ROMANIAN JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY
Volume **20**, Number 2, 2017, 124–135

The maximum flows in parametric dynamic networks

Nicoleta AVESALON

Department of Mathematics and Computer Science, Transilvania University of Brasov, Romania
Email: nicole.grigoras@gmail.com

Abstract. This article presents and solves the maximum parametric flow over time problem. The proposed approach consists in applying a parametric flow algorithm in the static time-space network, obtained by expanding the original dynamic network. A numerical example is presented.

Key-words: Dynamic network, parametric network, maximal flow.

1. Introduction

The static network flow models bridges several diverse and seemingly unrelated areas of combinatorial optimization. More often in scientific writing, flow in a network refers to the flow of electricity, phone calls, email messages, commodities being transported across truck routes, or other such kinds of flow [1]. However, in some other applications, time is an essential ingredient [2]. In this instance, to account properly for the evolution of the underlying system over time, we need to use dynamic network flow models. On the other hand, a natural generalization of the maximum flow problem can be obtained by making the capacities of some arcs functions of a single parameter [5, 9]. This class of problems is known as the parametric maximum network flow problem.

In this paper, the case of the maximum parametric dynamic flow in dynamic networks is considered. The proposed approach consists in transformation the maximum flow in parametric dynamic network problem in the maximum flow parametric static network.

Further on, in Section 2 some basic dynamic network notations and terminology are presented. In Section 3 the parametric maximum flow in static network is exposed, while in Section 4 is presented the algorithm for solving the parametric maximum flow in dynamic network. In Section 5 an example is given.

In the presentation to follow, some familiarity with flow algorithms is assumed and many details are omitted, since they are straight forward modifications of known results.

2. The maximum flows in dynamic networks

Let G = (N, A, u) be a static network with the node set N = {1,...,i,...,j,...,n}, the arc set $A = \{a_1, ..., a_k, ..., a_m\}, a_k = (i, j)$ and the capacity function $u : A \to \mathbf{R}^+$, where $\mathbf{R}$ is real number set.

To define the maximal static flow problem, we distinguish two special nodes in the static network $G = (N, A, u)$: a source node 1 and a sink node n.

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

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 $G_h = (N, A, h, u_h)$:

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

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

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

$$0 \leqslant f_h(i, j; t) \leqslant u_h(i, j; t), \text{ for all } (i, j) \in A \text{ and for all } t \in H \tag{2.2}$$

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

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 $f_h(i, j; t) = 0, (i, j) \in A, t \in \{T - h(i, j; t) + 1, ..., T\}$.

In other words, a dynamic flow f_h from 1 to n is any flow f_h from 1 to n in which not more than $u_h(i, j; t)$ flow units starting from node i at time t and arriving at node j at time $t + h(i, j; t)$, for all arcs (i, j) and all t. Note that in a dynamic flow, units may be departing from the source at time 0,1,...,$T', T' < T$. A maximum dynamic flow for T time periods from 1 to n is any dynamic flow from 1 to n in which the maximum possible number of flow units arrive at the sink node n during the first T time periods.

We will show how to transform the maximum dynamic flow problem in the dynamic network G = (N, A,h, u_h) into a static flow problem on a static network $G'_H = (N'_H, A'_H, u'_H)$, called the reduced expanded network.

For a given dynamic network $G_h = (N, A, h, u_h)$, we form the expanded network $G_H = (N_H, A_H, u_H)$ as follows. We make T+1 copies i_t, $t = 0, 1, ..., T$ of each node i in G_h. Node i_t in G_H represents node i in G_h at time t. For each (i, j) in G_h, there are arcs (i_t, j_θ), $\theta = t + h(i, j; t)$, $t = 0, 1, ..., T - h(i, j; t)$ with capacity $u(i_t, j_\theta)$= $u_h(i, j; t)$ in G_H. The arc (i_t, j_θ) in G_H represents the potential movement of a commodity from node i to node j in time $h(i, j; t)$. The number of nodes in G_H is $n(T + 1)$, and number of arcs is limited by $m(T + 1) -$

$\sum_A \bar{h}(i,j)$, where $\bar{h}(i,j) = min\{h(i,j,0),...,h(i,j;T)\}$. It is easy to see that any dynamic flow in dynamic network G_h is equivalent to a static flow in static network G_H from the source nodes $1_0, 1_1, ..., 1_T$ to the sink nodes $n_0, n_1, ..., n_T$, 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 supersource node 1^* and a supersink node n^* constructing superexpanded network $G_H^* = (N_H^*, A_H^*, u_H^*)$, where $N_H^* = N_H \cup \{1^*, n^*\}$, $A_H^* = A_H \cup \{(1^*, 1_t)|t = 0,1,...,T\} \cup \{(n_t, n^*)|t = 0,1,...,T\}$, $u_H^*(i_t, j_\theta) = u_H(i_t, j_\theta)$ for all $(i_t, j_\theta) \in A_H, u_H^*(1^*, 1_t) = u_H^*(n_t, n^*) = \infty, t = 0,1,...,T$.

Now, we construct the reduced expanded network $G'_H = (N'_H, A'_H, u'_H)$ as follows. We define the function $h^*, h^* : A_H^* \to \mathbf{N}, h^*(1^*, 1_t) = h^*(n_t, n^*) = 0, t = 0,1,...,T, h^*(i_t, j_\theta) = h(i,j;t), t = 0,1,...,T - h(i,j,t)$. Let $d^*(1^*, i_t)$ be the length of the shortest path from the source node 1^* to the node i_t in network G_H^* and $d^*(i_t, n^*)$ the length of the shortest path from node i_t to the sink node n^*, with respect to h^*. The computation of $d^*(1^*, i_t)$ and $d^*(i_t, n^*)$ for all $i_t \in N_H^*$ is performed by means of the usual shortest path algorithms. In network G'_H we rewrite the nodes $1^*, n^*$ by $1'$ respectively n'. We have $N'_H = \{1', n'\} \cup \{i_t | i_t \in N_H, d^*(1^*, i_t) + d^*(i_t, n^*) \leqslant T\}, A'_H = \{(1', 1_t)|d^*(1_t, n^*) \leqslant T\} \cup \{(n_t, n')|d^*(1^*, n_t) \leqslant T\} \cup \{i_t, j_\theta)|(i_t, j_\theta) \in A_H, d^*(1^*, i_t) + h^*(i_t, j_\theta) + d^*(j_\theta, n^*) \leqslant T\}$ and u'_H is restriction of u_H^* at A'_H.

It is easy to see that the network G'_H 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'_H cannot contain any circuit, and are therefore acyclic always.

In the most general dynamic model, the parameter h(i) = 1 is waiting time at node i, and the parameter u(i;t) is defined as the capacity of the node i, which represents 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 (1), (2), (3) is equivalent to the maximum static flow problem in static network G'_H as follows:

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

$$0 \leqslant f'(i_t, j_\theta) \leqslant u'(i_t, j_\theta), \text{ for all } (i_t, j_\theta) \in A'_H, \quad (2.5)$$

$$\max v', \quad (2.6)$$

where by convention $i_t = 1'$ for t= -1 and $i_t = n'$ for t=T+1.
For more details we recommend the works [2], [3], [5], [6], [10].

3. The maximum flow in parametric static networks

A natural generalization of the maximum flow problem in static networks can be obtained by making the 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 parametric maximum flow problem can alternately be defined as to find all the breakpoints and their corresponding maximum flows and minimum cuts. Hamacher and Foulds [5] investigated an approach for determining in each iteration an improvement of the flow defined on the whole interval of the parameter while for the same problem Ruhe [9] proposed a piece-by-piece approach, computing at each step a maximum flow to be added to the current flow in order to preserve its maximally for greater parameter values. Parpalea and Ciurea [8] presented an algorithm for the maximum flow problem in static networks with constant lower bounds and linear upper bound functions of a single parameter.

The approach presented in this section was proposed in [8]. A static network G = (N, A, u) with the upper bounds u 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 $\bar{G} = (N, A, \bar{u})$. The upper bound function (capacity 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 parametric part of the upper bound function $\bar{u}$ and $u_0 : A \to \mathbf{R}^+$ is 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 $U(i, j) \geqslant -u_0(i, j)/\Lambda, (i, j) \in A$.

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

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

$$0 \leqslant \bar{f}(i, j; \lambda) \leqslant \bar{u}(i, j; \lambda), (i, j) \in A\ , \tag{3.2.d}$$

$$max v(\lambda) \tag{3.3}$$

For maximum flow problem in parametric static network $\bar{G} = (N, A, \bar{u})$, the subintervals $J_k = [\lambda_k, \lambda_{k+1}], k = 0, ..., K$ of the parameter λ values can be determined such as a minimum 1-n cut in the non-parametric static network $G_k = (N, A, u_k), u_k(i, j) = \bar{u}(i, j; \lambda_k)$, also to represent a minimum 1-n cut for all the parameter λ values within the subinterval J_k. A parametric 1-n cut partitioning in parametric static network $\bar{G} = (N, A, \bar{u})$ is defined as a finite set of cuts $[S_k, T_k], k = 0, ..., K$ together with a partitioning of the interval $[0, \Lambda]$ of the parameter in disjoints subintervals $J_k, k = 0, ..., K$, such that $J_0 \cup ... \cup J_k = [0, \Lambda]$. It is denoted by $[S_k; J_k], k = 0, ..., K$. The capacity of a parametric 1-n cut partitioning is defined as:

$$\bar{c}[S_k; J_k] = \sum_{(S_k, T_k)} \bar{u}(i, j; \lambda), k = 0, ..., K \tag{3.4}$$

Definition 3. 1. *A parametric 1-n cut partitioning $[S_k; J_k]$ with the subinterval J_k assuring that every cut is a minimum cut $[S_k^*, T_k^*]$ within the subinterval $[\lambda_k, \lambda_{k+1}]$ is referred to as parametric minimum 1-n cut and is denoted by $[S_k^*, J_k], k = 0, ...K$.*

Definition 3. 2. *For a flow $\bar{f}$ in parametric static network $\bar{G} = (N, A, \bar{u})$ the parametric residual capacity $\bar{r}(i, j; \lambda)$ of any of the arcs (i, j)$\in A$ is given by:*

$$\bar{r}(i, j; \lambda) = \bar{u}(i, j; \lambda) - \bar{f}(i, j; \lambda) + \bar{f}(j, i; \lambda) = \bar{\alpha}(i, j) + \lambda\bar{\beta}(i, j), \tag{3.5}$$

$\lambda \in J_k, k = 0, ...K$

Definition 3. 3. *For set $\bar{s}(i, j) = \{\lambda | \bar{r}(i, j; \lambda) > 0\}, (i, j) \in A$ and for a flow $\bar{f}$ in parametric static network $\bar{G} = (N, A, \bar{u})$, the static network denoted by $\tilde{\tilde{G}}$ = (N,$\tilde{A}$, $\bar{r}$), which $\tilde{A} = \{(i, j) | (i, j) \in A, \bar{s}(i, j) \neq \phi\}$ is named the parametric residual static network with respect to given flow $\bar{f}$.*

If $(i, j) \in A$ and $(i, j) \notin \tilde{A}$, then $\bar{s}(i, j) = \phi$.

Definition 3. 4. *A partly conditional augmenting directed path $\tilde{\tilde{P}}_k$ is a directed path $\tilde{\tilde{P}}_k$ from the source node 1 to sink node k in the parametric residual static network $\tilde{\tilde{G}}$, with the restriction:*

$$\bar{s}(\tilde{\tilde{P}}_k) = \bigcap_{\tilde{\tilde{P}}_k} \bar{s}(i, j) \neq \phi \tag{3.6}$$

If k=n, then $\tilde{\tilde{P}} = \tilde{\tilde{P}}_n$ is a conditional augmenting directed path from the source node 1 to the sink node n. The parametric residual capacity of a conditional augmenting directed path $\tilde{\tilde{P}}$ is $\bar{r}(\tilde{\tilde{P}}; \lambda) = min\{\bar{r}(i, j; \lambda) | (i, j) \in \tilde{\tilde{P}}, \lambda \in \bar{s}(\tilde{\tilde{P}})$.

Theorem 3. 1 (Theorem of Conditional Augmenting Directed Path)**.** *A flow $\bar{f}$ is a maximum flow in parametric static network $\bar{G}$ if and only if the parametric residual static network $\tilde{\tilde{G}}$ contains no conditional augmenting directed path $\tilde{\tilde{P}}$.*

If network $\tilde{\tilde{G}}$ contains no directed path $\tilde{\tilde{P}}$, then the maximum flow in network $\bar{G}$ is computed as:

$$\bar{f}(i, j; \lambda) = max\{\bar{u}(i, j; \lambda) - \bar{r}(i, j; \lambda), 0\} \tag{3.7}$$

If $\bar{f} = 0$, then obviously $\tilde{\tilde{G}} = \bar{G}$

The partitioning algorithm for the parametric maximum flow (PAPMF) is presented in Figure 3.1(see [8])

```
PAPMF;
BEGIN
    f̄0 := 0;
    compute the network G̃̃;
    B:={0}; k := 0; λk := 0;
    REPEAT
      SADP( k, λk, B);
      k:=k+1;
    UNTIL( λk = Λ);
END.
```

Figure 3.1.a. Partitioning algorithm for the parametric maximum flow.

(PAPMF)

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

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

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

(13) THEN $\lambda_{k+1} := \lambda'$;
(14) END;
(15) $\alpha_k(p(i), i) := \alpha_k(p(i), i) - \alpha_k(\tilde{\bar{P}})$; $\beta_k(p(i), i) := \beta_k(p(i), i) - \beta_k(\tilde{\bar{P}})$;
(16) $\alpha_k(i, p(i)) := \alpha_k(i, p(i)) + \alpha_k(\tilde{\bar{P}})$; $\beta_k(i, p(i)) := \beta_k(i, p(i)) + \beta_k(\tilde{\bar{P}})$;
(17) i:=p(i);
(18) END;
(19) END;

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

In order to avoid working with piecewise linear functions, the PAPMF works in parametric residual static networks defined for subintervals of the parameter values where the parametric residual capacities of all arcs remain linear functions. The parametric residual network $\tilde{\bar{G}}$ for a subinterval $J_k = [\lambda_k, \lambda_{k+1}]$ is denoted by $\tilde{\bar{G}}_k$. In k-th step of the PAPMF, 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 $\alpha_k(i, j) = \bar{\alpha}(i, j) + \lambda_k \bar{\beta}(i, j), \beta_k(i, j) = \bar{\beta}(i, j)$ 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 $\alpha_k(i, p(i))$, $\beta_k(i, p(i))$, $\alpha_k(p(i), i), \beta_k(p(i), i)$. Obviously in $\tilde{\bar{G}}_k$ we have $\bar{r}_k(\tilde{\bar{P}}, \lambda) = \alpha_k(\tilde{\bar{P}}) + (\lambda - \lambda_k)\beta_k(\tilde{\bar{P}})$.

Theorem 3. 2 (Theorem of Correctness)**.** *The partitioning algorithm for the parametric maximum flow computes correctly a parametric maximum flow for* $\lambda \in [0, \Lambda]$ *in the parametric static network* $\bar{G} = (N, A, \bar{u})$.

Theorem 3. 3 (Theorem of Complexity)**.** *The partitioning algorithm for the parametric maximum flow runs in O(K*n^2*m) 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 maximum flow problem in the non-parametric static network can be solved in O(nm) time (see [7]).

4. The maximum flow in parametric dynamic networks

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

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

The upper bounds $U_h(i, j; t)$ must satisfy the constraint

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

The maximum flow problem in parametric dynamic network $\bar{G}_h$ is to compute flow function $\bar{f}_h : A \times H \times I \to \mathbf{R}^+$ that satisfies the following constraints:

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

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

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

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

For the maximum flow in parametric dynamic network problem, the parametric dynamic residual capacities with respect to given flow $\bar{f}_h$ are defined as follows:

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

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

There are at least two approaches for solving the maximum flow in parametric dynamic network problem. The first approach consists in applying a classical parametric flow algorithm (see [4, 8, 9]) in the parametric reduced expanded network $\bar{G}'_H = (N'_H, A'_H, \bar{u'}_H)$ which is constructed similarly with construction of the network $G'_H = (N'_H, A'_H, u'_H)$ in Section 2. In this section we use the algorithm presented in Section 3. The second approach, consists in applying a parametric flow algorithm in dynamic network.

The *Algorithm for the Maximum Parametric Dynamic Flow*(AMPDF) is presented in Figure 4.1.

(1) AMPDF
(2) BEGIN
(3) construct the network $\bar{G}'_H$;
(4) apply the PAPMF in network $\bar{G}'_H$;
(5)END.

Figure 4.1. Algorithm for maximum parametric dynamic flow(AMPDF)

Theorem 4. 1 (Theorem of Correctness)**.** *The algorithm for maximum parametric dynamic flow computes correctly a maximum flow in parametric dynamic networks.*

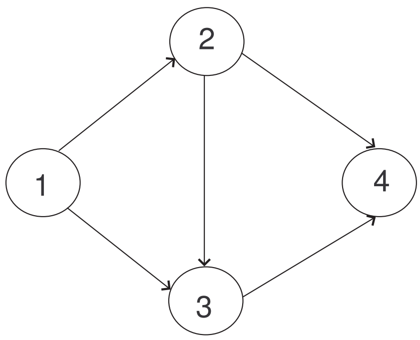
Proof. This theorem results from the fact that the maximum parametric dynamic flow problem on a parametric dynamic network $\bar{G}_h = (N, A, h, \bar{u}_h)$ is equivalent to the maximum static flow problem on parametric static network $\bar{G}'_H = (N'_H, A'_H, u'_H)$ and Theorem 3.2 $\square$

Theorem 4. 2 (Theorem of Complexity)**.** *The algorithm for maximum parametric dynamic flow 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 runs in $O(K(n'_H)^2m'_H)$. From Section 2 we obtain $n'_H = O(nT)$ and $m'_H = O(mT)$. Therefore we obtain that algorithm runs in $O(KT^3n^2m)$ time. $\square$

5. Example

The parametric dynamic network is presented in Figure 5.1.a and time horizon being set to T=3, therefore H=$\{0, 1, 2, 3\}$.



(a)

$a_k = (i,j)$	$h(a_k;t)$	$u_{0h}(a_k;t)$	$U_h(a_k;t)$
$a_1 = (1,2)$	$\begin{cases} 1, & t = 0 \\ 2, & t \geqslant 1 \end{cases}$	4	4
$a_2 = (1,3)$	$\begin{cases} 1, & 0 \leqslant t < 2 \\ 2, & t \geqslant 2 \end{cases}$	$\begin{cases} 9, & 0 \leqslant t < 2 \\ 6, & t \geqslant 2 \end{cases}$	-4
$a_3 = (2,3)$	1	3	-2
$a_4 = (2,4)$	$\begin{cases} 1, & 0 \leqslant t < 2 \\ 2, & t \geqslant 2 \end{cases}$	4	2
$a_5 = (3,4)$	$\begin{cases} 2, & 0 \leqslant t < 2 \\ 1, & t \geqslant 2 \end{cases}$	8	$\begin{cases} 0, & 0 \leqslant t < 2 \\ -2, & t \geqslant 2 \end{cases}$

(b)

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

The transit times 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 [0,1], i.e., Λ=1.

The parametric superextended network $\bar{G}_H^* = (N_H^*, A_H^*, \bar{u}_H^*)$ is presented in Figure 5.2.

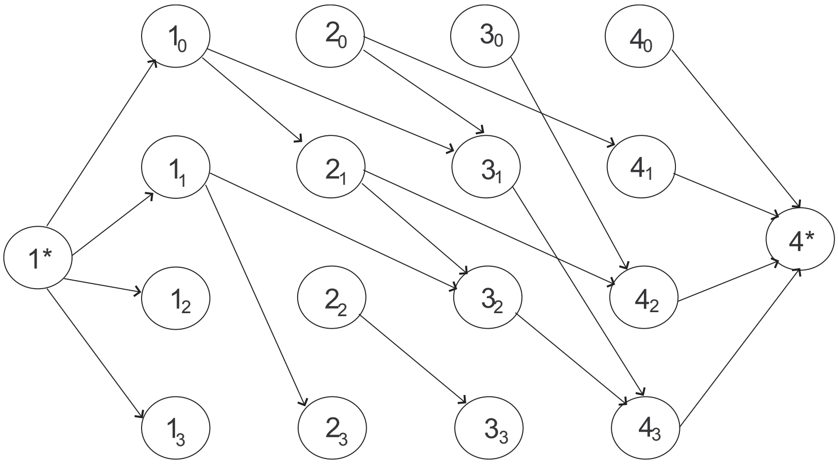


Figure 5.2. Parametric time super extended network $\bar{G}_H^* = (N_H^*, A_H^*, \bar{u}_H^*)$.

The parametric reduced expanded network $\bar{G}'_H = (N'_H, A'_H, \bar{u}'_H)$ is presented in Figure 5.3.

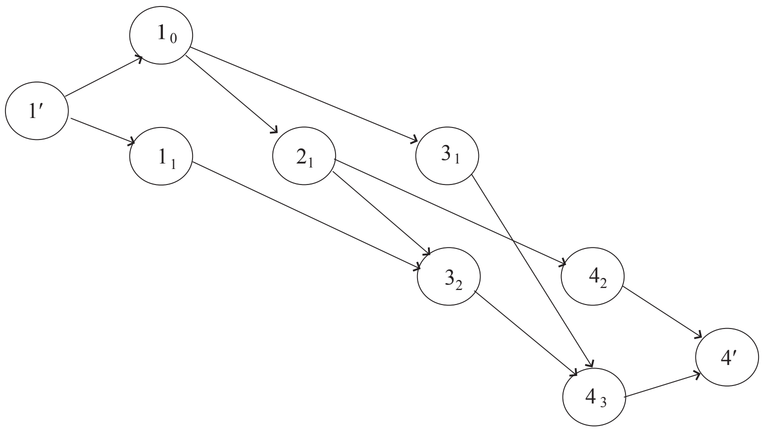


Figure 5.3. Parametric time reduced network $\bar{G}'_H = (N'_H, A'_H, \bar{u}'_H)$

We have: $\bar{u}_H(1', 1_0; \lambda) = \bar{u}_H(1', 1_1; \lambda) = \bar{u}_H(4_2, 4'; \lambda) = \bar{u}_H(4_3, 4'; \lambda) = \infty$, $\bar{u}_H(1_0, 2_1; \lambda) = 4+4\lambda$, $\bar{u}_H(1_0, 3_1; \lambda) = 9-4\lambda$, $\bar{u}_H(1_1, 3_2; \lambda) = 9-4\lambda$, $\bar{u}_H(2_1, 3_2; \lambda) = 3-2\lambda$, $\bar{u}_H(2_1, 4_2; \lambda) = 4 + 2\lambda$, $\bar{u}_H(3_1, 4_3; \lambda) = 8$, $\bar{u}_H(3_2, 4_3; \lambda) = 8 - 2\lambda$, $\lambda \in [0, 1]$.

Applying the PAPMF in network $\bar{G}'_H$ we obtain the results which are synthetically indicated in table from Figure 5.4.

k	$[\lambda_k, \lambda_{k+1}]$	$\tilde{\bar{P}}'$	$\bar{r}(\tilde{\bar{P}}'; \lambda)$	$\bar{v}'(\lambda)$
0	$[0, 1/4]$	$(1', 1_0, 2_1, 4_2, 4')$	$4 + 2\lambda$	20
		$(1', 1_0, 3_1, 4_3, 4')$	8	
		$(1', 1_1, 3_2, 4_3, 4')$	$8 - 2\lambda$	
1	$[1/4, 1/2]$	$(1', 1_0, 2_1, 4_2, 4')$	$4 + 2\lambda$	$21 - 4\lambda$
		$(1', 1_0, 3_1, 4_3, 4')$	$9 - 4\lambda$	
		$(1', 1_1, 3_2, 4_3, 4')$	$8 - 2\lambda$	
2	$[1/2, 1]$	$(1', 1_0, 2_1, 4_2, 4')$	$4 + 2\lambda$	$21 - 4\lambda$
		$(1', 1_0, 3_1, 4_3, 4')$	$9 - 4\lambda$	
		$(1', 1_1, 3_2, 4_3, 4')$	$9 - 4\lambda$	
		$(1', 1_0, 2_1, 3_2, 4_3, 4')$	$2\lambda - 1$	

Figure 5.4. Results of PAPMF in network $\bar{G}'_H$

The flow $\bar{f}'$ is obtained from $\bar{r}(\tilde{\bar{P}}'; \lambda)$ functions of parameter λ.

The graph of $\bar{v}'(\lambda)$ is presented in Figure 5.5.

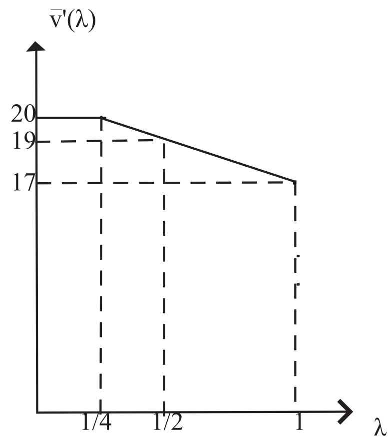


Figure 5.5 The graph of $\bar{v}'(\lambda)$

References

[1] AHUJA R., MAGNANTI T., ORLIN J., *Network Flows. Theory, algorithms and applications*, Prentice Hall, Inc., Englewood Clifss, New Jersey, 1993.

[2] ARONSON J. A., *A survey of dynamic network flows*, Annals of Operations Research, 20, pp.1–66, 1989.

[3] CAI X., SHA D., WONG C., Time-varying Network Optimization, Springer, 2007.

[4] CIUREA E., *Second best temporally repeated flows,* The Korean Journal of Computational and Applied Mathematics, **9**(1), pp. 77–86, 2002.

[5] HAMACHER H., FOULDS L., *Algorithms for Flows with Parametric Capacities*, ZOR-Methods and Models of Operations Research, **3**, pp. 21–37, 1989.

[6] NASRABADI E., HASHEMI S., *Minimum cost time-varying network flow problems*, Optimization Methods and Software, **25**(3), pp. 429–447, 2010.

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

[8] PARPALEA M., CIUREA E., *Partitioning algorithm for the parametric maximum flow*, Applied Mathematics, 4, Special Issue on Computer Mathematics, (10A), pp. 3–10, 2013.

[9] RUHE G., *Algorithmic Aspects of Flows in Networks*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1991.

[10] SKUTELLA M., *An Introduction to Network Flows Over Time*, Research Trends in Combinatorical Optimization, Springer, pp. 451–482, 2008.