Bull. Math. Soc. Sci. Math. Roumanie
Tome 56(104) No. 4, 2013, 387–401

Shortest conditional decreasing path algorithm for the parametric minimum flow problem

by
Eleonor Ciurea and Mircea Parpalea

Abstract

This article presents a decreasing directed paths algorithm for the parametric minimum flow problem. The algorithm always finds a shortest conditional decreasing directed path from the source node to the sink node in a parametric residual network and decreases the flow along the corresponding path in the original parametric network. On each of the iterations, the shortest conditional decreasing path (SCDP) algorithm computes a subinterval of the parameter value within a decreasing of flow is possible and the maximum amount by which the flow can be decreased. The complexity of the SCDP algorithm is $O(n^2m^2K + nmK^2)$ where $K - 1$ is the number of breakpoints of the piecewise linear minimum flow value function, n and m being respectively the number of nodes and the number of arcs in the network.

Key Words: Network flow, parametric minimum flow problem, decreasing paths algorithm.
2010 Mathematics Subject Classification: Primary 90B10; Secondary 90C35, 90C47, 05C35, 68R10.

1 Introduction

For the parametric maximum flow problem with zero lower bounds and linear capacity functions of a parameter λ, Hamacher and Foulds [8] investigated an augmenting paths approach for determining in each iteration an improvement of the flow defined on the whole interval of the parameter. For the same problem, Ruhe [9] proposed a "piece-by-piece" approach. Gallo, Grigoriadis, and Tarjan [7] and Ahuja et al. [2], [3] have pointed out that the parametric problem has many applications, in multiprocessor scheduling with release times and deadlines, integer programming problems, computing subgraph density and network vulnerability and partitioning a data base between fast and slow memory. Ciurea et al. [4], [5] investigated the non-parametric minimum flow problem. The approach presented in this article refers to the minimum flow problem in a network with linear lower bound functions of a single parameter λ. Further on, this paper is organized as follows: Section 2 contains the basic network flow terminology and some results used in the rest of the paper. More specialized terminology is developed in later sections. Section 3 deals with the parametric minimum flow problem. Section 4 presents the shortest conditional decreasing path algorithm for the parametric minimum flow problem. Section 5 gives an example of how the algorithm works on a network with linear lower bound functions of a parameter. In the presentation to follow, some familiarity with flow algorithms is assumed and many details are omitted, since they are straightforward modifications of known results. The notions and results presented in Section 2 and Section 3 are taken from [1], [2], [4], [5] and [6].

2 Terminology and preliminaries

Definitions and Notation

Given a capacitated network $G = (N, A, \ell, u, s, t)$ with $N = \ldots, i, \ldots$ being the set of nodes i and $A = \ldots, a, \ldots$ 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 and the *lower bound* function are two nonnegative functions, $u(a)$ and $\ell(a)$ 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 next 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 $t \in T$.

The minimum flow problem

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] and [5].
Let f be a feasible solution for the minimum flow problem. Supposing that an arc $(i, j) \in A$ carries $f(i, j)$ units of flow, the *residual capacity* $\hat{r}(i, j)$ of any arc $(i, j) \in A$, with respect to a given flow f, for the minimum flow problem is given by:

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

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:

$$\hat{c}[S, T] = \ell(S, T) - u(T, S). \tag{4}$$

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 1. *(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 2. *(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 The parametric minimum flow problem

A natural generalization of the minimum flow problem is obtained by making the lower bounds $\ell(i, j)$ for some of the arcs $(i, j) \in A$ linear functions of a single, nonnegative, real parameter λ:

$$\ell(i, j; \lambda) = \ell_0(i, j) - \lambda \cdot L(i, j), \tag{5}$$

where $L(i, j)$ is a real valued function associated with each arc $(i, j) \in A$, referred to as the *parametric part of the lower bound* of the arc (i, j). The nonnegative value $\ell_0(i, j)$ is the lower bound of the arc (i, j) for $\lambda = 0$: $\ell(i, j; 0) = \ell_0(i, j)$ with $0 \le \ell_0(i, j) \le u(i, j)$.
For the problem to be correctly formulated, the lower bound function of every arc $(i, j) \in A$ must respect the condition $0 \le \ell_0(i, j; \lambda) \le u(i, j)$ for the entire interval of the parameter values, i.e. $\forall (i, j) \in A$ and $\forall \lambda \in [0, \Lambda]$. It follows that the parametric part of the lower bounds $L(i, j)$ must satisfy the constraints: $\frac{1}{\Lambda}(\ell_0(i, j) - u(i, j)) \le L(i, j) \le \frac{1}{\Lambda}\ell_0(i, j)$, $\forall (i, j) \in A$.
The parametric minimum flow (PMinF) problem is to compute all minimum flows for every possible value of $\lambda \in [0, \Lambda]$:

$$minimize \quad v(\lambda) \quad for\ all \quad \lambda \in [0, \Lambda] \qquad with: \tag{6}$$

$$\sum_{j|(i,j)\in A} f(i, j; \lambda) - \sum_{j|(j,i)\in A} f(j, i; \lambda) = \begin{cases} v(\lambda), & i = s \\ 0, & i \ne s, t \\ -v(\lambda), & i = t \end{cases} \tag{7}$$

$$\ell(i,j;\lambda) \le f(i,j;\lambda) \le u(i,j), \qquad \forall (i,j) \in A. \tag{8}$$

On the set of piecewise linear functions $F(\lambda) : [0,\Lambda] \to \Re^{+}$ an ordering cannot be defined for the whole interval $[0,\Lambda]$ since two piecewise linear functions are not necessarily comparable. For any two piecewise linear functions of a parameter λ, $F_1(\lambda)$ and $F_2(\lambda)$, a partition B of the interval of the parameter values of the form $0 = \lambda_0 < \lambda_1 < \ldots < \lambda_{K+1} = \Lambda$ must be defined such that on each of the generated sub-intervals $[\lambda_k, \lambda_{k+1}]$, $k = 0,\ldots,K$ an ordering to be defined as:

$$F_1(\lambda) \le F_2(\lambda) \; for \; [\lambda_k, \lambda_{k+1}] \; \Leftrightarrow F_1(\lambda) \le F_2(\lambda) \quad \forall \lambda \in [\lambda_k, \lambda_{k+1}], \tag{9}$$

i.e. the two functions have no crossing points within any of the sub-intervals $(\lambda_k, \lambda_{k+1})$, the only crossing points taking place for λ_k, $k = 0,\ldots,K$. To be mentioned that the breakpoints and the crossing points of the two piecewise linear functions do not necessarily take place for the same λ_k values.

Based on the above considerations, for the parametric minimum flow problem, the sub-intervals of the parameter values with the property that within each of the sub-intervals $[\lambda_k, \lambda_{k+1}]$ the maximum $s-t$ cut for the non-parametric network with constant lower bound functions $\ell(i,j;\lambda_k)$ remains a maximum $s-t$ cut for all the λ values in $[\lambda_k, \lambda_{k+1}]$ are denoted by J_k.

Definition 1. *A parametric cut partitioning $[S_k; J_k]$ is a finite set of cuts $[S_k, T_k]$, $k = 0,\ldots,K$ together with a partitioning of the interval of the parameter $[0,\Lambda]$ in sub-intervals J_k such that $J_0 \cup \ldots \cup J_K = [0,\Lambda]$. The capacity of a parametric $s-t$ cut partitioning for the minimum flow problem is a piecewise linear function $\hat{c}[S_k; J_k]$ defined for all λ values of every sub-interval $\lambda \in J_k$, $k = 0,\ldots,K$:*

$$\hat{c}[S_k; J_k] = \sum_{(i,j)\in(S_k,T_k)} \ell(i,j;\lambda) - \sum_{(i,j)\in(T_k,S_k)} u(i,j) \tag{10}$$

A parametric $s-t$ cut with the sub-intervals J_k assuring that every $s-t$ cut is a maximum cut $[\hat{S}_k, \hat{T}_k]$ within the sub-interval J_k is referred to as a *parametric maximum $s-t$ cut* for the entire interval of the parameter values $[0,\Lambda]$ and is denoted by $[\hat{S}_k; J_k]$. Thus a parametric maximum cut $[\hat{S}_k; J_k]$ is a set of maximum cuts $[\hat{S}_k, \hat{T}_k]$ and $\hat{c}[\hat{S}_k; J_k] = \hat{c}[\hat{S}_k; \hat{T}_k]$ for all λ of every sub-interval J_k, $k = 0,\ldots,K$. As defined, the capacity of a parametric maximum $s-t$ cut partitioning for the minimum flow problem is a linear function within every sub-interval J_k.

Theorem 3. *(Parametric Min-Flow Max-Cut Theorem): If there is a feasible flow in the parametric network, the value function of the parametric minimum flow from a source s to a sink t in a capacitated network with parametric lower bounds equals the capacity of the parametric maximum $s-t$ cut. [6]*

Let $f(\lambda) = (\ldots f(i,j;\lambda),\ldots)_{(i,j)\in A}$ be a vector of feasible flow functions defined on the interval $[0,\Lambda]$. Supposing that an arc $(i,j) \in A$ carries a flow $f(i,j;\lambda)$, the existing flow can be reduced either by *pulling* the flow $f(i,j;\lambda) - \ell(i,j;\lambda)$ from node j to node i over the arc (i,j) or by *pushing* the flow $u(j,i) - f(j,i;\lambda)$ from j to i along the arc (j,i). These flows are computed as differences between piecewise linear functions of λ. The *parametric residual capacity* $\hat{r}(i,j;\lambda)$ of any of the arcs $(i,j) \in A$, with respect to a given flow $f(\lambda)$, is given by:

$$\hat{r}(i,j;\lambda) = u(j,i) - f(j,i;\lambda) + f(i,j;\lambda) - \ell(i,j;\lambda), \quad \lambda \in J_k \tag{11}$$

with $k = 0, \ldots, K(i,j)$ where $K(i,j)$ is the number of sub-intervals where the piecewise linear parametric residual capacity function of the arc (i,j) maintains a constant slope. Based on these ideas, for a network $G(\lambda) = (N, A, \ell(\lambda), u, s, t)$ and a feasible solution $f(\lambda)$, the network denoted by $\hat{G}(f(\lambda)) = (N, \hat{A}(\lambda))$, with $\hat{A}(\lambda)$ being the set of arcs consisting only of arcs with $\hat{r}(i,j;\lambda) > 0$ for at least a sub-interval of $[0, \Lambda]$, is referred to as the *parametric residual network* with respect to the given flow $f(\lambda)$ for the parametric minimum flow problem. The sub-intervals $\hat{I}(i,j) \subseteq [0, \Lambda]$ where a decreasing of flow along an arc (i,j) in $\hat{G}(f(\lambda))$ is possible based on $f(\lambda)$, are defined as:

$$\hat{I}(i,j) = \{\lambda | \hat{r}(i,j;\lambda) > 0\} \quad for \quad (i,j) \in \hat{A}(\lambda). \tag{12}$$

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

Definition 2. *A conditional decreasing directed path $\hat{P}(\lambda)$ in $\hat{G}(f(\lambda))$ is a directed path $\hat{P}$ from the source node s to the sink node t such that: $\hat{I}(\hat{P}) = \bigcap_{(i,j)\in\hat{P}} \hat{I}(i,j) \neq \emptyset$.*

Definition 3. *A partly conditional decreasing directed path $\hat{P}(i,\lambda)$ in $\hat{G}(f(\lambda))$ is a conditional decreasing directed path from the source node s to a node i with $i \neq t$.*

Definition 4. *The parametric residual capacity of a conditional decreasing directed path $\hat{P}(\lambda)$ in $\hat{G}(f(\lambda))$ is the inner envelope of the parametric residual capacities $\hat{r}(i,j;\lambda)$ of all arcs composing the conditional decreasing directed path for all $\lambda \in \hat{I}(\hat{P})$:*

$$\hat{r}(\hat{P}(\lambda)) = \min_{\lambda\in\hat{I}(\hat{P})} \{\hat{r}(i,j;\lambda) | (i,j) \in \hat{P}(\lambda)\}, \quad \lambda \in J_k \tag{13}$$

with $k = 0, \ldots, K(\hat{P}(\lambda))$ where $K(\hat{P}(\lambda))$ is the number of sub-intervals where the piecewise linear parametric residual capacity function of the conditional decreasing directed path $\hat{P}(\lambda)$ maintains a constant slope.

It must be mentioned that, denoting by $K(i,j)$ the number of sub-intervals where the piecewise linear parametric residual capacity function of the arc (i,j) maintains a constant slope, $K(\hat{P}(\lambda))$ generally respects the following relation: $K(\hat{P}(\lambda)) \geq \sum_{(i,j)\in\hat{P}} K(i,j)$.

Theorem 4. *(Conditional Decreasing Path Theorem): A flow $\hat{f}(\lambda)$ is a parametric minimum flow if and only if the parametric residual network $\hat{G}(\hat{f}(\lambda))$ contains no conditional decreasing directed path from the source node to the sink node. [6]*

If the parametric residual network $\hat{G}(\hat{f}(\lambda))$ contains no conditional decreasing directed path from the source node to the sink node, the parametric minimum flow in network $G(\lambda)$ can be determined from the parametric residual capacities using the expression: $\hat{f}(i,j;\lambda) = \ell(i,j;\lambda) + \max\{\hat{r}(i,j;\lambda) - u(j,i) + \ell(j,i;\lambda), \quad 0\}$.

4 Shortest decreasing paths algorithm for the parametric minimum flow problem

The shortest decreasing path algorithm for the parametric minimum flow problem presented in this paper determines in each of the iterations an improvement of the flow over the sub-interval of the parameter values given by a shortest conditional decreasing directed path in the parametric residual network.

Finding the minimum length of a conditional decreasing directed path

Let $N = \{0, 1, 2, \ldots, n-1\}$ be the set of nodes in the parametric residual network with $s = 0$ and $t = n - 1$. For increasing values of $h = 1, 2, \ldots, n - 1$, the sub-intervals $\hat{I}_{jh}$ on which the flow can be decreased over a partly conditional decreasing directed path from the source node s to node j, $\hat{P}(j, \lambda)$ in $\hat{G}(f(\lambda))$ with exactly h arcs are computed for each node $j \in N - \{0\}$ as:

$$\hat{I}_{jh} := \{\lambda \mid \exists\ \hat{P}(j, \lambda) \in \hat{G}(f(\lambda))\ \textit{with exactly } h \textit{ arcs}\} \tag{14}$$

To find a recursive way of computing the sub-intervals $\hat{I}_{jh}$, the following relation is considered: $\hat{I}_{00} = [0, \Lambda]$, $\hat{I}_{j0} = \emptyset$ for all $j = 1, \ldots, n - 1$,

$$\hat{I}_{jh} := \bigcup_{i \mid (i,j) \in \hat{A}(\lambda)} (\hat{I}_{ih-1} \bigcap \hat{I}(i, j)), \quad j = 1, \ldots, n - 1; \quad h = 1, \ldots, n - 1. \tag{15}$$

The set of the sub-intervals $\hat{I}_{jh}$ in relation (15) can be regarded as being the elements of a matrix $\hat{I} := [\hat{I}_{jh}]_{(n-1)\times(n-1)}$ of sub-intervals of $[0, \Lambda]$. The algorithm starts with $\hat{I} := [\hat{I}_{jh} = \emptyset]_{(n-1)\times(n-1)}$ and as soon as $\hat{I}_{n-1h} \neq \emptyset$ is reached for a certain value of h, the recursion stops indicating that a conditional decreasing directed path of length h can be found. On the other hand, if the condition $\hat{I}_{n-1n-1} = \emptyset$ is reached, since any elementary path from the source node s to the sink node t has at most $n - 1$ arcs, no conditional decreasing directed path exists and the current flow is a parametric minimum flow. The procedure LENGTH presented below determines whether there exists a nonempty sub-interval of the parameter values $\hat{I}_{n-1h}$ over which an improvement of the flow can be obtained. Based on this sub-interval both the value h of the minimum length of a conditional decreasing directed path and the path itself can be derived.

```
procedure LENGTH(Î, h, C)
begin
  while (C = 0) do
    begin;
      for j := 1 to n − 1 do Î_jh := ⋃_{i|(i,j)∈Â(λ)} (Î_ih−1 ⋂ Î(i, j)) ;
      if (Î_n−1h ≠ ∅) then C:=1
      else if (h < n − 1) then h := h + 1
            else C := 2;
    end;
end;
```

Theorem 5. *If the procedure* LENGTH *ends with $\hat{I}_{n-1h} \neq \emptyset$ then h is the length of the shortest conditional decreasing directed path from the source node to the sink node in the parametric residual network $\hat{G}(f(\lambda))$.*

Proof. The algorithm stops as soon as $\hat{I}_{n-1h} \neq \emptyset$ which means that a conditional decreasing directed path consisting of exactly h arcs exists in the parametric residual network $\hat{G}(f(\lambda))$. It is obvious that h is minimal with this property since otherwise the algorithm would have stopped earlier. For an arbitrary chosen value of the parameter $\lambda^* \in \hat{I}_{n-1h}$, in the non-parametric residual network $\hat{G}(f(\lambda^*))$ a breadth first search determines the elements $h(i)$ such that $h(i) = \hat{d}(i)$ for all nodes $i \in N$. For $h = 0$, $\lambda^* \in \hat{I}_{00}$ and $h(0) = \hat{d}(0) = 0$. Assuming the statement to be true for $h = k$ and starting from a node i with $h(i) = \hat{d}(i) = k$, for all nodes j with $\lambda^* \notin \hat{I}_{jk'}$ for all $k' \leq k$ and for which there exists an arc $(i,j) \in \hat{G}(f(\lambda^*))$ holds that $h(j) = h(i) + 1 = \hat{d}(i) + 1 = k + 1 = \hat{d}(j)$. □

It has to be reminded that according to definition 2 a shortest conditional decreasing directed path $\hat{P}(\lambda)$ in the parametric residual network $\hat{G}(f(\lambda))$ is not a shortest directed path $\hat{P}$ in $\hat{G}(f(\lambda))$ since for some arcs $(i,j) \in \hat{P}$ might hold that $\bigcap_{(i,j)\in\hat{P}} \hat{I}(i,j) = \emptyset$.

Finding a conditional decreasing directed path

After procedure *length* ends with a minimum length h of a possible conditional decreasing directed path $\hat{P}(\lambda)$, the problem of finding the conditional decreasing directed path itself is investigated. Let $\hat{I}_{n-1h} \neq \emptyset$ for an $h \leq n-1$, where h is minimal with this property. For every arbitrary chosen value $\lambda^* \in \hat{I}_{n-1h}$ there must exist a node i with $(i, n-1) \in \hat{A}(\lambda)$ so that both $\lambda^* \in \hat{I}_{ih-1}$ and $\lambda^* \in \hat{I}(i, n-1)$ since $\hat{I}_{n-1h} := \bigcup_{i|(i,n-1)\in\hat{A}(\lambda)} (\hat{I}_{ih-1} \bigcap \hat{I}(i, n-1))$. Therefore, for an arc $(i, n-1) \in \hat{P}(\lambda)$ we have:

$$\lambda^* \in \hat{I}_{ih-1} \bigcap \hat{I}(i, n-1)). \tag{16}$$

On the other hand $\lambda^* \notin \hat{I}_{ih-2}$ since otherwise the recursion would have stopped after $h-1$ iterations (i.e. h is minimal with the property $\hat{I}_{n-1h} \neq \emptyset$). Continuing in the same logic it results that $\lambda^* \notin \hat{I}_{ih-3}$ since otherwise the recursion would have stopped after $h-2$ iterations etc. Based on this reasoning, a vector $\hat{p} = (\hat{p}(0), \ldots, \hat{p}(n-1))$ can be defined in order to memorize the conditional decreasing directed path based on relation (16). In the beginning of the algorithm, the following initialisation is performed: $\hat{p} := (n, \ldots, n)$. Let $\hat{p}(h) := n-1$ be set and for an arbitrary chosen value $\lambda^* \in \hat{I}_{n-1h}$, if $\lambda^* \in \hat{I}_{jh-1} \bigcap \hat{I}(j, n-1))$ then the arc $(j, n-1)$ is added to $\hat{P}(\lambda)$, i.e. $\hat{p}(h-1) := j$ is set. Then a new node $i \in N - \{j, n-1\}$ with $\lambda^* \in \hat{I}_{ih-2} \bigcap \hat{I}(i,j))$ is searched, $\hat{p}(h-2) := i$ is set and so on. Based on the values of the vector $\hat{p}$, the following conditional decreasing directed path will be obtained: $\hat{P}(\lambda) := (\hat{p}(0) = 0, \hat{p}(1), \ldots, \hat{p}(h-1), \hat{p}(h) = n-1)$. The procedure DPATH for determining a conditional decreasing directed path $\hat{P}(\lambda)$ is the following:

procedure DPATH$(\hat{I}, h, \hat{p})$
begin
$\lambda^* := (\max\{\lambda \mid \lambda \in \hat{I}_{n-1h}\} + \min\{\lambda \mid \lambda \in \hat{I}_{n-1h}\})/2$;
  $N^* := N - \{n-1\}$;
  $\hat{p}(h) := n-1$;
  for $k := h-1$ **down to** 1 **do**
    begin;
      select a node $i \in \{N^* \mid \lambda^* \in \hat{I}_{ik} \cap \hat{I}(i, \hat{p}(k+1))\}$;
      $\hat{p}(k) := i$;
      $N^* := N^* - \{i\}$;
    end;
  $\hat{p}(0) := 0$;
end;

Theorem 6. *The procedure* DPATH *computes correctly a conditional decreasing directed path of h arcs from the source node to the sink node in the parametric residual network $\hat{G}(f(\lambda))$.*

Proof. If $\hat{I}_{n-1h} \neq \emptyset$ then for every $\lambda^* \in \hat{I}_{n-1h}$ it results that $\lambda^* \in \hat{I}_{ih-1}$ and $\lambda^* \in \hat{I}(i, n-1)$ for at least a node i since $\hat{I}_{n-1h} := \bigcup_{i|(i,n-1)\in\hat{A}(\lambda)} (\hat{I}_{ih-1} \bigcap \hat{I}(i, n-1))$. For an iteration k of the procedure DPATH, the existence of an ending point i of a partly conditional directed path in $\hat{G}(f(\lambda))$ such that $\lambda^* \in \hat{I}_{ik} \cap \hat{I}(i, \hat{p}(k+1))$ is assured by $\lambda^* \in \hat{I}_{ik}$ and $\lambda^* \in \hat{I}(i, \hat{p}(k+1))$ for at least one node i because otherwise, from the recursive relation $\hat{I}_{jk+1} := \bigcup_{i|(i,j)\in\hat{A}(\lambda)} (\hat{I}_{ik} \bigcap \hat{I}(i,j))$, would result that $\lambda^* \notin \hat{I}_{jk'}$ for any $k' > k$ and any node $j \in N - \{0\}$ which contradicts the initial assumption: $\lambda^* \in \hat{I}_{n-1h}$ for $h > k$. Hence, the directed path is a conditional decreasing directed path in $\hat{G}(f(\lambda))$ for at least the sub-interval $\hat{I}(\hat{P}) = \bigcap_{(i,j)\in\hat{P}} \hat{I}(i,j) = \{\lambda^*\}$. □

Decreasing the flow over a conditional decreasing directed path

After a conditional decreasing directed path $\hat{P}(\lambda)$ in the parametric residual network $\hat{G}(f(\lambda))$ is memorised in vector $\hat{p}$, the flow is decreased along the corresponding decreasing path in $G(\lambda)$. For an arc $a = (i,j)$, the piecewise linear parametric residual capacity function $\hat{r}(a;\lambda)$ can be represented considering the ordered list of its $K(a)$ breakpoints $B_a := \{\lambda_k\}$, $0 = \lambda_0 < \lambda_1 < \ldots < \lambda_{K(a)+1} = \Lambda$ and the corresponding ordered list of residual capacity values computed for every breakpoint $\lambda_k \in B_a$: $R_a := \{\hat{r}(a;\lambda_k) \mid \lambda_k \in B_a\}$. Similarly, the piecewise linear parametric residual capacity function $\hat{r}(\hat{P}(\lambda))$ of a conditional decreasing directed path $\hat{P}(\lambda)$ in $\hat{G}(f(\lambda))$ can be represented as the ordered list $R_{\hat{P}(\lambda)} := \{\hat{r}(\hat{P}(\lambda_k)) \mid \lambda_k \in B_{\hat{P}(\lambda)}\}$ with $B_{\hat{P}(\lambda)}$ being the list $0 = \lambda_0 < \lambda_1 < \ldots < \lambda_{K(\hat{P}(\lambda))+1} = \Lambda$ containing, in increasing order, its $K(\hat{P}(\lambda))$ breakpoints. As mentioned before, $K(\hat{P}(\lambda)) \geq \sum_{a\in\hat{P}(\lambda)} K(a)$, i.e. $\bigcup_{a\in\hat{P}(\lambda)} B_a \subseteq B_{\hat{P}(\lambda)}$. In order to compute the piecewise linear parametric residual capacity function $\hat{r}(\hat{P}(\lambda))$ of a conditional decreasing directed path $\hat{P}(\lambda)$, the procedure PRC (*parametric residual capacity*) repeatedly compares two piecewise linear parametric residual capacity functions and sets the parametric residual capacity of the conditional decreasing directed path to their inner envelope. It starts by setting the parametric residual capacity of the conditional decreasing directed path to the parametric residual capacity of the first arc $a \in \hat{P}(\lambda)$ and successively computes the inner envelope of the current parametric residual capacity of the conditional decreasing directed path and the parametric residual capacity of each of the following arcs. The procedure stops after all the arcs in $\hat{P}(\lambda)$ have been investigated. The algorithm starts with $K(a) = 0$ (i.e. no breakpoint), $B_a = \{0, \Lambda\}$ and $R_a = \{\hat{r}(a;0), \hat{r}(a;\Lambda)\}$ for every arc $a \in \hat{A}(\lambda)$. Every time the piecewise linear parametric residual capacity function of a new arc $a \in \hat{P}(\lambda)$ is analysed, the procedure PRC updates the list of breakpoints $B_{\hat{P}(\lambda)}$ with the breakpoints in B_a and then compares, for each of the sub-intervals generated by $B_{\hat{P}(\lambda)}$, the two linear residual capacity functions finding the lowest of them. Any new breakpoint which may occur within a sub-interval $J_k = [\lambda_k, \lambda_{k+1}]$ is added to $B_{\hat{P}(\lambda)}$.

procedure PRC($\hat{p}$)
begin
$a := (0, \hat{p}(1))$; $R_{\hat{P}(\lambda)} := R_a$; $B_{\hat{P}(\lambda)} := B_a$;
for $q := 1$ **to** $h - 1$ **do**
begin
$a := (\hat{p}(q), \hat{p}(q+1))$; $B_a^* := B_a$;
while $(B_a^* \neq \emptyset)$ **do**
begin
$\lambda^* := \min\{\lambda \mid \lambda \in B_a^*\}$
if $(\lambda^* \notin B_{\hat{P}(\lambda)})$ **then** ADD NEW(λ^*, $B_{\hat{P}(\lambda)}$, $R_{\hat{P}(\lambda)}$);
remove λ^* from B_a^*;
end;
$\lambda_\ell := \min\{\lambda \mid \lambda \in B_{\hat{P}(\lambda)}\}$;
if $(\lambda_\ell \notin B_a)$ **then** ADD NEW(λ_ℓ, B_a, R_a);
repeat
begin
$\lambda_r := \min\{\lambda \mid \lambda \in B_{\hat{P}(\lambda)},\ \lambda > \lambda_\ell\}$;
if $(\lambda_r \notin B_a)$ **then** ADD NEW(λ_r, B_a, R_a);
if $([\hat{r}(a; \lambda_\ell) - \hat{r}(\hat{P}(\lambda_\ell))] \cdot [\hat{r}(a; \lambda_r) - \hat{r}(\hat{P}(\lambda_r))] < 0)$ **then**
begin
$\lambda_{mid} := \frac{\lambda_\ell \cdot [\hat{r}(\hat{P}(\lambda_r)) - \hat{r}(a;\lambda_r)] + \lambda_r \cdot [\hat{r}(a;\lambda_\ell) - \hat{r}(\hat{P}(\lambda_\ell))]}{\hat{r}(\hat{P}(\lambda_r)) - \hat{r}(a;\lambda_r) + \hat{r}(a;\lambda_\ell) - \hat{r}(\hat{P}(\lambda_\ell))}$;
ADD NEW(λ_{mid}, $B_{\hat{P}(\lambda)}$, $R_{\hat{P}(\lambda)}$);
ADD NEW(λ_{mid}, B_a, R_a);
end;
$\hat{r}(\hat{P}(\lambda_\ell)) := \min\{\hat{r}(a; \lambda_\ell),\ \hat{r}(\hat{P}(\lambda_\ell))\}$;
$\lambda_\ell := \lambda_r$
end;
until $(\lambda_\ell = \Lambda)$;
$\hat{r}(\hat{P}(\Lambda)) := \min\{\hat{r}(a; \Lambda),\ \hat{r}(\hat{P}(\Lambda))\}$;
end;
end;

The procedure ADD NEW adds a new breakpoint in the piecewise linear parametric residual capacity function by adding a new λ^* value to the list B and a new $\hat{r}(\lambda^*)$ value to the list R.

procedure ADD NEW(λ^*, B, R)
begin
add λ^* to B;
$\lambda' := \max\{\lambda \mid \lambda \in B, \lambda < \lambda^*\}$; $\lambda'' := \max\{\lambda \mid \lambda \in B, \lambda > \lambda^*\}$;
$\hat{r}(\lambda^*) := \hat{r}(\lambda') + (\lambda^* - \lambda') \cdot [\hat{r}(\lambda'') - \hat{r}(\lambda')]/(\lambda'' - \lambda')$;
add $\hat{r}(\lambda^*)$ to R;
end;

After the piecewise linear parametric residual capacity function $\hat{r}(\hat{P}(\lambda))$ of the conditional decreasing directed path is computed, the procedure UPDATE decreases the flow along the corresponding path in $G(\lambda)$ which reflects in updating the parametric residual network $\hat{G}(f(\lambda))$. For each of the arcs composing the conditional decreasing directed path $a \in \hat{P}(\lambda)$, the procedure checks, in increasing order of the λ_k values, if the breakpoints in $B_{\hat{P}(\lambda)}$ belong to B_a and if not, procedure ADD NEW adds the breakpoints to B_a. Then the values of the parametric residual capacity of the conditional decreasing directed path $\hat{P}(\lambda)$ computed for each of its breakpoints are subtracted from the parametric residual capacity values of all arcs $a = (i, j)$ and added to the parametric residual capacity values of arcs $b = (j, i)$. Simultaneously, every time two consecutive breakpoints with zero parametric residual capacity values are found, the sub-interval bounded by the two λ values is subtracted from $\hat{I}(a)$.

```
procedure UPDATE(p̂)
begin
 λ_ℓ := min{λ | λ ∈ B_{P̂(λ)}, λ > 0};
  repeat
    begin
      for q := 0 to h − 1 do
        begin
          a := (p̂(q), p̂(q + 1)); b := (p̂(q + 1), p̂(q));
          if (λ_ℓ ∉ B_a) then ADD NEW(λ_ℓ, B_a, R_a);
          if (λ_ℓ ∉ B_b) then ADD NEW(λ_ℓ, B_b, R_b);
        end;
      λ_r := min{λ | λ ∈ B_{P̂(λ)}, λ > λ_ℓ}; λ_ℓ := λ_r;
    end;
  until (λ_ℓ = Λ);
 λ_ℓ := min{λ | λ ∈ B_{P̂(λ)}};
  for q := 0 to h − 1 do
    begin
      a := (p̂(q), p̂(q + 1)); b := (p̂(q + 1), p̂(q));
      r̂(a; λ_ℓ) := r̂(a; λ_ℓ) − r̂(P̂(λ_ℓ)); r̂(b; λ_ℓ) := r̂(b; λ_ℓ) + r̂(P̂(λ_ℓ));
    end;
  repeat
    begin
      λ_r := min{λ | λ ∈ B_{P̂(λ)}, λ > λ_ℓ};
      for q := 0 to h − 1 do
        begin
          a := (p̂(q), p̂(q + 1)); b := (p̂(q + 1), p̂(q));
          r̂(a; λ_r) := r̂(a; λ_r) − r̂(P̂(λ_r)); r̂(b; λ_r) := r̂(b; λ_r) + r̂(P̂(λ_r));
          if (r̂(a; λ_ℓ) + r̂(a; λ_r) = 0) then Î(a) := Î(a) − [λ_ℓ, λ_r];
          if (r̂(b; λ_ℓ) + r̂(b; λ_r) ≠ 0) then
            if (r̂(b; λ_ℓ) = 0) then Î(b) := Î(b) + (λ_ℓ, λ_r]
            else if (r̂(b; λ_r) = 0) then Î(b) := Î(b) + [λ_ℓ, λ_r)
                else Î(b) := Î(b) + [λ_ℓ, λ_r];
        end;
      λ_ℓ := λ_r;
    end;
  until (λ_ℓ = Λ);
end;
```

Shortest conditional decreasing paths (SCDP) algorithm

The first phase of finding a parametric minimum flow consists in establishing a feasible flow if one exists in a nonparametric network $G' = (N, A, \ell', u, s, t)$ obtained from the initial network $G(\lambda) = (N, A, \ell(\lambda), u, s, t)$ by only replacing the parametric lower bound functions of every arc $(i, j) \in A$ with the non-parametric lower bounds: $\ell'(i, j) = \max\{\ell(i, j; \lambda) \mid \lambda \in [0.\Lambda]\}$, i.e. $\ell'(i, j) = \ell_0(i, j)$ for $L(i, j) \geq 0$ and $\ell'(i, j) = \ell_0(i, j) - \Lambda \cdot L(i, j)$ for $L(i, j) < 0$. After a feasible flow f_0 is established, the parametric residual network for this feasible flow $\hat{G}(f_0(\lambda))$ is computed. The piecewise linear parametric residual capacity function $\hat{r}(i, j; \lambda)$ of every arc $a = (i, j)$ is initialised as $\hat{r}(i, j; \lambda) = u(j, i) - f(j, i; \lambda) + f(i, j; \lambda) - \ell(i, j; \lambda)$, i.e.

for $B_a = \{0, \Lambda\}$ the corresponding lists of residual capacity values $R_a = \{\hat{r}(a; 0), \hat{r}(a; \Lambda)\}$ with $\hat{r}(i, j; 0) = u(j, i) - f_0(j, i) + f_0(i, j) - \ell(i, j; 0)$ and $\hat{r}(i, j; \Lambda) = u(j, i) - f_0(j, i) + f_0(i, j) - \ell(i, j; \Lambda)$. The sub-interval of the parameter values for every arc is initialised as $\hat{I}(a) := [0, \Lambda]$, $\forall\, a \in \hat{A}(\lambda)$. In the second phase the algorithm repeatedly computes the minimum length h of a conditional decreasing directed path, constructs the shortest conditional decreasing directed path, starting from the sink node t and adding arcs till the source node is reached, memorised in vector $\hat{p}$ and finally once the conditional decreasing directed path found, the flow is decreased and the parametric residual network is updated. As soon as the sink node t cannot be reached, even with a directed path of $n-1$ arcs, i.e. $\hat{I}_{n-1n-1} = \emptyset$, the algorithm stops and the obtained flow is a parametric minimum flow. The shortest conditional decreasing paths (SCDP)algorithm is presented above.

```
SCDP ALGORITHM
Begin
  let f_0 be a feasible flow in network G';
  compute the parametric residual network Ĝ(f_0(λ));
  Î_00 := [0, Λ]; h := 1;
  for j := 1 to n − 1 do Î_j0 := ∅;
  repeat
     C := 0;
     LENGTH(Î, h, C);
     if (C = 1) then
       begin
           DPATH(Î, h, p̂);
           PRC(p̂);
           UPDATE(p̂);
       end;
  until (C = 2);
End.
```

Theorem 7. *If there is a feasible flow in the network $G(\lambda) = (N, A, \ell(\lambda), u, s, t)$, the shortest conditional decreasing paths algorithm computes correctly a minimum flow.*

Proof. The algorithm terminates when the sink node t cannot be reached in the parametric residual network even with a conditional decreasing directed path of $n-1$ arcs, i.e. $\hat{I}_{n-1n-1} = \emptyset$. From the Conditional Decreasing Path Theorem (4) it results that the flow is a parametric minimum flow. □

Theorem 8. *The complexity of the shortest conditional decreasing paths algorithm is $O(n^2m^2K + nmK^2)$, where K is the number of breakpoints of the piecewise linear minimum flow value function.*

Proof. Procedure LENGTH investigates all the m arcs of the network for every node in a conditional decreasing path in each of the n iterations thus the complexity of finding the length of a conditional decreasing directed path is $O(n^2m)$. A conditional decreasing directed path is constructed in $O(n^2)$ time since in each of the n iterations all the n nodes are investigated. The complexity of computing the parametric residual capacity of a conditional decreasing directed path and updating the parametric residual network is $O(nK)$ with $K-1$ being the maximum number of breakpoints of a piecewise linear parametric residual capacity of a conditional decreasing directed path. Thus, the total complexity of performing one step of decreasing the flow is $O(n^2m + nK)$. The number of steps of decreasing the flow is $O(mK)$ since each step reduces the residual capacity to zero of at least one sub-interval of the parameter values bounded by two consecutives breakpoints and for at least one arc. Thus the total complexity of the SCDP algorithm is $O(n^2m^2K + nmK^2)$. □

5 Example

The algorithm is illustrated on the network presented in Figure 1.a and for the parameter λ taking values in the interval $[0, 1]$, i.e. $\Lambda = 1$. The source node is $s = 1$ and the sink node is $t = 3$. The feasible flow f_0 is computed in the non-parametric network $G' = (N, A, \ell', u, s, t)$ presented in Figure 1.b and obtained from the initial network $G(\lambda) = (N, A, \ell(\lambda), u, s, t)$ by replacing the parametric lower bound functions with their maximum values over the entire interval of the parameter $\lambda \in [0, 1]$.

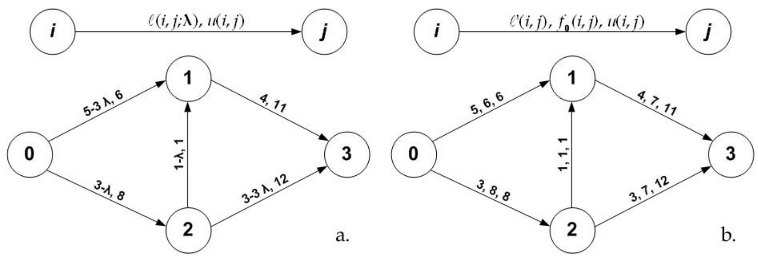

Figure 1: a. The parametric network $G(\lambda)$ with linear lower bound functions and constant upper bounds; b. The feasible flow $f_0(i, j)$ in network $G'(\lambda) = \{N, A, \ell', u, s, t\}$

The parametric residual network $\hat{G}(f_0(\lambda))$ is presented in Figure 2.a. The sub-intervals of the parameter values available for decreasing the flow on every arc are initialised as $\hat{I}(a) = [0, 1]$ for all arcs $a \in \hat{A}(\lambda)$. For practical reasons, in the parametric residual network, the direct arcs are denoted by a_i and the backward arcs are denoted by b_i with $i = 1, \dots, 5$: $(0, 1) = a_1$, $(0, 2) = a_2$, $(1, 3) = a_3$, $(2, 1) = a_4$, $(2, 3) = a_5$ and $(1, 0) = b_1$, $(2, 0) = b_2$, $(3, 1) = b_3$, $(1, 2) = b_4$, $(3, 2) = b_5$ respectively. Using these notations, the lists B_a and R_a are initialized as follows: $B_{a_i} := \{0, 1\}$, $R_{a_i} := \{\hat{r}(a_i; 0), \hat{r}(a_i; \Lambda)\}$, $B_{b_i} := \{0, 1\}$, $R_{b_i} := \{\hat{r}(b_i; 0), \hat{r}(b_i; \Lambda)\}$ for $i = 1, \dots, 5$.

Iteration 1: For the parametric residual network $\hat{G}(f_0(\lambda))$, based on the values of $\hat{I} = [\hat{I}_{jh}]_{(3)\times(3)}$ illustrated in Figure 2.b, procedure LENGTH finds the minimum length of a conditional decreasing directed path $h = 2$ since $\hat{I}_{32} \neq \emptyset$. Starting from $\hat{p}(h) = n - 1 = 3$, procedure DPATH searches the predecessor node belonging to the conditional decreasing directed path by investigating all arcs ending in node $\hat{p}(h) = 3$ and starting from a node with a nonempty value of $\hat{I}_{ih-1} = \hat{I}_{i1}$ i.e. the nonempty elements of the first column of the matrix $\hat{I}$. Choosing $\lambda^* = (0 + 1)/2 = 1/2$ as an arbitrary value $\lambda^* \in \hat{I}_{32}$, both nodes $i = 1$ and $i = 2$ can be selected since the condition $\lambda^* \in \hat{I}_{ih-1} \bigcap \hat{I}(i, \hat{p}(h))$ holds for both arcs $(1, 3)$ and $(2, 3)$. Let node 1 be selected, i.e. $\hat{p}(h-1) = \hat{p}(1) = 1$. Since the last value of the iteration index in procedure DPATH is reached, $\hat{p}(h-2) = \hat{p}(0) = 0$ is set. The conditional decreasing directed path $\hat{P}(\lambda) = (0, 1, 3)$ memorized in $\hat{p} = (0, 1, 3, 4)$ is obtained and the parametric residual capacity $\hat{r}(\hat{P}(\lambda))$

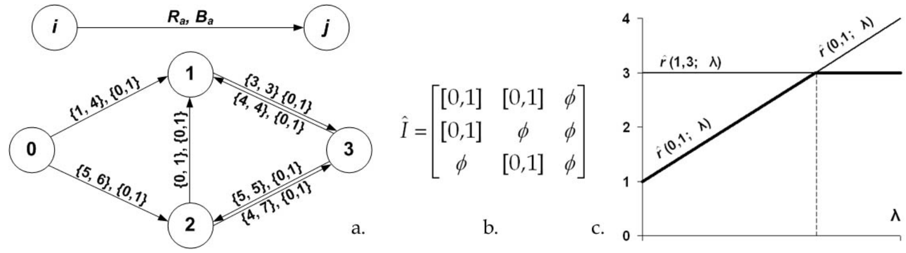

Figure 2: Illustrating the first iteration of the SCDP algorithm. The parametric residual network $\hat{G}(f_0(\lambda))$; b. The sub-intervals of the parameter values $\hat{I} = [\hat{I}_{jh}]$ in the parametric residual network; c. The parametric residual capacity function of the conditional decreasing directed path $\hat{P}(\lambda) = (0, 1, 3)$.

is computed by procedure PRC$((0, 1, 3, 4))$ as follows: $B_{\hat{P}(\lambda)} := B_{a_1} = \{0, 1\}$, $R_{\hat{P}(\lambda)} := R_{a_1} = \{1, 4\}$. The next arc of the conditional decreasing directed path is the arc $a_3 = (1, 3)$ for which all the λ values in $B_{\hat{P}(\lambda)} = \{0, 1\}$ also belong to $B_{a_3} = \{0, 1\}$ so that there is no need to add new breakpoints and the two current variables $\lambda_\ell = \min\{0, 1\} = 0$ and $\lambda_r = \min\{0, 1 \mid \lambda > 0\} = 1$ are computed. The procedure PRC verifies if there is a crossing point of the two parametric residual capacities between $\lambda_\ell = 0$ and $\lambda_r = 1$, i.e. if $[\hat{r}(a_3; \lambda_\ell) - \hat{r}(\hat{P}(\lambda_\ell))] \cdot [\hat{r}(a_3; \lambda_r) - \hat{r}(\hat{P}(\lambda_r))] < 0$.
Since $[3 - 1] \cdot [3 - 4] = -2$, a new breakpoint is found for $\lambda_{mid} = 2/3$ with $\hat{r}(\hat{P}(\lambda_{mid})) = 3$. The new breakpoint is added both to $B_{\hat{P}(\lambda)}$ and B_{a_3} which become $B_{\hat{P}(\lambda)} = \{0, 2/3, 1\}$, $B_{a_3} = \{0, 2/3, 1\}$ and the corresponding parametric residual capacities are updated to $R_{\hat{P}(\lambda)} = \{1, 3, 4\}$ and $R_{a_3} = \{3, 3, 3\}$. Then $\hat{r}(\hat{P}(0))$ is set to $\min\{\hat{r}(\hat{P}(0)), \hat{r}(a_3; 0)\} = \min\{1, 3\} = 1$, and λ_ℓ is set to the value of $\lambda_r = 1$. Since now $\lambda_\ell = \Lambda$ the procedure stops after setting $\hat{r}(\hat{P}(1))$ to $\min\{\hat{r}(\hat{P}(1)), \hat{r}(a_3; 1)\}$, i.e. $\hat{r}(\hat{P}(1)) := \min\{4, 3\} = 3$. The resulting piecewise linear parametric residual capacity function $\hat{r}(\hat{P}(\lambda))$, presented in Figure 2.c. as the inner envelope of the parametric residual capacities of all arcs composing the conditional decreasing directed path $\hat{P}(\lambda) = (0, 1, 3)$, is represented as $R_{\hat{P}(\lambda)} = \{1, 3, 3\}$ for $B_{\hat{P}(\lambda)} = \{0, 2/3, 1\}$ with $K(\hat{P}(\lambda)) = 1$.

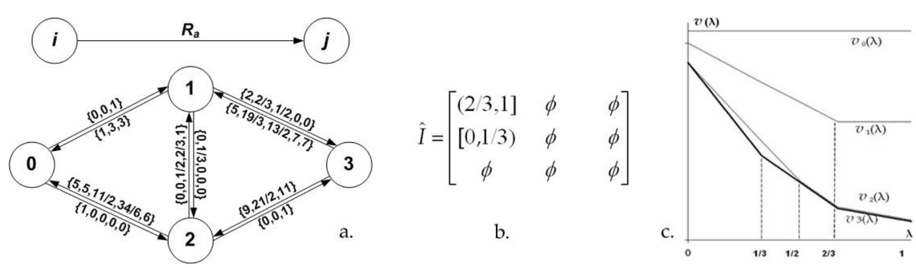

Figure 3: a. The parametric residual network $\hat{G}(f_3(\lambda))$; b. The sub-intervals of the parameter values $\hat{I} = [\hat{I}_{jh}]$ in the parametric residual network; c. The evolution of the piecewise linear flow value function $v_i(\lambda)$ after each of the iterations.

The algorithm makes then a call to procedure UPDATE$((0, 1, 3, 4))$ which updates the parametric residual network $\hat{G}(f(\lambda))$ according to the decreasing of flow along the corresponding path in $G(\lambda)$. This step ends with: $R_{a_1} = \{0, 0, 1\}$, $R_{b_1} = \{1, 3, 3\}$, $R_{a_3} = \{2, 0, 0\}$ and $R_{b_3} = \{5, 7, 7\}$. Finally, the sub-intervals available for decreasing the flow are updated to $\hat{I}(0,1) = [1,1] - [0,2/3] = (2/3,1]$ and $\hat{I}(1,0) = \emptyset \cup [0,2/3] = [0,2/3]$, respectively $\hat{I}(1,3) = [0,2/3)$ and $\hat{I}(3,1) = [0,1]$. After two more iterations, the updated parametric residual network $\hat{G}(f_3(\lambda))$ for the new parametric flow $f_3(\lambda)$ is presented in Figure 3.a. with the lists B_a having the following values: $B_{a_1} = B_{b_1} = \{0, 2/3, 1\}$, $B_{a_2} = B_{b_2} = \{0, 1/3, 1/2, 2/3, 1\}$, $B_{a_3} = B_{b_3} = \{0, 1/3, 1/2, 2/3, 1\}$, $B_{a_4} = B_{b_4} = \{0, 1/3, 1/2, 2/3, 1\}$, $B_{a_5} = B_{b_5} = \{0, 1/2, 1\}$.

By investigating the sub-intervals of the parameter values $\hat{I} = [\hat{I}_{jh}]$ in the updated parametric residual network it can easily be noticed that $\hat{I}_{n-1n-1} = \hat{I}_{33} = \emptyset$ (see Figure 3.b) which means that no conditional decreasing directed path exists in the parametric residual network and the algorithm stops.

The parametric residual capacity functions are presented in Figure 4 for all the sub-intervals of the parameter values defining the parametric maximum partitioning cut: $[\hat{S}_k; J_k]$, $k = 0, \ldots, 3$. For each of the sub-intervals J_k the parametric minimum flow value function equals the parametric capacity function of the corresponding maximum cut: $\hat{c}[\hat{S}_k; J_k] = \ell(\hat{S}_k, \hat{T}_k) - u(\hat{T}_k, \hat{S}_k)$. From these residual capacities, the parametric minimum flow $\hat{f}(\lambda)$ is obtained.

For the parameter value $\lambda = 1/2$ the piecewise linear flow value function does not change the slope but the parametric flow distributes differently over the parametric network arcs. The decreasing of the piecewise linear flow value function after each of the three iterations of the SCDP algorithm is represented in Figure 3.c.

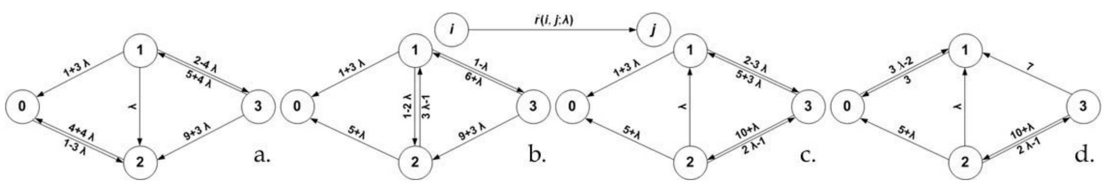

Figure 4: The parametric residual network for each of the sub-intervals J_k, $k = 0, \ldots, 3$ of the parametric maximum cut.

References

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

[2] Ahuja,R., Stein,C., Tarjan,R.E., Orlin,J., Improved algorithms for bipartite network flow. SIAM Journal on Computing, 23(5), (1994), 906-933.

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

[4] Ciurea,E., Ciupala,L., About preflow algorithms for the minimum flow problem. WSEAS Transactions on Computer Research, (2008), 1(3), 35-42.

[5] Ciurea,E., Ciupala,L., Sequential and parallel algorithms for minimum flows. Journal of Applied Mathematics and Computing, (2004), 15(1-2), 53-75.

[6] Ciurea,E., Parpalea,M., Balancing Algorithm for the Minimum Flow Problem in Parametric Bipartite Networks, in Proc. of 14th WSEAS International Conference on Computers, Corfu, Greece, (2010), 226-231.

[7] Gallo,G., Grigoriadis,M.D. and Tarjan,R.E., A fast parametric maximum flow algorithm and applications. SIAM Journal on Computing 18, (1989), 30-55.

[8] Hamacher,H.W., Foulds,L.R., Algorithms for Flows with Parametric Capacities. ZOR - Methods and Models of Operations Research, (1989), 33, 21-37.

[9] Ruhe,G., Complexity Results for Multicriterial and Parametric Network Flows Using a Pathological Graph of Zadeh. Zeitschrift fr Operations Research, (1988), 32, 9-27.

Received: 31.01.2011,

Revised: 30.05.2011,

Accepted: 08.07.2011.

Transilvania University of Brasov,
Department of Theoretical Computer Science,
Bd. Iuliu Maniu 50, 500091, Brasov ROMANIA,
Email: e.ciurea@unitbv.ro
parpalea@gmail.com