Generalized Parametric Multi-Terminal Flows Problem

Pascal Berthomé[1], Madiagne Diallo[2], and Afonso Ferreira[3*]

[1] Laboratoire de Recherche en Informatique (LRI), UMR 8623, CNRS, Université Paris-Sud, 91405, Orsay-Cedex, France, Pascal.Berthome@lri.fr
[2] Laboratoire PRiSM, Université de Versailles, 45 Av. des Etats-Unis, 78035 Versailles-Cedex, France, Madiagne.Diallo@prism.uvsq.fr
[3] CNRS, I3S, INRIA-Sophia Antipolis, Projet Mascotte, 2004 Rt. des Lucioles, BP93, F-06902 Sophia Antipolis, France, Afonso.Ferreira@inria.fr

Abstract. Given an undirected edge-weighted n-nodes network in which a single edge-capacity is allowed to vary, Elmaghraby studied the sensitivity analysis of the multi-terminal network flows. The procedure he proposed requires the computation of as many Gomory-Hu cut trees as the number of critical capacities of the edge, leading to a pseudo-polynomial algorithm.
In this paper, we propose a fully polynomial algorithm using only two Gomory-Hu cut trees to solve the Elmaghraby problem and propose an efficient generalization to the case where k edge capacities can vary. We show that obtaining the all-pairs maximum flows, for the case where k edge capacities vary in the network, is polynomial whenever $k = O(\text{polylog } n)$, since we show that it can be solved with the computation of 2^k Gomory-Hu trees.

Keywords: Multi-Terminal Flows, Gomory-Hu Cut Tree, Parametric Flows, Dynamic Networks, Max-Flow, Min-Cut, Sensitivity Analysis, Network Management.

1 Introduction

In the late 1950's, the single source–single terminal maximum flow problem was popularized by the resolution of Ford and Fulkerson [9]. They specially showed the connection between the maximum flow and the min cut problems in extension of Menger's theorem.

In the setting of a connected, undirected graph with constant edge-weights, the multi-terminal network flows problem consists in finding the all pairs maximum flows in the network. Clearly, this problem is solvable with $n(n-1)/2$ single source–single terminal maximum flow computations. In 1961, Gomory and Hu [10] delivered an ingenious method to solve this maximum flow analysis problem using only $n-1$ maximum flow computations. They summarized their

* Partially supported by the Europeans projects RTN ARACNE and FET CRESCCO, and by the French AS *Dynamo*.

results in a tree referred to as GH cut tree (in the literature) reflecting the all pairs maximum flows in the network. Later in 1990, Gusfield [11] provided a simpler procedure to obtain the GH cut tree but using also $n-1$ maximum flow computations.

In 1964, Elmaghraby [8] was the first to extend the former problem to the case where a single edge is allowed to vary in capacity. He did the sensitivity analysis of the multi-terminal network flows, *i.e.*, the analysis of the capacity variation effects on the all pairs maximum flows in the network. As far as the variation is concerned, some maximum flows will be affected by the variation and others not. This give rise to the concept of *critical capacities*, *i.e.*, values for which a current flow changes of behavior by either beginning to vary with the parametric capacity or stopping to vary and remaining constant for the rest of the parameterization process. Thus the former sensitivity analysis problem turns to be the one of finding all the *critical capacities*, since the flows remain unchanged when the parametric capacity vary between two *critical capacities*. To solve this problem, Elmaghraby proposed an algorithm that computes as many GH cut trees as the number of possible critical capacities, leading to a pseudo-polynomial algorithm.

Recently, Diallo and Hamacher [6], showed that the analysis in [8] fails to determine all affected flows and provided an algorithmic improvement on the Elmaghraby analysis without improving the pseudo-polynomiality of the algorithm. In [5], it is proposed some pseudo-polynomial heuristics for the generalization of the Elmaghraby method.

In this paper we improve all methods and propose an efficient generalization of the sensitivity analysis of multi-terminal network flows. To do so, we start first by showing a fully polynomial algorithm that requires the computation of only two Gomory-Hu trees in order to solve the Elmaghraby sensitivity analysis problem. Second, since our algorithm is very simple compared to the existing pseudo-polynomial solution, it can be generalized to many varying capacities. Our main result is then that obtaining the all-pairs maximum flows for any value of the parameters, in the case where k edge capacities vary in the network, is polynomial whenever $k = O(\text{polylog } n)$, since we show that it can be solved with the computation of 2^k Gomory-Hu trees. In particular, the critical capacities become not a central key but a simple consequence of our results, since for more than one parametric edge-capacity, this notion of critical capacity may become more complex than obtaining directly the values of the maximum flows.

The multi-terminal network flows problem with constant capacity has many known applications in the fields of transports, energy and telecommunications (see for example [4, 5, 8] and references therein). The parametric multi-terminal network flows problem also reflects in these fields problems including link breakdown, capacity improvement, bandwidth reservation, network expansion.

In the remainder, we present in Section 2 some basic definitions and briefly describe the main ideas of Elmaghraby's method, and its improvements by Diallo and Hamacher. Section 3 is devoted to the details of our method in the case of a single edge-capacity variation. In Section 4, we provide the generalization to several parametric edge-capacities. We close the paper with concluding remarks and perspectives. Interested readers are referred to the technical report [2] where some examples illustrate our algorithms.

2 Definitions and Preliminaries

In this section we provide a precise definition of a GH cut tree and some preliminary results. Throughout this paper, we assume that the reader is familiar with general concepts of graph theory and network flows. For example, we refer to [1, 9, 12].

2.1 Definitions

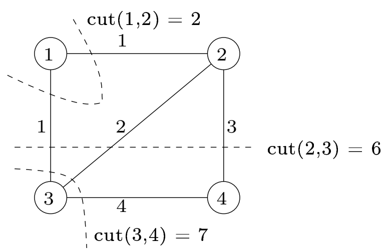
Let G be a connected undirected graph. We denote by $V(G)$ the vertex-set of G and by $E(G)$ its edge-set. A *network* is intended to be G associated to an edge-weight function $c : E(G) \rightarrow R^+$ called *edge-capacity* function. A flow from a source vertex s to a terminal vertex t in G is given by a function $f : E(G) \rightarrow R^+$. The flow has to be conserved in each vertex, excepted in vertices s and t, *i.e*, $\forall u \in V \setminus \{s,t\}$, $\sum_{v \in V} f(u,v) = 0$. We notice that the graphs we deal with are symmetric thus each undirected edge $[u,v]$ can be replaced by two directed arcs (u,v) and (v,u), both with the same arc capacity, *i.e.*, $c(u,v) = c(v,u)$. For each $(u,v) \in E(G)$, $f(u,v) \leq c(u,v)$. For the sake of simplicity, we denote the maximum flow from a source s to a terminal t as f_{st}.

Definition 1. (GH cut tree) *Given a network $G = (V,E)$, a GH cut tree $T = (V,F)$ obtained from G is a weighted tree with the same set of vertices V with the two following properties:*

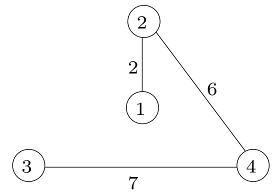
equivalent flow tree: *the value of the maximum flow between any $s,t \in G$ is equal to the value of the maximum flow in T between s and t, i.e., the smallest of the capacities of the edges on the unique path from s to t in T; thus the maximum flows between all pairs of vertices in G are represented in T;*

cut property: *the removal of any edge of capacity c from T separates its vertices into two classes, where the cut in G given by this partition has capacity c as well.*

In Figure 1(b), we illustrate a GH cut tree T obtained from the given network G with the illustrated minimum cuts in Figure 1(a). As shown in [10], $n-1$ min-cut computations are sufficient to obtain the global structure of the GH cut tree. We notice that GH cut trees are not unique. Algorithms to compute a GH cut tree are provided in [10] and in [11]. An experimental study of minimum cut algorithms and a comparison of algorithms producing GH cut trees are provided in [3].



(a) G



(b) GH tree

Fig. 1. A graph and one of its Gomory-Hu cut trees

2.2 Sensitivity analysis and all critical capacities

A natural extension of the multi-terminal network flows problem with constant edge-capacities is the analysis of the effects of a single edge-capacity parameterization on the all pairs maximum flows. This was effectively the aim of Elmaghraby in [8]. He mainly described a procedure to analyze the case where a single capacity decreases linearly and gave a sketch of the procedure to solve the capacity increasing case.

The sensitivity analysis problem solved in [8] can be stated as follows. Given the former network description, and an edge $e = (i, j)$ of $E(G)$ with a capacity given by $c(e) = \bar{c} - \epsilon$, $0 \leq \epsilon \leq \bar{c}$, where $\bar{c}$ is the initial capacity, one has to determine the set of all pairs maximum flows for all values of ϵ.

The Elmaghraby procedure can be briefly described as follows. With the decrease on the capacity $c(e)$, it is clear that some maximum flow values will be modified and others not. If the edge e is present in a minimum cut, this implies a reduction in the respective maximum flow value. As far as the variation is concerned, some maximum flow values that were constant may begin to decrease linearly with respect to ϵ. The value of $c(e)$ for which some flow changes of behavior is called *critical capacity.* Thus, in the interval between two critical capacities prevails the *status quo*, *i.e.* , no maximum flow behavior change occurs. With this remark, Elmaghraby transfers the sensitivity analysis of multi-terminal network flows to the problem of determining all the critical capacities, since in each such interval a GH cut tree computation provides the desired maximum flows.

In order to obtain the first critical capacity $\hat{\lambda}^0$, a GH cut tree is computed with the parameter set to zero. From this cut tree, an incident-like $E(G) \times (n-1)$ matrix is constructed [7]. This matrix defines the minimum cuts in the cut tree in terms of the original edges in G. Thus, based on this matrix, one identifies the set $\mathcal{C}$ of the minimum cuts that contain the parameterized edge and its complement set $\bar{\mathcal{C}}$ in the cut tree. The critical capacity $\hat{\lambda}^0$ is then the smallest value of the parameter λ for which a cut would move from $\bar{\mathcal{C}}$ to $\mathcal{C}$. With $\hat{\lambda}^0$ at hand, in order to compute the next critical value, we decrease the parametric capacity in G by

$\hat{\lambda}^0$ and repeat the same procedure with this updated capacity. The analysis ends when the parametric capacity is null, *i.e.*, $\epsilon = \bar{c}$.

In [8], it is described with detail how to obtain the critical capacities. We just emphasize that *to obtain each critical capacity, a GH cut tree computation is needed, thus leading to a pseudo-polynomial algorithm.*

Furthermore, when the parameter is negative, *i.e.*, when the capacity of the edge increases, Elmaghraby proposed to set the capacity of the tested edge to a big enough value and do the backward process using the decreasing case procedure to determine the critical capacities. What again leads to another pseudo-polynomial algorithm with some additional operations.

In [6], Diallo and Hamacher showed that the $E(G) \times (n-1)$ matrix used by Elmaghraby [7] to determine the minimum cuts that contain the investigated edge may fail to provide all such minimum cuts. They provided a counter-example and deliver a very simple algorithm to generally test whether or not a minimum cut between a given pair of vertices contains a given edge.

In the sequel, we show that if a single capacity is varying, then two GH cut trees are enough to compute all the critical capacities. Furthermore, the method we provide delivers simultaneously the critical capacities for both the capacity decreasing and capacity increasing problems. Nevertheless, the advantage of Elmaghraby's method is that one can obtain a minimum cut for any value of the parameter.

3 Computing critical capacities with two GH cut trees

In this section, we show that if a single capacity is parameterized, then only two GH cut trees are needed to obtain all the critical capacities. To do so, we first provide a way to obtain the critical capacity with respect to a single maximum flow, and then, in Section 3.2, we explore this result to obtain the critical capacities for the all pairs maximum flows.

3.1 Critical capacity with respect to a single maximum flow

For s and t two vertices of G, we define $f_{s,t}(\lambda)$ as the value of the maximum flow between s and t when the capacity of the edge e is λ. We denote by $f^0_{s,t}$ (or simply f^0) the maximum flow $f_{s,t}(0)$, *i.e.* , when the edge e is removed from the network, and by $f^\infty_{s,t}$ (or simply f^∞) the $\lim_{\lambda \to \infty} f_{s,t}(\lambda)$, *i.e.* , the maximum flow when there is no constraint on the edge e. This latter value is finite for all pairs, except when $(s,t) = e$. It can be simply computed by setting the capacity of the tested edge e to the sum of the capacities of its adjacent edges.

One interesting point is to observe the global behavior of the function $f_{s,t}(\lambda)$, for a given pair (s,t), for which the maximum value changes of behavior during the parameterization. As shown in Figure 2, it is composed by two distinct parts:

- as far as the capacity λ of the edge e increases, the maximum flow increases in the same way. During this stage, the parameterized edge is present in any s,t-minimum cut;

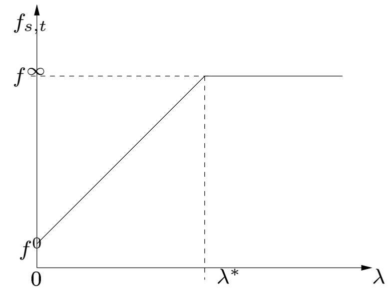


Fig. 2. behavior of a sensitive maximum flow function $f_{s,t}(\lambda)$

– at some value λ^* of λ, namely the critical capacity, the maximum flow becomes saturated (except if $\{s,t\} = e$). During this stage, the parameterized edge is out of all s,t-minimum cuts.

However, a maximum flow f_{st} may never depend on the parameter, in this case, $f^0_{s,t} = f^{\infty}_{s,t}$, thus we admit that its critical capacity is $\lambda^*_{s,t} = 0$.

Notice that $f_{ij} \to \infty$ as $\lambda \to \infty$, thus by convention we also admit that $f^{\infty}_{i,j} = \infty$.

Lemma 1. *Let $G = (V,E)$ be a network and $e = (i,j)$ one edge of $E(G)$ with parametric capacity $\lambda \geq 0$. Let p and q be two vertices of G. The critical capacity $\lambda^*_{p,q}$ exists if $\{p,q\} \neq \{i,j\}$ and satisfies:*

$$\lambda^*_{p,q} = f^{\infty}_{p,q} - f^0_{p,q}. \tag{1}$$

Proof. This is a direct consequence of the behavior of the maximum flow function: it grows linearly from $f^0_{p,q}$ up to $f^{\infty}_{p,q}$, thus the breakpoint is when the capacity equals to $f^{\infty}_{p,q} - f^0_{p,q}$. □

Corollary 1. *The critical capacity λ^* of λ for an arbitrary maximum flow can be computed using only two maximum flow computations.*

Proof. Using Lemma 1, we deduce that only f^0 and f^{∞} are necessary to compute λ^*. Furthermore, in order to compute f^{∞}, the result of f^0 can be used as initial value. □

Corollary 2. *Let $G = (V,E)$ be a network and $e = (i,j)$ one edge of $E(G)$ with capacity $\lambda \geq 0$. Let s and t be two vertices of G. The maximum flow $f_{s,t}(\lambda)$ verifies:*

$$f_{s,t}(\lambda) = \begin{cases} f^0_{s,t} + \lambda & \text{if } \lambda < f^{\infty}_{s,t} - f^0_{s,t} \\ f^{\infty}_{s,t} & \text{otherwise} \end{cases} \tag{2}$$

or more simply:

$$f_{s,t}(\lambda) = \min(f^0_{s,t} + \lambda, f^{\infty}_{s,t}). \tag{3}$$

Note that, in the case of a network for which the investigated edge is a cut-edge, then, the previous formula is also valid, since when the investigated edge is removed, the maximum flow between two vertices, one in each side of the cut-edge, is null due to disconnectivity, and the critical capacity is simply f^{∞}.

3.2 Critical capacities for the all pairs maximum flows

Theorem 1. *Let $G = (V, E)$ be a network with n nodes. If only one edge capacity is allowed to vary, then the set of all critical capacities can be computed using two GH cut trees followed by $O(n^2)$ operations.*

Proof. Two remarks shall be made. First, for a given single pair of vertices s and t of the network, Lemma 1 provides a simple way to compute the unique critical capacity if we know the values of the maximum flows $f^0_{s,t}$ and $f^\infty_{s,t}$. Second, the GH procedure provides an efficient way to compute the all pairs maximum flows.

Thus, the desired result can be obtained by the computation of a GH cut tree in the absence of the investigated edge e in order to obtain all the $f^0_{s,t}$, $\forall s, t \in V(G)$, and the computation of a second GH cut tree where the capacity of the edge e is set to ∞. This latter computation provides all the $f^\infty_{s,t}$ $\forall s, t \in V(G)$. With these two values of maximum flows at hand, using Lemma 1, one get all critical capacities by computing and sorting increasingly the differences

$$f^\infty_{s,t} - f^0_{s,t}, \quad \forall s, t \in V(G).$$

The final step considers $n(n-1)$ pairs of vertices, thus it takes $O(n^2)$ operations to be performed. □

The algorithm can be directly obtained from this proof. Notice that it does not matter the case (increasing or decreasing) we deal with. Once the algorithm is performed, the critical values for both type of cases can be obtained.

4 Analyzing the effects of several parametric capacities

In this section, we study the case where more than one capacity vary either or not independently.

4.1 Analyzing the effects of two parametric capacities

We examine in detail the case where the capacities of two edges vary independently. The main result of this section is that the algorithm given by the former Theorem 1 can also be applied in the current case, and only four (actually $2^{\mathbf{2}}$) maximum flow computations are needed to compute any maximum flow value, whatever the value of the capacities of the selected edges are.

The general problem in this section can be formally stated as follows. Given a network $G = (V, E)$ and two distinct edges e_1 and e_2, we want to determine the maximum flow between all pairs of vertices when the capacity of e_1 is λ and the capacity of e_2 is μ, where $\lambda, \mu \geq 0$ are varying.

On the effects of two parametric capacities on a single maximum flow. In this paragraph, we consider two selected vertices s and t in the network and provide a way to compute $f_{s,t}(\lambda, \mu)$. Before stating our results, one can remark that all the partial functions $\lambda \mapsto f_{s,t}(\lambda, \mu_0)$, with μ_0 fixed, have the same profile as illustrated in Figure 2: the maximum flow first increases up to a saturation step (critical capacity), and then stagnates. The partial functions $\mu \mapsto f_{s,t}(\lambda_0, \mu)$, with λ_0 fixed, behave analogously.

As previously, we denote $f_{s,t}^{0,0}$ the maximum flow between s and t when both edges e_1 and e_2 are removed (respective capacities set to 0) from the network. The flows $f_{s,t}^{0,\infty}$, $f_{s,t}^{\infty,0}$ and $f_{s,t}^{\infty,\infty}$ can be defined in a similar way considering the capacities $c(e_1)$ and $c(e_2)$ set to 0 or to ∞. In other words we denote the maximum flows $f_{s,t}^{0,0}$, $f_{s,t}^{0,\infty}$, $f_{s,t}^{\infty,0}$ and $f_{s,t}^{\infty,\infty}$ as *extreme flows*.

Theorem 2. *Let $G = (V, E)$ be a network, e_1 and e_2 two different edges of E, and s and t two distinct vertices of V. Then, the maximum flow $(f_{s,t}(\lambda, \mu))$ between s and t with the capacity of e_1 set to λ and the capacity of e_2 set to μ can be directly obtained from the four maximum flows $f_{s,t}^{0,0}$, $f_{s,t}^{0,\infty}$, $f_{s,t}^{\infty,0}$ and $f_{s,t}^{\infty,\infty}$. The maximum flow $(f_{s,t}(\lambda, \mu))$ can be computed as follows:*

$$f_{s,t}(\lambda, \mu) = \min(f_{s,t}^{0,0} + \mu + \lambda, f_{s,t}^{0,\infty} + \lambda, f_{s,t}^{\infty,0} + \mu, f_{s,t}^{\infty,\infty}). \tag{4}$$

Proof. The main point of this proof is to decompose the computation of the general maximum flow into several computations of simple maximum flows and use Corollary 2 to obtain the desired values.

Thus, let us consider that μ is fixed. As noted previously, the partial function $\lambda \mapsto f_{s,t}(\lambda, \mu)$ can be obtained if both maximum flows $f_{s,t}(0, \mu)$ and $f_{s,t}(\infty, \mu)$ are known by using Corollary 2 or its closed form given in Equation 3:

$$f_{s,t}(\lambda, \mu) = \min(f_{s,t}(0, \mu) + \lambda, f_{s,t}(\infty, \mu)) \tag{5}$$

At this step, it remains to compute $f_{s,t}(0, \mu)$ and $f_{s,t}(\infty, \mu)$. For the former, we consider the partial function $\mu \mapsto f_{s,t}(0, \mu)$. Again, this function can be described using Corollary 2:

$$f_{s,t}(0, \mu) = \min(f_{s,t}(0, 0) + \mu, f_{s,t}(0, \infty)) \tag{6}$$

Using their definitions, Equation 6 can be rewritten as:

$$f_{s,t}(0, \mu) = \min(f_{s,t}^{0,0} + \mu, f_{s,t}^{0,\infty}) \tag{7}$$

Similarly, $f_{s,t}(\infty, \mu_0)$ can be obtained by the following equation:

$$f_{s,t}(\infty, \mu) = \min(f_{s,t}^{\infty,0} + \mu, f_{s,t}^{\infty,\infty}) \tag{8}$$

Consequently, we have:

$$f_{s,t}(\lambda, \mu) = \min(\min(f_{s,t}^{0,0} + \mu, f_{s,t}^{0,\infty}) + \lambda, \min(f_{s,t}^{\infty,0} + \mu, f_{s,t}^{\infty,\infty})) \tag{9}$$

Since $a+\min(b, c) = \min(a+b, a+c)$ and $\min(\min(a, b), \min(c, d)) = \min(a, b, c, d)$, the previous equation simplifies to:

$$f_{s,t}(\lambda, \mu) = \min((f_{s,t}^{0,0} + \mu + \lambda, f_{s,t}^{0,\infty} + \lambda, f_{s,t}^{\infty,0} + \mu, f_{s,t}^{\infty,\infty}). \tag{10}$$

$\square$

As previously, the proof yields an algorithm to compute the resulting maximum flows.

Effects on the all pairs maximum flows. From the property given by Theorem 2, we can deduce the all pairs maximum flows theorem for two parametric capacities, given below.

Theorem 3. *Let $G = (V, E)$ be a network, and e_1 and e_2 be two different edges of E. Then, the all pairs parametric maximum flow problem can be solved using the computation of four GH cut trees if the capacity of both edges e_1 and e_2 vary.*

Proof. This is a direct consequence of Theorem 2. We need to compute all the $f_{s,t}^{0,0}$, $f_{s,t}^{0,\infty}$, $f_{s,t}^{\infty,0}$ and $f_{s,t}^{\infty,\infty}$ for all the pairs (s, t) of vertices. The set of $f_{s,t}^{0,0}$, for all the vertices s and t can be obtained by the computation of a GH cut tree considering the network G in which have been removed both edges e_1 and e_2. The three other GH cut trees can be obtained similarly considering the presence or the removal of each tested edge with the infinite capacity.

Once the four GH cut-trees are computed, all values of maximum flows can be obtained using Theorem 2. The four GH cut trees provide the four extremal maximum flows and the desired maximum flow can be obtained using Equation 4. □

4.2 On the generalization to several parametric capacities

In this section, we consider the case where the capacities of more than two edges vary. Let e_1, e_2, $\ldots$, e_k be the selected edge capacities that will be parametrized by λ_1, λ_2, $\ldots$ and λ_k, $k \leq m$, where m is the number of edges in the network.

Theorem 4. *Let $G = (V, E)$ be a network, k be an integer, and e_1, e_2, $\ldots$, e_k be k different edges. All pairs maximum flows for all values of the parameters can be computed by using 2^k GH cut tree computations if the capacities of the edges vary independently.*

Proof. This result can be obtained by a simple recurrence. The basic case $k = 1$ is solved in Section 3.2. The sketch of the general case strictly follows the proof of Theorem 3. The main idea is to consider as fixed one of the parameters and use the recursion hypothesis for this case, leading to 2^{k-1} maximum flow computations. Then, it remains to develop each maximum flow computation in terms of the final dimension. Thus, for each computation, two maximum flows are necessary by using Corollary 2, leading to 2^k maximum flow computations.

The graphs on which the GH cut trees have to be computed are the variations of the initial graphs where the considered edges are either removed or their capacity set to infinity. □

5 Conclusion

In this paper, we have shown how to obtain efficiently allways the all pairs maximum flows when the capacities of some edges are parameterized. Previously, to solve the single parametric capacity case, as many GH cut tree computations were needed as the number of critical capacities, whereas in our approach, only two GH cut trees are required. Our work provides on the one hand a major improvement for the single varying capacity case and on the other hand, provides an efficient algorithm to solve the generalized varying capacities case.

An algorithmic improvement on our paper would be to provide a more efficient way to reuse a GH cut tree to compute a next one. We know that $(n-1)$ maximum flow computations are required to compute one GH cut tree. However, since there exists an important relationship between all the graphs for which we compute the GH cut trees, there should be a way to compute less than $2^k(n-1)$ maximum flows overall. We can expect a significant improvement on computation time in this way.

References

1. C. Berge. *Graphs and Hypergraphs.* North Holland, Amsterdam, 1973.
2. P. Berthomé, M. Diallo, and A. Ferreira. Generalized parametric multi-terminal flow problem. Technical Report LRI-1355, LRI, March 2003. available at http://www.lri.fr/˜ berthome/.
3. C.S. Chekuri, A.V. Goldberg, D.R. Karger, M.S. Levine, and C. Stein. Experimental study of minimum cut algorithms. In *ACM Symposium On Discrete Algorithms*, pages 324–333, New Orleans, Louisiana, 5–7 January 1997.
4. J. Cohen and E.P. Duarte Jr. Fault-tolerant routing of TCP/IP PDU's on general topology backbones. In *Design of Reliable Communication Networks*, Budapest, Hungary, October 2001.
5. M. Diallo. Parametric multi-terminal networks flow analysis. Master's thesis, Dpt of Mathematics, University of Kaiserslautern, Germany, October 2000.
6. M. Diallo and H.W. Hamacher. Parametric capacity and multi-terminal network flows. Technical Report 60, AG-Hamacher, University of Kaiserslautern, Germany, 2002.
7. S.E. Elmaghraby. On the relationship between the cut-tree and the fundamental cut-set of multi-terminal flow networks. *Journal of Franklin Institute*, October 1964.
8. S.E. Elmaghraby. Sensitivity analysis of multi-terminal network flows. *J. ORSA*, 12:680–688, 1964.
9. L.R. Ford and D.R. Fulkerson. *Flows in Networks.* Princeton Univ. Press, Princeton, NJ, 1973.
10. R. E. Gomory and T. C. Hu. Multi-terminal network flows. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):551–570, December 1961.
11. D. Gusfield. Very simple methods for all pairs network flow analysis. *SIAM Journal of Computing*, 19:143–155, 1990.
12. T. C. Hu. *Combinatorial Algorithms.* Addison-Wesley Publ. Comp., Reading, Mass., 1982.