

Minimum Flows in Parametric Dynamic Networks - the Static Approach

Grigoraş (Avesalon) Nicoleta

Transilvania University of Braşov, 50091 Braşov, Iuliu Maniu, 50, Romania

E-mail: nicole.grigoras@gmail.com

Keywords: dynamic network, parametric network, minimum flow

Received: October 15, 2018

The problems of flows in parametric networks extend the classical problems of optimal flow to some special kind of networks where capacities of certain arcs are not constants but depending on several parameters. Consequently, these problems consist of solving a range of ordinary (nonparametric) optimal flow problems for all the parameter values within certain sub-intervals of the parameter values. Although classical network flow models have been widely used as valuable tools for many applications [1], they fail to capture the essential property of the dynamic aspect of many real-life problems, such as traffic planning, production and distribution systems, communication systems, evacuation planning, etc. In all these cases, time is an essential component, either because the flows take time to pass from one location to another, or because the structure of the network changes over time. Accordingly, the dynamic flow models seem suited to catch and describe different real-life dynamic problems such as network-structure changing over time or timely decision-making, but, because of their complexity, these models have not been as thoroughly investigated as those of classical flows.

This article presents and solves the problem of the minimum flows in a parametric dynamic network. The proposed approach consists in applying a parametric flow algorithm in the reduced expanded network which is obtained by expanding the original dynamic network. A numerical example is also presented for a better understanding of the used approach.

Povzetek: V članku je predstavljena metoda za rešitev problema najmanjšega pretoka v parametričnih dinamičnih mrežah.

1 Introduction

The parametric maximum flow problem, as well as that of the related minimum flow one represent generalizations of ordinary problems for the maximum, respectively minimum flow in which the upper/lower bounds of some arcs depend on a single parameter, being monotonically increasing (or decreasing) functions of the parameter. For the parametric maximum flow problem with zero lower bounds and linear capacity functions of a single parameter, G.Ruhe proposed [16] an original 'piece-by-piece' approach while papers [11] and [13] solve the problems of parametric minimum, respectively maximum flows via an partitioning approach. Finally, the same partitioning approach is extended to a discrete time dynamic network in paper [3]. This class of problems is known as the flow in parametric networks problem. Beside the applications of the ordinary, nonparametric flow problems, the applications of those of flows in parametric networks may include: multiprocessor scheduling with release times and deadlines, integer programming problems, computing sub-graph density and network vulnerability and partitioning a data base between fast and slow memory, product selection, flow sharing, database record segmentation in large shared databases, optimizing field repair kits, etc.[15]

Besides this, the static network flow models arise in a num-

ber of combinatorial applications that on the surface might not appear to be maximum flow problems at all. The problem also arises directly in problems as far reaching as machine scheduling, the assignment of computer modules to computer processors, tanker scheduling etc. [1]. However, in some other applications, time is an essential ingredient, whether it involves problems like maximum or minimum flows in time-varying networks [2], [3], dynamic flows of minimum cost [12], general problems of temporally repeated flows [6] such as earliest arrival flow [17] or dynamic resource allocation problem for large-scale transportation network evacuation [7]. Starting from the complexities of the classic algorithms, an important series of other papers concern with the complexities of dynamic network algorithms. Cai et al. [4] proved that the complexity of finding a shortest (quickest) dynamic flow augmenting path, by exploring the forward and reverse arcs successively, is $O(nmT^2)$. For algorithms which explores the two sub-networks (the forward sub-network, consisting of the set of direct arcs and the reverse sub-network consisting of the set of reverse arcs simultaneously, Miller-Hooks and Patterson [8] also reported a complexity of $O(n^2T^2)$. By using special node addition and selection procedures, Nasrabadi and Hashemi [9] succeeded to reduce significantly the number of node time pair that needs to be visited. The worst-case complexity of their algorithm is

$O(nT(n + T))$. Finally, Orlyn reported [10] an extremely good running time of $O(nm)$ for the problem of maximum flow. In this case we need to use dynamic network flow models. Due to the powerful versatility of dynamic flow algorithms, it is not surprising that these algorithms are often more difficult to design than their static counterparts [10] but still they are also very challenging problems. The approaches for solving the minimum parametric flow over time problem via applying classical algorithms can be grouped in two main categories:

i) by applying a non-parametric minimum dynamic flow algorithm [12] in dynamic residual networks generated by partitioning the interval of the parameter values [9], [14]. This first category approach was also used [3] for finding a maximum parametric flow in discrete-time dynamic networks;

ii) by applying a static (classical sequential or parallel [5]) parametric flow algorithm for the maximum [13], [16] or for the minimum [11] flow in a (static network) reduced expanded network, which is obtained by expanding the original dynamic network.

In this paper, the case of the minimum flows in parametric dynamic networks is considered. The proposed approach consists in transformation of the problem of minimum flow in parametric dynamic network into that of the minimum flow in parametric static network. This problem generalizes the problems of flow in dynamic network and of flow in parametric static networks through the following assumptions: (1) the dynamic network and corresponding expanded static network are with lower bounds, (2) we address the minimum flow problem on a dynamic network with time varying transit times as well as time varying lower and upper bounds on arc i.e. the dynamic network is not stationary.

The remainder of this paper is organized as follows. In Section 2 some dynamic network notations and terminology are presented. Then, in Section 3 we expose, the minimum flow in parametric static network, while in Section 4 the minimum flow in parametric dynamic network problem is presented. In Section 5 an example is given. In the presentation to follow, some familiarity with flow problems is assumed and many details are omitted.

2 The minimum flows in dynamic networks.

Let $G = (N, A, l, u)$ be a static network with the nod set $N = \{1, \dots, i, \dots, j, \dots, n\}$, the arc set $A = \{a_1, \dots, a_k, \dots, a_m\}$, $a_k = (i, j)$, the lower bound function $l : A \rightarrow \mathbf{R}^+$ and the upper bound (capacity) function $u : A \rightarrow \mathbf{R}^+$, where \mathbf{R} is the real number set. To define the minimal 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 natural number set and let $H = \{0, 1, \dots, T\}$ be the set of periods, where T is a finite time horizon, $T \in$

\mathbf{N} . Let $h : A \times H \rightarrow \mathbf{N}$ be the transit time function, $l_h : A \times H \rightarrow \mathbf{R}^+$ the time lower bound function and $u_h : A \times H \rightarrow \mathbf{R}^+$ the time upper bound function. For each arc $(i, j) \in A$ the $h(i, j; t)$ 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 starting from node i at time t and arriving at node j at time $\theta = t + h(i, j; t)$ for all arcs (i, j) and all t . The minimal dynamic flow problem for T time periods is to determine a flow function $f_h : A \times H \rightarrow \mathbf{N}$, which should satisfy the following conditions in dynamic network $G_h = (N, A, h, l_h, u_h)$:

$$\sum_{i=1}^T (\sum_j f_h(i, j; t) - \sum_k \sum_\tau f_h(k, i; \tau)) = v_H, \tag{2.1.a}$$

$$\sum_j f_h(i, j; t) - \sum_k \sum_\tau f_h(k, i; \tau) = 0, \tag{2.1.b}$$

$i \neq 1, n, t \in H,$

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

$$l_h(i, j; t) \leq f_h(i, j; t) \leq u_h(i, j; t), \tag{2.2}$$

$\forall (i, j) \in A \text{ and } \forall t \in H$

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

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

Obviously, the problem of finding a minimum dynamic flow is more complex than the problem of finding a minimum static flow. Fortunately, this complication can be solved by rephrasing the dynamic flow problem into a static flow problem in a static network $G' = (N', A', l', u')$ called reduced expanded network. First, we form the expanded network $G_H = (N_H, A_H, l_H, u_H)$ with $N_H = \{i_t | i \in N, t \in H\}$, $A_H = \{(i_t, j_\theta) | (i, j) \in A, t = 0, 1, \dots, T - h(i, j; t)\}$, $l_H(i_t, j_\theta) = l_h(i, j; t)$, $u_H(i_t, j_\theta) = u_h(i, j; t)$, $(i_t, j_\theta) \in A_H$.

We have $|N_H| = n(T + 1)$ and $|A_H| \leq m(T + 1) - \sum_A \bar{h}(i, j)$, where $\bar{h}(i, j) = \min\{h(i, j; 0), \dots, h(i, j; T)\}$. 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 $1_0, 1_1, \dots, 1_T$ to the sink nodes n_0, n_1, \dots, n_T in static network G_H and vice versa [2]. 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^* building 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 $h^* : A_H^* \rightarrow \mathbf{N}$, $h^*(1^*, 1_t) = h^*(n_t, n^*) = 0$, $t \in H$, $h^*(i_t, j_\theta) = h(i, j; t)$, $(i_t, j_\theta) \in A_H$. 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 sub-network 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 always acyclic.

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:

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

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

$$\min v', (2.6)$$

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], [17].

3 The minimum flow in parametric static networks

A natural generalization of the minimum flow problem in static networks can be obtained by making the lower bounds of some arcs function of a single parameter. Since the minimum flow value function in a parametric network is a continuous piecewise linear function of the parameter, the parametric minimum flow problem can alternately be

defined as to find all the breakpoints and their corresponding minimum flow and maximum cuts. The approach presented in this section is presented in [11], the first approach for minimum flow in parametric static network.

A static network $G = (N, A, l, u)$ with the lower bounds $l(i, j)$ 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 $\bar{G} = (N, A, \bar{l}, u)$. The upper bound function $\bar{l} : A \times \mathbf{R}^+ \rightarrow \mathbf{R}^+$ is defined by the relation:

$$\bar{l}(i, j; \lambda) = l_0(i, j) + \lambda \cdot L(i, j), \lambda \in [0, \Lambda] = I (3.1)$$

where $L : A \rightarrow \mathbf{R}$ is the parametric part of the upper bound function \bar{l} and $l_0 : A \rightarrow \mathbf{R}^+$ is the non parametric part of the function \bar{l} with $\bar{l}(i, j; 0) = l_0(i, j)$, $(i, j) \in A$. The $L(i, j)$ and $l_0(i, j)$ must satisfy $-l_0(i, j)/\Lambda \leq L(i, j) \leq (u(i, j) - l_0(i, j))/\Lambda$ and $0 \leq l_0(i, j) \leq u(i, j)$, $(i, j) \in A$.

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

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

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

$$\min \bar{v}(\lambda) (3.4)$$

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

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

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

$$\bar{r}(i, j; \lambda) = u(j, i) - \bar{f}(j, i; \lambda) + \bar{f}(i, j; \lambda) - \bar{l}(i, j; \lambda), \lambda \in I_k, k = 0, 1, \dots, K (3.6)$$

For a flow \bar{f} in 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 $\tilde{G} = (N, \tilde{A}, \bar{r})$, with $\tilde{A} = \{(i, j) | (i, j) \in A, \bar{s}(i, j) \neq \phi\}$ is named the parametric residual static network. If $(i, j) \in A$ and $(i, j) \notin \tilde{A}$, then $\bar{s}(i, j) = \phi$.

Let \tilde{P} be a directed path from the source node 1 to the sink node n in the parametric residual static network \tilde{G} . If \tilde{P} verifies the restriction:

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

then \tilde{P} is named conditional decreasing directed path. The parametric residual capacity of a conditional decreasing directed path \tilde{P} is $\bar{r}(\tilde{P}; \lambda) = \min\{\bar{r}(i, j; \lambda) | (i, j) \in \tilde{P}, \lambda \in \bar{s}(\tilde{P})\}$. From paper [11] we have the theorem:

Theorem 1 (11). *A flow \bar{f} is a minimum flow in parametric static network \tilde{G} if and only if the parametric residual static network \tilde{G} contains no conditional decreasing directed path \tilde{P} .*

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

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

The first phase of finding a minimum flow in network \tilde{G} consists in establishing a feasible flow, if one exists, in non-parametric network $\hat{G} = (N, A, \hat{l}, u)$ with $\hat{l}(i, j) = l_0(i, j)$ for $L(i, j) \leq 0$ and $\hat{l}(i, j) = l_0(i, j) + \Lambda \cdot L(i, j)$ for $L(i, j) > 0$. After a nonparametric feasible flow \hat{f} (see [1]) we compute the parametric residual network \tilde{G}_0 for this flow \hat{f} .

The parametric residual capacities in \tilde{G}_0 can be written as $\bar{r}_0(i, j; \lambda) = \alpha_0(i, j) + \lambda\beta_0(i, j)$, where $\alpha_0(i, j) = u(j, i) - \hat{f}(j, i) + \hat{f}(i, j) - l_0(i, j)$ and $\beta_0(i, j) = L(i, j), \lambda \in I_0 = [0, \lambda_1]$.

The second phase of the algorithm starts with the parametric residual network $\tilde{G}_0, \lambda_0 = 0$ and $I_0 = [0, \lambda_1]$.

The algorithm for minimum flow in a parametric static network (the algorithm MFPSN) is presented in Figure 3.1.

```
(01) Algorithm MFPSN;
(02) BEGIN
(03)   compute a feasible flow  $f_0$  in network  $G_0$ ;
(04)   compute parametric residual network  $\tilde{G}_0$ ;
(05)   B:={0}; k := 0;  $\lambda_k := 0$ ;
(06)   REPEAT
(07)     SDDP( k,  $\lambda_k$ , B);
(08)     k:=k+1;
(09)   UNTIL ( $\lambda_k = \Lambda$ );
(10) END.
```

Figure 3.1.a. The algorithm for the minimum flow in parametric static network

```
(01) PROCEDURE SDDP( k,  $\lambda_k$ , B);
(02) BEGIN
(03)   compute the network  $\tilde{G}_k$ ;
(04)   compute exact distance labels  $\tilde{d}(i)$  in  $\tilde{G}_k$ ;
(05)   p:=( n+1, ..., n+1);  $\alpha_k(\tilde{P}) := 0$ ;  $\beta_k(\tilde{P}) := 0$ ;
(06)    $\lambda_{k+1} := \Lambda$ ; j:=n;
(07)   WHILE  $\tilde{d}(n) < n$  DO
(08)     BEGIN
(09)       IF(exists an admissible arc (i, j))
(10)         THEN BEGIN
(11)           p(i):=j;
(12)           j:=i;
(13)           IF (j=1)
(14)             THEN BEGIN
(15)               RCDADP(p,  $\lambda_{k+1}$ , B,  $\alpha_k(\tilde{P}), \beta_k(\tilde{P})$ );
(16)               j:=n;
(17)             END;
(18)           END
(19)         ELSE BEGIN
(20)            $\tilde{d}(j) := \min\{\tilde{d}(i) + 1 | (i, j) \in \tilde{A}_k\}$ ;
(21)           IF j  $\neq$  n
(22)             THEN j:= p(j);
(23)           END;
(24)         END;
(25)       compute the flow  $\bar{f}_k$ ;
(26)       add  $\lambda_{k+1}$  to the list B;
(27)     END;
```

Figure 3.1.b. The Procedure Shortest Decreasing Directed Path (SDDP)

```
(01) PROCEDURE RCDADP(p,  $\lambda_{k+1}$ , B,  $\alpha_k(\tilde{P}), \beta_k(\tilde{P})$ );
(02) BEGIN
(03)   compute  $\tilde{P}$  based on p;
(04)    $\alpha_k(\tilde{P}) := \min\{\alpha_k(i, j) | (i, j) \in \tilde{P}\}$ ;
(05)    $\beta_k(\tilde{P}) := \min\{\beta_k(i, j) | (i, j) \in \tilde{P}\}$ ;
(06)   i:=1;
(07)   WHILE i  $\neq$  n DO;
(08)     BEGIN
(09)       IF( $\beta_k(p(i), i) < \beta_k(\tilde{P})$ )
(10)         THEN BEGIN
(11)            $\lambda' := \lambda_k + (\alpha_k(i, p(i)) - \alpha_k(\tilde{P})) / (\beta_k(\tilde{P}) - \beta_k(i, p(i)))$ ;
(12)           IF ( $\lambda' < \lambda_{k+1}$ )
(13)             THEN  $\lambda_{k+1} := \lambda'$ ;
(14)           END;
(15)            $\alpha_k(i, p(i)) := \alpha_k(i, p(i)) - \alpha_k(\tilde{P})$ ;
(16)            $\beta_k(i, p(i)) := \beta_k(i, p(i)) - \beta_k(\tilde{P})$ ;
(17)            $\alpha_k(p(i), i) := \alpha_k(p(i), i) + \alpha_k(\tilde{P})$ ;
(18)            $\beta_k(p(i), i) := \beta_k(p(i), i) + \beta_k(\tilde{P})$ ;
(19)         END;
(20)       i:=p(i);
(21)     END;
```

Figure 3.1.c. The Procedure Residual Capacity of Conditional Decreasing Directed Path(RCDADP).

In order to avoid working with piecewise linear functions, the algorithm MFPSN works in parametric residual static networks defined for sub-intervals of the parameter

values where the parametric residual capacities of all arcs remain linear functions. The parametric residual network \tilde{G} for a sub-interval $I_k = [\lambda_k, \lambda_{k+1}]$ is denoted by \tilde{G}_k . In k -th step of the algorithm MFPSN, the SDDP procedure computes the parametric residual static network \tilde{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 decreasing directed path \tilde{P} in network \tilde{G}_k . The RCDADP procedure computes $\bar{r}_k(\tilde{P}, \lambda), \lambda_{k+1}$ and alters $\alpha_k(i, p(i)), \beta_k(i, p(i)), \alpha_k(p(i), i), \beta_k(p(i), i)$. Obviously in \tilde{G}_k we have $\bar{r}_k(\tilde{P}, \lambda) = \alpha_k(\tilde{P}) + (\lambda - \lambda_k) \cdot \beta_k(\tilde{P})$.

In paper [11] the following two theorems are presented.

Theorem 2 (Theorem of Correctness). *The algorithm MFPSN computes correctly a minimum flow in parametric static network $G = (N, A, \bar{l}, u)$ and $\lambda \in I = [0, \Lambda]$.*

Theorem 3 (Theorem of Complexity). *The algorithm MFPSN 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.*

For further details of this section we recommend the work [11].

We remark the fact that the maximum flow problem in the nonparametric static network can be solved in $O(nm)$ time (see [10]). In this case the algorithm MFPSN runs in $O(Knm)$ time.

4 The minimum flow in parametric dynamic networks

A dynamic network $G_h = (N, A, h, l_h, u_h)$ for which the lower bounds $l_h(i, j; t)$ 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 $\tilde{G}_h = (N, A, h, \bar{l}_h, u_h)$. The parametric lower bound function $\bar{l}_h : A \times H \times I \rightarrow \mathbf{R}^+$ is defined by the relation:

$$\bar{l}_h(i, j; t; \lambda) = l_{0h}(i, j; t) + \lambda \cdot L_h(i, j; t), \quad (i, j) \in A, t \in H, \lambda \in I, \tag{4.1}$$

The parametric part $L_h(i, j; t)$ and the $l_{0h}(i, j; t)$, must satisfy the constraints:

$$\begin{aligned} -l_{0h}(i, j; t)/\Lambda &\leq L_h(i, j; t) \\ L_h(i, j; t) &\leq (u_h(i, j; t) - l_{0h}(i, j; t))/\Lambda, \text{ and} \\ 0 &\leq l_{0h}(i, j; t) \leq u_h(i, j; t), \\ (i, j) &\in A, t \in H. \end{aligned}$$

The minimum flow problem in parametric dynamic network $G_h = (N, A, h, \bar{l}_h, u_h)$ is to compute flow function $f_h : A \times H \times I \rightarrow \mathbf{R}^+$ that satisfies the following constraints:

$$\sum_{i=0}^T (\sum_j \bar{f}(i, j; t; \lambda) - \sum_k \sum_\tau \bar{f}_h(k, i; \tau; \lambda)) = \bar{v}_H(\lambda), \quad (4.4.a)$$

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

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

$$\bar{l}_h(i, j; t; \lambda) \leq \bar{f}_h(i, j; t; \lambda) \leq u_h(i, j; t), \quad (i, j) \in A, t \in H, \lambda \in I \tag{4.3}$$

$$\min \bar{v}_H(\lambda), \lambda \in I \tag{4.4}$$

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

In network $\tilde{G}_h = (N, A, h, \bar{l}_h, u_h)$ 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 $\bar{l}_h(j, i; \theta; \lambda) = u_h(j, i; \theta) = 0, \theta = t + h(i, j; t), t \in H, \lambda \in I, h(j, i; \theta) = -h(i, j; t)$, if $0 \leq t \leq T - h(i, j; t)$ and $h(j, i; \theta) = \infty$, if $T - h(i, j; t) + 1 \leq t \leq T$.

The parametric dynamic residual capacities with respect to given flow f_h are defined as follow:

$$\begin{aligned} \bar{r}_h(i, j; t; \lambda) &= u_h(j, i; t) - \bar{f}_h(j, i; t; \lambda) + \bar{f}_h(i, j; t; \lambda) - \bar{l}_h(i, j; t; \lambda), \\ (i, j) &\in A, t \in H, \lambda \in I \end{aligned} \tag{4.5}$$

For the minimum flow in parametric dynamic network $\tilde{G}_h = (N, A, h, \bar{l}_h, u_h)$ with respect to given flow f_h is defined as $\tilde{G}_h = (N, \tilde{A}, \tilde{r}_h)$, 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 MFPSN presented in Section 3 in parametric static reduced expanded network $\tilde{G}' = (N', A', \bar{l}', u')$ which is constructed similar with construction of the network $G' = (N', A', l', u')$ presented in Section 2.

The algorithm for the minimum flow in parametric dynamic network (the algorithm MFDPN) is presented in Figure 4.1.

- (1) ALGORITHM MFDPN;
- (2) BEGIN
- (3) construct the network \tilde{G}' ;
- (4) apply the algorithm MFPSN in network \tilde{G}' ;
- (5) END.

Figure 4.1. The algorithm for minimum flow in parametric dynamic network.

Theorem 4 (Theorem of Correctness). *The algorithm MFDPN computes correctly a minimum flow in parametric dynamic network $G_h = (N, A, h, \bar{l}_h, u_h)$ and $\lambda \in I$.*

Proof. This theorem results from the fact that the minimum flow in parametric dynamic network with lower bounds $\tilde{G}_h = (N, A, h, \bar{l}_h, u_h)$ is equivalent with the maximum flow in parametric static network $\tilde{G}' = (N', A', l', u')$ and Theorem 2. □

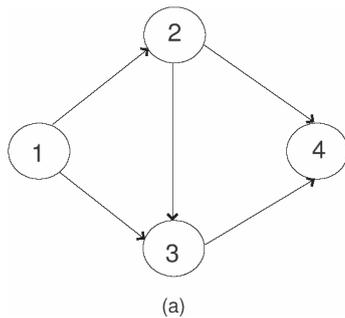
Theorem 5 (Theorem of Complexity). *The algorithm MF-PDN runs in $O(KT^3n^2m)$ time, where $K+1$ is the number for λ values in the set B at the end of the algorithm.*

Proof. From Theorem 3 results that the algorithm MFPSN runs in $O(K(n'_H)^2m'_H)$ time. From Section 2 we obtain $n' = O(nT)$ and $m' = O(mT)$. Therefore we obtain that algorithm MF-PDN runs in $O(KT^3n^2m)$ time. \square

In accordance with the remark in Theorem 3, we note that the algorithm MF-PDN runs in $O(KnmT^2)$ time.

5 Example

The support parametric dynamic network is presented in Figure 5.1.(a) and the time horizon is set to $T = 3$, therefore $H=\{0, 1, 2, 3\}$. The transit times $h(i, j; t)$ and the dynamic upper bounds (capacities) $u_h(i, j; t)$, as well as the parametric dynamic lower bounds $l_h(i, j; t; \lambda) = l_{0h}(i, j; t) + \lambda \cdot L_h(i, j; t)$ the for all arcs in \tilde{G}_h are indicated in the two tables in Figure 5.1.b. The interval of parameter λ values is set to $[0, 1]$, i.e., $\Lambda=1$.



(i, j)	$h(i, j; t)$	$U_h(i, j; t)$
(1, 2)	$1, t = 0$	5
	$2, t = 1, 2, 3$	
(1, 3)	$1, t = 0, 1$	5
	$2, t = 2, 3$	
(2, 3)	$1, t = 0, 1, 2, 3$	5
(2, 4)	$1, t = 0, 1$	5
	$2, t = 2, 3$	
(3, 4)	$2, t = 0, 1$	5
	$1, t = 2, 3$	

(i, j)	$l_{0h}(i, j; t)$	$L_h(i, j; t)$
(1, 2)	$3, t = 0$	$-2, t = 0$
	$0, t = 1, 2, 3$	$0, t = 1, 2, 3$
(1, 3)	$1, t = 0, 1$	$4, t = 0$
	$0, t = 2, 3$	$1, t = 1$
		$0, t = 2, 3$
(2, 3)	$0, t = 0, 1, 2, 3$	$3, t = 1$
(2, 4)	$0, t = 0, 1, 2, 3$	$0, t = 0, 2, 3$
		$0, t = 0, 1, 2, 3$
(3, 4)	$0, t = 0$	$-2, t = 2$
	$2, t = 1, 2$	$0, t = 0, 1, 3$
	$0, t = 3$	

(b)

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

The support graph for parametric super-extended network \tilde{G}_H^* is presented in Figure 5.2.

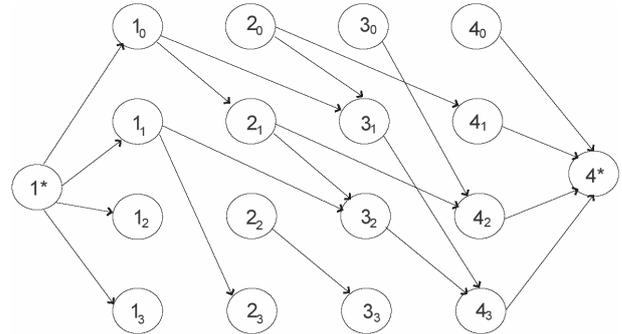


Figure 5.2. The support graph for parametric superextended network \tilde{G}_H^* .

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

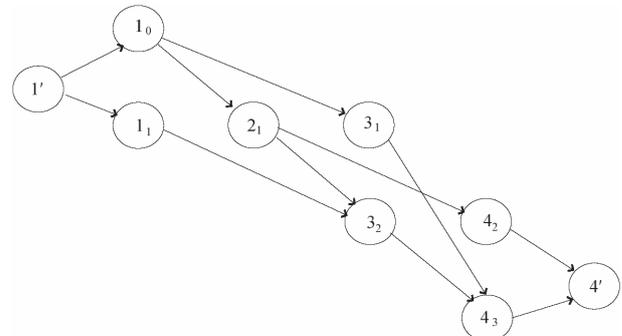


Figure 5.3. The support graph for parametric reduced expanded network \tilde{G}' .

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

(i_t, j_θ)	$l'(i_t, j_\theta)$	$L'(i_t, j_\theta)$	$u'(i_t, j_\theta)$	$\hat{f}'(i_t, j_\theta)$
$(1', 1_0)$	0	0	∞	10
$(1', 1_1)$	0	0	∞	2
$(1_0, 2_1)$	3	-2	5	5
$(1_0, 3_1)$	1	3	5	5
$(1_1, 3_2)$	1	1	5	2
$(2_1, 3_2)$	0	3	5	3
$(2_1, 4_2)$	0	0	5	2
$(3_1, 4_3)$	2	0	5	5
$(3_2, 4_3)$	2	-2	5	5
$(4_2, 4')$	0	0	∞	2
$(4_3, 4')$	0	0	∞	10

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

In the first phase we determine in static residual network \tilde{G}' : $\tilde{P}'_1 = (1', 1_0, 2_1, 4_2, 4')$, $\hat{r}'(\tilde{P}'_1) = 2$; $\tilde{P}'_2 = (1', 1_0, 3_1, 4_3, 4')$, $\hat{r}'(\tilde{P}'_2) = 5$; $\tilde{P}'_3 = (1', 1_1, 3_2, 4_3, 4')$, $\hat{r}'(\tilde{P}'_3) = 2$; $\tilde{P}'_4 = (1', 1_0, 2_1, 3_2, 4_3, 4')$, $\hat{r}'(\tilde{P}'_4) = 3$. The feasible flow \hat{f}' in network \tilde{G}' is presented in the table from Figure 5.4.

In the second phase by applying algorithm MF-PSN in network \tilde{G}' we obtain in the parametric static residual network \tilde{G}' the following directed path: $\tilde{P}'_1 = (1', 1_1, 3_2, 2_1, 4_2, 4')$, $\tilde{P}'_2 = (1', 1_0, 2_1, 4_2, 4')$, $\tilde{P}'_3 = (1', 1_0, 3_1, 4_3, 4')$, $\tilde{P}'_4 = (1', 1_0, 2_1, 3_2, 4_3, 4')$ in $\tilde{G}'_0, \tilde{G}'_1, \tilde{G}'_2$. The results of this example are synthetically indicated in the table from Figure 5.5. The graphic of $\bar{v}'(\lambda)$ is presented in Figure 5.6.

k	λ_k	\tilde{P}'_i	$\bar{r}_k(\tilde{P}'_i)$	λ_{k+1}	$\bar{v}'(\lambda)$
0	0	\tilde{P}'_1	$1 - \lambda$	1	$6 - \lambda$
		\tilde{P}'_2	$1 + \lambda$	1	
		\tilde{P}'_3	3	1/4	
		\tilde{P}'_4	$1 + \lambda$	1/4	
1	1/4	\tilde{P}'_1	$1 - \lambda$	1	$5 + 3\lambda$
		\tilde{P}'_2	$1 + \lambda$	1	
		\tilde{P}'_3	$4 - 4\lambda$	1	
		\tilde{P}'_4	$1 + \lambda$	3/5	
2	3/5	\tilde{P}'_1	$1 + \lambda$	1	$2 + 8\lambda$
		\tilde{P}'_2	$1 + \lambda$	1	
		\tilde{P}'_3	$4 - 4\lambda$	1	
		\tilde{P}'_4	$4 - 4\lambda$	1	

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

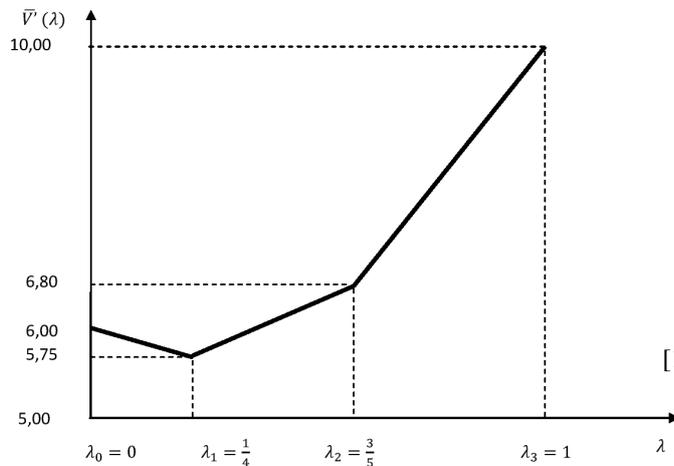


Figure 5.6. The graphic of $\bar{v}'(\lambda)$

References

[1] Ahuja R., Magnanti T., Orlin J. (1993) *Network Flows. Theory, algorithms and applications*, Prentice Hall, Inc., Englewood Cliffs, New Jersey.

[2] Aronson J.A. (1989) A survey of dynamic networks flows, *Annals of Operations Research*, **20** (5), pp.1–66. <https://doi.org/10.1007/bf02216922>

[3] Avesalon N., Ciurea E., Parpalea M. (2017), Maximum parametric flow in discrete-time dynamic networks, *Fundamenta Informaticae*, **156** (2), pp.125–139. <https://doi.org/10.1007/bf02216922>

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

[5] Ciurea E., Ciupală L. (2004) Sequential and parallel algorithms for minimum flow, *Journal of Applied Mathematics and Computing*, **15**(1-2), pp.53–75. <https://doi.org/10.1007/bf02935746>

[6] Ciurea E. (2002) Second best temporally repeated flows, *The Korean Journal of Computational and Applied Mathematics*, **9**(1), pp.77–86. <https://doi.org/10.1007/bf03012341>

[7] He X., Zheng H., Peeta S. (2015) Model and solution algorithm for the dynamic resource allocation problem for large-scale transportation network evacuation, *Transportation Research Part C.: Emerging Technologies*, **59**, pp.233–247. <https://doi.org/10.1016/j.trc.2015.05.005>

[8] Miller-Hooks E., Patterson S.S. (2004) On solving quickest time problems in time-dependent dynamic networks, *Journal of Mathematical Modelling and Algorithms*, **3**, pp.39–71. <https://doi.org/10.1023/b:jmma.0000026708.57419.6d>

[9] Nasrabadi E., Hashemi S.M. (2010) Minimum cost time-varying network flow problems, *Optimization Methods and Software*, **25** (3), pp.429–447. <https://doi.org/10.1080/10556780903239121>

[10] Orlin J. (2013) Max Flows in $O(nm)$ time, or better, *Proceeding of the forty-fifth Annual ACM Symposium on Theory of Computing*, ACM Press, New York, pp.765–774. <https://doi.org/10.1145/2488608.2488705>

[11] Parpalea M., Ciurea E. (2016) Minimum parametric flow. A partitioning approach, *British Journal of Applied Science and Technology*, **13** (6), pp.1–8. <https://doi.org/10.9734/bjast/2016/22636>

[12] Parpalea M., Ciurea E. (2011) The quickest maximum dynamic flow of minimum cost, *International Journal of Applied Mathematics and Informatics*, **3** (5), pp.266–274.

[13] Parpalea M., Ciurea E. (2013), Partitioning algorithm for the parametric maximum flow, *Applied Mathematics*, **4** (10A), pp.3–10. <https://doi.org/10.4236/am.2013.410a1002>

- [14] Parpalea M., Avesalon N., Ciurea E. (2018) Minimum parametric flow in time dependent dynamic networks, *Revue d'Automatique, d'Informatique et de Recherche Opérationnelle - Theoretical Informatics and Applications (RAIRO: ITA)*, **52** (1), pp.43–53.<https://doi.org/10.1051/ita/2018002>
- [15] Rashidi H., Tsang E. (2015) *Vehicle Scheduling in Port Automation: Advanced Algorithms for Minimum Cost Flow Problems*, CRC Press, 2 edition, Boca Raton, London, New York.<https://doi.org/10.1201/b18984>
- [16] Ruhe G. (1991) *Algorithmic Aspects of Flows in Networks*, Kluwer Academic Publisher, Dordrecht, The Netherlands.<https://doi.org/10.1007/978-94-011-3444-6>
- [17] Zheng H., Chiu Y.C., Mirchandani P.B. (2015) On the System Optimum Dynamic Traffic Assignment and Earliest Arrival Flow Problems, *Transportation Science*, **49**, pp.13–27.<https://doi.org/10.1287/trsc.2013.0485>