările de drumuri minime, iar coloana arată sosirile acestor drumuri minime.

Pentru problema analoagă, a determinării matricii valorilor maxime ale drumurilor din graf, vom nota ou M această matrice, iar cu M_{ij} elementul de pe linia x_i și de pe coloana x_j a acestei matrice, definit astfel:

$$M_{ij} = \begin{cases} 0, & \text{daca } i = j, \\ -\infty, & \text{dacă nu exista nici un drum în graf, de la } x_i \text{ la } x_j \\ \text{valoarea maxima a drumurilor de la } x_i \text{ la } x_j & \text{în restul cazurilor} \end{cases}$$

Metodele de rezolvare a problemelor (i) și (ii) folosesc așa-numita **matrice a valorilor**, notată cu W . Această matrice este o matrice pătrată de ordinul n , al cărei element w_{ij} , de pe linia x_i și de pe coloana x_j , se definește astfel:

$$w_{ij} = \begin{cases} 0, & \text{daca } i = j \text{ (pentru elementele de pe diagonala principală)} \\ \pm \infty, & \text{dacă } i \neq j \text{ și nu există arcul } (x_i, x_j) \text{ în graf,} \\ w(x_i, x_j), & \text{dacă graful conține arcul } (x_i, x_j). \end{cases}$$

Observația 2.1.10. În cazul când $i \neq j$ și graful nu conține arcul (x_i, x_j) , se pune $w_{ij} = \infty$, în cazul problemei de drumuri minime și $w_{ij} = -\infty$, în cazul problemei de drumuri maxime.

Observația 2.1.1.1. Pentru cazul unui graf neorientat, notațiile de mai sus rămân valabile, dar transformând mai întâi acest graf într-unul orientat, prin înlocuirea fiecărei muchii $[x_i, x_j]$ cu două arce de sens contrar, (x_i, x_j) și (x_j, x_i) , cu aceeași valoare (egală cu valoarea muchiei corespunzătoare).

Exemplul 2.1.3. Fie graful orientat G , din figura 2.1.3, valorile arcelor fiind scrise pe arce. Matricea valorilor acestui graf, scrisă în scopul rezolvării, problemei de drumuri minime, este cea din tabelul 2.1.1. Se observă că toate elementele de pe diagonala principală sunt egale cu zero, în locurile corespunzătoare arcelor existente în graf se găsesc valorile acestor arce, iar celelalte elemente sunt infinite.

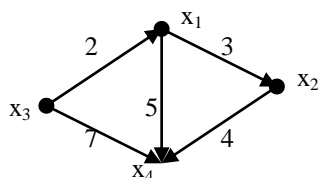


Fig. 2.1.3

De exemplu: $w_{12} = 3$, deoarece graful conține arcul (x_1, x_2) , cu valoarea trei, dar $w_{21} = \infty$, deoarece graful nu conține arcul (x_2, x_1) . Dacă matricea valorilor ar fi fost scrisă în scopul determinării valorilor drumurilor maxime, ea ar fi diferit de cea din tabelul 2.1.1 doar prin faptul că în locul elementelor egale cu ∞ , ar fi fost $-\infty$.

Tabelul 2.1.1

	x_1	x_2	x_3	x_4
x_1	0	3	∞	7
x_2	∞	0	3	5
x_3	∞	∞	0	4
x_4	∞	∞	∞	0

Prin cercetare directă, ne putem convinge că matricele valorilor minime, respectiv maxime, ale drumurilor (între oricare două vârfuri ale grafului G , sunt cele din tabelele 2.1.2 și 2.1.3.

Tabelul 2.1.2.

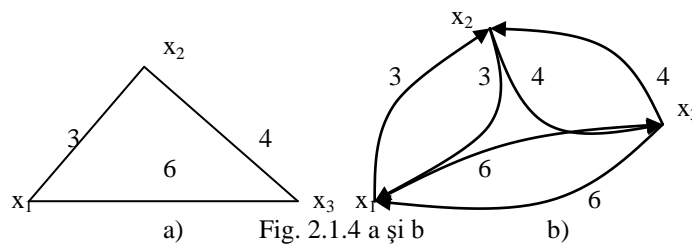
	x_1	x_2	x_3	x_4
x_1	0	2	5	7
x_2	∞	0	3	5
x_3	∞	∞	0	4
x_4	∞	∞	∞	0

Tabelul 2.1.3.

	x_1	x_2	x_3	x_4
x_1	0	2	5	9
x_2	$-\infty$	0	3	7
x_3	$-\infty$	$-\infty$	0	4
x_4	$-\infty$	$-\infty$	$-\infty$	0

(M)

Exemplul 2.1.4. Graful neorientat din figura 2.1.4 a, poate fi înlocuit cu graful orientat din figura 2.1.4 b, a cărei matrice a valorilor este cea din tabelul 2.1.4. Evident, în graful dat se poate pune numai problema determinării valorilor minime ale drumurilor dintre oricare două vârfuri.



Tabelul 2.1.4

	x_1	x_2	x_3
x_1	0	3	6
x_2	3	0	4
x_3	6	4	0

(W)

Deoarece problemele (i) și (ii) nu se pot rezolva cu ușurință prin cercetare directă, la grafuri de dimensiuni mari, este nevoie să folosim unii algoritmi în acest sens, algoritmi ce vor fi prezentați în cele ce urmează. La baza acestora stă *principiul de optimalitate al lui Bellman*, care este un principiu general în matematică și care aplicat la problema noastră se enunță astfel:

“Orice drum optim în graf este format din subdrumuri optime din acel graf”.

Vom ilustra aplicarea acestui principiu în cadrul descrierii fiecărui algoritm de drum optim în parte.

În cele ce urmează, prezint doi dintre algoritmi de tipul (i). Ei vor da, pe lângă valoarea minimă a drumurilor dintre cele două vârfuri specificate și alte valori minime analoage, ce nu se cer în problemă.

2.2. Algoritmi pentru determinarea drumurilor minime

2.2.1. Algoritmul Bellman-Kalaba

Fie $G=(V, E, w)$, unde $V=\{x_1, x_2, \dots, x_n\}$, un graf orientat. Interesează găsirea valorilor minime ale tuturor drumurilor de la fiecare vârf al grafului, x_i ($i=1, 2, \dots, n$), la un

anumit vârf fixat, fie acesta x_n (dacă vârfurile fixate sunt altul decât x_n , îl trecem pe ultimul loc). Pornim cu matricea $W = \|w_{ij}\|$.

Fie $\overline{m_{in}^{(k)}}$ valoarea minimă a drumurilor formate din cel mult k arce, între x_i și x_n și fie m_{in} valoarea minimă a tuturor drumurilor de la x_i la x_n , indiferent de numărul de arce.

Teorema 2.2.1.1. Fie $G = (V, E, w)$, $V = \{x_1, x_2, \dots, x_n\}$ un graf orientat ponderat și x_n un vârf fixat. Atunci:

$$\overline{m_{in}^{(k+1)}} = \min_{\{j: j \neq i\}} \{w_{ij} + \overline{m_{jn}^{(k)}}\} \text{ pentru } i \neq n.$$

Demonstrație. Evident, orice drum de valoare minimă de la x_i la x_n , format din cel mult $k + 1$ arce trebuie să fie format dintr-un arc (x_i, x_j) , $i \neq j$ și un drum de la x_j la x_n , de cel mult k arce, care trebuie să aibă valoarea minimă față de toate celelalte drumuri formate din k arce de la x_j la x_n (conform principiului de optimalitate al lui Bellman).

Teorema 2.2.1.2. Fie $G = (V, E, W)$, $V = \{x_1, x_2, \dots, x_n\}$ un graf orientat ponderat și x_n , un vârf fixat. Dacă $(\exists) k \in \mathbb{N}$, astfel încât oricare ar fi vârfurile x_i , să avem:

$$\overline{m_{in}^{(k)}} = \overline{m_{in}^{(k+1)}}, \text{ atunci } \overline{m_{in}^{(k)}} = m_{in}, (\forall) x_i \in X.$$

Demonstrație. Dacă vârfurile x_i și x_n sunt fixate și două numere naturale $k, p \in \mathbb{N}$ sunt în relația $k \geq p$, atunci evident $\overline{m_{in}^{(k)}} \geq \overline{m_{in}^{(p)}}$. Trebuie arătat că, în condițiile teoremei: $\overline{m_{in}^{(k)}} = \overline{m_{in}^{(k+s)}}$, $s \in \mathbb{N}$. Acest lucru se arată prin inducție matematică, după valorile numărului natural s .

Verificarea, este imediată pentru $s = 1$, fiind chiar relația din enunț.

În etapa de demonstrare, presupunem adevărată relația $\overline{m_{in}^{(k)}} = \overline{m_{in}^{(k+s)}}$, pentru un s oarecare și trebuie să arătăm că este adevărată relația:

$$\overline{m_{in}^{(k)}} = \overline{m_{in}^{(k+s+1)}}.$$

Dar, aplicând rezultatul teoremei 2.2.1.1, avem:

$$\overline{m_{in}^{(k+s+1)}} = \min_{\{j: j \neq i\}} \{w_{ij} + \overline{m_{jn}^{(k+s)}}\} = \min_{\{j: j \neq i\}} \{w_{ij} + \overline{m_{jn}^{(k)}}\} = \overline{m_{in}^{(k+1)}} = \overline{m_{in}^{(k)}}, \text{ pentru } x_i \neq x_n.$$

În acest șir de egalități am aplicat, de asemenea, ipoteza de inducție și condiția dată în enunțul teoremei.

Dacă $x_i = x_n$, proprietatea este evidentă.

Pe aceste teoreme se bazează algoritmul **Bellman Kalaba**, conținând următoarele etape:

Pas 1. Se construiește matricea valorilor W , corespunzând grafului ponderat dat.

Pas 2. Se atașează noi linii în partea de jos a matricei W , notate $\overline{m_{in}^{(1)}}$, $\overline{m_{in}^{(2)}}$, ... care dau valorile minime ale drumurilor formate din 1, 2, ..., arce de la oricare vârf x_i la vârfurile x_n , fixate și anume:

a) Linia $\overline{m_{in}^{(1)}}$ conține valorile minime ale drumurilor de la oricare vârf x_i la x_n , formate dintr-un singur arc deci conține valorile arcelor incidente interiorului x_n . Aceste valori vor fi date, evident, de coloana lui x_n , din matricea W .

b) Presupunând completată o linie oarecare $\overline{m_{in}^{(k)}}$ se trece la calculul liniei $\overline{m_{in}^{(k+1)}}$, pe baza relației dată de teorema 2.2.1.1, pentru fiecare element în parte pentru $i \neq n$, dar evident adevărată și în cazul poziției diagonale când unul din arce este considerat degenerat (de valoare zero).

Evident $\overline{m_{in}^{(k)}} = 0$, pentru orice $k \in \mathbb{N}$

c) Atașarea liniilor continuă pentru $k \leq n - 1$ până când $\overline{m_{in}^{(k)}} = \overline{m_{in}^{(k+1)}}$, și deci, conform teoremei 2.2.1.2, s-au obținut aici tocmai valorile minime căutate, indiferent de numărul de arce străbătute.

Pas. 3. Se determină efectiv vârfurile și arcele din drumul minim de la x_i la vârful fixat x_n , astfel: adunăm element cu element linia x_i din W cu linia $\overline{m_{in}^{(k+1)}}$, căutând cea mai mică valoare a sumei. Presupunem că $w_{i1} + \overline{m_{i1n}^{(k+1)}} = \overline{m_{in}^{(k+1)}}$. Atunci, evident, arcul (x_i, x_{i1}) este primul arc din drumul minim căutat. Când suma minimă se obține în dreptul mai multor coloane, atunci există mai multe drumuri corespunzând valorii minime și se urmărește până la capăt fiecare variantă în parte.

Analog se adună linia x_i cu linia $\overline{m_{in}^{(k+1)}}$ căutând arcul care asociat unui drum minim de la vârful atins la x_n , să dea valoarea $\overline{m_{i1n}^{(k+1)}}$ găsită anterior. Se determină astfel al doilea arc (x_{i1}, x_{i2}) din succesiunea căutată, procedând la fel până se află un arc cu vârful x_n , extremitate terminală. Rezultă drumul de valoare minimă, unic sau nu, de forma $(x_i, x_{i1}, x_{i2}, \dots, x_{ip}, x_n)$.

Observația 2.2.1.1. Algoritmul se poate aplica și în cazul determinării valorilor maxime ale drumurilor, înlocuind peste tot minimul prin maximum și luând în considerare matricea W corespunzătoare (eu $-\infty$ în loc de $+\infty$).

Exemplul 2.2.1.1. Fie graful orientat din figura 2.2.1.1, arcele fiind căi de transport între diferitele puncte, iar valorile, scrise pe arce reprezentând costurile unitare de transport de-a lungul arcelor corespunzătoare se cere drumul pe care să se facă transportul, între x_1 și x_6 , încât cheltuielile să fie minime. Trebuie aflată deci, mai întâi, valoarea minimă a drumurilor de la x_1 la x_6 , precum și drumul (drumurile) ce realizează această valoare minimă.

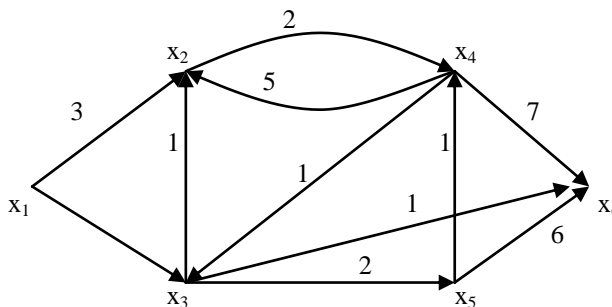


Fig. 2.2.1.1.

Matricea valorilor, W , a grafului dat, este scrisă în tabelul 2.2.1.1. Sub acest tabel, în continuarea matricei W , sunt calculate cinci linii suplimentare, ce sunt explicate parțial în continuare.

Tabelul 2.2.1.1.

	x_1	x_2	x_3	x_4	x_5	x_6
x_1	0	3	7	∞	∞	∞
x_2	∞	0	∞	2	∞	∞
x_3	∞	1	0	∞	2	1
x_4	∞	3	1	0	∞	7
x_5	∞	∞	∞	1	0	6
x_6	∞	∞	∞	∞	∞	0
$m_{i6}^{(1)}$	∞	∞	1	7	6	0

$m_{i6}^{(2)}$	8	9	1	2	6	0
$m_{i6}^{(3)}$	8	4	1	2	3	0
$m_{i6}^{(4)}$	7	4	1	2	3	0
$m_{i6}^{(5)}$	7	4	1	2	3	0

Linia notată cu $m_{i6}^{(1)}$ corespunde cu coloana x_6 a matricei W , fiind însă transpusă.

Elementele liniei $m_{i6}^{(1)}$ au fost calculate succesiv astfel:

$$m_{16}^{(2)} = \min\{0+\infty; 3+\infty; 7+1; \infty+7; \infty+6; \infty+0\}=8,$$

$$m_{26}^{(2)} = \min\{\infty+\infty; 0+\infty; \infty+1; 2+7; \infty+6; \infty+0\}=9,$$

$$m_{36}^{(2)} = \min\{\infty+\infty; 1+\infty; 0+1; \infty+7; 2+6; 1+0\}=1,$$

$$m_{46}^{(2)} = \min\{\infty+\infty; 3+\infty; 1+1; 0+7; \infty+6; 7+0\}=2,$$

$$m_{56}^{(2)} = \min\{\infty+\infty; \infty+\infty; \infty+1; 1+7; 0+6; 6+0\}=6,$$

$$m_{66}^{(2)} = \min\{\infty+\infty; \infty+\infty; \infty+1; \infty+7; \infty+6; 0+0\}=0,$$

Asemănător, au fost calculate celelalte linii suplimentare. Deoarece liniile $m_{i6}^{(4)}$ și $m_{i6}^{(5)}$ sunt identice, calculul acesta s-a oprit la linia $m_{i6}^{(5)}$; primul element al acestei linii, încadrată în tabelul 2.2.1.1, reprezintă valoarea minimă a drumurilor de la x_1 la x_6 : $m_{16} = 7$.

Pentru a determina drumul (drumurile) ce realizează această valoare, procedăm astfel:

a) Printre sumele elementelor de pe linia x_1 , cu elementele corespunzătoare ale liniei $m_{i6}^{(5)}$, cea mai mică este egală cu 7 și se atinge pentru valoarea $w_{12} = 3$; înseamnă că primul arc al drumului căutat este (x_1, x_2) .

b) Printre sumele dintre elementele corespunzătoare ale liniilor x_2 și $m_{i6}^{(5)}$, cea mai mică este egală cu 4 și se atinge pentru valoarea $w_{24}=2$; înseamnă că al doilea arc al drumului minim este (x_2, x_4) .

c) Printre sumele elementelor corespunzătoare ale liniilor x_4 și $m_{i6}^{(5)}$, cea mai mică este egală cu 2, atinsă pentru $w_{43}=1$; cel de al treilea arc al drumului minim este deci (x_4, x_3)

d) Printre sumele elementelor corespunzătoare ale liniilor x_3 și $m_{i6}^{(5)}$, cea mai mică este egală cu unu și este atinsă pentru $w_{36}=1$; ultimul arc al drumului minim este deci (x_3, x_6) .

Din cele de mai sus, rezultă că drumul de valoare minimă, de la x_1 la x_6 , este unic și anume: $d = (x_1, x_2, x_4, x_3, x_6)$, având o valoare de șapte unități. Pe acest drum, urmează să se organizeze transportul de la x_1 la x_6 .

2.2.2. Algoritmii lui Yen

Pentru calculul valorilor minime de la un vârf al grafului fixat, fie acesta x_1 , la celelalte vârfuri, x_2, x_3, \dots, x_n , din G , adică $m_{12}, m_{13}, \dots, m_{1n}$, se poate proceda și astfel:

Pasul 1. Se scrie matricea valorilor W și se determină cea mai mică dintre valori, care pleacă din x_1 . Fie:

$$w_{1i0} = \min_{i=2, \dots, n} \{w_{1i}\}, \text{ deci } w_{1i0} = m_{1i0}.$$

Pasul 2. Toate elementele din prima linie a matricei W , exceptând $w_{11}=0$ și w_{1i_0} , se transformă astfel:

$$w_{1i} = \min_{\substack{i \neq 1 \\ i \neq i_0}} \{ w_{1i}; w_{1i_0} + w_{i_0i} \}.$$

Din matricea astfel obținută, se suprimă linia de rang i_0 și coloana de rang i_0 , păstrând pentru celelalte linii și coloane, rangurile lor inițiale. Dacă matricea obținută este de ordinul 2, atunci w_{12} din această matrice reprezintă valoarea minimă de la x_1 la x_k , unde k este rangul coloanei a doua (liniei a doua) din această matrice și algoritmul ia sfârșit. Dacă matricea obținută are cel puțin ordinul trei, atunci se repetă pași 1^o și 2^o asupra acestei matrici, ca și asupra matricei W , cu prima linie a acestei matrici.

Fiecare etapă de aplicare a algoritmului în cei doi pași scade ordinul matricei cu câte o unitate și dă la fiecare etapă câte o valoare minimă m_{1i} ($i=1, 2, 3, \dots, n$).

Justificarea algoritmului. În primul pas al primei, etape de aplicare a algoritmului $w_{1i_0} = \min_{i \geq 2} \{ w_{1i} \} = m_{1i_0}$, deoarece, orice drum de valoare minimă de la x_1 la x_{i_0} trebuie să înceapă cu un arc incident exterior lui x_1 , deci valoarea sa este cel puțin egală cu valoarea celui mai scurt arc de forma (x_1, x_{i_0}) care este în particular un drum de la x_1 la x_{i_0} .

După, o etapă a algoritmului, se obține o matrice corespunzând unui subgraf $G_1=(V_1, E_1)$ al lui G , unde $V_1=V \setminus \{x_{i_0}\}$. Vom arăta cum se determină valorile minime de la x_1 la x_i (cu $i \neq i_0$) în graful G . Fie un vârf oarecare, $x_i \neq x_{i_0}$. Există două posibilități, pe care le cercetăm în continuare.

Cazul 1. În graful G există un drum de la x_1 la x_i care nu trece prin x_{i_0} , fie aceasta $d=(x_1, x_{j1}, x_{j2}, \dots, x_i)$ un drum minim. Rezultă că $w_{1j1} \leq w_{1i_0} + w_{1j1}$, deoarece în caz contrar am putea înlocui arcul (x_1, x_{j1}) printr-un drum format din două arce, (x_1, x_{i_0}) și (x_{i_0}, x_{j1}) și care este mai scurt, contrazicând ipoteza. Prin înlocuirile de la pasul 2^o, elementul w_{1j1} , din W rămâne neschimbat, deci valoarea drumului $(x_1, x_{j1}, x_{j2}, \dots, x_i)$ este aceeași în G și G_1 . În graful G_1 însă, nu pot exista distanțe mai mici de la x_1 la x_i , cu $i \neq i_0$, deoarece oricărui drum $(x_1, x_{j1}, \dots, x_{js}, x_i)$ din G_1 îi corespunde în G același drum, dacă $w_{1j1} \leq w_{1i_0} + w_{i_0j1}$, sau îi corespunde drumul $(x_1, x_{i_0}, x_{j1}, \dots, x_{js}, x_i)$ din G dacă $w_{1j1} \leq w_{1i_0} + w_{i_0j1}$ cu valoare egală. Deci distanțele minime se conservă în G și G_1 .

Cazul 2. În graful G , orice drum minim de la x_1 la x_i trece prin x_{i_0} . Fie un astfel de drum: $d(x_1, x_i) = (x_1, x_{j1}, x_{j2}, \dots, x_{i_0}, x_{k1}, x_{k2}, \dots, x_i)$.

Deoarece valoarea arcului (x_1, x_{i_0}) este mai mică sau egală cu valoarea arcului (x_1, x_{j1}) , rezultă că și drumul:

$$d^1(x_1, x_i) = (x_1, x_{i_0}, x_{k1}, x_{k2}, \dots, x_i)$$

este un drum de valoare minimă de la x_1 la x_i în G , în care x_{i_0} nu mai apare încă odată între $x_{k1}, x_{k2}, \dots, x_i$. Există evident relația: $w_{1k1} > w_{1i_0} + w_{i_0k1}$, deoarece în caz contrar și drumul $d''(x_1, x_i) = (x_1, x_{k1}, x_{k2}, \dots, x_i)$ este drum de valoare minimă în G de la x_1 la x_i și nu trece prin x_{i_0} , contrazicând ipoteza.

În pasul 2 al algoritmului Yen s-a înlocuit w_{1k1} prin: $w_{1i_0} + w_{i_0k1}$, deci valoarea drumului d^1 în G este egală cu valoarea drumului d'' în G_1 , deci se conservă distanțele minime și în acest caz.

Exemplul 2.2.2.1. Fie graful orientat din figura 2.2.2.1, ce reprezintă posibilitățile de transport al unui produs, între punctele x_1, x_2, x_3, x_4 și x_5 . Valorile de pe arce reprezentând costurile transportului pe arcele respective, să se determine drumul de la x_1 la x_5 , încât costul total de transport să fie minim.

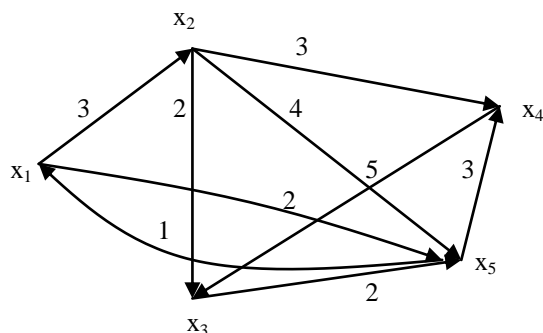


Fig. 2.2.2.1.

Matricea valorilor grafului este cea din tabelul 2.2.2.1. Avem: $m_{11}=0$ și $\min_{i \geq 2} \{w_{1i}\} = \{3, \infty, \infty, 2\}=2$, deci $x_{i0}=x_5$ și $m_{15}=2$, iar drumul este $d(x_1, x_5)=(x_1, x_5)$.

Tabelul 2.2.2.1

	x_1	x_2	x_3	x_4	x_5
x_1	0	3	∞	∞	2
x_2	∞	0	2	3	4
x_3	∞	∞	0	0	2
x_4	∞	∞	5	0	∞
x_5	1	∞	∞	3	0

Apoi, se suprimă din tabelul 2.2.2.1. linia și coloana corespunzând vârfului x_5 , ceea ce conduce la tabelul 2.2.2.2, în care s-au înlocuit elementele din prima linie ca la pasul 2⁰ al algoritmului:

Tabelul 2.2.2.2.

	x_1	x_2	x_3	x_4
x_1	0	3	∞	5
x_2	∞	0	2	3
x_3	∞	∞	0	∞
x_4	∞	∞	5	0

$$\tilde{w}_{12} = \min\{3; 2 + \infty\} = 3$$

$$\tilde{w}_{13} = \min\{\infty; 2 + \infty\} = \infty$$

$$\tilde{w}_{14} = \min\{\infty; 2 + 3\} = 5$$

Din prima linie a tabelului 2.2.2.2., se alege: $\min_{i \geq 2} \{w_{1i}\} = \min\{3, \infty, 5\}=3$, care corespunde vârfului x_2 . Avem deci $m_{12}=3$ și $d(x_1, x_2)=(x_1, x_2)$.

Acum se suprimă linia și coloana corespunzătoare lui x_2 și modificăm elementele din prima linie a matricei din tabelul 2.2.2.2., cu formula de la pasul 2⁰ a algoritmului:

$$\approx w_{13} = \min\{\infty; 3 + 2\} = 5,$$

$$\approx w_{14} = \min\{5; 3 + 3\} = 5,$$

Obținem tabelul 2.2.2.3, în care avem: $m_{13}=m_{14}=5$ și $d(x_1, x_3)=(x_1, x_2, x_3)$; $d(x_1, x_4)=(x_1, x_5, x_4)$.

Tabelul 2.2.2.3.

	x_1	x_3	x_4
x_1	0	5	5
x_3	∞	0	∞
x_4	∞	5	0

În general, la acest algoritm găsirea drumului minim se poate face așa cum se explică mai jos. Drumul cel mai scurt, de la x_i la x_j are lungimea m_{ij} , astfel că: $m_{ij} = m_{ik} + w_{kj}$, dacă (x_k, x_j) este ultimul arc al drumului minim. Pentru distanța minimă m_{ik} de la x_i la x_k se procedează analog pentru găsirea ultimului arc al drumului. Dacă nu există nici un indice k cu proprietatea $k \neq i, k \neq j$ care să verifice egalitatea: $m_{ij} = m_{ik} + w_{kj}$, atunci singurul drum minim de la x_i la x_j este arcul: (x_i, x_j) .

Observația 2.2.2.1. Pentru aflarea drumului de valoare maximă, se folosește inițial matricea W corespunzătoare și se alege de fiecare dată valoarea maximă de pe linia tabelului rămas.

În continuare, prezint trei algoritmi ce rezolvă problemele (ii), deci care conduc la calcularea matricei M , a valorilor minime ale drumurilor dintre oricare două vârfuri ale grafului.

2.2.3. Algoritmul Roy - Floyd

Fie $G = (V, E, w)$, unde $X = \{x_1, x_2, \dots, x_n\}$.

Algoritmul Roy-Floyd determină atât valorile distanțelor minime între oricare două vârfuri ale grafului, cât și drumurile minime corespunzătoare și se bazează pe următoarea teoremă:

Teorema 2.2.3.1. Dacă, într-un graf oarecare G , este îndeplinită inegalitatea: $w_{ij} \leq w_{ik} + w_{kj}$, pentru orice $i \neq j, i \neq k, j \neq k$, atunci valoarea minimă a drumurilor de la x_i la x_j este valoarea w_{ij} .

Demonstrație. Presupunem că există cel puțin un drum de valoare finită de la x_i la x_j , fie acesta:

$d(x_i, x_j) = (x_i, x_{k1}, x_{k2}, \dots, x_{k8}, x_j)$ și din ipoteza teoremei se deduce din aproape în aproape că:

$$\begin{aligned} w_{ik2} &\leq w_{ik1} + w_{k1k2}, \\ w_{ik3} &\leq w_{ik2} + w_{k2k3} \leq w_{ik1} + w_{k1k2} + w_{k2k3} \\ &\dots \\ w_{ij} &\leq w_{ik1} + w_{k1k2} + \dots + w_{ksj}. \end{aligned}$$

de unde justetea concluziei teoremei.

Algoritmul pentru aflarea matricei M , a valorilor minime a drumurilor, are etapele următoare:

Pasul 1. Se asociază grafului matricea $W = (w_{ij}), i, j = 1, 2, \dots, n$ și matricea $C \parallel C_{ij}$, $i, j = 1, 2, \dots, n$, unde C_{ij} sunt mulțimi de noduri sau vârfuri din care pleacă arce, deci:

$$C_{ij} = \begin{cases} \{x^i\}, & \text{dac } w_{ij} \text{ finit} \\ \emptyset, & \text{dac } w_{ij} = \infty. \end{cases}$$

Pasul 2. Ținând seama de afirmația teoremei, se determină matricile $W_k = \| w_{ij}^{(k)} \|$ $i, j=1, 2, \dots, n, k=1, 2, \dots, n$ în care se modifică valorile w_{ij} care nu satisfac teorema și deci se pot micșora, astfel:

$$w_{ij}^{(1)} = \begin{cases} w_{ij}, & \text{dac } i \neq j \text{ și } i = 1, \text{ sau } j = 1, \\ 0, & \text{dac } i = j, \\ \min\{ w_{ij}, w_{i1} + w_{1j} \}, & \text{dac } i \neq j, i \neq 1; j \neq 1. \end{cases}$$

În general, pentru $k=2, 3, \dots, n$, punem.

$$w_{ij}^{(k)} = \begin{cases} 0, & \text{dac } i = j, \\ w_{ij}^{(k-1)}, & \text{dac } i \neq j; i = k \text{ sau } j = k \\ \min\{ w_{ij}^{(k-1)}; w_{ik}^{(k-1)} + w_{kj}^{(k-1)} \}, & \text{dac } i \neq j, i \neq k; j \neq k. \end{cases}$$

Analog se află matricile $C_k, k=1, 2, \dots, n$. Mai precis, elementele matricei $C_k = \| C_{ij}^{(k)} \|, i, j = 1, 2, \dots, n$, se află astfel:

$$C_{ij}^{(k)} = \begin{cases} C_{ij}^{(k-1)}, & \text{daca } w_{ij}^{(k-1)} < w_{ik}^{(k-1)} + w_{kj}^{(k-1)}, \\ C_{ij}^{(k-1)} \cup C_{kj}^{(k-1)}, & \text{daca } w_{ij}^{(k-1)} = w_{ik}^{(k-1)} + w_{kj}^{(k-1)}, \\ C_{kj}^{(k-1)}, & \text{daca } w_{ij}^{(k-1)} > w_{ik}^{(k-1)} + w_{kj}^{(k-1)}, \end{cases}$$

cu convenția: $C_{ij}^{(0)} = C_{ij}, w_{ij}^{(0)} = w_{ij}$

În final, $W_n = M$, iar C_n dă efectiv drumurile minime, ținând cont de mulțimile vârfurilor vecine cu fiecare vârf x_j , date de elementele $C_{ij}^{(n)}$, reconstituind drumul în sens invers.

Exemplul 2.2.3.1. Fie graful din figura 2.2.2.1 folosit și în exemplul precedent. Trebuie să găsim costurile minime de transport, între oricare două puncte ale grafului. Cu alte cuvinte, trebuie să se determine matricea valorilor minime dintre vârfurile grafului și drumurile corespunzătoare acestor trasee optime. Matricile W și C , din primul pas al algoritmului, se găsesc în tabelele 2.2.3.1 și 2.2.3.1'.

Tabelul 2.2.3.1

0	3	∞	∞	2
∞	0	2	3	4
∞	∞	0	∞	2
∞	∞	5	0	∞
1	∞	∞	3	0

Tabelul 2.2.3.1'

C:

x_1	x_1			x_1
	x_2	x_2	x_2	x_2
		x_3		x_3
		x_4	x_4	
x_5			x_5	x_5

Aplicăm pasul 2 al algoritmului pe rând, pentru $k = 1, 2, \dots, n$. Avem succesiv rezultatele conținute în tabelele 2.2.3.2 -2.2.3.6'

Tabelul 2.2.3.2

W_1 :

0	3	∞	∞	2
∞	0	2	3	4
∞	∞	0	∞	2
∞	∞	5	0	∞
1	4	∞	3	0

Tabelul 2.2.3.2'

C_1 :

x_1	x_2			x_1
	x_2	x_2	x_2	x_2
		x_3		x_3
		x_4	x_4	
x_5	x_1		x_5	x_5

Tabelul 2.2.3.3

0	3	5	6	2
∞	0	2	3	4
∞	∞	0	∞	2
∞	∞	5	0	∞
1	4	6	3	0

$W_2:$

Tabelul 2.2.3.3'

x_1	x_1	x_2	x_2	x_1
	x_2	x_2	x_2	x_2
		x_3		x_3
		x_4	x_4	
x_5	x_1	x_2	x_5	x_5

$C_2:$

Tabelul 2.2.3.4

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

$W_3:$

Tabelul 2.2.3.4'

x_1	x_1	x_2	x_2	x_1
	x_2	x_2	x_2	x_2, x_3
		x_3		x_3
		x_4	x_4	x_3
x_5	x_1	x_2	x_5	x_5

$C_3:$

Tabelul 2.2.3.5

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

$W_4:$

Tabelul 2.2.3.5'

x_1	x_1	x_2	x_2	x_1
	x_2	x_2	x_2	x_2, x_3
		x_3		x_3
		x_4	x_4	x_3
x_5	x_1	x_2	x_5	x_5

$C_4:$

Tabelul 2.2.3.6

0	3	5	5	2
5	0	2	3	4
3	6	0	5	2
8	11	5	0	7
1	4	6	3	0

$w_5:$

Tabelul 2.2.3.6'

x_1	x_1	x_2	x_2	x_1
x_5	x_2	x_2	x_2	x_2, x_3
x_5	x_5	x_3	x_5	x_3
x_5	x_5	x_4	x_4	x_3
x_5	x_1	x_2	x_5	x_5

$C_5:$

Să alegem câteva distanțe minime din tabelul Tabelul 2.2.3.6. - 2.2.3.6'

a) $m_{14} = 5$ și $d(x_1, x_4) = (x_1, x_5, x_4)$, citit pe prima linie din C_5 , reconstituind drumul de la sfârșit astfel: vârful x_4 este precedat de x_5 aflat în căsuța (1, 4), vârful x_5 este precedat de x_1 aflat în căsuța (1, 5), care este vârful inițial al drumului.

b) $m_{25} = 4$ și $d(x_2, x_5) = (x_2, x_5)$, sau $d(x_2, x_5) = (x_2, x_3, x_5)$, deci două drumuri cu aceeași lungime minimă 4, aflate pe linia lui x_2 , din tabelul 2.2.3.6, ca mai sus.

c) $m_{42} = 11$ $d(x_4, x_2) = (x_4, x_3, x_5, x_1, x_2)$.

Drumurile aflate sunt cele pe care trebuie organizat transportul, între vârfurile corespunzătoare.

2.2.4. Algoritmul lui Dantzig

Algoritmul lui Dantzig este un procedeu de calcul recursiv pentru găsirea distanțelor minime în subgrafurile cu 2, 3, ..., n vârfuri. În cazul în care funcția $w: E \rightarrow \mathbb{R}$ poate lua și valori negative, algoritmul poate detecta și eventualele circuite negative, a căror semnificație

este că distanța între oricare două vârfuri poate fi făcută oricât de mică parcurgând circuitele respective de un număr de ori suficient de mare, deci nu există o cea mai mică distanță finită între vârfurile circuitelor.

Fie graful ponderat $G = (V, E, w)$ și $G^{(k)} = (V_k, E_k, w_k)$ subgraful lui G în care mulțimea vârfurilor este: $V_k = \{x_1, x_2, \dots, x_k\}$, $k \leq n$. Fie, de asemenea, $M^{(k)} = \|m_{ij}^{(k)}\|$, $i, j = 1, 2, \dots, k$, matricea distanțelor minime în subgraful $G^{(k)}$.

Pasul 1. Se face $k = 2$ și $m_{12}^{(2)} = w_{12}$; $m_{21}^{(2)} = w_{21}$. Dacă $w_{12} + w_{21} < 0$ ne oprim, s-a detectat circuitul negativ (x_1, x_2, x_1) și problema distanțelor minime în G își pierde sensul. În caz contrar, pentru $w_{12} + w_{21} \geq 0$ se ia $m_{11}^{(2)} = m_{22}^{(2)} = 0$ și se trece la pasul 2^0 , succesiv pentru $k = 3, 4, \dots, n$.

Pasul 2. Cunoscând matricea $M^{(k-1)}$, se află elementele matricei $M^{(k)}$, prin recurență, astfel:

$$a) m_{ik}^{(k)} = \min_{j=1, 2, \dots, k-1} \{m_{ij}^{(k-1)} + w_{jk}\}, \text{ pentru } i=1, 2, \dots, k-1.$$

$$b) m_{kj}^{(k)} = \min_{i=1, 2, \dots, k-1} \{m_{ij}^{(k-1)} + w_{ki}\}, \text{ pentru } j=1, 2, \dots, k-1.$$

Se verifică dacă există un indice i , anume $1 \leq i \leq k - 1$, pentru care: $m_{ik}^{(k)} + m_{ki}^{(k)} < 0$ și în acest caz ne oprim, existând un circuit negativ în subgraful cu mulțimea de vârfuri V_k care trece prin vârfurile x_i și x_k . În caz contrar, elementele de pe diagonală sunt egale cu zero și se calculează elementele:

$$c) m_{ij}^{(k)} = \min\{m_{ij}^{(k-1)}; m_{ik}^{(k)} + m_{kj}^{(k)}\}, \text{ pentru } i \neq j; i, j = 1, 2, \dots, k-1$$

La sfârșitul algoritmului, elementele $m_{ij}^{(n)}$ dau distanțele minime în G , unde $m_{ii}^{(n)} = 0$, oricare ar fi $i = 1, 2, \dots, n$ și $k = 2, 3, \dots, n$. Dacă graful nu conține arce cu lungimea negativă, nu se mai verifică la pasul 2^0 existența circuitelor negative.

Justificarea algoritmului. Prin inducție după k , arătăm că valorile $m_{ik}^{(k)}$ și $m_{kj}^{(k)}$ date de formulele a), b), c) dau distanțele minime în subgraurile cu mulțimile de vârfuri $V_2, V_3, \dots, V_n \equiv V$ (pentru $k=2, 3, \dots, n$, respectiv).

Pentru $k=2$, subgraful $G^{(2)}$ are numai două vârfuri, $V_2 = \{x_1, x_2\}$ și distanțele minime între ele coincid cu distanțele directe (prin arce). Presupunând că nu există circuite negative, facem ipoteza că formulele sunt adevărate până la $k - 1$ și demonstrăm valabilitatea lor pentru $G^{(k)}$. Distanța minimă între un vârf x_i ($i = 1, 2, \dots, k - 1$) și x_k , în G este fie distanța directă w_{ik} , obținută din formula a) $j = i$, fie valoarea unui drum minim $d(x_i, x_k) = (x_i, x_{k1}, \dots, x_{ks}, x_{j0}, x_k)$, format cu vârfuri din V_k . Dar $(x_i, x_{k1}, \dots, x_{ks}, x_{j0})$ este un drum minim în $G^{(k-1)}$, a cărui lungime este $m_{ij0}^{(k-1)}$ conform ipotezei de inducție, deoarece, în caz contrar, acest drum s-ar putea înlocui printr-un drum mai scurt de la x_i la x_j în $G^{(k-1)}$ și care prelungit cu arcul (x_{j0}, x_k) ar genera un drum mai scurt decât $(x_i, x_{k1}, \dots, x_{ks}, x_{j0}, x_k)$, contrazicând ipoteza. Deci $m_{ik}^{(k)} = m_{ij0}^{(k-1)} + w_{j0k} = j = \min_{1, 2, \dots, k-1} \{m_{ij}^{(k-1)} + w_{ij}\}$ adică formulele a) sunt adevărate pentru orice $i = 1, 2, \dots, k-1$.

Dacă ar există un indice i ($1 \leq i \leq k - 1$) pentru care $m_{ik}^{(k)} + m_{ki}^{(k)} < 0$, atunci ar exista un circuit negativ care trece prin vârful x_k în $G^{(k)}$, deci și în graful G .

Valabilitatea formulelor b) se face în mod asemănător. Pentru formulele c), considerăm un drum de valoare minimă de la x_i la x_j în subgraful $G^{(k)}$. Dacă acest drum nu trece prin vârful x_k , atunci el este un drum minim în subgraful $G^{(k-1)}$ și are valoarea $m_{ij}^{(k-1)}$ conform ipotezei de inducție. Dacă drumul trece prin x_k , atunci el se compune din două subdrumuri, unul de la x_i la x_k , iar celălalt de la x_k la x_j , minimalitatea sa impunând

condiția ca ambele subdrumuri să fie drumuri minime în subgraful G^k (principiul lui Bellman).

Deci, valoarea drumului minim de la x_i la x_j este în acest caz $m_{ik}^{(k)} + m_{kj}^{(k)}$, unde $m_{ik}^{(k)}$ și $m_{kj}^{(k)}$ sunt date de formulele a) și b).

În concluzie, formula c) este adevărată pentru orice $i, j=1, 2, \dots, k$.

Exemplul 2.2.4.1. Reluând exemplul rezolvat anterior prin algoritmul lui Roy-Floyd, algoritmul lui Dantzig dă matricile drumurilor minime în subgrafurile $G^{(2)}, G^{(3)}, G^{(4)}, G^{(5)} \subset G$, ce se găsesc în tabelele 2.2.4.1 – 2.2.4.4. Subgrafurile însele sunt reprezentate în figura 2.2.4.1 a, b, c, respectiv în figura 2.2.2.1 ($G^{(5)}$).

	x_1	x_2
$M^{(2)}$	0	3
	x_2	∞ 0

	x_1	x_2	x_3
$M^{(3)}$	0	3	5
	x_2	∞ 0	2
	x_3	∞ ∞ 0	

	x_1	x_2	x_3	x_4
$M^{(4)}$	0	3	5	6
	x_2	∞ 0	2	3
	x_3	∞ ∞ 0	∞	∞
	x_4	∞ ∞ 5	0	

	x_1	x_2	x_3	x_4	x_5
$M^{(5)}$	0	3	5	5	2
	x_2	5	0	2	3
	x_3	3	6	0	5
	x_4	8	11	5	0
	x_5	1	4	∞	∞

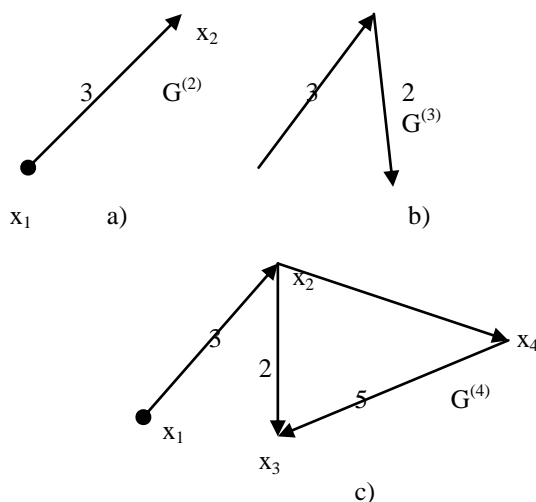


Fig.2.2.4.1

Găsirea drumurilor minime se poate face așa cum urmează. Pentru $m_{ij}^{(n)} = m_{ij}$ se ia linia lui x_i din W , care dă valorile directe de la x_i la alte vârfuri și se adună cu coloana x_j din $M^{(n)} \equiv M$, care dă valorile minime de la diferite vârfuri până la x_i și se caută sumele de valoare m_{ij} . Se află un prim arc al drumului căutat, (x_i, x_1) și se lucrează analog cu linia x_i din W și coloana x_j din M căutând suma m_{ij} pentru cel de al doilea arc al drumului ș.a.m.d. În exemplul dat, avem:

$$m_{42} = 11 \text{ și } d(x_4, x_2) = (x_4, x_3, x_5, x_1, x_2).$$

2.3. Arbori

2.3.1. Definiții și elemente de bază

În clasa grafurilor conexe, arborii reprezintă grafurile cele mai simple (ca structură) și, de asemenea, cele mai frecvent utilizate în practică. De studiul lor s-au ocupat matematicieni și fizicieni de seamă: Cayley a studiat arborii pentru aplicațiile lor în chimia organică, iar Kirchhoff, a studiat această categorie de grafuri pornind de la studiul rețelelor electrice.

Termenul de arbore a fost introdus de Cayley în 1857, plecând de la o analogie botanică.

Definiția 2.3.1.1. Se numește **arbore** un graf conex și fără cicluri.

Exemplul 2.3.1.1. În figura 2.3.1.1. este desenat un arbore. Se observă că oricum am elimina o muchie, graful își pierde proprietatea de conexitate, și oriunde am adăuga o muchie, apare un ciclu. Acest lucru este valabil în orice arbore.

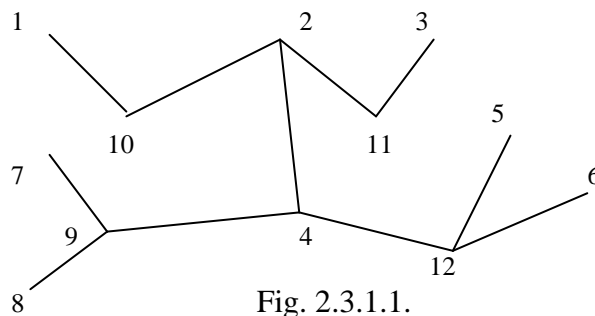


Fig. 2.3.1.1.

Teorema 2.3.1.1. Fie un graf $G=(V, E)$. Următoarele afirmații sunt echivalente:

- 1) G este arbore.
- 2) G este un graf conex, minimal cu aceasta proprietate (eliminând o muchie oarecare se obține un graf neconex).
- 3) G este un graf fără cicluri, maximal cu aceasta proprietate (dacă se adaugă o muchie se obține un graf care are măcar un ciclu).

Demonstrație. Ca să demonstrăm teorema enunțată mai sus, este suficient să parcurgem următoarele implicații: 1) \Rightarrow 2), 2) \Rightarrow 1), 1) \Rightarrow 3) și 3) \Rightarrow 1)

Într-adevăr, după ce vom demonstra ce ne-am propus, vom putea scrie:

$$2) \Leftrightarrow 1) \Leftrightarrow 3).$$

și, deci, prin tranzitivitatea relației de echivalență, obținem și 2) \Leftrightarrow 3).

1) \Rightarrow 2) *ipoteză:* G conex și fără cicluri

concluzia: G conex și minimal cu această proprietate

Se observă că proprietatea de conexitate este comună ipotezei și concluziei. Presupunem prin reducere la absurd că în E există o muchie $e = [x, y]$ prin a cărei eliminare să se obțină un graf G_1 care este totuși conex. Rezultă din conexitatea lui G_1 că în acest graf există un lanț între oricare două vârfuri, deci și între x și y , fie acesta L_{xy} . Din acesta putem obține un lanț elementar L_{xy} . Dacă adăugăm muchia eliminată, ne situăm din nou în G și, în

plus, se obține un ciclu, ceea ce este absurd. Urmează că G este într-adevăr minimal cu proprietatea de conexitate.

2) \Rightarrow 1) *Ipoteză:* G conex și minimal cu aceasta proprietate

Concluzia: G conex și fără cicluri

Din nou proprietatea de conexitate este comună ipotezei și concluziei. Mai rămâne să demonstrăm că G este fără cicluri. Presupunem prin reducere la absurd că în G există un ciclu $C = [x_1, x_2, \dots, x_k, x_1]$. Eliminăm muchia $[x_1, x_2]$ și argumentăm că se obține un graf G_1 , tot conex. Într-adevăr, pentru că G este conex, între orice două vârfuri există un lanț. În lanțurile care folosesc muchia, $[x_1, x_2]$ (doar acestea sunt afectate de eliminarea muchiei), vom înlocui această muchie cu lanțul $[x_2, x_3, \dots, x_k, x_1]$. Urmează deci că G_1 este într-adevăr conex, ceea ce vine în contradicție cu ipoteza. Rezultă că presupunerea făcută a fost, falsă și deci G este fără cicluri.

1) \Rightarrow 3) *Ipoteză:* G conex și fără cicluri

Concluzia: G fără cicluri și maximal cu această proprietate.

Proprietatea lui G de a fi fără cicluri este comună ipotezei și concluziei. Deoarece G este conex, rezultă că între oricare două vârfuri distincte și neadiacente x și y există un lanț. Din el construim un lanț elementar care să lege x și y . Dacă am adăuga muchia $[x, y]$ atunci ar apărea un ciclu, deci G este fără cicluri, maximal.

3) \Rightarrow 1) *Ipoteză:* G fără cicluri și maximal cu această proprietate

Concluzia: G conex și fără cicluri.

Din nou ipoteza și concluzia au o parte comună. Mai rămâne să demonstrăm conexitatea lui G . Fie x și y două vârfuri neadiacente în G . Dacă am adăuga muchia $[x, y]$, am obține, conform ipotezei, un graf care conține un ciclu C (în care apar, evident, vârfurile x și y), $C = [x, y, t, \dots, x]$. Eliminând muchia $[x, y]$, ne plasăm din nou în G , și, în plus, din ciclul C putem evidenția un lanț $L = [x, t, \dots, x]$ care leagă cele două vârfuri x și y . În concluzie, graful G este conex. Demonstrația teoremei este acum terminată.

Definiția 2.3.1.2. Fie G un graf. Un graf parțial H al său care în plus este și arbore se numește **arbore parțial**.

Corolarul 2.3.1.1. Un graf $G=(V, E)$ conține un arbore parțial, dacă și numai dacă G este conex.

Demonstrație:

I. Presupunem că G conține un arbore parțial $H=(V, T)$. Atunci, cum H este conex (fiind arbore), pentru G el este graf parțial. Rezultă că prin adăugarea la H a muchiilor din $E-T$ pentru a reconstitui pe G din H , proprietatea de conexitate se păstrează. Deci G este conex.

II. Dacă G este conex atunci există două cazuri:

1. G este arbore, caz în care $H=G$.

2. G nu este arbore. Rezultă că el conține cicluri, deci există în G măcar un ciclu C . Fie o muchie $e=[x, y]$ a acestui ciclu. Prin eliminarea muchiei e , proprietatea de conexitate se păstrează. Se obține astfel un graf G_1 . Dacă G_1 este arbore, atunci $H=G_1$. Dacă nu, procedăm similar cu G_1 . După un număr finit de pași (la fiecare pas se elimină o muchie), obținem arborele parțial căutat H .

Propoziția 2.3.1.1. Orice arbore $H = (V, E)$ cu $n \geq 2$ vârfuri conține cel puțin două vârfuri terminale.

Demonstrație: Presupunem prin absurd, că există un arbore H cu $n \geq 2$ vârfuri și care conține cel mult un vârf terminal, adică un vârf al cărui grad este 1. Alegem în H un lanț elementar de lungime maximă (care conține numărul maxim de muchii posibil); fie acesta $L=[x_1, x_2, \dots, x_k]$. Rezultă că cel mult o extremitate este de gradul 1, deci cel puțin una din extremități are gradul cel puțin 2; fie aceasta x_1 . Mai există în H măcar un vârf în afară de x_2 cu care x_1 este adiacent; fie z unul din ele. Dacă z nu ar aparține lanțului L , am deduce că L

ar mai putea, fi prelungit cu măcar o muchie, contradicție. Urmează deci că z este pe lanțul L , $L=[x_1, x_2, \dots, z, \dots, x_k]$. Din acest lanț am putea construi un ciclu $C=[x_1, x_2, \dots, z, x_1]$, ceea ce este absurd. Contradicția provine din presupunerea existenței a cel mult unui vârf terminal. Demonstrația este terminată.

Propoziția 2.3.1.2. *Orice arbore cu n vârfuri are $n-1$ muchii.*

Demonstrație. Să observăm mai întâi (ca exemplu) că arborele din figura 2.3.1.1 are 12 vârfuri și 11 muchii.

Facem demonstrația prin inducție după n . Pentru $n=1$, există un singur arbore și el nu are muchii, deci numărul său de muchii este $1-1=0$.

Presupunem proprietatea din enunț adevărată pentru orice arbore cu n vârfuri și să o demonstrăm pentru arborii cu $n+1$ vârfuri.

Fie H un arbore cu $n+1$ vârfuri. Conform propoziției anterioare, rezultă că există în H un vârf terminal; fie acesta x . Eliminând din H vârful x împreună cu muchia, incidentă cu el, obținem un arbore H_1 , cu n vârfuri (într-adevăr, H_1 este conex și fără cicluri). Pentru el aplicăm ipoteza inductivă (el având n vârfuri) și deducem că H_1 are $n-1$ muchii. Adăugând la loc muchia eliminată, refacem arborele H și constatăm în plus că el are $(n-1)+1=n=(n+1)-1$ muchii, ceea ce și voiam să demonstrăm.

Din cele două etape ale inducției, rezultă că afirmația din enunț este, demonstrată.

Ne propunem să vedem cum rezolvăm algoritmic unele probleme legate de arbori.

O primă problemă este aceea a verificării dacă un graf este arbore. Verificăm mai întâi dacă el este conex, folosind, de exemplu una din metodele de parcurgere. Dacă el este, conex, pentru a fi arbore ar fi suficient ca graful să aibă $n-1$ muchii, lucru ușor de verificat, chiar înainte de verificarea conexității.

O altă problemă interesantă este aceea a verificării dacă un graf este fără cicluri.

Definiția 2.3.1.3. *Un graf $G=(V, E)$ care nu conține cicluri se numește **aciclic**.*

Algoritmul pe care îl propunem se bazează pe faptul că reconstituim mulțimea V de vârfuri, plecând inițial de la extremitățile primei muchii, Se adaugă la V noi vârfuri adiacente cu vârfuri deja adăugate anterior la Y . Dacă în tentativa de a căuta asemenea vârfuri, găsim unul în V care este adiacent cu un vârf care este de asemenea în V , deducem ușor că am depistat existența unui ciclu și ne oprim. Dacă nu găsim nici o muchie care să aibă o extremitate în V și o extremitate în afara lui V , se adaugă la V extremitățile unei muchii oarecare, neutilizate încă. Căutarea vârfurilor, pentru a le adăuga în V , o facem deci printre extremitățile muchiilor grafului. Se pornește inițial cu mulțimea Y a indicilor muchiilor grafului, exceptând prima muchie pe care o folosim pentru inițializarea mulțimii V . Pe măsură ce prelucrăm anumite muchii (adăugând la V o extremitate sau chiar pe amândouă), scoatem din Y indicii lor.

În continuare algoritmul scris în pseudocod. Presupunem că muchiile grafului sunt date în vectorul u .

Pas 1. aciclic:=true; {dacă vom descoperi ca $G = (V,E)$ conține cicluri, vom schimba valoarea variabilei aciclic în false}

$V=\{u[1].x,u[1].y\}$; {punem în X , extremitățile primei muchii;}

$Y=\emptyset$; {inițializăm Y cu mulțimea vidă;}

Pas 2. pentru $i:=2,m$ execută $Y:=Y \cup \{i\}$; {punem în Y indicii tuturor muchiilor, exceptând 1}

Pas 3. atâta timp cât (aciclic=true) and ($Y \neq \emptyset$) execută

$g:=false$; {se folosește pentru a testa dacă am găsit o muchie cu o extremitate în V }

pentru $j:=2,m$ execută {determină prima muchie care are cel puțin o extremitate în V }

dacă $j \in Y$ atunci

```

x1:=u[j].x;y1:=u[j].y;
dacă (x1∈V) or (y1∈V) atunci g:=true break;
dacă g=false atunci determină prima muchie j din Y
V:=V∪{u[j].x,u[j].y}
Y:=Y-{j}

Altfel, dacă (x1∈V) and (y1∈V) atunci
aciclic:=false; {G are cicluri}
altfel determină extremitatea t (a muchiei j) care nu este în V
V:=V∪{t}; {adaug t la V};
Y:=Y-{j}; {scot muchia j din Y}

```

Pas 4. Dacă aciclic=true **atunci** scrie 'graf aciclic'
altfel scrie 'graful conține cicluri'

De asemenea s-ar putea pune problema construirii unui arbore parțial pentru un graf conex. Se poate folosi ideea din algoritmul anterior. De data aceasta alegem o muchie numai dacă ea are o extremitate în V și o extremitate în afara mulțimii V. Precizăm încă odată că în algoritm, V este o variabilă de tip mulțime cu ajutorul căreia noi alegem sau nu o muchie sau un vârf, și nu semnificația tradițională de mulțime a tuturor vârfurilor grafului G.

Definiția 2.3.1.4. Un graf G care nu, conține cicluri se numește **pădure**.

Exemplul 2.3.1.2. În figura 2.3.1.2 este desenat un graf care este pădure. Denumirea introdusă se justifică prin faptul că fiecare componentă conexă a unui graf fără cicluri este un arbore.

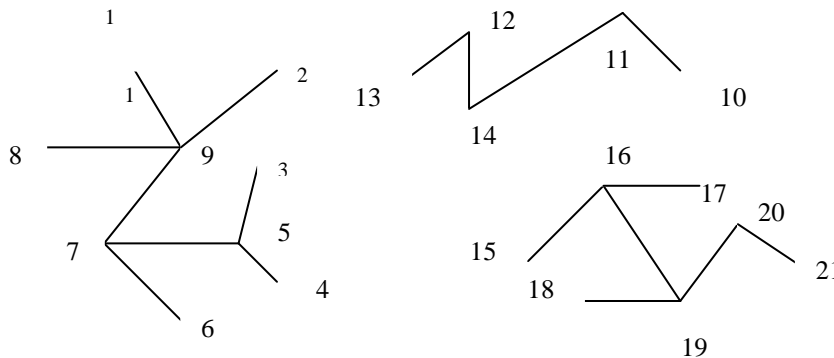


Figura 2.3.1.2

2.3.2. Arbore parțial de cost minim. Algoritmi

Fie un graf $G=(V, E)$ conex, $V = \{1, 2, \dots, n\}$, și o funcție $C:E \rightarrow \mathbb{R}_+$ asociază fiecărei muchii u , un număr real pozitiv $c(u)$, numit **costul său**.

Definiția 2.3.2.1. Pentru un graf parțial $H=(V, T)$ al lui G , costul său reprezintă suma costurilor muchiilor sale, adică:

$$c(H) = \sum_{u \in E} c(u)$$

Exemplul 2.3.2.1. Fie graful $G=(V, E)$, a cărui reprezentare grafică este dată în figura 2.3.2.1, costul fiecărei muchii fiind scris pe ea. Pentru $H=(V, T)$ unde:

$$T = \{[1,2], [3,5], [4,3], [6,7]\},$$

$$c(H) = c([1,2]) + c([3,5]) + c([4,3]) + c([6,7]) = 2 + 2 + 1 + 5 = 10$$

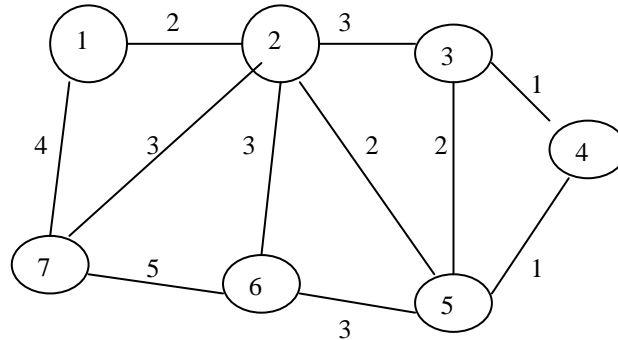


Fig. 2.3.2.1.

Problemă: Să se determine un graf parțial H al lui G care să fie conex și să aibă costul minim.

Observația 2.3.2.1. Pentru un arbore parțial de cost minim folosim notația prescurtată **APM**.

Problema enunțată este cunoscută și sub numele de “**problema conectării cu cost minim a orașelor**”, deoarece putem să interpretăm cele n vârfuri ale grafului ca fiind n orașe, iar costul muchiei $[i, j]$ ca reprezentând costul conectării, directe a orașelor i și j . Un arbore parțial conex de cost minim reprezintă modalitatea optimă din punct de vedere financiar de a lega direct unele perechi de orașe astfel încât în final orice două orașe să fie conectate (direct sau prin intermediul altora).

Propoziția 2.3.2.1. Pentru graful G conex, cu funcția de cost c , există un graf parțial H conex și de cost minim, care, în plus, este arbore.

Demonstrație. Dacă G este arbore, atunci $H=G$ deoarece orice muchie s-ar elimina, proprietatea de conexitate s-ar pierde. Dacă G nu este arbore, el conține un număr finit de grafuri parțiale care să fie în plus și conexe, fie acestea G_1, G_2, \dots, G_k . Îl alegem pe cel de cost minim și îl notăm cu H .

Presupunem prin reducere la absurd că H nu este arbore. Cum el este conex, rezultă că are cel puțin un ciclu. Eliminăm o muchie u aparținând unui ciclu, obținem un graf parțial H_1 , de asemenea conex și în plus $c(H_1) = c(H) - c(u)$, de unde rezultă că $c(H_1) < c(H)$, ceea ce este absurd. Urmează că H este arbore.

Pentru rezolvarea acestei probleme, sunt cunoscuți mai mulți algoritmi. Ne rezumăm numai la unul.

2.3.2.1. Algoritmul lui Kruskal

Acest algoritm a fost stabilit de Kruskal în anul 1956 și de cele mai multe ori referirea la algoritm se face folosind numele autorului

Se pornește inițial cu un graf parțial al grafului G care nu conține nici o muchie, deci conține n vârfuri izolate. Se poate așadar considera ca sunt n arbori disjuncți $H_1, H_2, \dots, H_i = (\{i\}, \emptyset)$, $i = 1, \dots, n$. La pasul k ($k=0, 1, \dots, n-2$) al algoritmului avem $n-k$ arbori disjuncți, fie aceștia H_1, H_2, \dots, H_{n-k} , $H_i = (X_i, E_i)$, astfel încât $V_1 \cup V_2 \cup \dots \cup V_{n-k} = V$ și $V_i \cap V_j = \emptyset \forall i \neq j$. Obținem astfel prin *unificarea* a doi dintre arborii existenți, aleși într-un anumit mod, $n-k-1$ arbori disjuncți. Alegerea celor doi arbori ce vor fi *unificați* se face în felul următor: dintre toate muchiile nealuate încă, se, selectează aceea de cost minim care are cele două

extremități în două mulțimi diferite V_i și V_j (această condiție se impune pentru ca adăugarea unei muchii să nu provoace apariția unui ciclu în graful parțial de cost minim ce se construiește din aproape în aproape). Prin adăugarea acestei muchii u , din arborii H_i și H_j , se va forma un nou arbore $H'=(V',T')$, $V'=V_i \cup V_j$, $T'=T_i \cup T_j \cup \{u\}$. Evident, după pasul $n-2$ obținem un singur arbore.

Detalii de implementare pentru algoritmul lui Kruskal

Pentru graful conex dat se folosește al treilea mod de reprezentare, iar pentru a ști în fiecare moment care sunt vârfurile ce aparțin aceluiași subarbore parțial H_i , se asociază tuturor vârfurilor lui H , aceeași valoare; evident pentru subarbori parțiali distincți vârfurile acestora vor avea asociate valori distincte. Folosim pentru acest scop vectorul L cu n componente.

struct Muchie

```
{
  int a,b;
  float cl;
};
```

.....
Muchie u[50];

Int L[30];

Etaple algoritmului lui Kruskal sunt:

Pas 1. $L[i]=i$, $i = 1,2, \dots, n$. Inițial fiecare vârf face parte, singur, dintr-un subarbore, $H_i=(\{i\}, \emptyset)$.

Pas 2. Se ordonează vectorul u crescător după costul c al muchiilor.

Pas 3. $ct=0$; {costul total};

$k=0$; {numărul total de muchii alese};

$i=1$; {indice pentru vectorul ce conține muchiile};

Pas 4. Pentru a alege cele $n-1$ muchii ale arborelui parțial de cost minim se parcurg elementele vectorului u astfel:

Fie $v=[u[i].a, u[i].b]$ muchia ce urmează să fie analizată. Inițial i , este 1. Dacă în vectorul L în extremitățile acestei muchii, avem valori egale, rezultă că ele aparțin deja aceluiași subarbore, deci alegerea ei ar provoca apariția unui ciclu. Prin urmare, această muchie nu este, aleasă. Dacă, din contră, în extremitățile, ei vectorul L are valori diferite, înseamnă că ele fac parte din doi subarbori diferiți, fie aceștia $H_a=(V_a,E_a)$ și $H_b=(V_b,E_b)$, care pot fi unificați, operație care presupune că în toate vârfurile aparținând celor doi subarbori să facem să apară aceeași valoare, provenită din, primul sau din al doilea subarbore. Se obține subarborile $H_c=(V_a \cup V_b, E_a \cup E_b \cup \{v\})$. Acest pas se repeta până am reușit să, alegem cele $n-1$ muchii ale arborelui parțial de cost minim.

Scriem într-o formă condensată pasul 4:

Pas 4. atâta timp cât $k < n - 1$ execută

dacă $L[u[i].x] \neq L[u[i].y]$ atunci $k:=k+1$ {alege muchia i };

$ct:=ct+u[i].c$

$w:=L[u[i].x]$ {urmeaza unificarea celor doi subarbori};

$v:=L[u[i].y]$

pentru $j:=1, n$ execută

dacă $L[j]=v$ atunci $L[j]:=w$;

$i:=i+1$;

Pas 5. scrie ct.

Pentru graful din figura 2.3.2.1. inițial avem:

$u_1 \quad u_2 \quad u_3 \quad u_4 \quad u_5 \quad u_6 \quad u_7 \quad u_8 \quad u_9 \quad u_{10} \quad u_{11}$

$u = ([2,5], [1,7], [6,2], [2,7], [6,5], [2,3], [5,4], [1,2], [3,4], [3,5], [7,6])$

După ordonarea elementelor vectorului, în ordinea crescătoare a lui c , avem:

$u = ([5,4], [3,4], [3,5], [1,2], [2,5], [2,3], [6,5], [2,6], [2,3], [1,7], [7,6])$

$c = (1, 1, 2, 2, 2, 3, 3, 3, 3, 4, 5)$

Se inițializează L cu $(1, 2, 3, 4, 5, 6, 7)$.

În acest moment avem $H_i = (\{i\}, \emptyset)$, pentru $i = 1, \dots, 7$.

Se alege muchia $[5, 4]$, L devine: $(1, 2, 3, 5, 5, 6, 7)$.

Au fost unificați subarborii H_4 și H_5 , obținându-se subarborii $H_5 = (\{4,5\}, \{[5,4]\})$.

Ceilalți subarbori H_i cu $i \in \{1,2,3,6,7\}$, rămân neschimbați.

Se alege muchia $[3,4]$ și L devine: $(1,2,3,3,3,6,7)$.

Obținem $H_3 = (\{3,4,5\}, [3,4], [5,4])$, iar $H_i = (\{i\}, \emptyset)$, pentru $i \in \{1,2,6,7\}$.

În, acest moment urmează la rând muchia $[3,5]$ care nu poate fi luată deoarece avem $L[3] = L[5]$ (vârfurile 3 și 5 aparțin deja aceluiași subarborii H_3) și alegerea ei ar provoca apariția unui ciclu.

Se merge mai departe și se alege muchia $[1,2]$. L devine: $(1,1,3,3,3,6,7)$.

Obținem $H_1 = (\{1,2\}, \{ [1,2] \})$. Apoi se alege $[2,5]$ și $L = (1, 1, 1, 1, 1,6,7)$. În acest moment avem trei subarbori: $H_1 = (\{1,2,3,4,5\}, \{ [1,2], [2,5], [3,4], [5,4] \})$ iar H_6 și H_7 rămân neschimbați.

Muchia $[2,3]$ nu poate fi luată pentru că ar crea un ciclu (vârfurile 2 și 3 aparțin deja aceluiași subarborii H_1). Se alege pe rând, muchiile $[6,5]$ (obținându-se subarborii $H_1 = (\{1,2,3,4,5,6\}, \{ [1,2], [2,5], [3,4], [5,4], [6,5] \})$) și, în sfârșit, $[2,7]$, L devenind: $(2,2,2,2,2,2,2)$, ceea ce corespunde arborelui parțial de cost minim $H = (\{1,2,3,4,5,6,7\}, \{ [1,2], [2,5], [3,4], [5,4], [6,5], [2,7] \})$. Nu are importanța care subarborii se alege pentru a transfera în vârfurile celui alt subarborii (în componentele corespunzătoare lor din vectorul L) aceeași valoare.

Programul C++ care determină **APM** folosind algoritmul lui Kruskal este următorul:

```
#include<iostream.h>
struct muchie
{
    int x,y;
    float cost;
};
muchie u[20];
int i,j,m,n,k,L[20],ct,v,w,aux;
void citire()
{ cout<<"\n Dati numarul de noduri";
  cin>>n;
  cout<<"\n Dati numarul de muchii";
  cin>>m;
  for(i=1;i<=m;i++)
  { cout<<"\n Dati muchia "<<i<<endl;
```

```

        cin >>u[i].x;
        cin >>u[i].y;
        cout<<"\n dati costul muchiei "<<i<<endl;
        cin>>u[i].cost;
    }
}
main()
{
citire();
    int m1=m;
    for (i=1;i<m1;i++)
        for (j=i+1;j<=m1;j++)
            if (u[i].cost>u[j].cost)
                {
                    aux=u[i].cost;
                    u[i].cost=u[j].cost;
                    u[j].cost=aux;
                }
        cout<<endl;
    for(i=1;i<=m;i++)
        cout<<u[i].cost<<" ";
    cout<<endl;
    for (int t=1;t<=n;t++)
        L[t]=t;
    ct=0;
    k=0;
    i=1;
    while (k<n-1)
    {
        if (L[u[i].x]!=L[u[i].y])
        {
            k++;
            ct=ct+u[i].cost;
            cout<<u[i].x<<" "<<u[i].y<<" ";
            v=L[u[i].y];
            w=L[u[i].x];
            for (j=1;j<=n;j++)
                if (L[j]==v) L[j]=w;
        }
        i++;
    }
    cout<<"\n Costul total este " <<ct;
}

```

2.3.3. Arbori binari de căutare optimali

O clasa foarte importantă de arbori cu rădăcină o constituie arborii binari.

Definiția 2.3.3.1. *Un arbore binar este un arbore cu rădăcina în care gradul oricărui vârf este cel mult egal cu doi.*

Putem defini recursiv arborii binari astfel :

Definiția 2.3.3.2. *Un arbore binar este un arbore care fie este vid, fie constă dintr-un nod rădăcina și doi arbori binari disjuncți numiți subarborii stâng, respectiv subarborii drept.*

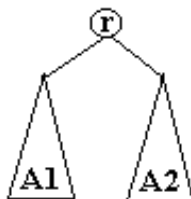


Fig. 2.3.3.1.

A_1 = subarbore stâng; A_2 = subarbore drept.

Se face o distincție clară între subarborii drept și cel stâng. Dacă subarborii stâng este nevid, rădăcina lui se numește fiul stâng al rădăcinii. Analog, dacă subarborii drept este nevid, rădăcina lui se numește fiul drept al rădăcinii.

De exemplu,

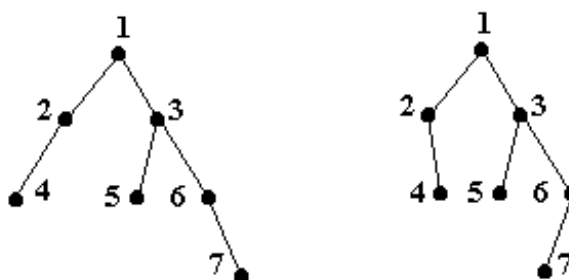


Fig. 2.3.3.2.

sunt doi arbori binari distincți.

Fie $a_1 < a_2 < \dots < a_n$ o mulțime statică de chei. Să se construiască un arbore binar de căutare optimal. Deci, odată arborele construit, nu se vor efectua operații de inserare sau ștergere, ci numai operații de căutare. Pentru evaluarea optimalității unui arbore binar de căutare se impune definirea unei funcții cost pe mulțimea tuturor arborilor binari de căutare cu cheile $a_1 < a_2 < \dots < a_n$.

Notăm cu p_i probabilitatea de a căuta valoarea a_i . În cazul în care s-ar efectua numai căutări cu succes, costul arborelui ar fi:

$$\sum_{i=1}^n p_i \cdot (\text{nivel}(a_i) + 1)$$

deoarece pentru căutarea valorii a_i în arbore, numărul de comparații necesare este egal cu numărul de noduri de pe drumul de la rădăcină la nodul cu cheia a_i , deci, în ipoteza că nivelurile sunt numerotate de la zero, este egal cu $\text{nivel}(a_i) + 1$.

Dar nu se execută numai căutări cu succes, de aceea trebuie să includem în costul arborelui și costul căutărilor fără succes. Să partiționăm mulțimea valorilor care nu sunt în arbore în n clase, astfel:

$$C_0 = \{ x \mid x < a_1 \}$$

$$C_n = \{ x \mid x > a_n \}$$

$$C_i = \{ x \mid a_i < x < a_{i+1} \}, \forall i \in \{1, 2, \dots, n-1\}.$$

Notăm cu q_i probabilitatea ca valoarea căutată să fie în clasa C_i , $i \in \{0, 1, \dots, n\}$.

$$\sum_{i=1}^n p_i + \sum_{i=0}^n q_i = 1$$

Costul căutărilor fără succes va fi:

$$\sum_{i=0}^n q_i \cdot \text{nivel}(\text{nod}_{\text{extern } i})$$

deoarece orice căutare fără succes se termină pe un nod extern. Nodul extern i , $0 \leq i \leq n$, (considerând numerotarea de la stânga la dreapta) este nodul corespunzător clasei C_i .

Definiția 2.3.3.3. Numim **arbore binar de căutare optimal** pentru mulțimea $a_1 < a_2 < \dots < a_n$ un arbore binar de căutare de cost minim, care minimizează expresia

$$\sum_{i=1}^n p_i \cdot (\text{nivel}(a_i) + 1) + \sum_{i=0}^n q_i \cdot \text{nivel}(\text{nod}_{\text{extern } i})$$

Exemplul 2.3.3.1. Fie $a_1=5$, $a_2=10$, $a_3=20$. Arborii binari de căutare pentru această mulțime de valori sunt:

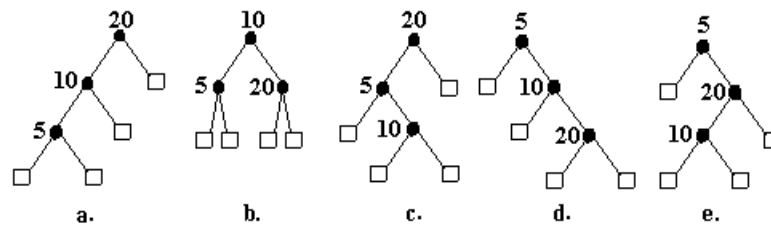


Fig. 2.3.3.3.

Dacă vom considera $p_1=p_2=p_3=1/7$ și $q_0=q_1=q_2=q_3=1/7$, deci orice valoare este căutată în arbore cu aceeași probabilitate, obținem:

- cost(arbore a.) = 15/7
- cost(arbore b.) = 13/7
- cost(arbore c.) = 15/7
- cost(arbore d.) = 15/7
- cost(arbore e.) = 15/7

Așa cum era de așteptat, arborele binar de căutare optimal este b.

Dar, pentru $p_1=0.5$; $p_2=0.1$; $p_3=0.05$ și $q_0=0.15$; $q_1=0.1$; $q_2=0.05$; $q_3=0.05$ obținem :

- cost(arbore a.) = 2.65
- cost(arbore b.) = 1.9
- cost(arbore c.) = 2.05
- cost(arbore d.) = 1.5
- cost(arbore e.) = 1.6

Deci, arborele binar de căutare optimal este arborele d.

Pentru determinarea arborelui binar de căutare optimal pentru o mulțime dată de valori $a_1 < a_2 < \dots < a_n$, o primă soluție ar fi să generăm toți arborii binari de căutare pentru mulțimea de valori dată, să calculăm costul fiecărui arbore și să selectăm arborele de cost minim.

Dar numărul arborilor binari cu n vârfuri date este :

$$B(n) = O\left(\frac{4^n}{n^{3/2}}\right)$$

ceea ce face ca aplicarea acestei metode să nu fie posibilă pentru valori mari ale lui n .

Făcând unele observații referitoare la proprietățile arborilor binari de căutare optimali, un algoritm mult mai eficient se bazează pe metoda programării dinamice.

Se notează pentru orice $\forall 1 \leq i < j \leq n$:

$$T_{i,j} = \begin{cases} \text{arborele binar de cautare optimal pentru } a_{i+1} \dots a_j, \text{ daca } i < j \\ \text{arborele vid, daca } i = j \\ \text{nedefinit, daca } i > j \end{cases}$$

$$C_{i,j} = \begin{cases} \text{costul arborelui } T_{i,j}, \text{ daca } i < j \\ 0 \text{ daca } i = j \\ \text{nedefinit, daca } i > j \end{cases}$$

$$r_{i,j} = \begin{cases} \text{radacina arborelui } T_{i,j}, \text{ daca } i < j \\ 0 \text{ daca } i = j \\ \text{nedefinit, daca } i > j \end{cases}$$

$$w_{i,j} = \begin{cases} q_i + \sum_{k=i+1}^j (q_k + p_k) = \text{ponderea arborelui } T_{i,j}, \text{ daca } i < j \\ q_i, \text{ daca } i = j \\ \text{nedefinit, dacai } > j \end{cases}$$

Conform acestor notații:

T_{0n} este arborele binar de căutare optimal pentru a_1, \dots, a_n ;

c_{0n} este costul arborelui binar de căutare optimal T_{0n} ;

r_{0n} este rădăcina arborelui binar de căutare optimal pentru a_1, \dots, a_n ;

$w_{0n} = 1$.

Fie $T_{i,j}$ este un arbore binar de căutare optimal pentru a_{i+1}, \dots, a_j și $r_{i,j} = k$ ($i < k \leq j$). $T_{i,j}$ are doi subarbori : subarborele stâng, care conține cheile a_{i+1}, \dots, a_{k-1} , și subarborele drept, care conține cheile a_{k+1}, \dots, a_j .

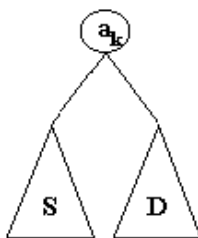


Fig. 2.3.3.4.

Atunci:

$$c_{ij} = p_k + \text{cost}(S) + \text{cost}(D) + \text{pondere}(S) + \text{pondere}(D)$$

$$\text{pondere}(S) = \text{pondere}(T_{i,k-1}) = w_{i,k-1}$$

$$\text{pondere}(D) = \text{pondere}(T_{k,j}) = w_{k,j}$$

$$p_k + w_{i,k-1} + w_{k,j} = w_{i,j}$$

$$\text{Deci, } c_{ij} = \text{cost}(S) + \text{cost}(D) + w_{ij}.$$

Din această relație deducem că $T_{i,j}$ este optimal dacă și numai dacă S și D sunt optimali, deci $S=T_{i,k-1}$ și $D=T_{k,j}$ (altfel înlocuind S sau D cu un arbore de cost mai mic, am obține un arbore pentru a_{i+1}, \dots, a_j de cost mai mic decât $c_{i,j}$, în contradicție cu ipoteza că $T_{i,j}$ este optimal).

Problema are prin urmare substructură optimală pe care o caracterizăm prin următoarea relație de recurență :

$$C_{i,j} = C_{i,k-1} + C_{k,j} + w_{i,j} \quad (*)$$

Cum $T_{i,j}$ este optimal, rădăcina sa $r_{i,j}=k$ trebuie să fie selectată astfel încât să se obțină un cost minim.

Vom folosi această relație pentru a obține T_{0n} în mod *bottom-up*, conform principiilor programării dinamice.

De exemplu, pentru $n=4$ și

	0	1	2	3	4
a=	-	5	10	15	20
p=	-	3/16	3/16	1/16	1/16
q=	2/16	3/16	1/16	1/16	1/16

Utilizând relația (*) calculăm $w_{i,j}$, $c_{i,j}$, $r_{i,j}$, pentru $\forall i,j \in \{0,1,2,3,4\}$, $i \leq j$:

$w_{0,0}=2/16$ $c_{0,0}=0$ $r_{0,0}=0$	$w_{1,1}=3/16$ $c_{1,1}=0$ $r_{1,1}=0$	$w_{2,2}=1/16$ $c_{2,2}=0$ $r_{2,2}=0$	$w_{3,3}=1/16$ $c_{3,3}=0$ $r_{3,3}=0$	$w_{4,4}=1/16$ $c_{4,4}=0$ $r_{4,4}=0$
$w_{0,1}=8/16$ $c_{0,1}=8/16$ $r_{0,1}=1$	$w_{1,2}=7/16$ $c_{1,2}=7/16$ $r_{1,2}=2$	$w_{2,3}=3/16$ $c_{2,3}=3/16$ $r_{2,3}=3$	$w_{3,4}=3/16$ $c_{3,4}=3/16$ $r_{3,4}=4$	----- ----- -----
$w_{0,2}=12/16$ 6 $c_{0,2}=19/16$ $r_{0,2}=1$	$w_{1,3}=9/16$ $c_{1,3}=12/16$ $r_{1,3}=2$	$w_{2,4}=5/16$ $c_{2,4}=8/16$ $r_{2,4}=3$	----- ----- -----	----- ----- -----
$w_{0,3}=14/16$ 6 $c_{0,3}=25/16$ $r_{0,3}=2$	$w_{1,4}=11/16$ 6 $c_{1,4}=19/16$ $r_{1,4}=2$	----- ----- -----	----- ----- -----	----- ----- -----
$w_{0,4}=1$ $c_{0,4}=2$ $r_{0,4}=2$	----- ----- -----	----- ----- -----	----- ----- -----	----- ----- -----

Deci arborele binar de căutare optimal este:

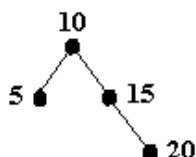


Fig. 2.3.3.5.

Algoritm de determinare a unui arbore binar de căutare optimal :

calcul(r , c , w , p , q)

pentru $i \leftarrow 0, n-1$ executa //inițializare

$w[i, i] \leftarrow q[i]$

$c[i, i] \leftarrow 0$

$r[i, i] \leftarrow 0$

$w[i, i+1] \leftarrow q[i] + q[i+1] + p[i+1]$

$c[i, i+1] \leftarrow w[i, i+1]$

```

    r[i, i+1] ← i+1
sfarsit_pentru
c[n,n] ← 0 r[n,n] ← 0 w[n,n] ← q[n]
pentru d ← 2, n executa
    pentru i ← 0, n-d executa
        j ← i+d
        w[i, j] ← w[i, j-1]+p[j]+q[j]
        k ← i+1
        pentru t ← i+2, j executa
            daca c[i, k-1]+c[k, j] > c[i, t-1]+c[t, j] atunci k ← t
        sfarsit_daca
        sfarsit_pentru
        r[i, j] ← k
        c[i, j] ← c[i, k-1]+c[k, j]
    sfarsit_pentru
sfarsit_pentru
sfarsit_calcul

```

Observația 2.3.3.1. Construirea unui arbore binar de căutare optimal prin metoda programării dinamice are complexitatea timp de $O(n^3)$.

Se poate îmbunătăți timpul de execuție a algoritmului, utilizând un rezultat datorat lui D. E. Knuth, conform căruia putem restrânge domeniul în care se caută valoarea ce minimizează expresia lui $C_{i,j}$ de la intervalul $[i+1, j]$ la intervalul $[r_{i, j-1}, r_{i+1, j}]$. În acest caz procedura de calcul va fi $O(n^2)$.

2.4. Aplicații

Aplicația 2.4.1. Într-o secție a unei întreprinderi, se aplică asupra unui produs anumite operații de prelucrare. Prelucrările ce se pot face, tehnologiile posibile de ordonare a lor și costurile unitare asociate fiecărei prelucrări sunt date într-un model-graf ponderat cu reprezentarea geometrică dată prin figura 2.4.1.

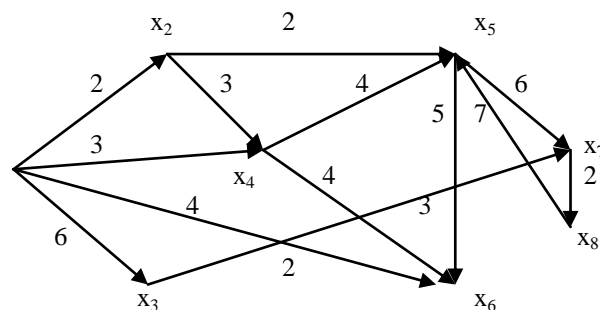


Fig. 2.4.1

Să se determine tehnologia corespunzătoare celui mai scăzut cost unitar total.

Rezolvare. Sensul economic al elementelor grafului dat impune ca acesta să nu aibă circuite, ceea ce verificăm cu ajutorul matricei drumurilor.

Tabelul 2.4.1 conține matricea arcelor grafului dat, cu ajutorul căreia s-a calculat matricea drumurilor, din tabelul 2.4.2

Tabelul 2.4.1

	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇	x ₈
x ₁	0	1	1	1	0	1	0	0
x ₂	0	0	0	1	1	0	0	0
x ₃	0	0	0	0	0	0	1	0
x ₄	0	0	0	0	1	1	0	0
x ₅	0	0	0	0	0	1	1	1
x ₆	0	0	1	0	0	0	1	0
x ₇	0	0	0	0	0	0	0	1
x ₈	0	0	0	0	0	0	0	0

Tabelul 2.4.2

	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇	x ₈
x ₁	0	1	1	1	1	1	1	1
x ₂	0	0	1	1	1	1	1	1
x ₃	0	0	0	0	0	0	1	1
x ₄	0	0	0	0	1	1	1	1
x ₅	0	0	1	0	0	1	1	1
x ₆	0	0	1	0	0	0	1	1
x ₇	0	0	0	0	0	0	0	1
x ₈	0	0	0	0	0	0	0	0

Trebuie determinat drumul de valoare minimă de, la x₁ (care evident reprezintă starea inițială a produsului) la x₈ (care reprezintă starea finală).

Aplicăm algoritmul Bellman-Kalaba. Matricea valorilor, împreună cu liniile suplimentare, se găsesc în tabelul 2.4.3.

Tabelul 2.4.3.

	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇	x ₈
x ₁	0	2	6	3	∞	4	∞	∞
x ₂	∞	0	∞	3	2	∞	∞	∞
x ₃	∞	∞	0	∞	∞	∞	3	∞
x ₄	∞	∞	∞	0	∞	2	5	∞
x ₅	∞	∞	∞	∞	0	5	6	7
x ₆	∞	∞	2	∞	∞	0	6	∞
x ₇	∞	∞	∞	∞	∞	∞	0	2
x ₈	∞	∞	∞	∞	∞	∞	∞	0
$m_{i8}^{(1)}$	∞	∞	∞	∞	7	∞	2	0
$m_{i8}^{(2)}$	∞	9	5	9	7	8	2	0
$m_{i8}^{(3)}$	11	9	5	9	7	7	2	0
$m_{i8}^{(4)}$	11	9	5	9	7	7	2	0

Din cercetarea acestui tabel, se observă că tehnologia de la x₁ la x₈, cu cost minim, corespunde costului 11. Căutăm arcele care formează drumul de valoare minimă, adunând întâi linia x₁ la $m_{i8}^{(4)}$ și căutând sumele de valoare minimă, adică 11, avem: 2 + 9 = 6 + 5 = 4 + 7; deci sunt trei variante, anume drumul începe fie cu arcul (x₁, x₂), fie cu arcul (x₁, x₃), fie cu arcul (x₁, x₆). Urmărim prima variantă, adunând linia x₂ la m_{i8}, și căutând sumele de valoare 9, avem 2 + 7 = 9, deci al doilea arc din prima variantă este (x₂, x₅) și rămâne de la x₅ la x₈, un drum de lungime minima 7, care este tocmai arcul (x₅, x₈). Drumul în prima variantă, format din 3 arce, este: (x₁, x₂, x₅, x₈). Analog procedăm în celelalte două variante și obținem drumurile: (x₁, x₃, x₇, x₈) și (x₁, x₃, x₃, x₇, x₈). Geometric, cele trei drumuri optime găsite, formează graful parțial din figura 2.4.2

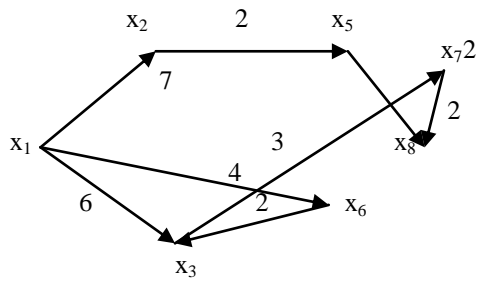


Fig. 2.4.2

Aplicația 2.4.2. Pentru aprovizionarea cabanelor dintr-un complex turistic alpin, se folosesc atât posibilitățile de transport rutier pe drumurile forestiere cât și transportul pe teleferic. Condițiile existente fac ca pe unele trasee să se poată circula în ambele sensuri, iar pe altele într-un singur sens, bine determinat. Aceste posibilități sunt oglindite într-un model graf prin muchii sau, respective arce, considerând că muchia conține două arce de sens contrar. Se cunosc distanțele pe fiecare porțiune a diverselor trasee. Știind că aprovizionarea se poate face din centrele x_1 și x_2 , Să se găsească cele mai scurte trasee ce trebuiesc parcurse de la aceste centre la toate cabanele complexului, modelul graf fiind cel din figura 2.4.3

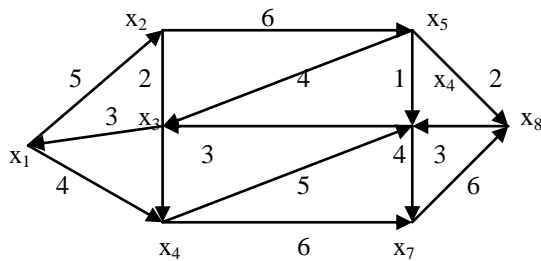


Fig. 2.4.3

Rezolvare. Graful mixt poate fi transformat în graf orientat, înlocuind fiecare muchie prin două arce de sens contrar și de aceeași valoare, ceea ce conduce la graful orientat din figura 2.4.4.

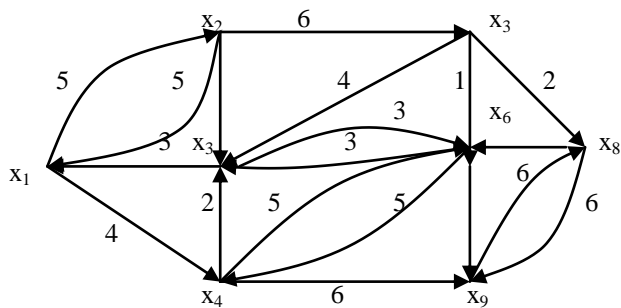


Fig. 2.4.4

Aplicăm algoritmul lui Dantzig, pentru aflarea tuturor rutelor minime. Pornim cu matricea valorilor W , din tabelul 2.4.4.

Tabelul 2.4.4.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1	0	5	∞	4	∞	∞	∞	∞
x_2	5	0	2	∞	6	∞	∞	∞
x_3	3	∞	0	∞	∞	3	∞	∞
x_4	∞	∞	2	0	∞	5	6	∞
x_5	∞	∞	4	∞	0	1	∞	∞
x_6	∞	∞	3	5	∞	0	4	∞
x_7	∞	∞	∞	∞	∞	∞	0	6
x_8	∞	∞	∞	∞	∞	3	6	0

Deducem recursiv matricele $M^{(k)}$, $k = 2, 3, \dots, 8$, ce se găsesc în tabelele 2.4.5 - 2.4.11. Ultimul dintre aceste tabele conține matricea M căutată.

Tabelul 2.4.5.

	x_1	x_2
x_1	0	5
x_2	5	0

Tabelul 2.4.6.

	x_1	x_2	x_3
x_1	0	5	7
x_2	5	0	2
x_3	3	8	0

Tabelul 2.4.7.

	x_1	x_2	x_3	x_4
x_1		5	6	4
x_2	5	0	2	9
x_3	3	8	0	7
x_4	5	10	2	0

Tabelul 2.4.8.

	x_1	x_2	x_3	x_4	x_5
x_1	0	5	6	4	11
x_2	5	0	2	9	6
x_3	3	8	0	7	4
x_4	5	10	2	0	16
x_5	7	12	4	11	0

Tabelul 2.4.9.

	x_1	x_2	x_3	x_4	x_5	x_6
x_1	0	5	6	4	11	9
x_2	5	0	2	9	6	5
x_3	3	8	0	8	14	3
x_4	5	10	2	0	16	5
x_5	7	12	4	6	0	1
x_6	6	11	3	5	17	0

Tabelul 2.4.10.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
x_1	0	5	6	4	11	9	10
x_2	5	0	2	9	6	5	9
x_3	3	8	0	7	14	3	7
x_4	5	10	2	0	16	5	6
x_5	7	12	4	6	0	1	5
x_6	6	12	3	5	17	0	4
x_7	∞	∞	∞	∞	∞	∞	0

Tabelul 2.4.11

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1	0	5	6	4	11	9	10	12
x_2	5	0	2	9	6	5	9	8
x_3	3	8	0	7	14	3	7	13
x_4	5	10	2	0	16	5	6	12
x_5	7	12	4	6	0	1	5	2
x_6	6	11	3	5	17	0	4	10
x_7	15	20	12	14	26	0	9	6
x_8	9	14	6	8	20	3	6	0

Drumurile de valori minime, care pleacă din centrul x_1 , sunt:

$$d(x_1, x_2) = (x_1, x_2), \text{ cu } m_{12} = 5;$$

$$d(x_1, x_3) = (x_1, x_4, x_3), \text{ cu } m_{13} = 6;$$

$$d(x_1, x_4) = (x_1, x_4), \text{ cu } m_{14} = 4;$$

$$d(x_1, x_5) = (x_1, x_2, x_5), \text{ cu } m_{15} = 11;$$

$$d(x_1, x_6) = (x_1, x_4, x_6), \text{ cu } m_{16} = 9;$$

$$d(x_1, x_7) = (x_1, x_4, x_7), \text{ cu } m_{17} = 10;$$

$$d(x_1, x_8) = (x_1, x_2, x_5, x_8), \text{ cu } m_{18} = 13;$$

Drumurile de valori minime, care pleacă din centrul x_4 , sunt:

$$d(x_4, x_1) = (x_4, x_4, x_1), \text{ cu } m_{41} = 5;$$

$$d(x_4, x_2) = (x_4, x_3, x_1, x_2), \text{ cu } m_{42} = 10;$$

$$d(x_4, x_3) = (x_4, x_3), \text{ cu } m_{43} = 2;$$

$$d(x_4, x_5) = (x_4, x_3, x_1, x_2, x_5), \text{ cu } m_{45} = 16;$$

$$d(x_4, x_6) = (x_4, x_6), \text{ cu } m_{46} = 5;$$

$$d(x_4, x_7) = (x_4, x_7), \text{ cu } m_{47} = 6;$$

$$d(x_4, x_8) = (x_4, x_7, x_8), \text{ cu } m_{48} = 12;$$

CAPITOLUL 3

DRUMURI HAMILTONIENE ÎN GRAFURI

3.1. Transport pe drumuri hamiltoniene

Adesea, transportul unor produse trebuie organizat cu ajutorul unei rețele de transport cunoscute (un graf), astfel încât să se treacă prin toate punctele importante luate în considerație (prin toate vârfurile, grafului), netrecând de, două sau mai multe ori prin niciunul dintre acestea. De exemplu, putem fi puși în fața necesității de a organiza aprovizionarea unor magazine, (cantine; puncte de lucru ale unei întreprinderi etc.) cu ajutorul unui singur camion, care să treacă succesiv, pe la fiecare dintre acestea, folosind o rețea de comunicații cunoscute. În plus dacă o astfel de organizare poate fi făcută în mai multe feluri se poate pune problema găsirii unei soluții care să fie cât mai convenabilă (optimă) din punct de vedere, al cheltuielilor făcute, al timpului consumat etc.

Definiția 3.1.1. a) Fie $G = (V, E)$ un graf. Un lanț, elementar, ciclu elementar, drum elementar sau circuit elementar în graful G ce conține toate nodurile lui G se numește **hamiltonian**.

b) Un graf neorientat se numește **hamiltonian** dacă are (cel puțin) un ciclu hamiltonian.

c) Un graf orientat se numește **hamiltonian** dacă are (cel puțin) un circuit hamiltonian.

În problema dată ca exemplu, mai sus, se poate cere drumul cu care să se aprovizioneze punctele respective, fără a considera drumurile camionului de la și la garaj (caz în care problema este de determinare a unui drum hamiltonian prescurtat DH), sau se poate lua în considerare și întoarcerea camionului în punctul de plecare (caz în care problema este de determinare a unui circuit hamiltonian, prescurtat CH). În plus, dacă se iau în considerare și costurile de transport, suntem conduși la problema determinării unui drum hamiltonian de valoare minimă prescurtat DHM, respectiv la problema determinării unui circuit hamiltonian de valoare minimă - prescurtat CHM. În sfârșit, dacă graful folosit este neorientat, denumirilor și notațiilor de mai sus le corespund următoarele: lanț hamiltonian (LH), ciclu hamiltonian (CH), lanț hamiltonian de valoare minimă (LHM), ciclu hamiltonian de valoare minimă (CHM).

Vom începe, studiul sistematic al unora dintre problemele *puse*, cu problema determinării drumului hamiltonian (drumurilor hamiltoniene), într-un graf orientat și neponderat. Determinarea acestor drumuri hamiltoniene se face în mod diferențiat, în grafuri fără circuite și în grafuri cu circuite.

a) *Problema drumurilor hamiltoniene în grafuri fără circuite.*

În construcția matricei drumurilor triangularizate, D' , am putut observa că primul element diferit de zero din fiecare linie, ca și ultimul element diferit de zero din fiecare coloană corespund în graful dat, $G = (V, E)$.

Teorema 3.1.1.(Y. Chen). Fie $G = (V, E)$ un graf orientat și fără circuite, cu n vârfuri. Condiția necesară și suficientă pentru ca G să conțină un drum Hamiltonian este ca numărul elementelor egale cu unu din matricea D , să fie:

$$\sum_{i=1}^n p(x_i) = \frac{n(n-1)}{2}.$$

Demonstrație. Presupunem că există un drum hamiltonian: $DH=(x_{i_1}, x_{i_2}, \dots, x_{i_n})$

corespunzând permutării: $\begin{pmatrix} 1 & 2 & \dots & n \\ i_1 & i_2 & \dots & i_n \end{pmatrix}$.

Deoarece graful nu are circuite, rezultă că x_{i_1} atinge $(n-1)$ vârfuri, deci pe linia lui x_{i_1} , în matricea D sunt $(n-1)$ elemente egale cu unu. Vârful x_{i_2} , atinge $(n-2)$ vârfuri, deci linia corespunzătoare lui din D are $(n-2)$ elemente egale cu unu ș.a.m.d., pentru fiecare vârf, al drumului hamiltonian numărul de elemente egale cu unu scade cu câte o unitate, până la vârful x_{i_n} , care este ultimul vârf al drumului hamiltonian, deci are puterea de atingere zero (în caz contrar, graful nu ar corespunde ipotezei de a nu avea circuite). Deci numărul total de elemente egale cu unu în matricea D este suma unei progresii aritmetice cu rația unu, adică:

$$(n-1)+(n-2)+, \dots, +2+1 = \frac{n(n-1)}{2}.$$

Reciproc, presupunem că suma cifrelor de unu din D , sau suma puterilor de atingere, este $\frac{n(n-1)}{2}$. Considerăm forma triangularizată, D' , în care notăm vârfurile x_1', x_2', \dots, x_n' ,

deduse prin permutarea elementelor x_1, x_2, \dots, x_n în ordinea descrescătoare a puterilor de atingere. În D' toate elementele egale cu unu se află deasupra diagonalei principale și cum deasupra diagonalei principale sunt $\frac{n(n-1)}{2}$ locuri, înseamnă că nici o cifră zero nu se află

deasupra diagonalei principale. Putem scrie o succesiune de arce, corespunzând primului element diferit de zero din fiecare linie: (x_1', x_2') , (x_2', x_3') , ..., (x_{n-1}', x_n') , succesiune care formează un drum hamiltonian $DH = (x_1', x_2', \dots, x_n')$.

Teorema 3.1.2. Într-un graf $G=(V, E)$, orientat și fără circuite, dacă există un drum hamiltonian, acesta este unic.

Demonstrație. Fie $DH=(x_1', x_2', \dots, x_n')$, unde vârfurile se succed în ordinea descrescătoare a puterilor de atingere. Orice altă succesiune a vârfurilor conține cel puțin o inversare a indicilor. Presupunem prin absurd că ar exista cel puțin un arc de forma (x_i', x_j') cu $j < i$, adică linia x_j' precede linia x_i' în D' , astfel că $d'_{ji}=1$, deci situat deasupra diagonalei principale în D' . Deducem de aici existența unui drum $d(x_j', x_i')$. Cum însă succesiunea de sus conține, drumul (x_i', x_j') , rezultă că graful are circuit, contrar ipotezei. Deci presupunerea că în graf drumul hamiltonian nu este unic, este falsă.

Exemplul 3.1.1. Prelucrarea unui obiect artizanal poate trece prin șase faze distincte, notate într-un model de graf prin $x_i (i=1, 2, \dots, 6)$, iar posibilitățile de trecere de la o fază la alta sunt reprezentate prin arce (vezi figura 3.1.1.). Este posibil ca obiectul să treacă prin toate fazele de prelucrare? Dacă acest lucru este posibil, să se indice tehnologia corespunzătoare.

Rezolvare. Matricea arcelor grafului dat este cea din tabelul 3.1.1 Cu ajutorul ei s-a calculat matricea drumurilor, din tabelul 3.1.2, Deoarece aceasta din urmă are numai zerouri pe diagonala principală, graful dat nu are circuite.

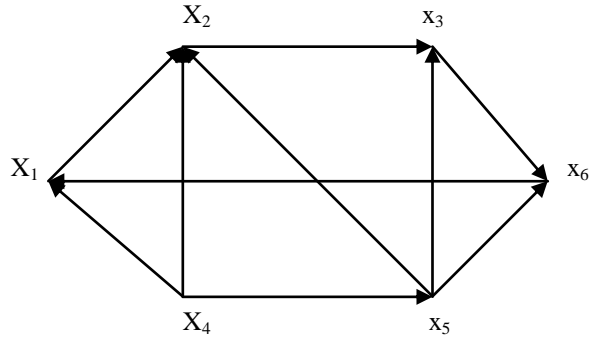


Fig.3.1.1

Tabelul 3.1.1.

Tabelul 3.1.2.

(A)

	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆
x ₁	0	1	0	0	0	0
x ₂	0	0	1	0	0	0
x ₃	0	0	0	0	0	0
x ₄	1	1	0	0	1	0
x ₅	0	1	1	0	0	1
x ₆	1	0	1	0	0	0

(D)

	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	p(x _i)
x ₁	0	1	1	0	0	0	2
x ₂	0	0	1	0	0	0	1
x ₃	0	0	0	0	0	0	0
x ₄	1	1	1	0	1	1	5
x ₅	1	1	1	0	0	1	4
x ₆	1	1	1	0	0	0	3

$$\sum p(x_i) = 15$$

Calculăm $\frac{n(n-1)}{2} = \frac{6 \cdot 5}{2} = 15 = \sum_{i=1}^6 p(x_i)$ ceea ce se observă direct, în tabelul 3.1.2.

Deci graful dat are un drum hamiltonian, anume:

$$DH=(x_4, x_5, x_6, x_1, x_2, x_3)$$

Tabelul 3.1.3.

(D')

	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	p(x _i)
x ₁	0	1	1	1	1	1	2
x ₂	0	0	1	1	1	1	1
x ₃	0	0	0	1	1	1	0
x ₄	0	0	0	0	1	1	5
x ₅	0	0	0	0	0	1	4
x ₆	0	0	0	0	0	0	3

Succesiunea respectivă corespunde la tehnologia căutată.

Observația 3.1.1. Forma triangularizată a matricei drumurilor este reprezentată în tabelul 3.1.3.

b) *Problema drumurilor hamiltoniene în grafuri oarecare* (cu circuite)

În grafuri oarecare se pot determina drumurile hamiltoniene pe mai multe căi. În cele ce urmează prezentăm doi algoritmi uzuali.

3.2. Algoritmi pentru determinarea drumurilor hamiltoniene

3.2.1. Algoritmul ce folosește graful condensat

Algoritmul pleacă de la o serie de observații evidente, în legătură cu graful condensat G^* și anume:

Observația 3.2.1.1. Dacă într-un graf G există cel puțin un drum hamiltonian atunci și graful său condensat G^* , conține un astfel de drum. Drumul hamiltonian din G trebuie să parcurgă, succesiv, toate vârfurile oricărei componente tare conexe, iar de la o componentă la alta ordinea este dictată de ordinea componentelor în drumul hamiltonian din G^* .

Observația 3.2.1.2. Dacă graful condensat G^* nu conține nici un drum hamiltonian atunci nici graful G nu conține nici un drum hamiltonian. Această observație este negația logică a propoziției precedente.

Observația 3.2.1.3. Dacă graful condensat G^* conține un drum hamiltonian nu rezultă că și graful G conține drum hamiltonian în mod necesar. În funcție de legăturile directe (arcele din G) între componentele tare conexe, graful G poate să aibă sau poate să nu conțină drum hamiltonian.

Algoritmul de deducere a drumurilor hamiltoniene din G , prin reducerea la drumul hamiltonian din G^* , cere parcurgerea următoarelor etape:

Pas. 1. Găsirea tuturor componentelor tare conexe. Fie mulțimea lor: $V^* = \{C_1, C_2, \dots, C_k\}$.

Pas. 2. Determinarea matricei arcelor A^* pentru G^* și matricei drumurilor D^* .

Pas.3. Cercetarea existenței DH^* în G^* , pe baza teoremei Chen, verificând egalitatea:

$$\sum_{i=1}^k p(C_i) = \frac{k(k-1)}{2}$$

Pas. 4. Dacă G^* nu conține un drum hamiltonian, atunci nici graful G nu conține un drum hamiltonian.

Dacă G^* conține un drum hamiltonian fie $DH^* = (C_1, C_2, \dots, C_k)$; determinăm drumurile hamiltoniene din fiecare componentă tare conexă și le legăm prin arcele care leagă componentele succesive în toate modurile posibile. Cum am mai afirmat, în G poate să existe sau poate să nu existe drum hamiltonian, în funcție de aceste legături.

Exemplul 3.2.1.1. La un șantier hidroenergetic, trebuie livrate materiale de construcții în cinci puncte de lucru. Aceste puncte de lucru au fost legate prin niște căi de acces, croite după posibilitățile permise de teren, ca în graful din figura 3.2.1.1. Se pune problema dacă basculantele pot trece pe la toate punctele de lucru să descarce materialele necesare.

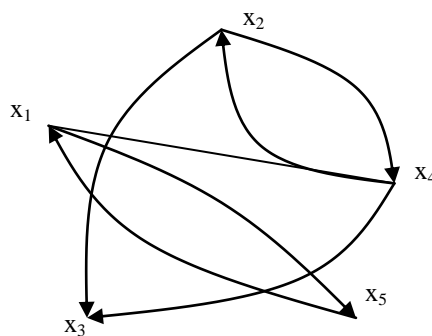


Fig. 3.2.1.1.

Rezolvare. Graful are circuite. Să aplicăm algoritmul bazat pe construirea grafului condensat. În tabelul 3.2.1.1. sunt efectuate calculele pentru determinarea componentelor tare conexe.

Tabelul 3.2.1.1.

	x ₁	x ₂	x ₃	x ₄	x ₅	W ₁	W ₂
x ₁	0	0	0	1	1	1	//////
x ₂	0	0	1	1	0	0	1
x ₃	0	0	0	0	0	0	0
x ₄	0	1	1	0	0	0	1
x ₅	1	0	0	0	0	1	//////
W ₁	1	1	1	1	1		
W ₂	//////	1	1	1	//////		

Avem:

$$C_1 = C_{(x_1)}: (W_1 \cup W'_1) \cup \{x_1\} = \{x_1, x_5\},$$

$$C_2 = C_{(x_2)}: (W_2 \cap W'_2) \cup \{x_2\} = \{x_2, x_4\},$$

$$C_3 = C_{(x_3)}: \{x_3\}.$$

Deci $G^* = (V^*, E^*)$, unde $V^* = \{C_1, C_2, C_3\}$.

Cu ajutorul matricelor intermediare din tabelele 3.2.1.2 și 3.2.1.3, determinăm matricea A^* din tabelul 3.2.1.4. Avem deci: $E^* = \{(C_1, C_2), (C_2, C_3)\}$.

Tabelul 3.2.1.2

	C ₁	C ₂	C ₃
x ₁	1	1	0
x ₂	0	1	1
x ₃	0	0	0
x ₄	0	1	1
x ₅	1	0	0

Tabelul 3.2.1.3

	C ₁	C ₂	C ₃
C ₁	1	1	0
C ₂	0	1	1
C ₃	0	0	0

Tabelul 3.2.1.4

	C ₁	C ₂	C ₃
C ₁	0	1	0
C ₂	0	0	1
C ₃	0	0	0

(A)

\tilde{A}^*

(A*)

Construim matricea drumurilor în G^* , cu algoritmul Chen, ca în tabelul 3.2.1.5.

Tabelul 3.2.1.5.

	C ₁	C ₂	C ₃	P(C _i)
C ₁	0	1	1	2
C ₂	0	0	1	1
C ₃	0	0	0	0

$$\sum_{i=1}^3 p(C_i) = 2 + 1 + 0 = 3 = \frac{3 \cdot 2}{2}$$

(graful condensat are $n^*=3$ vârfuri), în graful condensat există drumul hamiltonian ce urmează:

$$DH^* = (C_1, C_2, C_3).$$

Grafic, avem descompunerea grafului dat, ca în figura 3.2.1.2.

Ținând seama și de arcele ce leagă componentele consecutive din DH^* , reprezentate în figura 3.2.1.3. rezultă că există și în graful inițial, un drum hamiltonian, anume:

$$DH = (x_2, x_1, x_4, x_2, x_3).$$

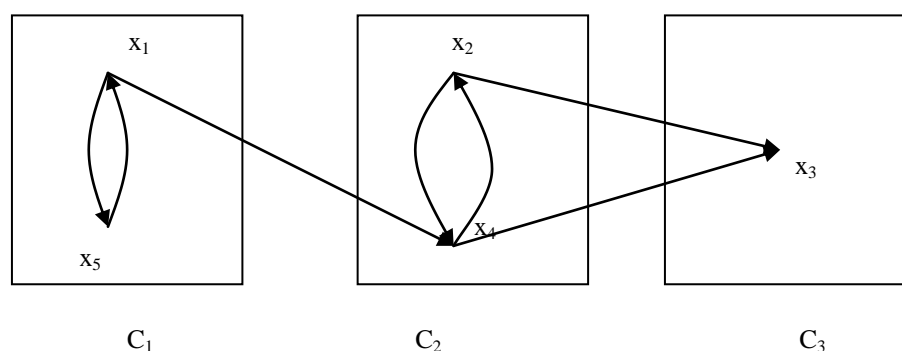


Fig.3.2.1.2.

Acesta este drumul pe care urmează să îl facă basculantele, pentru a trece pe la toate punctele.

3.2.2. Algoritmul lui Kaufmann (produs latin)

Fie un graf oarecare $G = (V, E)$. Construim matricele latine $M^{(1)}, M^{(2)}, \dots, M^{(n-1)}$, matrice pătrate de ordinul n , dacă graful are n noduri (vârfuri).

Matricea $M^{(1)}$ se construiește astfel:

- dacă $(x_i, x_j) \in E$, atunci punem secvența $x_i x_j$ în căsuța (i, j) a matricei $M^{(1)}$,
- dacă $(x_i, x_j) \notin E$, atunci punem 0 sau \emptyset în căsuța (i, j) a matricei.

Formăm $\tilde{M}^{(1)}$ din $M^{(1)}$, prin ștergerea primului vârf al secvenței $x_i x_j$ din orice căsuță (i, j) . Rezultă că matricea $\tilde{M}^{(1)}$ conține vârfurile care pot fi atinse prin arce de la fiecare vârf al grafului.

Succesiv, construim matricele: $M^{(2)}, M^{(3)}, \dots, M^{(n-1)}$, care vor da drumurile elementare între orice două vârfuri, formate din 2, 3, ..., respectiv $(n-1)$ arce, așadar matricea $M^{(n-1)}$ va conține toate drumurile hamiltoniene. În general, definim:

$$M^{(k)} = M^{(k-1)} L \tilde{M}^{(1)}, \quad k=2, 3, \dots, n-1,$$

semnul L reprezentând „produsul latin”, ce se efectuează în modul următor:

- în căsuța (i, j) a matricei $M^{(k)}$ se trece „produsul latin” al liniei i din matricea $M^{(k-1)}$, prin coloana j din matricea $\tilde{M}^{(1)}$,

- dacă la înmulțirea liniei i a lui $M^{(k-1)}$, cu coloana j a lui $\tilde{M}^{(1)}$, urmează să înmulțim secvența (x_i, x_k, \dots, x_s) cu vârful x_j scriem în căsuța (i, j) a lui $M^{(k)}$, secvența $(x_i, x_k, \dots, x_s, x_j)$ numai dacă în această secvență nu se repetă nici un vârf, în caz contrar renunțând la ea; dacă asemenea situație apare de mai multe ori, pentru aceeași căsuță (i, j) scriem în ea toate rezultatele de acest fel, unul sub altul,

- dacă la înmulțirea liniei i din $M^{(k-1)}$, cu coloana j din $\tilde{M}^{(1)}$ obținem numai secvențe completate ca mai sus, în care se repetă cel puțin un vârf, sau de fiecare data unul dintre „factori” este nul, în căsuța (i, j) scriem 0 sau \emptyset .

Vom face o justificare intuitivă a algoritmului, pe baza unui exemplu, folosind graful din, figura 3.2.2.1. Matricea $M^{(1)}$ ce conține posibilitățile de a ajunge cu un singur arc, între oricare două vârfuri ale grafului, este cea din tabelul 3.2.2.1. Să observăm că în fiecare căsuță (i, j) , a acestei matrice, se găsește sau secvența $x_i x_j$ (dacă, graful conține arc de la x_i la x_j), sau \emptyset (în caz contrar). Astfel trecerea greșită a unei secvențe în altă căsuță decât cea

corespunzătoare ei s-ar observa imediat. De exemplu, dacă secvența $x_1 x_2$ nu ar fi fost trecută în căsuța (1, 2), de pe linia, lui x_1 și coloana x_2 tabelul ar fi conținut o greșeală.

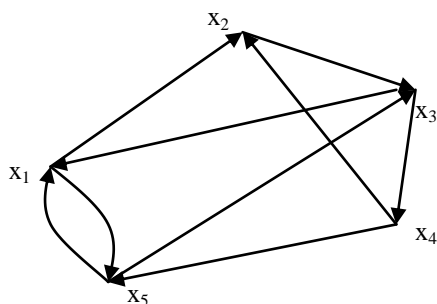


Fig. 3.2.2.1.

Tabelul 3.2.2.1.

$M^{(1)}$:

	x_1	x_2	x_3	x_4	x_5
x_1	\emptyset	x_1x_2	\emptyset	\emptyset	x_1x_5
x_2	\emptyset	\emptyset	x_2x_3	\emptyset	\emptyset
x_3	x_3x_1	\emptyset	\emptyset	x_3x_4	\emptyset
x_4	\emptyset	x_4x_2	\emptyset	\emptyset	x_4x_5
x_5	x_5x_1	\emptyset	x_5x_3	\emptyset	\emptyset

Atenție! În căsuța (i, j), a matricei $M^{(1)}$ sunt trecute toate (drumurile de la x_i la x_j , de câte un singur arc.

Matricea $\tilde{M}^{(1)}$ este cea din tabelul 3.2.2.2, ea fiind obținută din $M^{(1)}$ prin suprimarea primului vârf, de la fiecare secvență de vârfuri și aici observăm că în orice căsuță de pe coloana x_j nu poate fi decât 0 sau vârfurile x_j : în căsuța de ordin j a coloanei, se găsește x_j , dacă graful conține arcul (x_i, x_j) și 0 în caz contrar. Înseamnă că matricea $\tilde{M}^{(1)}$ conține eventualele prelungiri de drumuri cu câte un arc, primul vârf al arcului fiind suprimat pentru a nu se repeta în secvența (drumul) următoare.

Tabelul 3.2.2.2

$\tilde{M}^{(1)}$:

	x_1	x_2	x_3	x_4	x_5
x_1	\emptyset	x_2	\emptyset	\emptyset	x_5
x_2	\emptyset	\emptyset	x_3	\emptyset	\emptyset
x_3	x_1	\emptyset	\emptyset	x_4	\emptyset
x_4	\emptyset	x_2	\emptyset	\emptyset	x_5
x_5	x_1	\emptyset	x_3	\emptyset	\emptyset

Tabelul 3.2.2.3.

	X ₁	X ₂	X ₃	X ₄	X ₅
X ₁	∅	∅	X ₁ X ₂ X ₃ X ₁ X ₅ X ₃	∅	X ₅
X ₂	X ₂ X ₃ X ₁	∅	∅	X ₂ X ₃ X ₄	∅
X ₃	X ₁	X ₃ X ₁ X ₂ X ₃ X ₄ X ₂	∅	∅	X ₃ X ₁ X ₅ X ₃ X ₄ X ₅
X ₄	X ₄ X ₅ X ₁	∅	X ₄ X ₂ X ₃ X ₄ X ₅ X ₃	∅	∅
X ₅	X ₅ X ₃ X ₁	X ₅ X ₁ X ₂	∅	X ₅ X ₃ X ₄	∅

Produsul latin $M^{(2)}=M^{(1)}L_{\tilde{M}^{(1)}}$ este realizat în tabelul 3.2.2.3. Să observăm că în căsuța (i, j), a acestui tabel, se găsesc secvențe de câte trei vârfuri diferite, fiecare reprezentând câte un drum elementar de doua arce, de la vârful x_i , la vârful x_j . În continuare, câteva exemple de calcul, pentru aceste drumuri. Astfel, înmulțind latin linia x_1 , din $M^{(1)}$, cu coloana x_i , din $\tilde{M}^{(1)}$ avem:

	X ₁	X ₂	X ₃	X ₄	X ₅
X ₁	∅	X ₁ X ₂	∅	∅	X ₁ X ₅

	X ₁	X ₂	X ₃	X ₄	X ₅
X ₁	∅	∅	Cu	∅	→ ∅
X ₂	∅	X ₁ X ₂	Cu	∅	→ ∅
X ₃	X ₁	∅	Cu	X ₁	→ ∅
X ₄	∅	∅	Cu	∅	→ ∅
X ₅	X ₁	X ₁ X ₅	Cu	X ₁	→ X ₁ X ₅ X ₁

A apărut o singura secvență de trei vârfuri, la care renunțăm, deoarece se repetă vârful x_1 ; această secvență reprezintă de fapt un circuit ce trece prin vârful x_1 . Ca urmare, în căsuța (1, 1) din matricea $M^{(2)}$, a apărut ∅.

Înmulțind latin linia x_1 , din $M^{(1)}$, cu coloana x_3 din $\tilde{M}^{(1)}$, avem:

	X ₁	X ₂	X ₃	X ₄	X ₅
X ₁	∅	X ₁ X ₂	∅	∅	X ₁ X ₅

	X ₁	X ₂	X ₃	X ₄	X ₅
X ₁	∅	∅	Cu	∅	→ ∅
X ₂	∅	X ₁ X ₂	Cu	X ₃	→ X ₁ X ₂ X ₃
X ₃	X ₃	∅	Cu	∅	→ ∅
X ₄	∅	∅	Cu	∅	→ ∅
X ₅	X ₃	X ₁ X ₅	Cu	X ₃	→ X ₁ X ₅ X ₃

Deoarece au apărut două secvențe, de câte trei vârfuri distincte, în căsuța (1, 3) a matricei $M^{(2)}$ au fost trecute amândouă. Ele reprezintă cele doua drumuri posibile, de câte două arce, de la x_1 la x_3 . Cititorul este rugat să le observe și pe graful din figura 3.2.2.1. În plus, din modul cum au apărut ele, se poate observa cum funcționează înmulțirea latină:

- secvența x_1x_2 se prelungește cu vârful x_3 , adică drumul de un arc (x_1, x_2) se continuă cu arcul (x_2, x_3) , suprimarea lui x_2 în $\tilde{M}^{(1)}$ fiind necesară pentru ca el să nu se repete în secvența $x_1x_2x_3$,

- secvența x_1x_5 se prelungește, de asemenea, cu vârful x_3 (pe coloana x_3 din $\tilde{M}^{(1)}$ nu poate fi decât x_3), ducând la drumul de doua arce (x_1, x_5, x_3) .

Din cele de mai sus, rezultă că matricea $M^{(2)}$ conține toate drumurile elementare de câte doua arce, între oricare două, vârfuri ale grafului. În mod, analog, matricea $M^{(3)}=M^{(2)}L\tilde{M}^{(1)}$, din tabelul 3.2.2.4, conține drumurile elementare de câte trei arce din graful dat. Prezentăm trei exemple de calcul. Produsul latin între linia x_i din $M^{(2)}$ și coloana x_4 din $\tilde{M}^{(1)}$ se face astfel:

						x_4									
						\emptyset	\emptyset	Cu	\emptyset	\rightarrow	\emptyset				
						\emptyset	\emptyset	Cu	\emptyset	\rightarrow	\emptyset				
x_1	\emptyset	\emptyset	x_1x_2	\emptyset	\emptyset	x_4	x_1x_2	Cu	x_4	\rightarrow	$x_1x_2x_3$				
			x_3				x_3				x_4				
			x_1x_5				x_1x_5				$x_1x_5x_3$				
			x_3				x_3				x_4				
						\emptyset	\emptyset	Cu	\emptyset	\rightarrow	\emptyset				
						\emptyset	x_1x_5	Cu	\emptyset	\rightarrow	\emptyset				

Tabelul 3.2.2.4.

$M^{(3)}$:		x_1	x_2	x_3	x_4	x_5
	x_1	\emptyset	\emptyset	\emptyset	$x_1x_2x_3x_4$ $x_1x_5x_3x_4$	\emptyset
	x_2	\emptyset	\emptyset	\emptyset	\emptyset	$x_1x_2x_3x_4$ $x_1x_5x_3x_4$
	x_3	$x_3x_4x_5x_1$	\emptyset	\emptyset	\emptyset	\emptyset
	x_4	$x_4x_2x_3x_1$ $x_4x_5x_3x_1$	$x_4x_5x_1x_2$	\emptyset	\emptyset	\emptyset
	x_5	\emptyset	$x_5x_3x_1x_2$ $x_5x_3x_4x_2$	$x_5x_1x_2x_3$	\emptyset	\emptyset

După cum se vede, dacă într-o căsuță sunt două secvențe, ele se pot prelungi separat dacă nu se va repeta nici un vârf de fiecare dată, Produsul latin dintre linia x_2 din $M^{(2)}$ și coloana x_5 din $\tilde{M}^{(1)}$ se face astfel:

						x_5									
						x_5	x_2x_3	Cu	x_5	\rightarrow	$x_2x_3x_1$				
						\emptyset	x_4				x_5				
x_2	x_2x_3	\emptyset	\emptyset	x_2x_3	\emptyset	\emptyset	\emptyset	Cu	\emptyset	\rightarrow	\emptyset				
	x_1			x_4			\emptyset				\emptyset				
						x_3	x_2x_3	Cu	x_5	\rightarrow	$x_2x_3x_4$				
						\emptyset	x_4				x_5				
						\emptyset	\emptyset	Cu	\emptyset	\rightarrow	\emptyset				

Au apărut astfel două secvențe de câte patru vârfuri distincte, deci ambele se vor trece în căsuța (2, 5) din matricea $M^{(3)}$.

În sfârșit, dăm exemplul înmulțirii latine a liniei x_3 din $M^{(2)}$ cu coloana x_4 din $\tilde{M}^{(1)}$:

					X4							
					X1	∅	∅	Cu	∅	→	∅	
	X1	X2	X3	X4	X5	X2	∅	X3X1	Cu	∅	→	∅
							X2					
							X3X4					
							X2					
X3						X3	X1	∅	Cu	X1	→	∅
	∅	X3X1	∅	∅	X3X1							
		X1			X5							
		X3X4			X3X4							
		X2			X5							
						X4	∅	∅	Cu	∅	→	∅
						X5	X1	X3X1	Cu	X1	→	X3X1X5
								X5				X1
								X3X4				X3X4X5
								X5				X1

Dintre cele două secvențe apărute, s-a păstrat numai cea de a doua, în prima fiind repetat vârful x_1 (drumul corespunzător nu este elementar). Secvența păstrată a fost trecută în căsuța (3, 1) din $M^{(3)}$.

Tabelul 3.2.2.5 conține matricea $M^{(4)}=M^{(3)}L\tilde{M}^{(4)}$, în care se află drumurile elementare de câte patru arce, deci drumurile hamiltoniene din graful dat. Este evident că orice încercare de prelungire a acestor drumuri conduce numai la drumuri neelementare (se repetă de fiecare dată cel puțin vârful): deci la o matrice nulă.

Tabelul 3.2.2.5.

		X1	X2	X3	X4	X5
$M^{(3)}$:	X1	∅	X1X5X3X4X2	∅	∅	X1X2X3X4X5
	X2	X2X3X4X5X1	∅	∅	∅	∅
	X3	∅	X3X4X5X2X1	∅	∅	∅
	X4	∅	X4X5X3X1X2	X4X5X1X2X3	∅	X4X2X3X1X5
	X5	∅	∅	∅	X5X1X2X3X4	∅

Ca urmare a calculelor efectuate mai sus, rezultă că graful din figura 3.2.2.1 are opt drumuri hamiltoniene date de secvențele de vârfuri din tabelul 3.2.2.5.

Observații asupra algoritmului.

Observația 3.2.2.1. Pentru o simplificare a scrierii, putem să înlocuim secvențele de vârfuri prin secvențele corespunzătoare de indici. De exemplu, în loc de $x_1x_2x_3$, putem scrie pe scurt 123. În același scop, vârfurile se puteau nota inițial prin litere, de exemplu, prin A, B, C, D, ..., secvența amintită fiind notată prin ABC. De altfel, de la astfel de notații vine și numele de „produs latin”.

Observația 3.2.2.2. Algoritmul poate fi folosit și la determinarea circuitelor hamiltoniene, continuând cu încă un pas al aplicării lui și păstrând numai secvențele care se încheie cu același vârf cu care încep. Astfel, în exemplu de mai sus, efectuând produsul: $M^{(5)}=M^{(4)}L\tilde{M}^{(5)}$, dar păstrând secvențele de tipul amintit, se obține circuitul $(x_1, x_2, x_3, x_4, x_5, x_1)$, sub cinci forme (începând cu oricare dintre vârfurile intermediare). Să observăm că la acest pas ne interesează numai secvențele ce vor apare pe diagonală principală.

Observația 3.2.2.3. Dacă nu ne interesează matricele intermediare $M^{(k)}$, ci numai $M^{(n-1)}$ (n , numărul de vârfuri ale grafului), se poate aplica și o prescurtare de calcul. Astfel, definind pe $\tilde{M}^{(k)}$ ca matricea obținută din $M^{(k)}$, prin suprimarea primului vârf de la fiecare secvență a sa putem scrie și verifica ușor pe exemple:

$$M^{(4)}=M^{(2)}L\tilde{M}^{(2)} ; M^{(8)}=M^{(4)}L\tilde{M}^{(4)} \text{ etc.}$$

Evident, vom face asemenea prescurtări evitând depășirea lui $M^{(n-1)}$ în caz contrar obținând o matrice nulă. De exemplu, pentru un graf cu opt vârfuri, vom efectua operațiile în ordinea următoare: $M^{(2)}=M^{(1)}L\tilde{M}^{(1)}$; $M^{(4)}=M^{(2)}L\tilde{M}^{(2)}$; $M^{(6)}=M^{(4)}L\tilde{M}^{(4)}$; $M^{(7)}=M^{(6)}L\tilde{M}^{(6)}$. Matricea $M^{(7)}$ va da toate drumurile hamiltoniene din vârful cercetat.

Dăm ca exemplu în tabelul 3.2.2.6, matricea $\tilde{M}^{(2)}$, cu ajutorul căreia se regăsește, direct $M^{(4)}$. Produsul latin a două secvențe va fi făcut, în acest. caz, prin alăturarea lor, ținând seama de ordine, (prelungirea unei secvențe, cu altă secvență, nu, cu un vârf).

Tabelul 3.2.2.6.

	X ₁	X ₂	X ₃	X ₄	X ₅
X ₁	∅	∅	X ₂ X ₃ X ₅ X ₃	∅	∅
X ₂	X ₃ X ₁	∅	∅	X ₃ X ₄	∅
X ₃	∅	X ₁ X ₂ X ₄ X ₂	∅	∅	X ₁ X ₅ X ₄ X ₅
X ₄	X ₅ X ₁	∅	X ₂ X ₃ X ₅ X ₂	∅	∅
X ₅	X ₂ X ₁	X ₁ X ₂	∅	X ₃ X ₄	∅

Observația 3.2.2.4. Algoritmul produsului latin este foarte general, aplicabil grafurilor cu sau fără circuite, dar este greu de aplicat pentru un graf cu un număr mare de vârfuri, datorită scrierii unor secvențe lungi și în număr mare, în căsuțele matricelor. De aceea, propunem aplicarea prealabil și a algoritmului ce folosește componentele tare conexe, urmând ca în aceste componente să se determine drumurile hamiltoniene cu ajutorul produsului latin. De altfel, în exemplul luat mai sus, graful este tare conex și deci are o singură componentă tare conexă ceea ce impune determinarea drumurilor hamiltoniene numai cu metoda produsului latin.

Continuăm studiul drumurilor hamiltoniene în grafuri orientate, cu problema determinării unui ciclu hamiltonian de valoare minimă (CHM) într-un graf complet (care conține toate arcele posibile, adică între oricare două vârfuri există două arce de sensuri contrare). Fie $G = (V, E, w)$ un graf orientat cu n vârfuri, în care mulțimea E conține toate arcele (x_i, x_j) , cu $x_i, x_j \in V$. Graful este ponderat funcția $w : E \rightarrow \mathbb{R}^+$ având valori ce le vom nota convențional, astfel:

$$w(x_i, x_j)=c_{ij}, \text{ pentru } (x_i, x_j) \in E.$$

Valorile c_{ij} reprezintă costurile deplasării pe arce între vârfurile corespunzătoare, x_i și x_j .

Se poate deduce cu ușurință că numărul de drumuri, reprezentând circuite hamiltoniene în graful G , este egal cu $(n - 1)!$. Dintre acestea, trebuie găsit cel care are valoarea cea mai mică, respectiv cele (dacă sunt mai multe) cu valori minime. Prezentăm, în cele de mai jos un algoritm pentru aceasta, în care se folosește un procedeu de ramificare a cazurilor ce apar pe parcurs. Figura obținută în urma acestor ramificări poartă numele de arborescență. O arborescență are un vârf de plecare, numit sursa (rădăcină), de la care pleacă arce spre anumite vârfuri; de la fiecare dintre aceste vârfuri pornesc, de asemenea, arce spre

alte vârfuri s.a.m.d., ultimele vârfuri la care se ajunge fiind așa-numitele vârfuri (noduri) terminale ale arborescenței.

3.2.3. Algoritmul lui Little

Fie $S(0)$ mulțimea tuturor itinerariilor posibile din G , deci având $(n-1)!$ drumuri, reprezentând circuite hamiltoniene. Little și colaboratorii săi (Murty, Karel și Sweeney) au pornit de la ideea folosirii unei matrici de „costuri transformate”, matricea c_{ij}^* ; $i, j = 1, 2, \dots, n$ cu proprietatea că are cel puțin un zero pe fiecare linie și pe fiecare coloană. Evident, dacă s-ar putea găsi un circuit hamiltonian având numai costuri transformate nule, acesta ar fi un circuit optim, al cărui „cost”, γ , ar fi suma totală a reducerilor făcute (minorității pe linii, și coloane), din matricea inițială, ca la metoda repartizării de la problemele de alocări. Fiecare circuit hamiltonian din $S(0)$ va costa, deci, cel puțin γ . Mulțimea $S(0)$ se partiționează în două submulțimi disjuncte și se calculează câte o margine inferioară pentru fiecare din ele. Într-o etapă oarecare a algoritmului se alege submulțimea cu marginea inferioară mai mică și se împarte în două submulțimi disjuncte până la obținerea unei submulțimi care conține un singur circuit hamiltonian (drum), al cărui „cost” este marginea sa inferioară și este deci drumul optim (celelalte circuite hamiltoniene au margini inferioare mai mari). Este vorba, deci, de un procedeu de separare și evaluare progresivă (P.S.E.P.) sau „branch and bound” (ramifică și mărginește), constând din construirea unei arborescențe, în care succesorii fiecărui nod se stabilesc în funcție de evaluarea unei margini inferioare a drumurilor respective, adică în orice etapă se continuă construcția arborescenței alegând ca punct de ramificație al arborescenței vârful cu cea mai mică margine inferioară a costurilor, până la obținerea unui circuit hamiltonian optim.

Descrierea algoritmului lui Little

Fie $C = \|C_{ij}\|$, $i, j = 1, 2, \dots, n$, matricea „costurilor”. Condiția ca un vârf atins într-un drum să fie părăsit, pentru ca drumul să fie continuat spre alte vârfuri, se va realiza prin condiția matematică $C_{ij} = \infty$.

Pasul 1. Se deduce matricea „costurilor transformate”: $\|C_{ij}^*\|$, $i, j = 1, 2, \dots, n$, astfel:

- se aleg minoranții pe linii din matricea C : $\alpha_i = \min_{i \leq j \leq n} C_{ij}$ și se deduce matricea:

$C' = \|C'_{ij}\|$, $i, j = 1, 2, \dots, n$, cu elementele $C'_{ij} = C_{ij} - \alpha_i$, $j = 1, 2, \dots, n$, deci matricea în care fiecare linie are cel puțin un element nul;

- se aleg minoranții pe coloanele matricei C' , adică $\beta_j = \min_{i \leq j \leq n} (C'_{ij})$ și se scad din

elementele fiecărei coloane, obținând $C^*_{ij} = C'_{ij} - \beta_j$ $i = 1, 2, \dots, n$, deci cel puțin un element nul în fiecare coloană.

Matricea redusă este:

$$C^* = \|C^*_{ij}\|, \quad i, j = 1, 2, \dots, n,$$

iar „reducere totală” este:

$$r = \sum_{i=1}^n \alpha_i + \sum_{j=1}^n \beta_j.$$

Pasul 2. Pentru fiecare element $C^*_{hk} = 0$ se înregistrează penalitatea p_{hk} , corespunzătoare nefolosirii acestui element, pornind de la următorul raționament: dacă nu folosim arcul (x_h, x_k) , trebuie să folosim un alt element din linia h și un anumit element din coloana k , așa încât, costul nefolosirii arcului (x_h, x_k) să fie cel puțin egal cu suma celor mai mici „costuri transformate” din linia h și coloana k , exceptând pe C^*_{hk} , adică:

$$P_{hk} = \min_{j \neq k} \{C^*_{hj}\} + \min_{i \neq h} \{C^*_{ik}\}$$

Se înregistrează penalitatea p_{hk} în fiecare celulă în care $C_{hk}^*=0$, într-un colț ca în schema din figura 3.2.3.1.

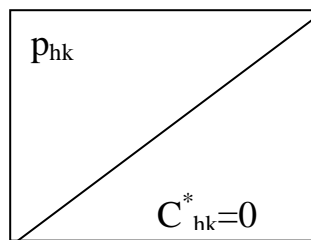


Fig. 3.2.3.1.

Pasul 3. Se alege celula cu penalitatea maximă; fie aceasta (h, k) , adică:

$$P_{hk} = \max_{c_{ij}^*=0} \{ p_{ij} \}$$

Dacă există mai multe celule cu penalitate maximă egală se alege una arbitrară dintre ele.

Mulțimea $S(0)$ se partiționează în două submulțimi disjuncte: $S(h,k)$, ce reprezintă mulțimea circuitelor hamiltoniene care trec prin arcul (h, k) , iar $\overline{S(h,k)}$ este mulțimea complementară.

Pasul 4. Se calculează marginile inferioare ale costurilor tuturor circuitelor în fiecare submulțime:

(4 a) Pentru submulțimea $S(h,k)$, deoarece prin nefolosirea arcului (x_h, x_k) , pe lângă reducerea r va apare un cost dat de penalitatea x_{hk} , marginea inferioară va fi:

$$\theta(h, k) = r + p_{hk}$$

(4 b) Pentru submulțimea $\overline{S(h, k)}$, se observă că, folosind arcul (x_h, x_k) nu se poate folosi arcul invers: (x_k, x_h) , pentru ca drumul să nu treacă de două ori prin același vârf. Pentru a evita utilizarea arcului (x_k, x_h) , înlocuim $C_{kh}^* = \infty$. De asemenea, trebuie să se țină seama de faptul că, alegând arcul (x_h, x_k) nu se mai poate utiliza nici un alt arc din linia h și din coloana k , deci se elimină linia h și coloana k din C^* . În matricea rămasă se calculează, reducerea referitoare la celula (h, k) , notată r_{hk} , reprezentând cantitatea cu care matricea rămasă poate fi redusă. Rezultă că marginea inferioară în submulțimea $\overline{S(h, k)}$ va fi:

$$\overline{\theta(h, k)} = r + r_{hk}$$

Cu aceasta, se, începe construcția arborescenței utilizate, de metodă, ca, în figura 3.2.3.2.

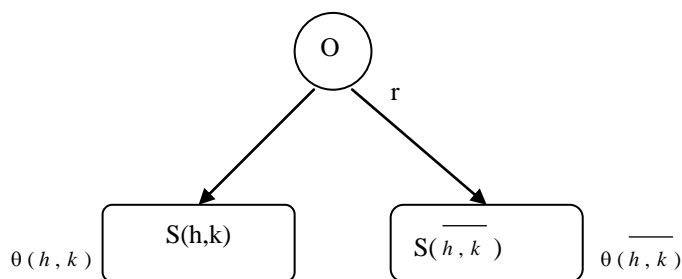


Fig.3.2.3.2.

Pasul 5. Se alege $\theta^* = \min \{ \overline{\theta(h, k)}, \theta(h, k) \}$.

Dacă $\theta^* = \overline{\theta(h, k)}$, se trece la pasul 2, folosind matricea redusă obținută în pasul (4 b).

Dacă $\theta^* = \theta(h, k)$, în matricea $\| c_{ij}^* \|$ se pune $c_{hk}^* = \infty$ și se reduce matricea rezultată, trecând cu ea la pasul 2.

Pasul 6. (urmează după pasul 2 în toate iterațiile, exceptând prima iterație, când după pasul 2 se execută pașii 3 și 4). Trebuie partiționată o mulțime $S(s, q)$ de margine inferioară θ' . Fie (u, v) celula în care $p_{uv} = \max p_{ij}$ din matricea redusă a etapei respective, în care liniile și coloanele corespunzând arcelor (m, n) , aparținând predecesorilor vârfului care trebuie ramificat, au fost eliminate.

Se face partiționarea în $S(u, v)$ și $\overline{S(u, v)}$ și se trece la pasul 7, ca în figura 3.2.3.3.

Pasul 7. Se calculează marginile, inferioare pentru submulțimile $S(u, v)$ și $\overline{S(u, v)}$ formate astfel:

(7 a) Pentru submulțimea $\overline{S(u, v)}$ marginea inferioară este:

$\overline{\theta(u, v)} = \theta' + p_{uv}$, unde θ' era marginea inferioară a mulțimii $S(s, q)$, pe care am partiționat-o.

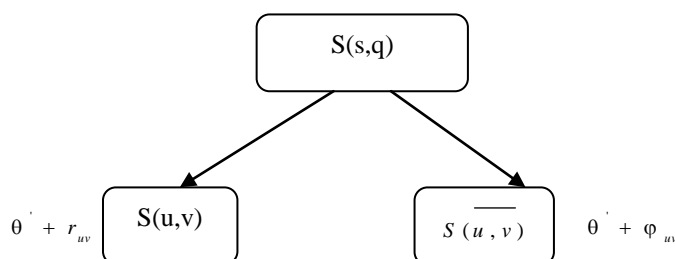


Fig. 3.2.3.3.

(7 b) Pentru $S(u, v)$ se elimină din matricea „costurilor transformate” a etapei respective linia u și coloana v , apoi se determină arcul (α, β) , care, împreună cu arcele incluse anterior în drumul de până aici ar forma un subciclu. Se alege deci costul $C^* = \infty$ și se reduce, iar matricea e astfel obținută, fie se face reducerea corespunzătoare r_{uv} . Se determină marginea inferioară a submulțimii $S(u, v)$ prin formula:

$$\theta(u, v) = \theta' + r_{uv}$$

Se continuă repetarea pașilor 5, 2, 6, 7 până la obținerea circuitului hamiltonian dorit, iar marginea inferioară a ultimei submulțimi (care nu mai poate fi ramificată sau partiționată) reprezintă „costul” minim al acestui circuit hamiltonian.

Observația 3.2.3.1. Problema poate avea mai multe soluții optime, continuând ramificările în vârfuluri abandonate în etape anterioare, care corespund la un moment dat unor margini inferioare minime.

Observația 3.2.3.2. Metoda este foarte generală, fiind aplicabilă la grafuri orientate complete, în care matricea inițială a costurilor C este nesimetrică.

În finalul paragrafului de față, vom prezenta metode de rezolvare a unor probleme de transport, cu ajutorul drumurilor hamiltoniene din grafuri neorientate. Una dintre noțiunile ajutoare importante, în determinarea unor astfel de drumuri, este aceea de arbore acoperire al unui graf dat.

Definiția 3.2.3.1. Fiind dat un graf G , conex și neorientat, cu n vârfuluri, se numește **arbore acoperire** al lui un graf parțial al lui G , ce este un arbore.

Observația 3.2.3.3. Vom nota, pentru ușurarea scrierii, muchiile grafului G prin e_1, e_2, \dots, e_m , înșiruirea lor fiind făcută fără a avea în vedere un criteriu anumit, ci printr-o convenție de numerotare. După cum se vede, s-a notat cu m numărul muchiilor, convenția ca G să fie conex impunând $m > n-1$. Este evident că aflarea unui arbore acoperire al grafului G se poate face considerând succesiv câte o muchie a grafului, care să nu formeze nici un ciclu, cu cele considerate până la momentul respectiv; după ce alegem în acest fel $(n-1)$ muchii, pe celelalte le suprimăm, fiind astfel conduși la un arbore acoperire al lui G .

Dacă graful G este ponderat, fiecărei muchii $[x_i, x_j]$ corespunzându-i o valoare w_{ij} , se poate pune problema determinării - dintre toți arborii acoperire posibili - a unuia (a unora) cu valoarea totală cea mai mică. În astfel de probleme, considerăm drept valoare a unui arbore acoperire T , suma valorilor de pe muchiile lui o notăm cu $w(T)$. Deoarece această problemă este importantă pentru practică, ea ajutând și la căutarea drumurilor hamiltoniene în grafuri neorientate, vom insista asupra rezolvării ei, prezentând o metodă în acest sens.

Fie un graf conex ponderat $G = (V, E, w)$ și fie w_{ij} ponderea muchiei $[x_i, x_j]$. Se pune problema găsirii **arborelui minimal (economic)**, adică **arborele acoperire** T pentru care avem:

$$\text{Min } w(T) = \sum_{(x_i, x_j) \in T} w_{ij}.$$

Lema Berge. Fie G un graf conex, finit, complet și cu valori distincte w_{ij} . Atunci se poate extrage din G un singur arbore economic.

Demonstrație. Arătăm existența arborelui minimal construindu-l efectiv în modul următor:

Se alege o muchie $e_1 = [x_i, x_j]$, cu $w_{ij} = w(e_1)$ cea mai mică dintre valorile funcției w și obținem arborele $\overline{T_1}$, format dintr-o singură muchie. Se alege apoi muchia e_2 , cu $e_2 \neq e_1$ și $w(e_2)$ cea mai mică dintre valorile muchiilor rămase după alegerea lui e_1 și se formează arborele $T_2 = \{e_1, e_2\}$. Se alege apoi e_3 , cu proprietățile: $e_3 \neq e_1, e_3 \neq e_2$, cu valoarea $w(e_3)$ cea mai mică din cele rămase, astfel ca să nu se formeze un ciclu și avem $T_3 = \{e_1, e_2, e_3\}$ ș.a.m.d., până se obține $\overline{T_{n-1}}$. Presupunem, prin absurd, că $\overline{T_{n-1}}$ nu este minim și deci că există $\overline{T_{n-1}}$, cu $w(\overline{T_{n-1}}) < w(T_{n-1})$ și $\overline{T_{n-1}} \neq T_{n-1}$. Rezultă că arborii diferă printr-o muchie; fie e_k muchia de cea mai mică valoare $w(e_k)$, în $\overline{T_{n-1}}$, și astfel se formează un graf, \tilde{G} , care conține un singur ciclu. În graful \tilde{G} eliminăm o altă muchie, diferită de e_k , fie aceasta e_p . Se obține un arbore cu valoarea:

$$w(\overline{T}) = w(T) + w(e_k) - w(e_p).$$

Cum am presupune că \overline{T} are cea mai mică valoare, deci $w(\overline{T}) < w(T)$, trebuie ca: $w(e_k) - w(e_p) > 0$, de unde $w(e_k) > w(e_p)$ și care contrazice faptul că s-a ales muchia e_k din $\overline{T_{n-1}}$ de cea mai mică valoare.

3.2.4. Algoritm euristic bazat pe arborele economic

Pornește de la arborele economic, care, în cazul existenței unor noduri de grad mai mare decât doi va micșora gradele acestor noduri, cu o creștere cât mai mică a lungimii totale a arborelui.

Fie $G = (V, E, w)$ un graf neorientat, complet cu n vârfuri și matricea valorilor W (de ordin $n \times n$).

Pasul 1. Se determină un arbore economic (arbore-acoperire de valoare minimă)

Dacă acest arbore economic are numai vârfuri de gradul 1 și 2, atunci el este un lanț hamiltonian, chiar un LHM.

Dacă arborele economic nu este lanț hamiltonian, se trece la pasul următor.

Pasul 2. Se micșorează treptat gradele vârfurilor care sunt cel puțin egale cu 3, astfel încât, creșterea valorii totale să fie minimalizată la fiecare iterație. Acest lucru se face prin operațiile A, B, C, de mai jos:

Observația 3.2.4.1. Scăderea treptată a gradelor vârfurilor x_i care au $g(x_i) \geq 3$, ducă în final la un lanț hamiltonian (adică un arbore cu gradele nodurilor numai 1 și 2), dar acesta nu este totdeauna un LHM al grafului G , dar este o valoare aproximativă apropiată de cea a unui LHM.

Operația A: Fie un nod z cu $g(z) \geq 3$ și fie x, y doua noduri terminale. Se introduce o muchie $[x, y]$ și se suprimă o muchie $[z, t]$ din ciclul format, cea care are cea mai mare valoare dintre muchiile adiacente lui z și aflate în acest ciclu.

Operația B: Fie un nod u cu $g(u) \geq 3$, adiacent cu un nod terminal y . Se aleg două noduri oarecare: u și v , ambele de grad cel mult 2 și se intercalează y între u și v , adică se introduc muchiile $[y, u]$ și $[x, v]$, suprimând muchiile $[z, y]$ și $[u, v]$.

Operația C: Fie un nod z cu $g(z) \geq 3$, adiacent cu un nod v neterminal. Dacă v nu se află pe lanțul unic care unește z cu un nod terminal y în arborele-acoperire, atunci se introduce muchia $[y, v]$ și se suprimă mușchia $[z, v]$.

Creșterile de cost ale arborelui sunt:

Operația A: $\alpha = w[x, y] - w[z, t]$

Operația B: $\beta = w[y, u] + w[y, v] - w[y, z] - w[u, v]$

Operația C: $w = w[y, v] - w[z, v]$.

Observația 3.2.4.2. Aplicarea repetată a acestor operații va ducă la obținerea unui lanț hamiltonian. Pentru ca lanțul hamiltonian obținut să fie cât mai mic, se aleg transformările care să minimizeze creșterea valorii arborelui în fiecare etapă.

Deci pasul 2 se aplică efectiv astfel:

(2 a) Se listează vârfurile terminale (de grad 1) din arborele economic obținut în pasul 1: y_1, y_2, \dots, y_k . Alegem toate perechile (y_i, y_j) , $j \neq i$ și se adaugă muchia $[y_i, y_j]$, suprimând o altă muchie adiacentă cu un nod z de grad $g(z) \geq 3$, cea mai lungă din ciclul format (operația A - figura 3.2.4.1.).

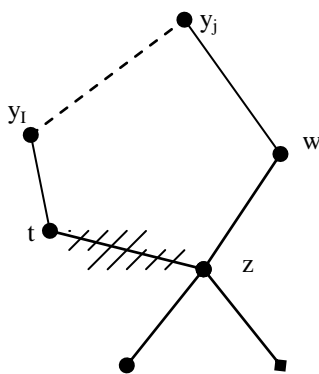


Fig. 3.2.4.1.

Se calculează creșterea minimă a lungimilor: $\alpha_0 = \min\{\alpha\}$, cu $\alpha = w(y_i, y_j) - w(z, t)$.

(2 b) Se aplică operația B (figura 3.2.4.2.), pentru toate vârfurile terminale y adiacente cu un nod z cu $g(z) \geq 3$ și se determină creșterea minimă a valorii lanțului pentru operațiile B: $\beta = \min\{\beta\}$

(2 c) Se aplică operația C (figura 3.2.4.3.), pentru toate vârfurile v neterminale și adiacente unui nod z de grad $g(z) \geq 3$, apoi se calculează creșterea minimă a valorii lanțului prin efectuarea operațiilor C: $Y_0 = \min\{Y\}$.

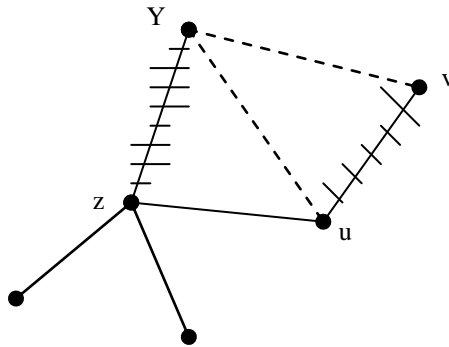


Fig. 3.2.4.2.

(2 d) Se alege $\min(\alpha_0, \beta_0, v_0)$ și se efectuează asupra arborelui operația corespunzătoare acestei valori. Când există mai multe operații care realizează această creștere minimă a valorii lanțului, se va alege cea transformare care produce scăderea gradului a două noduri de grad cel puțin trei (daca există).

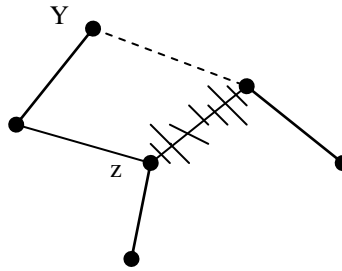


Fig. 3.2.4.3.

Dacă după transformarea efectuată, toate nodurile vor avea grad 1 și 2, algoritmul ia sfârșit (s-a obținut un lanț hamiltonian de valoare apropiată de cea minimă), în caz. contrar, se reia pasul 2, prin subpașii (2 a), (2 b), (2 c), (2 d).

Numărul maxim de iterații, până la obținerea unui lanț hamiltonian, este majorat de:

$$\sum_{g(z) \geq 3} [g(z) - 2]$$

Exemplul 3.2.4.1. Într-o problemă de colectare a corespondenței de la mai multe puncte, mașina pleacă de la centrul x_1 și trebuie să depună toate scrisorile colectate din punctele $x_2, x_4, x_5, x_6, x_7, x_8$ la poșta centrală x_3 . Trebuie determinat drumul optim care să înceapă de la unitatea x_1 și să se încheie la x_3 , trecând prin toate celelalte puncte de colectare a corespondenței. Datele sunt în tabelul 3.2.4.1 și reprezintă distanțele între puncte (în sute de metri).

Tabelul 3.2.4.1.

La De la	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇	x ₈
x ₁	0	2	7	15	10	8	9	21
x ₂	2	0	23	45	35	14	7	2
x ₃	7	23	0	8	11	17	13	22
x ₄	15	45	0	8	12	32	21	15
x ₅	10	35	11	12	0	42	38	19
x ₆	8	14	17	32	42	0	31	36
x ₇	9	7	13	21	38	31	0	18
x ₈	21	5	22	15	19	36	18	0

Rezolvare. Trebuie determinat un LHM_{1,3}, deci este o problemă de tipul P2. Pentru a aplica algoritmul prezentat trebuie transformată întâi problema într-o problema P₁, adăugând o constantă suficient de mare (de exemplu, k=300) la liniile 1 și 3, precum și la coloanele 1 și 3, exceptând elementele diagonale (penalizarea extremităților pentru lanțul hamiltonian dorit).

Datele transformate sunt redată în tabelul 3.2.4.2

Se determină arborele-economic reprezentat în figura 3.2.4.4. el având o valoare egală cu 660. Notăm: L = 660. Gradele vârfurilor sunt:

$$g(x_1)=1, g(x_4)=3, g(x_7)=1, g(x_2)=4, g(x_3)=1, g(x_8)=2, g(x_3)=1, g(x_6)=1,$$

Tabelul 3.2.4.2.

La De la	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇	x ₈
x ₁	0	302	607	315	310	308	309	321
x ₂	302	0	323	45	35	14	7	2
x ₃	607	323	0	308	311	317	313	322
x ₄	35	45	308	0	12	32	21	15
x ₅	310	35	311	12	0	42	38	19
x ₆	308	14	317	32	42	0	31	36
x ₇	309	7	313	21	38	31	0	18
x ₈	321	5	322	15	19	36	18	0

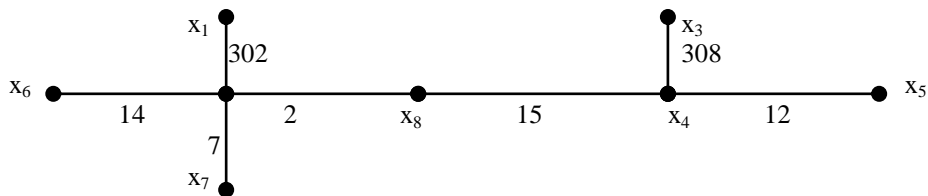


Fig. 3.2.4.4.

Nodurile x₂ și x₄ au gradele 4 și respectiv 3 trebuie să micșorăm aceste grade în pasul 2.

În cadrul primei iterații s-au efectuat calculele din tabelul 3.2.4.3, din care rezultă că avem:

$\min(\alpha_0, \gamma_0) = \alpha_0=3$. Se face operația A, de introducere a muchiei [x₃, x₄] și de suprimare a muchiei [x₃, x₄]. Rezultă. arborele din figura 3.2.4.5. de valoare L₁=L + 3 = 663.

Tabelul 3.2.4.3.

Operația	Muchii introduse	Muchii suprimate	Creșterea lungimii arborelui	Creșterea minimă la fiecare operație
A	$[x_1, x_3]$	$[x_3, x_4]$	$607-408=299$	* $\alpha_0=3$
	$[x_1, x_3]$	$[x_1, x_2]$	$607-302=305$	
	$[x_1, x_5]$	$[x_1, x_2]$	$310-302=8$	
	$[x_1, x_6]$	$[x_1, x_2]$	$308-302=6$	
	$[x_1, x_7]$	$[x_1, x_2]$	$309-302=7$	
	$[x_3, x_5]$	$[x_3, x_4]$	$311-308=3$	
	$[x_3, x_6]$	$[x_3, x_4]$	$317-308=9$	
	$[x_3, x_7]$	$[x_3, x_4]$	$313-308=5$	
	$[x_5, x_6]$	$[x_2, x_6]$	$42-14=28$	
	$[x_5, x_6]$	$[x_4, x_8]$	$42-15=27$	
	$[x_5, x_7]$	$[x_4, x_8]$	$38-15=223$	
	$[x_5, x_7]$	$[x_2, x_7]$	$38-7=31$	
$[x_6, x_7]$	$[x_2, x_6]$	$31-14=17$		
B Nu este aplicabilă la acest arbore.				
C	$[x_1, x_8]$	$[x_2, x_8]$	$321-2=319$	* $\gamma_0=4$
	$[x_7, x_8]$	$[x_2, x_8]$	$18-2=16$	
	$[x_3, x_8]$	$[x_2, x_8]$	$36-2=34$	
	$[x_3, x_8]$	$[x_4, x_8]$	$322-15=307$	
	$[x_5, x_8]$	$[x_4, x_8]$	$19-15=4$	

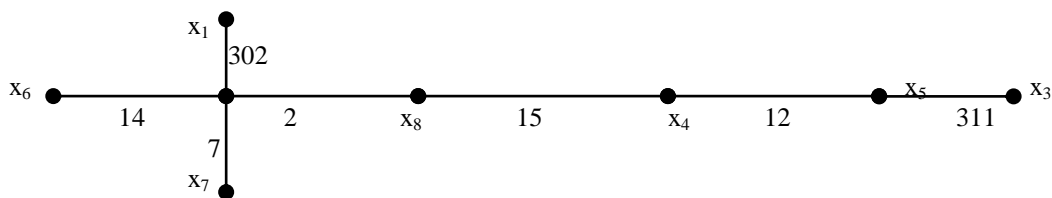


Fig. 3.2.4.5

Singurul nod de grad mai mare decât 2 este $x_2 : g(x_2)=4$. Calculele iterației a doua sunt redată în tabelul 3.2.4.4, tabel din care rezultă că avem:

$$\min(\alpha_0, \beta_0, \gamma_0)=6.$$

Tabelul 3.2.4.4.

Operația	Muchii introduse	Muchii suprimate	Creșterea lungimii arborelui	Creșterea minimă la fiecare operație
A	$[x_1, x_3]$	$[x_1, x_2]$	$607-302=305$	* $\alpha_0=6$
	$[x_1, x_6]$	$[x_1, x_2]$	$308-302=6$	
	$[x_2, x_7]$	$[x_1, x_2]$	$309-302=7$	
	$[x_3, x_5]$	$[x_2, x_6]$	$317-14=303$	
	$[x_3, x_7]$	$[x_2, x_7]$	$313-7=306$	
	$[x_6, x_7]$	$[x_2, x_6]$	$31-14=17$	
B	$[x_1, x_8], [x_1, x_4]$	$[x_1, x_2], [x_4, x_5]$	$(321+315)-(302+15)=319$	

	$[x_1, x_4], [x_1, x_5]$	$[x_1, x_2], [x_4, x_5]$	$(315+310)-$ $(302+12)=311$	$\beta_0=7$ *
	$[x_1, x_5], [x_1, x_3]$	$[x_1, x_2], [x_3, x_5]$	$(310+607)-$ $(302+311)=304$	
	$[x_6, x_8], [x_4, x_5]$	$[x_2, x_6], [x_4, x_8]$	$(36+32)-$ $(14+15)=39$	
	$[x_4, x_6], [x_5, x_6]$	$[x_3, x_6], [x_4, x_5]$	$(32+42)-$ $(14+12)=48$	
	$[x_5, x_6], [x_3, x_6]$	$[x_2, x_6], [x_3, x_5]$	$(32+317)-$ $(14+311)=34$	
	$[x_7, x_8], [x_4, x_7]$	$[x_2, x_7], [x_4, x_8]$	$(18+21)-$ $(7+15)=17$	
	$[x_4, x_7], [x_5, x_7]$	$[x_2, x_7], [x_4, x_5]$	$(21+38)-$ $(7+12)=40$	
	$[x_5, x_7], [x_3, x_7]$	$[x_2, x_7], [x_3, x_5]$	$(38+313)-$ $(7+311)=33$	
	$[x_1, x_8]$	$[x_2, x_8]$	$321-2=319$	$\gamma_0=16$ *
	$[x_6, x_8]$	$[x_2, x_8]$	$36-2=34$	
	$[x_7, x_8]$	$[x_2, x_8]$	$18-2=16$	

Se execută operația A corespunzătoare, adică muchia $[x_1, x_2]$ este suprimată și se introduce în arbore muchia $[x_1, x_6]$, obținând arborele din figura 3.2.4.6, de valoare $L_2=L_1 + 6=669$, în care nodul x_2 are gradul mai mic, anume $g(x_2)=3$.

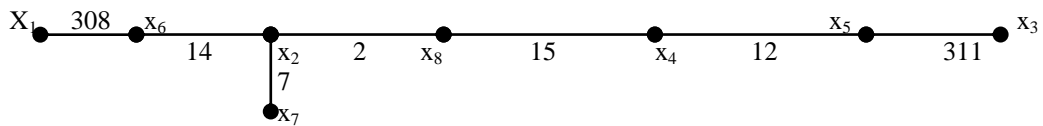


Fig. 3.2.4.6.

Trebuie micșorat gradul lui x_2 , singurul nod de grad mai mare decât 2, în următoarea iterație. Calculele corespunzătoare sunt redată în tabelul 3.2.4.5.. Avem:

$$\min(\alpha_0, \beta_0, \gamma_0) = \gamma_0 = 16$$

Tabelul 3.2.4.5

Operația	Muchii introduse	Muchii suprimate	Creșterea lungimii arborelui	Creșterea minimă la fiecare operație
A	$[x_1, x_3]$	$[x_2, x_7]$	$607-14=593$	$\alpha_0=295$
	$[x_1, x_7]$	$[x_2, x_6]$	$309-14=295$	
	$[x_3, x_7]$	$[x_2, x_6]$	$313-7=306$	
B	$[x_1, x_7], [x_6, x_7]$	$[x_2, x_7], [x_1, x_6]$	$(309+21)-$ $(7+308)=25$	$\beta_0=17$
	$[x_7, x_8], [x_4, x_7]$	$[x_2, x_7], [x_4, x_8]$	$(18+21)-$ $(7+15)=17$	
	$[x_4, x_7], [x_5, x_7]$	$[x_2, x_7], [x_4, x_5]$	$(21+38)-$ $(7+12)=40$	
	$[x_5, x_7], [x_3, x_7]$	$[x_2, x_7], [x_3, x_5]$	$(38+313)-$ $(7+311)=33$	
C	$[x_6, x_7]$	$[x_2, x_6]$	$31-14=17$	$\gamma_0=16$
	$[x_3, x_6]$	$[x_2, x_6]$	$317-14=303$	
	$[x_7, x_6]$	$[x_2, x_6]$	$18-2=16$	
	$[x_1, x_6]$	$[x_3, x_6]$	$321-2=319$	

Se execută operația C de suprimare a muchiei $[x_2, x_6]$ și introducerea muchiei $[x_7, x_8]$, obținând arborele din figura 3.2.4.7, în care toate nodurile au gradele 1 sau 2, deci este lanțul hamiltonian căutat, de valoare $L = L_2 + 16 = 635$. Se scad valorile $k = 300$, de pe muchiile $[x_1, x_6]$ și $[x_3, x_5]$, obținând LHM_{13} de valoare=85, în probleme de colectare inițială (vezi figura 3.2.4.8).

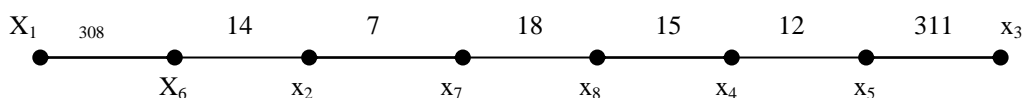


Fig. 3.2.4.7.

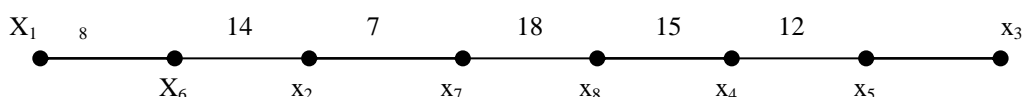


Fig. 3.2.4.8.

Direct, calculând toate valorile lanțurilor hamiltoniene de extremități fixate, x_1 și x_3 , problema ar avea mari dificultăți de calcul, existând $6! = 720$ lanțuri hamiltoniene cu aceste extremități.

3.3. Aplicații

Aplicația 3.3.1. Un produs finit poate fi verificat în privința respectării indicilor de calitate pe la șapte puncte de control. Procesul de producție este automatizat și trecerea se face pe bandă rulantă. Cum fiecare punct de control verifică o anumită calitate a produsului, bine determinată, găsiți posibilitatea de a folosi într-o anumită ordine aceste puncte de control, astfel încât verificarea sa fie completă. Graful corespunzător se găsește în figura 3.3.1.

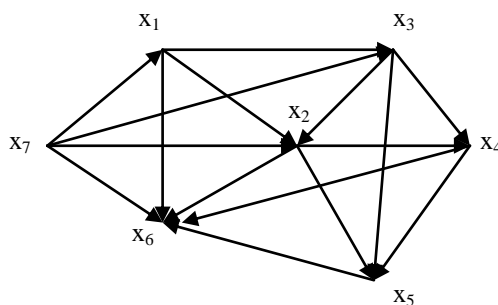


Fig. 3.3.1.

Rezolvare. Graful nu are circuite. Într-adevăr, folosind matricea arcelor din tabelul 3.3.1, se obține matricea drumurilor din tabelul 3.3.2, aceasta neavând nici un element egal cu unu pe diagonala principală. În tabelul 3.3.2. sunt calculate și puterile de atingere ale vârfurilor grafului.

Tabelul 3.3.1

<i>la</i> <i>De la</i>	x_1	x_2	x_3	x_4	x_5	x_6	x_7
x_1	0	0	1	0	1	1	0
x_2	0	0	0	1	1	1	0
x_3	0	1	0	1	1	0	0
x_4	0	0	0	0	1	1	0
x_5	0	0	0	0	0	1	0
x_6	0	0	0	0	0	0	0
x_7	1	1	1	0	0	1	0

(A):

Tabelul 3.3.2.

<i>la</i> <i>De la</i>	x_1	x_2	x_3	x_4	x_5	x_6	x_7	$P(x_i)$
x_1	0	1	1	1	1	1	0	5
x_2	0	0	0	1	1	1	0	3
x_3	0	1	0	1	1	1	0	4
x_4	0	0	0	0	1	1	0	2
x_5	0	0	0	0	0	1	0	1
x_6	0	0	0	0	0	0	0	0
x_7	1	1	1	1	1	1	0	6

(D):

După cum se vede, avem:

$$\sum_{i=1}^7 p(x_i) = 21 = \frac{7 \cdot 6}{2}$$

Ca urmare, graful dat are un drum hamiltonian, acesta având forma următoare:

$$DH=(x_7, x_1, x_3, x_2, x_4, x_5, x_6).$$

Aplicația 3.3.2. Într-o hală industrială procesul de producție se desfășoară pe un sistem de benzi rulante, care, leagă șase puncte de lucru, ca în figura 3.3.2. În fiecare din aceste puncte de lucru se execută câte o anumită operație. Întreprinderea, asimilează un produs nou, care trebuie să treacă prin toate cele șase operații, deci pe la toate punctele de lucru. Arătați dacă sistemul de benzi rulante existent în hală permite acest lucru, iar dacă nu, indicați o modificare tehnică cât mai simplă care să ducă la soluționarea problemei.

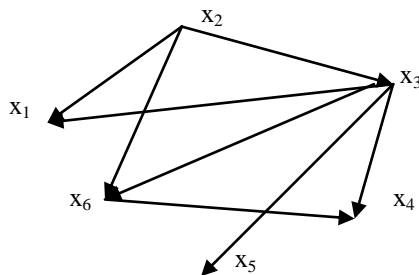


Fig. 3.3.2.

Rezolvare. Matricea arcelor grafului, din tabelul 3.3.3, conduce la matricea drumurilor, din, tabelul 3.3.4. Cercetând acest din urmă tabel, observăm că graful este fără

circuite și $\sum_{i=1}^6 p(x_i) = 12 \neq \frac{6 \cdot 5}{2}$. În consecință nu există drum hamiltonian. Pentru ca produsul să poată trece pe la toate punctele de lucru, căutăm să scriem forma triangularizată, ordonând vârfurile care au aceeași putere de atingere după criteriul unui număr mai mic de zerouri în coloana corespunzătoare din D, deoarece astfel numărul elementelor egale cu zero situate imediat deasupra diagonalei principale va fi mai mic.

Tabelul 3.3.3.

(A):

	x_1	x_2	x_3	x_4	x_5	x_6
x_1	0	0	0	0	0	0
x_2	1	0	1	0	0	1
x_3	1	0	0	1	1	1
x_4	0	0	0	0	1	0
x_5	0	0	0	0	0	0
x_6	0	0	0	1	0	0

Tabelul 3.3.4.

(D):

	x_1	x_2	x_3	x_4	x_5	x_6	$P(x_i)$
x_1	0	0	0	0	0	0	0
x_2	1	0	1	1	1	1	5
x_3	1	0	0	1	1	1	4
x_4	0	0	0	0	1	0	1
x_5	0	0	0	0	0	0	0
x_6	0	0	0	1	1	0	2

$$\sum p(x_i) = 12$$

Alegem ordinea ($x_2, x_3, x_6, x_4, x_5, x_1$) care va da un drum hamiltonian dacă se construiește o bandă rulantă între punctele x_5 și x_1 , după cum rezultă din forma triangularizată ce se găsește în tabelul 3.3.5

Cu construcția benzii de la x_5 la x_1 produsul trece prin toate cele 6 operații, în ordinea: $x_2, x_3, x_6, x_4, x_5, x_1$.

Tabelul 3.3.5.

(D):

	x_1	x_2	x_3	x_4	x_5	x_6
x_1	0	1	1	1	1	1
x_2	0	0	1	1	1	1
x_3	0	0	0	1	1	0
x_4	0	0	0	0	1	0
x_5	0	0	0	0	0	0
x_6	0	0	0	0	0	0

Dacă nu am fi respectat regula de a ordona vârfurile cu aceeași putere de atingere după numărul mai mic de zerouri din coloana corespunzătoare din D, atunci ordinea era: x_2, x_3, x_4, x_1, x_5 , dar se construiau două benzi rulante, între x_4 și x_1 , ca și între x_1 și x_5 , deci nu constituie cea mai simplă soluție tehnică.

Aplicația 3.3.3. Un inspector trebuie să facă un control asupra celor cinci unități de alimentație publică din cadrul unei stațiuni montane. Posibilitățile de acces între aceste unități sunt reprezentate în tabelul-graf din figura 3.3.3., prin arce. Se pune problema, cum trebuie să procedeze inspectorul pentru a controla toate unitățile, deci care este unitatea cu care trebuie să înceapă controlul și ordinea în care va controla celelalte unități.

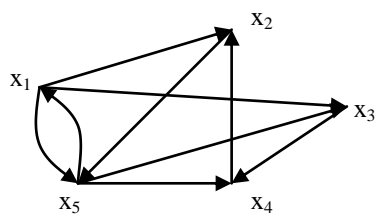


Fig. 3.3.3.

Rezolvare. Graful are, evident, circuite. Să aplicăm algoritmul Kaufmann. Pentru aceasta se folosesc matricile $M^{(1)}$ și $\tilde{M}^{(1)}$ din tabelele 3.3.6 și 3.3.7.

Tabelul 3.3.6.

$M^{(1)}$:

\emptyset	x_1x_2	x_1x_3	\emptyset	x_1x_5
\emptyset	\emptyset	\emptyset	\emptyset	x_2x_5
\emptyset	\emptyset	\emptyset	x_3x_4	\emptyset
\emptyset	x_4x_2	\emptyset	\emptyset	\emptyset
x_5x_1	\emptyset	x_5x_3	x_5x_4	\emptyset

Tabelul 3.3.6'.

$\tilde{M}^{(1)}$:

\emptyset	x_2	x_3	\emptyset	x_5
\emptyset	\emptyset	\emptyset	\emptyset	x_5
\emptyset	\emptyset	\emptyset	x_4	\emptyset
\emptyset	x_2	\emptyset	\emptyset	\emptyset
x_1	\emptyset	x_3	x_4	\emptyset

Cu rezultatele intermediare din tabelele 3.3.7 și 3.3.8 se obține rezultatul final din tabelul 3.3.9

Tabelul 3.3.7.

$M^{(2)}$:

\emptyset	\emptyset	$x_1x_2x_3$	$x_1x_3x_4$ $x_1x_5x_4$	$x_1x_2x_5$
$x_2x_5x_1$	\emptyset	$x_2x_5x_3$	$x_2x_3x_5$	\emptyset
\emptyset	$x_3x_4x_2$	\emptyset	\emptyset	\emptyset
\emptyset	\emptyset	\emptyset	\emptyset	$x_4x_2x_5$
\emptyset	$x_5x_1x_2$ $x_5x_4x_2$	$x_5x_1x_3$	$x_5x_3x_4$	\emptyset

Tabelul 3.3.8.

$M^{(3)}$:

\emptyset	$x_1x_3x_4x_2$ $x_1x_5x_4x_2$	$x_1x_2x_5x_3$	$x_1x_5x_3x_4$ $x_1x_2x_5x_4$	\emptyset
$x_2x_5x_1$	\emptyset	$x_2x_5x_1x_3$	$x_2x_5x_3x_4$	\emptyset
\emptyset	\emptyset	\emptyset	\emptyset	$x_3x_4x_2x_5$
\emptyset	\emptyset	$x_4x_2x_5x_2$	\emptyset	\emptyset

\emptyset	$x_5x_3x_4x_2$	\emptyset	$x_5x_1x_3x_4$	\emptyset
-------------	----------------	-------------	----------------	-------------

Tabelul 3.3.9.

$M^{(4)}$:

\emptyset	$x_1x_5x_3x_4x_2$	\emptyset	$x_1x_2x_5x_3x_4$	$x_1x_3x_4x_2x_5$
\emptyset	\emptyset	\emptyset	$x_2x_5x_1x_3x_4$	\emptyset
$x_3x_4x_2x_5$	\emptyset	\emptyset	\emptyset	\emptyset
\emptyset	\emptyset	$x_4x_2x_5x_1x_3$	\emptyset	\emptyset
\emptyset	$x_5x_1x_3x_4x_2$	\emptyset	\emptyset	\emptyset

Matrice $M^{(4)}$ dă toate drumurile hamiltoniene, în număr de șapte. Se vede că se poate începe controlul cu oricare unitate. Între aceste variante se alege una după un anumit criteriu de optim. Cele șapte drumuri hamiltoniene sunt următoarele:

$$DH_1=(x_1, x_5, x_3, x_4, x_2),$$

$$DH_2=(x_1, x_2, x_5, x_3, x_4),$$

$$DH_3=(x_1, x_3, x_4, x_2, x_5),$$

$$DH_4=(x_2, x_5, x_1, x_3, x_4),$$

$$DH_5=(x_3, x_4, x_2, x_3, x_1),$$

$$DH_6=(x_4, x_2, x_5, x_1, x_3),$$

$$DH_7=(x_5, x_1, x_3, x_4, x_2),$$

Aplicația 3.3.4. Trebuie stabilit drumul unui raliu care să treacă prin zece localități. Șoselele existente între localitățile respective au fie sens unic, fie ambele sensuri de circulație și au ca model un graf cu imaginea din figura 3.3.4. Trebuie stabilit din ce localitate trebuie să pornească raliul și care este drumul prin celelalte localități. Dacă s-a cronometrat și timpul mediu de parcurgere a fiecărei porțiuni de șosea cu automobilul (în ore), care este decizia cea mai bună de stabilire a drumului?

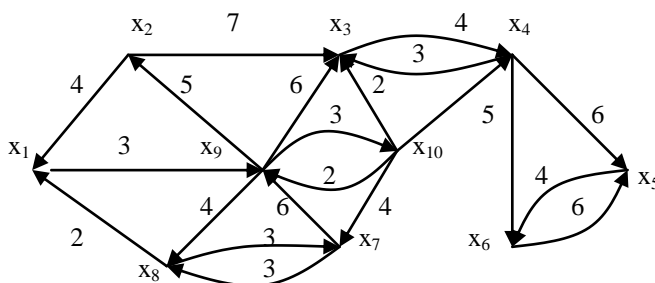


Fig.3.3.4.

Rezolvare. Trebuie cercetat dacă acest graf cu circuite admite existența unui drum hamiltonian, iar în cazul existenței a mai multor drumuri hamiltoniene trebuie determinat cel minim (DHM).

Algoritmul Kaufmann este mai greu de aplicat pentru întregul graf, deoarece matricea $M^{(9)}$ va avea ordinul 10×10 și în fiecare căsuță se poate trece \emptyset sau o succesiune de 10 litere. De aceea, construim întâi graful condensat și stabilim drumurile hamiltoniene în fiecare componentă tare conexă cu algoritmul lui Kaufmann.

Tabelul 3.3.8. conține calculele pentru determinarea componentelor tare conexe.

Tabelul 3.3.8.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	W'_1	W'_2	W'_3
--	-------	-------	-------	-------	-------	-------	-------	-------	-------	----------	--------	--------	--------

A:

x ₁	0	0	0	0	0	0	0	0	1	0	1	x	x
x ₂	1	0	1	0	0	0	0	0	0	0	1	x	x
x ₃	0	0	0	1	0	0	0	0	0	0	0	1	x
x ₄	0	0	1	0	1	1	0	0	0	0	0	1	x
x ₅	0	0	0	0	0	0	0	0	0	0	0	0	1
x ₆	0	0	0	0	1	1	0	0	0	0	0	0	1
x ₇	0	0	0	0	0	0	0	0	1	1	0	1	x
x ₈	1	0	0	0	0	0	1	0	0	0	1	x	x
x ₉	0	1	1	0	0	0	0	1	0	1	1	x	x
x ₁₀	0	0	1	1	0	0	1	0	1	0	1	x	x
W ₁	1	1	1	1	1	1	1	1	1	1			
W ₂	x	x	1	1	1	1	x	x	x	x			
W ₃	x	x	x	x	1	1	x	x	x	x			

Avem:

$$C_1 = C(x_1) = (W_1 \cap W'_1) \cup \{x_1\} = \{x_1, x_2, x_7, x_8, x_9, x_{10}\}$$

$$C_2 = C(x_3) = (W_2 \cap W'_2) \cup \{x_3\} = \{x_3, x_4\}$$

$$C_3 = C(x_5) = (W_3 \cap W'_3) \cup \{x_5\} = \{x_5, x_6\}$$

Cu ajutorul tabelor 3.3.9 și 3.3.10, s-a calculat matricea arcelor grafului condensat, din tabelul 3.3.11.

Tabelul 3.3.9.

	C ₁	C ₂	C ₃
x ₁	1	0	0
x ₂	1	1	0
x ₃	0	1	0
x ₄	0	1	1
(\tilde{A}): x ₅	0	0	1
x ₆	0	0	1
x ₇	1	1	0
x ₈	1	0	0
x ₉	1	1	0
x ₁₀	1	1	0

Tabelul 3.3.10.

	C ₁	C ₂	C ₃
C ₁	1	1	0
(\tilde{A}) C ₂	0	1	1
C ₃	0	0	1

Tabelul 3.3.11.

	C ₁	C ₁	C ₁
(A [*]): C ₁	0	1	0
C ₂	0	1	1
C ₃	0	0	0

Graful condensat are, deci, imaginea din figura 3.3.5.

Deoarece G^* nu are circuite, determinăm matricea D^* și puterile de atingere ale celor trei „vârfuri”, C_1, C_2, C_3 . Această matrice se găsește în tabelul 3.3.12.

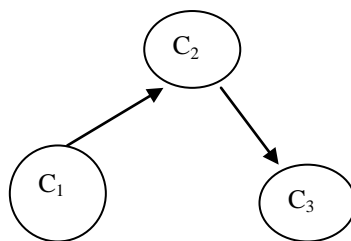


Fig. 3.3.5

Tabelul 3.3.12

	C_1	C_2	C_3	$P(C_i)$
(D^*) : C_1	0	1	1	2
C_2	0	0	0	1
C_3	0	0	0	0

Avem:

$$\sum_{i=1}^3 P(C_i) = 2 + 1 + 0 = 3 = \frac{3 \cdot 2}{2},$$

deci, există $DH^* = (C_1, C_2, C_3)$.

În C_1 aplicăm algoritmul Kaufmann pentru determinarea drumurilor hamiltoniene în acest subgraf, cu șase vârfuri. Pentru simplitatea scrierii, convenim să trecem secvențele numai prin intermediul indicilor, adică ij în loc de $x_i x_j$.

În tabelele 3.3.13-3.3.18. sunt efectuate calculele, ultima matrice furnizând rezultatul.

Astfel:

din matricea $-M^{(5)}$ citim cele două drumuri hamiltoniene din componenta C_1 anume:

$$DH_1 = \{x_2, x_1, x_9, x_{10}, x_7, x_8\}$$

$$DH_2 = \{x_{10}, x_7, x_8, x_1, x_9, x_2\}$$

Tabelul 3.3.13.

	x_1	x_2	x_7	x_8	x_9	x_{10}
$M^{(1)}$ x_1	\emptyset	\emptyset	\emptyset	\emptyset	1,9	\emptyset
x_2	2,1	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset
x_7	\emptyset	\emptyset	\emptyset	\emptyset	7,8	7,9
x_8	8,1	\emptyset	8,7	\emptyset	\emptyset	\emptyset
x_9	\emptyset	9,2	\emptyset	9,8	\emptyset	9,10
x_{10}	\emptyset	\emptyset	10,7	\emptyset	10,9	\emptyset

Tabelul 3.3.14.

	x_1	x_2	x_7	x_8	x_9	x_{10}
$M^{(1)}$ x_1	\emptyset	\emptyset	\emptyset	\emptyset	9	\emptyset
x_2	1	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset
x_7	\emptyset	\emptyset	\emptyset	8	9	\emptyset
x_8	1	\emptyset	7	\emptyset	\emptyset	\emptyset
x_9	\emptyset	2	\emptyset	8	\emptyset	10
x_{10}	\emptyset	\emptyset	7	\emptyset	9	\emptyset

Tabelul 3.3.15.

	X ₁	X ₂	X ₇	X ₈	X ₉	X ₁₀	
M ⁽²⁾ :	X ₁	∅	1,9,2	∅	1,9,8	∅	1,9,10
	X ₂	1	∅	∅	∅	2,1,9	∅
	X ₇	7,8,1	7,9,2	∅	7,9,8	∅	7,9,10
	X ₈	∅	∅	∅	∅	8,1,19 8,7,9	∅
		9,2,1		9,8,7			
	X ₉	9,8,1	∅	9,10,7	∅	∅	∅
	X ₁₀	∅	10,9,2	∅	10,7,8 10,9,8	10,7,9	∅

Tabelul 3.3.16.

	X ₁	X ₂	X ₇	X ₈	X ₉	X ₁₀	
M ⁽²⁾ :	X ₁	∅	∅	1,9,8 1,9,10,7	∅	∅	∅
	X ₂	∅	∅	∅	2,1,9,8	∅	2,1,9,10
		7,9,2,1	∅	∅	∅	7,8,1,9	∅
	X ₇	7,9,8,1					
	X ₈	∅	8,1,9,1 8,7,9,2				8,1,9,10 8,7,9,10
	X ₉	∅	∅	0	9,10,9,8	∅	∅
		10,9,2,1					
	X ₁₀	10,8,7,1 10,9,8,1	10,7,9,2		10,7,9,8	∅	∅
				10,9,8,7			

Tabelul 3.3.17

	X ₁	X ₂	X ₇	X ₈	X ₉	X ₁₀	
M ⁽⁴⁾ :	X ₁	∅	∅	∅	1,9,10,7,8	∅	∅
	X ₂	∅	∅	2,1,9,8,7	∅	∅	∅
				2,1,9,10,7			
	X ₇	∅	7,8,1,9,2	∅	∅	∅	7,8,1,9,10
	X ₈	8,7,9,2,1	∅	∅	∅	∅	∅
	X ₉	9,10,7,8,1	∅	0	∅	∅	∅
		10,7,9,2,1					
	X ₁₀	10,7,9,8,1	∅	∅	∅	10,7,8,1,9	∅

Tabelul 3.3.18.

	X ₁	X ₂	X ₇	X ₈	X ₉	X ₁₀	
M ⁽⁵⁾ :	X ₁	∅	∅	∅	∅	∅	
	X ₂	∅	∅	∅	2,1,9,10,7,8	∅	∅
	X ₇	∅	∅	∅	∅	∅	∅
	X ₈	∅	∅	∅	∅	∅	∅
	X ₉	∅	∅	∅	∅	∅	∅
	X ₁₀	∅	10,7,8,1,9,2	∅	∅	∅	∅

Componentele C_2 și C_3 au numai câte două vârfuri, deci drumurile lor hamiltoniene vor fi:

(x_3, x_4) și (x_4, x_3) în C_2

(x_5, x_6) și (x_6, x_5) în C_3

Cercetăm legăturile între ultimul vârf din DH_1 și DH_2 din C_1 , și primul vârf din componentele ce urmează în DH^* , anume C_2 , apoi între ultimul vârf din C_2 și primul din componenta ce sfârșește DH^* în G^* , adică, C_3 . Avem schema din tabelul 3.3.19.

Tabelul 3.3.19.

C_1	C_2	C_3
$x_2, x_1, x_9, x_{10}, x_7, x_8$ $x_{10}, x_7, x_8, x_1, x_9, x_2$	x_3, x_4 x_4, x_3	x_5, x_6 x_6, x_5

Rezultă două drumuri hamiltoniene pentru graful inițial:

$DH_1 = (x_{10}, x_7, x_8, x_1, x_9, x_2, x_3, x_4, x_5, x_6)$,

de valoare egală cu 40 ore,

$DH_2 = (x_{10}, x_7, x_8, x_1, x_9, x_2, x_3, x_4, x_5, x_6)$

de valoare egală cu 41 ore, deci trebuie ales raliul pe drumul dat de DH_1 .

BIBLIOGRAFIE

1. **C. Bălcău**, *Combinatorică și teoria grafurilor*, Editura Universității din Pitești, Pitești, 2007.
2. **E. Cerchez, M. Șerban**, *Programarea în Limbajul C / C++ pentru liceu*, POLIROM, 2006;
3. **E. Ciurea**, *Algoritmi. Introducere în algoritmica grafurilor*, Editura Tehnică, București, 2001;
4. **T.H. Cormen, C.E. Leiserson, R.R. Rivest**, *Introducere în Algoritmi*, Computers Libris Agora, 2001;
5. **L. Ezechil**, *Prelegeri de didactica generala*, Editura Paralela 45 Educational, Pitești 2003 ;
6. **D.E. Knuth**, *Tratat de programarea calculatoarelor. Algoritmi fundamentali*, Editura Tehnică, București, 1974;
7. **D. Logofătu**, *Algoritmi fundamentali în C++. Aplicații*, POLIROM, 2007;
8. **G.D. Mateescu, P.F. Moraru, C. Popescu**, *Informatică. Teste și variante de subiecte pentru bacalaureat*, Editura Donaris, Sibiu, 2001;
9. **O. Pătrășcoiu, Gh. Marian, N. Mitroi**, *Elemente de grafuri și combinatorică. Metode, algoritmi și programe*, Editura All București, 1994;
10. **C. Petre, D. Popa, Șt. Crăciunoiu, C. Iliescu**, *Metodica predării Informaticii și Tehnologiei Informației*, Editura Arves, Craiova, 2002;
11. **R. Pinte, D. Popescu Anastasiu**, *Bacalaureat la Informatică*, Editura L&S Infomat, București, 2005;
12. **I. Tomescu**, *Probleme de combinatorică și teoria grafurilor*, Editura Didactică și Pedagogică, București, 1981.