Annals of the University of Craiova, Mathematics and Computer Science Series
Volume 43(2), 2016, Pages 188–199
ISSN: 1223-6934

Maximum flows in parametric dynamic networks with lower bounds

Nicoleta Avesalon (Grigoras)

Abstract. This paper presents and solves the problem of the maximum flow in parametric dynamic networks with nonzero lower bounds. It also provides a rigorous formulation of the problem of the maximum flow in dynamic networks and in static parametric networks, both with nonzero lower bounds. The aim of this paper is to present a novel and efficient algorithm which is developed for solving the problem of maximum flow in parametric dynamic networks with nonzero lower bounds. The proposed approach consists in applying an algorithm for finding the maximum parametric flow in a reduced expanded network, i.e. the static network obtained by expanding the original dynamic network. The article also presents a numerical example on how the algorithm works in a dynamic network having dynamic upper bounds which vary linearly with a parameter.

Key words and phrases. Dynamic network, parametric network, maximum flow.

1. Introduction

The classical model of the static flows in networks arises in many combinatorial applications, some of which might not appear to be maximum flow problems. The static flows model arises directly in problems as machine scheduling, assignment of computer modules to computer processors, tanker scheduling [1] etc. In several other applications [2], [3], [4], [6], [7], [8], [9], [12], [16], where time is an essential ingredient, the related model of dynamic flow in networks needs to be use. While in the static problems, where there is no change of situation, the challenge is to find out more efficient and faster algorithms in order to tackle the large scale problems, within dynamic problems, the challenge is to respond faster to the changes which take place while the new problem is solved. A generalization of the dynamic maximum flow problem, in which the capacities of certain arcs are not only varying in time but also depend of a parameter [5], [9], is known as the problem of the maximum flow in parametric dynamic networks.

The present paper addresses the case of the maximum flows in parametric dynamic networks with nonzero lower bounds and proposes an approach for solving it which consists in transforming the maximum flow in the parametric dynamic network into the maximum flow in a related parametric static network. The algorithm developed in this paper solves the problem which arises by generalization of the dynamic network flow problem and of the parametric network flow problem, through the following assumptions:(1) both the dynamic network and the corresponding static expanded network have nonzero lower bounds, (2) the problem of the maximum flow will be

Received May 14, 2015. Revised October 26, 2016.

addressed in a non-stationary dynamic network, i.e. with time-varying transit times, as well as with lower and upper bounds of arcs which are also time-varying.

We mention that the static approach of the problem of the maximum flows in parametric dynamic networks with nonzero lower bounds has not been investigated so far by other researchers.

The remainder of this paper is organized as follows. Section 2 presents some basic notations and the terminology used for describing dynamic networks. Then, in Section 3, we expose the problem of the maximum flow in static parametric networks with nonzero lower bounds. The problem of the maximum flow in parametric dynamic networks with nonzero lower bounds is presented in Section 4. Finally, in Section 5 we give an example on how the algorithm works. In the presentation to follow, some familiarity with flow problems is assumed and therefore many details are omitted.

2. The maximum flows in dynamic networks.

Let $G = (N, A, l, u)$ be a static network with the set of nodes $N = \{1, ..., i, ..., j, ..., n\}$, the set of arcs $A = \{a_1, ..., a_k, ..., a_m\}, a_k = (i, j)$ the lower bound function $l : A \to \mathbf{R}^+$ and the upper bound (capacity) function $u : A \to \mathbf{R}^+$, where $\mathbf{R}$ is the set of real numbers. To define the maximal static flow problem, we distinguish two special nodes in the static network $G = (N, A, l, u)$: a source node 1 and a sink node n.

Let $\mathbf{N}$ be the set of natural numbers and let $H = \{0, 1, ...T\}$ be the set of periods, where T is a finite time horizon, $T \in \mathbf{N}$. Let $h : A \times H \to \mathbf{N}$ be the transit time function, $l_h : A \times H \to \mathbf{R}^+$ the time-dependent lower bound function and $u_h : A \times H \to \mathbf{R}^+$ the time-dependent upper bound function. For each arc $(i, j) \in A$, $h(i, j; t)$ represents the transit time of arc (i, j) at time t, $t \in H$.

A dynamic flow from source node 1 to sink node n is any flow from 1 to n in which not less than $l_h(i, j; t)$ and not more than $u_h(i, j; t)$ flow units are starting from node i at time t and are arriving at node j at time $\theta = t + h(i, j; t)$ for all arcs (i,j) and for all t values. The maximal dynamic flow problem for T time periods is to determine a flow function $f_h : A \times H \to \mathbf{N}$, which should satisfy the following conditions in dynamic network $G_h = (N, A, h, l_h, u_h)$:

$$\sum_{t=0}^{T}(\sum_{j} f_h(i, j; t) - \sum_{k}\sum_{\tau} f_h(k, i; \tau)) = v_H, \tag{2.1.a}$$

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

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

$$l_h(i, j; t) \leq f_h(i, j; t) \leq u_h(i, j; t), \text{ for all } (i, j) \in A \text{ and for all } t \in H \tag{2.2}$$

$$\max v_H, \tag{2.3}$$

where $\tau = t - h(k, i; \tau), v_H = \sum_{t=o}^{T} v(t)$, $v(t)$ is the flow value at time t and $f_h(i, j; t) = 0, (i, j) \in A, t \in \{T - h(i, j; t) + 1, ..., T\}$.

Obviously, the problem of finding a maximum dynamic flow is more complex than the problem of finding a maximum static flow. Happily, this complication can be solved by rephrasing the dynamic flow problem into a static flow problem in a static network $G' = (N', A', l', u')$ which is called reduced expanded network. First, we form the expanded network $G_H = (N_H, A_H, l_H, u_H)$ with $N_H = \{i_t | i \in N, t \in H\}$,$A_H = \{(i_t, j_\theta) | (i,j) \in A, t = 0, 1, ..., T - h(i,j;t)\}$, $l_H(i_t, j_\theta) = l_h(i,j;t)$,$u_H(i_t, j_\theta) = u_h(i,j;t)$, $(i_t, j_\theta) \in A_H$. We have $|N_H| = n(T+1)$ and $|A_H| \leq m(T+1) - \sum_A \bar{h}(i,j)$, where $\bar{h}(i,j) = min\{h(i,j;0), ..., h(i,j;T)\}$. Clearly, any dynamic flow from the source node 1 to the sink node n in dynamic network G_h is equivalent to a static flow from the source nodes $1_0, 1_1, ..., 1_T$ to the sink nodes $n_0, n_1, ..., n_T$ in static network G_H and vice versa. We can further reduce the multiple source, multiple sink problem in network G_H to the single source, single sink problem by introducing a supernode 1^* and a supersink node n^* and by building the superexpanded network $G_H^* = (N_H^*, A_H^*, l_H^*, u_H^*)$, where $N_H^* = N_H \cup \{1^*, n^*\}$, $A_H^* = A_H \cup \{(1^*, 1_t) | t \in H\} \cup \{(n_t, n^*) | t \in H\}$, $l_H^*(i_t, j_\theta) = l_H(i_t, j_\theta)$, $u_H^*(i_t, j_\theta) = u_H(i_t, j_\theta)$, $(i_t, j_\theta) \in A_H$, $l_H^*(1^*, 1_t) = l_H^*(n_t, n^*) = 0$, $u_H^*(1^*, 1_t) = u_H^*(n_t, n^*) = \infty$, $t \in H$. Next, we build the reduced expanded network $G' = (N', A', l', u')$ as follows. We define the function h^*, $h^* : A_H^* \to \mathbf{N}, h^*(1^*, 1_t) = h^*(n_t, n^*) = 0, t \in H$, $h^*(i_t, j_\theta) = h(i,j;t), (i_t, j_\theta) \in A_H$. Let $d^*(1^*, i_t)$ be the length of the shortest path from the source node 1^* to the node i_t and $d^*(i_t, n^*)$ the length of the shortest path from node i_t to the sink node n^* with respect to h^* in network G_H^* . The computation of $d^*(1^*, i_t)$ and $d^*(i_t, n^*)$, $i_t \in N_H^*$ is performed by means of the usual shortest path algorithms [1]. In network G' we rewrite the nodes $1^*, n^*$ by $1'$ respectively n'. We obtain $N' = \{1', n'\} \cup \{i_t | i_t \in N_H, d^*(1^*, i_t) + d^*(i_t, n^*) \leq T\}$, $A' = \{(1', 1_t) | 1_t \in N_H, d^*(1_t, n^*) \leq T\} \cup \{(n_t, n') | n_t \in N_H, d^*(1^*, n_t) \leq T\} \cup \{i_t, j_\theta) | (i_t, j_\theta) \in A_H, d^*(1^*, i_t) + h^*(i_t, j_\theta) + d^*(j_\theta, n^*) \leq T\}$ and l', u' are restriction of l_H^*, u_H^* at A'. It is easy to see that the network G' is always a partial subnetwork of G_H^*. Since an item released from a node at a specific time does not return to that location at the same or an earlier time, the networks G_H, G_H^*, G' cannot contain any circuit, and are therefore acyclic always.

In the most general dynamic model, the parameter $h(i) = 1$ is the waiting time at node i, and the parameters $l_h(i,t), u_h(i,t)$ are defined as lower bound and upper bound, which represents the minimum respectively the maximum amount of flow that can wait at node i from time t to t+1. This most general dynamic model is not discussed in this paper.

The maximum dynamic flow problem for T time periods in dynamic network G_h formulated in conditions (2.1), (2.2), (2.3) is equivalent with the maximum static flow problem in static network G' as follows:

$$\sum_{j_\theta} f'(i_t, j_\theta) - \sum f'(k_\tau, i_t) = \begin{cases} v', & \text{if } i_t = 1', & (2.4.a) \\ 0, & \text{if } i_t \neq 1', n', & (2.4.b) \\ -v', & \text{if } i_t = n', & (2.4.c) \end{cases}$$

$$l'(i_t, j_\theta) \leq f'(i_t, j_\theta) \leq u'(i_t, j_\theta),\ (i_t, j_\theta) \in A', \quad (2.5)$$
$$max\ v', \quad (2.6)$$

where by convention $i_t = 1'$ for t= -1 and $i_t = n'$ for t=T+1.

For further details we recommend the works [2], [3], [4], [6], [7], [10].

3. The maximum flow in parametric static networks

A natural generalization of the problem of the maximum flow in static networks can be obtained by making the upper bounds (capacities) of some arcs functions of a single parameter. Since the maximum flow value function in a parametric network is a continuous piecewise linear function of the parameter, the problem of the parametric maximum flow can alternately be defined as to find all the breakpoints and their corresponding maximum flows and minimum cuts. There are more approaches for solving the problem of the maximum flow in parametric static network [5],[8], [9]. The approach presented in this section is the one presented in [8].

A static network G = (N, A, l, u) with the upper bounds $u(i,j)$ of some arcs $(i,j) \in A$, functions of a real parameter λ is referred to as a parametric static network and is denoted by $\bar{G} = (N, A, l, \bar{u})$. The upper bound function $\bar{u} : A \times \mathbf{R}^+ \to \mathbf{R}^+$ is defined by relation:

$$\bar{u}(i,j;\lambda) = u_0(i,j) + \lambda \cdot U(i,j), \lambda \in [0, \Lambda] = I \tag{3.1}$$

where $U : A \to \mathbf{R}$ is the parametric part of the upper bound function $\bar{u}$ and $u_0 : A \to \mathbf{R}^+$ is the non parametric part of the function $\bar{u}, \bar{u}(i,j;0) = u_0(i,j), (i,j) \in A$. The $U(i,j)$ must satisfy $U(i,j) \geq (l(i,j) - u_0(i,j)/\Lambda, (i,j) \in A$.

The problem of the maximum flow in parametric static network $\bar{G} = (N, A, l, \bar{u})$ is to compute all maximum flows for every possible value of λ in I :

$$\sum_j \bar{f}(i,j;\lambda) - \sum_k \bar{f}(k,i;\lambda) = \begin{cases} \bar{v}(\lambda), & \text{if i=1} \qquad (3.2.a) \\ 0, & \text{if } i \neq 1, n \qquad (3.2.b) \\ -\bar{v}(\lambda), & \text{if i=n} \qquad (3.2.c) \end{cases}$$

$$l(i,j) \leq \bar{f}(i,j;\lambda) \leq \bar{u}(i,j;\lambda), (i,j) \in A \ , \tag{3.3}$$

$$max\ \bar{v}(\lambda) \tag{3.4}$$

For the maximum flow problem in parametric static network $\bar{G} = (N, A, l, \bar{u})$, the subintervals $I_k = [\lambda_k, \lambda_{k+1}], k = 0, 1, ..., K$ of the parameter λ values can be determined such as a minimum 1-n cut in the nonparametric static network $G_k = (N, A, l, u_k), u_k(i,j) = \bar{u}(i,j;\lambda_k)$, remains the minimum 1-n cut for all $\lambda \in I_k$. A parametric 1-n cut in parametric static network $\bar{G} = (N, A, l, \bar{u})$ can be defined as a finite set of cuts $[S_k, T_k], k = 0, 1, ..., K$ together with a partitioning of the interval I in disjoints subintervals $I_k, k = 0, 1, ..., K$, such that $I = I_0 \cup I_1 \cup ... \cup I_K$. The $[S_k, T_k]$ is denoted by $[S_k; I_k]$ for each $k, k = 0, 1, ..., K$ The capacity of $[S_k; I_k]$ is defined as:

$$\bar{c}[S_k; I_k] = \sum_{(S_k,T_k)} \bar{u}(i,j;\lambda) - \sum_{(T_k,S_k)} l(i,j), k = 0, 1, ..., K \tag{3.5}$$

A parametric minimum 1-n cut in network G_k is denoted by $[S_k^*; I_k], k = 0, 1, ..., K$. For a given flow $\bar{f}$ in the parametric network $\bar{G} = (N, A, l, \bar{u})$ the parametric residual capacity $\bar{r}(i,j;\lambda)$, $(i,j) \in A$ is given by:

$$\bar{r}(i,j;\lambda) = \bar{u}(i,j;\lambda) - \bar{f}(i,j;\lambda) + \bar{f}(j,i;\lambda) - l(j,i), \tag{3.6}$$

for $\lambda \in I_k, k = 0, 1, ..., K$.

For a flow $\bar{f}$ in the parametric static network $\bar{G}$, we define the set $\bar{s}(i,j) = \{\lambda | \bar{r}(i,j;\lambda) \ > 0\}$, $(i,j) \in A$. The static network $\tilde{\bar{G}} = (N, \tilde{A},\ \bar{r})$, with $\tilde{A} = \{(i,j) | (i,j) \in A, \bar{s}(i,j) \neq \phi\}$ is called the parametric residual static network. If

$(i,j) \in A$ and $(i,j) \notin \tilde{A}$, then $\bar{s}(i,j) = \phi$. Let $\tilde{\bar{P}}$ be a directed path from the source node 1 to the sink node n in the parametric residual static network $\tilde{\bar{G}}$. If $\tilde{\bar{P}}$ verifies the restriction:

$$\bar{s}(\tilde{\bar{P}}) = \bigcap_{\tilde{\bar{P}}} \bar{s}(i,j) \neq \phi \tag{3.7}$$

then $\tilde{\bar{P}}$ is called a conditional augmenting directed path. The parametric residual capacity of a conditional augmenting directed path $\tilde{\bar{P}}$ is

$$\bar{r}(\tilde{\bar{P}};\lambda) = min\{\bar{r}(i,j;\lambda) | (i,j) \in \tilde{\bar{P}}, \lambda \in \bar{s}(\tilde{\bar{P}})\}.$$

From paper [8] we have the theorem

Theorem 3.1. *A flow $\bar{f}$ is a maximum flow in parametric static network $\bar{G}$ if and only if the parametric residual static network $\tilde{\bar{G}}$ contains no conditional augmenting directed path $\tilde{\bar{P}}$.*

If residual static network $\tilde{\bar{G}}$ contains no conditional augmenting path $\tilde{\bar{P}}$, then the maximum flow in parametric network $\bar{G}$ is computed as:

$$\bar{f}(i,j;\lambda) = l(i,j) + max\{\bar{u}(i,j;\lambda) - \bar{r}(i,j;\lambda) - l(i,j), 0\} \tag{3.8}$$

The first phase of finding a maximum flow in network $\bar{G}$ consist in establishing a feasible flow, if one exists, in the nonparametric network $\bar{G} = (N, A, l, \hat{u})$ with $\hat{u}(i,j) = u_0(i,j)$ for $U(i,j) \geq 0$ and $\hat{u}(i,j) = u_0(i,j) + \Lambda \cdot U(i,j)$ for $U(i,j) < 0$. After a nonparametric feasible flow $\hat{f}$ (see [1]) is obtained, we compute the parametric residual network $\tilde{\bar{G}}_0$ for this flow $\hat{f}$.

The parametric residual capacities in $\tilde{\bar{G}}_0$ can be written as $\bar{r_0}(i,j;\lambda) = \alpha_0(i,j) + \lambda\beta_0(i,j)$, where $\alpha_0(i,j) = u_0(i,j) - \hat{f}(i,j) + \hat{f}(j,i) - l(j,i)$ and $\beta_0(i,j) = U(i,j)$.

The second phase of the algorithm starts with the parametric residual network $\tilde{\bar{G}}_0$, $\lambda_0 = 0$ and $I_0 = [0, \Lambda]$.

The algorithm for computing the maximum flow in a parametric static network with lower bounds (the algorithm MFPSNLB) is presented in Figure 3.1.

Algorithm MFPSNLB;
BEGIN
  compute a feasible flow f_0 in network G_0;
  compute the parametric residual network $\tilde{\bar{G}}_0$;
  B:={0}; $k := 0; \lambda_k := 0$;
  REPEAT
    SADP(k, λ_k, B);
    k:=k+1;
  UNTIL($\lambda_k = \Lambda$);
END.

Figure 3.1.a. The algorithm for the maximum flow in parametric static network with lower bounds

PROCEDURE SADP(k, λ_k, B);
BEGIN

compute the network $\tilde{\bar{G}}_k$;
compute the exact distance labels $\tilde{d}(i)$ in $\tilde{\bar{G}}_k$;
p=(n+1, n+1, ..., n+1); $\alpha_k(\tilde{\bar{P}}) := 0$; $\beta_k(\tilde{\bar{P}}) := 0$;
$\lambda_{k+1} := \Lambda$;i:=1;
WHILE $\tilde{d}(1) < n$ DO
  BEGIN
   IF(exists an admissible arc (i, j))
    THEN BEGIN
      p(j):=i;
      i:=j;
      IF(i=n)
        THEN BEGIN
          RCCADP(p, $\lambda_{k+1}, B, \alpha_k(\tilde{\bar{P}}), \beta_k(\tilde{\bar{P}})$);
          i:=1;
            END;
      END
    ELSE BEGIN
          $\tilde{d}(i) := min\{\tilde{d}(j) + 1|(i,j) \in \tilde{A}_k\}$;
          IF i$\neq$ s
           THEN i:= p(i);
        END;
  END;
compute the flow $\bar{f}_k$;
add λ_{k+1} to the list B;
END;

Figure 3.1.b. The Procedure Shortest Augmenting Directed Path (SADP)

PROCEDURE RCCADP(p, λ_{k+1}, B, $\alpha_k(\tilde{\bar{P}}), \beta_k(\tilde{\bar{P}})$);
BEGIN
compute $\tilde{\bar{P}}$ based on p;
$\alpha_k(\tilde{\bar{P}}) := min\{\alpha_k(i,j)|(i,j) \in \tilde{\bar{P}}\}$;
$\beta_k(\tilde{\bar{P}}) := min\{\beta_k(i,j)|(i,j) \in \tilde{\bar{P}}\}$;
i:=n;
WHILE i$\neq$ 1 DO;
  BEGIN
    IF($\beta_k(p(i), i) < \beta_k(\tilde{\bar{P}})$
      THEN BEGIN
        $\lambda' := \lambda_k + (\alpha_k(p(i), i) - \alpha_k(\tilde{\bar{P}}))/(\bar{\beta}_k(\tilde{\bar{P}}) - \beta_k(p(i), i))$;
        IF ($\lambda' < \lambda_{k+1}$)
          THEN $\lambda_{k+1} := \lambda'$;
          END;
    $\alpha_k(p(i), i) := \alpha_k(p(i), i) - \alpha_k(\tilde{\bar{P}})$; $\beta_k(p(i), i) := \beta_k(p(i), i) - \beta_k(\tilde{\bar{P}})$;
    $\alpha_k(i, p(i)) := \alpha_k(i, p(i)) + \alpha_k(\tilde{\bar{P}})$; $\beta_k(i, p(i)) := \beta_k(i, p(i)) + \beta_k(\tilde{\bar{P}})$;
    i:=p(i);
  END;

(19) END;

Figure 3.1.c. The Procedure Residual Capacity of Conditional Augmenting Directed Path(RCCADP).

In order to avoid working with piecewise linear functions, the algorithm MFPSNLB works in parametric residual static networks defined for subintervals of the parameter values where the parametric residual capacities of all arcs remain linear functions (and not piecewise linear). The parametric residual network $\tilde{\bar{G}}$ for a subinterval $I_k = [\lambda_k, \lambda_{k+1}]$ is denoted by $\tilde{\bar{G}}_k$. In k-th step of the algorithm MFPSNLB, the SADP procedure computes the parametric residual static network $\tilde{\bar{G}}_k$ where $\bar{r}_k(i,j;\lambda) = \alpha_k(i,j) + (\lambda - \lambda_k)\cdot\beta_k(i,j)$, with $\alpha_k(i,j) = \alpha_0(i,j) + \lambda_k\beta_0(i,j), \beta_k(i,j) = \beta_0(i,j)$ and computes the shortest augmenting directed path $\tilde{\bar{P}}$ in network $\tilde{\bar{G}}_k$. The RCCADP procedure computes $\bar{r}_k(\tilde{\bar{P}},\lambda), \lambda_{k+1}$ and alters $\alpha_k(i,p(i))$, $\beta_k(i,p(i))$, $\alpha_k(p(i),i), \beta_k(p(i),i)$. It is obvious that in $\tilde{\bar{G}}_k$ we have $\bar{r}_k(\tilde{\bar{P}},\lambda) = \alpha_k(\tilde{\bar{P}}) + (\lambda-\lambda_k)\cdot\beta_k(\tilde{\bar{P}})$.

Theorem 3.2 (Theorem of Correctness). *The algorithm MFPSNLB computes correctly a maximum flow in parametric static network with lower bounds* $\bar{G} = (N, A, l, \bar{u})$ *and* $\lambda \in I = [0, \Lambda]$.

Theorem 3.3 (Theorem of Complexity). *The algorithm MFPSNLB runs in O(Kn^2m) time, where K+1 is the number of* λ *value in the set B at the end of the algorithm.*

We remark the fact that the problem of the maximum flow in nonparametric static network can be solved in O(nm) time (see 7). In this case the algorithm MFPSNLB runs in O(Knm).

4. The maximum flow in parametric dynamic networks.

A dynamic network $G_h = (N, A, h, l_h, u_h)$ for which the upper bounds (capacities) $u_h(i,j;t)$ of some arcs $(i,j) \in A$ are functions of a real parameter $\lambda \in I = [0, \Lambda]$ is referred to as a parametric dynamic network and is denoted by $\bar{G}_h = (N, A, h, l_h, \bar{u}_h)$. The parametric upper bound function $\bar{u}_h : A \times H \times I \to \mathbf{R}^+$ is defined by relation

$$\bar{u}_h(i,j;t;\lambda) = u_{0h}(i,j;t) + \lambda U_h(i,j;t), (i,j) \in A, t \in H, \lambda \in I, \qquad (4.1)$$

The parametric part $U_h(i,j;t)$ must satisfy the constraint:

$$U_h(i,j;t) \geq (l_h(i,j;t) - u_{0h}(i,j;t)/\Lambda, (i,j) \in A, t \in H.$$

The problem of the maximum flow in parametric dynamic network $G_h = (N, A, h, l_h, \bar{u}_h)$ is to compute a flow function $\bar{f}_h : A \times H \times I \to \mathbf{R}^+$ that satisfies the following constraints:

$$\sum_{t=0}^{T}(\sum_j \bar{f}(i,j;t;\lambda) - \sum_k \sum_\tau \bar{f}_h(k,i;\tau;\lambda) = \bar{v}_H(\lambda), \text{ i=1} \qquad (4.2.a)$$

$$\sum_j \bar{f}(i,j;t;\lambda) - \sum_k \sum_\tau \bar{f}_h(k,i;\tau;\lambda) = 0, i \neq 1, n, t \in H \qquad (4.2.b)$$

$$\sum_{t=0}^{T}(\sum_j \bar{f}(i,j;t;\lambda) - \sum_k \sum_\tau \bar{f}_h(k,i;\tau;\lambda) = -\bar{v}_H(\lambda), \text{ i=n} \qquad (4.2.c)$$

$$l_h(i,j;t) \leq \bar{f}_h(i,j;t;\lambda) \leq \bar{u}_h(i,j;t;\lambda), (i,j) \in A, t \in H, \lambda \in I \qquad (4.3)$$

$\max \bar{v}_H(\lambda), \lambda \in I$ (4.4)
where $\bar{f}_h(i,j;t;\lambda) = 0, (i,j) \in A, t \in \{T - h(i,j;t) + 1, ..., T\}, \lambda \in I$.

In network $\bar{G}_h = (N, A, h, l_h \bar{u}_h)$ we consider the following assumption: if $(i,j) \in A$ then $(j,i) \in A$. This assumption is non-restrictive because if $(i,j) \in A$ and $(j,i) \notin A$ we consider that $(j,i) \in A$ with $l_h(j,i;\theta) = u_h(j,i;\theta;\lambda) = 0, \theta = t + h(i,j;t), t \in H, \lambda \in I, h(j,i;\theta) = -h(i,j;t)$, if $0 \le t \le T - h(i,j;t)$ and $h(j,i;\theta) = \infty$, if $T - h(i,j;t) + 1 \le t \le T$.

The parametric dynamic residual capacities with respect to a given flow $\bar{f}_h$ are defined as follow:

$\bar{r_h}(i,j;t;\lambda) = \bar{u}_h(i,j;t;\lambda) - \bar{f}_h(i,j;t;\lambda) + \bar{f}_h(j,i;\theta;\lambda) - l_h(j,i;\theta),$ (4.5)
$(i,j) \in A, t \in H, \lambda \in I$

For the maximum flow in parametric dynamic network $\bar{G}_h = (N, A, h, l_h, \bar{u}_h)$, the parametric dynamic residual network with respect to a given flow $\bar{f}_h$ is defined as $\tilde{\bar{G}}_h = (N, \tilde{A}, \bar{r}_h)$, where $\tilde{A} = \{(i,j) | (i,j) \in A, \bar{r}_h(i,j;t;\lambda)\, 0, t \in H, \lambda \in I\}$.

In this paper, the proposed approach consists in applying the algorithm MFP-SNLB presented in Section 3 in parametric static reduced expanded network $\bar{G}' = (N', A', l', \bar{u'})$ which is constructed similar with the construction of the network $G' = (N', A', l', u')$ presented in Section 2.

The algorithm for the maximum flow in parametric dynamic network with lower bounds (the algorithm MFPDNLB) is presented in Figure 4.1.

(1) ALGORITHM MFPDNLB;
(2) BEGIN
(3) construct the network $\bar{G}'$;
(4) apply the algorithm MFPDNLB in network $\bar{G}'$;
(5)END.

Figure 4.1. The algorithm for maximum flow in parametric dynamic network with lower bounds.

Theorem 4.1 (Theorem of Correctness). *The algorithm MFPDNLB computes correctly a maximum flow in parametric dynamic network with lower bounds* $\bar{G}_h = (N, A, h, l_h, \bar{u}_h)$ *and* $\lambda \in I$.

Proof. This theorem results from the fact that the maximum flow in parametric dynamic network with lower bounds $\bar{G}_h = (N, A, h, l_h, \bar{u}_h)$ is equivalent with the maximum flow in parametric static network $\bar{G}' = (N', A', l', \bar{u}')$ and Theorem 3.2. □

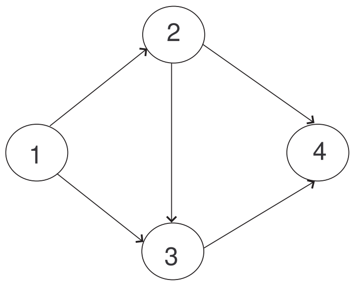
Theorem 4.2 (Theorem of Complexity). *The algorithm MFPDNLB runs in* $O(KT^3n^2m)$ *time, where K+1 is the number for* λ *values at the end of the algorithm.*

Proof. From Theorem 3.3 we have that the algorithm MFPDNLB runs in $O(K(n'_H)^2 m'_H)$ time. From Section 2 we obtain $n' = O(nT)$ and $m' = O(mT)$. Therefore we obtain that algorithm MFPDNLB runs in $O(KT^3n^2m)$ time. □

In accordance with the remark at Theorem 3.3 we note that the algoritm MF-PDNLB run in $O(KnmT^2)$ time.

5. Example.

The parametric dynamic network is presented in Figure 5.1. with the time horizon set to T=3, therefore H=$\{0, 1, 2, 3\}$. The transit times h(i,j;t), the dynamic lower bounds $l_h(i, j; t)$ and the parametric dynamic upper bounds (capacities) $\bar{u}_h(i, j; t; \lambda) = u_{0h}(i, j; t) + \lambda U_h(i, j; t)$ for all arcs in $\bar{G}_h$ are indicated in Figure 5.1.b. The interval of parameter λ values is set to [0,1], i.e., Λ=1.



(a)

(i,j)	$h(i,j;t)$	$l_h(i,j;t)$	$u_{0h}(i,j;t)$	$U_h(i,j;t)$
$(1,2)$	$1, t=0$ $2, t=1,2,3$	$1, t=0$ $0, t=1,2,3$	$4, t=0,1,2,3$	$4, t=0,1,2,3$
$(1,3)$	$1, t=0,1$ $2, t=2,3$	$2, t=0,1$ $0, t=2,3$	$9, t=0,1$ $6, t=2,3$	$-4, t=0,1,2,3$
$(2,3)$	$1, t=0,1,2,3$	$0, t=0,1,2,3$	$3, t=0,1,2,3$	$-2, t=0,1,2,3$
$(2,4)$	$1, t=0,1$ $2, t=2,3$	$0, t=0,1,2,3$	$4, t=0,1,2,3$	$2, t=0,1,2,3$
$(3,4)$	$2, t=0,1$ $1, t=2,3$	$0, t=0,3$ $1, t=1,2$	$8, t=0,1,2,3$	$0, t=0,1$ $-2, t=2,3$

(b)

Figure 5.1. The parametric dynamic network $\bar{G}_h$.

The support graph for parametric superextended network $\bar{G}_H^*$ is presented in Figure 5.2.

The support graph for parametric reduced expanded network $\bar{G}'$ is showed in Figure 5.3.

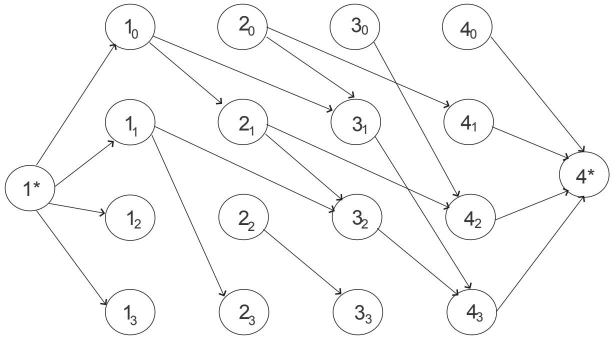


Figure 5.2. The support graph for parametric superextended network $\bar{G}^*_H$.

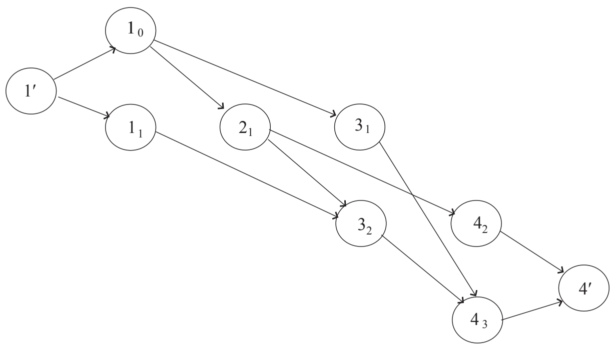


Figure 5.3. The support graph for parametric reduced expanded network $\bar{G}'$.

The lower bounds $l'(i_t, j_\theta)$ and the parametric upper bounds $\bar{u'}(i_t, j_\theta; \lambda) = u'_0(i_t, j_\theta) + \lambda U'(i_t, j_\theta)$ for all arcs in $\bar{G}'$ are indicated in table from Figure 5.4.

(i_t, j_θ)	$l'(i_t, j_\theta)$	$u'_0(i_t, j_\theta)$	$U'(i_t, j_\theta)$	$\hat{f}'(i_t, j_\theta)$
$(1', 1_0)$	0	∞	0	3
$(1', 1_1)$	0	∞	0	2
$(1_0, 2_1)$	1	4	4	1
$(1_0, 3_1)$	2	9	-4	2
$(1_1, 3_2)$	2	9	-4	2
$(2_1, 3_2)$	0	3	-2	0
$(2_1, 4_2)$	0	4	2	1
$(3_1, 4_3)$	1	8	0	2
$(3_2, 4_3)$	1	8	-2	2
$(4_2, 4')$	0	∞	0	1
$(4_3, 4')$	0	∞	0	4

Figure 5.4. The $l'(i_t, j_\theta), \bar{u}'(i_t, j_\theta; \lambda) = u'_0(i_t, j_\theta) + \lambda U'(i_t, j_\theta)$ and $\hat{f}'(i_t, j_\theta)$ in network $\bar{G}'$.

In the first phase we determine in static residual network $\tilde{\hat{G}}'$: $\tilde{\hat{P}}'_1 = (1', 1_0, 2_1, 4_2, 4')$, $\hat{r}'(\tilde{\hat{P}}'_1) = 1$; $\tilde{\hat{P}}'_2 = (1', 1_0, 3_1, 4_3, 4'), \hat{r}'(\tilde{\hat{P}}'_2) = 2$; $\tilde{\hat{P}}'_3 = (1', 1_1, 3_2, 4_3, 4'), \hat{r}'(\tilde{\hat{P}}'_3) = 2$. The feasible flow $\hat{f}'$ in network $\hat{G}'$ is presented in table from Figure 5.4.

In the second phase, by applying algorithm MFPSNLB in network $\bar{G}'$, we obtain in the parametric static residual network $\tilde{\bar{G}}'$ the following directed paths: $\tilde{\bar{P}}'_1 = (1', 1_0, 2_1, 4_2, 4')$, $\tilde{\bar{P}}'_2 = (1', 1_0, 3_1, 4_3, 4')$, $\tilde{\bar{P}}'_3 = (1', 1_1, 3_2, 4_3, 4')$ in $\tilde{\bar{G}}'_0$,$\tilde{\bar{G}}_1$ and $\tilde{\bar{P}}'_1$, $\tilde{\bar{P}}'_2$, $\tilde{\bar{P}}'_3$, $\tilde{\bar{P}}'_4 = (1', 1_0, 3_1, 4_3, 4')$ in $\tilde{\bar{G}}'_2$. The results of this example are synthetically presented in table from Figure 5.5. We notice the fact that $\tilde{\hat{P}}'_i = \tilde{\bar{P}}'_i, i = 1, 2, 3, 4$. The flow $\bar{f_k}'$ is obtain by $\bar{f_k}'(i_t, j_\theta; \lambda) = \bar{r_k}'(i_t, j_\theta; \lambda) + \hat{f}'(i_t, j_\theta; \lambda), (i_t, j_\theta) \in A, \lambda \in I_k$. The $\bar{r_k}'(i_t, j_\theta; \lambda)$ are obtained from $\bar{r_k}'(\tilde{\bar{P}}_i; \lambda)$. The graphic of $\bar{v}'(\lambda)$ is presented Figure 5.6.

k	λ_k	$\tilde{\bar{P}}'_i$	$\bar{r}_k(\tilde{\bar{P}}'_i;\lambda)$	λ_{k+1}	$\hat{r}'(\tilde{\hat{P}}'_i)$	$\bar{v}'(\lambda)$
0	0	$\tilde{\bar{P}}'_1$	$3+2\lambda$	1	1	20
		$\tilde{\bar{P}}'_2$	6	1/4	2	
		$\tilde{\bar{P}}'_3$	$6-2\lambda$	1/4	2	
1	1/4	$\tilde{\bar{P}}'_1$	$3+2\lambda$	1	1	$21-4\lambda$
		$\tilde{\bar{P}}'_2$	$7-4\lambda$	1	2	
		$\tilde{\bar{P}}'_3$	$6-2\lambda$	1/2	2	
2	1/2	$\tilde{\bar{P}}'_1$	$3+2\lambda$	1	1	$21-4\lambda$
		$\tilde{\bar{P}}'_2$	$7-2\lambda$	1	2	
		$\tilde{\bar{P}}'_3$	$7-2\lambda$	1	2	
		$\tilde{\bar{P}}'_4$	1	0	$2\lambda-1$	

Figure 5.5. Results of applying the algorithm in $\bar{G}'$.

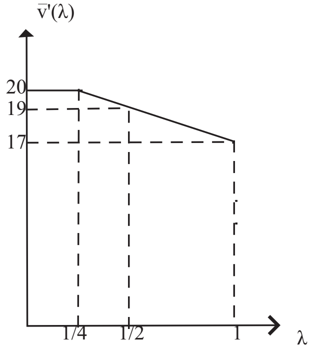


Figure 5.6. The graphic of $\bar{v}'(\lambda)$.

6. Conclusions

In this paper, we have presented an original version of the general problem of the maximum flows in parametric dynamic networks, namely the one with nonzero lower bounds, a model that is more closely related to real problems. The algorithm which has been developed proved that the parametric flow in dynamic networks can be more conveniently addressed in a static way by transforming dynamic networks into related expanded ones. An example is also given to support this type of approaches.

The question arises whether the approach used in this paper is also adequate for the parametric problem of the minimum flow in dynamic networks or this new problem requires a more suitable approach. Another problem requesting a response is related to the efficiency of the addressed algorithm in relation with the complexity of a genuine dynamic approach. These problems will orient the future work of the authors.

References

[1] R. Ahuja, T. Magnanti, J.B.Orlin, *Network Flows. Theory, algorithms and applications*, Prentice Hall, Engleewood Clifss, New Jersey, 1993.

[2] J.A. Aronson, A survey of dynamic networks flows, *Annals of Operations Research* **20** (1989), 1–66.

[3] X. Cai, D. Sha, C. Wong, *Time-varying Network Optimization*, Springer (2007).

[4] E. Ciurea, Second best temporally repeated flows, *The Korean Journal of Computational and Applied Mathematics* **9** (2002), no. 1, 77–86.

[5] H. Hamacher, L. Foulds, Algorithms for Flows with Parametric Capacities, *ZOR-Methods and Models of Operations Research* **3**(1989), 21–37.

[6] E. Nasrabadi, S. Hashemi, Minimum cost time-varying flow problems, *Optimization Methods and Software* **25** (2010), no. 3, 429–447.

[7] J.B. Orlin, Max Flows in O(nm) time, or better, *Proceseeding of the Annual ACM Symposium on Theory of Computing*, ACM Press (2013), New York, 765–774.

[8] M. Parpalea, E. Ciurea, Partitioning algorithm for the parametric maximum flow, *Applied Mathematics* **4**, Special Issue on Computer Mathematics (2013), no. 10A, 3–10.

[9] G. Ruhe, *Algorithmic Aspects of Flows in Networks*, Kluwer Academic Publisher, Dordrecht, 1991.

[10] M. Skutella, *An introduction to Network Flows Over Time*, Research Trends in Combinatorical Optimization (2008), Springer, 451–482.

(Nicoleta Avesalon (Grigoras)) Department of Mathematics and Computer Science,
Transilvania University of Brasov
50 Iuliu Maniu, Brasov, 50091, Romania
E-mail address: nicole.grigoras@gmail.com