Universitatea
Transilvania
din Brașov

ȘCOALA DOCTORALĂ INTERDISCIPLINARĂ

Facultatea: Matematică şi Informatică

Camelia SCHIOPU

Clase de fluxuri dinamice

Dynamic flows classes

REZUMAT / ABSTRACT

Conducător ştiințific

Prof.dr. Eleonor CIUREA

BRAȘOV, 2019

Universitatea Transilvania din Brașov

COMPONENȚA

Comisiei de doctorat

Numită prin ordinul Rectorului Universității Transilvania din Braşov

Nr. ..... din .....

PREŞEDINTE:	**Conf.dr. Eugen PĂLTĂNEA** Decanul Facultății de Matematică și Informatică Universitatea *Transilvania* din Brașov
CONDUCĂTOR ŞTIINȚIFIC:	**Prof.dr. Eleonor CIUREA** Universitatea *Transilvania* din Brașov
REFERENȚI:	**Prof.dr. Ioan TOMESCU** Universitatea din București **Prof.dr. Cornelius CROITORU** Universitatea „Alexandru Ioan Cuza" din Iași **Conf.dr. Laura CIUPALĂ** Universitatea *Transilvania* din Brașov

Data, ora şi locul susținerii publice a tezei de doctorat: 20.09.2019 ora 12.00, sala PI1.

Eventualele aprecieri sau observații asupra conținutului lucrării vor fi transmise electronic, în timp util, pe adresa camelia.s@unitbv.ro.

Totodată, vă invităm să luați parte la şedința publică de susținere a tezei de doctorat.

Vă mulțumim!

Universitatea Transilvania din Brașov

CUPRINS

	Pg. teza	Pg. rezumat
INTRODUCERE	5	7
1. FLUXURI ÎN REȚELE	8	10
1.1 Fluxuri în rețele statice oarecare	8	10
1.1.1 Noțiuni introductive	8	10
1.1.2 Fluxul maxim	9	11
1.1.3 Fluxul minim	12	13
1.2 Fluxuri în rețele statice speciale	15	15
1.2.1 Fluxuri maxime în rețele statice planare	15	15
1.2.2 Fluxuri minime în rețele statice planare	19	17
1.2.3 Fluxuri maxime în rețele statice bipartite	19	18
1.2.4 Fluxuri minime în rețele statice bipartite	22	20
1.3 Fluxuri în rețele dinamice oarecare	24	21
1.3.1 Fluxul maxim în rețele dinamice oarecare	24	21
1.3.2 Fluxul minim în rețele dinamice oarecare	28	26
2. FLUXURI ÎN REȚELE DINAMICE PLANARE	29	27
2.1 Fluxuri în rețele dinamice planare. Abordarea statică	29	27
2.1.1 Construcția rețelei statice expandate reduse cu drumuri dinamice	29	27
2.1.2 Fluxul maxim în rețele dinamice planare cu margine inferioară zero	30	28

2.2 Fluxuri în rețele dinamice planare. Abordarea dinamică 34 29

2.2.1 Fluxul maxim în rețele dinamice planare cu margine inferioară zero 34 29

2.2.2 Fluxul maxim în rețele dinamice planare cu margine inferioară pozitivă 38 30

2.2.3 Fluxul minim în rețele dinamice planare 46 34

3. FLUXURI ÎN REȚELE DINAMICE BIPARTITE 49 36

3.1 Fluxuri în rețele dinamice bipartite. Abordarea statică 49 36

3.1.1 Fluxuri maxime în rețele dinamice bipartite cu margine inferioară zero 49 36

3.1.2 Fluxuri maxime în rețele dinamice bipartite cu margine inferioară pozitivă 54 38

3.1.3 Fluxuri minime în rețele dinamice bipartite 58 40

3.2 Fluxuri în rețele dinamice bipartite. Abordarea dinamică 62 42

3.2.1 Fluxul maxim în rețele dinamice bipartite cu marginea inferioară zero 62 42

4. APLICAȚII ŞI CONCLUZII 66 44

4.1 Aplicații 66 44

4.2 Concluzii 68 46

BIBLIOGRAFIE SELECTIVĂ 70 50

ANEXE 77 54

Anexa 1. Scurt rezumat al tezei 78 54

Anexa 2. Curriculum vitae 80 56

Universitatea Transilvania din Brașov

CONTENTS

	Thesis page	Abs. page
INTRODUCTION	5	7
1. FLOWS IN NETWORKS	8	10
1.1 Flows in general static networks	8	10
1.1.1 Introductory notions	8	10
1.1.2 Maxium flow	9	11
1.1.3 Minimum flow	12	13
1.2 Flows in special static networks	15	15
1.2.1 Maximum flows in planar static networks	15	15
1.2.2 Minimum flows in planar static networks	19	17
1.2.3 Maximum flows in bipartite static networks	19	18
1.2.4 Minimum flows in bipartite static networks	22	20
1.3 Flows in general dynamic networks	24	21
1.3.1 Maximum flow in general dynamic networks	24	21
1.3.2 Minimum flow in general dynamic networks	28	26
2. FLOWS IN PLANAR DYNAMIC NETWORKS	29	27
2.1 Flows in planar dynamic networks. The static approach	29	27
2.1.1 Construction of the static reduced expanded network with dynamic paths	29	27
2.1.2 Maximum flows in planar dynamic networks with lower bound zero	30	28

2.2 Flows in planar dynamic networks. The dynamic approach 34 29

2.2.1 Maximum flows in planar dynamic netorks with lower bound zero 34 29

2.2.2 Maximum flows in planar dynamic netorks with positive lower bound 38 30

2.2.3 Minimum flows in planar dynamic networks 46 34

3. FLOWS IN BIPARTITE DYNAMIC NETWORKS 49 36

3.1 Flows in bipartite dynamic networks. The static approach 49 36

3.1.1 Maximum flows in bipartite dynamic networks with lower bound zero 49 36

3.1.2 Maximum flows in bipartite dynamic networks with positive lower bound 54 38

3.1.3 Minimum flows in bipartite dynamic networks 58 40

3.2 Flows in bipartite dynamic networks. The dynamic approach 62 42

3.2.1 Maximum flow in bipartite dynamic networks with lower bound zero 62 42

4. APPLICATIONS AND CONCLUSIONS 66 44

4.1 Applications 66 44

4.2 Conclusions 68 46

SELECTED BIBLIOGRAPHY 70 50

APPENDICES 77 54

Appendix 1. PhD. Thesis Abstract 78 54

Appendix 2. Curriculum vitae 80 56

Introducere

Studiul fluxurilor în reţele implică o abordare interdisciplinară la graniţa dintre domenii de cercetare importante: matematici aplicate, informatică, inginerie, management pe de o parte şi cercetări operaţionale pe de altă parte, cunoscând o dezvoltarea importantă nu doar datorită numeroaselor aplicaţii directe ale problemelor din viaţa reală, ci şi datorită faptului că au fost dezvoltaţi algoritmi eficienţi pentru probleme mai generale din domeniul cercetării operaţionale sau al algoritmicii grafurilor. Studiile privind fluxurile în reţele sunt extinse, iar cercetătorii au făcut, în ultimii ani, îmbunătăţiri continue ale algoritmilor pentru rezolvarea mai multor clase de probleme.

Modelele de fluxuri în reţele dinamice nu au fost investigate într-o formă atât de detaliată ca modelele clasice de flux, din cauza complexităţii modelelor dinamice ale fluxului de reţea, în comparaţie cu cele statice. Fluxurile dinamice sunt utilizate pe scară largă pentru a modela structurarea reţelelor, a problemelor de luare a deciziilor în timp: probleme în comunicarea electronică, producţie şi distribuţie, planificarea economică, programarea locurilor de muncă şi transportul. Deşi există metode foarte eficiente de obţinere a soluţiilor pentru problemele legate de fluxul static, problemele de flux în reţele dinamice s-au dovedit mai dificil de rezolvat.

În această lucrare se formulează şi se cercetează modelele dinamice de aflare a fluxului maxim şi a fluxului minim în reţele planare şi bipartite. În continuare prezentăm structura acestei lucrări.

Capitolul 1, **Fluxuri în reţele**, descrie noţiuni generale privind fluxurile în reţele, noţiuni utilizate pe parcursul acestei lucrări. Acest capitol prezintă algoritmi pentru determinarea fluxurilor în reţele statice oarecare (paragraful 1.1) şi în reţele statice speciale (paragraful 1.2), precum şi algoritmi pentru determinarea fluxurilor în reţele dinamice oarecare (paragraful 1.3). Algoritmii prezentaţi în acest capitol stau la baza dezvoltării unor algoritmi prezentaţi în capitolele 2 şi 3.

În capitolul al doilea, **Fluxuri în reţele dinamice planare**, am prezentat modul în care este construită reţeaua statică expandată redusă cu drumuri dinamice. De asemenea, am studiat algoritmi pentru flux maxim, cazul cu margine inferioară zero, folosind abordarea statică

(paragraful 2.1). În paragraful 2.2 am prezentat algoritmi pentru flux maxim, cazul cu margine inferioară zero, cât şi cel cu margine inferioară pozitivă şi algoritmi pentru flux minim, folosind abordarea dinamică. Algoritmii propuşi sunt prezentaţi, în detaliu, în pseudocod. Pentru fiecare algoritm am prezentat câte un exemplu sugestiv. Rezultatele acestui capitol au fost publicate sau sunt trimise spre publicare în lucrările: [30], [31], [66], [67], [69], [70], [71], [73] şi [33].

Capitolul al treilea, **Fluxuri în reţele dinamice bipartite**, este dedicat studiului fluxurilor în reţele dinamice bipartite. În această secţiune am studiat problema fluxului maxim şi cea a fluxului minim în reţele dinamice bipartite folosind abordarea statică (paragraful 3.1). În paragraful 3.2 am studiat problema fluxului maxim în reţele dinamice bipartite cu margine inferioară zero folosind abordarea dinamică. Am rezolvat aceste probleme prin reformularea problemelor în probleme în reţeaua statică bipartită. Rezultatele acestui capitol au fost publicate sau sunt trimise spre publicare în lucrările [32], [65], [68], [70], [72] şi [74].

În ultimul capitol, **Aplicaţii şi concluzii**, sunt prezentate aplicaţiile şi concluziile obţinute în urma cercetării claselor de fluxuri dinamice, fiind descrise contribuţiile aduse temei abordate. În paragraful 4.1 este prezentată o aplicaţie referitor la evacuarea unei clădiri modelată ca o problemă de flux dinamic. Această aplicaţie reprezintă conţinutul unui articol al autoarei acestei teze [66]. În paragraful 4.2 este realizată o analiză comparativă a complexităţii algoritmilor, referitor la cele două abordări, şi anume: abordarea statică şi cea dinamică. Tot aici sunt enunţate câteva probleme de cercetare deschise, legate de rezultatele capitolelor anterioare.

Articolele publicate sau trimise spre publicare sunt următoarele:

a. articole publicate în reviste:

- [65] Annals of the University of Craiova, Mathematics and Computer Science Series (categoria C);
- [67] Bulletin of the Transilvania University of Braşov Series V (categoria D);
- [68] Bulletin of the Transilvania University of Braşov Series III (categoria D);
- [70] The International Journal of Computers Communications and Control (categoria C);
- [71] The International Journal of Computers Communications and Control (categoria C);
- [73] Fundamenta Informaticae, Annales Societatis

Mathematicae Polonae (categoria B).

b. articole prezentate şi publicate în proceedingurile unor conferinţe internaţionale:

- [66] ICMA - The 14^{th} International Conference on Mathematics and its Applications, 5-7th November 2015, Timişoara;
- [72] Proceedings in IEEE Xplore of the 6^{th} International Conference on Computers, Comunications and Control (categoria D);
- [33] 6^{th} International Conference on Control, Decision and Information Technologies, 2019 (categoria D).

c. articole trimise la reviste şi conferinţe spre publicare:

- [32] Informatica an International Journal of Computing and Informatics (categoria C);
- [69] Romanian Journal of Information Science and Technology (categoria C);
- [74] Bulletin of the Transilvania University of Braşov Series III (categoria D).

d. articole conexe realizate de-a lungul studiilor doctorale:

- [31] Journal of Circuits, Systems, and Computers, trimis spre publicare (categoria C);
- [30] Proceedings in IEEE Xplore of the 5^{th} International Conference on Control, Decision and Information Technologies (CoDIT'18) (categoria D).

* * *

Aduc mulţumiri domnului profesor dr. Eleonor Ciurea, coordonatorul acestei teze de doctorat, pentru încrederea pe care a avut-o în mine, pentru răbdarea şi înţelegerea de care a dat dovadă de-a lungul anilor de pregătire a studiilor mele doctorale, pentru sprijinul constant acordat în activitatea ştiinţifică şi pentru faptul că a fost mereu alături de mine şi m-a stimulat să public rezultatele cercetării în reviste şi conferinţe cu vizibilitate internaţională.

1. Fluxuri în reţele

1.1 Fluxuri în reţele statice oarecare

1.1.1 Noţiuni introductive

Definitia 1.1. *Se numeşte graf orientat un triplet* $G = (N, A, q)$ *format din mulţimea arcelor* A*, mulţimea nodurilor* N *şi dintr-o aplicaţie* $q : A \to N \times N$ *numită funcţie de incidenţă care asociază o pereche ordonată* $(i, j) \in N \times N$ *cu* $i \neq j$ *pentru orice arc* $a \in A$*.*

Definitia 1.2. *Dacă oricare pereche ordonată* $(i, j) \in N \times N$ *este imaginea a cel mult* q*,* $q > 1$*, elemente din* A*, atunci* $G = (N, A, q)$ *se numeşte multigraf orientat.*

Definitia 1.3. *Un graf orientat cu* $q = 1$ *se numeşte digraf şi se notează cu* $G = (N, A)$*.*

Definitia 1.4. *Se numeşte lanţ de la nodul* i_1 *la nodul* i_{k+1} *o secvenţă* $L = (i_1, a_1, i_2, \cdots, i_k, a_k, i_{k+1})$*,* $i_j \in N$*,* $a_j \in A$*,* $j = \overline{1, k+1}$ *cu proprietatea că fiecare arc* a_j *este de forma* (i_k, i_{k+1}) *sau* (i_{k+1}, i_k)*. Dacă nodurile din secvenţă sunt distincte atunci lanţul se numeşte elementar.*

Definitia 1.5. *Un lanţ* $L = (i_1, a_1, i_2, \cdots, i_k, a_k, i_{k+1})$ *al grafului orientat* $G = (N, A, q)$*, în care fiecare arc* a_j *este arc direct, adică este de forma* $a_j = (i_j, i_{j+1})$*,* $j = 1, \ldots, k$*, se numeşte lanţ elementar orientat sau drum elementar şi se notează cu* D*.*

Definitia 1.6. *Un lanţ orientat cu* $i_{k+1} = i_1$ *se numeşte ciclu orientat sau circuit şi se notează cu* $\mathring{D}$*.*

Definitia 1.7. *O tăietură este o partiţionare a mulţimii nodurilor* N *în două submulţimi* X *şi* $\bar{X} = N - X$*.*

Definitia 1.8. *O tăietură* $[X, \bar{X}]$ *cu* $1 \in X$ *şi* $n \in \bar{X}$ *se numeşte tăietură* $1 - n$*.*

Definitia 1.9. *O reţea statică* $G = (N, A, l, u)$ *este un digraf* $G = (N, A)$*,* $N = \{1, \ldots, i, \ldots, n\}$*,* $A = \{a_1, \ldots, a_k, \ldots, a_m\}$ *în care se definesc funcţiile* $l : A \to \mathbb{R}_+$ *numită margine inferioară şi* $u : A \to \mathbb{R}_+$ *numită margine superioară (capacitate) cu* $l \leq u$*. Nodul* 1 *reprezintă nodul sursă şi nodul* n *reprezintă nodul stoc.*

Definitia 1.10. *Un flux este o funcţie* $f : A \to \mathbb{R}_+$ *care satisface*

următoarele condiţii:

$$f(i,N) - f(N,i) = \begin{cases} v, & \text{dacă } i = 1 \\ 0, & \text{dacă } i \neq 1, n \\ -v, & \text{dacă } i = n \end{cases} \tag{1.1a}$$

cu valoarea fluxului $v \geq 0$, $f(i,N) = \sum_{j \in N} f(i,j)$, $f(N,i) = \sum_{k \in N} f(k,i)$ *şi* $f(i,j) = 0$ *pentru perechile* $(i,j) \notin A$. *Dacă sunt îndeplinite şi condiţiile:*

$$l(i,j) \leq f(i,j) \leq u(i,j), \quad (i,j) \in A \tag{1.1b}$$

atunci f *este un flux admisibil.*

Definitia 1.11. *Capacitatea reziduală* $r(i,j)$ *pentru problema fluxului maxim a oricărui arc* $(i,j) \in A$, *în reţeaua* $G = (N, A, l, u)$ *este dată de fluxul adiţional maxim care poate fi trimis de la nodul* i *la nodul* j *utilizând arcele* (i,j) *şi* (j,i) *şi este* $r(i,j) = u(i,j) - f(i,j) + f(j,i) - l(j,i)$.

1.1.2 Fluxul maxim

Problema fluxului maxim în reţele statice cu margini inferioare pozitive îşi propune să găsească o soluţie care trimite fluxul maxim de la un nod (sursă) la alt nod (destinaţie), cu constrângerea că marginile inferioare şi superioare ale arcelor, în reţea, sunt satisfăcute. Mai mulţi algoritmi eficienţi au fost dezvoltaţi pentru rezolvarea acestei probleme şi sunt prezentaţi în lucrarea autorilor Ahuja, Magnanti şi Orlin [2] şi în lucrările altor autori prezentaţi în bibliografie.

Definitia 1.12. *Pentru problema fluxului maxim definim capacitatea tăieturii* $1-n$ $[X, \bar{X}]$ *astfel:*

$$c[X, \bar{X}] = \sum_{(X, \bar{X})} u(i,j) - \sum_{(\bar{X}, X)} l(j,i) \tag{1.2}$$

Dacă folosim notaţia $\sum_{(X,\bar{X})} b(i,j) = b(X, \bar{X})$ atunci pentru capacitatea tăieturii $[X, \bar{X}]$ putem folosi relaţia

$$c[X, \bar{X}] = u(X, \bar{X}) - l(\bar{X}, X) \tag{1.3}$$

Dacă $l(i,j) = 0$ pentru $(i,j) \in A$ atunci $c[X, \bar{X}] = u(X, \bar{X})$ este relaţia de definiţie a capacităţii unei tăieturi în cazul în care marginile inferioare sunt nule.

Definitia 1.13. *O tăietură* $1-n$ *a cărei capacitate este minimă între toate tăieturile* $1-n$ *se numeşte tăietură* $1-n$ *minimă şi se notează* $[X^*, \bar{X}^*]$.

Teorema 1.1. *Dacă f este un flux de valoare v în reţeaua $G = (N, A, l, u)$ şi $[X, \bar{X}]$ o tăietură $1 - n$ atunci f verifică relaţiile:*

$$v = f[X, \bar{X}] \leq c[X, \bar{X}], \quad f[X, \bar{X}] = f(X, \bar{X}) - f(\bar{X}, X) \tag{1.4}$$

Teorema 1.2. *Într-o reţea $G = (N, A, l, u)$ valoarea fluxului maxim de la nodul sursă 1 la nodul stoc n este egală cu capacitatea tăieturii $1 - n$ minime $[X^*, \bar{X^*}]$, adică:*

$$v = f[X^*, \bar{X^*}] = c[X^*, \bar{X^*}] \tag{1.5}$$

Pentru problema fluxului maxim un preflux f este o funcţie $f : A \to \mathbb{N}$ care satisface următoarele condiţii:

$$f(N, i) - f(i, N) \geq 0, i \in N - \{1, n\} \tag{1.6a}$$

$$l(i, j) \leq f(i, j) \leq u(i, j), (i, j) \in A \tag{1.6b}$$

Un pseudoflux este o funcţie care satisface doar constrângerea 1.6b.

Pentru un preflux f excesul fiecărui nod $i \in N$ este

$$e(i) = f(N, i) - f(i, N) \tag{1.7}$$

şi dacă $e(i) > 0$, $i \in N - \{1, n\}$ atunci vom spune că nodul i este un nod activ. Dacă $e(i) = 0$ atunci i se numeşte nod echilibrat. Un preflux f care satisface condiţia $e(i) = 0$, $\forall\, i \in N - \{1, n\}$ este un flux. Astfel, un flux este un caz particular de preflux.

Reţeaua reziduală în raport cu fluxul (prefluxul) f este $\tilde{G} = (N, \tilde{A}, r)$ cu $\tilde{A} = \{(i, j) | (i, j) \in A, r(i, j) > 0\}$. În reţeaua reziduală $\tilde{G} = (N, \tilde{A}, r)$ definim funcţia distanţă $d : N \to \mathbb{N}$. Spunem că funcţia distanţă este validă dacă satisface următoarele două condiţii :

$$d(n) = 0 \tag{1.8a}$$

$$d(i) \leq d(j) + 1, (i, j) \in \tilde{A} \tag{1.8b}$$

Ne referim la $d(i)$ ca la eticheta distanţă a nodului i. Spunem că un arc $(i, j) \in \tilde{A}$ este admisibil dacă satisface condiţia că $d(i) = d(j) + 1$; ne referim la toate celelalte arce ca fiind arce inadmisibile. De asemenea, ne referim la un drum de la nodul 1 la nodul k constând numai din arce admisibile ca la un drum admisibil.

În timp ce problema fluxului admisibil cu margine inferioară zero întotdeauna are o soluţie admisibilă (deoarece fluxul zero este admisibil), problema cu margine inferioară pozitivă poate să nu aibă soluţii admisibile. Prin urmare, problema fluxului maxim cu margini inferioare pozitive poate fi rezolvată în două faze:

– (F1) această fază determină un flux admisibil dacă există unul;

– (F2) această fază converteşte un flux admisibil într-un flux maxim.

Problema din fiecare fază se reduce la rezolvarea unei probleme de flux maxim cu margini inferioare zero. În consecinţă, este posibilă rezolvarea problemei de flux maxim cu margini inferioare pozitive prin rezolvarea a două probleme de flux maxim, fiecare cu margine inferioară zero.

În urma aplicării unui algoritm pentru flux maxim, la terminarea algoritmului se obţin capacităţi reziduale optime din care se poate construi fluxul maxim, astfel:

$$f(i,j) = l(i,j) + \max\{0, u(i,j) - r(i,j) - l(i,j)\}$$
$$f(j,i) = l(j,i) + \max\{0, u(j,i) - r(j,i) - l(j,i)\}.$$

Complexitatea unor algoritmi depinde de $\bar{u} = max\{u(i,j)|(i,j) \in A\}$.

1.1.3 Fluxul minim

Problema fluxului minim constă în determinarea unui flux f pentru care v este minim.

Pentru problema fluxului minim un preflux f este o funcţie $f : A \to \mathbb{N}$ care satisface următoarele condiţii:

$$f(i,N) - f(N,i) \leq 0, i \in N - \{1,n\} \tag{1.9a}$$

$$l(i,j) \leq f(i,j) \leq u(i,j), (i,j) \in A \tag{1.9b}$$

Pentru un preflux f deficitul fiecărui nod $i \in N$ este

$$e(i) = f(i,N) - f(N,i) \tag{1.10}$$

şi dacă $e(i) < 0$, $i \in N - \{1,n\}$ atunci vom spune că nodul i este un nod activ.

Presupunem, fără pierderea generalităţii, că dacă $(i,j) \in A$ atunci $(j,i) \in A$ (dacă $(j,i) \notin A$ considerăm că $(j,i) \in A$ cu $l(j,i) = u(j,i) = 0$).

Pentru problema fluxului minim, capacitatea $c[X,\bar{X}]$ a tăieturii $1-n$ este $c[X,\bar{X}] = l(X,\bar{X}) - u(\bar{X},X)$. Ne referim la o tăietură $1-n$ a cărei capacitate este maximă de-a lungul tuturor tăieturilor $1-n$ ca la o tăietură maximă.

Teorema 1.3. *Valoarea fluxului minim de la nodul sursă* 1 *la nodul stoc* n *în reţeaua* $G = (N,A,l,u)$ *este egală cu capacitatea tăieturii maxime* $1-n$.

În problema fluxului minim, capacitatea reziduală $\hat{r}(i,j)$ pentru orice arc $(i,j) \in A$ cu privire la un flux dat (preflux) f este dată de $\hat{r}(i,j) =$

$u(j,i) - f(j,i) + f(i,j) - l(i,j)$. Reţeaua reziduală este $\hat{G} = (N, \hat{A}, \hat{r})$ cu $\hat{A} = \{(i,j)|(i,j) \in A, \hat{r}(i,j) > 0\}$.

Problema fluxului minim în reţeaua $G = (N, A, l, u)$ poate fi rezolvată în două faze:

- (F1) se determină un flux admisibil dacă există;
- (F2) pornind de la un flux admisibil, se determină un flux minim;

Există trei abordări pentru rezolvarea problemei fluxului minim:

(a) folosind algoritmi cu drumuri de micşorare a fluxului;
(b) folosind algoritmi cu prefluxuri;
(c) folosind algoritmi minmax (algoritmi pentru fluxul maxim aplicaţi de la nodul n la nodul 1).

Definim în continuare noţiunea de etichetă distanţă în reţeaua reziduală $\hat{G}(f) = (\hat{N}, \hat{A})$, unde $\hat{A} = \{(i,j)|(i,j) \in A, \hat{r}(i,j) > 0\}$.

Definitia 1.14. *Se numeşte funcţie distanţă în reţeaua reziduală $\hat{G}(f) = (\hat{N}, \hat{A})$ relativ la prefluxul f o funcţie $\hat{d} : \hat{N} \to \mathbb{N}$ cu $\hat{N} = N$.*

Definitia 1.15. *Funcţia distanţă este validă dacă satisface următoarele condiţii:*

$$\hat{d}(1) = 0 \tag{1.11a}$$

$$\hat{d}(j) \leq \hat{d}(i) + 1, (i,j) \in \hat{A}, r(i,j) > 0 \tag{1.11b}$$

Valoarea $\hat{d}(i)$ se numeşte eticheta distanţă a nodului i, iar condiţiile (1.11a) şi (1.11b) se numesc condiţii de validitate.

Definitia 1.16. *Etichetele distanţă se numesc exacte dacă pentru fiecare nod i, $\hat{d}(i)$ este egală cu lungimea celui mai scurt drum de la nodul 1 la nodul i în reţeaua reziduală $\hat{G}(f)$.*

Definitia 1.17. *Un arc (i,j) din reţeaua reziduală $\hat{G}(f)$ se numeşte arc admisibil dacă el satisface condiţia: $\hat{d}(j) = \hat{d}(i) + 1$, altfel arcul se numeşte arc inadmisibil.*

Definitia 1.18. *Un drum de la nodul sursă 1 la nodul stoc n din reţeaua reziduală $\hat{G}(f)$ se numeşte drum admisibil dacă el conţine numai arce admisibile, altfel se numeşte drum inadmisibil.*

Fluxul minim în reţeaua reziduală $\hat{G}(f)$ este determinat de relaţiile:

$$f(i,j) = l(i,j) + \max(\hat{r}(i,j) - u(j,i) + l(j,i), 0) \tag{1.12a}$$

$$f(j,i) = l(j,i) + \max(\hat{r}(j,i) - u(i,j) + l(i,j), 0) \tag{1.12b}$$

1.2 Fluxuri în reţele statice speciale

1.2.1 Fluxuri maxime în reţele statice planare

Un digraf $G = (N, A)$ se numeşte planar dacă poate fi reprezentat în plan astfel încât fiecărui nod să-i corespundă un punct al planului, iar arcelor să le corespundă curbe simple ce unesc punctele corespunzătoare nodurilor şi în plus aceste arce să se intersecteze (eventual) numai în noduri. Analog se defineşte un graf simplu neorientat planar. Arcele (muchiile) unui graf planar determină contururi (frontiere) care mărginesc regiuni numite feţe. Există o singură regiune din plan fără contur numită faţă nemărginită.

Testarea planarităţii unui digraf este o problemă rezolvată prin descrierea unor algoritmi în timp liniar, (Hopcroft şi Tarjan [53]).

În acest subparagraf ne vom referi la o clasă specială a digrafurilor planare cunoscută sub denumirea de digrafuri planare $(1, n)$ (nodul sursă 1 şi nodul destinaţie n se găsesc pe frontiera feţei nemărginite).

Fie $G = (N, A, l = 0, u)$ o reţea statică cu mulţimea nodurilor $N = \{1, \ldots, i, \ldots, j, \ldots, n\}$, mulţimea arcelor $A = \{a_1, \ldots, a_k, \ldots, a_m\}$, $a_k = (i, j)$, funcţia margine inferioară $l : A \to \mathbb{N}$, $l = 0$ şi funcţia margine superioară (capacitate) $u : A \to \mathbb{R}_+$.

Reamintim că un flux în reţeaua statică $G = (N, A, l = 0, u)$ este o funcţie $f : A \to \mathbb{R}_+$ care satisface următoarele condiţii:

$$\sum_j f(i,j) - \sum_k f(k,i) = \begin{cases} v, & \text{if } i = 1 \\ 0, & \text{if } i \neq 1, n \\ -v, & \text{if } i = n \end{cases} \tag{1.13a}$$

$$0 \leq f(i,j) \leq u(i,j), \quad (i,j) \in A \tag{1.13b}$$

pentru orice $v \geq 0$.

Ne vom referi la v ca la valoarea fluxului static f. Problema fluxului maxim constă în determinarea unui flux f care să maximizeze valoarea lui v.

Dacă reţeaua planară $G = (N, A, l = 0, u)$ sau echivalent $G = (N, A, u)$ este conexă şi are n noduri, m arce şi q feţe, atunci $m < 3n$ şi $q = m - n + 2$.

Prezentăm, în continuare problema fluxului în reţeaua planară orientată $(1, n)$. Pentru asta vom defini mai întâi reţeaua orientată duală $G' = (N', A', u')$. Adăugăm în reţeaua G arcul $(n, 1)$ cu $u(n, 1) = 0$ care împarte faţa nemărginită în două feţe: o faţă mărginită şi o faţă

nemărginită. În acest caz avem $q^{'} = q + 1$ feţe. Apoi adăugăm un nod i' în interiorul fiecărei feţe x a reţelei G. Fie $1'$ şi n' nodurile în reţeaua orientată dual ă G' corespunzătoare noii feţe mărginite, respectiv noii feţe nemărginite. Fiecare arc $(i, j) \in A$ se află pe frontiera dintre cele două feţe x şi y. Corespunzător acestui arc, reţeaua G' conţine două arce opuse (i', j') şi (j', i'). Dacă arcul (i, j) este în sensul acelor de ceasornic al feţei x, definim costul $c'(i', j') = u(i, j)$ şi costul $c'(j', i') = 0$. Definim costurile arcelor în mod contrar dacă (i, j) este un arc în sens invers acelor de ceasornic al feţei x. Reţeaua dual ă orientată G' conţine arcele $(1', n')$ şi $(n', 1')$ care vor fi şterse din reţeaua G'. Mulţimea $A' = \{(i', j'), (j', i') | i', j' \in N'\}$, (i', j') şi (j', i') corespund lui $(i, j) \in A$. Există o corespondenţă unu la unu între tăietura $1 - n$ în reţeaua G şi drumul de la nodul $1'$ la nodul n' în reţeaua G'. Mai mult decât atât capacitatea tăieturii este egală cu costul corespunzător drumului. Prin urmare, putem obţine o tăietură $1 - n$ minimă $[X, \bar{X}]$ şi $c[X, \bar{X}]$ în reţeaua G prin determinarea celui mai scurt drum P' şi $c(P')$ de la nodul $1'$ la nodul n' în reţeaua G'. Putem rezolva problema celui mai scurt drum în reţeaua G' folosind algoritmul Dijkstra, (Ahuja, Magnanti şi Orlin [2]).

Prezentăm mai jos algoritmul pentru determinarea fluxului maxim în reţeaua statică planară $G = (N, A, u)$.

Fie $d'(i')$ distanţa celui mai scurt drum de la nodul $1'$ la nodul i' în reţeaua orientată dual ă $G' = (N', A', u')$. Algoritmul pentru fluxul maxim în reţeaua statică planară orientată $(1, n)$ este algoritmul (MFPSN), (Ford şi Fulkerson [40], Hassin [51], Itai şi Shiloach [54]):

Algoritmul 1: Algoritmul pentru determinarea fluxului maxim în reţeaua statică planară orientată $(1, n)$

```
   MFPSN;
1  BEGIN
2     construieşte reţeaua G';
3     DIJKSTRA (G', d');
4        for (i, j) ∈ A do
5           f(i, j) := d'(j') − d'(i');
6  END;
```

Teorema 1.4. *Algoritmul MFPSN determină un flux maxim în reţeaua G.*

Teorema 1.5. *Complexitatea algoritmului MFPSN este $O(n^2)$.*

Utilizând algoritmul Frederickson ([41]), procedura MFPSN determină un flux maxim în timp $O(n^{1.5})$.

Reamintim că problema fluxului maxim într-o reţea generală cu mai multe noduri sursă şi mai multe noduri stoc este echivalentă cu

problema fluxului maxim într-o reţea extinsă cu un singur nod sursă şi un singur nod stoc. Pentru reţele planare această extindere poate distruge planaritatea reţelei. De aceea este necesară o altă metodă. Fie $N = N_1 \cup N_2 \cup N_3$ cu N_1 mulţimea nodurilor sursă, N_2 mulţimea nodurilor tranzit şi N_3 mulţimea nodurilor stoc.

În continuare prezentăm algoritmul pentru determinarea fluxului maxim în reţele statice planare orientate cu toate nodurile sursă şi toate nodurile stoc situate pe faţa exterioară, (Miller şi Naor [62]).

Algoritmul 2: Algoritmul pentru determinarea fluxului maxim în reţele statice planare orientate cu toate nodurile sursă şi toate nodurile stoc situate pe faţă exterioară.

```
   MFDPSNSSOF;
1  BEGIN
2     DIV(N_1, N_3, X, Y);
3     MFXY(G, X, Y, f_1^*, G̃_1);
4     MFC(G̃_1, C̃, f_2^*);
5     MFG(f_2^*, G̃_2, f_3^*, f_0^*);
6  END.
```

Din lucrarea Miller şi Naor [62] prezentăm următoarele două teoreme.

Teorema 1.6. *Algoritmul MFDPSNSSOF calculează în mod corect un flux maxim.*

Teorema 1.7. *Timpul de funcţionare al algoritmului MFDPSNSSOF este* $O(n \log^{1.5} n)$.

1.2.2 Fluxuri minime în reţele statice planare

În continuare considerăm că reţeaua $G = (N, A, l, u)$ este o reţea $(1, n)$ planară orientată.

Reţeaua $G' = (N', A', l', u')$ este reţeaua duală a reţelei $(1, n)$ planare $G = (N, A, l, u)$ unde $N' = \{1', \ldots, i', \ldots, n'\}$, $A' = \{(i', j'), (j', i') | i', j' \in N\}$, (i', j') şi (j', i') corespund lui $(i, j) \in A$, cu $n' = |N'| = m - n + 3$ şi $m' = |A'| = 2m$.

Dacă arcul (i, j) este în sensul acelor de ceasornic în faţa x, definim $l'(i', j') = l(i, j)$ şi $u'(j', i') = -u(i, j)$. Definim l' şi u' în mod contrar dacă (i, j) este un arc în sens invers arcelor de ceasornic al feţei x, (Ciurea şi Georgescu [25]).

Algoritmul pentru fluxul minim $\mathring{f}$ în reţeaua statică planară $(1, n)$ $G = (N, A, l, u)$ este prezentat mai jos.

Algoritmul 3: Algoritmul pentru determinarea fluxului minim în reţeaua statică planară $(1, n)$ $G = (N, A, l, u)$.

```
   mFPSN;
1  BEGIN
2     BELLMAN-FORD (G', d');
3     for (i, j) ∈ A do
4        f°(i, j) := d'(j') − d'(i');
5  END.
```

Din lucrarea Ciurea şi Georgescu [25] prezentăm următoarele două teoreme:

Teorema 1.8. *Dacă fluxul $\mathring{f}$ determinat cu algoritmul mFPSN satisface condiţiile $f(i,j) \leq u(i,j)$ pentru toate arcele $(i,j) \in A$, atunci fluxul $\mathring{f}$ este fluxul minim admisibil.*

Teorema 1.9. *Algoritmul mFPSN are complexitatea $O(n^2)$.*

Remarcăm faptul că dacă fluxul minim $\mathring{f}$ obţinut cu algoritmul mFPSN este inadmisibil ($u(i,j) < f(i,j)$ pentru unul sau mai multe arce $(i,j) \in A$), atunci nu există flux admisibil în reţeaua G.

1.2.3 Fluxuri maxime în reţele statice bipartite

Reţelele statice bipartite apar în contexte practice cum ar fi: problema eliminării în competiţiile de baseball, testarea fiabilităţii reţelei, şi prin urmare, este interesant să găsim algoritmi eficienţi pentru această clasă de probleme.

Definitia 1.19. *Spunem că digraful $G = (N, A)$ este digraf bipartit dacă mulţimea nodurilor N admite o partiţie în două submulţimi N_1 şi N_2 astfel încât pentru fiecare arc $(i,j) \in A$ avem fie $i \in N_1$ şi $j \in N_2$, fie $i \in N_2$ şi $j \in N_1$, unde $N_1 \neq \emptyset$, $N_2 \neq \emptyset$, $N_1 \cap N_2 = \emptyset$, $N_1 \cup N_2 = N$.*

Notăm cu $G = (N_1 \cup N_2, A, l, u)$ reţeaua bipartită. Fie $n_1 = |N_1|$ şi $n_2 = |N_2|$. Deducem că $|N| = n_1 + n_2$.

Una dintre problemele care se pune este dacă un graf dat este sau nu un graf bipartit. Din fericire există multe metode simple pentru rezolvarea acestei probleme. Una dintre acestea are la bază următoarea proprietate ce caracterizează reţelele bipartite, (Jungnickel [56]).

Proprietatea 1. *Un graf G este graf bipartit dacă şi numai dacă fiecare ciclu în G conţine un număr par de arce.*

Prin îmbunătăţirea compexităţii algoritmilor Dinic [36] şi Karzanov, Naor [57], Gusfield, Martel şi Fernandez-Baca [49] au descris algoritmi pentru flux maxim pe reţele bipartite. Ahuja, Orlin, Stein şi Tarjan [1] au adus îmbunătăţiri importante şi au demonstrat că este posibil să se obţină o nouă limită de timp pentru reţele bipartite. În lucrarea

[49] Gusfield, Martel şi Fernandez-Baca descriu probleme care pot fi rezolvate cu ajutorul reţelelor cu flux în grafuri bipartite.

Fără nicio pierdere de generalitate, presupunem că $n_1 \leq n_2$. De asemenea, presupunem că nodul sursă 1 aparţine mulţimii N_2 (dacă nodul sursă 1 aparţine mulţimii N_1, atunci putem crea un nou nod sursă $1' \in N_2$, şi putem adăuga un arc $(1', 1)$ cu $l(1', 1) = 0$, $u(1', 1) = \infty$). O reţea bipartită se numeşte neechilibrată dacă $n_1 << n_2$ şi echilibrată în caz contrar.

Ahuja, Orlin, Stein şi Tarjan [1] au obţinut timpi de execuţie îmbunătăţiţi prin modificarea algoritmilor. Această modificare se aplică numai pentru algoritmii de înaintare a prefluxului şi se numeşte regula de înaintare pe două arce. În concordanţă cu această regulă, întotdeauna se înaintează fluxul de la un nod în N_1 pe două arce la un alt nod din N_1, într-o etapă numită *bi-înaintare*, astfel încât să nu se acumuleaze excese la noduri în N_2.

Versiunea modificată a algoritmului preflux FIFO pentru determinarea fluxului maxim în reţeaua bipartită este numit Algoritmul preflux FIFO bipartit. O bi-înaintare este o înaintare peste două arce admisibile consecutive. El mută excesul dintr-un nod $i \in N_1$ în alt nod $k \in N_1$. Prin această abordare algoritmul mută fluxul de-a lungul drumului $\tilde{D} = (i, j, k), j \in N_2$, şi asigură faptul că niciun nod din N_2 nu a avut vreodată un exces. O înaintare a α unităţi de la nodul i la nodul j scade atât $e(i)$, cât şi $r_0(i, j)$ cu α unităţi şi creşte $e(j)$ şi $r_0(j, i)$ cu α unităţi (Ahuja, Orlin, Stein şi Tarjan [1]). Precizăm faptul că pentru $l = 0$ şi $f = 0$ obţinem $r = u$.

În lucrarea [1] Ahuja, Orlin, Stein şi Tarjan prezintă algoritmul preflux FIFO (BFIFOP) redat, în detaliu, mai jos.

Algoritmul 4: Algoritmul preflux FIFO bipartit (BFIFOP) pentru fluxul maxim

```
   BFIFOP;
1  BEGIN
2    INIŢIALIZARE;
3    while Q ≠ ∅ do
4      BEGIN
5        selectează nodul i de la începutul listei Q;
6        BIÎNAINTARE/REETICHETARE(i);
7      END;
8  END.

   PROCEDURA INIŢIALIZARE;
1  BEGIN
2    f =: 0; Q := ∅; se construieşte reţeaua reziduală G̃;
```

```
3    se înaintează u(1, j) unităţi de flux pe fiecare arc (1, j) ∈ A şi se adaugă
     nodul j la sfârşitul listei Q dacă e(j) > 0 şi j ≠ n;
4    se calculează etichetele distanţă exacte d(i) de la nodul n la nodul i în
     reţeaua reziduală; d(1) = 2n_1 + 1;
5  END.

   PROCEDURA BIÎNAINTARE/REETICHETARE(i);
1  BEGIN
2    if există un arc admisibil (i, j)
3      BEGIN
4        selectează un arc admisibil (i, j);
5        if există un arc admisibil (j, k)
6          BEGIN
7            selectează arcul admisibil (j, k);
8            înaintează α := min {e(i), r(i, j), r(j, k)} unităţi de flux de-a
             lungul drumului (i, j, k) şi adaugă nodul k la lista Q
             dacă k ∉ Q şi k ≠ n;
9          END
10       else d(j) := min{d(k) + 1|(j, k) ∈ A, r(j, k) > 0};
11     END
12   else d(i) := min{d(j) + 1|(i, j) ∈ A, r(i, j) > 0};
13 END;
```

Teorema 1.10. *Algoritmul preflux BFIFOP determină un flux maxim în reţeaua* $G = (N_1 \cup N_2, A, u)$.

Teorema 1.11. *Algoritmul preflux BFIFOP are complexitatea* $n_1^2 n$.

Mai multe informaţii se găsesc în lucrarea Ahuja, Orlin, Stein şi Tarjan [1]. Precizăm faptul că acest algoritm se execută pe reţeaua G cu $l = 0$, un singur nod sursă 1 şi un singur nod stoc n.

1.2.4 Fluxuri minime în reţele statice bipartite

Pentru problema fluxului minim, fără nicio pierdere de generalitate, presupunem că $n_2 \leq n_1$. De asemenea, presupunem că nodul sursă 1 aparţine lui N_2 (dacă nodul sursă 1 aparţine lui N_1, atunci putem crea un nou nod sursă $1' \in N_2$, şi putem adăuga un arc $(1', 1)$ cu $l(1', 1) = 0$, $u(1', 1) = \infty$). Pentru problema fluxului minim o reţea bipartită se numeşte neechilibrată dacă $n_2 << n_1$ şi echilibrată în caz contrar.

Definitia 1.20. *Se numeşte bi-înapoiere cu* $i, k \in N_2$ *şi* $j \in N_1$, *o înapoiere cu p unităţi de flux pe drumul* (i, j, k) *în reţeaua bipartită.*

Autorii lucrării [28], Ciurea, Georgescu şi Marinescu, obţin îmbunătăţiri ale timpilor de execuţie prin modificarea algoritmilor. Această modificare se aplică numai pentru algoritmul FIFO de înapoiere a fluxului şi se numeşte regula de înapoiere pe două arce. Deoarece se înapoiază întotdeauna fluxul pe drumul (i, j, k) cu $i, k \in N_2$

şi $j \in N_1$ rezultă că niciun deficit nu se acumulează la noduri din N_1.

Algoritmul 5: Algoritmul preflux FIFO bipartit (BFIFOPP) pentru fluxul minim

```
   BFIFOPP;
1  BEGIN
2    INIŢIALIZARE;
3    while Q ≠ ∅ do
4      BEGIN
5        selectează nodul k de la începutul listei Q;
6        BIÎNAPOIERE/REETICHETARE(k);
7      END;
8  END.
   PROCEDURA INIŢIALIZARE;
1  BEGIN
2    fie f un flux admisibil în G; Q := ∅; se construieşte reţeaua reziduală Ĝ;
3    se calculează etichetele distanţă exacte d̂(i) de la 1 la i în Ĝ;
4    d̂(n) = 2n_2 + 1;
5    se retrag r̂(i,n) de flux pe arcele ((i,n) ∈ Ã) şi se adaugă i la sfârşitul
     listei Q dacă e(i) < 0 şi i ≠ 1;
6  END.

   PROCEDURA BIÎNAPOIERE/REETICHETARE(k);
1  BEGIN
2    if există un arc admisibil (j,k)
3      BEGIN
4        selectează un arc admisibil (j,k);
5        if există un arc admisibil (i,j)
6          BEGIN
7            selectează un arc admisibil (i,j);
8            înapoiază α := min{−e(k), r(j,k), r(i,j)} unităţi de flux de-a
             lungul drumului (i,j,k) şi adaugă nodul i la începutul listei Q
             dacă i ∉ Q şi i ≠ 1;
9          END
10       else d(j) := min{d(i) + 1|(i,j) ∈ A, r(i,j) > 0};
11     END
12   else d(k) := min{d(j) + 1|(j,k) ∈ A, r(j,k) > 0};
13 END;
```

Teorema 1.12. *Algoritmul preflux BFIFOPP determină un flux minim în reţeaua* $G = (N_1 \cup N_2, A, u)$.

Teorema 1.13. *Algoritmul preflux BFIFOPP are complexitatea* $n_2^2 n$.

1.3 Fluxuri în reţele dinamice oarecare

1.3.1 Fluxul maxim în reţele dinamice oarecare

Definitia 1.21. *O reţea* $D = (N, A, h, e, q, H)$ *cu* N *mulţimea nodurilor,* A *mulţimea arcelor,* h *funcţia timp* $h : A \times H \to \mathbb{N}$, e *funcţia margine inferioară* $e : A \times H \to \mathbb{N}$, q *funcţia margine superioară (capacitate)* $q : A \times H \to \mathbb{N}$ *şi* H *mulţimea perioadelor de timp*

$H = \{0, 1, \ldots, T\}$ *se numeşte reţea dinamică.*

Definitia 1.22. *O reţea* $D = (N, A, h, e, q, H)$ *este numită reţea dinamică staţionară dacă funcţiile* h, e, q *ale tuturor arcelor* $(i, j) \in A$ *sunt independente de timp* $h(i, j; \theta) = h(i, j)$, $e(i, j; \theta) = e(i, j)$ *şi* $q(i, j; \theta) = q(i, j)$ $\forall (i, j) \in A$ *şi* $\forall \theta \in \{0, 1, \ldots, T\}$.

Modelele de reţele dinamice apar în multe probleme incluzând sistemele de producţie - distribuţie, planificare economică, sistemele energetice, problema de transport şi problema evacuării clădirilor.

Literatura de specialitate tratează problema fluxului maxim numai în reţele dinamice $D = (N, A, h, e = 0, q, H)$.

Fie $D = (N, A, h, q, H)$ o reţea dinamică cu $e = 0$, mulţimea nodurilor $N = \{1, \ldots, n\}$, mulţimea arcelor $A = \{a_1, \ldots, a_m\}$, 1 nodul sursă şi n nodul stoc. Parametrul $h(i, j; t)$ este timpul de tranzit necesar pentru a traversa un arc (i, j). Parametrul $q(i, j; t)$ reprezintă cantitatea maximă a fluxului care poate traversa de-a lungul arcului (i, j) când fluxul pleacă de la nodul i la momentul t şi soseşte la nodul j la momentul $\theta = t + h(i, j; t)$.

Problema fluxului dinamic maxim pentru T perioade de timp constă în determinarea funcţiei $g : A \times H \to \mathbb{N}$, care satisface următoarele condiţii în reţeaua dinamică $D = (N, A, h, e = 0, q, H)$:

$$\sum_{t=0}^{T} (g(1, N; t) - \sum_{\tau} g(N, 1; \tau)) = w \tag{1.14a}$$

$$g(i, N; t) - \sum_{\tau} g(N, i; \tau) = 0, i \neq 1, n, \ t \in H \tag{1.14b}$$

$$\sum_{t=0}^{T} (g(n, N; t) - \sum_{\tau} g(N, n; \tau)) = -w \tag{1.14c}$$

$$0 \leq g(i, j; t) \leq q(i, j; t), \quad (i, j) \in A \quad , \quad t \in H \tag{1.15}$$

$$max \quad w, \tag{1.16}$$

unde $\tau = t - h(k, i; t)$, $w = \sum_{t=0}^{T} v(t)$, $v(t)$ este valoarea fluxului la momentul t şi $g(i, j; t) = 0$ pentru toţi $t \in \{T - h(i, j; t) + 1, \ldots, T\}$.

Problema găsirii fluxului maxim în reţeaua dinamică $D = (N, A, h, e = 0, q, H)$ este mult mai complexă decât problema găsirii fluxului maxim în reţeaua statică $G = (N, A, u)$. Din fericire, această situaţie a fost rezolvată prin reformularea problemei din reţeaua dinamică D într-o problemă în reţeaua statică $R = (V, E, u)$ numită reţea statică expandată (Ahuja, Magnanti şi Orlin [2], Ford şi Fulkerson [39]).

Reţeaua statică expandată a reţelei dinamice $D = (N, A, h, e = 0, q, H)$ este o reţea $R = (V, E, u)$ cu $V = \{i_t | i \in N, t \in H\}$, $E = \{(i_t, j_\theta) | (i, j) \in A, t \in \{0, 1, \ldots, T - h(i, j; t)\}, \theta = t + h(i, j; t), \theta \in H\}$, $u(i_t, j_\theta) = q(i, j; t), (i_t, j_\theta) \in E$. Numărul de noduri în reţeaua statică expandată R este $n(T + 1)$ şi numărul de arce este limitat de $m(T + 1) - \sum\limits_{A} \mathring{h}(i, j)$, unde $\mathring{h}(i, j) = min\{h(i, j; 0), \ldots, h(i, j; T)\}$. Este uşor de observat că orice flux în reţeaua dinamică D de la nodul sursă 1 la nodul destinaţie n este echivalent cu un flux în reţeaua statică expandată R de la nodurile sursă $1_0, 1_1, \ldots, 1_T$ la nodurile stoc $n_0, n_1, \ldots, n_T$ şi vice-versa. Putem reduce, în continuare, problema surselor multiple şi a stocurilor multiple în reţeaua statică expandată R la un singur nod sursă şi un singur nod stoc prin introducerea unui supernod sursă 0 şi a unui supernod stoc $n + 1$ construind reţeaua statică super-expandată $R_2 = (V_2, E_2, u_2)$, unde $V_2 = V \cup \{0, n + 1\}$, $E_2 = E \cup \{(0, 1_t) | t \in H\} \cup \{(n_t, n + 1) | t \in H\}$, $u_2(i_t, j_\theta) = u(i_t, j_\theta)$, $(i_t, j_\theta) \in E$, $u_2(0, 1_t) = u_2(n_t, n + 1) = \infty$, $t \in H$.

Acum, vom construi reţeaua statică expandată redusă $R_1 = (V_1, E_1, u_1)$ după cum urmează: definim funcţia $h_2 : E_2 \longrightarrow \mathbb{N}$, cu $h_2(0, 1_t) = h_2(n_t, n + 1) = 0$, $t \in H$, $h_2(i_t, j_\theta) = h(i, j; t)$, $(i_t, j_\theta) \in E$. Fie $d_2(0, i_t)$ lungimea celui mai scurt drum de la nodul sursă 0 la nodul i_t, şi $d_2(i_t, n + 1)$ lungimea celui mai scurt drum de la nodul i_t la nodul stoc $n + 1$, referitor la h_2 în reţeaua R_2. Lungimile $d_2(0, i_t)$ şi $d_2(i_t, n+1)$ pentru toţi $i_t \in V$ se calculează cu un algoritm al celui mai scurt drum. Reţeaua $R_1 = (V_1, E_1, u_1)$ are $V_1 = \{0, n + 1\} \cup \{i_t | i_t \in V, d_2(0, i_t) + d_2(i_t, n + 1) \leq T\}$, $E_1 = \{(0, 1_t) | d_2(1_t, n + 1) \leq T, t \in H\} \cup \{(i_t, j_\theta) | (i_t, j_\theta) \in E, d_2(0, i_t) + h_2(i_t, j_\theta) + d_2(j_\theta, n + 1) \leq T\} \cup \{(n_t, n + 1) | d_2(0, n_t) \leq T, t \in H\}$ şi u_1 este restricţia lui u_2 la E_1.

Remarcăm faptul că reţeaua statică expandată redusă R_1 este întotdeauna o subreţea parţială a super reţelei statice expandate R_2. Un flux dinamic pentru T perioade în reţeaua dinamică D este echivalent cu un flux în reţeaua statică expandată redusă R_1. Este uşor de observat că reţelele statice R, R_2, R_1 nu pot conţine niciun circuit, şi, prin urmare, sunt aciclice.

În cazul general, parametrul $h\,(i) = 1$ este timpul de aşteptare în nodul i, iar parametrul $q(i; t)$ este funcţia margine superioară pentru fluxul $g(i; t)$ care poate aştepta la nodul i de la momentul t la $t + 1$.

Problema fluxului maxim pentru T perioade de timp în reţeaua dinamică D formulată în condiţiile (1.14), (1.15) şi (1.16) este echivalentă cu problema fluxului maxim în reţeaua statică expandată redusă R_1 după cum urmează:

$$\sum_{j_\theta} f_1(i_t, j_\theta) - \sum_{k_\tau} f_1(k_\tau, i_t) = \begin{cases} v_1(i_t), & \text{if } i_t = 1_t \\ 0, & \text{if } i_t \neq 1_t, n_t \\ -v_1(i_t), & \text{if } i_t = n_t \end{cases} \tag{1.17a}$$

$$0 \leq f_1(i_t, j_\theta) \leq u_1(i_t, j_\theta), \quad (i_t, j_\theta) \in E_0 \tag{1.17b}$$

$$\max v_1, \tag{1.17c}$$

cu $v_1 = \sum_t v_1(1_t)$.

În cazul $h(i, j; t) = h(i, j), t \in H$ distanţele dinamice $d(1, i; t), d(i, n; t)$ devin distanţe statice $d(1, i), d(i, n)$.

Există două abordări referitor la determinarea unui flux maxim în reţeaua dinamică $D = (N, A, h, e = 0, q, H)$: abordarea statică şi abordarea dinamică. Abordarea statică constă în determinarea unui flux maxim în reţeaua statică expandată redusă $R_1 = (V_1, E_1, u_1)$. Abordarea dinamică este utilizată în cazul staţionar. Nu este necesară construirea reţelei statice expandate reduse pentru rezolvarea problemei fluxului dinamic maxim pentru orice T. Fluxul dinamic maxim în cazul staţionar poate fi generat din fluxul f de valoare maximă şi timp minim în reţeaua statică $D = (N, A, h, e = 0, q, H)$, unde $h(i, j)$ este costul (timpul) pentru orice arc $(i, j) \in A$.

Algoritmul pentru problema fluxului dinamic maxim în cazul staţionar (SMDF) este prezentat mai jos.

Algoritmul 6: Algoritmul pentru determinarea fluxului dinamic maxim în cazul staţionar (SMDF)

```
   SMDF;
1  BEGIN
2     MVMCF (D, f);
3     DFEF (f, r(P_1), ..., r(P_k));
4     RF (r(P_1), ..., r(P_k));
5  END.
```

Fluxul obţinut cu algoritmul SMDF se numeşte flux repetat temporal pentru motivul evident că acesta constă în expedieri repetate de-a lungul aceluiaşi drum de la nodul 1 la nodul n. Valoarea fluxului maxim pentru fluxul repetat temporal obţinut cu algoritmul SMDF este:

$$v_H = (T + 1)v - \sum_A h(i, j) f(i, j) \tag{1.18}$$

unde v este valoarea maximă a fluxului obţinut cu MVMCF.

Teorema 1.14. *Algoritmul SMDF calculează corect fluxul dinamic maxim în reţeaua $D = (N, A, h, e = 0, q, H)$.*

Teorema 1.15. *Algoritmul SMDF are complexitatea $O(n^2 m^3 \log n)$.*

Definitia 1.23. *Fiind dat un flux admisibil g în reţeaua dinamică $D = (N, A, h, e, q, H)$, reţeaua reziduală dinamică în raport cu fluxul g este definită ca $D(g) = (N, A(g))$, cu:*

$$A(g) = A^+(g) \cup A^-(g)$$

unde

$$A^+(g) = \{(i,j)|(i,j) \in A,\ \exists t \leq T - h(i,j;t) cu\ q(i,j;t) - g(i,j;t) > 0\} \tag{1.19a}$$

$$A^-(g) = \{(i,j)|(j,i) \in A,\ \exists t \leq T - h(j,i;t) cu\ g(j,i;t) - e(j,i;t) > 0\} \tag{1.19b}$$

Oricare arc (i,j) al reţelei reziduale dinamice $D(g)$ care aparţine mulţimii arcelor directe $(i,j) \in A^+(g)$ are acelaşi timp de traversare $h(i,j;t)$ ca şi arcul corespunzător din reţeaua dinamică originară G. Oricare arc (i,j) al reţelei reziduale dinamice $D(g)$ care aparţine mulţimii arcelor artificiale inverse $(i,j) \in A^-(g)$ are următoarea caracteristică:

$$h(i,j;t) + h(j,i;t)) = -h(j,i;t), (j,i) \in A, 0 \leq t + h(j,i;t) \leq T, \quad g(j,i;t) > 0 \tag{1.20}$$

Definitia 1.24. *Capacitatea reziduală a unui arc (i,j) din reţeaua reziduală dinamică $D(g)$ este definită astfel:*

$$r(i,j;t) = u(i,j;t) - f(i,j;t), (i,j) \in A, 0 \leq t + h(i,j;t) \leq T \tag{1.21a}$$

$$r(i,j;t + h(j,i;t)) = f(j,i;t) - e(j,i;t), (j,i) \in A, 0 \leq t + h(j,i;t) \leq T \tag{1.21b}$$

Definitia 1.25. *Un drum $P((i_1;t),(i_{K+1};t+h(P;t))) = ((i_1;t),(i_2;t+h(i_1,i_2;t)),\ldots,(i_{K+1};\ t + h(P;t)))$ de la perechea nod-timp $(i_1;t)$ la perechea nod-timp $(i_{K+1};t+h(P;t))$ în reţeaua reziduală dinamică $D(g)$ este numit drum dinamic de mărire a fluxului dacă $r(i_k,i_{k+1};t_k) > 0$ cu $t_1 = t$ şi $t_{k+1} = t_k + h(i_k,i_{k+1};t_k), k = 1,\ldots,K$.*

Definitia 1.26. *Fiind dat un flux admisibil f în reţeaua dinamică $D = (N, A, h, e = 0, q, H)$, capacitatea reziduală a unui drum dinamic de mărire a fluxului $P((i_1;t),(i_{k+1};t + h(P;t)))$ este definită ca:*

$$r(P;t) = \min_{l \leq k \leq K} \{r(i_k,i_{k+1});t_k)|t_1 = t, t_{k+1} = t_k + h(i_k,i_{k+1};t_k)\} \tag{1.22}$$

Un drum dinamic $P((i_1;t),(i_{K+1};t\ +\ h(P;t))) = ((i_1;t),(i_2;t\ +\ h(i_1,i_2;t)),\ldots,(i_{K+1};t + h(P;t)))$ de mărire a fluxului este un circuit dinamic dacă $i_{K+1} = i_1$ şi $h(P;t) = 0$. Pentru detalii invităm cititorul să consulte lucrarea [8].

1.3.2 Fluxul minim în reţele dinamice oarecare

În lucrarea [18] se tratează o problemă specială a fluxului minim într-o reţea dinamică oarecare.

Problema fluxului dinamic minim pentru T perioade de timp constă în determinarea funcţiei $g : A \times H \to \mathbb{N}$, care satisface următoarele condiţii în reţeaua dinamică $D = (N, A, h, e, q)$:

$$\sum_{t=0}^{T}(g(1,N;t) - \sum_{\tau} g(N,1;\tau)) = \bar{w} \tag{1.23a}$$

$$g(i,N;t) - \sum_{\tau} g(N,i;\tau) = 0, i \neq 1,n,\ t \in H \tag{1.23b}$$

$$\sum_{t=0}^{T}(g(n,N;t) - \sum_{\tau} g(N,n;\tau)) = -\bar{w} \tag{1.23c}$$

$$e(i,j;t) \leq g(i,j;t) \leq q(i,j;t), \quad (i,j) \in A \quad , \quad t \in H \tag{1.24}$$

$$min \quad \bar{w}, \tag{1.25}$$

unde $\tau = t - h(k,i;t)$, $\bar{w} = \sum\limits_{t=0}^{T} v(t)$, $v(t)$ este valoarea fluxului la momentul t şi $g(i,j;t) = 0$ pentru toţi $t \in \{T - h(i,j;t) + 1, \ldots, T\}$.

În acest subparagraf considerăm că problema este staţionară.

$$h(i,j;t) = h(i,j), \quad e(i,j;t) = e(i,j), \quad q(i,j;t) = \infty, \quad (i,j) \in A, \quad t \in H \tag{1.26}$$

Remarcăm faptul că fluxul repetat temporal poate să nu fie optim pentru problema fluxului dinamic minim. Pentru mai multe detalii invităm cititorul să consulte lucrarea [18].

2. Fluxuri în reţele dinamice planare

2.1 Fluxuri în reţele dinamice planare. Abordarea statică.

2.1.1 Construcţia reţelei statice expandate reduse cu drumuri dinamice

Această construcţie aparţine autoarei acestei teze şi este conţinută în lucrarea [68].

Fie reţeaua dinamică $D = (N, A, h, e, q, H)$.

Problema fluxului dinamic maxim pentru T perioade de timp constă în determinarea funcţiei flux $g : A \times H \to \mathbb{N}$, care trebuie să satisfacă următoarele condiţii:

$$\sum_{t=0}^{T}(\sum_{j} g(1,j;t) - \sum_{k}\sum_{\tau} g(k,1;\tau)) = w \tag{2.1a}$$

$$\sum_{j} g(i,j;t) - \sum_{k}\sum_{\tau} g(k,i;\tau) = 0, i \neq 1, n,\ t \in H \tag{2.1b}$$

$$\sum_{t=0}^{T}(\sum_{j} g(n,j;t) - \sum_{k}\sum_{\tau} g(k,n;\tau)) = -w \tag{2.1c}$$

$$e(i,j;t) \leq g(i,j;t) \leq q(i,j;t), \quad (i,j) \in A \ , \ t \in H \tag{2.2}$$

$$max \quad w, \tag{2.3}$$

unde $\tau = t - h(k,i;t)$, $w = \sum_{t=0}^{T} v(t)$, $v(t)$ este valoarea fluxului la momentul t şi $g(i,j;t) = 0$ pentru toate momentele $t \in \{T - h(i,j;t) + 1, \ldots, T\}$.

Vom construi, reţeaua statică expandată redusă $R_1 = (V_1, E_1, l_1, u_1)$ utilizând noţiunea de cel mai scurt drum dinamic. Problema celui mai scurt drum dinamic este prezentată de Cai, Sha şi Wong [8]. Fie $d(1,i;t)$ lungimea celui mai scurt drum dinamic la momentul t de la nodul sursă 1 la nodul i şi $d(i,n;t)$ lungimea celui mai scurt drum la momentul t de la nodul i la nodul stoc n, referitor la h în reţeaua dinamică D. Vom considera $H_i = \{t | t \in H, d(1,i;t) \leq t \leq T - d(i,n;t)\}, i \in N$, şi $H_{i,j} = \{t | t \in H, d(1,i;t) \leq t \leq T - h(i,j;t) - d(j,n;\theta)\}, (i,j) \in A$. Reţeaua statică redusă cu surse

multiple şi stocuri multiple $R_0 = (V_0, E_0, l_0, u_0)$ are $V_0 = \{i_t | i \in N, t \in H_i\}$, $E_0 = \{(i_t, j_\theta) | (i,j) \in A, t \in H_{i,j}\}$, $l_0(i_t, j_\theta) = e(i,j;t)$, $u_0(i_t, j_\theta) = q(i,j;t), (i_t, j_\theta) \in E_0$. Reţeaua statică expandată redusă $R_1 = (V_1, E_1, l_1, u_1)$ este construită din reţeaua R_0 după cum urmează: $V_1 = V_0 \cup \{0, n+1\}, E_1 = E_0 \cup \{(0,1_t) | 1_t \in V_0\} \cup \{(n_t, n+1) | n_t \in V_0\}$, $l_1(0,1_t) = l_1(n_t, n+1) = 0$, $u_1(0,1_t) = u_1(n_t, n+1) = \infty$, $1_t, n_t \in V_0$ şi $l_1(i_t, j_\theta) = l_0(i_t, j_\theta)$ şi $u_1(i_t, j_\theta) = u_0(i_t, j_\theta)$, $(i_t, j_\theta) \in E_0$.

2.1.2 Fluxul maxim în reţele dinamice planare cu margine inferioară zero

Rezultatele din acest subparagraf aparţin autoarei acestei lucrări şi sunt conţinute în lucrarea [69], trimisă spre publicare.

Vom considera, în această secţiune că reţeaua dinamică $D = (N, A, h, e = 0, q, H)$ este $(1, n)$ planară. Vom construi reţeaua expandată redusă cu multiple noduri sursă şi multiple noduri stoc $R_0 = (V_0, E_0, u_0)$ conform descrierilor din subcapitolul 2.1.1.

Teorema 2.1. *Reţeaua expandată redusă statică cu multiple noduri sursă şi multiple noduri stoc $R_0 = (V_0, E_0, u_0)$ este o reţea planară statică.*

Observăm că R_0 este, de asemenea, o reţea planară statică cu multiple noduri sursă şi multiple noduri stoc când toate nodurile sursă şi toate nodurile stoc se găsesc pe aceeaşi frontieră a unei singure feţe sau cu multiple noduri sursă şi multimple noduri stoc când nodurile sursă şi nodurile stoc se află pe frontiera unor feţe diferite. În continuare vom considera primul caz, şi anume cazul în care toate nodurile sursă şi toate nodurile stoc se găsesc pe aceeaşi frontieră a unei singure feţe (faţa exterioară).

Prezentăm mai jos algoritmul pentru determinarea fluxului maxim în reţele planare orientate cu multiple noduri sursă şi multiple noduri stoc situate pe aceeaşi faţă (faţa exterioară).

Algoritmul 7: Algoritmul pentru determinarea fluxului maxim în reţele dinamice planare cu multiple noduri sursă şi multiple noduri stoc situate pe faţa exterioară.

```
   MFPDN;
1  BEGIN
2     CONS(D, R_0);
3     MFDPSNSSOF(R_0, f*);
4  END.
```

În continuare prezentăm corectitudinea şi complexitatea algoritmului MFPDN.

Teorema 2.2. *Algoritmul MFPDN calculează în mod corect fluxul maxim în reţeaua dinamică planară orientată D.*

Teorema 2.3. *Algoritmul MFPDN are complexitatea* $O(nTlog^{1.5}nT)$

Rezolvarea problemei fluxului maxim cu $e > 0$ şi a problemei fluxului minim prin abordarea statică constituie o temă de cercetare viitoare.

2.2 Fluxuri în reţele dinamice planare. Abordarea dinamică.

2.2.1 Fluxul maxim cu margine inferioară zero

Rezultatele din acest subparagraf aparţin parţial autoarei acestei lucrări şi sunt prezentate în articolul [71] publicat în numărul 11(2) din 2016 al revistei International Journal of Computers Comunications & Control.

În acest subparagraf considerăm reţeaua dinamică $D = (N, A, h, e = 0, q, H)$ în cazul staţionar. În acest caz folosim algoritmul SMDF prezentat în paragraful 1.3.

Reamintim că un ciclu (circuit) $\overset{\circ}{P}$ de cost mediu este $\sum (c(i,j)|(i,j) \in \overset{\circ}{P}) \,/|\overset{\circ}{P}|$, şi că ciclul mediu minim este ciclu cu cel mai mic cost mediu în reţeaua D ([2] etc.). Dacă se utilizează algoritmul programării dinamice, atunci algoritmul pentru determinarea ciclului mediu minim are complexitatea $O(nm)$ [2]. De asemenea, reamintim că algoritmul de eliminare a ciclului mediu minim este o variantă îmbunătăţită a algoritmului lui Klein de eliminare a circuitelor de cost negativ. Algoritmul eliminării ciclului mediu minim ca şi algoritmul Klein porneşte cu un flux maxim. Pentru informaţii suplimentare privind aceşti algoritmi, cititorul poate consulta lucrarea [2].

În cazul când reţeaua D este planară, un flux maxim se determină cu algoritmul MFPSN prezentat în paragraful 1.2.1. Pentru generarea unui flux dinamic maxim staţionar în reţeaua planară D se utilizează algoritmul SMDF prezentat în paragraful 1.3.1. Acest nou algoritm (SMDF modificat) îl denumim SMPDF (determină fluxul maxim cu MFPSN).

În continuare prezentăm teorema de corectitudine şi teorema de complexitate.

Teorema 2.4. *Algoritmul SMPDF calculează corect fluxul maxim în reţeaua dinamică planară staţionară.*

Teorema 2.5. *Algoritmul SMPDF aplicat în reţeaua dinamică planară staţionară are complexitatea* $O(n^5 logn)$.

2.2.2 Fluxul maxim în reţele dinamice planare cu margine inferioară pozitivă

Rezultatele din acest subparagraf aparţin parţial autoarei acestei lucrări şi sunt prezentate în lucrarea [73].

În acest subparagraf considerăm problema fluxului maxim în reţele dinamice planare staţionare cu margine inferioară pozitivă.

Există două inconveniente pentru această problemă. Primul constă în faptul că deşi în reţeaua planară statică $(1,n)$ $D=(N,A,c=h,l,u)$ există un flux admisibil, este posibil ca în reţeaua expandată redusă $R_0=(V_0,E_0,l_0,u_0)$ să nu existe nici un flux admisibil. Al doilea inconvenient constă în faptul că este posibil ca fluxul repetat în timp al fluxului admisibil de timp minim în reţeaua D, să nu fie un flux admisibil în reţeaua R_0 (sau R_1), deşi în R_0 există un flux admisibil.

Presupunem că fluxul $f=\mathring{f}$, determinat cu algoritmul mFPSN prezentat în subparagraful 1.2.2 în reţeaua statică $D=(N,A,c=h,l,u)$ este admisibil. Fie P_s un drum determinat cu procedura DFEF din algoritmul SMDF prezentat în subparagraful 1.3.1 cu $r(P_s)$ fluxul, $h(P_s)$ timpul de tranzit şi $\nu(P_s)=(T+1)-h(P_s)$ numărul de repetiţii ale drumului P_s, $s=1,\ldots,k$. Considerăm că $h(P_1)\leq\ldots\leq h(P_k)$. Dacă drumul P_s este $P_s=((1,x),\ldots,(y,i),(i,j),\ldots,(z,n))$, atunci definim $h_i(P_s)=h(1,x)+\ldots+h(y,i)$ şi $\bar{h}_i(P_s)=h(P_s)-h_i(P_s)$. În mod evident că $h_n(P_s)=h(P_s)$. Fie $H_i, i\in N$ mulţimea $H_i=\{t|t\in H, d(1,i;t)\leq t\leq T-d(i,n;t)\}$ cu precizarea că reţeaua dinamică D este staţionară. Definim mulţimile $H_i^s=\{t,|t\in H_i, h_i(P_s)\leq t\leq T-\bar{h}_i(P_s)\}=\{h_i(P_s),h_i(P_s)+1,\ldots,T-\bar{h}_i(P_s)\}$ cu $|H_i^s|=\nu(P_s), i\in P_s$ şi $H_i^s=\emptyset$ pentru $i\notin P_s, s=1,\ldots,k$. În mod corespunzător drumurile în reţeaua R_0 sunt $P_s^t=(1_t,\ldots,i_{\gamma+t},\ldots,n_{\eta+t}), t=0,\ldots,k_s$, $k_s=\nu(P_s)-1$,$\gamma=h_i(P_s)$, $\eta=h_n(P_s)=h(P_s)$, $s=1,\ldots,k$. Dacă reţeaua dinamică este staţionară, atunci avem $d(1,i;t)=d(1,i)$, $d(i,n;t)=d(i,n)$ pentru toate nodurile $i\in N$ şi sunt calculate cu un algoritm de determinare a celui mai scurt drum.

Considerăm că arcele din A şi din E_0 sunt aranjate într-o anumită ordine. Fie $e=(e(i,j)), q=(q(i,j)), g=(g(i,j;t))$ şi $l_0=(l_0(i_t,j_\theta)), u_0=(u_0(i_t,j_\theta)), f_0=(f_0(i_t,j_\theta))$, vectorul margine inferioară, vectorul margine superioară şi vectorul flux în care aceste componente sunt ordonate în aceeaşi ordine cu ordinea arcelor din mulţimile A şi E_0. Generăm fluxul g dintr-un flux static în reţea statică $D=(N,A,c=h,e,q)$ şi nu construim reţeaua statică R_0, dar ne referim adesea la existenţa sa.

Definim lista $E_0^{'} = (a_1^{'}, a_2^{'}, \ldots, a_\alpha^{'})$ cu proprietatea că $a_i^{'} \in E_0$, $f_0(a_i^{'}) < l_0(a_i^{'})$, $i = 1, 2, \ldots, \alpha$, unde f_0 este fluxul repetat temporal în reţeaua R_0 generat de fluxul admisibil şi de timp minim f^*. Fluxul f_0 este determinat de algoritmul SMDF prezentat în subparagraful 1.3.1 cu modificarea că procedura MVMCF foloseşte algoritmul mFPSN prezentat în subparagraful 1.2.2 care determină un flux admisibil $f = \mathring{f}$ şi algoritmul de anulare a ciclului de cost (timp) mediu minim pentru determinarea unui flux de timp minim admisibil f^* în reţeaua statică $D = (N, A, c = h, e, q)$. Fie $k_i^{'}$ numărul drumurilor P_s^t care conţin arce $a_i^{'}$, $i = 1, 2, \ldots, \alpha$. Dacă există arcul $a_j^{'} \in E_0^{'}$ şi $a_j^{'} \notin P_s^t$, $s = 1, 2, \ldots, k$, $t = 0, \ldots, k_s$, atunci putem determina uşor drumul P_{k+1}^t care conţine arcul $a_j^{'}$. Selectăm arcele $a_i^{'}$ din $E_0^{'}$ în ordinea crescătoare a numerelor $k_i^{'}$. Fie $\mathcal{P}_0$, $\mathcal{P}_0 = \{P_s^t | (i_t, j_\theta) \in P_s^t\}, (i_t, j_\theta) \in E_0^{'}$.

Dacă $E_0^{'} = \emptyset$, f_0 este un flux admisibil în reţeaua R_0, altfel determinăm un flux admisibil $\mathring{f}_0$ în reţeaua R_0 cu procedura FFR_0 prezentată în algoritmul 8.

Algoritmul 8: Algoritmul pentru determinarea unui flux admisibil în reţeaua R_0.

$FFR_0(l_0, u_0, f_0, E_0^{'}, \mathcal{P}_0, \mathring{f}_0, k_1^{'}, \ldots, k_\alpha^{'})$;
BEGIN
  $\mathring{f}_0 := f_0; \beta := 1$;
  repeat
    selectează un arc $a_i^{'}$ din $E_0^{'}$ cu $k_i^{'}$ minim;
    selectează drumul P_s^t din $\mathcal{P}_0$ cu $a_i^{'} \in P_s^t$;
    fie $\hat{P}_s^t := P_s^t - \{a_i^{'}\}$;
    $r(\hat{P}_s^t) := min\{u_0(i_t, j_\theta) - f_0(i_t, j_\theta) | (i_t, j_\theta) \in \hat{P}_s^t\}$;
    if $l_0(a_i^{'}) - f_0(a_i^{'}) \leq r(\hat{P}_s^t)$
      BEGIN
        $r(P_s^t) := l_0(a_i^{'}) - f_0(a_i^{'})$;
        creşte $\mathring{f}_0$ cu $r(P_s^t)$ de-a lungul drumului P_s^t;
        elimină din $E_0^{'}$ arcele $a_i^{'}$ cu $l_0(a_i^{'}) \leq \mathring{f}_0(a_i^{'})$;
      END
    else $\beta := 0$;
    if $\beta = 0$;
      Exit;
  until $E_0^{'} = \emptyset$;
  if $\beta = 1$;
    then $\mathring{f}_0$ este un flux admisibil în R_0
    else nu există flux admisibil în R_0;
END.

Teorema 2.6. *Dacă $E_0^{'} \neq \emptyset$ şi există un flux admisibil în reţeaua dinamică planară orientată $(1, n)$ $D = (N, A, h, e, q)$ atunci procedura*

FFR_0 determină un flux admisibil în reţeaua D, altfel procedura specifică faptul că nu există un flux admisibil în D.

Teorema 2.7. *Procedura FFR_0 are complexitatea $O(n^2T^2)$.*

Prezentăm pentru această problemă un algoritm pentru fluxul admisibil maxim de timp minim $\overset{*}{f}{}^*$ în reţeaua statică $D = (N, A, c = h, e, q)$ care generează un flux admisibil maxim $\overset{*}{f}_0$ în reţeaua statică R_0. Algoritmul pentru fluxul admisibil de timp minim $\overset{*}{f}{}^*$ în reţeaua statică $D = (N, A, c = h, e, q)$ este în esenţă acelaşi ca algoritmul lui Ahuja-Orlin al reţelelor stratificate pentru problema fluxului maxim ([2]). Facem următoarele modificări la algoritmul prezentat de Ahuja-Orlin al reţelelor stratificate: etichetele distanţă $d(i)$ sunt distanţele de la nodul 1 cu privire la cost $c = h$, care pot fi rezolvate cu un algoritm clasic. Remarcăm că acest algoritm este o variantă a algoritmului de drumuri minime succesive (Ahuja, Magnanti şi Orlin [2]). Algoritmul de generare a unui flux admisibil maxim $\overset{*}{f}_0$ în reţeaua statică R_0 este o variantă a algoritmului Wilkinson prezentat în lucrarea Wilkinson [79]. Varianta algoritmului Wilkinson (VWA) este prezentată în algoritmul 9.

Algoritmul 9: Varianta algoritmului Wilkinson.

```
    VWA;
 1  BEGIN
 2    f*^* := f^*; f*_0 := f_0;
 3    construim reţeaua reziduală D̃ cu privire la f*^*;
 4    obţinem etichetele distanţă exacte d(i) în D̃;
 5    for i ∈ N do
 6      b(i) :=true;
 7    i := 1;
 8    while d(n) ≤ T do
 9      if b(1) = true
10        then if există un arc admisibil (i, j)
11               then BEGIN
12                      ÎNAINTARE (i);
13                      if i = n
14                      then BEGIN
15                             MĂRIRE;
16                             i := 1;
17                             END;
18                      END
19             else ÎNAPOIERE(i)
20        else BEGIN
21             determină etichetele distanţă exactă d(i) în D̃;
22              for i ∈ N do
23                b(i) :=true;
24              i := 1;
```

```
25            END;
26 END.

   PROCEDURE ÎNAINTARE(i);
1  BEGIN
2     p̃(j) := i;
3     i := j;
4  END;

   PROCEDURE ÎNAPOIERE(i);
1  BEGIN
2     b(i) := false;
3     if i ≠ 1
4       then i := p̃(i);
5  END;

   PROCEDURE MĂRIRE;
1  BEGIN
2     identifcă un drum de mărire P̃ folosind lista predecesorilor p̃;
3     r(P̃) := min{r(i, j)|(i, j) ∈ Ã};
4     actualizează reţeaua reziduală D̃;
5     identifică lanţurile C_0^t din R_0 corespunzătoare drumului P̃;
6     determină r(C_0^t) pentru toţi t;
7     măreşte fluxul f*_0 cu r(C_0^t) de-a lungul lanţului C_0^t pentru toţi t;
8  END;
```

Reţeaua $\tilde{D}$ este reţeaua reziduală a reţelei statice $D = (N, A, c = h, e, q)$. Amintim că un arc (i, j) în reţeaua reziduală $\tilde{D}$ este admisibil dacă satisface condiţiile că $d(j) = d(i) + 1$ şi nodul j nu este blocat ($b(j) =$ true) cu $d(i)$, $d(j)$ distanţele privind costul $c = h$. Lanţurile C_0^t în reţeaua statică R_0 sunt identificate în acelaşi mod ca şi drumurile P_s^t corespunzătoare drumului P_s care este prezentat mai sus.

Teorema 2.8. *Algoritmul VWA determină un flux maxim admisibil în reţeaua dinamică planară* $(1, n)$ $D = (N, A, h, e, q, H)$.

Teorema 2.9. *Algoritmul VWA are complexitatea* $O(n^2T^2\bar{q})$, *unde* $\bar{q} = max\{q(i, j)|(i, j) \in A\}$.

Prezentăm, mai jos, algoritmul complet pentru fluxul admisibil maxim în reţeaua dinamică planară $(1, n)$ (CAMFFPDN).

Algoritmul 10: Algoritmul complet pentru fluxul admisibil maxim în reţeaua dinamică planară $(1, n)$

```
   CAMFFPDN;
1  BEGIN
2     SMDF(D, f°, f°*, f_0);
3     if f_0 nu este admisibil
4       then FFR_0(R_0, f_0, E'_0, P_0, f°_0, k'_1, ..., k'_α)
```

4 VWA($D, \overset{\circ}{f}{}^*, R_0, f_0, \overset{*}{f}{}^*, \overset{*}{f}_0$);
5 END.

Teorema 2.10. *Algoritmul CAMFFPDN determină un flux admisibil maxim în reţeaua dinamică planară* $(1, n)$ $D = (N, A, h, e, q, H)$.

Teorema 2.11. *Algoritmul CAMFFPDN are complexitatea* $O(n^2T^2\bar{q})$.

2.2.3 Fluxul minim în reţele dinamice planare

Rezultatele din acest subparagraf aparţin parţial autoarei acestei lucrări şi sunt prezentate în lucrarea [33] în curs de publicare.

Considerăm problema fluxului minim în reţeaua dinamică staţionară $(1, n)$ $D = (N, A, h,\ e, q, H)$.

Remarcăm faptul că dacă fluxul minim $f = \overset{\circ}{f}$ obţinut cu algoritmul mFPSN prezentat în subparagraful 1.2.2 este inadmisibil $(u(i, j) < \overset{\circ}{f}(i, j))$ pentru unul sau mai multe arce $((i, j) \in A)$, atunci nu există flux admisibil în reţeaua D.

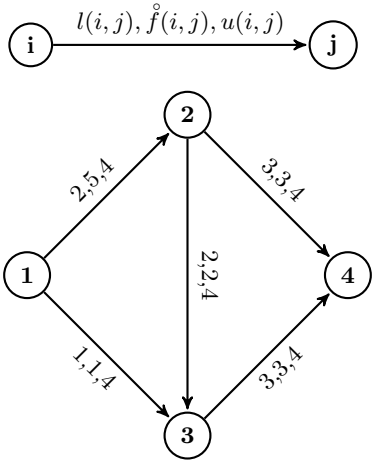

Figura 2.1: Fluxul minim $\overset{\circ}{f}$ nu este admisibil.

De exemplu, fluxul minim $f = \overset{\circ}{f}$ obţinut cu algoritmul mFPSN pentru reţeaua din Figura 2.1 este un flux indamisibil deoarece $u(1, 2) = 4 < 5 = \overset{\circ}{f}(1, 2)$. Dacă considerăm $X = \{1, 3, 4\}$, $\bar{X} = \{2\}$ avem $u(X, \bar{X}) = 4 < 5 = l(X, \bar{X})$ şi din teorema fluxului admisibil prezentată în lucrarea Ahuja, Magnanti şi Orlin [2] rezultă că în reţeaua D nu există un flux admisibil.

Precizăm că fluxul admisibil $\overset{\circ}{f}_0$ determinat cu procedura FFR_0 este un flux admisibil minim în reţeaua R_0, deoarece fluxul $\overset{\circ}{f}_0$ în R_0 este generat de fluxul admisibil minim de timp minim $\overset{\circ}{f}^*$ în reţeaua statică $D = (N, A, c = h, l = e, u = q)$ şi de metoda prin care este calculat $\overset{\circ}{f}_0$ în procedura FFR_0. Fluxul admisibil minim în reţeaua R_0 este echivalent cu un flux admisibil în reţeaua dinamică $D = (N, A, h, e, q)$.

Să considerăm teoretic, că pot exista anumite reţele dinamice planare de dimensiuni foarte mari în care $\overset{\circ}{f}_0$ determinat cu procedura FFR_0 nu este minim. În acest caz se foloseşte algoritmul de mai jos.

Algoritmul $10'$**:** Algoritmul complet pentru fluxul admisibil minim în reţeaua dinamică planară $(1, n)$

```
   CAmFFPDN;
1  BEGIN
2    SMDF(D, f°, f*, f_0);
3    if f_0 nu este admisibil
4      then FFR_0(R_0, f_0, E_0', P_0, f°_0, k_1', ..., k_α')
5      if f_0 nu este un flux minim
6        then VWAm(D, f*, R_0, f*, f*_0);
7  END.
```

Procedurile SMDF şi FFR_0 sunt procedurile precizate în algoritmul 10. Dacă fluxul admisibil f_0 nu este un flux minim atunci se aplică procedura VWAm (varianta algoritmului Wilkinson pentru fluxul minim) care determină un flux admisibil minim $\overset{\circ}{f}_0$ pornind de la fluxul admisibil iniţial f_0.

Teorema 2.12. *Algoritmul CAmFFPDN determină un flux admisibil minim în reţeaua dinamică planară* $(1, n)$ $d = (N, A, h, e, q, H)$

Până în prezent, autoarea lucrării, nu a găsit un exemplu în care $\overset{\circ}{f}_0$ determinat cu procedura FFR_0 să nu fie flux minim.

3. Fluxuri în reţele dinamice bipartite

3.1 Fluxuri în reţele dinamice bipartite. Abordarea statică

3.1.1 Fluxuri maxime în reţele dinamice bipartite cu marginea inferioară zero

Rezultatele din acest subparagraf aparţin autoarei acestei teze şi sunt publicate în lucrarea [65].

Considerăm că reţeaua dinamică $D = (N, A, h, e = 0, q, H)$ este bipartită.

Construim reţeaua expandată redusă statică $R_0 = (V_0, E_0, l_0, u_0)$.
Teorema 3.1. *Dacă reţeaua dinamică $D = (N, A, h, e = 0, q, H)$ este bipartită, atunci reţeaua statică expandată redusă $R_0 = (V_0, E_0, u_0)$ este bipartită.*

Fie w_1, w_2, ε_0 cu $w_1 = |W_1|$, $w_2 = |W_2|$, $\varepsilon_0 = |E_0|$. Dacă $n_1 << n_2$ este evident că $w_1 << w_2$. În reţeaua statică bipartită R_0 determinăm un flux maxim f_0 cu o generalizare a algoritmului preflux FIFO bipartit prezentat în subparagraful 1.2.3 (algoritmul 4).

Reamintim faptul că algoritmul preflux FIFO ar putea efectua mai multe înaintări saturate urmate, fie de o înaintare saturată, fie de o operaţie de reetichetare. Ne referim la această secvenţă de operaţii ca la o examinare a nodurilor. Algoritmul analizează nodurile active în ordinea FIFO. Algoritmul menţine lista Q a nodurilor active într-o coadă. Prin urmare, algoritmul selectează un nod i de la începutul listei Q, execută înaintările pentru acest nod, şi adaugă noi noduri active la sfârşitul cozii Q. Algoritmul analizează nodul i până când acesta devine inactiv sau este reetichetat. În ultimul caz, adăugăm nodul i la finalul cozii Q. Algoritmul se termină când coada Q a nodurilor active este goală.

Versiunea generalizată a algoritmului preflux FIFO pentru fluxul maxim în reţele bipartite este numită algoritmul preflux FIFO bipartit generalizat. O biînaintare este o înaintare peste două arce admisibile consecutive. Se trimite excesul de la un nod $i_t \in W_1$ la alt nod $k_\tau \in W_1$. Această abordare se referă la faptul că algoritmul trimite fluxul pe drumul $\tilde{D} = (i_t, j_\theta, k_\tau), j_\theta \in W_2$, şi se asigură că niciun nod din W_2 nu

a avut vreodată exces. O înaintare de α unităţi de la nodul i_t la nodul j_θ scade atât $e(i_t)$, cât şi $r_0(i_t, j_\theta)$ cu α unităţi şi creşte atât $e(j_\theta)$, cât şi $r_0(j_\theta, i_t)$ cu α unităţi, unde $\alpha = \min\{e(i_t), r_0(i_t, j_\theta), r_0(j_\theta, i_t)\}$.

Precizăm că menţinem lista arcelor $E_0^+(i_t) = \{(i_t, j_\theta)|(i_t, j_\theta) \in E_0\}$. Putem aranja arcele în aceste liste arbitrar, dar ordinea decisă rămâne neschimbată pe tot parcursul algoritmului. Fiecare nod i_t are un arc curent, care este un arc în $E_0^+(i_t)$ şi este următorul arc pentru testarea admisibilităţii. Iniţial, arcul curent al nodului i_t este primul arc în $E_0^+(i_t)$. De fiecare dată când algoritmul încearcă să găsească un arc admisibil de la nodul i_t, se testează dacă arcul nodului curent este admisibil. Dacă nu, acesta desemnează următorul arc din lista de arce curente. Algoritmul repetă această procedură până când fie găseşte un arc admisibil, fie se ajunge la sfârşitul listei arcelor.

Algoritmul preflux FIFO bipartit generalizat (GBFIFOP1) este prezentat mai jos.

Algoritmul 11: Algoritmul preflux FIFO bipartit generalizat (GBFIFOP1)

```
   GBFIFOP1;
1  BEGIN
2    INIŢIALIZARE;
3    while Q ≠ ∅ do
4      BEGIN
5        selectează nodul i_t de la începutul listei Q;
6        BIÎNAINTARE/REETICHETARE(i_t);
7      END;
8  END.

   PROCEDURA INIŢIALIZARE;
1  BEGIN
2    f =: 0; Q := ∅;
3    calculează etichetele distanţă exacte d(i_t);
4    for t ∈ H_1 do
5      BEGIN
6        f_0(1_t, j_θ) := u_0(1_t, j_θ) şi adaugă nodul j_θ la sfârşitul listei Q
         pentru toate arcele (1_t, j_θ) ∈ E_0;
7        d(1_t) := 2w_2 + 1;
8      END
9  END;

   PROCEDURA BIÎNAINTARE/REETICHETARE(i_t);
1  BEGIN
2    selectează primul arc (i_t, j_θ) din E_0^+(i_t) cu r_0(i_t, j_θ) > 0;
3    β := 1;
4    repeat
5      if (i_t, j_θ) este arc admisibil
```

```
6           if (j_θ, k_τ) este arc admisibil
7              BEGIN
8                 înaintează α := min{e(i_t), r_0(i_t, j_θ), r_0(j_θ, k_τ)}
                  unităţi ale fluxului peste arcele (i_t, j_θ), (j_θ, k_τ);
9                 if k_τ ∉ Q
10                   adaugă nodul k_τ la sfârşitul listei Q;
11             END
12          else if (j_θ, k_τ) nu este ultimul arc din E_0^+(j_θ) cu
                    r_0(j_θ, k_τ) > 0
13                  selectează următorul arc din E_0^+(j_θ)
14             else d(j_θ) := min{d(k_τ) + 1|(j_θ, k_τ) ∈ E_0^+(j_θ),
                    r_0(j_θ, k_τ) > 0} ;
15          if e(i_t) > 0
16             if (i_t, j_θ) nu este ultimul arc din E_0^+(j_θ) cu r_0(i_t, j_θ) > 0
17                selectează următorul arc din E_0^+(i_t)
18             else
19                BEGIN
20                   d(i_t) := min{d(j_θ) + 1|(i_t, j_θ) ∈ E_0^+(j_θ),
                     r_0(i_t, j_θ) > 0};
21                   β := 0;
22                END
23       END
24   until e(i_t) = 0 sau β = 0
25   if e(i_t) > 0
26      adaugă nodul i_t la sfârşitul listei Q;
27 END;
```

Precizăm că orice drum în reţeaua reziduală $\tilde{R_0} = (V_0, \tilde{E_0}, r_0)$ poate avea cel mult $2w_2 + 1$ arce. Prin urmare, am stabilit $d(1_t) := 2w_2 + 1$ în procedura INIŢIALIZARE.

Corectitudinea algoritmului GBFIFOP1 rezultă din corectitudinea algoritmului pentru flux maxim în reţeaua bipartită (Ahuja, Orlin, Stein şi Tarjan [1]).

Teorema 3.2. *Algoritmul GBFIFOP1 care determină un flux maxim într-o reţea dinamică bipartită $D = (N, A, h, e = 0, q)$ are complexitatea $O(n_1 m T^2 + n_1^3 T^3)$.*

3.1.2 Fluxuri maxime în reţele dinamice bipartite cu margine inferioară pozitivă

Rezultatele din acest subparagraf sunt originale şi sunt publicate în lucrările [70] şi [72].

În aceast subparagraf reţeaua dinamică $D = (N, A, h, e, q, H)$ este bipartită.

Construim reţeaua statică expandată redusă $R_0 = (V_0, E_0, l_0, u_0)$ şi reamintim faptul că reţeaua R_0 este o reţea bipartită cu $V_0 = W_1 \cup W_2$,

$W_1 = \{i_t | i \in N_1, t \in H_i\}$, $W_2 = \{i_t | i \in N_2, t \in H_i\}$. Fie w_1, w_2, ε_0, $w_1 = |W_1|$, $w_2 = |W_2|$, $\varepsilon_0 = |E_0|$. Dacă $n_1 << n_2$ atunci evident că $w_1 << w_2$. În reţeaua statică bipartită R_0 determinăm un flux maxim f_0 cu o generalizare a algoritmului preflux bipartit FIFO.

Generalizăm algoritmul GBFIFOP1 prezentat în subparagraful 3.1.1 pentru cazul $l_0 > 0$. Există mai multe noduri sursă $1_t, t \in H_1$ şi mai multe noduri stoc $n_t, t \in H_n$. De asemenea, prezentăm în detaliu pseudocodul.

Algoritmul GBFIFOP1 bipartit generalizat (GBFIFOP2) este prezentat mai jos:

Algoritmul 12: Algoritmul preflux FIFO generalizat bipartit (GBFIFOP2)

```
   GBFIFOP2;
1  BEGIN
2    INIŢIALIZARE;
3    while Q ≠ ∅ do
4      BEGIN
5        selectează nodul i_t de la începutul listei Q;
6        BIÎNAINTARE/REETICHETARE(i_t);
7      END;
8  END.

   PROCEDURA INIŢIALIZARE;
1  BEGIN
2    f_0 este un flux admisibil în R_0; Q := ∅;;
3    calculează etichetele distanţă exacte d(i_t);
4    for t ∈ H_1 do
5      BEGIN
6        f_0(1_t, j_θ) := u_0(1_t, j_θ) şi adaugă nodul j_θ la sfârşitul listei Q
         pentru toate arcele (1_t, j_θ) ∈ E_0;
7        d(1_t) := 2w_2 + 1;
8      END
9  END;

   PROCEDURA BIÎNAINTARE/REETICHETARE(i_t);
1  BEGIN
2    selectează primul arc (i_t, j_θ) din E_0^+(i_t) cu r_0(i_t, j_θ) > 0;
3    β := 1;
4    repeat
5      if (i_t, j_θ) este arc admisibil
6        BEGIN
7          selectează primul arc (j_θ, k_τ) din E_0^+(j_θ) cu r_0(j_θ, k_τ) > 0;
8          if (j_θ, k_τ) este arc admisibil
9            BEGIN
10             înaintează α := min{e(i_t), r_0(i_t, j_θ), r_0(j_θ, k_τ)}
               unităţi ale fluxului peste arcele (i_t, j_θ), (j_θ, k_τ);
```

```
11                  if k_τ ∉ Q
12                     adaugă nodul k_τ la sfârşitul listei Q;
13                  END
14               else if (j_θ, k_τ) nu este ultimul arc din E_0^+(j_θ) cu
                        r_0(j_θ, k_τ) > 0
15                     selectează următorul arc din E_0^+(j_θ)
16                  else d(j_θ) := min{d(k_τ) + 1|(j_θ, k_τ) ∈ E_0^+(j_θ),
                        r_0(j_θ, k_τ) > 0};
17               if e(i_t) > 0
18                  if (i_t, j_θ) nu este ultimul arc din E_0^+(j_θ) cu r_0(i_t, j_θ) > 0
19                     selectează următorul arc din E_0^+(i_t)
20                  else
21                     BEGIN
22                        d(i_t) := min{d(j_θ) + 1|(i_t, j_θ) ∈ E_0^+(j_θ),
                          r_0(i_t, j_θ) > 0};
23                        β := 0;
24                     END
25            END
26      until e(i_t) = 0 sau β = 0
27      if e(i_t) > 0
28         adaugă nodul i_t la sfârşitul listei Q;
29   END;
```

Precizăm faptul că în prima fază se determină un flux admisibil în R_0 cu algoritmul GBFIFOP1 modificând problema conform cu explicaţiile date în subparagraful 1.1.2 (a se vedea şi [2]).

Remarcăm faptul că orice drum în reţeaua reziduală $\tilde{R}_0 = (V_0, \tilde{E}_0, r_0)$ poate avea cel mult $2w_1 + 1$ arce. De aceea am stabilit $d(1_t) := 2w_1 + 1$ în procedura INIŢIALIZARE.

Corectitudinea algoritmului GBFIFOP2 rezultă din corectitudinea algoritmului pentru fluxul maxim în reţele bipartite Ahuja, Orlin, Stein, Tarjan [1].

Teorema 3.3. *Algoritmul GBFIFOP2 care determină un flux maxim într-o reţea dinamică* $D = (N, A, h, e, q, H)$ *are complexitatea* $O(n_1 m T^2 + n_1^3 T^3)$.

3.1.3 Fluxuri minime în reţele dinamice bipartite

Rezultatele acestui subparagraf aparţin parţial autoarei acestei teze şi sunt conţinute în lucrarea [32] trimisă spre publicare.

În acest subparagraf reţeaua dinamică $D = (N, A, h, e, q, H)$ este bipartită.

Construim reţeaua expandată redusă statică $R_0 = (V_0, E_0, l_0, u_0)$ cu $V_0 = W_1 \cup W_2$, $W_1 = \{i_t | i \in N_1, t \in H_i\}$, $W_2 = \{i_t | i \in N_2, t \in H_i\}$.

Conform Teoremei 3.1 rezultă că dacă reţeaua dinamică $D = (N, A, h, e, q, H)$ este bipartită, atunci reţeaua expandată redusă statică $R_0 = (V_0, E_0, l_0, u_0)$ este bipartită.

Fie w_1, w_2, ε_0 cu $w_1 = |W_1|$, $w_2 = |W_2|$, $\varepsilon_0 = |E_0|$. Dacă $n_2 << n_1$ atunci observăm că $w_2 << w_1$. În reţeaua statică bipartită R_0 determinăm un flux minim f_0 cu algoritmul preflux FIFO bipartit generalizat pentru fluxul minim.

Algoritmul (GBFIFOP3) preflux FIFO bipartit generalizat pentru fluxul minim este prezentat mai jos:

Algoritmul 13: Algoritmul preflux FIFO bipartit generalizat pentru fluxul minim (GBFIFOP3)

```
   GBFIFOP3;
1  BEGIN
2    INIŢIALIZARE;
3    while Q ≠ ∅ do
4      BEGIN
5        selectează nodul j_θ de la începutul listei Q;
6        BIÎNAPOIERE/REETICHETARE(j_θ);
7      END;
8  END.

   PROCEDURE INIŢIALIZARE;
1  BEGIN
2    f_0 este un flux admisibil în R_0; Q := ∅;
3    calculează funcţia distanţă exactă d̂ în reţeaua reziduală Ĝ;
4    for λ ∈ H_n do
5      BEGIN
6        f_0(j_θ, n_λ) := l_0(j_θ, n_λ) şi adaugă nodul j_θ la sfârşitul listei Q
         pentru toate arcele (j_θ, n_λ) ∈ E_0;
7        d̂(n_λ) := 2w_2 + 1;
8      END;
9  END;

   PROCEDURA BIÎNAPOIERE/REETICHETARE(j_θ);
1  BEGIN
2    selectează primul arc (i_t, j_θ) din E_0^-(j_θ) cu r̂_0(i_t, j_θ) > 0;
3    β := 1;
4    repeat
5      if (i_t, j_θ) este arc admisibil
6        BEGIN
7          selectează primul arc (k_τ, i_t) din E_0^-(i_t) cu r̂_0(k_τ, i_t) > 0;
8          if (k_τ, i_t) este arc admisibil
9            BEGIN
10             înapoiază α := min {−e(j_θ), r̂_0(i_t, j_θ), r̂_0(k_τ, i_t)}
               unităţi de flux peste arcele (i_t, j_θ), (k_τ, i_t);
11           END
```

```
12              if k_τ ∉ Q
13                 adaugă nodul k_τ la sfârşitul listei Q;
14              else if (k_τ, i_t) nu este ultimul arc din E_0^-(i_t) cu
                        r̂_0(k_τ, i_t) > 0
15                      selectează următorul arc din E_0^-(i_t)
16                   else d̂(i_t) := min{d(k_τ) + 1|(k_τ, i_t) ∈ E_0^-(i_t),
                r̂_0(k_τ, i_t) > 0};
17                      if e(j_θ) > 0
18              if (i_t, j_θ) nu este ultimul arc din E_0^-(j_θ) cu r̂_0(i_t, j_θ) > 0
19                 selectează următorul arc din E_0^-(j_θ)
20              else
21                 BEGIN
22                    d̂(j_θ) := min{d(i_t) + 1|(i_t, j_θ) ∈ E_0^-(j_θ),
                      r̂_0(i_t, j_θ) > 0};
23                    β := 0;
24                 END
25    until e(j_θ) = 0 sau β = 0
26    if e(j_θ) > 0
27       adaugă nodul j_θ la sfârşitul listei Q;
28 END;
```

Remarcăm faptul că orice drum în reţeaua reziduală $\tilde{R}_0 = (V_0, \tilde{E}_0, r_0)$ poate avea cel mult $2w_2+1$ arce. Prin urmare, stabilim $\hat{d}(n_\lambda) := 2w_2+1$ în procedura INIŢIALIZARE.

Corectitudinea algoritmului GBFIFOP3 rezultă din corectitudinea algoritmului pentru flux minim în reţele bipartite (Ciurea, Georgescu şi Marinescu [28]).

Teorema 3.4. *Algoritmul GBFIFOP3 care determină un flux minim în reţeaua dinamică* $D = (N, A, h, e, q, H)$ *are complexitatea* $O(n_2 m T^2 + n_2^3 T^3)$.

3.2 Fluxuri în reţele dinamice bipartite. Abordarea dinamică

3.2.1 Fluxul maxim în reţele dinamice bipartite cu marginea inferioară zero

Rezultatele din acest subparagraf aparţin autoarei şi sunt publicate în lucrarea [68].

În acest subparagraf considerăm fluxul maxim în reţele dinamice bipartite în cazul staţionar $h(i,j;t) = h(i,j)$, $q(i,j;t) = q(i,j)$, $(i,j) \in A, t \in H$. Folosim algoritmul SMDF care a fost prezentat în subparagraful 1.3.1. În acest subparagraf reţeaua dinamică $D = (N, A, h, e = 0, q, H)$ este bipartită.

Considerăm reţeaua statică bipartită $G = (N, A, c = h, l = 0, u)$ unde

$c(i,j) = h(i,j)$, $u(i,j) = q(i,j), (i,j) \in A$. Procedura MVMCF din algoritmul SMDF efectuează algoritmul pentru fluxul de cost minim şi valoare maximă f^* în reţeaua statică bipartită. Versiunea modificată a algoritmului de scalare a costului pentru reţeaua statică bipartită G începe cu orice flux admisibil. În acest caz fluxul admisibil este un flux maxim f. Determinăm un flux f de valoare maximă cu versiunea modificată a algoritmului preflux FIFO care are complexitatea $O(n_1 m + n_1^3)$. Versiunea modificată a algoritmului de scalare a costului are complexitatea $O(n_1 m + n_1^3 \log(n_1 \bar{c})) = O(n_1 m + n_1^3 \log(n_1 \bar{h}))$, $\bar{h} = max\{h(i,j) | (i,j) \in A\}$.

Teorema 3.5. *Algoritmul SMDF calculează în mod corect fluxul maxim în reţeaua dinamică staţionară bipartită.*

Teorema 3.6. *Algoritmul SMDF aplicat pe reţeaua dinamică staţionară bipartită are complexitatea* $O(max\{n_1 m + n_1^3 \log(n_1 \bar{h}), nT\})$.

4. Aplicaţii şi concluzii

4.1 Aplicaţii

Fluxurile dinamice apar în rezolvarea multor probleme, incluzând sisteme de producţie-distribuţie, planificare economică, sisteme energetice, sisteme de trafic şi sisteme de evacuare a clădirilor. În continuare vom prezenta o aplicaţie referitor la evacuarea unei clădiri. Această aplicaţie reprezintă conţinutul unui articol al autoarei acestei teze [66].

Vom arăta cum o problemă de evacuare a unei clădiri este modelată ca o problemă de flux dinamic. În acest exemplu, obiectul care urmează a fi evacuat este reprezentat de o reţea numită reţea de evacuare în care nodurile corespund părţilor relevante ale obiectului de evacuare, iar arcele reprezintă conexiuni între aceste părţi. Unul sau mai multe noduri se disting ca noduri sursă, iar altele ca noduri destinaţie. Grupul de persoane evacuate este modelat ca flux dinamic. Un model de reţea dinamică al obiectelor de evacuare are următoarele componente: orizontul de timp şi unităţile de bază, nodul, arcul, sursa şi destinaţia. Orizontul de timp T este împărţit în perioade de timp finite $t = 0, 1, ..., T$. Nodurile sunt folosite pentru a modela punctele şi/sau locaţiile conectate. Există câteva alternative de luat în considerare la definirea unui nod. Nodurile care reprezintă locaţii care găzduiesc un număr semnificativ de persoane sunt nodurile sursă, iar ieşirile clădirii sunt noduri destinaţie în reţea. Oferta unui nod sursă este dată de numărul de persoane evacuate din locaţia asociată nodului. Fiecare nod din reţea are o capacitate care reprezintă numărul maxim de persoane evacuate care pot ocupa în siguranţă un nod la un moment dat.

Pentru detalii se poate consulta lucrarea [50].

Exemplu. În Figura 4.1 este prezentată o secţiune de plan a primului etaj al unui cămin studenţesc. Această secţiune are treisprezece camere şi un coridor. Este conectată la uşa de ieşire din clădire, de la parter, printr-o scară. O altă ieşire, mai precis ieşirea de urgenţă, din clădire este localizată la capătul din stânga al coridorului.

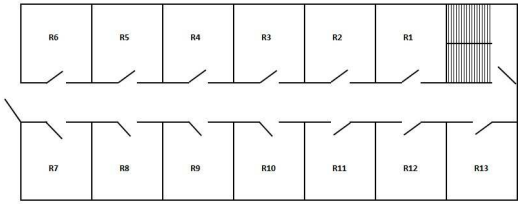

Figura 4.1: Planul secţiune de la etajul întâi al clădirii

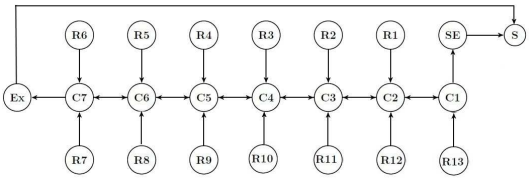

Figura 4.2: Reprezentarea reţelei

În Figura 4.2 am reprezentat reţeaua corespunzătoare planului de secţiune de la etaj unde: $R1 - R13$ reprezintă camerele, $C1 - C7$ reprezintă coridorul, SE – scările de ieşire, Ex – uşa de iesire de la capătul din stânga al coridorului, şi S un nod suplimentar ca zonă de siguranţă comună. Pentru a defini arcele de reţea, se consideră o situaţie în care unele persoane sunt situate în camera $R9$ chiar înainte de anunţarea unei evacuări. De îndată ce vor afla că va fi o evacuare, persoanele vor ieşi din camera lor şi vor intra în zona de coridor modelată de nodul $C5$. Apoi, o parte dintre persoane vor alege să se mute la $C6$ deoarece ei vor să iasă afară folosindu-se de uşa de ieşire de la capătul din stânga al coridorului, în timp ce alţii aleg să se mute la $C4$ deoarece ei vor să iasă afară folosindu-se de uşa de ieşire de la parter. Situaţia similară este luată în considerare şi pentru alte încăperi. Aceste posibilităţi de mişcare determină direcţia arcelor din modelul de reţea. În timpul procesului de evacuare, o conexiune între două noduri poate fi temporară datorită, de exemplu, blocării prin foc sau fum. În acest caz, un arc care reprezintă o conexiune trebuie, de asemenea, să fie temporar, capacitatea arcului scade în timp şi devine zero după câteva momente. Prin urmare, timpul necesar pentru traversarea arcului de la un nod la altul poate fi variabil în timp. Aplicaţia analizează evacuarea unei clădiri cu număr de ocupanţi cunoscut sau necunoscut. Atunci când numărul de locuitori dintr-o clădire este greu de estimat, suntem interesaţi să găsim numărul maxim de persoane evacuate care pot fi trimise în zona de siguranţă într-un anumit orizont de timp T.

4.2 Concluzii

În lucrarea de faţă am abordat problema claselor de fluxuri dinamice axându-ne, în special, pe probleme de flux în reţele dinamice planare şi reţele dinamice bipartite având la bază contribuţiile originale ale autoarei. În capitolele 2 şi 3 au fost prezentate principalele rezultate ale autoarei, publicate sau trimise spre publicare în literatura de specialitate.

În capitolul 2 este prezentată construcţia reţelei statice expandate reduse cu drumuri dinamice, precum şi abordarea statică a fluxului maxim în reţele dinamice planare cu margine inferioară zero. De asemenea, se prezintă abordarea dinamică a determinării fluxului maxim în reţele dinamice planare cu margine inferioară zero şi în reţele cu margine inferioară pozitivă, precum şi problema fluxului minim în această clasă de probleme.

În capitolul 3 am abordat problema fluxurilor în reţele dinamice bipartite, atât în ceea ce priveşte abordarea statică, cât şi abordarea dinamică.

Pentru fiecare problemă prezentată în capitolele 2 şi 3 am propus câte un algoritm prezentat, în mod detaliat, în pseudocod, precum şi câte un exemplu sugestiv pentru claritatea fiecărei probleme expuse.

În Tabelul 4.1, de mai jos, am prezentat, sintetizat, complexitatea algoritmilor descrişi în capitolele 2 şi 3.

Reţea	**Abordare**	**Flux**	**Margine inferioară**	**Complexitate**
dinamică planară	statică	maxim	$e = 0$	$O(nT \log^{1.5} nT)$
			$e > 0$	$-$
		minim	$e > 0$	$-$
	dinamică	maxim	$e = 0$	$O(n^5 \log n)$
			$e > 0$	$O(n^2T^2\bar{q})$
		minim	$e > 0$	$O(n^2T^2)$
dinamică bipartită	statică	maxim	$e = 0$	$O(n_1mT^2 + n_1^3T^3)$
			$e > 0$	$O(n_1mT^2 + n_1^3T^3)$
		minim	$e > 0$	$O(n_2mT^2 + n_2^3T^3)$
	dinamică	maxim	$e = 0$	$O(max\{n_1mT^2 + n_1^3 \log(n_1\bar{h}), nT\})$
			$e > 0$	$-$
		minim	$e > 0$	$-$

Tabelul 4.1: Complexitatea algoritmilor pentru determinarea fluxului maxim şi a fluxului minim în reţele dinamice planare şi reţele dinamice bipartite

Multe probleme interesante de flux în reţele dinamice planare şi reţele dinamice bipartite sunt încă nestudiate, şi anume:

- problema fluxului maxim în reţele dinamice planare cu margine inferioară pozitivă, abordarea statică;
- problema fluxului minim în reţele dinamice planare, abordarea statică;
- problema fluxului maxim în reţele dinamice bipartite cu margine inferioară pozitivă, abordarea dinamică;
- problema fluxului minim în reţele dinamice bipartite, abordarea dinamică;
- determinarea fluxului maxim sau minim în reţele dinamice bipartite cu alţi algoritmi;
- alte probleme de fluxuri în reţele dinamice.

Cercetările viitoare ale autoarei acestei teze se vor direcţiona şi spre problemele prezentate mai sus.

Rezultate originale:

1. studiul din **capitolul 2**, **secţiunea 2.2.2** privind fluxul maxim în reţele dinamice planare cu margine inferioară pozitivă a fost publicat în lucrarea: **C. Schiopu**, E. Ciurea, Maximum flows in planar dinamic networks with lower bounds, Fundamenta informaticae, Vol. 163(2), 2018, pp. 189-204 - jurnal indexat ISI cu **Factor de Impact 0.725 şi SRI: 0.531**
2. studiul realizat în **secţiunea 2.2.1** cu privire la abordarea dinamică a fluxului maxim în reţele dinamice planare cu margine inferioară zero a fost publicat în lucrarea: **C. Schiopu**, E. Ciurea, The maximum flows in planar dinamic networks, International Journal of Computers Communications & Control, Vol. 11(2), 2016, pp. 282-291 - jurnal indexat ISI cu **Factor de Impact 1.290 şi SRI: 0.338**
3. studiul realizat în **secţiunea 2.2.3** cu privire la abordarea dinamică a fluxului minim în reţele dinamice planare se regăseşte în lucrarea: E. Ciurea, **C. Schiopu**, The minimum flows in directed planar networks, lucrare prezentată în cadrul International Conference on Control, Decision and Information Technologies 2019, Franţa - conferinţă indexată ISI
4. studiul realizat în **secţiunea 3.1.1** cu privire la abordarea statică privind fluxul maxim în reţele dinamice bipartite cu marginea inferioară zero a fost publicat în lucrarea: **C. Schiopu**, The maximum flows in bipartite dynamic networks. The static approach., Annals of the University of Craiova - Mathematics

and Computer Science Series, Vol. 43(2), 2016, pp. 200-209 - jurnal indexat în WOS, însă fără factor de impact

5. studiul din **secţiunea 3.1.2**, privind fluxul maxim în reţele dinamice bipartite cu margine inferioară pozitivă a fost publicat în lucrarile: **C. Schiopu**, E. Ciurea, Two flow problems in dynamic network, International Journal of Computers Communications & Control, Vol. 12(1), 2017, pp. 103-115 - jurnal indexat ISI cu **Factor de Impact 1.290 şi SRI: 0.338** şi **C. Schiopu**, E. Ciurea, The maximum flows in bipartite dynamic networks with lower bounds. The static approach. în Proceedings in IEEE Xplore of the 6^{th} International Conference on Computers, Communications and Control (ICCCC), Oradea, România, 2016, pp. 10-15.
6. construcţia reţelei statice expandate reduse cu drumuri dinamice şi studiul din **secţiunea 3.2.1** cu privire la abordarea dinamică a fluxului maxim în reţele dinamice bipartite cu margine inferioară zero a fost publicat în lucrarea: **C. Schiopu**, The maximum flows in bipartite dynamic networks, Bulletin of the Transilvania University of Braşov Vol. 7(56), 2014, pp. 193-202

Rezultate originale trimise spre publicare:

1. studiul realizat în **secţiunea 2.1.2** cu privire la fluxul maxim în reţele dinamice planare cu margine inferioară zero se regăseşte în lucrarea: **C. Schiopu**, Maximum flows in planar dynamic networks. The static approach - lucrare trimisă spre recenzie la revista Romanian Journal of Information Science and Technology - **Factor de Impact 0.288 şi SRI: 0.197**
2. studiul din **secţiunea 3.1.3** cu privire la fluxurile minime în reţele dinamice bipartite se regăseşte în lucrarea: E. Ciurea, **C. Schiopu**, The minimum flows in bipartite dynamic Networks. The static approach. - lucrare trimisă spre recenzie la revista Informatica an International Journal of Computing and Informatics

Alte studii realizate de-a lungul studiilor doctorale sunt:

1. studiul fluxului minim în reţele orientate planare $s-t$ în lucrarea: E. Ciurea, O. Georgescu, **C. Schiopu** Minimum flows in directed s-t planar networks with arcs and nodes capacities publicată în proceeding-ul International Conference on Control, Decision and Information Technologies 2018, Grecia - conferinţă indexată ISI.
2. studiul fluxului minim în reţele planare $1-n$ statice orientate şi în reţele dinamice cu arce şi noduri capacitate în lucrarea: E.

Ciurea, O. Georgescu, **C. Schiopu** Minimum flows in directed 1-n planar static and dynamic networks with arcs and nodes capacities - lucarare trimisă spre recenzie la revista Journal of Circuits, Systems and Computers - **Factor de Impact 0.595 şi SRI: 0.183**

3. studiul privind problema fluxului maxim în reţele dinamice publicat în lucrările: **C. Schiopu** Applications of the maximal dynamic network flow problem în proceeding-ul 14th International Conference on Mathematics and Applications 2015, Politehnica University of Timişoara, România şi **C. Schiopu** An applications of freight systems, Bulletin of the Transilvania University of Braşov Vol. 8(57), 2015, pp. 499-506.
4. studiul privind problema fluxului maxim în reţele dinamice bipartite cuprins în lucrarea: **C. Schiopu** Maximum Flows in Bipartite Dynamic Network - trimisă spre recenzie la Bulletin of the Transilvania University of Braşov.

Bibliografie

[1] R. Ahuja and J. Orlin and C. Stein and R. Tarjan: Improved algorithms for bipartite network flows. SIAM Journal of Computing *23* (1994), 5, 906–933.

[2] R. Ahuja and T. Magnanti and J. Orlin: Network Flows. Theory, algorithms and applications. Prentice Hall, Inc., Upper Saddle River, NJ, 1993.

[3] J.E. Aronson: A survey of dynamic network flows. Annals of Operation Research *20*(1) (1989), 1–66.

[4] M. S. Bazaraa and J. J. Jarvis and H. D. Sherali: Linear Programming and Network Flows (Fourth Edition). Wiley, Hoboken, NJ, 2010.

[5] J.A. Bondy and U.S.R. Murty: Graph theory with applications. Macmillan, London, 1976.

[6] G. Borradaile and P.N. Klein and S. Mozes and Y. Nussbaum and C. Wulf-Nilsen: Multiple-source multiple-sink maximum flow in directed planar graph in near-lineare time. In: Proceedings of the 2011 IEEE 52^{nd} Annual Symposium on Foundations of Computer Science, Palm Springs, California, USA, 2011, pp. 170-179.

[7] G. Borradaile and P. Klein: An $O(nlogn)$ algorithm for maximum st-flow in a directed planar graph. J. ACM *56* (2006), 2, 1–30.

[8] X. Cai and D. Sha and C. Wong: Time-varying Network Optimization. Springer, 2007.

[9] W.K. Chen: Theory of Nets: Flows in Networks. Wiley, New York, 1990.

[10] L. Ciupală: Algoritmi fundamentali din teoria grafurilor. Aplicaţii. Editura Universităţii Transilvania Braşov, Braşov, 2007.

[11] L. Ciupală and E. Ciurea : A parallel algorithm for the minimum flow problem in bipartite networks. In: Proceedings of the 12^{th} WSEAS International Conference on COMPUTERS, Heraklion, Greece, 2008, pp. 203–207.

[12] L. Ciupală and E. Ciurea : About preflow algorithms for the minimum flow problem. In: Journal WSEAS Transaction on Computer Research, *3* (2008), pp. 35–42.

[13] L. Ciupală and E. Ciurea : Sequential and parallel deficit scaling algorithms for the minimum flow in bipartite networks. In: Journal WSEAS Transaction on Computer Research, *7*(10) (2008), pp. 1545–1554.

[14] L. Ciupală and E. Ciurea : A highest-label preflow algorithm for the minimum flow problem. In: Proceedings of the 11^{th} WSEAS International Conference on COMPUTERS, Agios Nikolaos, Crete Island, Greece, 2007, pp. 168–171.

[15] L. Ciupală and E. Ciurea: A parallel FIFO preflow algorithm for the minimum flow problem. International Journal of Computers, Communications and Control *1* (2006), 5, 135–139.

[16] E. Ciurea: Algoritmi: Introducere în algoritmica grafurilor. Editura Tehnică, Bucureşti, 2001.

[17] E. Ciurea: Les problemes des flots dynamiques. Cahiers du CERO *26* (1984), 3–9.

[18] E. Ciurea: An algorithm for minimal dynamic flow. Korean Journal of Computaional and Applied Mathematics *7* (2000), 2, 259-269.
[19] E. Ciurea: Second best temporally repeated flow. Korean Journal of Computaional and Applied Mathematics *9*(1) (2002), 77–86.
[20] E. Ciurea and L. Ciupală: Algoritmi. Introducere în algoritmica fluxurilor în reţele. Editura Matrix Rom, Bucureşti, 2006.
[21] E. Ciurea and L. Ciupală: A C-scaling algorithm for the minimum flow problem. In: Proceedings of the Seventh Balkan Conference on Operational Research, 2005, pp. 1–4.
[22] E. Ciurea and L. Ciupală: A L-scaling algorithm for the minimum flow problem. In: Proceedings of the Seventh International Conference on Operation Informatics in Economy, 2005, pp. 556–561.
[23] E. Ciurea and L. Ciupală: Sequential and parallel algorithms for minimum flows. Journal of Applied Mathematics and Computing *1*5 (2004), 1-2, 53–75.
[24] E. Ciurea and L. Ciupală: Algorithms for minimum flows. Computer Science Journal of Moldova *9* (2001), 3(27), 275–290.
[25] E. Ciurea and O. Georgescu: Minimm flows in directed s-t planar networks. Bulletin Mathematique de la Société des Sciences Mathématics de Roumaine *5*3(101) (2010), 4, 305-313.
[26] E. Ciurea and O. Georgescu and M. Iolu: Algorithms for minimum flows. Dynamic tree implementation. Studia Universitas Babes Bolyai, Informatică *L*III (2008), 1, 73–82.
[27] E. Ciurea and O. Georgescu: Minimum flow in unit capacity networks. Analele Universităţii Bucureşti *L*V (2006), 1, 11–20.
[28] E. Ciurea and O. Georgescu and D. Marinescu: Improved algorithms for minimum flows in bipartite networks. International Journal of Computers *2* (2008), 4, 351–360.
[29] E. Ciurea and O. Georgescu and D. Marinescu: Minimum flows in bipartite networks. In: Proceedings of the 10^{th} WSEAS International Conference on Mathematical and Computational Methods in Science and Engineering (MACMESE '08), Bucharest, România, 2008, pp. 491–495.
[30] E. Ciurea and O. Georgescu and **C. Schiopu**: Minimum flows in directed s-t planar networks with arcs and nodes capacities. In: Proceedings of the 5^{th} International Conference on Control, Decision and Information Technologies (CoDIT '18), Thessaloniki, Greece, 2018, pp. 110–115.
[31] E. Ciurea and O. Georgescu and **C. Schiopu**: Minimum flows in directed 1-n planar static and dynamic networks with arcs and nodes capacities. Journal of Circuits, Systems, and Computers, trimisă spre publicare.
[32] E. Ciurea and **C. Schiopu**: The Minimum Flows in Bipartite Dynamic Networks. The Static Approach. Informatica an International Journal of Computing and Informatics, trimisă spre publicare.
[33] E. Ciurea and **C. Schiopu**: Minimum flows in directed planar dynamic networks. 6^{th} International Conference on Control, Decision and Information Technologies, Paris, France, 2019, în curs de publicare.
[34] C. Croitoru: Tehnici de bază în optimizarea combinatorie. Editura Universităţii Al. I. Cuza, Iaşi, 1992.
[35] G. Demoucron and Y. Malgrange and R. Pertuiset: Graphes planaires: reconnaissance et construction de représentations planaires topologiques. Rev. Francaise Recherche Opérationnelle *8* (1964), 33–47.
[36] E. Dinic: Algorithm for solution of a problem of maximum flow in network with power estimation. Soviet Mathematics Doklady *1*1 (1970), 5, 1277–1280.

[37] S. Even and R. Tarjan: Network flow and testing graph connectivity. SIAM Journal on Computing *4* (1975), 4, 507–518.
[38] J. Fakcharoenphol and S. Rao: Planar graphs, negative weight edges, shortest paths, and near linear time. Journal of Computer and System Sciences *72* (2006), 5, 868–889.
[39] L. Ford and D. Fulkerson: Flows in Networks. Princenton University Press, Princenton, NJ, 1962.
[40] L. Ford and D. Fulkerson: Maximal flows through a network. Canadian Journal Mathematics *8* (1956), 399–404.
[41] G. Frederickson: Fast algorithms for shortest path in planar graphs, with applications. SIAM Journal on Computing *16*(6) (1987), 1004–1022.
[42] O. Georgescu: Algorithms for minimum flows. Dynamic tree implementation. Bulletin of the Transilvania University of Braşov. Series III: Mathematics, Informatics, Physics*1*,(50) (2008), 4, 513–524.
[43] O. Georgescu: Minimum flow in network using dynamic tree implementation. In: Proceedings of the 3^{rd} Annual South-East European Doctoral Student Conference, 2008, pp. 192–204.
[44] O. Georgescu and E. Ciurea: Minimum flows in directed $s-t$ planar networks. Bull. Math. Soc. des Sci. Math. Roumanie Tome *53*(101) (2010), 4, 305–313.
[45] O. Georgescu and E. Ciurea: An algorithm for minimum flows. In: WSEAS Transaction on Computers, 2008, pp. 2042–2051.
[46] O. Georgescu and E. Ciurea: Decreasing path algorithm for minimum flow. Dynamic tree implementation. In: Proceedings of the 12^{st} WSEAS International Conference on Computers, 2008, pp. 235–240.
[47] A. Gibbons: Algorithmic graph theory. Cambridge Univ. Press, Cambridge, 1984.
[48] A.V. Goldberg and R.E. Tarjan: A new approach to the maximum flow problem. Journal of ACM *35* (1988), 4, 921–940.
[49] D. Gusfield and C. Martel and D. Fernandez-Baca: Fast algorithms for bipartite network flow. SIAM Journal on Computing *16*(2) (1987), 237–251.
[50] H.W. Hamacher and S.A. Tjandra: Mathematical modelling of evacuation problems: a state of the art. Berichte des Fraunhofer ITWM (2001), 24, 1–38.
[51] R. Hassin: Maximum flow in (s,t) planar networks. Information Processing Letters *13* (1981), 3, 107-117.
[52] R. Hassin and D.B. Johnson: An $O(nlog^2n)$ algorithm for maximum flow in undirected planar networks. SIAM Journal on Computing *14* (1985), 3, 612–624.
[53] J. Hopcroft and R.E. Tarjan: Efficient planarity testing. JACM *21*(4) (1974), 1, 549–568.
[54] A. Itai and Y. Shiloach: Maximum flow in planar networks. SIAM Journal on Computing *8* (1979), 2, 135–150.
[55] D.B. Johnson and S. Venkatesan: Using divide and conquer to find flows in directed planar networks in $O(n^{3/2}\log n)$ time. In: Proceedings of the 20^{th} Annual Allerton Conference on Communication, Control, and Computing, Allerton House, Monticello, 1982, pp. 898-905.
[56] D. Jungnickel: Graphs, networks and algorithms. Springer, Berlin, 1999.
[57] A Karzanov and J. Naor: Determining the maximal flow in a network by the method of preflows. Doklady Mathematics *15* (1974), 434-437.
[58] S. Khuller and J. Naor: Flows in planar graphs with vertex capacities. Algorithmica *11* (1994), 3, 200-225.

[59] P.N. Klein: Multiple-source shortest path in planar graphs. In: proceedings of the 16^{th} Annual ACM-SIAm Symposium on Discrete Algorithm, 2005, pp. 146-155.
[60] P.N. Klein and S. Subramanian: A fully dynamic approximation scheme for shortest paths in planar graphs. Algorithmica *22* (1998), 3, 235-249.
[61] G.L. Miller: Finding small simple cycle separators fo 2-connected planar graphs. Journal of Computer and System Sciences *32* (1986), 3, 265–279.
[62] G.L. Miller and J. Naor: Flow in planar graphs with multiple sources and sinks. SIAM Journal on Computing *24* (1995), 5, 1002–1017.
[63] S. Mozes and C. Wulff-Nilsen: Shortest path in planar graphs with real lengths in $O(nlog^2n/loglogn)$ time. In: Proceedings of the 18^{th} European Symposium on Algorithms, Liverpool, UK, 2010, pp. 206–217.
[64] H.F. Salehi and S. Khadayifar and M.A. Raayatpanoh: Minimum flow problem on network flows with time-varying bounds. Applied Mathematical Modelling *36* (2012), 9, 4414–4421.
[65] **C. Schiopu**: The maximum flows in bipartite dynamic networks. The static approach. Annals of the University of Craiova - Mathematics and Computer Science Series *43* (2016), 2, 200–209
[66] **C. Schiopu**: Applications of the maximal dynamic network flow problem. In: Proceedings of the 14th International Conference on Mathematics and its Applications Politehnica University of Timişoara, România, 2015, pp. 113-120.
[67] **C. Schiopu**: An application of freight systems. Bulletin of the Transilvania University of Braşov *8*(57) (2015), 2, 499–506
[68] **C. Schiopu**: The maximum flows in bipartite dynamic networks. Bulletin of the Transilvania University of Braşov *7*(56) (2014), 2, 193–202.
[69] **C. Schiopu**: Maximum flows in planar dynamic networks. The static approach. Romanian Journal of Information Science and Technology, trimisă spre publicare.
[70] **C. Schiopu** and E. Ciurea: Two flow problems in dynamic networks. International Journal of Computers Communications & Control *12*(1) (2017), 1, 103–115
[71] **C. Schiopu** and E. Ciurea: The maximum flows in planar dynamic networks. International Journal of Computers Communications & Control *11*(2) (2016), 2, 282–291
[72] **C. Schiopu** and E. Ciurea: The maximum flows in bipartite dynamic networks with lower bounds. The static approach. In: Proceedings in IEEE Xplore of the 6^{th} International Conference on Computers, Communications and Control (ICCCC), Oradea, România, 2016, pp. 10–15.
[73] **C. Schiopu** and E. Ciurea: Maximum flows in planar dynamic networks with lower bounds. Fundamenta Informaticae, *163* (2018), 2, 189–204
[74] **C. Schiopu**: Maximum Flows in Bipartite Dynamic Networks. Bulletin of the Transilvania University of Braşov Series III, trimisă spre publicare.
[75] M. Skutella: An Introduction to Network Flows Over Time. In book: Research Trends in Combinatorial Optimization, (2009), 451-482.
[76] D.D. Sleator and R.E. Tarjan: A data structure for dynamic trees. J. Comput. System Sci. *26* (1983), 3, 362–391.
[77] S.A. Tjandra: Dynamic Network Optimization with Application to the Evacuation Problem. Dissertation, Universitat Kaiserslautern, 2003.
[78] I. Tomescu: Probleme de combinatorică şi teoria grafurilor. Editura Didactică şi Pedagogică, Bucureşti, România, 1981.
[79] W. Wilkinson: An Algorithm for universal maximal dynamic flows in a network. Oper. Res. *19* (1971), 7, 1602–1612.

Anexe

Anexa 1. Scurt rezumat al tezei

Teza de doctorat, intitulată *Clase de fluxuri dinamice*, este structurată în 4 capitole, un capitol care conţine noţiuni utilizate pe parcursul acestei lucrări, 2 capitole care conţin rezultate originale şi un capitol cu aplicaţii şi concluzii. Primul capitol, intitulat *Fluxuri în reţele*, descrie noţiuni generale privind fluxurile în reţele prezentând algoritmi pentru determinarea fluxurilor în reţele statice oarecare şi în reţele statice speciale, precum şi algoritmi pentru determinarea fluxurilor în reţele dinamice oarecare. În capitolul 2, intitulat *Fluxuri în reţele dinamice planare*, am prezentat modul în care este construită reţeaua statică expandată redusă cu drumuri dinamice şi am dezvoltat algoritmi pentru flux maxim şi algoritmi pentru flux minim folosind abordarea statică si cea dinamică. Algoritmii propuşi în acest capitol sunt prezentaţi în detaliu, în pseudocod. În capitolul 3, intitulat *Fluxuri în reţele dinamice bipartite*, am studiat problema fluxului maxim şi cea a fluxului minim în reţele dinamice bipartite folosind abordarea statică şi fluxul maxim în reţele dinamice bipartite cu margine inferioară zero folosind abordarea dinamică. În ultimul capitol sunt prezentate concluziile finale obţinute în urma cercetării claselor de fluxuri dinamice, fiind descrise contribuţiile aduse temei abordate, realizând, totodată, o analiză comparativă a complexităţii algoritmilor referitor la abordarea statică şi cea dinamică şi prezentând o aplicaţie a fluxurilor dinamice. Rezultatele cuprinse în această teză pot fi extinse şi în alte probleme de fluxuri în reţele dinamice.

* *

*

The PhD thesis entitled, *Dynamic Flow Classes*, is structured into 4 chapters: a chapter that contains notions used during this work, 2 chapters that contain original results and a chapter with conclusions and further research. The first chapter, entitled *Flows in Networks*, describe general concepts of network flows presenting algorithms for determination of flows in general static network and special static network, as well as algorithms for determining flows in general dynamic networks. In the second chapter, entitled *Flows in planar dynamic*

networks, we presented how is constructing the static reduced expanded network with dynamic paths and we have developed algorithms for maximum and minimum flows using static approach and dynamic approach. The algorithms proposed in this chapter are presented, in detail, in the pseudocode. In Chapter 3, entitled *Flows in Dynamic Bipartite Networks*, we studied the problem of the maximum flow and the minimum flow in the bipartite dynamic networks using static approach and the maximum flow in dynamic bipartite networks with lower bound zero using dynamic approach.

In the last chapter are presented the final conclusions obtained from the research of dynamic flow classes, describing the contributions to the topic approached, while also performing a comparative analysis of the complexity of the algorithms regarding the static and the dynamic approach and presenting an application of dynamic flows. The results contained in this thesis can be extended to other problems of flows in dynamic networks.

Anexa 2. Curriculum Vitae

Informaţii personale	**Camelia SCHIOPU** E-mail: camelia.s@unitbv.ro
Experienţa profesională	**2008-prezent** – asistent universitar în cadrul Departamentului de Management şi informatică economică al Facultăţii de Ştiinţe Economice şi Administrarea Afacerilor a Universităţii *Transilvania* din Braşov **2005-2011** – analist în cadrul Departamentului de Informatizare al Universităţii *Transilvania* din Braşov **1998-2005** – informatician în cadrul serviciului Secretariat Rectorat al Universităţii *Transilvania* din Braşov
Educaţie	**2014-prezent** – doctorand, domeniul informatică, Facultatea de Matematică şi Informatică, Universitatea *Transilvania*, Braşov **2009-2010** – studii postuniversitare în specializarea Antreprenoriat şi Supply Chain Management Facultatea de Ştiinţe Economice şi Administrarea Afacerilor, Universitatea *Transilvania* din Braşov, **2005-2007** - studii de mastrat în specializarea Tehnologii Internet Facultatea de Matematică şi Informatică, Universitatea *Transilvania* din Braşov, **1999-2003** – studii de licenţă în informatică, Facultatea de Matemtică şi Informatică, Universitatea *Transilvania*, Braşov **1995-1998** – studii superioare de scurtă durată în informatică, Colegiul Universitar, Forestier, Economic şi de Informatică, Universitatea *Transilvania*, Braşov
Competenţe lingvistice	**Limbi străine cunoscute/Autoevaluare:** Engleză – înţelegere-mediu, vorbire-mediu, scriere-mediu Franceză – înţelegere-începător, vorbire-începător, scriere-începător
Experienţă în cercetare	3 lucrări ştiinţifice în reviste ISI 10 lucrări ştiinţifice în reviste BDI 7 participări la conferinţe internaţionale - membru în echipa proiectului de cercetare *Modele şi indicatori de evaluare în contextul societăţii a calităţii cercetării ştiinţifice, în universităţi bazate pe cunoaştere*, în perioada 2006-2008

Personal information	**Camelia SCHIOPU** E-mail: camelia.s@unitbv.ro
Work experience	**2008-now** – Assistant professor, Department of Management and Economic Informatics, Faculty of Economic Sciences and Business Administration, *Transilvania* University of Braşov **2005-2011** – analyst in the IT departament of the *Transilvania* University of Braşov **1998-2005** – informatician at the Rectorate Secretariat *Transilvania* University of Braşov
Education	**2014-now** – PhD student in Informatics, Faculty of Mathematics and Computer Science, *Transilvania* University of Braşov **2009-2010** – Postgraduate studies in specializarea Entrepreneurship and Supply Chain Management, Faculty of Economic Sciences and Business Administration, *Transilvania* University of Braşov **2005-2007** - Master studies in Internet Technology Faculty of Mathematics and Computer Science, *Transilvania* University of Braşov **1999-2003** – Bachelor degree in Informatics, Faculty of Mathematics and Computer Science, *Transilvania* University of Braşov **1995-1998** – Short-term higher education in informatics, University College, Forestry, Economic and Informatics, *Transilvania* University of Braşov
Personal skills	**Foreign languages/Self-evaluation:** English – understanding-medium, speaking-medium, writing-medium
Experience in research	3 scientific papers in ISI journals 10 scientific papers in BDI journals 7 participation at international conferences - team member of the projects *Models and indicators for assessing the quality of scientific research in universities in the context of knowledge, based society* during 2006-2008