Suport curs algoritmica grafurilor
VII. Rețele de flux

Așa cum se pot modela hărți rutiere folosind grafuri și căuta drumul minim între două puncte, un graf orientat se poate interpreta ca o rețea de flux.

Intuitiv putem să ne imaginăm un material curgând de la o sursă (unde este produs) către o destinație (unde e consumat). Sursa (*source*) produce material constant și destinația (*sink*) consumă material la aceeași rată. Fluxul materialului, la un moment dat, este rata cu care se mișcă.

Rețelele de flux pot modela multe probleme: curgerea lichidelor, traseul unor componente pe linia de asamblare, curentul într-o rețea electrică, informația într-o rețea de comunicații.

Ne putem gândi la arcele din graf ca la un canal de transport pentru material, fiecare canal are definit o capacitate pentru rata maximă cu care materialul curge pe canal (ex. $20A$ într-un cablu electric, $350Mbit/s$ pentru un cablu CAT5E). Vârfurile grafului reprezintă intersecția canalelor de transport, materialul curge prin vârfuri fără să se acumuleze.

Rata cu care materialul intră într-un vârf este egală cu rata cu care iese din vârf, fluxul se conservă. În problema fluxului maxim trebuie determinată rata maximă cu care se poate transporta material de la sursă la destinație fără a încălca diferite constrângeri aplicate capacităților.

7.1 Rețele de flux

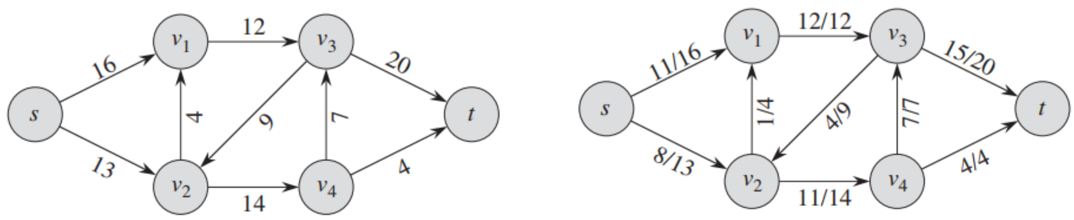
O rețea de flux $G = (V, E)$ este un graf orientat în care fiecare arc $(u, v) \in E$ are o capacitate pozitivă $c(u, v) \geq 0$.

- Dacă E conține arcul (u,v), E nu trebuie să conțină și arcul (v,u);
- dacă $(u, v) \notin E \Longrightarrow c(u, v) = 0$, nu există bucle;
- se disting 2 vârfuri într-o rețea de flux:
  - s = sursa (source)
  - t = destinație (sink)

Se presupune că fiecare vârf se află pe un drum de la s la t ($\forall v \in V, s \rightsquigarrow v \rightsquigarrow t$).

Definiţie 7.1.1 (Reţea de flux).
Fie $G = (V, E)$ o reţea de flux cu funcţia de capacitate c, G e graf orientat. Fie s sursa din reţea şi t vârful destinaţie. Un flux este o funcţie $f : V \times V \rightarrow \mathbb{R}$ ce satisface restricţiile:

- *se respectă capacitatea arcului:* $\forall u, v \in V, 0 \leq f(u, v) \leq c(u, v)$;



(a) (b)

Figura 1: (a) O reţea de flux $G = (V, E)$ ce modelează problema de transport între fabrică (vârful s) şi depozit (vârful t). Ponderea fiecărui arc reprezintă capacitatea de transport pe zi pe ruta respectivă. (b) Un flux în reţeaua de flux de la (a) de valoare $|f| = 19$. Fiecare arc este etichetat sub forma $f(u, v)/c(u, v)$ (simbolul "/" este folosit doar pentru a separa fluxul de capacitate).

- *conservarea fluxului:* $\forall u \in V \setminus \{s, t\}, \sum_{v \in V} f(v, u) = \sum_{v \in V} f(u, v)$ *(fluxul nu se acumulează sau pierde într-un vârf).*

Dacă $(u, v) \notin E \implies f(u, v) = 0$. Valoarea pozitivă $f(u, v)$ se numește fluxul de la u la v, valoarea $|f|$ a fluxului f se definește ca

$$|f| = \sum_{v \in V} f(s, v) - \sum_{v \in V} f(v, s).$$

adică diferențele între valoarea totală a fluxului ce iese minus valoarea totală a fluxului ce intră, notația $|.|$ reprezintă valoarea fluxului și nu modul sau cardinal.

Pentru o rețea de flux

$$\sum_{v \in V} f(v, s) = 0$$

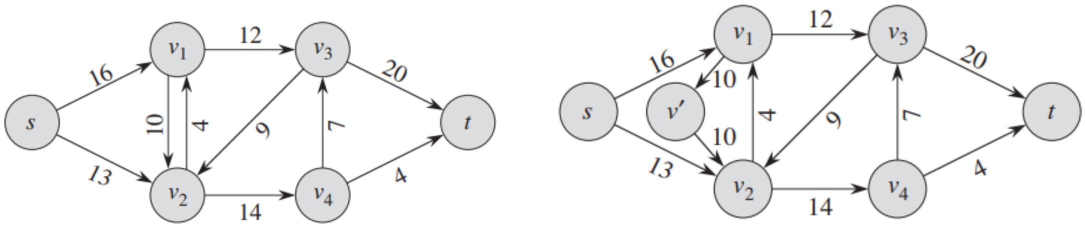
nu există arce ce intră în sursa s. În problema fluxului maxim se dă o rețea de flux G cu vârfurile s și t și se cere să se găsească un flux maxim.

Exemplu Se modelează problema de transport din figura 1a. O companie deține o fabrică (sursa s) în Timișoara ce produce mingi de tenis și un depozit (destinația t) în Cluj.

Compania închiriază spațiu pe camioane de la o altă firmă pentru a transporta mingile de tenis. Camioanele călătoresc pe rute prestabilite (arce) între orașe (vârfuri) $\rightarrow$ capacitate limitată de transport. Compania poate expedia maxim $c(u, v)$ mingi pe zi între două orașe. Trebuie să se determine numărul maxim de lăzi ce pot fi transportate pe zi și să se producă doar atât. Nu contează durata transportului, ce contează este ca p lăzi să iasă din fabrică pe zi și p lăzi să ajungă în depozit pe zi. Problema se poate modela cu o rețea de transport (figura 1b), capacitatea arcelor trebuie respectată altfel lăzile s-ar acumula în orașele intermediare.

Arce antiparalele De exemplu, compania de transport oferă posibilitatea închirierii de spațiu pentru 10 lăzi pe ruta $v_1 \rightsquigarrow v_2$ (situație prezentată în figura 2).

Rețeaua încalcă constrângerea/presupunerea inițială $(v_1, v_2) \in E \rightarrow (v_2, v_1) \notin E$ (figura 2a). Arcele (v_1, v_2) și (v_2, v_1) se numesc arce antiparalele. Dacă se vrea modelarea problemei ca o rețea de flux trebuie să se transforme rețeaua: se alege un arc antiparalel (de ex. (v_1, v_2) și se împarte în două (v_1, v') și (v', v_2) ambele cu aceeași capacitate $c(v_1, v_2) = c(v_1, v') = c(v', v_2)$ (figura 2b).

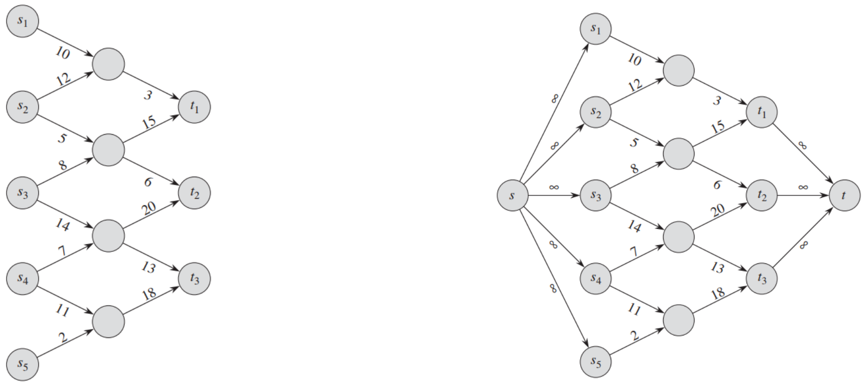


(a) (b)

Figura 2: Conversia unei reţele de flux ce conţine arce paralele într-o reţea fără astfel de arce. (a) O reţea de flux $G = (V, E)$ ce conţine arcele paralele (v_1, v_2) şi (v_2, v_1). (b) Reţeaua de flux fără arce paralele obţinută prin adăugarea vârfului v'.

Rețele cu noduri sursă și destinație multiple De exemplu, compania producătoare de mingi de tenis poate avea mai multe fabrici și depozite:

- un set de m surse $\{s_1, s_2, \ldots, s_m\}$;
- un set de n destinații $\{t_1, t_2, \ldots, t_n\}$.



(a) (b)

Figura 3: Reţea de flux cu surse şi destinaţii multiple.

Pentru a rezolva problema se adaugă un vârf sursă, supersursa s, și arcele (s, s_i) cu capacitățile $c(s, s_i) = \infty, \forall i = 1, \ldots, m$. Se adaugă și un vârf destinație, superdestinație t, și arcele (t_j, t) cu capacitățile $c(t_i, t) = \infty, \forall i = 1, \ldots, n$. s oferă atât flux cât e necesar pentru fiecare s_i și t consumă atât flux cât primește de la fiecare t_i, situația este ilustrată de figura 3.

7.2 Metoda Ford-Fulkerson

O metodă ce permite rezolvarea problemei de flux maxim, există mai multe implementări și timpi de rulare diferiți. Metoda Ford-Fulkerson depinde de 3 concepte:

- rețea reziduală;
- drum de creștere - cale reziduală;
- tăieturi.

Metoda Ford-Fulkerson crește iterativ valoarea fluxului. Pornește cu $f(u, v) = 0, \forall u, v \in V$ și în fiecare pas se crește fluxul prin găsirea unui drum ”de creștere” (*cale reziduală* sau *drum rezidual*) într-o rețea reziduală G_f asociată lui G. După ce se cunosc arcele unei căi reziduale în G_f, se pot identifica arce în G pentru care se poate schimba fluxul astfel încât să crească valoarea fluxului total. În fiecare pas se poate ca fluxul să și descrească. Acești pași se repetă atât timp cât există drumuri reziduale în rețeaua reziduală.

METODA_FORD_FULKERSON(G, s, t)
1: inițializare flux cu 0
2: **while** există o cale reziduală p în graful rezidual G_f **do**
3: crește fluxul f de pe p
4: **return** f

7.3 Rețea reziduală/Graf rezidual

Intuitiv fie o rețea de flux $G = (V, E)$ și fluxul f, rețeaua reziduală G_f e formată din arce a căror capacitate arată cum pot schimba fluxul arcelor din $G \implies$ un arc din rețeaua de flux admite un flux suplimentar egal cu capacitatea legăturii minus fluxul de pe arc.

Dacă această valoare este pozitivă se trasează arcul în G_f cu ”capacitatea reziduală” $e_f(u, v) = c(u, v) - f(u, v)$. Singurele arce din G ce se regăsesc în G_f sunt cele ce admit mai mult flux.

G_f poate conține și arce ce nu se regăsesc în G. Algoritmul poate ajunge în situația în care trebuie să scadă fluxul unui arc pentru a maximiza fluxul total din rețea. Pentru a reprezenta o posibilă descreștere a fluxului $f(u, v)$ unui arc din G se adaugă un arc (v, u) în G_f cu capacitate reziduală $c_f(v, u) = f(u, v)$.

În rețeaua reziduală se pot trasa arce ce admit flux în direcția opusă lui (u, v) și cel mult anulând fluxul de pe (u, v). Se poate trimite înapoi flux $\iff$ se reduce fluxul de pe un arc (operație necesară).

Formal fie rețeaua de flux $G = (V, E)$ cu s și t. Fie f un flux în G și perechea de vârfuri $u, v \in V$. Definim capacitatea reziduală $c_f(u, v)$:

$$c_f(u, v) = \begin{cases} c(u, v) - f(u, v), & \text{dacă } (u, v) \in E \\ f(v, u), & \text{dacă} (v, u) \in E \\ 0, & \text{în rest} \end{cases}$$

Deoarece dacă $(u, v) \in E$ se presupune că $(v, u) \notin E \implies$ doar un caz apare pentru fiecare arc.

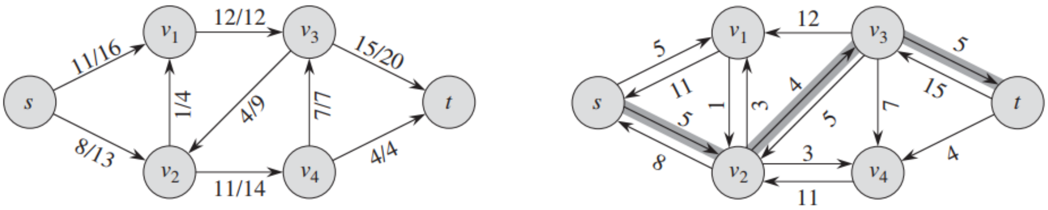
De exemplu: $c(u, v) = 16$ și $f(u, v) = 11 \implies$ se poate crește $f(u, v)$ cu până la la $c_f(u, v) = 5$ unități până să se încalce constrângerea capacității arcului. De asemenea se vrea ca algoritmul să poată întoarce max 11 unități de flux de la u la $v \implies c_f(v, u) = 11$.

Definiţie 7.3.1 (Graf rezidual/Reţea reziduală).
fie o reţea de flux $G = (V, E)$ şi fluxul f prin G, numim reţea reziduală a lui G (sau graf rezidual al lui G), indusă de f, o reţea de flux $G_f = (V, E_f)$ unde:

$$E_f = \{(u, v) \in V \times V | c_f(u, v) > 0\}.$$

Fiecare arc din rețeaua reziduală, *arc rezidual*, poate admite un flux pozitiv. O *cale reziduală* este un drum $s \rightsquigarrow t$ în G_f și $c_f(u, v)$ este capacitatea reziduală. Arcele din E_f sunt cele din E sau inverse

$$|E_f| \leq 2|E|.$$



(a) G (b) G_f

Figura 4: Reţea de flux şi graful rezidual.

G_f e similară cu o rețea de flux, cu capacități date de c_f, G_f nu satisface definiția dată pentru o rețea de flux deoarece poate conține arcele (u, v) și (v, u) (singura diferență dintre G și G_f).

Un flux în rețeaua reziduală G_f dă indicații unde se poate adăuga unități de flux în rețeaua de flux originală G. Figura 4 prezintă o rețea de flux și graful rezidual asociat.

Definiţie 7.3.2 (îmbunătăţirea fluxului)**.**
Dacă f este un flux în G şi f' este un flux în G_f, definim $f \uparrow f'$ îmbunătăţirea fluxului f de f' ca o funcţie de la $V \times V$ la $\mathcal{R}$ definită de

$$(f \uparrow f')(u, v) = \begin{cases} f(u, v) + f'(u, v) - f'(v, u), & \textit{dacă } (u, v) \in E \\ 0, & \textit{în rest} \end{cases}$$

Intuitiv se urmărește definiția rețelei reziduale:

- se mărește fluxul pe (u, v) cu $f'(u, v)$ dar se scade cu $f'(v, u)$ deoarece fluxul de pe arcul (v, u) în G_f înseamnă descreșterea fluxului în rețeaua originală;
- împingerea fluxului pe arcul invers se mai numește anulare.

De exemplu dacă se trimit 5 lăzi de la $u \to v$, 3 lăzi de la $v \to u$ din perspectiva rezultatului final se trimit doar 2 lăzi de la $u \to v$ și 0 de la $v \to u$.

Lema 7.1.
Fie $G = (V, E)$ o reţea de flux cu s şi t şi f un flux în G. Fie G_f reţeaua reziduală a lui G indusă de f şi fie f' un flux în G_f. Funcţia $f \uparrow f'$ este un flux în G cu valoarea $|f \uparrow f'| = |f| + |f'|$.

7.4 Creșterea fluxului pe un drum

Fie o rețea de flux $G = (V, E)$ și un flux f, o cale reziduală (care îmbunătățește) p este un drum de la s la t în graful rezidual G_f. Din definiția grafului rezidual, se poate crește fluxul pe un arc (u, v) de pe un *"drum mai bun"* p cu max $c_f(u, v)$ fără a încălca constrângerea capacității pe oricare $(u, v) \in E$.

Drumul reprezentat de arcele îngroșate din figura 4b este un drum ce îmbunătățește fluxul, se poate crește fluxul de pe fiecare arc din p până la 4 unități fără a încălca constrângerea capacității deoarece capacitatea minimă $c_f(v_2, v_3) = 4$.

Capacitate reziudală este valoarea maximă cu care se poate îmbunătăți fluxul pe fiecare arc din p, este dată de:

$$c_f(p) = min\{c_f(u, v) | (u, v) \in p\}.$$

Lema 7.2.
fie $G = (V, E)$ o reţea de flux, fie f un flux în G şi p o cale reziduală, $p \in G_f$. Definim funcţia $f_p : V \times V \to \mathbb{R}$

$$f_p(u, v) = \begin{cases} c_f(p), & \textit{dacă } (u, v) \in p \\ 0, & \textit{în rest} \end{cases}$$

Corolar 7.3.
Fie $G = (V, E)$ o reţea de flux şi f un flux în G şi p în G_f. Fie f_p definit de Lema (7.2) şi presupunem că se îmbunătăţeşte f cu f_p, Atunci funcţia $f \uparrow f_p$ este un flux în G cu valoarea $|f \uparrow f_p| = |f| + |f_p| > |f|$.

7.5 Tăieturi și fluxuri în rețele

Metoda Ford-Fulkerson îmbunătățește iterativ fluxul până când găsește fluxul maxim. Cum știm că am găsit fluxul maxim? Fluxul e maximal dacă graful rezidual nu conține drum ce poate îmbunătăți fluxul.

O tăietură (S, T) a rețelei de flux $G = (V, E)$ este o partiție a lui V în S și $T = V \setminus S$ astfel încât $s \in S$ și $t \in T$. Dacă f este un flux atunci fluxul net $f(S, T)$ peste tăietura (S, T) este definit de

$$f(S,T) = \sum_{u \in S} \sum_{v \in T} f(u,v) - \sum_{u \in S} \sum_{v \in T} f(v,u)$$

iar capacitatea tăieturii (S, T) este

$$c(S,T) = \sum_{u \in S} \sum_{v \in T} c(u,v).$$

Figura 5 prezintă o tăietură în graf.

Lema 7.4.
fie f un flux într-o reţea de flux G şi (S, T) o tăietură în G, fluxul prin tăietura (S, T) este fluxul prin reţea $f(S, T) = |f|$.

Teorema 7.5 (Teorema flux maxim, tăietura minimă)**.**
fie $G = (V, E)$ o reţea de flux. Următoarele afirmaţii sunt echivalente:

1. *f este o funcţie de flux în G_f astfel încât $|f|$ este flux maxim total în G;*

2. *reţeaua reziduală G_f nu are căi reziduale (drumuri $s \rightsquigarrow t$);*

3. *există o tăietură (S, T) a lui G astfel încât $|f| = c(S, T)$.*

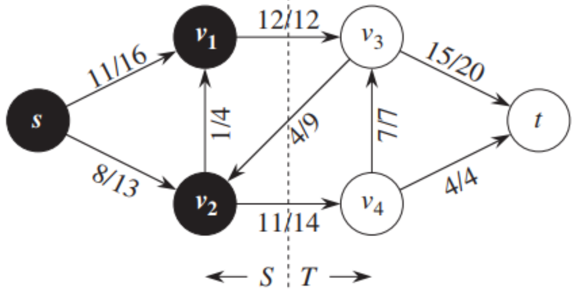


Figura 5: Tăietură într-o reţea de flux.

7.6 Algoritmul Ford-Fulkerson

Timpul de rulare depinde de calculul căii reziduale, dacă se utilizează algoritmul *BFS* pentru a determina calea reziduala $p \Longrightarrow$ algoritmul rulează în timp polinomial.

Algoritmul *Ford-Fulkerson* este (atributul $(u, v).f$ indică fluxul arcului):

FORD_FULKERSON(G, s, t)

1: **for** $(u, v) \in E$ **do**
2:     $(u, v).f = 0$
3: **while** există o cale reziduală p de la s la t în graful rezidual G_f **do**
4:     $c_f(p) = min\{c_f(u, v) | (u, v) \in p\}$
5:     **for** fiecare arc $(u, v) \in p$ **do**
6:         **if** $(u, v) \in E$ **then**
7:             $(u, v).f = (u, v).f + c_f(p)$
8:         **else**
9:             $(v, u).f = (v, u).f - c_f(p)$
10: **return** f

Fiecare iterație a buclei *while* durează $O(V + E') = O(E)$ dacă folosim *DFS* sau *BFS* pentru a determina un drum rezidual. Bucla *while* se execută de cel mult $|f^*|$ ori, f^* - fluxul maxim în rețea.Complexitatea în timp pentru Ford-Fulkerson este $O(E|f^*|)$ (dacă se folosește *DFS* pentru a determina un drum rezidual în graful rezidual).

Figura 6 prezintă iterațiile algoritmului *Ford-Fulkerson* (liniile $3 - 10$).

7.7 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.

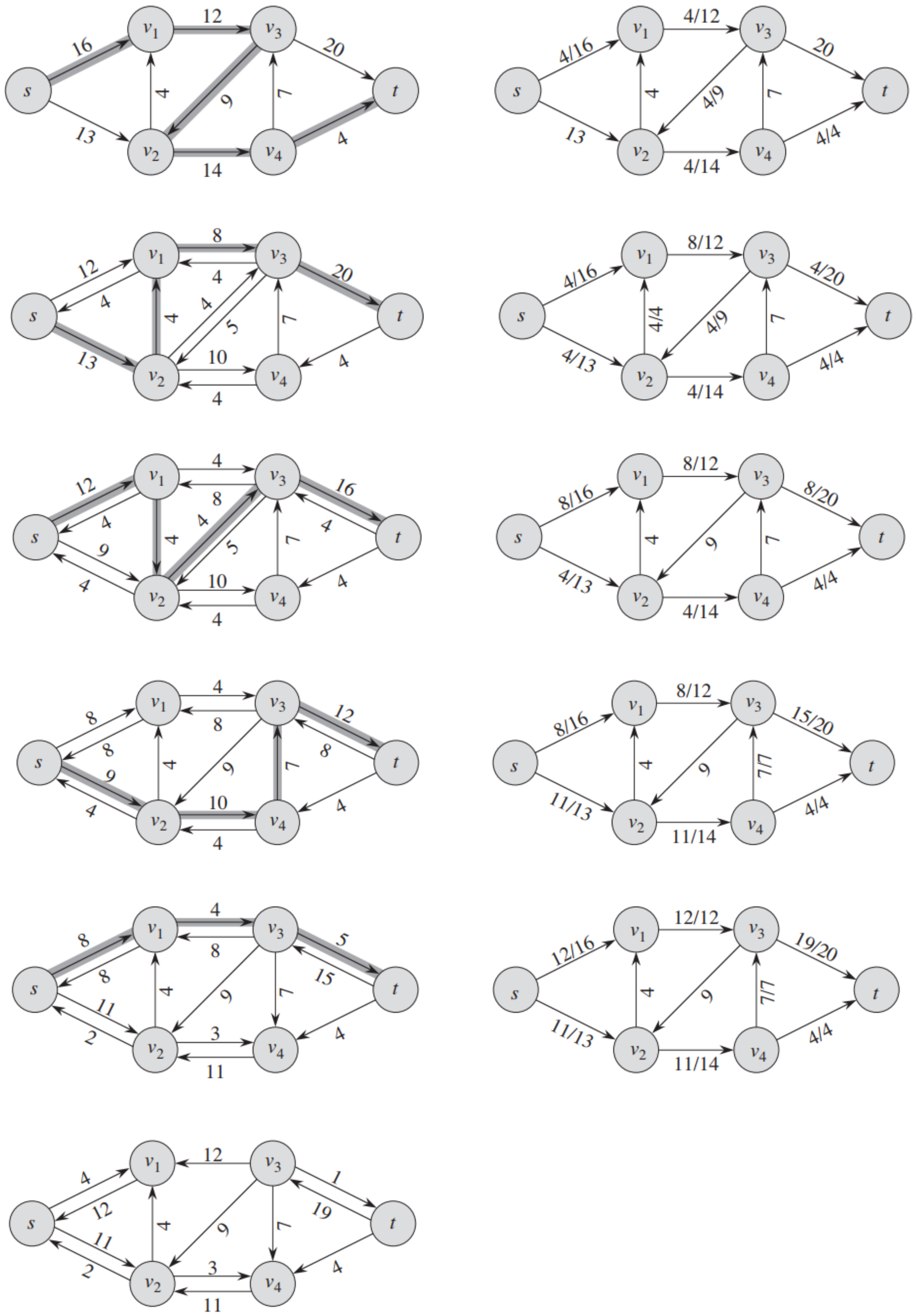


Figura 6: Paşii urmaţi de algoritmul Ford-Fulkerson. Grafurile din partea stângă reprezintă reţeaua reziduală iar grafurile din partea dreaptă prezintă fluxul obţinut în urma îmbunătăţirilor descoperite în graful rezidual.