Recent changes 🔊 Login

Search



Responsabili: ■ ■Darius Neaţu

Stefan Popa

Objective laborator

• formalizarea noțiunilor de rețea de transport și flux în rețea

Laborator 11 - Flux Maxim

 prezentarea unei metode de rezolvare a problemei de flux maxim analiza unei implementări a metodei oferite

Importanță – aplicații practice

depăși capacitatea sa de transport. Rețelele de transport pot modela curgerea lichidului în sisteme cu țevi, deplasarea pieselor pe benzi rulante,

deplasarea curentului prin rețele electrice, transmiterea informațiilor prin rețele de comunicare etc. Descrierea problemei și a rezolvărilor

Un graf orientat poate fi utilizat pentru modelarea unui proces de transport într-o rețea între un producător s și un

consumator t. Destinația nu poate consuma mai mult decât se produce, iar cantitatea trimisă pe o cale nu poate

O problemă des întâlnită într-o rețea de transport este cea a găsirii fluxului maxim posibil prin arcele rețelei astfel încât:

1. să nu fie depășite capacitățile arcelor 2. fluxul să se conserve în drumul său de la *s* la *t*

Definiție 1

O rețea de transport este un graf orientat G = (V, E) cu proprietățile: 1. există două noduri speciale în V: s este nodul sursă (sau producătorul) și t este nodul terminal (sau

consumatorul). 2. este definită o funcție totală de capacitate c: V×V → R+ astfel încât:

c (u,v) = 0 dacă (u,v) ∉ E c (u,v) ≥ 0 daca (u,v) ∈ E

3. pentru orice nod $v \in V \setminus \{s,t\}$ există cel puțin o cale $s \rightarrow v \rightarrow t$. **Definiție 2**

Numim flux în rețeaua G = (V,E) o funcție totală f: V×V → R cu proprietățile:

1. Restricție de capacitate: • $f(u,v) \le c(u,v)$, $\forall (u,v) \in V$ - fluxul printr-un arc nu poate depăși capacitatea acestuia

c(u,v).

Exemplu:

Definiție 3

 $G_f = (V, E_f)$, astfel încât

2. Antisimetrie:

• f(u,v) = -f(v,u), $\forall u \in V$, $\forall v \in V$ 3. Conservarea fluxului:

■ Σ f(u,v) = 0, \forall u \in V \setminus {s,t}, v \in V Un flux negativ de la u la v este unul virtual, el nu reprezintă un transport efectiv, ci doar sugerează că există un transport fizic de la v la u (este o convenție asemănătoare cu cea făcută pentru intensitățile curenților într-o rețea

un nod este 0 (ținând cont de convenția de semn stabilită).

Numim capacitate reziduală a unui arc $c_f(u,v) = c(u,v) - f(u,v)$

muchie in graf:

muchii reziduale: Daca avem arcul $(u,v) \in V$ cu c(u,v) = 15 și f(u,v) = 10, se pot transporta $c_f(u,v) = 5$ unități suplimentare fără a încălca restricția de capacitate. Dar, conform definiției, deși arcul (v,u) ∉ V vom avea totuși o capacitate reziduală $c_f(v,u) = c(v,u) - f(v,u) = 0 - (-10) = 10$: as putea transporta 10 unități în sens opus care să le anuleze pe cele

electrică). Ultima proprietate ne spune că la trecerea printr-un nod fluxul se conservă: suma fluxurilor ce intră într-

și o interpretăm ca fiind cantitatea de flux adițional care poate fi transportat de la u la v, fără a depăși capacitatea

10 ale fluxului direct pe muchia (u,v).

Este important de observat că E_f și E pot fi disjuncte: un arc rezidual (u,v) apare în rețeaua reziduală doar dacă capacitatea sa este strict pozitivă (ceea ce nu implică existența arcului în rețeaua originală). Un drum de ameliorare este o cale (u1, u2, ..., uk), unde u1 = s și uk = t, în graful rezidual cu $c_f(ui, ui+1) > 0$, ∀i=1,2,...,k-1. Practic, un drum de ameliorare va reprezenta o cale în graf prin care se mai poate pompa flux adițional de la sursa la destinație.

prezentăm și cea mai utilizată metodă de rezolvare.

Algoritmul Ford-Fulkerson

■ $E_f = \{(u,v) \in V \mid c_f(u,v) = c(u,v) - f(u,v) > 0\}$

Aceasta este o metodă iterativă de găsire a fluxului maxim într-un graf care pleacă de la ideea: cât timp mai există un drum de ameliorare (o cale de la sursă la destinație) pot pompa pe această cale un flux suplimentar egal cu capacitatea reziduală a căii.

Output | fmax | |fmax| ← 0 $f(u,v) \leftarrow 0, \forall (u,v) \in V \times V$

|fmax| += cf(p)

for-each $(u,v) \in p$ $f(u,v) \leftarrow f(u,v) + cf(p)$ $f(v,u) \leftarrow -f(u,v)$ return |fmax|

elementelor marginite de o astfel de taietura. Exemplu:

Rețeaua inițială:

Edmonds-Karp

Input G(V,E), s, t

 $f(u,v) \leftarrow 0, \forall (u,v) \in V \times V$

Output |fmax|

|fmax| ← 0

Rețeaua reziduală:

*-----

Observăm că deși muchia (d,c) are capacitate 0 în rețeaua originală (ea nu ar putea transporta flux) în rețeaua

reziduală avem c(d,c) = 1 ceea ce îi permite să facă parte din drumul de ameliorare p1 = (s, a, b, d, c, t) de

capacitate $c_f(p1) = 1$, astfel că adăugarea acestui flux suplimentar pe muchia (d,c) nu va duce la încălcarea

while true $p(s \rightarrow t) \leftarrow BFS(Gf, s, t)$ if not $\exists p(s \rightarrow t)$ break; find $cf(p) = min\{cf(u,v) \mid cf(u,v) \in p\}$ |fmax| += cf(p)for-each $(u,v) \in p$ $f(u,v) \leftarrow f(u,v) + cf(p)$

 $c_f(p) = min(c_f(s,c), c_f(c,d), c_f(d,t)) =$ p = s -> c -> d -> t=min(3-0,2-0,1-0)=min(3,2,1)=1

Graful si calea de ameliorare gasita

 $p = s \rightarrow c \rightarrow d \rightarrow t$

p = s -> a -> b -> c -> e -> t

Capacitatea reziduala a caii de ameliorare

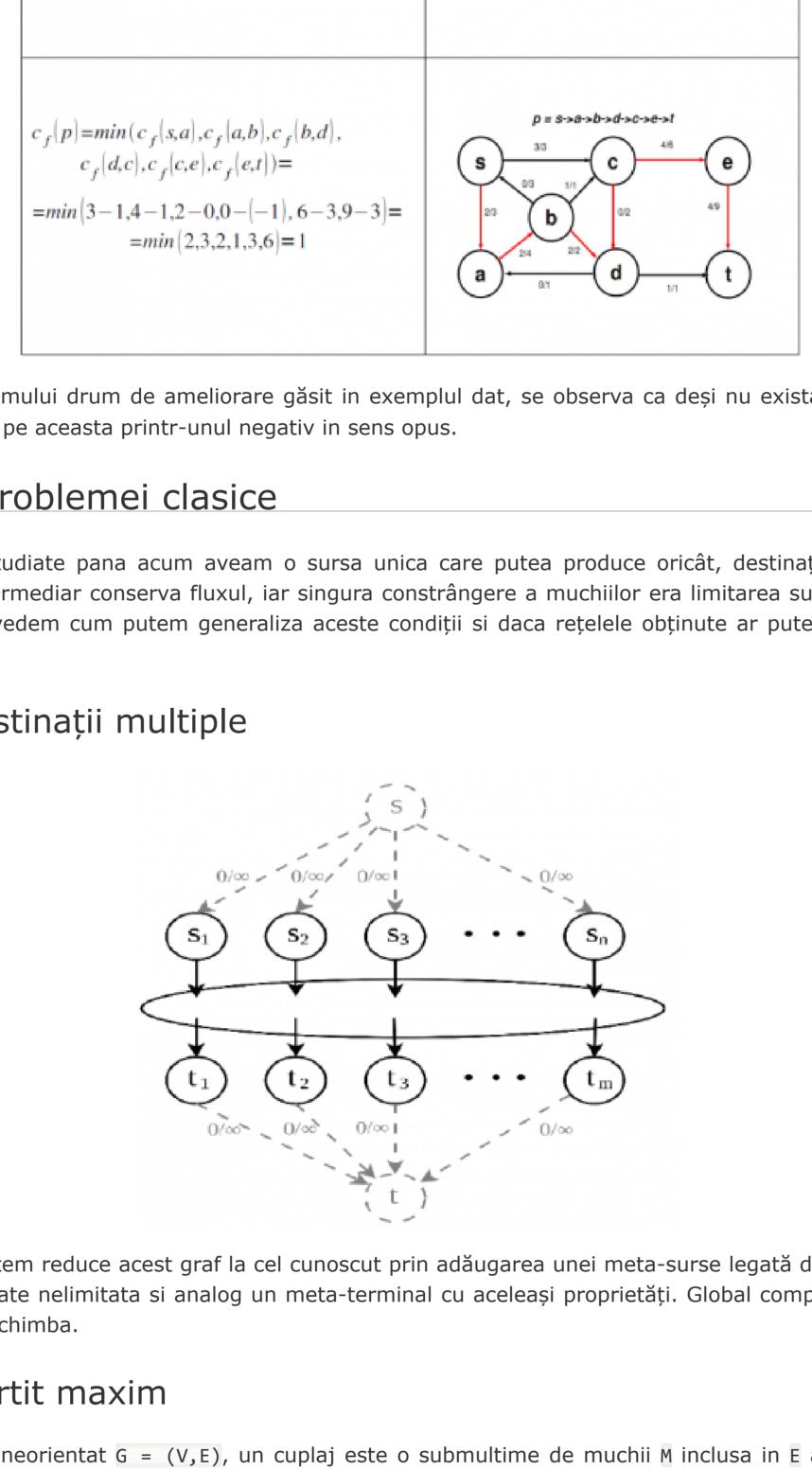
 $c_f(p) = min(c_f(s,c), c_f(c,e), c_f(e,t)) =$

=min(3-1,6-0,9-0)=min(2,6,9)=2

$$\begin{split} c_f(p) = & \min(c_f(s,a), c_f(a,b), c_f(b,c), \\ c_f(c,e), c_f(e,t)) = \end{split}$$

=min(3-0.4-0.1-0.6-2.9-2)=

=min(3,4,1,4,7)=1



La nivelul unei rețele întregi se adaugă un nod sursă cu muchii către toate nodurile ce consumau flux si un nod

[2, 9]d(v)d(v) - 2 d(w) + 2

Există cazuri în care aș vrea ca datele de pe muchiile rețelei sa facă parte dintr-un anumit interval [inf, sup]. Pe

Plecând tot de la condițiile de conservare la nivel de nod putem translata intervalul [inf, sup] în [0, sup-inf] și

să considerăm că nodul sursă a consumat inf unități de flux iar nodul destinație a produs inf unități. Din exterior

entitatea alcătuită din doua noduri și o muchie este văzută ca acționând în același fel asupra fluxului ce o

[2] Introducere in analiza algoritmilor, Cristian A. Giumale – Cap. V Algoritmi pr grafuri: Fluxuri maxime intr-un graf [3] Un articol de la MIT:
A Labeling Algorithm for the maximum flow network problem [4] Resurse wiki – Edmonds Karp Max Flow - Min Cut Theorem

In acest laborator vom folosi scheletul de laborator din arhiva skel-lab11.zip.

Inainte de a rezolva exercitiile, asigurati-va ca ati citit si inteles toate precizarile din sectiunea Precizari laboratoare 07-12.

• evitati **depunctari** la lab/teme/test

• evitati **buguri**

mai bun algoritm în prezent pentru cuplajul bipartit maxim se executa in O(√V*E).

Se da un graf orientat cu n noduri si m arce. Graful are pe arce capacitati pozitive. Folositi **Edmonds-Karp** pentru a gasi **fluxul maxim** intre nodul sursa **1** si nodul destinatie **n** in graful dat.

■ *n* <= 1.000 m <= 5.000• Nu exista arc de la **n** la **1**. timp de executie • C++: 1s Java: 2s

Proiect Catalog Test practic

Regulamente PA 2020

Diverse

Hall of PA

• [TEME] Configuratie vmchecker • [skel_graph] Precizari laboratoare 07-12

Laboratoare

• 00: Introducere și Relaxare skel-lab00.zip • 01: Divide et Impera skel-lab01.zip

sol-lab01.zip • 02: Greedy

skel-lab02.zip sol-lab02.zip • 03: Programare Dinamică 1 skel-lab03.zip

sol-lab03.zip skel-lab04.zip

• 04: Programare Dinamică 2 sol-lab04.zip • 05: Backtracking

skel-lab05.zip

sol-lab05.zip • 06: Minimax

skel-lab06.zip ■ ■ sol-lab06.zip • 07: Parcurgerea Grafurilor. Sortare Topologică skel-lab07.zip

sol-lab07.zip 08: Aplicații DFS skel-lab08.zip sol-lab08.zip • 09: Drumuri minime skel-lab09.zip sol-lab09.zip

skel-lab10.zip 11: Flux Maxim skel-lab11.zip ■ 12: A* skel-lab12.zip **Materiale Suplimentare Test Practic**

• 10: Arbori minimi de acoperire

Articole Probleme **Crash-course Opțional** Debugging şi Structuri de Date

Schelet: Debugging şi Structuri de Date

Laborator 11 - Flux Maxim

Obiective laborator

Importanță – aplicații

Descrierea problemei şi a

Algoritmul Ford-

Table of Contents

practice

rezolvărilor

Fulkerson Implementarea Edmonds-Karp Variații ale problemei clasice Surse si destinații multiple

 Rețea cu limite inferioare de capacitate Concluzii și observații Referinte Exercitii Flux maxim

BONUS

Cuplaj bipartit maxim

Rețea cu noduri ce nu

conservă fluxul

Așa cum era de intuit, capacitatea reziduală a unui drum de ameliorare p este cantitatea maximă de flux ce se poate transporta de-a lungul lui: $c_f(p) = \min\{c_f(u,v) \mid (u,v) \in p\}$ Acum că am introdus noțiunile necesare pentru formalizarea problemei de flux maxim într-un graf, putem să

Acest algoritm reprezintă mai mult un șablon de rezolvare pentru că nu detaliază modul în care se alege drumul de ameliorare din rețeaua reziduală. Ford_Fulkerson Input G(V,E), s, t

while $\exists a \text{ path } p(s ---> t) \text{ in } Gf \text{ such that } cf(u,v) > \emptyset, \forall (u,v) \in p$

find $cf(p) = min\{cf(u,v) \mid cf(u,v) \in p\}$

Complexitatea va fi O(E * f max) pentru că în ciclul while putem găsi, în cel mai rău caz, doar cai care duc la creșterea fluxului cu doar o unitate la fiecare pas. Ne punem acum problema dacă acest algoritm este corect, dacă la final vom avea cu adevărat fluxul maxim posibil prin graf. Corectitudinea algoritmului derivă imediat din teorema Flux maxim - tăietura minimă. Numim o **tăietură a unui graf** o partiție (S,T) a nodurilor sale cu proprietatea $s \in S$ si $t \in T$. Teorema flux maxim – tăietura minimă: Pentru o rețea de flux G(V,E) următoarele afirmații sunt echivalente: 1. f este fluxul maxim în G 2. Rețeaua reziduală G_f nu conține drumuri de ameliorare 3. Există o tăietură (S,T) a lui G astfel încât fluxul net prin tăietură este egal cu capacitatea acelei tăieturi. Obs: Prin orice tăietură fluxul este egal cu cel maxim pentru că nu există o altă cale pe care ar putea ajunge flux

de la sursă la destinație și care să nu treacă prin tăietură (ar încălca tocmai definiția ei); sau, altfel spus, valoarea

unui flux într-o rețea este dată de fluxul oricărei tăieturi. Astfel, fluxul total va fi mărginit de cea mai mică

capacitate a unei tăieturi. Dacă este îndeplinit punctul 3. al teoremei atunci știm că acea tăietură nu poate fi decât

una de capacitate minimă. Ultima incercare de a gasi o cale de la sursa la drena va rezulta in gasirea doar a

restricției de capacitate; sau am fi putut alege drumul
$$p2=(s, a, c, t)$$
 cu $c_f(p2)=1$ sau $p3=(s,a,b,d,t)$ cu $c_f(p3)=1$. Performanța algoritmului ține de modul în care va fi ales drumul de ameliorare, se poate întâmpla ca pentru $|fmax|$ de valoarea mare o alegere nepotrivită să ducă la timpi de execuție foarte mari. Implementarea Edmonds-Karp

Așa cum am văzut, algoritmul Ford-Fulkerson nu definește o metodă de alegere a drumului de ameliorare pe baza căruia se modifică fluxul în graf. Implementarea Edmonds-Karp alege întotdeauna cea mai scurtă cale folosind o căutare în lățime în graful rezidual unde fiecare arc are ponderea 1. Se poate demonstra că lungimea căilor găsite astfel crește monoton cu fiecare nouă ameliorare.

 $f(v,u) \leftarrow -f(u,v)$ return |fmax| Plecând de la ideea că drumurile de ameliorare găsite au lungimi din ce în ce mai mari se poate arăta că în această implementare fluxul se mărește de cel mult 0(V*E) ori ([1], pag.513). Complexitatea algoritmului va fi 0(V*E²). Să luăm un exemplu de rulare al acestui algoritm. Vom considera starea rețelei după ce a fost găsita prima cale de pompare flux(inițial toate arcele sunt etichetate cu 0/0 conform notației stabilite):

se va transforma in:

destinație dinspre toate nodurile ce produceau flux.

Rețea cu limite inferioare de capacitate

o astfel de muchie fluxul trebuie să respecte inegalitatea inf ≤ f ≤ sup

Exemplu:

traversează.

Iată un exemplu de transformare:

Concluzii și observații

Referinte

grafuri: Flux maxim

[5] Aplicatii flux maxim

Exercitii

Bineînțeles că toate aceste transformări pot fi combinate pentru a se ajunge la rețeaua de flux clasică.

Laboratorul de față s-a vrut a fi doar o introducere în domeniul fluxurilor într-un graf - modalitate de a reprezenta

probleme de circulație a materialelor atât de frecvent întâlnite. In [1] și [2] găsiți și alți algoritmi interesanți

împreună cu studiul complexității lor (algoritmul de pompare preflux, algoritmul "mutare-in-fata"). Spre exemplu, cel

[1] Introducere in algoritmi, Thomas H. Cormen, Charles E. Leiserson, Ronald R. Rivest - Capitolul VI Algoritmi pe

Prin citirea acestor precizari va asigurati ca: cunoasteti conventiile folosite

• 1 <= c <= 110.000, unde c este capacitatea unui arc

Rezolvati problema narta pe infoarena. pa/laboratoare/laborator-11.txt · Last modified: 2020/05/30 19:30 by darius.neatu

Flux maxim

Old revisions

Restrictii si precizari:

Rezultatul se va returna sub forma unui singur numar F, reprezentand fluxul maxim ce poate fi trimis prin retea. **BONUS**

> Media Manager Back to top (CC) BY-SA CHIMERIC DE WSC CSS DOKUWIKI SET FIREFOX RSS XML FEED WSC XHTML 1.0

Obs: In cazul ultimului drum de ameliorare găsit in exemplul dat, se observa ca deși nu exista muchia (d,c) se simulează un flux pe aceasta printr-unul negativ in sens opus. Variații ale problemei clasice In rețelele clasice studiate pana acum aveam o sursa unica care putea produce oricât, destinația unica consuma oricât, orice nod intermediar conserva fluxul, iar singura constrângere a muchiilor era limitarea superioara a fluxului prin capacitate. Sa vedem cum putem generaliza aceste condiții si daca rețelele obținute ar putea fi reduse la una clasica. Surse si destinații multiple Se observă ca putem reduce acest graf la cel cunoscut prin adăugarea unei meta-surse legată de sursele mici prin muchii de capacitate nelimitata si analog un meta-terminal cu aceleași proprietăți. Global comportamentul rețelei de flux nu se va schimba. Cuplaj bipartit maxim Fiind dat un graf neorientat G = (V,E), un cuplaj este o submultime de muchii M inclusa in E astfel incat pentru toate varfurile $v \in V$, exista cel mult o muchie in M incidenta in v. Spunem ca un varf $v \in M$ este cuplat de cuplajul M daca exista o muchie in M incidenta in v. Un cuplaj maxim este un cuplaj de cardinalitate maxima. In cazul grafurilor bipartite, multimea de varfuri poate fi partitionata in V = L U R, unde L si R sunt disjuncte si toate muchiile din E sunt intre L si R. Problema poate fi rezolvata cu ajutorul notiunii de flux, construind reteaua de transport G' = (V', E') pentru graful bipartit G. Vom alege sursa s si destinatia t ca fiind noi varfuri care nu sunt in V si vom construi V' = V U {s,t}. Arcele orientate ale lui G' sunt date de: • $E' = \{ (s, u) : u \in L \} \cup \{ (u, v) : u \in L, v \in R \text{ si } (u, v) \in E \} \cup \{ (v, t) : v \in R \}$ Pentru a completa constructia, vom atribui fiecarei muchii din E' capacitatea unitate. Rețea cu noduri ce nu conservă fluxul Spre deosebire de s si t care produc/consumă oricât, un nod intermediar ar putea produce sau consuma o cantitate constanta de flux la trecerea prin el. In acest caz vom avea un invariant la nivel de nod care ia forma: f_{in} - f_{out} = di, unde di este cantitatea produsă suplimentar (> 0) sau solicitată (< 0) de un nod.</p>

Am putea transforma egalitatea in: • $(f_{in} + |di|) - f_{out} = 0$, dacă di < 0 • $f_{in} - (f_{out} + |di|) = 0$, daca di > 0 Altfel spus, un nod ce consumă flux poate fi transformat intr-unul ce conservă fluxul și are un in-arc adițional de capacitate [di], iar unul ce produce flux va avea un out-arc de aceeași capacitate.