British Journal of Applied Science & Technology
13(6): 1-8, 2016, Article no.BJAST.22636
ISSN: 2231-0843, NLM ID: 101664541

SCIENCEDOMAIN *international*
www.sciencedomain.org

Minimum Parametric Flow – A Partitioning Approach

Mircea Parpalea[1*] and Eleonor Ciurea[2]

[1]National College Andrei Şaguna, Braşov, Romania.
[2]Department of Computer Science, Transilvania University of Braşov, Romania.

Authors' contributions

This work was carried out in collaboration between authors. Both authors designed, prepared and approved the final manuscript.

Article Information

DOI: 10.9734/BJAST/2016/22636
Editor(s):
(1) Samir Kumar Bandyopadhyay, Department of Computer Science and Engineering, University of Calcutta, India.
(2) Singiresu S. Rao, Prof. at Department of Mechanical and Aerospace Engineering, University of Miami, Coral Gables, USA.
Reviewers:
(1) Ahmet Sayar, Kocaeli University, Turkey.
(2) Luiza Amalia Pinto Cantao, Universidade Estadual Paulista, Brazil.
(3) S. S. Shinde, Savitribai Phule Pune University, India.
(4) Anonymous, Southern Illinois University, Carbondale, USA.
Complete Peer review History: http://sciencedomain.org/review-history/12820

Method Article

Received 15th October 2015
Accepted 9th December 2015
Published 28th December 2015

ABSTRACT

The present paper proposes a partitioning type approach for the parametric minimum flow problem which is based on the classical decreasing directed paths method. On each of its iterations, the algorithm finds a decreasing directed path from source node to sink node in a range of parametric residual networks which are consecutively defined for subintervals of the parameter values and, by decreasing the flow along the corresponding paths in the original parametric network, splits the interval of the parameter values in subintervals generated by the breakpoints of the piecewise linear parametric residual capacity function of the decreasing directed path. Further on, the algorithm reiterates for every generated subinterval in increasing order of the parameter values.

Keywords: Minimum flow; parametric network; decreasing paths.

1. INTRODUCTION

The problem of the parametric maximum flow with zero lower bounds and linear capacity functions has constantly been investigated and several algorithms exist (*e.g*. Hamacher and Foulds [1], Ruhe [2,3], Gallo et al. [4] or Zhang et al. [5,6]) to solve different instances the problem. Although it has its own applications, the parametric minimum flow problem was

[*]Corresponding author: E-mail: parpalea@gmail.com, e.ciurea@unitbv.ro;

addressed in literature considerably less often than the parametric maximum flow problem. Actually, the parametric minimum flow problem (or, in general any parametric flow problem) represents a kind of generalisation of the classical nonparametric problem to the case where the lower bounds (or correspondingly the upper bounds) of some arcs depend of a certain parameter. Consequently, the problem consists in solving the nonparametric minimum flow problem for all the parameter values within a certain interval. If all the lower bound functions linearly depend of the parameter, the minimum parametric flow value function will result in a continuous piecewise linear function of the same parameter. The partitioning type approach, which is presented in this paper, proposes an original algorithm for computing the minimum flow in networks with linear upper bound functions. As Bichot and Siarry [7] 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" [7].

The structure of this article is the following: Section 2 gathers some basic terminology elements regarding the network flow problem. The terminology and definitions in this section are taken from [8]. Section 3 reminds some necessary definitions regarding the parametric minimum flow problem. Section 4 suggests a possible application of the parametric minimum flow problem. Section 5 describes the proposed algorithm which finds a parametric minimum flow. Section 6 gives an example of a parametric network with linear lower bounds functions in order to show the evolution of the proposed algorithm. Finally, Section 7 presents some conclusions and a generalization of the problem.

2. TERMINOLOGY AND PRELIMINARIES

Let $G=(N,A,\ell,u,s,t)$ be a capacitated network with $n=|N|$ nodes $N=\{\ldots,i,\ldots\}$ and $m=|A|$ arcs $A=\{\ldots,a,\ldots\}$. The set of nodes N contains two special nodes: the source node *s* and the sink node *t*. If an arc $a\in A$ connects nodes $i,j\in N$ then $a=(i,j)$. For every arc $a=(i,j)\in A$ two nonnegative real functions are defined: the *upper bound* $u(a)$ and *lower bound* $\ell(a)$. A *flow* is a function $f:A\rightarrow\Re^+$ satisfying the conditions:

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

In the previous equation (1), $v\geq 0$ is referred to as the value of the flow f.

A *feasible flow* if a function $f:A\rightarrow\Re^+$ satisfying the flow bound constraints: $\ell(i,j)\leq f(i,j)\leq u(i,j)$, for every arc $(i,j)\in A$. A *cut*, denoted [*S*, *T*], is a partition of the set of nodes into two subsets *S* and *T*=*N*−*S*. An arc (i,j) is called a *forward arc* of the cut if $i\in S$ and $j\in T$. If $i\in T$ and $j\in S$ then the arc (i,j) is called a *backward arc* of the cut. The set of forward arcs of the cut is denoted (*S*, *T*) and (*T*, *S*) denotes the set of backward arcs. A cut [*S*, *T*] is called an *s*−*t* cut if $s\in S$ and $t\in T$. The minimum flow problem consists in finding a flow which minimises its value. For a network with nonzero lower bounds, the problem is usually solved in two stages: (1) building a feasible flow in the network; (2) starting from a given feasible flow, establishing the minimum flow.

3. PARAMETRIC MINIMUM FLOW

The parametric minimum flow problem can be regarded as a generalisation of the nonparametric problem where the lower bounds of some arcs depend of a nonnegative, real parameter λ:

$$\overline{\ell}(i,j;\lambda)=\ell_0(i,j)-\lambda\cdot\pounds(i,j) \tag{2}$$

In expression (2), by $\pounds(i,j)$ is denoted a real valued function called the *parametric part of the lower bound* of the arc (i,j). The significance of $\ell_0(i,j)$ is the value of $\overline{\ell}(i,j;\lambda)$ for $\lambda=0$, observing the condition that $0\leq\ell_0(i,j)\leq u(i,j)$. The same restriction also holds for the lower bound of every arc $(i,j)\in A$, *i.e.*

$0 \le \bar{\ell}(i,j;\lambda) \le u(i,j)$, $\forall \lambda \in [0,\Lambda]$. This results in: $(\ell_0(i,j) - u(i,j))/\Lambda \le \pounds(i,j) \le \ell_0(i,j)/\Lambda$, $\forall\, (i,j) \in A$.

Definitions 1 to 7, which follow in this section, are adapted from reference [8] while theorem 1 is taken from reference [9].

Definition 1. *A PARAMETRIC NETWORK denoted $\bar{G} = (N, A, \bar{\ell}, u, s, t)$, is a directed network with some arcs having lower bounds which depend of a real parameter.*

Definition 2. *The PARAMETRIC MINIMUM FLOW problem consists in solving the nonparametric minimum flow problem for all the parameter values within a certain interval*:

$$minimize \quad \bar{v}(\lambda) \quad for\ all \quad \lambda \in [0,\Lambda] \tag{3.a}$$

with

$$\sum_{j|(i,j)\in A} \bar{f}(i,j;\lambda) - \sum_{j|(j,i)\in A} \bar{f}(j,i;\lambda) = \begin{cases} \bar{v}(\lambda), & i = s \\ 0, & i \ne s,t \\ -\bar{v}(\lambda), & i = t \end{cases} \tag{3.b}$$

and

$$\ell(i,j;\lambda) \le f(i,j;\lambda) \le u(i,j) \quad \forall\ (i,j) \in A \quad . \tag{3.c}$$

Definition 3. *For the parametric minimum flow problem, the PARAMETRIC RESIDUAL CAPACITY $\bar{r}'(i,j;\lambda)$ 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 from node i to node j can be diminished over the arcs (i,j) and (j,i)*. It is given by:

$$\bar{r}'(i,j;\lambda) = u(j,i) - \bar{f}(j,i;\lambda) + \bar{f}(i,j;\lambda) - \bar{\ell}(i,j;\lambda). \tag{4}$$

Definition 4. *The PARAMETRIC RESIDUAL NETWORK of a parametric network $\bar{G}$, with respect to a feasible flow $\bar{f}$, is the network denoted $\bar{G}'(\bar{f}) = (N, A'(\bar{f}))$.*

In the previous definition (4), $A'(\bar{f})$ represents the set of only those arcs having $\bar{r}'(i,j;\lambda) > 0$ for a subinterval of the parameter values.

Definition 5. *The subinterval in which the flow can be diminished over an arc (i,j) is denoted $I(i,j) \subseteq [0,\Lambda]$:*

$$I(i,j) = \{\lambda \mid \bar{r}'(i,j;\lambda) > 0\} \quad for\ (i,j) \in A'(\bar{f}) \quad . \tag{5}$$

Definition 6. *A directed path P from source to sink in network $\bar{G}'(\bar{f})$, is called CONDITIONAL DECREASING DIRECTED PATH $\bar{P}$ if it meets the restriction $I(\bar{P}) = \bigcap_{(i,j)\in\bar{P}} I(i,j) \ne \phi$.*

Definition 7. *The PARAMETRIC RESIDUAL CAPACITY OF A CONDITIONAL DECREASING DIRECTED PATH is denoted $\bar{r}'(\bar{P})$ and is defined as:*

$$\bar{r}'(\bar{P}) = \min_{\lambda \in I(\bar{P})} \{\bar{r}'(i,j;\lambda) \mid (i,j) \in \bar{P}\}. \tag{6}$$

In general, both $\bar{r}'(i,j;\lambda)$ and $\bar{r}'(\bar{P})$ are piecewise linear functions. Denoting by $K(i,j)$ the number of linear segments of $\bar{r}'(i,j;\lambda)$ and by $K(\bar{P})$ the equivalent number for $\bar{r}'(\bar{P})$, results that $K(\bar{P}) \ge \sum_{(i,j)\in\bar{P}} K(i,j)$. Since any $\bar{P}$ is elementary, follows that: $K(\bar{P}) \le n-2$.

Theorem 1. [9] (CONDITIONAL DECREASING PATH THEOREM) *A flow $\bar{f}_{\min}$ is a parametric minimum flow if and only if the parametric residual network $\bar{G}'(\bar{f}_{\min})$ contains no conditional decreasing directed path from the source node to the sink node.*

Based on the optimal parametric residual network $\bar{G}'(\bar{f}_{\min})$, the parametric minimum flow is computed as:

$$\bar{f}_{\min}(i,j;\lambda) = \bar{\ell}(i,j;\lambda) + max\,\{\bar{r}'_{\min}(i,j;\lambda) - u(j,i) + \bar{\ell}(j,i;\lambda), \quad 0\}. \tag{7}$$

4. APPLICATIONS

Considering that the problem of the parametric minimum flow represents an extension of the classical problem of the minimum flow, the applications of the parametric minimum flows cover all those instances of the classical flow problem where network characteristics are depending linearly of a parameter. This subsection briefly presents the problem of scheduling some works on different machines. The problem has many practical applications if the machines are considered to be workers, oil ship, freight trucks, wagons, airplanes or even processors etc.

Let X be the set of works that must be made by a set Y of machines. Each work $x_i \in X$ is performed by a machine $y_k \in Y$. There is a strict order in scheduling the works, meaning that the work x_i must start at time moment $\tau(x_i)$ and be finished at time moment $\tau'(x_i)$. Moreover, there exists a time interval $\tau''(x_i, x_k)$ between the moment of finishing the work x_i and the moment of starting the work x_k. The aim of the problem is to find an optimal scheduling for a variable amount of work which uses as few machines as possible and to determine the load of the machines for such an optimal scheduling.

This problem can be formulated as a problem of parametric minimum flow in a network $\overline{G} = (N, A, \overline{\ell}, u, s, t)$ which is built in the following way: for each of the works x_i, $i = 1, \ldots, n$, the network contains a pair of nodes x_i' and x_i'' together with the arc (x_i', x_i'') having the lower bound function given by $\ell(x_i', x_i''; \lambda) = \lambda \cdot u_i$ with nonnegative value u_i representing the maximum value of the amount of work x_i, *i.e.* $u(x_i', x_i'') = u_i$, and the real parameter $\lambda \in [0,1]$. A source node s and a sink node t are also added to the network. The source node is connected through an arc having $\ell(s, x_i') = 0$ and $u(s, x_i') = u_i$, $i = 1, \ldots, n$, by each of the nodes x_i'. Similarly, from each of the nodes x_i'', an arc is built toward the sink node, having $\ell(x_i'', t) = 0$ and $u(x_i'', t) = u_i$, $i = 1, \ldots, n$. If $\tau'(x_i'') + \tau''(x_i'', x_j') \leq \tau(x_j')$ then the arc (x_i'', x_k') with $\ell(x_i'', x_k') = 0$ and $u(x_i'', x_k') = u_i$ is also added to the network. The optimal scheduling of the work can be found by solving a minimum flow problem in the parametric network described above.

5. PARTITIONING ALGORITHM FOR FINDING THE PARAMETRIC MINIMUM FLOW

Every iteration of the proposed algorithm achieves an improvement of the flow within a subinterval of the parameter values. This subinterval is constantly updated so that in its inside the residual capacities of all network arcs show no breakpoints. A range of parametric residual networks that take into account the previously stated requirement are successively defined by the algorithm. By doing so, the difficulty to compare or to subtract two piecewise linear functions can be avoided. The parametric residual network defined for a subinterval of the type $J_k = [\lambda_k, \lambda_{k+1}]$ is denoted by $\overline{G}_k'(\bar{f})$, having all the parametric residual capacities explicitly written as linear functions, $\bar{r}'(i, j; \lambda) = \alpha_k(i, j) + (\lambda - \lambda_k) \cdot \beta_k(i, j)$. The value $\alpha_k(i, j)$ is computed as $\alpha_k(i,j) = u(j,i) - f_0(j,i) + f_0(i,j) - \ell_0(i,j) - \lambda_k \cdot \pounds(i,j)$ while the slope of the parametric residual capacity is $\beta_k(i, j) = \pounds(i, j)$. On each of its iterations, the algorithm finds a shortest directed path P from source to sink in parametric residual network $\overline{G}_k'(\bar{f})$. When a directed paths P is found, the algorithm builds its parametric residual capacity $\bar{r}'(P)$ by computing $\alpha_k(P) := min\ \{\alpha_k(i, j) \mid (i, j) \in P\}$ and. $\beta_k(P) := min\ \{\beta_k(i,j) \mid (i,j) \in P\ \ and\ \ \alpha_k(i,j) = \alpha_k(P)\}$

Then the upper limit λ_{k+1} of the subinterval $J_k = [\lambda_k, \lambda_{k+1}]$ is updated to value of the parameter which corresponds to the first crossing point (if one exists within J_k) between the parametric residual capacity of the directed path and the residual capacities of all arcs composing the directed path P, so that the function $\bar{r}'(P)$ to remain linear, without breakpoints within J_k. This will do that for all arcs $(i,j)\in P$ with $(\beta_k(i,j)<\beta_k(P))$ the following value of the upper limit λ_{k+1} to be computed:

$$\lambda_{k+1}=min\{\lambda_{k+1},\ \lambda_k+(\alpha_k(i,j)-\alpha_k(P))/(\beta_k(P)-\beta_k(i,j))\}$$

. Finally, the parametric residual network $\bar{G}_k'(\bar{f})$ is updated for all arcs $(i,j)\in P$ by subtracting the parametric residual capacity $\bar{r}'(P)$ of the directed path from the direct arcs and adding it to the reverse ones: $\alpha_k(i,j):=\alpha_k(i,j)-\alpha_k(P)$; $\beta_k(i,j):=\beta_k(i,j)-\beta_k(P)$; $\alpha_k(j,i):=\alpha_k(j,i)+\alpha_k(P)$; $\beta_k(j,i):=\beta_k(j,i)+\beta_k(P)$.

As soon as the parametric residual network $\bar{G}_k'$ contains no directed path from the source node to the sink node, the algorithm reiterates over the next subinterval, until the whole interval of the parameter values is covered. Due to the fact that $\bar{r}'(P)\le\bar{r}'(i,j;\lambda)$ for all the arcs composing any directed path P within J_k, every directed path P in the parametric residual network $\bar{G}_k'(\bar{f})$ is also a conditional decreasing directed path $\bar{P}$ in $\bar{G}'(\bar{f})$ for the subinterval J_k.

Parametric Min Flow Algorithm

```
(1)  build a feasible flow $f_0$ in network $G^*$;
(2)  $B:=\{0\}$; $k:=0$; $\lambda_k:=0$;
(3)  REPEAT
(4)      build the parametric residual network $\bar{G}_k'(f_0)$;
(5)      $\lambda_{k+1}:=\Lambda$;
(6)      WHILE (exists a directed path $P$ in $\bar{G}_k'(f_0)$) DO
(7)          build a directed path $P$ in $\bar{G}_k'(f_0)$;
(8)          build the parametric residual capacity $\bar{r}'(P)$ of the directed path $P$;
(9)          update the value of $\lambda_{k+1}$;
(10)         update the parametric residual network $\bar{G}_k'(\bar{f})$;
(13)     compute $\bar{f}_{\min}$ for $J_k=[\lambda_k,\lambda_{k+1}]$;
(14)     add $\lambda_{k+1}$ to B;
(15)     $k:=k+1$;
(16) UNTIL ($\lambda_k=\Lambda$);
```

In the first line of the algorithm, if that exists, a feasible flow f_0 in a nonparametric network is built. The nonparametric network, which is denoted by $G^*=(N,A,\ell^*,u,s,t)$, is obtained from the parametric one by replacing the parametric lower bounds $\bar{\ell}(i,j;\lambda)$ with the constants $\ell^*(i,j)$ as shown in the followings: $\ell^*(i,j)=max\ \{\bar{\ell}(i,j;\lambda)\mid\lambda\in[0,\Lambda]\}$, *i.e.* $\ell^*(i,j)=\ell_0(i,j)-\Lambda\cdot£(i,j)$ for $£(i,j)<0$ and $\ell^*(i,j)=\ell_0(i,j)$ for $£(i,j)\ge 0$. For this phase, see the algorithms presented in the papers of Ahuja, Magnanti and Orlin [10] or Ciurea and Ciupală [11]. Further on, with $k=0$ the algorithm builds the parametric residual network $\bar{G}_k'(f_0)$ for $J_0=[0,\Lambda]$, *i.e.* $\bar{r}'(i,j;0)=\alpha_0(i,j)=u(j,i)-f_0(j,i)+f_0(i,j)-\ell_0(i,j)$ for $\lambda_0=0$. During its successive iterations an ordered list $B=\{\lambda_0=0,\lambda_1,\ldots,\lambda_{K+1}=\Lambda\}$ of the parameter values which define the successive parametric residual networks $\bar{G}_k'(\bar{f})$ is used. Initially, for $k=0$, the list is initialised as $B:=\{0\}$. As soon as the parametric residual network $\bar{G}_k'(\bar{f})$ contains no directed paths, the value λ_{k+1} of the lower limit of the next subinterval $J_k=[\lambda_{k+1},\Lambda]$, corresponding to the parametric residual network $\bar{G}_{k+1}'(\bar{f})$, *is added to* B . At those moments the parametric minimum flow $\bar{f}_{\min}(i,j;\lambda)$ is computed for the subinterval $J_k=[\lambda_k,\lambda_{k+1}]$ and the algorithm goes on iterating within the next subinterval $J_k=[\lambda_{k+1},\Lambda]$ until the value $\lambda_{k+1}=\Lambda$ is reached. After all the necessary updating steps within the subinterval $J_k=[\lambda_k,\lambda_{k+1}]$, from the optimal residual capacities, the algorithm computes the parametric minimum flow $\bar{f}_{\min}(i,j;\lambda)$ for all $(i,j)\in A$.

Theorem 2. (*THEOREM OF CORRECTNESS*) *If there exists a feasible flow in the parametric network*

$\overline{G}=(N,A,\bar{\ell},u,s,t)$, PARAMETRIC MIN FLOW algorithm correctly computes a minimum parametric flow within the parameter values interval $\lambda\in[0,\Lambda]$.

Proof. The algorithm runs in successive parametric residual networks $\overline{G}'_k(\bar{f})$ where both $\bar{r}'(i,j;\lambda)$, $\forall(i,j)\in A$ and $\bar{r}'(P)$ are linear functions without intersections within $J_k=[\lambda_k,\lambda_{k+1}]$. It follows that the evolution of the algorithm is similar to the one of the nonparametric algorithm and thus, its correctness in every subinterval J_k results directly from the previously mentioned similarity.

Theorem 3. (*THEOREM OF COMPLEXITY*) *The complexity of PARAMETRIC MIN FLOW algorithm is $O(Kn^2m)$, with $K=|B|-1$.*

Proof. In every subinterval of $[0,\Lambda]$ the algorithm actually performs a non-parametric successive shortest decreasing directed path algorithm with the complexity $O(n^2m)$ (see Ahuja et al. [10]). Consequently, the complexity of the algorithm is $O(Kn^2m)$.

Considering that Orlin reported solving the maximum flow problem in a special type of nonparametric static network with $m<n^{1.06}$ (see [12]) in $O(nm)$ time, results that the problem of the parametric minimum flow solved in best time could have a complexity of $O(Knm)$.

6. EXAMPLE

For the network presented in Fig. 1(a) with λ taking values in [0, 1], *i.e.* Λ=1, the source is node $s=0$ and the sink is node $t=3$. The nonparametric network G^* is presented in Fig. 1(b). The parametric residual network $\overline{G}'_0(f_0)$ is presented in Fig. 2(a) where the parametric residual capacity function $\bar{r}'(i,j;\lambda)=\alpha_0(i,j)+\lambda\cdot\beta_0(i,j)$ for every arc (i,j) is indicated.

After the initialisations step (2) of the algorithm, in the parametric residual network $\overline{G}'_0(f_0)$, the following directed paths: $P_1=(0,1,3)$, $P_2=(0,2,3)$ and $P_3=(0,2,1,3)$ with their corresponding parametric residual capacities $\bar{r}'(P_1)=1+3\lambda$, $\bar{r}'(P_2)=4+3\lambda$ and $\bar{r}'(P_3)=\lambda$ are consecutively built. For each of the directed paths, the value of the upper limit of the subinterval of the parameter is updated to $\lambda_1 = min\{1,\ 2/3\}=2/3$, $\lambda_1 = min\{2/3,\ 1/2\}=1/2$ and finally to $\lambda_1 = min\{1/2,\ 1/3\}=1/3$. Since there is no directed path from the source to the sink node in the updated parametric residual network, which is presented in Fig. 2(b), the parametric minimum flow $\bar{f}_{\min}$ (see Fig. 3(a)) for the subinterval $J_0=[0,1/3]$ is computed and the first step of the algorithm ends after the final value $\lambda_1=1/3$ is added to the list of breakpoints, which becomes $B=\{0,\ 1/3\}$, and counter is incremented to $k:=1$.

In the second step, starting with $J_1=[1/3,1]$, the algorithm builds, in the parametric residual network $\overline{G}'_1(f_0)$, the directed paths $P_1=(0,1,3)$, $P_2=(0,2,3)$ and $P_3=(0,2,1,3)$ with the parametric residual capacities $\bar{r}'(P_1)=2+3\lambda$, $\bar{r}'(P_2)=5+3\lambda$ and $\bar{r}'(P_3)=1/3-2\lambda$ which narrows the subinterval of the parameter values to $J_1=[1/3,1/2]$ and leads to the parametric minimum flow $\bar{f}_{\min}$ for the considered subinterval, presented in Fig. 3(b). After two more steps, the algorithm ends with the parametric minimum flow $\bar{f}_{\min}$ computed for the subintervals $J_2=[1/2,\ 2/3]$ and $J_3=[2/3,1]$ as presented in Fig. 3(c) and (d).

As it was defined in equation (3.a) of definition 2, the piecewise linear minimum flow value function $\bar{v}(\lambda)$ is presented Fig. 3(e), for the parametric network $\overline{G}$ shown in Fig. 1(a) and for the whole range of values of the parameter $\lambda\in[0,\Lambda]$. It can be easily seen that for the value $\lambda=1/2$ the function $\bar{v}(\lambda)$ does not change its slope but the parametric flow only distributes differently over the arcs of the network $\overline{G}$.

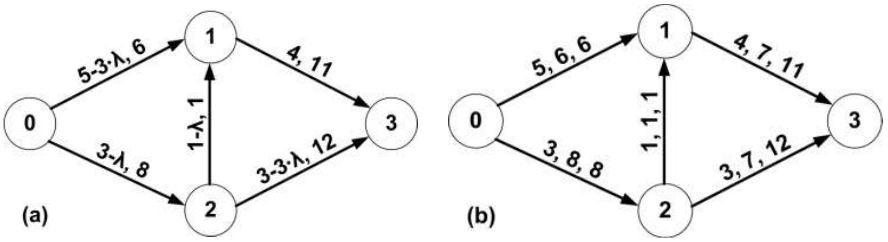

Fig. 1. (a) The parametric network $\overline{G}$ *;* (b) The non-parametric network G^{*}.

(a): *For every arc* (i, j) *, the linear lower bound function* $\overline{\ell}(i, j; \lambda)$ *and the constant upper bound value* $u(i, j)$ *are indicated;*

(b): *The three numbers on each arc* (i, j) *denote* $\ell^{*}(i, j)$*,* $f_0(i, j)$ *and* $u(i, j)$.

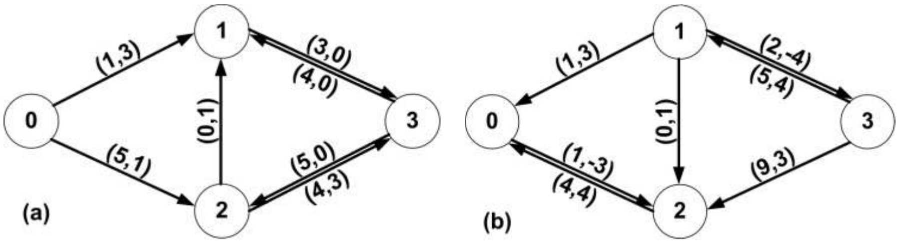

Fig. 2. The parametric residual networks (a) $\overline{G}_0'(f_0)$ and (b) $\overline{G}_0'(\bar{f}_{\min})$.

The sets of two numbers indicated for every arc (i, j) *denote* $\left(\alpha_0(i, j),\ \beta_0(i, j)\right)$**.**

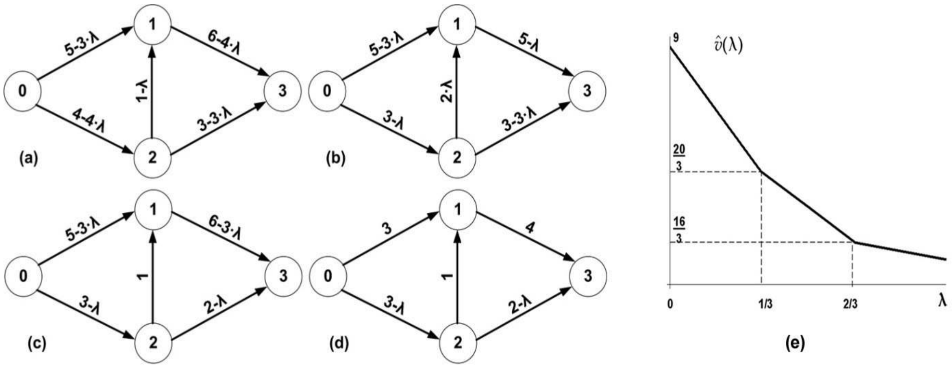

Fig. 3. (a)-(d) The parametric minimum flow $\bar{f}_{\min}$ for subinterval of the parametric values*;* (e) The piecewise linear minimum flow value function $\overline{v}(\lambda)$ for the considered parametric network $\overline{G}$.

(a): $J_0 = [0, 1/3]$*; (b):* $J_1 = [1/3, 1/2]$*; (c):* $J_2 = [1/2, 2/3]$*; (d):* $J_3 = [2/3, 1]$.

7. CONCLUSIONS

Since the proposed approach for the parametric minimum flow problem is based on an algorithm which works in the parametric residual network, the changes occurring by considering parametric upper bounds instead of constant ones are easy to be dealt. Thus the proposed algorithm can be extended to networks with both lower and upper parametric bounds. The main advantage of the proposed approach consists in working with linear instead of piecewise linear functions, *i.e.* the residual capacity of every arc in the residual network is explicitly written as a linear function. Considering that the parametric upper bounds are linear functions instead of constants, the parametric residual capacity of every arc in the residual network also remains written as a linear function, allowing the running of the algorithm in a similar manner. Consequently, the proposed algorithm remains valid (with the appropriate modifications) for the case of both parametric upper bounds and parametric lower bounds.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCES

1. Horst W Hamacher, Les R Foulds. “Algorithms for flows with parametric capacities. ZOR – Methods and Models of Operations Research. 1989;33:21-37.
2. Günther Ruhe. Complexity results for multicriterial and parametric network flows using a pathological graph of Zadeh. Zeitschrift für Operations Research. 1988;32:9-27.
3. Günther Ruhe. Characterization of all optimal solutions and parametric maximal flows in networks. Optimization. 1985;16(1): 51-61.
4. Giorgio Gallo, Michael D Grigoriadis, Robert E Tarjan. A fast parametric maximum flow algorithm and applications. SIAM Journal on Computing. 1989;18:30-55.
5. Bin Zhang. A new balancing method for solving parametric maximum flow problems. Stanford EE Computer Systems Colloquium; 2007.
6. Bin Zhang, Julie Ward, Qi Feng. A simultaneous parametric maximum-flow algorithm with vertex balancing. Palo Alto: HP Laboratories, Technical Report HPL. 2005;121.
7. Charles-Edmond Bichot, Patrick Siarry. Graph partitioning: Optimisation and applications. ISTE – Wiley; 2011.
8. Mircea Parpalea, Eleonor Ciurea. Partitioning algorithm for the parametric maximum flow. Applied Mathematics (Special Issue on Computer Mathematics). 2013;4(10A):3-10.
DOI: 10.4236/am.2013.410A1002
9. Mircea Parpalea. Min-Max algorithm for the parametric flow problem. Bulletin of the Transilvania University of Brasov, Series III: Mathematics, Informatics, Physics. 2010;3(52):191-198.
10. Ravindra K Ahuja, Thomas L Magnanti, James B Orlin. Network flows. Theory, algorithms and applications. Prentice Hall Inc., Englewood Cliffs, New Jersey; 1993.
11. Eleonor Ciurea, Laura Ciupalǎ. Sequential and parallel algorithms for minimum flows. Journal of Applied Mathematics and Computing. 2004;15(1-2):53-75.
12. James B Orlin. Max flows in O(nm) time, or better. Proceedings of the forty-fifth annual ACM symposium on Theory of computing, ACM Press, New York. 2013;765-774.

Peer-review history:
The peer review history for this paper can be accessed here:
http://sciencedomain.org/review-history/12820