DOI: 10.1515/auom-2016-0022
An. Şt. Univ. Ovidius Constanţa

VERSITA
Vol. 24(1),2016, 371–381

A PARAMETRIC NETWORK APPROACH FOR CONCEPTS HIERARCHY GENERATION IN TEXT CORPUS

L.SÂNGEORZAN, M.M.PARPALEA and M.PARPALEA

Abstract

The article presents a preflow approach for the parametric maximum flow problem, derived from the rules of constructing concepts hierarchy in text corpus. Just as generating a taxonomy can be equivalently reduced to ranking concepts within a text corpus according to a defined criterion, the proposed preflow bipush-relabel algorithm computes the maximum flow - the optimum flow that respects certain ranking constraints. The parametric preflow algorithm for generating two level concepts hierarchy in text corpus works in a parametric bipartite association network and, on each step, the maximum possible amount of flow is pushed along conditional augmenting two-arcs directed paths in the parametric residual network, for the maximum interval of the parameter values. The obtained parametric maximum flow generates concepts hierarchies (taxonomies) in text corpus for different degrees of association values described by the parameter values.

1 Introduction

In order to enhance search or browse features, large documents collections are often organized in taxonomies concepts organized in hierarchical structures. Because the manually created taxonomies are extremely time-consuming and

Key Words: Parametric maximum flow, Bipartite network, Preflow bipush-relabel algorithm, Text corpus, Concepts hierarchy, Taxonomy.
2010 Mathematics Subject Classification: Primary 90B10, 90C35; Secondary 90C47, 05C35, 68R10.
Received: 09.09.2014
Accepted: 02.10.2014

costly, they are often incomplete and outdated. Thus, it is highly desirable the taxonomies to be automatically generated. Generally, there are two approaches to taxonomy generation: the query-independent approach, offering a consistent view of the whole corpus, and the query-dependent approach which allows concepts to be organized differently depending on the query [8]. The approach presented in our paper focuses only on the query-dependent approach, being based on statistical co-occurrence in determining the relationship between topics. The algorithm builds taxonomies where each topic is related to distinct subtopics with different degrees of association values, described by the parameter values. The special case of the maximum flow problem [1], [2] in bipartite graphs is motivated by its many natural application problems [3]. Although the improved algorithms for unbalanced bipartite graphs have been known for about 20 years, we are unaware of any published work that deals with flow algorithms in parametric bipartite networks. Further on, the paper is organized as follows: Section 2 deals with the basic terminology and some results concerning flows in parametric networks; Section 3 presents some issues related to the parametric bipartite concepts association network; Section 4 describes the parametric preflow bipush-relabel algorithm for the maximum flow problem in bipartite networks with linear upper bound functions.

2 Flows in parametric networks

The parametric maximum flow problem is an extension of the classical maximum flow problem [5] in which the capacities of certain arcs are functions of a parameter λ . The parametric maximum flow problem is to compute all maximum flows for every possible value of the parameter.

2.1 Terminology and preliminaries

Let $G = (N, A, u, s, t)$ be a capacitated network with $n = |N|$ nodes and $m = |A|$ arcs, N being the set of nodes i and A being the set of arcs a, so that every arc in A can be written as $a = (i, j)$ with $i, j \in N$. The capacity (*upper bound*) function is a nonnegative function, $u(a)$ associated with each arc $a \in A$. The network has two special nodes: a source node s and a sink node t. A *cut* is a partition of the node set N into two subsets S and $T = N - S$, denoted by $[S, T]$. 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 (T, S) denote the set of backward arcs. A cut $[S, T]$ is an $s - t$ cut if $s \in S$ and $t \in T$. A network G is called *bipartite* if its node set N can be partitioned into two subsets N_1 and N_2 such that all arcs have one endpoint in

N_1 and the other in N_2. Let $n_1 \in |N_1|$ and $n_2 \in |N_2|$. A bipartite network is referred to as *unbalanced* if $n_1 \ll n_2$ or $n_2 \ll n_1$ and *balanced* otherwise. For a bipartite network G, the extended network $\tilde{G} = (\tilde{N}, \tilde{A})$, with $\tilde{N} = N \cup \{s, t\}$ and $\tilde{A} = A \cup \{(s, j) | \forall j \in N_1\} \cup \{(i, t) | \forall i \in N_2\}$, is constructed by adding two special nodes, the source node s and the sink node t, considering that $s \in |N_2|$ and $t \in |N_1|$. For all arcs (s, j) and (i, t) the upper bounds of the arcs are $\tilde{u}(s, j) := \infty$, respectively $\tilde{u}(i, t) := \infty$ while the upper bounds of all the other arcs are $\tilde{u}(i, i) := u(i, j)$.

2.2 The parametric maximum flow

The parametric flow problem consists in generalising the classic problem of flows in networks by transforming the upper bounds of some arcs $(i, j) \in A$ of the network $G = (N, A, u, s, t)$ in linear functions of a real parameter.

Definition 1. *A directed network $G = (N, A, u, s, t)$ 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 $\bar{G} = (N, A, \bar{u}, s, t)$.*

For a parametric network $\bar{G}$, the *parametric upper bound* (*capacity*) function $\bar{u} : A \times [0, \Lambda] \to \Re^+$ associates to each arc $(i, j) \in A$, for each of the parameter values $\lambda \in [0, \Lambda]$, the real number $\bar{u}(i, j; \lambda)$, referred to as the parametric upper bound of arc (i, j): $\bar{u}(i, j; \lambda) = u_0(i, j) + \lambda \cdot U(i, j)$, $(i, j) \in A$, where $U : A \to \Re$ 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). The nonnegative value $u_0(i, j)$ is the upper bound of the arc (i, j) for $\lambda = 0$, i.e. $\bar{u}(i, j; 0) = u_0(i, j)$ with $0 \leq u_0(i, j)$. For the problem to be correctly formulated, the upper bound function of every arc $(i, j) \in A$ must respect the condition $0 \leq \bar{u}(i, j; \lambda)$ 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 upper bounds $U(i, j)$ must satisfy the constraint: $U(i, j) \geq -u_0(i, j)/\Lambda$, $\forall (i, j) \in A$.

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 $\bar{v}(i; \lambda)$ referred to as the value of node i for each of the parameter λ values.

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

$$\sum_{j|(i,j)\in A} \bar{f}(i, j; \lambda) - \sum_{j|(j,i)\in A} \bar{f}(j, i; \lambda) = \bar{v}(i; \lambda),\ \forall i \in N,\ \forall \lambda \in [0, \Lambda], \quad (1)$$

$$0 \leq \bar{f}(i, j; \lambda) \leq \bar{u}(i, j; \lambda), \qquad \forall (i, j) \in A, \qquad \forall \lambda \in [0, \Lambda], \quad (2)$$

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

The parametric maximum flow (PMF) problem is to compute all maximum flows for every possible value of λ in $[0,\Lambda]$:

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

$$\sum_{j|(i,j)\in A} \bar{f}(i,j;\lambda) - \sum_{j|(j,i)\in A} \bar{f}(j,i;\lambda) = \begin{cases} \bar{v}(\lambda), & i = s \\ 0, & i \neq s,t \\ -\bar{v}(\lambda), & i = t \end{cases} \tag{4}$$

$$0 \leq \bar{f}(i,j;\lambda) \leq \bar{u}(i,j;\lambda), \qquad \forall (i,j) \in A. \tag{5}$$

Definition 3. *A* parametric $s-t$ cut partitioning *denoted by* $[S_k;J_k]$, $k = 0,\ldots,K$, *is defined as a finite set of cuts* $[S_k,T_k]$, $k = 0,\ldots,K$, *together with a partitioning of the interval* $[0,\Lambda]$ *of the parameter in disjoints subintervals* J_k, $k = 0,\ldots,K$, *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, $k = 0,\ldots,K$, *defined as:*

$$\tilde{\bar{c}}[S_k;J_k] = \sum_{(i,j)\in(S_k,T_k)} \bar{u}(i,j;\lambda), \quad k = 0,\ldots,K. \tag{6}$$

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* $[\tilde{S}_k;J_k]$, $k = 0,\ldots,K$.

Theorem 1. [5] *The value function* $\tilde{\bar{v}}$ *of the parametric maximum flow* $\tilde{\bar{f}}$ *from a source* s *to a sink* t *in network* $\bar{G}$ *equals the capacity* $\tilde{\bar{c}}$ *of the parametric minimum* $s-t$ *cut* $[\tilde{S}_k;J_k]$, $k = 0,\ldots,K$.

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 is given by:* $\tilde{\bar{r}}(i,j;\lambda) = \bar{u}(i,j;\lambda) - \bar{f}(i,j;\lambda) + \bar{f}(j,i;\lambda)$.

Definition 7. *The subintervals* $\tilde{I}(i,j) \subseteq [0,\Lambda]$ *where an augmentation of the flow* $\bar{f}(i,j;\lambda)$ *is possible along the arc* (i,j) *are defined as:* $\tilde{I}(i,j) = \{\lambda | \tilde{\bar{r}}(i,j;\lambda) > 0\}, \quad (i,j) \in A$.

Definition 8. *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 $\tilde{A}(\bar{f}) = \{(i,j)|(i,j) \in A,\ \tilde{I}(i,j) \neq \emptyset\}$ 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 9. *The* parametric excess *of a node $i \in N$ is defined as:*

$$\bar{e}(i;\lambda) = \sum_{j|(j,i)\in A} \bar{f}(j,i;\lambda) - \sum_{j|(i,j)\in A} \bar{f}(i,j;\lambda). \tag{7}$$

Definition 10. *The sets $\tilde{I}(i) = \{\lambda|\bar{e}(i;\lambda) > 0\}$ for $i \in N - \{s,t\}$ describe subintervals of $[0,\Lambda]$ where the excess of node i is positive.*

In the residual network $\tilde{\bar{G}}(\bar{f})$ the *distance function* $\tilde{d} : N \to \aleph$ is a function from the set of nodes to the nonnegative integers. A distance function is said to be *valid* if it satisfies the following conditions: $\tilde{d}(t) = 0$ and $\tilde{d}(i) \leq \tilde{d}(j) + 1$, $\forall (i,j) \in \tilde{A}$.

Definition 11. *An arc $(i,j) \in \tilde{A}$ in the parametric residual network $\tilde{\bar{G}}(\bar{f})$ is referred to as* conditionlly admissible *if both $\tilde{d}(i) = \tilde{d}(j)+1$ and $\tilde{I}(i,j) \bigcap \tilde{I}(i) \neq \emptyset$; otherwise it is* conditionally inadmissible.

3 Concepts association network

The first step of concepts hierarchy generation in text corpus consists in extracting a set of candidate concepts from the corpus and computing the co-occurrences between concepts. Based on the most frequent concepts appearing in the titles, the set of candidate concepts can be easily obtained. For each concept pair, its co-occurrence within the document abstract can be also computed.

Once the set of concepts is extracted and their co-occurrence is calculated, the concepts association network, encompassing all concepts, is constructed. The network is built as a bipartite directed graph $G = (N_1 \cup N_2, A)$ with weighted arcs. Each concept c_i is a node in G, i.e. $\forall i \leq n_1 + n_2, c_i \in N_1 \cup N_2$. There exists an arc $a_{i,j} = (c_i, c_i) \in A$ between the topics c_i and c_j if and only if c_i co-occurs with c_j in at least one document. The source node of the extended bipartite network $G = (N \cup \{s,t\}, A)$ represents the topic of the query-dependent taxonomy.

The *co-occurrences of the concepts pairs* within documents are computed as $\rho_{i,j} = count(c_i, c_j)$, i.e. the number of documents that contain both c_i and c_i in the title or document abstract.

The *external degree of a concept* c_i, denoted by k_i, represents the total number of arcs emerging from node c_i.

The *strength of a concept*, representing the concepts importance, is simply the sum of all its co-occurrences:

$$\sigma_i = \sum_{c_j | (c_i, c_j) \in A} count(c_i, c_j). \tag{8}$$

The *parametric upper bound* $\bar{u}(c_i, c_j; \lambda) = u_0(c_i, c_j) + \lambda \cdot U(c_i, c_j)$ of an arc (c_i, c_j) is computed as:

$$\bar{u}(c_i, c_j; \lambda) = \frac{\sigma_i}{k_i} + \lambda \cdot \rho_{i,j}, \qquad \lambda \in [0, \Lambda]. \tag{9}$$

For any arc $a_{i,j} \in A$ of the bipartite network, the node (endpoint) corresponding to a concept c_i with a greater strength σ_i belongs to subset N_1 and the other one belongs to N_2. For the extended network $\tilde{G} = (\tilde{N}, \tilde{A})$, constructed by adding the source and the sink nodes, the capacities of the arcs (s, c_i) are set to $\bar{u}(s, c_i) := \sigma_i$ while those of the arcs (c_j, t) are $\bar{u}(c_j, t) := \infty$.

4 Maximum flow problem in bipartite parametric network

All push-relabel algorithms, maintaining a preflow at every stage, work by examining active nodes and pushing excesses from these nodes to nodes estimated to be closer to the sink t. If t is not reachable, however, an attempt is made to push the deficit back to s. If eventually there will be no excess on any node other than s, the preflow is a flow which is also a maximum one. A push (augmentation of the flow on an arc) of $\delta(\tilde{P}; \lambda)$ units of flow over an arc $(i, j) \in \tilde{P}$ is referred to as *saturating* if $\delta(\tilde{P}; \lambda) = \tilde{\bar{u}}(i, j; \lambda)$ and *nonsaturating* otherwise. A nonsaturating push from node i decrease the excess $\bar{e}(i; \lambda)$ to zero. The process of modifying the value of a distance label of a node is referred to as a *relabel* operation.

Definition 12. *A push of* $\delta(\tilde{P}; \lambda)$ *units of flow on a two arcs path* $\tilde{P} = ((i, j), (j, k)) = (i, j, k)$ *in the bipartite network with* $j \in N_1$ *and* $i, k \in N_2$ *is referred to as a* bipush.

4.1 Conditional augmenting two-arcs directed paths in the residual network

A *conditional augmenting two-arcs directed path* $\tilde{P}$ in $\tilde{\bar{G}}(\bar{f}) = (N, \tilde{A})$ is a directed path formed by two consecutive conditionally admissible arcs (i, j) and (j, k) from the active node i, with $j \in N_1$ and $i, k \in N_2$, such that:

$$\tilde{I}(\tilde{P}) = \tilde{I}(i) \bigcap \tilde{I}(i, j) \bigcap \tilde{I}(j, k) \neq \emptyset. \tag{10}$$

Theorem 2. *If no conditional augmenting two-arcs directed path exists in* $\tilde{\bar{G}}(\tilde{\bar{f}})$ *then* $\tilde{\bar{f}}$ *is a maximum parametric flow.*

Proof. Suppose that no conditional augmenting two-arcs directed path exists in $\tilde{\bar{G}}(\tilde{\bar{f}})$ and yet $\tilde{\bar{f}}$ is not a maximum parametric flow, i.e., there exists $\lambda^* \in [0, \Lambda]$ such that $\bar{v}(\lambda^*) < \tilde{\bar{v}}(\lambda^*)$. Consequently, $\tilde{\bar{f}}(\lambda^*)$ is therefore a feasible but not maximum flow with respect to the fixed upper bounds $\bar{u}(i, j; \lambda^*)$. Hence, there exists an augmenting directed path $\tilde{P}$ with respect to $\tilde{\bar{f}}(\lambda^*)$. But the definition of $\tilde{\bar{G}}(\tilde{\bar{f}})$ yields that $\tilde{P}$ is also a conditional augmenting directed path with $\tilde{I}(\tilde{P}) = \{\lambda^*\}$ which contradicts the non-existence of a conditional augmenting two-arcs directed path. Thus $\tilde{\bar{f}}$ is a maximum parametric flow. □

In order to avoid working with conditional augmenting two-arcs directed paths, i.e. directed paths that respect the condition (10), our algorithm uses a partitioning technique which allows him to work independently on each subinterval of the value range of the parameter.

4.2 Parametric bipartite bipush-relabel (PBB) algorithm

Using a multi-thread parallel implementation, after computing the breakpoint of a parametric residual capacity of a conditional decreasing two-arcs directed path, the algorithm continues to run independently on each subinterval between two consecutive breakpoints, on independent threads.

In the *INITIALISATION* step of the algorithm, for each arc $(s, i) \in \tilde{A}$ the flow is set to its upper bound value, $\bar{f}(s, i; \lambda) := \bar{f}(s, i; \lambda)$ and thus nodes i become active nodes (i.e. $\tilde{e}(i; \lambda) := \bar{u}(s, i; \lambda)$) and are added to a list L.

After the initialization step, the algorithm performs the *BIPUSH* procedure for any selected node from the list L, while adding newly created active nodes to L. The algorithm terminates when L becomes empty. The procedure consecutively finds admissible two-arcs directed paths (i, j, k) and pushes the flow over them. The step of pushing the flow, which augments the flow over the two-arcs directed path with $\delta = \min\{\bar{e}(i; \lambda),\ \bar{r}(i, j; \lambda),\ \bar{r}(j, k; \lambda)\}$, consists in the following:

$\tilde{I}(i) := \tilde{I}(i) - J_1$, $\tilde{I}(i,j) := \tilde{I}(i,j) - J_2$, $\tilde{I}(j,k) := \tilde{I}(j,k) - J_3$, $\tilde{I}(k) := J_1$;
$\bar{r}(i,j;\lambda) := \bar{r}(i,j;\lambda) - \delta$, $\bar{r}(j,i;\lambda) := \bar{r}(j,i;\lambda) + \delta$;
$\bar{r}(j,k;\lambda) := \bar{r}(j,k;\lambda) - \delta$, $\bar{r}(k,j;\lambda) := \bar{r}(k,j;\lambda) + \delta$;

$$\bar{e}(i;\lambda) := \bar{e}(i;\lambda) - \delta,\ \bar{e}(k;\lambda) := \bar{e}(k;\lambda) + \delta,\ \delta = \begin{cases} \bar{e}(i;\lambda), & for\ \lambda \in J_1 \\ \bar{r}(i,j;\lambda), & for\ \lambda \in J_2 \\ \bar{r}(j,k;\lambda), & for\ \lambda \in J_3 \end{cases}.$$

During each stage of pushing of flow, the range of parameter values is partitioned in the subintervals $J_1 := \{\lambda | \bar{e}(i;\lambda) \leq \min\{\bar{r}(i,j;\lambda),\ \bar{r}(j,k;\lambda)\}\}$, $J_2 := \{\lambda | \bar{r}(i,j;\lambda) \leq \min\{\bar{e}(i;\lambda),\ \bar{r}(j,k;\lambda)\}\}$ and $J_3 := J - (J_1 \bigcup J_2)$ within which the algorithm independently reiterates.
Putting all together, the *parametric bipartite preflow bipush-relabel* (PBB) algorithm can be written as:

```
ALGORITHM PBB
INITIALISATION
BIPUSH(L, [0, Λ])
```

The algorithm starts with $\tilde{I}(i,j) = [0, \Lambda]$ for all arcs $(i,j) \in A$ and stops when the residual network contains no active nodes.

Theorem 3. (Theorem of correctness) *The parametric bipartite preflow bipush-relabel (PBB) algorithm computes correctly a parametric maximum flow.*

Proof. The proof of the theorem follows from the correctness of the *preflow bipush-relabel* algorithm for each of the subintervals of the parameter values. When the algorithm terminates, let S_p be the set of all nodes which are reachable from the source node s within the subinterval J_p. For the resulting cuts $[S_p, T_p]$ and the intervals J_p, $p = 1, 2, \cdots K$, the following observations hold:
(i) IF $i \in S_p$, $j \in T_p$ and $(i,j) \in A$ THEN $J_p \bigcap \tilde{I}(i,j) = \emptyset$ for otherwise node j could be reached from s in $\tilde{\bar{G}}(\tilde{\bar{f}})$. Hence, $\tilde{\bar{f}}(i,j;\lambda) = \bar{u}(i,j;\lambda)$; $\forall \lambda \in J_p$, by the definition of $\tilde{I}(i,j) = 0$.
(ii) IF $i \in T_p$, $j \in S_p$ and $(i,j) \in A$ THEN $J_p \bigcap \tilde{I}(j,i) = \emptyset$ for otherwise node i could be reached from s in $\tilde{\bar{G}}(\tilde{\bar{f}})$. Hence, $\tilde{\bar{f}}(i,j;\lambda) = 0$; $\forall \lambda \in J_p$. Summarizing, $\sum_{(i,j)\in(S_p,T_p)} \tilde{\bar{f}}(i,j;\lambda) = \sum_{(i,j)\in(S_p,T_p)} \bar{u}(i,j;\lambda)$ and $\sum_{(i,j)\in(T_p,S_p)} \tilde{\bar{f}}(i,j;\lambda) = 0$, thus the obtained flow is a parametric maximum flow which equals the parametric capacity of the minimum $s-t$ cut. □

4.3 Complexity issues

A breakpoint is a place where the slope of the optimal value function of a parametric optimization problem is changing. One approach to express complexity

of a parametric problem is to count the number of breakpoints (K). Unfortunately, in the worst case the number of breakpoints may be exponential in the size of the problem. The example originates from the pathological graph of Zadeh [6] which was used to show that the complexity may be exponential. For the parametric maximum flow problem in bipartite networks however the parametric bipartite preflow bipush-relabel (PBB) algorithm overcomes this inconvenient by using the multi-thread parallel implementation of a non-parametric approach. The main idea of this implementation is to assign a thread to each newly generated subinterval which will carry out the problem forward from the current configuration of the problem. Each thread continues the problem on its own subinterval separately from all the others without any need of synchronizing. The only common variables are the distance labels and, with the purpose to achieve the above independence of the threads, for each of the new generated subintervals a copy of the current distance labels values is generated so that they can be independently modified in the further parallel evolution of the algorithm.

Theorem 4. (Theorem of complexity) *The parametric bipartite preflow bipush-relabel (PBB) algorithm solves the parametric maximum flow problem in* $O(n_2^3 + n_2 m + Kn)$ *time.*

Proof. The initialization procedure and the computation of the exact distance labels are performed in $O(n_2)$ and $O(m)$ time respectively. The complexity of the non-parametric bipush-relabel algorithm, equal to the complexity of the parametric one for a non-partitioned subinterval of the parameter values is $O(n_2^3 + n_2 m)$. New copies of distance labels values are generated every time a breakpoint occurs, i.e. copying distance labels takes $O(Kn)$ time where K is the number of breakpoints. Thus, the total complexity of the algorithm is $O(n_2^3 + n_2 m + Kn)$. □

5 Conclusions

Based on the parametric maximum flow, the sets S_k, $k = 0, ..., K$, generated by the parametric $s - t$ cut partitioning, gather the concepts in classes which are ordered in hierarchies, i.e. taxonomy. Thus, maximum flow problem in parametric networks turns out to be an important scenario in practice since the complexity of its solving algorithm can be reduced to that of the equivalent non-parametric algorithm considering the approach of network partitioning.

References

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

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

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

[4] Goldberg,A., *Two-Level Push-Relabel Algorithm for the Maximum Flow Problem* Lecture Notes in Computer Science, Springer, vol.5564(2009), 212–225

[5] Parpalea,M., *Min-Max algorithm for the parametric flow problem*, Bulletin of the Transilvania University of Brasov, Series III: Mathematics, Informatics, Physics, Vol.3, **52** (2010), 191–198

[6] Ruhe,G., *Complexity Results for Multicriterial and Parametric Network Flows Using a Pathological Graph of Zadeh*, Zeitschrift fur Oper. Res. **32** (1988), 9–27

[7] Sângeorzan,L., Parpalea,M., Parpalea,M.M., *Partitioning Preflow-pull Algorithm for the Parametric Net-work Flow Problem a Linguistic Rule-based Constraints Optimisation Approach*, Recent Advances in Neural Networks, Fuzzy Systems and Evolutionary Computing, Proceedings of the 11th WSEAS International Conference on Neural Networks, Iasi, Romania, 2010, 111–116

[8] Treeratpituk,P., Khabsa,M., Giles,C.L., *Graph-based Approach to Automatic Taxonomy Generation (GraBTex)*, CoRR, arXiv:1307.1718 (2014), [cs.IR]

Livia SÂNGEORZAN,
Faculty of Mathematics and Computer Science,
Transilvania University of Brasov,
Eroilor blvd 25, 500030 Brasov, Romania.
Email: sangeorzan@unitbv.ro

Mihaela Marinela PARPALEA,
Faculty of Letters,
Transilvania University of Brasov,
Eroilor blvd 25, 500030 Brasov, Romania.
Email: mihaela.parpalea@unitbv.ro

Mircea PARPALEA,
National College Andrei Şaguna,
Şaguna str. 1, 590000, Brasov, Romania.
Email: parpalea@gmail.com