Bulletin of the *Transilvania* University of Braşov • Vol 12(61), No.1 - 2019
Series III: Mathematics, Informatics, Physics, 157-176
https://doi.org/10.31926/but.mif.2019.12.61.1.13

PARAMETRIC FLOWS IN STATIC NETWORKS

Mircea PARPALEA[1]

Dedicated to the 75th birthday of Professor Eleonor Ciurea

Abstract

Parametric flow problems may be regarded as some extensions of the classical maximum/minimum flow problems in which the capacities of certain arcs are not fixed but are functions of a parameter. Consequently, these problems consist of solving several nonparametric, ordinary maximum/minimum flow problems for all the parameter values within certain subintervals of the parameter values. During the last few decades, research that has been conducted resulted in new solution methods and in improvements of the algorithms for known methods. This paper presents the continuous improvements that have been made during the last years as well as recent techniques and algorithms related to parametric maximum and minimum flow problems. The paper also constitutes a synthesis of the different approaches that have been proposed by us for solving the parametric flow problem in static networks.

2000 *Mathematics Subject Classification:* 90B10, 90C35, 90C47.

Key words: maximum / minimum flow, parametric network, network partitioning algorithms.

1 Introduction

In our everyday lives we encounter many different types of real networks, including electrical networks, telephone cable, highway networks, railways, manufacturing networks or computer networks. Flow networks are very useful to model any real world problem confronting with any commodity flowing through a network (pipes, electric wires, highways, rails, and so on).

Generally, networks consist of special points called nodes and links connecting pairs of nodes called arcs. Network flow problems have always been among the best studied [1] combinatorial optimization problems. Maximum flow problem is

[1]National College *Andrei Şaguna*, Braşov, Romania, e-mail: parpalea@gmail.com

the classical network flow problem in network flow theory and has been extensively investigated. In this problem, the maximum flow which can be moved from the source to the sink is calculated without exceeding the maximum capacity. Once the maximum flow problem is solved, it can be used to solve other network flow problems too [14].

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.

Beside the applications of the ordinary maximum/minimum flows, those of parametric flows include: multiprocessor scheduling with release times and deadlines [10], integer programming problems [2], computing sub-graph density and network vulnerability and partitioning a data base between fast and slow memory [3], product selection [5], [20], flow sharing [10], database record segmentation in large shared databases [9], and optimizing field repair kits [16].

For the parametric maximum flow problem, Gallo, Grigoriadis and Tarjan [10] use a modified version of the push-relabel method using amortization and graph contraction and obtain an algorithm that solves the parametric maximum flow problem in the same asymptotic running time as the original algorithm. Applying their idea to the network of King, Rao and Tarjan [15] they obtained the same $O(nm \cdot log_{m/(nlogn)} n)$ complexity limit for the parametric flow problem as for the ordinary maximum flow problem.

Using a divide and conquer approach that uses an ordinary push-relabel algorithm [12] on a network with n vertices, m arcs, and integer upper bounds bounded by U, Tarjan et al. [23] achieved a running time which is bounded by $O(min\left\{n^{2/3}, m^{1/2}\right\} \cdot m \cdot log(n^2/m) \cdot logU)$, that is by a factor (i.e. $min\left\{n, log(nU)\right\}$) worse than that of the ordinary maximum flow algorithm [11].
For the parametric maximum flow problem with linear capacity functions of a single parameter, Hamacher and Foulds [13] investigated an augmentation path approach for determining in each iteration an improvement of the flow defined on the whole interval of the parameter. For the same problem, Ruhe [21], [22] proposed a piece-by-piece approach which assumes that the maximum flow is known for a given value of the parameter and computes the supplementary maximum flow to be added to the current flow in order to preserve the optimality of the flow for greater parameter values and also the maximum value of the parameter for which the computed flow is maximal.
Zhang, Ward and Feng [24], [25] proposed a balancing technique based algorithm for the parametric flow problem in a special bipartite network. With an enhancement suggested by Tarjan et al.[23] the algorithm runs in $O(mn^2 \cdot log(nU))$ time.

The partitioning type approach, which will be presented in this paper, proposes original algorithms for computing both maximum and minimum flows in parametric networks with linear upper / lower bound functions of a single parameter. As Bichot and Siarry showed, the parametric flow problem is of genuine practical and theoretical interest since graph partitioning applications are described on a wide variety of subjects as: data distribution in parallel-computing, VLSI circuit design, image processing, computer vision, route planning, air traffic control, mobile networks, social networks, etc [4].

The rest of this paper has the following structure and content. Section 2 reviews the preliminaries and terminology for ordinary flows in static networks, as follows: Subsection 2.1 deals with maximum flow in static networks and Subsection 2.2 is dedicated to the minimum flow in static networks. Section 3 describes in its Subsection 3.1 the parametric maximum flow problem while the parametric minimum flow problem is described in Subsection 3.2. Subsection 3.3 presents some personal, efficient implementations of the approaches that resolve the parametric flow problem in static networks. Finally, Section 4 brings in an original example which illustrate the partitioning approach for both the parametric maximum and minimum flow in static network and Section 5 contains some concluding remarks.

2 Terminology and preliminaries

Given a capacitated network $G = (N, A, \ell, u, s, t)$ with $N = \{\dots, i, \dots\}$ being the set of nodes i and $A = \{\dots, a, \dots\}$ being the set of arcs a so that for every $a \in A$, $a = (i, j)$ with $i, j \in N$, let $n = |N|$ and $m = |A|$. The *upper bound* function $u(a)$ and the *lower bound* function $\ell(a)$ are two nonnegative functions associated with each arc $a = (i, j) \in A$. The network has two special nodes: a source node s and a sink node t. A flow is a function $f : A \to \Re^+$ satisfying the following conditions:

$$\sum_{j|(i,j)\in A} f(i,j) - \sum_{j|(j,i)\in A} f(j,i) = \begin{cases} v, & i = s \\ 0, & i \neq s, t \\ -v, & i = t \end{cases} \tag{1}$$

for some $v \geq 0$, where v is referred to as the value of the flow f. Any flow on a directed network satisfying the flow bound constraints:

$$\ell(i,j) \leq f(i,j) \leq u(i,j), \qquad \forall (i,j) \in A \tag{2}$$

for every arc $(i, j) \in A$ is referred to as a *feasible flow.*

A *cut* is a partition of the node set N into two subsets S and $T = N - S$, denoted by $[S, T]$. Alternatively, a cut can be defined as the set of arcs whose endpoints belong to different subsets S and T. An arc $(i, j) \in A$ with $i \in S$ and $j \in T$ is referred to as a *forward arc* of the cut while an arc $(i, j) \in A$ with $i \in T$

and $j \in S$ as a *backward arc* of the cut. Let (S, T) denote the set of forward arcs in the cut and let (T, S) denote the set of backward arcs. A cut $[S, T]$ is an $s - t$ cut if $s \in S$ and $t \in T$.

2.1 Maximum flows in static networks

The maximum flow problem is to determine a flow $\tilde{f}$ for which v is maximized. For a feasible flow f in network $G = (N, A, \ell, u, s, t)$, the residual capacity of any arc $(i, j) \in A$ for the maximum flow problem represents the maximum additional flow that can be sent from node i to node j over both arcs (i, j) and (j, i). For the maximum flow problem, the *residual capacity* $\tilde{r}(i, j)$ of any arc $(i, j) \in A$, with respect to a given flow f, is given by:

$$\tilde{r}(i, j) = u(i, j) - f(i, j) + f(j, i) - \ell(j, i). \tag{3}$$

For a network $G = (N, A, \ell, u, s, t)$ and a feasible solution f, the network denoted by $\tilde{G}(f) = (N, \tilde{A})$, where $\tilde{A}$ is the set of *residual arcs* (i, j) with $\tilde{r}(i, j) > 0$ is referred to as the *residual network* with respect to the given flow f for the maximum flow problem. From the residual capacities $\tilde{r}(i, j)$, the flow can be determined using the following expression:

$$f(i, j) = \ell(i, j) + max\left\{u(i, j) - \tilde{r}(i, j) - \ell(i, j),\ 0\right\}. \tag{4}$$

For the maximum flow problem, the capacity $\tilde{c}[S, T]$ of an $s-t$ cut $[S, T]$ is defined as:

$$\tilde{c}[S, T] = u(S, T) - \ell(T, S). \tag{5}$$

The $s - t$ cut with the lowest capacity value among all $s - t$ cuts is referred to as a *minimum cut* $[\tilde{S}, \tilde{T}]$.

Theorem 1. *(Max-Flow Min-Cut Theorem): If there is a feasible flow in the network, the value of the maximum flow from a source s to a sink t in a capacitated network equals the capacity of the minimum* $s - t$ *cut,* $\tilde{v} = \tilde{c}[\tilde{S}, \tilde{T}]$ *.*

A path in $G = (N, A, \ell, u, s, t)$ from the source node s to the sink node t is referred to as an *augmentation path* if the corresponding directed path in the residual network consists only of arcs with positive residual capacities. There is a one-to-one correspondence between augmentation paths P in G and directed paths $\tilde{P}$ from s to t in the residual network $\tilde{G}(f)$. For a directed path $\tilde{P}$ in $\tilde{G}(f)$ we have $\tilde{r}(\tilde{P}) = \min\{\tilde{r}(i, j) | (i, j) \in \tilde{P}\}$.

Theorem 2. *(Augmentation Path Theorem): A flow* $\tilde{f}$ *is a maximum flow if and only if the residual network* $\tilde{G}(\tilde{f})$ *contains no directed path from the source node to the sink node.*

2.2 Minimum flows in static networks

The minimum flow problem is to determine a flow $\hat{f}$ for which v is minimized. The problem can be solved in two phases:
(1) establishing a feasible flow;
(2) from a given feasible flow, establishing the minimum flow. For the first phase see the algorithm presented in [1].

Let f be a feasible solution for the minimum flow problem in network $G = (N, A, \ell, u, s, t)$. Supposing that an arc $(i, j) \in A$ carries $f(i, j)$ units of flow, the residual capacity of any arc $(i, j) \in A$, with respect to the given flow f, for the minimum flow problem, represents the maximum amount of flow by which the flow from node i to node j can be decreased over both arcs (i, j) and (j, i). The *residual capacity* $\hat{r}(i, j)$ of any arc $(i, j) \in A$ is given by:

$$\hat{r}(i, j) = u(j, i) - f(j, i) + f(i, j) - \ell(i, j). \tag{6}$$

For a network $G = (N, A, \ell, u, s, t)$ and a feasible solution f, the network denoted by $\hat{G}(f) = (N, \hat{A})$, where $\hat{A}$ is the set of *residual arcs* corresponding to the feasible solution f and consisting only of arcs (i, j) with $\hat{r}(i, j) > 0$, is referred to as the *residual network* with respect to the given flow f for the minimum flow problem. The capacity of an $s - t$ cut $\hat{c}[S, T]$ is defined, for the minimum flow problem, as:

$$\hat{c}[S, T] = \ell(S, T) - u(T, S). \tag{7}$$

The $s - t$ cut with the greatest capacity value among all $s - t$ cuts is referred to as a *maximum cut* and is denoted by $[\hat{S}, \hat{T}]$.

Theorem 3. *(Min-Flow Max-Cut Theorem): If there is a feasible flow in the network, the value of the minimum flow from a source s to a sink t in a capacitated network with nonnegative lower bounds equals the capacity of the maximum $s - t$ cut.*

A path in $G = (N, A, \ell, u, s, t)$ from the source node s to the sink node t is referred to as a *decreasing path* if the corresponding directed path in the residual network consists only of arcs with positive residual capacities. There is a one-to-one correspondence between decreasing paths P in G and directed paths $\hat{P}$ from s to t in the residual network $\hat{G}(f)$. For a directed path $\hat{P}$ in $\hat{G}(f)$ we have $\hat{r}(\hat{P}) = \min\{\hat{r}(i, j) | (i, j) \in \hat{P}\}$.

Theorem 4. *(Decreasing Path Theorem): A flow $\hat{f}$ is a minimum flow if and only if the residual network $\hat{G}(\hat{f})$ contains no directed path from the source node to the sink node.*

3 Parametric flows in static networks

The problem of parametric flows in static networks concerns the general problem of parametric flows. Given a capacitated directed network with non-negative capacities and/or lower bounds, the natural generalization of the ordinary problem of maximum/minimum flow in static networks is obtained by making the upper/lower bounds for some of the network arcs linearly depend on a single, non-negative, real parameter. The general parametric flow problem is to compute the maximum/minimum flows for every possible value of the parameter within a given interval. The above mentioned problem looks like the ordinary maximum/minimum flow problem in static networks with the decisive differences that the flow variables of the parametric problem are piecewise linear functions instead of real numbers and the upper/lower bounds are linear functions instead of constants.

Definitions included in the following sections are taken from/adapted after, references [18] and [17] while the proof for theorems can be found in reference [19].

3.1 Parametric maximum flows in static networks

The parametric maximum flow problem consists in transforming the classic problem of maximum network flow by adapting it to a network $G = (N, A, \ell, u, s, t)$ where the upper bounds of some arcs $(i, j) \in A$ depend on a real parameter.

Definition 1. *A directed network $G = (N, A, \ell, u, s, t)$ for which the upper bounds u of some arcs $(i, j) \in A$ are functions of a real parameter λ is referred to as a* parametric network *and is denoted by $\bar{G} = (N, A, \ell, \bar{u}, s, t)$.*

For a parametric network $\bar{G}$, the *parametric upper bound* (capacity) function $\bar{u} : A \times [0, \Lambda] \to \Re^{+}$ of an arc $(i, j) \in A$ computes the real numbers $\bar{u}(i, j; \lambda)$, called upper bound of arc (i, j), for all the parameter λ values in a given interval $[0, \Lambda]$:

$$\bar{u}(i, j; \lambda) = u_0(i, j) + \lambda \cdot U(i, j) \geq \ell(i, j), \; for\, all \lambda \in [0, \Lambda]. \tag{8}$$

Here, $U : A \to \Re$ is a real valued function associating to each arc $(i, j) \in A$ the real number $U(i, j)$, referred to as the *parametric part of the upper bound* of the arc (i, j). Obviously, the nonnegative value $u_0(i, j)$ is the upper bound value of the arc (i, j) computed for $\lambda = 0$, i.e. $\bar{u}(i, j; 0) = u_0(i, j)$. From the above mentioned restriction, it derives that the parametric part of the upper bounds $U(i, j)$ must satisfy the constraint: $U(i, j) \geq (\ell(i, j) - u_0(i, j))/\Lambda$, $\forall (i, j) \in A$. The parametric flow value function $\bar{v} : N \times [0, \Lambda] \to \Re$ associates to each of the nodes $i \in N$ a real number $\bar{v}(i; \lambda)$ called value of node i for each of the parameter λ values.

Definition 2. *A feasible flow in parametric network $\bar{G} = (N, A, \ell, \bar{u}, s, t)$ is called* parametric flow *and is defined as a function $\bar{f} : A \times [0, \Lambda] \to \Re^{+}$ satisfying the following constraints:*

$$\sum_{j|(i,j)\in A} \bar{f}(i, j; \lambda) - \sum_{j|(j,i)\in A} \bar{f}(j, i; \lambda) = \bar{v}(i; \lambda), \; \forall i \in N, \forall \lambda \in [0, \Lambda]. \tag{9}$$

$$\ell(i, j) \leq \bar{f}(i, j; \lambda) \leq \bar{u}(i, j; \lambda), \; \forall (i, j) \in A, \forall \lambda \in [0, \Lambda]. \tag{10}$$

where $\sum_{i\in N} \bar{v}(i; \lambda) = 0$, $\forall \lambda \in [0, \Lambda]$.

The parametric maximum flow (PMaxF) problem is to compute all maximum flows for every possible value of the parameter, i.e. $\forall\lambda \in [0,\Lambda]$:

$$maximize\ \bar{v}(\lambda)\ for\ all\ \lambda \in [0,\Lambda], \tag{11}$$

$$\sum_{j|(i,j)\in A} \bar{f}(i,j;\lambda) - \sum_{j|(j,i)\in A} \bar{f}(j,i;\lambda) = \begin{cases} \bar{v}(\lambda), & i = s \\ 0, & i \neq s,t \\ -\bar{v}(\lambda), & i = t, \end{cases} \tag{12}$$

$$\ell(i,j) \leq \bar{f}(i,j;\lambda) \leq \bar{u}(i,j;\lambda),\ \forall (i,j) \in A, \tag{13}$$

Let F denote the set of piecewise linear functions f_i with $f_i : [0,\Lambda] \to \Re^+$. An ordering relation over the elements of F is defined as: $f_i \leq f_j \rightleftharpoons f_i(\lambda) \leq f_j(\lambda)$ for all $\lambda \in [0,\Lambda]$. If the two piecewise linear functions f_i and f_j have at least a breakpoint, then neither $f_i \leq f_j$ nor $f_i \geq f_j$ hold for the entire interval $[0,\Lambda]$ and consequently, the two functions may not necessarily be comparable. For the case when two piecewise linear functions have a number of K crossing points taking place for the values λ_k, $k = 1,\cdots,K$ of the parameter, the interval $[0,\Lambda]$ can be partitioned into $K+1$ subintervals of the type λ_k, λ_{k+1}, $k = 0,\cdots,K$, with $\lambda_0 = 0$ and $\lambda_{K+1} = \Lambda$, so that within every subinterval one of the following two cases would hold: $f_i \leq f_j$ or $f_i \geq f_j$, i.e. the two linear functions become comparable within subintervals.

Definition 3. *A parametric $s-t$ cut partitioning, denoted by $[S_k; J_k]$, $k = 0,\cdots,K$, is defined as a finite set of cuts $[S_k, T_k]$ together with a partitioning of the interval $[0,\Lambda]$ of the parameter in disjoint subintervals $J_k = [\lambda_k, \lambda_{k+1}]$, so that $J_0 \cup \cdots \cup J_K = [0,\Lambda]$.*

Definition 4. *For the parametric maximum flow problem, the capacity $\tilde{\bar{c}}[S_k; J_k]$ of a parametric $s-t$ cut partitioning is a linear function on every subinterval J_k, $k = 0,\cdots,K$, defined as:*

$$\tilde{\bar{c}}[S_k; J_k] = \sum_{(i,j)\in(S_k,T_k)} \bar{u}(i,j;\lambda) - \sum_{(j,i)\in(T_k,S_k)} \ell(j,i),\ k = 0,\cdots,K. \tag{14}$$

Definition 5. *A parametric $s-t$ cut partitioning $[S_k; J_k]$ with the subintervals J_k assuring that every cut is a minimum cut $[\tilde{S}_k, \tilde{T}_k]$ within the subinterval $[\lambda_k, \lambda_{k+1}]$ is referred to as a parametric minimum $s-t$ cut and is denoted by $[\tilde{S}_k; J_k]$, $k = 0,\cdots,K$.*

Theorem 5. *(Parametric max-flow min-cut theorem [19]): If there is a feasible flow in the parametric network $\bar{G}$, the value function $\tilde{\bar{v}}$ of the parametric maximum flow $\tilde{\bar{f}}$ from a source s to a sink t equals the capacity $\tilde{\bar{c}}$ of the parametric minimum $s-t$ cut $[\tilde{S}_k; J_k]$, $k = 0,\cdots,K$.*

Definition 6. *For the parametric maximum flow problem, the parametric residual capacity $\tilde{\bar{r}}(i,j;\lambda)$ of any of the arcs $(i,j) \in A$ with respect to a given parametric* *flow $\bar{f}$ represents the maximum additional flow that can be sent from node i to node j over the arcs (i,j) and (j,i) and it is given by:*

$$\tilde{\bar{r}}(i,j;\lambda) = \bar{u}(i,j;\lambda) - \bar{f}(i,j;\lambda) + \bar{f}(j,i;\lambda) - \ell(i,j). \tag{15}$$

The subintervals $\tilde{I}(i,j) \subseteq [0,\Lambda]$ for $(i,j) \in A$, where $\tilde{\bar{r}}(i,j;\lambda) > 0$, i.e. an augmentation of the flow $\bar{f}(i,j;\lambda)$ is possible along the arc (i,j), are defined as $\tilde{I}(i,j) = \{\lambda \mid \tilde{\bar{r}}(i,j;\lambda) > 0\}$.

Definition 7. *Given a feasible flow $\bar{f}$ in the parametric network $\bar{G}$, the network denoted by $\tilde{\bar{G}}(\bar{f}) = (N, \tilde{A}(\bar{f}))$ with $\tilde{A}(\bar{f}) = \left\{(i,j) \mid (i,j) \in A, \tilde{I}(i,j) \neq \emptyset\right\}$ being the set consisting only of arcs with positive parametric residual capacities, is referred to as the parametric residual network with respect to the given flow $\bar{f}$ for the parametric maximum flow problem.*

If an arc $(i,j) \in A$ does not belong to $\tilde{\bar{G}}(\bar{f})$, then $\tilde{I}(i,j) := \emptyset$ is set.

Definition 8. *A conditional augmentation directed path is denoted by $\tilde{\bar{P}}$ and is a directed path $\tilde{P}$ from the source s to the sink t in the parametric residual network $\tilde{\bar{G}}(\bar{f})$ with the restriction that:*

$$\tilde{I}(\tilde{\bar{P}}) = \bigcap_{(i,j)\in\tilde{\bar{P}}} \tilde{I}(i,j) \neq \emptyset. \tag{16}$$

Definition 9. *The parametric residual capacity of a conditional augmentation directed path $\tilde{\bar{P}}$ represents the minimum value of the parametric residual capacity $\tilde{\bar{r}}(i,j;\lambda)$ among all arcs (i,j) composing the conditional augmentation directed path for all the parameter λ values in the subinterval $\tilde{I}(\tilde{\bar{P}})$:*

$$\tilde{\bar{r}}(\tilde{\bar{P}};\lambda) = \min_{(i,j)\in\tilde{\bar{P}}} \left\{\tilde{\bar{r}}(i,j;\lambda) \mid \lambda \in \tilde{I}(\tilde{\bar{P}})\right\}. \tag{17}$$

Theorem 6. *(Conditional augmentation path theorem [19]): A parametric flow $\tilde{\bar{f}}$ is a maximum parametric flow if and only if the parametric residual network $\tilde{\bar{G}}(\tilde{\bar{f}})$ contains no conditional augmentation directed path.*

3.2 Parametric minimum flows in static networks

The parametric minimum flow problem can be regarded as a generalisation of the non-parametric, ordinary minimum flow problem for the case of the parametric networks where the lower bounds of some or all arcs $(i,j) \in A$ depend on a nonnegative real parameter λ. For the parametric minimum flow problem, a parametric network denoted by $\bar{G} = (N, A, \bar{\ell}, u, s, t)$ represents a directed network for which the lower bounds of some arcs depend on a real parameter.

Definition 10. *In a parametric network* $\bar{G} = (N, A, \bar{\ell}, u, s, t)$*, the parametric lower bound function* $\bar{\ell} : A \times [0, \Lambda] \to \Re^{+}$ *of an arc* $(i, j) \in A$ *computes the real numbers* $\bar{\ell}(i, j; \lambda)$*, called lower bound of arc* (i, j)*, for all the parameter* λ *values in a given interval* $[0, \Lambda]$*:*

$$\bar{\ell}(i, j; \lambda) = \ell_0(i, j) - \lambda \cdot L(i, j) \leq u(i, j), \ for\ all\ \lambda \in [0, \Lambda]. \tag{18}$$

In the above expression, $L : A \to \Re$ denotes a real valued function which is called the *parametric part of the lower bound* of the arc $(i, j) \in A$ and which must meet the following condition: $[\ell_0(i, j) - u(i, j)]/\Lambda \leq L(i, j) \leq \ell_0(i, j)/\Lambda$, $\forall(i, j) \in A$. Evidently, $\ell_0(i, j)$ represents the value of the function $\bar{\ell}(i, j; \lambda)$ for $\lambda = 0$ and consequently, it must hold that $0 \leq \ell_0(i, j) \leq u(i, j)$, $\forall(i, j) \in A$.

The parametric minimum flow (PMinF) problem consists of solving the nonparametric minimum flow problem for all the parameter values within a certain interval $[0, \Lambda]$:

$$minimize\ \bar{v}(\lambda)\ for\ all\ \lambda \in [0, \Lambda], \tag{19}$$

$$\bar{\ell}(i, j; \lambda) \leq \bar{f}(i, j; \lambda) \leq u(i, j),\ \forall(i, j) \in A, \tag{20}$$

under restrictions (12).

Definition 11. *For the parametric minimum flow problem, the capacity* $\hat{\bar{c}}[S_k; J_k]$ *of a parametric* $s - t$ *cut partitioning is a linear function on every subinterval* J_k*,* $k = 0, \cdots, K$*, defined as:*

$$\hat{\bar{c}}[S_k; J_k] = \sum_{(i,j) \in (S_k, T_k)} \bar{\ell}(i, j; \lambda) - \sum_{(j,i) \in (T_k, S_k)} u(j, i),\ k = 0, \cdots, K. \tag{21}$$

Definition 12. *A parametric* $s-t$ *cut partitioning* $[S_k; J_k]$ *with the subintervals* J_k *assuring that every cut is a maximum cut* $[\hat{S}_k, \hat{T}_k]$ *within the subinterval* $[\lambda_k, \lambda_{k+1}]$ *is referred to as a parametric maximum* $s - t$ *cut and is denoted by* $[\hat{S}_k; J_k]$*,* $k = 0, \cdots, K$*.*

Theorem 7. *(Parametric min-flow max-cut theorem [19]): If there is a feasible flow in the parametric network* $\bar{G}$*, the value function* $\hat{\bar{v}}$ *of the parametric minimum flow* $\hat{\bar{f}}$ *from a source* s *to a sink* t *equals the capacity* $\hat{\bar{c}}$ *of the parametric maximum* $s - t$ *cut* $[\hat{S}_k; J_k]$*,* $k = 0, \cdots, K$*.*

Definition 13. *For the parametric minimum flow problem, the parametric residual capacity* $\hat{\bar{r}}(i, j; \lambda)$ *of any of the arcs* $(i, j) \in A$ *with respect to a given parametric flow* $\bar{f}$ *represents the maximum amount by which the flow sent from node* i *to node* j *can be decreased over the arcs* (i, j) *and* (j, i) *and it is given by:*

$$\hat{\bar{r}}(i, j; \lambda) = u(j, i) - \bar{f}(j, i; \lambda) + \bar{f}(i, j; \lambda) - \bar{\ell}(i, j; \lambda). \tag{22}$$

The subintervals $\hat{I}(i, j) \subseteq [0, \Lambda]$ for $(i, j) \in A$, where $\hat{\bar{r}}(i, j; \lambda) > 0$, i.e. a decrease of the flow $\bar{f}(i, j; \lambda)$ is possible along the arc (i, j), are defined as $\hat{I}(i, j) = \{\lambda \mid \hat{\bar{r}}(i, j; \lambda) > 0\}$.

Definition 14. *Given a feasible flow* $\bar{f}$ *in the parametric network* $\bar{G}$*, the network denoted by* $\hat{\hat{G}}(\bar{f}) = (N, \hat{A}(\bar{f}))$ *with* $\hat{A}(\bar{f}) = \left\{(i,j) \mid (i,j) \in A, \hat{I}(i,j) \neq \emptyset\right\}$ *being the set consisting only of arcs with positive parametric residual capacities, is referred to as the parametric residual network with respect to the given flow* $\bar{f}$ *for the parametric minimum flow problem.*

Definition 15. *A conditional decreasing directed path is denoted by* $\hat{\hat{P}}$ *and is a directed path* $\hat{P}$ *from the source* s *to the sink* t *in the parametric residual network* $\hat{\hat{G}}(\bar{f})$ *with the restriction that:*

$$\hat{I}(\hat{\hat{P}}) = \bigcap_{(i,j)\in\hat{\hat{P}}} \hat{I}(i,j) \neq \emptyset. \tag{23}$$

Definition 16. *The parametric residual capacity of a conditional decreasing directed path* $\hat{\hat{P}}$ *represents the minimum value of the parametric residual capacity* $\hat{\hat{r}}(i,j;\lambda)$ *among all arcs* (i,j) *composing the conditional decreasing directed path for all the parameter* λ *values in the subinterval* $\hat{I}(\hat{\hat{P}})$*:*

$$\hat{\hat{r}}(\hat{\hat{P}};\lambda) = \min_{(i,j)\in\hat{\hat{P}}} \left\{\hat{\hat{r}}(i,j;\lambda) \mid \lambda \in \hat{I}(\hat{\hat{P}})\right\}. \tag{24}$$

Theorem 8. *(Conditional decreasing path theorem [19]): A parametric flow* $\hat{\hat{f}}$ *is a minimum parametric flow if and only if the parametric residual network* $\hat{\hat{G}}(\hat{\hat{f}})$ *contains no conditional decreasing directed path.*

3.3 Algorithmic approaches for the parametric flows in static networks

The current section presents several different approaches that have been proposed for solving the parametric flow problem in static networks.

The "Shortest conditional decreasing path algorithm" for the parametric minimum flow problem [7] determines in each stage an improvement of the flow over the subinterval of the parameter values which derives directly from the shortest conditional decreasing directed path in the parametric residual network. In its first phase, the algorithm establishes a feasible flow, if such a flow exists in the given parametric network. In the second phase, the algorithm repeatedly searches a shortest conditional decreasing directed path and, when one is found, the flow is decreased along the corresponding paths in the original parametric network and the parametric residual network is updated. As soon as the sink node cannot be reached from the source, the algorithm stops and the obtained flow represents a parametric minimum flow.

SHORTEST CONDITIONAL DECREASING PATH ALGORITHM
BEGIN
  find a feasible flow f_0 in network $\bar{G} = \{N, A, \bar{\ell}, u, s, t\}$;
  compute the parametric residual network $\hat{\bar{G}}(f_0)$;
  REPEAT
    compute the minimum length (h) of a conditional decreasing directed path;
    IF (exists $(\hat{I}_{n-1,h} \neq \emptyset)$ within the flow can be improved) THEN
      BEGIN
        build the conditional decreasing directed path $\hat{\bar{P}}$;
        compute the parametric residual capacity $\hat{\bar{r}}(\hat{\bar{P}}; \lambda)$;
        update the parametric residual network ;
      END;
  UNTIL $(h = n - 1)$ and $(\hat{I}_{n-1,h} = \emptyset)$;
END.

The "SEQUENTIAL ALGORITHM" for the parametric minimum flow problem [6] is based on a "*piece-by-piece*" approach which seeks, in each of its steps, a maximum improvement of the flow and the next maximum interval of the parameter ensuring that the computed flow is a parametric minimum flow. Assuming that the minimum flow is known for a given value of the parameter, the following two subproblems have to be solved: computing the maximum amount of flow that has to be subtracted from the current flow in order for this to preserve its optimality for greater parameter values; computing the maximum value of the parameter for which the newly computed flow remains optimal. The algorithm consists of applying a non-parametric, ordinary maximum flow algorithm for a sequence of parameter values in increasing order. An initial minimum flow is computed for a given value of the parameter and then the algorithm repeatedly finds a maximum amount by which the flow can be decreased over the next interval of the parameter values so that the maximum cut does not change. This maximum amount of flow is computed as a maximum flow in a derived network G_k^* with properly set lower and upper bounds:

$$\begin{array}{ll} \ell_k^*(i,j) = 0 & \text{for} \quad \hat{f}(i,j;\lambda_k) = u(i,j); \\ \ell_k^*(i,j) = -\infty & \text{for} \quad \hat{f}(i,j;\lambda_k) < u(i,j). \end{array}$$

$$\begin{array}{ll} u_k^*(i,j) = L(i,j) & \text{for} \quad \hat{f}(i,j;\lambda_k) = \ell(i,j;\lambda_k); \\ u_k^*(i,j) = \infty & \text{for} \quad \hat{f}(i,j;\lambda_k) > \ell(i,j;\lambda_k). \end{array}$$

On each of its iterations, the algorithm computes a new breakpoint of the piecewise linear minimum flow value function and the corresponding parametric minimum flow.

SEQUENTIAL ALGORITHM FOR PARAMETRIC MINIMUM FLOW
BEGIN
  find a feasible flow f_0 in network $\bar{G}_0 = \{N, A, \bar{\ell}(\lambda = 0), u, s, t\}$;
  $k := 0$; $\lambda_k := 0$;
  compute the MINIMUM flow $\hat{f}_k$ in network $\bar{G}_0$;
  WHILE $\lambda_k < \Lambda$ DO
    BEGIN
      compute the derived network G^*_k ;
      compute the MAXIMUM flow $\tilde{f}^*_k$ in network G^*_k;
      compute the parameter subinterval δ_k that maintains the optimality of $\hat{f}_k$;
      $\hat{f}_{k+1} := \hat{f}_k - \delta_k \cdot \tilde{f}^*_k$;
      $\lambda_{k+1} := \lambda_k + \delta_k$;
      $k := k + 1$;
    END
END.

A parametric bipartite network is called monotone if the lower bounds of the out of the source arcs are non-increasing functions of a parameter, the lower bounds of the arcs into the sink are non-decreasing functions of the parameter, while the lower bounds of the remaining arcs are constants. The "BALANCING ALGORITHM" for the minimum flow problem in monotone parametric bipartite networks decreases the flow over simple decreasing directed paths. The proposed algorithm [8] does not work directly in the original network but in the parametric residual network and finds a particular state of the residual network from which the minimum flow and the maximum cut for any of the parameter values are obtained. The approach implements a round-robin algorithm looping over a list of nodes until an entire pass ends without changes of the flow.

The "PARTITIONING ALGORITHM" for the parametric maximum flow [18] / minimum flow [17] problem represents an original approach of the directed paths type algorithms. From a feasible flow, established in the first stage of the algorithm, the partitioning algorithm finds, on every iteration of its second stage, an augmentation/decreasing directed path from the source node to the sink node in a parametric residual network defined only for a subinterval of the parameter values, it improves the flow along the corresponding paths in the original parametric network and splits the interval of the parameter values into subintervals which are generated by the breakpoints of the piecewise linear parametric residual capacity function of the augmentation/decreasing directed path. Further on, the algorithm reiterates within each of the generated subintervals, in increasing order of the parameter values.

PARTITIONING ALGORITHM
BEGIN
find a feasible flow f_0 in network $\bar{G}$;
$k := 0$; $\lambda_k := 0$;
REPEAT
compute the parametric residual network $\bar{G}_k(f_0)$;
$\lambda_{k+1} := \Lambda$;
WHILE (exists a directed path $\bar{P}$ in network $\bar{G}_k(f_0)$) DO
BEGIN
build a directed path $\bar{P}$ in network $\bar{G}_k(f_0)$;
compute the parametric residual capacity $\bar{r}(\bar{P})$;
compute the upper limit λ_{k+1} that maintain the linearity of $\hat{f}_k$;
update the the parametric residual network $\bar{G}_k(\bar{f}_k)$;
END;
compute the optimal flow for $J_k = [\lambda_k, \lambda_{k+1}]$;
$k := k + 1$;
UNTIL $(\lambda_k = \Lambda)$;
END.

The "PARAMETRIC MIN-MAX ALGORITHM" [19] solves the minimum flow problem in a parametric network with linear lower bound functions by computing a parametric maximum flow from the sink node to the source node. Given a feasible flow, a minimum flow from the source node to the sink node can be determined by establishing a maximum flow from the sink node to the source node in the residual network defined as for the parametric maximum flow problem. The algorithm does not work directly in the original parametric network but in the parametric residual network defined as for the parametric maximum flow problem.

PARAMETRIC MIN-MAX ALGORITHM
BEGIN
find a feasible flow f_0 in network $\bar{G}$;
compute the parametric residual network $\tilde{\bar{G}}(f_0)$;
compute the parametric MAXIMUM flow $\bar{f}^* := \tilde{\bar{f}}$ from t to s in $\tilde{\bar{G}}(f_0)$;
$\hat{\bar{f}} := \bar{f}^*$ is a parametric MINIMUM flow from s to t in network $\bar{G}$;
END.

4 Example

Further on, we will illustrate the partitioning algorithm for the parametric maximum flow in the static network presented in Figure 1 with the source node $s = 0$ and the sink node $t = 3$. The parameter λ takes values in the interval $[0, 1]$, i.e. $\Lambda = 1$.

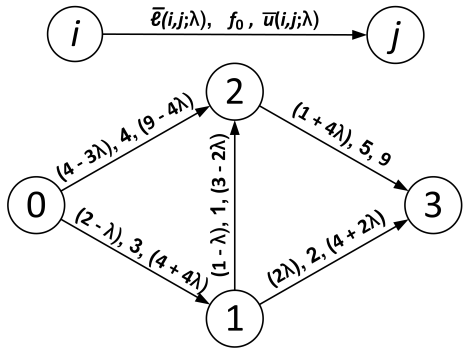

Figure 1: The parametric network $\bar{G} = (N, A, \bar{\ell}, \bar{u}, s, t)$. Above each arc (i, j), the parametric lower bound function $\bar{\ell}(i, j; \lambda)$, the feasible flow f_0 and the parametric upper bound function $\bar{u}(i, j; \lambda)$ are indicated.

Considering the feasible flow f_0 which is indicated in Figure 1, the residual network $\tilde{\tilde{G}}(f_0)$ for the parametric maximum flow problem and the residual network $\tilde{\hat{G}}(f_0)$ for the parametric minimum flow problem, with $\lambda_0 = 0$ and $\lambda_1 = \Lambda$, are presented in Figure 2. The residual capacity of every arc is written as $\tilde{\tilde{r}}(i, j; \lambda) = \tilde{\alpha}(i, j) + \lambda \cdot \tilde{\beta}(i, j)$. Here, $\tilde{\alpha}(i, j) = \tilde{\tilde{r}}(i, j; 0)$ and $\tilde{\beta}(i, j) = U(i, j)$.

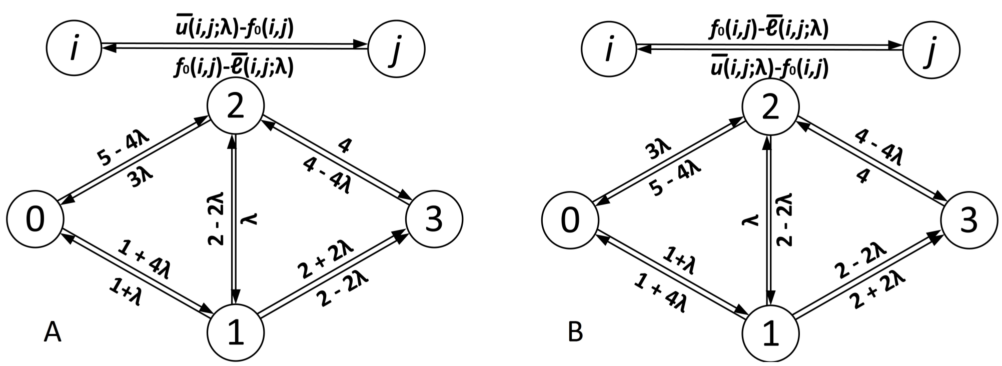

Figure 2: A. The residual network $\tilde{\tilde{G}}(f_0)$ for the parametric maximum flow problem; B. The residual network $\tilde{\hat{G}}(f_0)$ for the parametric minimum flow problem.

In the parametric residual network $\tilde{\tilde{G}}(f_0)$, (see. Figure 2.A) the directed path $\tilde{\tilde{P}} = (0, 1, 3)$ is built with the parametric residual capacity $\tilde{\tilde{r}}_0(\tilde{\tilde{P}}) = 1 + 4\lambda$, i.e. $\tilde{\alpha}_0(\tilde{\tilde{P}}) = 1$ and $\tilde{\beta}_0(\tilde{\tilde{P}}) = 4$. Since for the value $\lambda^* = 1/2$, the parametric residual capacity $\tilde{\tilde{r}}_0(\tilde{\tilde{P}}) = 1 + 4\lambda$ of the directed path crosses the parametric residual capacity $\tilde{\tilde{r}}_0(1, 3; \lambda)$ of the arc $(1, 3)$ and this parameter value respects the restriction $\lambda^* \leq \Lambda$, the upper limit of the subinterval of the parameter values is updated to $\lambda_1 := \lambda^* = 1/2$.

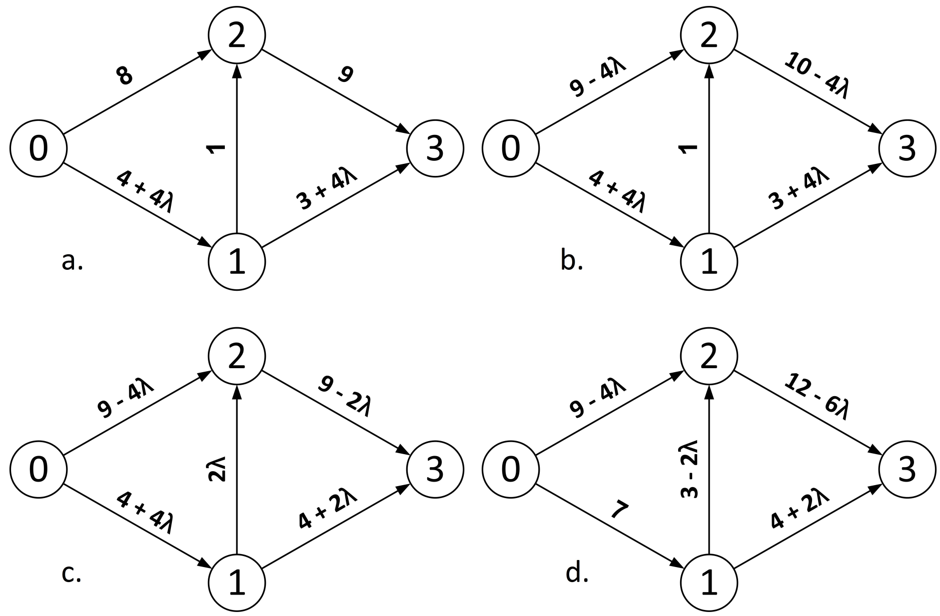

Figure 3: The parametric maximum flow for each of the subintervals J_k, $k = 0, 1, 2, 3$ of the parameter values: (a) $J_0 = [0, 1/4]$; (b) $J_1 = [1/4, 1/2]$; (c) $J_2 = [1/2, 3/4]$; (d) $J_3 = [3/4, 1]$.

In the next step, the parametric residual network $\tilde{\tilde{G}}_0(f_0)$ is updated, i.e. the values $\tilde{\alpha}_0(i, j)$ and $\tilde{\beta}_0(i, j)$ are updated for both arcs $(1, 3)$ and $(0, 1)$. Afterwards, in a similar way, the new directed path $\tilde{\tilde{P}} = (0, 2, 3)$ is built and the value of λ_1 is updated to $\lambda_1 := \lambda^* = 1/4 \leq 1/2$. Since at this stage no other directed path can be found, the maximum parametric flow $\tilde{\tilde{f}}_0$ is computed for the parameter values in the subinterval $J_0 = [\lambda_0, \lambda_1] = [0, 1/4]$ (see Figure 3.a) and the value of the counter is incremented to $k := 1$.
Because $\lambda_1 \neq \Lambda$, the algorithm reiterates within the new interval $[1/4, 1]$. After performing three more iterations, the algorithm ends with the parametric maximum flow $\tilde{\tilde{f}}_k$, separately computed for the subintervals $J_1 = [1/4, 1/2]$, $J_2 = [1/2, 3/4]$, $J_3 = [3/4, 1]$, as shown in Figures 3.b, c and d.
Now, considering the lower bounds of the parametric network in Figure 1 and the corresponding residual network for the parametric minimum flow, a similar algorithm is executed for computing a parametric minimum flow.
In the parametric residual network $\hat{\hat{G}}(f_0)$, (see. Figure 2.B) the directed path $\hat{\hat{P}} = (0, 1, 3)$ is built with the parametric residual capacity $\hat{\hat{r}}_0(\hat{\hat{P}}) = 1 + \lambda$. For the value $\lambda^* = 1/3$, the parametric residual capacity $\hat{\hat{r}}_0(\hat{\hat{P}}) = 1 + \lambda$ of the directed path crosses the parametric residual capacity $\hat{\hat{r}}_0(1, 3; \lambda)$ of the arc $(1, 3)$. The computed value of the parameter respects the restriction $\lambda^* \leq \Lambda$ and consequently, the upper limit of the subinterval of the parameter values is updated to $\lambda_1 := \lambda^* = 1/3$. According to the parametric residual capacity $\hat{\hat{r}}_0(\hat{\hat{P}})$, the residual network $\hat{\hat{G}}_0(f_0)$ is updated for both arcs $(1, 3)$ and $(0, 1)$. Then, in a similar way,

the new directed path $\hat{\hat{P}} = (0, 2, 3)$ is built and the value $\lambda^* = 4/7$ is found but because the restriction $\lambda^* \leq \lambda_1$ i.e. $4/7 \leq 1/3$ does not hold, the value $\lambda_1 = 1/3$ is maintained. Since at this stage no other directed path can be found, the minimum parametric flow $\hat{\hat{f}}_0$ is computed for the parameter values in the subinterval $J_0 = [\lambda_0, \lambda_1] = [0, 1/3]$ (see Figure 4.a) and the value of the counter is incremented to $k := 1$.

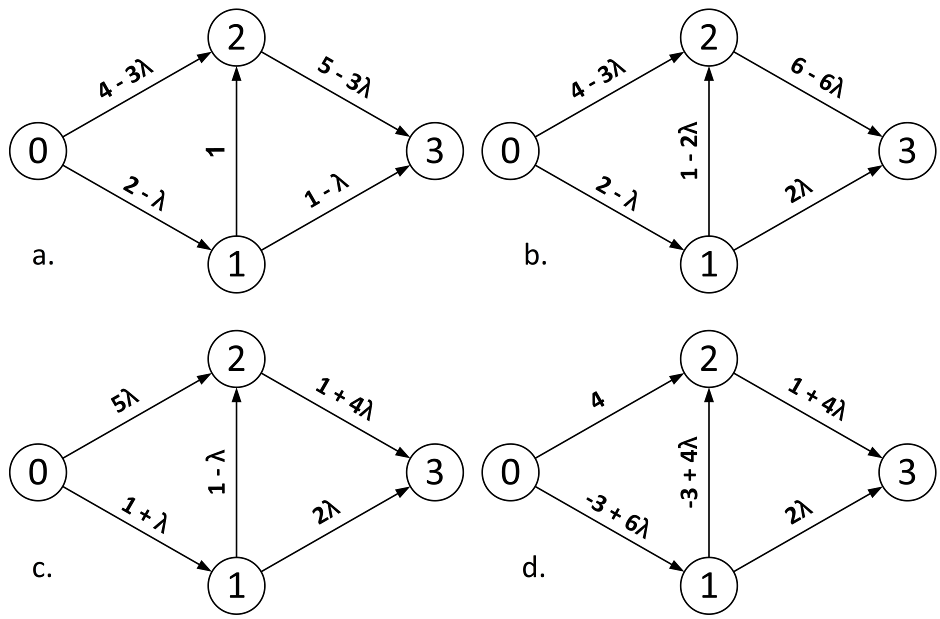

Figure 4: The parametric minimum flow for each of the subintervals J_k, $k = 0, 1, 2, 3$ of the parameter values: (a) $J_0 = [0, 1/3]$; (b) $J_1 = [1/3, 1/2]$; (c) $J_2 = [1/2, 4/5]$; (d) $J_3 = [4/5, 1]$.

For the new subinterval $\lambda_1 \neq \Lambda$, the algorithm reiterates within the new interval $[1/3, 1]$. The directed paths $\hat{\hat{P}} = (0, 1, 3)$, $\hat{\hat{P}} = (0, 1, 2, 3)$ and $\hat{\hat{P}} = (0, 2, 3)$ are consecutively built and parameter value λ_2 is updated to the value $\lambda = 1/2$. For the subinterval $J_1 = [\lambda_1, \lambda_2] = [1/3, 1/2]$ the minimum parametric flow $\hat{\hat{f}}_1$ is computed (see Figure 4.b) and the value of the counter is incremented to $k := 2$. After performing two more iterations, the algorithm ends with the parametric minimum flow $\hat{\hat{f}}_k$, separately computed for the subintervals $J_2 = [1/2, 4/5]$ and $J_3 = [4/5, 1]$, as shown in Figures 4.c and d.
The piecewise linear flow value function $\bar{v}(\lambda)$, computed according to equation (12), is presented Figure 5, both for the parametric maximum flow $\tilde{\tilde{f}}$, denoted by $\tilde{\tilde{v}}(\lambda)$, and for the parametric minimum flow $\hat{\hat{f}}$, denoted by $\hat{\hat{v}}(\lambda)$. These flow value functions have been computed for the parametric network $\bar{G} = (N, A, \bar{\ell}, \bar{u}, s, t)$ shown in Figure 1 for the whole range of values of the parameter $\lambda \in [0, \Lambda]$. As it can be easily seen, even if for some of the parameter values the functions $\bar{v}(\lambda)$ do not change their slopes, the parametric maximum or minimum flows distribute differently over the network arcs.

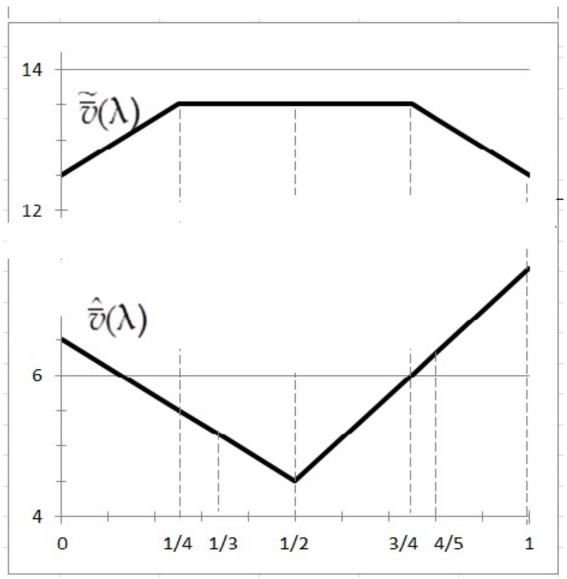

Figure 5: The piecewise linear maximum ($\tilde{\bar{v}}(\lambda)$) and minimum ($\hat{\bar{v}}(\lambda)$) flow value functions for the parametric network $\bar{G} = (N, A, \bar{\ell}, \bar{u}, s, t)$.

5 Concluding remarks

The main advantage of the partitioning approach consists in operating with linear functions instead of the difficult to handle piecewise linear functions, i.e. the residual capacity of every arc in the residual network is explicitly written as a linear function. Even though considering both the upper bounds and the lower ones as parametric linear functions instead of constants, the parametric residual capacity of every arc in the residual network still remains written as a linear function, allowing the running of the algorithm in a similar manner.

Based on the fact that the partitioning approach for the parametric flow problem is based on algorithms which work in the parametric residual network, the changes that follow from taking into account both upper and lower parametric bounds are equally treated and thus, easy to be dealt with. Consequently, algorithms presented can be extended to networks with both lower and upper parametric bounds. From the above considerations it result that the partitioning algorithms remain valid (with appropriate modifications) for the cases of parametric both upper bounds and lower bounds.

References

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

[2] Ahuja,R.K., Orlin,J.B., Stein,C., Tarjan,R.E., *Improved algorithms for bipartite network flow*, SIAM Journal on Computing **23** (1994), No.5, 906-933

[3] Ahuja,R.K., Orlin,J.B., *Distance-Directed Augmenting Path Algorithms for Maximum Flow and Parametric Maximum Flow Problems*, Naval Research Logistics, **38** (1990), 413-430

[4] Bichot,C-E., Siarry,P., *Graph Partitioning: Optimisation and Applications*, ISTE Wiley, 2011

[5] Balinski,M.L., *On a selection problem*, Management Science **17** (1970), No.3, 230-231

[6] Ciurea,E., Parpalea,M., *A sequential algorithm for finding the solution of the parametric minimum flow problem*, Carpathian Journal of Mathematics **28** (2012), No.1, 47-58

[7] Ciurea,E., Parpalea,M., *Shortest conditional decreasing path algorithm for the parametric minimum flow problem*, Bulletin Mathmatique de la Socit des Sciences Mathmatiques de Roumanie, **56(104)** (2013), No.4, 387-401

[8] Ciurea,E., Parpalea,M., Balancing Algorithm for the Minimum Flow Problem in Parametric Bipartite Networks in: *Latest Trends on Computers (vol.I)*, 14th WSEAS International Conference on Computers (part of the 14th WSEAS CSCC Multiconference), Corfu Island, Greece (2010) 226-231

[9] Eisner,M.J., Severance,D.G., *Mathematical techniques for efficient record segmentation in large shared databases*, J. ACM **23** (1976), No.4, 619-635

[10] Gallo,G., Grigoriadis,M.D., Tarjan,R.E., *A fast parametric maximum flow algorithm and applications*, SIAM Journal of Computing **18** (1989), No.1, 30-55

[11] Goldberg,A.V., Rao,S., *Beyond the flow decomposition barrier*, J. ACM **45** (1989), No.5, 783-797

[12] Goldberg,A.V., Tarjan,R.E., *A new approach to the maximum-flow problem*, J.ACM **35** (1988), No.4, 921-940

[13] Hamacher,H.W., Foulds,L.R., *Algorithms for flows with parametric capacities*, ZOR-Methods and Models of Operations Research **33** (1989) 21-37

[14] Hochbaum,D.S., The Pseudoflow Algorithm and the Pseudoflow-Based Simplex for the Maximum Flow Problem in: Bixby,R.E., Boyd,E.A., Rios-Mercado,R.Z. (eds.) *Integer Programming and Combinatorial Optimization*, LNCS **1412**, Springer, Heidelberg (1998), 325-337

[15] King,V., Rao,S., Tarjan,R.E., *A Faster Deterministic Maximum Flow Algorithm*, J.Algorithms **17** (1994), 447-474

[16] Mamer,J., Smith,S., *Optimizing field repair kits based on job completion rate*, Management Science **28** (1982), No.11, 1328-1333

[17] Parpalea,M., Ciurea,E., *Minimum Parametric Flow-A Partitioning Approach*, British Journal of Applied Science & Technology **13** (2015), Issue.6, 1-8

[18] Parpalea,M., Ciurea,E., *Partitioning Algorithm for the Parametric Maximum Flow*, Applied Mathematics **4** (2013), Special Issue on Computer Mathematics, No.10A, 3-10

[19] Parpalea,M., *Min-Max Algorithm for the Parametric Flow Problem*, Bulletin of the Transilvania University of Braov **3(52)**, Series III: Mathematics, Informatics, Physics, (2010) 191-198.

[20] Rhys,J.M.W., *A selection problem of shared fixed costs and network flows*, Management Science 17(3), 200-207 (1970)

[21] Ruhe,G., *Characterization of all optimal solutions and parametric maximal flows in networks*, Optimization, 16(1), 51-61, 1985.

[22] Ruhe,G., *Complexity results for multicriterial and parametric network flows using a pathological graph of Zadeh*, Zeitschrift fr Operations Research, 32, 9-27, 1988.

[23] Tarjan,R.E., Ward,J., Zhang,B., Zhou,Y., Mao,J., Balancing Applied to Maximum Network Flow Problems in: Azar,Y., Erlebach,T. (eds.) *European Symposium of Algorithms 2006*, Lecture Notes in Computer Science **4168** (2016), Springer, Berlin, Heidelberg, 612-623

[24] Zhang,B., *A new balancing method for solving parametric maximum flow problems*, Stanford EE, Computer Systems Colloquium, 2007

[25] Zhang,B., Ward,J., Feng,Q., *A simultaneous parametric maximum flow algorithm with vertex balancing*, Technical Report HPL-2005-121, HP Labs, 2005