Suport curs algoritmica grafurilor
VIII. Rețele de flux (II) - Algoritmi de flux maxim

8.1 Algoritmul Edmonds-Karp

Algoritmul Ford-Fulkerson se poate îmbunătăți, se aplică Ford-Fulkerson dar calea reziduală în G_f este determinată folosind *BFS*. Se alege calea reziduală ca drumu minim $s \rightsquigarrow t$ în G_f cu ponderea arcurilor 1. Algoritmul astfel implementat se numește Edmonds-Karp.

Timpul de rulare este $O(VE^2)$.

Fie $\delta_f(u, v)$ drumul de cost minim de la u la v în G_f cu ponderea 1 pentru fiecare arc.

Lema 8.1.
Dacă algoritmul Edmonds-Karp este rulat pe o reţea de flux $G = (V, E)$ cu vârfurile sursă s şi destinaţie t, pentru toate vârfurile $v \in V \setminus \{s, t\}$, drumul minim $\delta_f(s, v)$ în graful rezidual G_f creşte monoton cu fiecare îmbunătăţire de flux.

Următoarea teoremă limitează superior numărul iterațiilor algoritmului.

Teorema 8.2.
Dacă algoritmul Edmonds-Karp este rulat pe o reţea de flux $G = (V, E)$ cu vârfurile sursă s şi destinaţie t, numărul total de îmbunătăţiri ale fluxului găsite de algoritm este $O(VE^2)$.

8.2 Algoritmul de pompare de preflux (push-relabel sau preflow-push)

Algoritmul face parte dintr-o clasa de algoritmi care încep calculele pe baza unui preflux existent în rețea, urmând apoi distribuirea prefluxului pe arce astfel încât, în final, prefluxul să îndeplinească restricțiile unui flux și să fie maxim.

Funcționarea algoritmului este similară cu curgerea lichidelor printr-un sistem de conducte de diverse capacități care leagă puncte (vârfuri) aflate la diverse înălțimi. Inițial vârful sursă s al rețelei este cel mai înalt, celelalte vârfuri fiind la înălțimea 0. Destinația t rămâne în permanență la înălțimea 0. Prefluxul este inițializat încărcând la capacitate maximă toate conductele ce pornesc din s. În cursul funcționării, se poate întâmpla ca fluxul (lichidul) strâns la un vârf u din conductele ce alimentează vârful să depășească posibilitatea de eliminare prin conductele de scurgere la care este conectat vârful (restricția de conservare de flux nu mai este îndeplinită). Excesul de flux este stocat într-un rezervor $u.e$ al nodului, cu capacitate nelimitată teoretic. Pentru a echilibra rețeaua și a elimina supraîncărcarea vârfurilor, sunt efectuate două operații de baza:

1. Mărirea înălțimii unui nod. Atunci când un nod supraîncărcat u are o conductă de scurgere orientată spre un vecin v, care nu este încărcată la capacitate maximă, u este înălțat mai sus decât v, astfel încât fluxul să poată curge din u în v prin conducta (u, v).

2. Pomparea fluxului unui nod. Un nod u supraîncărcat, aflat la o înălțime mai mare decât un vecin v și conectat cu v printr-o conductă subîncărcată, pompează flux din rezervorul $u.e$ spre v prin conducta (u, v).

Treptat, înălțimile nodurilor cresc monoton și pot depăși înălțimea sursei s. În acest moment fluxul în exces din rețea se pompează sursei. Numărul operațiilor de mărire a înălțimii vârfurilor și de pompare a fluxului este limitat, iar atunci când nu mai este posibilă nici o operație de acest fel, fluxul din rețea îndeplinește restricțiile impuse și este maxim.

Operații de bază

Fie $G = (V, E)$ un graf orientat cu n vârfuri și m arce, care desemnează o rețea de flux, s este vârful sursă și t este vârful destinație iar $c : V \times V \to \mathbb{R}$ este capacitatea arcelor, astfel încât $c(u, v) \geq 0$ pentru orice arc $(u, v) \in E$, iar $c(u, v) = 0$ dacă $(u, v) \notin E$.

Se notează cu $G_f = (V, E_f)$ graful rezidual al rețelei G, iar $c_f(u, v)$ este capacitatea reziduală a arcului (u, v).

Definiţie 8.2.1.
Se numeşte preflux într-o reţea $G = (V, E)$ o funcţie $f : V \times V \to \mathbb{R}$ asociată reţelei G, astfel încât sunt satisfăcute restricţiile:

1.	$f(u, v) \leq c(u, v)$	$\forall (u, v) \in E$	*- respectare capacitate arc;*
2.	$f(u, v) = -f(u, v)$	$\forall u \in V$ *şi* $v \in V$	*- simetrie preflux;*
3.	$\sum_{v \in V} f(u, v) \geq 0$	$\forall u \in V \setminus \{s\}$	*- supraîncărcare pozitivă.*

Definiţie 8.2.2.
Se numeşte supraîncărcare a unui vârf cantitatea

$$u.e = \sum_{v \in V} f(v, u) - f(u, v) > 0.$$

Dacă $u.e > 0$ *spunem că vârful* $u \in V \setminus \{s\}$ *este supraîncărcat.*

$u.e$ arată fluxul în exces stocat într-un rezervor virtual al vârfului.

În cursul procesului de calcul al fluxului maxim prin G înălțimea vârfurilor crește treptat, conform unei funcții de înălțime.

Definiţie 8.2.3.
O funcţie $h : V \to \mathbb{N}$ este o funcţie de înălţime pentru o reţea de flux $G = (V, E)$ şi satisface următoarele restricţii:

- $h(s) = |V|$*;*
- $h(t) = 0$*;*
- $u.h \leq v.h + 1$ *pentru orice arc rezidual* $(u, v) \in E_f$*.*

Lema 8.3.
Fie $G = (V, E)$ o reţea de flux cu fluxul f, iar $h : V \to \mathbb{N}$ o funcţie de înălţime a vârfurilor reţelei. Dacă pentru orice pereche de vârfuri $u, v \in V$ avem $u.h > v.h + 1$, atunci $(u, v) \notin E_f$ (arcul nu este rezidual = arc în reţeaua reziduala G_f).

Cele două operații esențiale ale algoritmului sunt de *pompare* a excedentului de flux al unui vârf și de *înălțare* a unui vârf. Cele două operații sunt aplicabile doar vârfurilor $V \setminus \{s, t\}$ și doar dacă sunt satisfăcute anumite condiții.

Pompare a excedentului de flux are loc doar dacă diferența de înălțime dintre u și v este exact 1, ($u.h = v.h + 1$) și dacă există exces de flux $c_f(u, v) > 0$. Dacă h este o funcție de înălțime și $u.h > v.h + 1$ arcul rezidual (u, v) nu există și pomparea nu are sens. Procedura de pompare este:

POMPARE(u,v)

1: \\ condiție de aplicare: $u \notin \{s, t\} \wedge u.e > 0 \wedge c_f(u, v) > 0 \wedge u.h = v.h + 1$
2: \\ acțiune: pompează cantitatea de flux $\Delta_f(u, v) = min(u.e, c_f(u, v))$
3: $\Delta_f(u, v) = min(u.e, c_f(u, v))$
4: **if** $(u, v) \in E$ **then**
5: $\quad (u, v).f = (u, v).f + \Delta_f(u, v)$
6: **else**
7: $\quad (v, u).f = (v, u).f - \Delta_f(u, v)$
8: $u.e = u.e - \Delta_f(u, v)$
9: $v.e = v.e + \Delta_f(u, v)$

Deoarece vârful u are un exces de flux și capacitatea arcului (u, v) este pozitivă se poate crește valoarea fluxului de la u la v cu valoarea $\Delta_f(u, v) = min(u.e, c_f(u, v))$ fără ca $u.e$ să devină negativă sau capacitatea $c(u, v)$ să fie depășită. Cantitatea de flux Δ_f pompată de arcul rezidual (u, v) asigură respectarea pozitivității excesului de flux din u în v și a capacității arcului (u, v) din G. Linia 3 determină valoarea lui Δ_f, liniile 4-6 actualizează fluxul f (linia 5 crește fluxul pe arcul (u, v) și linia 6 descrește fluxul deoarece arcul (v, u) rezidual este inversul arcului din rețeaua de flux G). Liniile 7-8 actualizează excesul de flux pentru vârfurile u și v.

Prefluxul f prin rețea continuă să-și păstreze proprietățile după aplicarea procedurii de pompare. Spunem că pomparea este saturată dacă după pompare $c_f(u, v) = 0$, arcul (u, v) este folosit la întreaga capacitate. Un arc saturat nu mai apare în rețeaua reziduală G_f. Alternativ pomparea este nesaturată dacă după pompare $c_f(u, v) > 0$, această situație are loc atunci când $u.e < c_f(u, v)$.

Înălțarea unui vârf se aplică dacă $u.e > 0$ și $u.h \leq v.h$ pentru toate arcele $(u, v) \in E_f$. Altfel spus, se înalță vârful u dacă pentru vârful v pentru care există capacitate reziduală de la u la v fluxul nu poate fi pompat de la u la v deoarece v nu este la un nivel inferior lui u.

ÎNĂLȚARE(u)

\\ condiție de aplicare: $u \notin \{s, t\} \wedge u.e > 0 \wedge [u.h \leq v.h | \forall v \in V, (u, v) \in E_f]$
\\ acțiune: mărește înălțimea $u.h$
$u.h = 1 + min\{v.h | (u, v) \in E_f\}$

Când se aplică procedura de înălțare E_f trebuie să conțină cel puțin un arc care pleacă din u. Aceasta reiese din :

$$u.e = \sum_{v \in V} f(v, u) - \sum_{v \in V} f(u, v).$$

Deoarece toate fluxurile sunt pozitive trebuie să existe un vârf v astfel încât $(u, v).f > 0$, dar $c_f(u, v) > 0 \Rightarrow (u, v) \in E_f$.

Cele două operații sunt aplicabile doar vârfurilor supraîncărcate din mulțimea $V \setminus \{s, t\}$. Lema 8.4 arată că pentru un vârf supraîncărcat $u \in V \setminus \{s, t\}$ există cel puțin o operație de *pompare* sau *înălțare* care poate fi aplicată vârfului u.

Lema 8.4.
Fie h o funcţie de înălţime în reţeaua de flux G. Atunci, un nod supraîncărcat $u \in V \setminus \{s, t\}$ poate participa fie într-o operaţie de pompare a prefluxului, fie într-o operaţie de înălţare a vârfului.

Concluzia lemei 8.4 spune că în momentul în care cele două operații nu mai pot fi aplicate vârfurilor rețelei de flux G, în G nu mai există vârfuri supraîncărcate.

Algoritmul generic de pompare de flux

INITIALIZARE_PREFLUX(G,s,t)
```
\\ inițializare f(u,v) și h(u,v), ∀u,v ∈ V
for fiecare v ∈ V do
    v.h = 0
    v.e = 0
for fiecare (u,v) ∈ E do
    (u,v).f=0
s.h = |V|
for fiecare v ∈ s.Adj do
    (s,v).f = c(s,v)
    v.e = c(s,v)
    s.e = s.e − c(s,v)
```

POMPARE_PREFLUX(G,s,t)
```
INITIALIZARE_PREFLUX(G,s,t)
while TRUE do
    if ∃u ∉ {s,t} ∧ u.e > 0 ∧ c_f(u,v) > 0 ∧ u.h = v.h + 1 then
        POMPARE(u,v)
        continue
    if ∃u ∉ {s,t} ∧ u.e > 0 ∧ [u.h ≤ v.h|∀v ∈ V, (u,v) ∈ E_f] then
        INALTARE(u)
        continue
    break
```

Procedura de inițializare creează un preflux inițial f definit de:

$$(u,v).f = \begin{cases} c(u,v), & \text{dacă } u = s, \\ 0, & \text{în rest.} \end{cases}$$

Totodată algoritmul începe cu o funcție de înălțime h definită de:

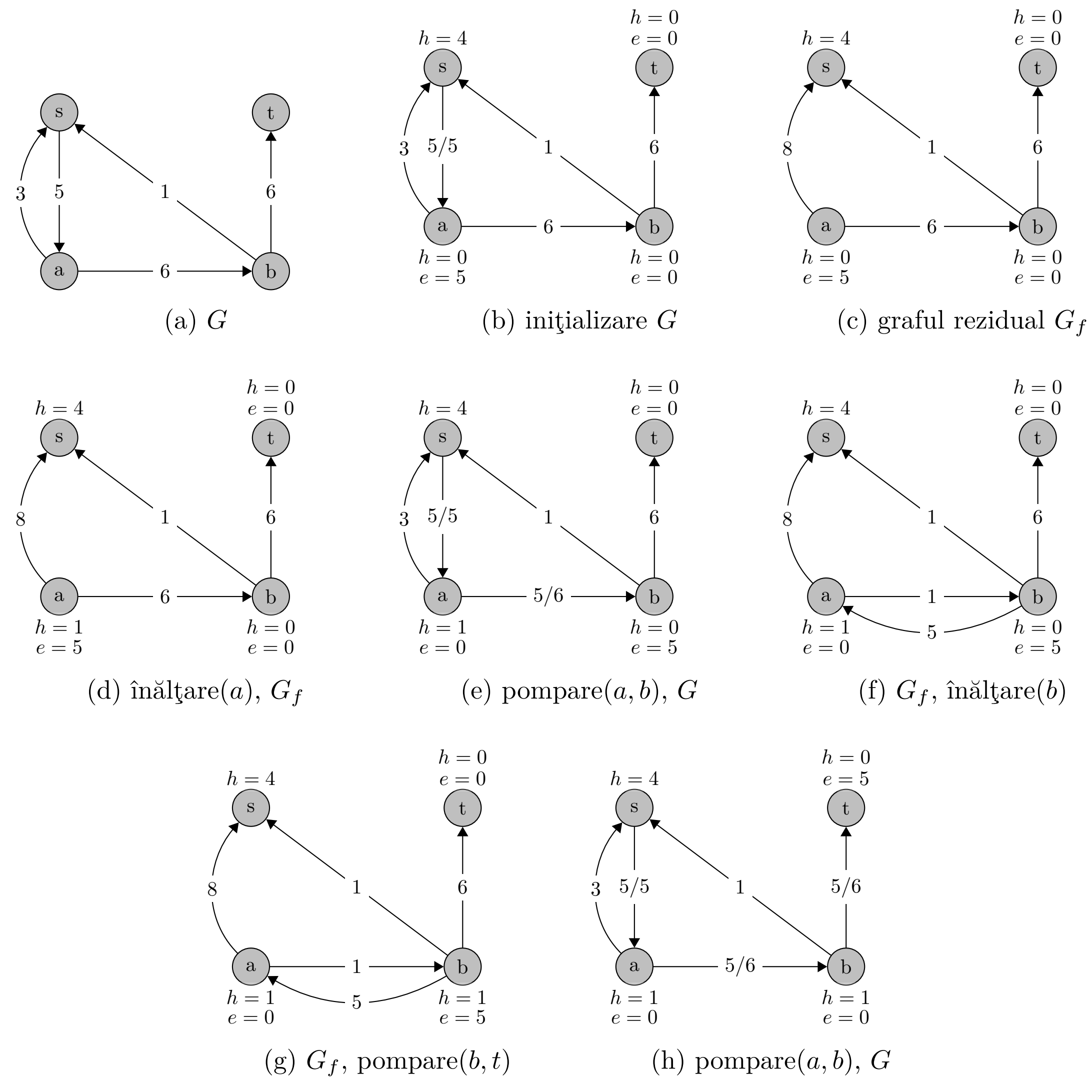
$$u.h = \begin{cases} |V|, & \text{dacă } u = s, \\ 0, & \text{în rest.} \end{cases}$$

Algoritmul se termină atunci când pentru orice nod din $u \in V \setminus \{s,t\}$ avem $u.e = 0$. La terminare prefluxul din rețea este fluxul maxim.

Complexitatea în timp a algoritmului este $O(V^2E)$.

Figura 1 prezintă un exemplu pentru algoritmul de pompare-preflux. Pentru graful original (figura 1a) sunt prezentate atributele înălțime și exces pentru fiecare vârf, pasul de inițializare, grafurile reziduale rezultate și efectul procedurilor de înălțare și pompare (figurile 1b-1h).

O versiune mai rapidă este algoritmul de pompare-topologică (relabel-to-front), având complexitatea în timp $O(V^3)$.



(a) G (b) iniţializare G (c) graful rezidual G_f

(d) înălţare(a), G_f (e) pompare(a, b), G (f) G_f, înălţare(b)

(g) G_f, pompare(b, t) (h) pompare(a, b), G

Figura 1: Exemplu: paşii urmaţi de algoritmul *pompare-preflux*.

8.3 Algoritmul de pompare-topologică (relabel-to-front)

Față de algoritmul de pompare_preflux, pomparea topologică impune o disciplină strictă de secvențiere a operațiilor elementare de pompare a prefluxului prin rețeaua G și de înălțare a vârfurilor rețelei. Această disciplină este conformă următorilor pași care, cu excepția inițializărilor, constituie partea centrală a algoritmului de pompare topologică.

1. Vârful curent prelucrat este selectat dintr-o listă $L(V)$ care conține toate vârfurile din $V \setminus \{s, t\}$. Inițial, ordinea vârfurilor din listă este oarecare.

2. Prelucrarea vârfului curent, cel de la vârful listei $L(V)$, fie el $u = head(L(V))$, urmărește eliminare completă a excesului de preflux $u.e$, dacă $u.e > 0$. Prelucrarea constă în vizitarea vecinilor lui u cu intenția aplicării operației *POMPARE(u,v)* și în mărirea treptată a înălțimii lui u, astfel încât operațiile de pompare să poată fi efectuate. Această prelucrare, numită *DESCĂRCARE*, este principala operație executată de pomparea topologică.

3. Dacă *DESCĂRCARE(u)* pentru vârful curent din $L(V)$ a condus la eliminarea efectivă a supraîncărcării vârfului, dacă $u.e > 0$ înainte de operație și $u.e = 0$ după operație, atunci u

este deplasat la începutul listei $L(V)$, iar procesul de selecție de la pasul (1) este reluat cu vârful succesor lui u din noua listă $L(V)$.

4. Repetarea pașilor (1-3) se termină atunci când parcurgând lista $L(V)$ se ajunge la capătul ei. Această situație apare atunci când $u.e = 0$ pentru orice vârf u din $L(V)$. În acest moment $t.e$ desemnează fluxul total prin rețeaua G și este maxim.

```
POMPARE_TOPOLOGICA(G,s,t)
INITIALIZARE_PREFLUX(G,s,t)
L = V \ {s,t}
for fiecare u ∈ V \ {s,t} do
    u.curent = u.N.head
u = L.head
while u ≠ NIL do
    înălțime_veche = u.h
    DESCARCARE(u)
    if u.h > înălțime_veche then
        mută u în capul listei L
    u.next
```

```
DESCARCARE(u)
while u.e > 0 do
    v = u.curent
    if v == NIL then
        INALTARE(u)
        u.curent = u.N.head
    else if c_f > 0 ∧ u.h == v.h + 1 then
        POMPARE(u,v)
    else
        u.curent = v.urmatorul_vecin
```

Figurile 2 și 3 prezintă un graf pentru care s-a aplicat procedura de $DESCARCARE(u)$ pentru a scăpa de excesul de flux din vârful y. Este nevoie de 15 iterații ale buclei *while* din procedura $DESCARCARE(u)$ pentru a pompa tot excesul din vârful y. Figurile prezintă doar vârfurile adiacente vârfului y și arcele ce leagă vârful y. Pentru figuri, numărul din interiorul vârfului reprezintă excesul de flux iar vârfurile sunt desenate pe nivele ce reprezintă înălțimea vârfului. Lista vecinilor vârfului y la începutul fiecărei iterații este ilustrată în dreapta fiecărei rețea de flux, numărul iterației este este reprezentat de coloanele de pe linia 1. Inițial (figura 2a) vârful y are în exces 19 unități de flux ce trebuie pompate ($y.e = 19$) și vârful curent în care se încearcă pomparea este s, ($y.curent = s$). Iterațiile $1, 2$ și 3 doar modifică atributul $y.curent$ deoarece nu există arce pe care se poate pompa excesul de flux. În iterația 4 $y.curent = NIL$ vârful y este înălțat și $y.curent$ este resetat la începutul listei (figura 2b). Procesul continua și în iterația 7 când $y.curent = z$ se pot pompa 8 unități de flux pe arcul (y, z) în vârful z, deoarece în această iterație se pompează exces de flux atributul $y.curent$ nu se modifică. Figura 2c prezintă iterațiile 8 și 9 după care $y.curent = NIL \Longrightarrow$ vârful y este înălțat. Procesul continua 2c-3c până când tot excesul de flux este pompat din vârful y.

8.4 Referințe

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2009. Introduction to Algorithms, Third Edition (3rd ed.). The MIT Press.
2. Geir Agnarsson and Raymond Greenlaw. 2006. Graph Theory: Modeling, Applications, and Algorithms. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
3. Mark Newman. 2010. Networks: An Introduction. Oxford University Press, Inc., New York, NY, USA.
4. Cristian A. Giumale. 2004. Introducere în analiza algoritmilor, teorie și aplicație. Polirom.

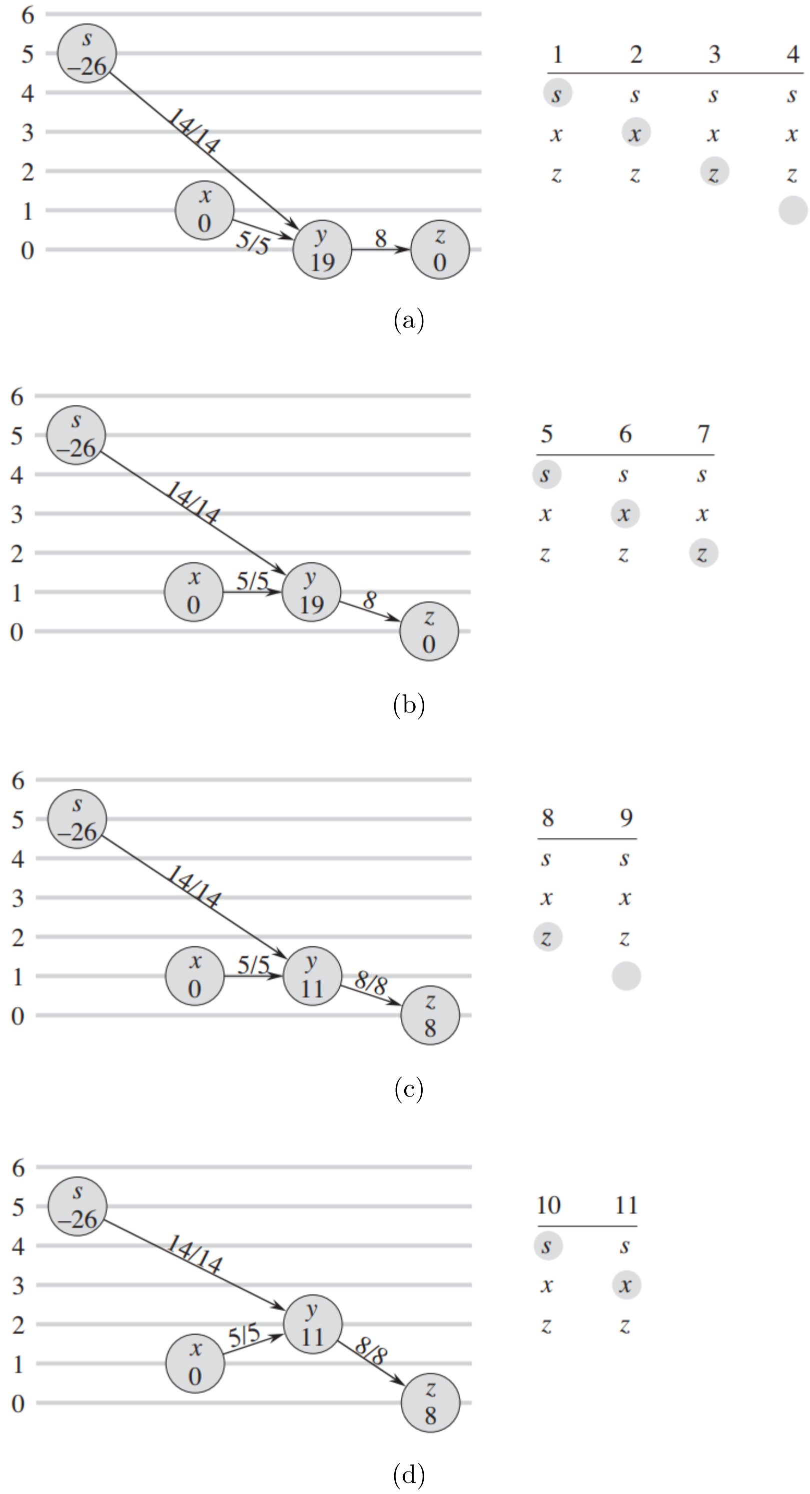


Figura 2: Paşii urmaţi de procedura de DESCARCARE(u) (explicaţiile se regăsesc în text).

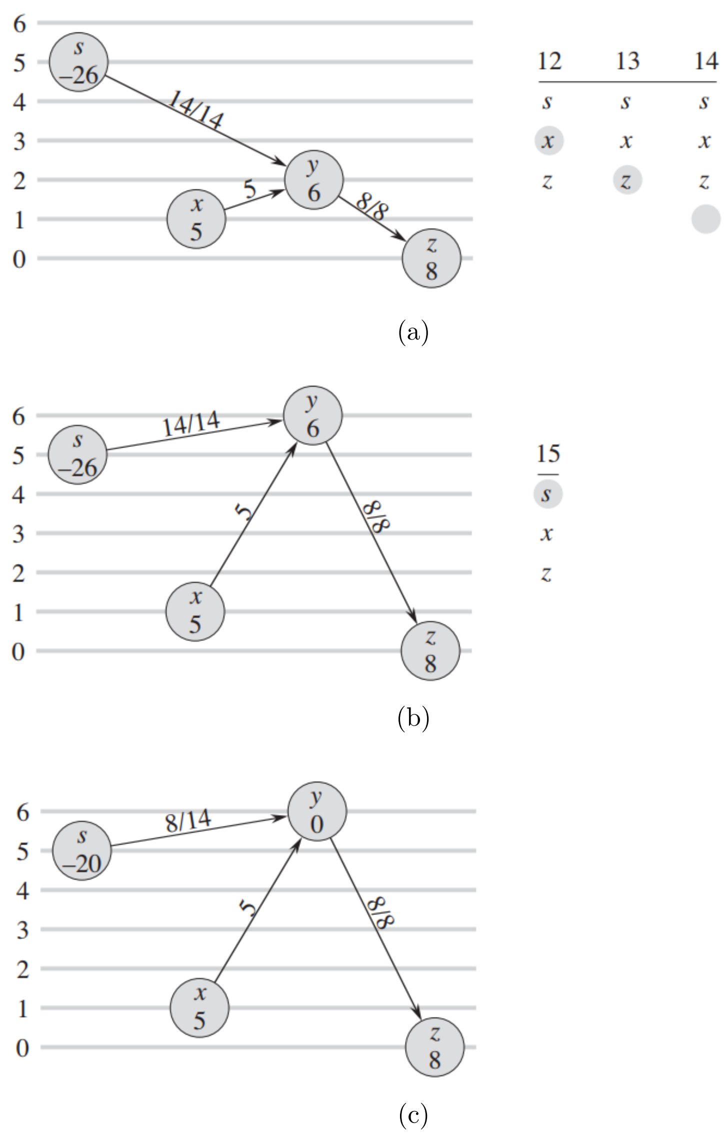


Figura 3: Paşii urmaţi de procedura de DESCARCARE(u). Continuarea exemplului din figura 2 (explicaţiile se regăsesc în text).