RAPPORT DE RECHERCHE

LRI

GENERALIZED PARAMETRIC MULTI-TERMINAL FLOW PROBLEM

BERTHOME P / DIALLO M / FERREIRA A

Unité Mixte de Recherche 8623
CNRS-Université Paris Sud-LRI

03/2003

Rapport de Recherche N° 1355

CNRS – Université de Paris Sud
Centre d'Orsay
LABORATOIRE DE RECHERCHE EN INFORMATIQUE
Bâtiment 650
91405 ORSAY Cedex (France)

Generalized Parametric Multi-Terminal Flow Problem

Pascal Berthomé
Laboratoire de Recherche en Informatique (LRI)
UMR 8623, CNRS, Université Paris-Sud
91405, Orsay-Cedex, France
Pascal.Berthome@lri.fr

Madiagne Diallo
Laboratoire PRiSM, UMR 8140
Université de Versailles
45 Av. des Etats-Unis
78035 Versailles-Cedex, France
Madiagne.Diallo@prism.uvsq.fr

Afonso Ferreira*
CNRS, Projet MASCOTTE
I3S & Inria Sophia Antipolis
2004, Rt des Lucioles, 06902 Sophia Antipolis
afonso.ferreira@sophia.inria.fr

18th March 2003

Abstract

Given an undirected and connected network in which some edges have a parametric capacity varying independently or not, we study the multi-terminal flow problem, i.e., the all pairs maximum flows analysis problem for any value of the parametric capacities.

When all the capacities in the network are constant, it is known that the multi-terminal flow problem can be solved with the computation of one Gomory-Hu cut tree. In the presence of k parametric capacities varying in the network, we show that 2^k Gomory and Hu cut tree computations are enough to solve the multi-terminal flow problem. With these results, we improve all results until now presented in the single parametric case and extend them to a generalized setting.

Keywords: Multi-Terminal Flows, Cut Tree, Parametric Flows, Max-Flow, Min-Cut, Sensitivity Analysis, Network Management

Résumé

Notre étude se situe dans un réseau connexe pour lequel la capacité de certaines arêtes peut évoluer de manière indépendante ou non. Dans ce cadre, nous considérons le problème du flot multiterminal paramétré, c'est-à-dire, le problème de l'analyse des flots maximum entre toutes les paires de sommets pour n'importe quelle affectation des capacités paramétriques.

Quand toutes les capacités dans le réseau sont données, le problème du flot multi-terminal peut être résolu par le calcul d'un arbre de Gomory-Hu. S'il existe k capacité paramétriques, nous montrons que 2^k arbres de Gomory-Hu sont suffisants pour résoudre le problème de multiflots paramétré. Ce faisant, nous améliorons tous les résultats déjà connus pour une seule capacité paramétrique et étendons ce résultat pour plusieurs capacités.

Mots-Clés: Flots Multi-terminaux, Arbre de coupe, Flots paramétrés, Flot Maximum, Coupe Minimale, Analyse de sensibilité, Gestion de réseau.

*Partially supported by the European projects RTN ARACNE and FET CRESCCO, and by the AS CNRS Dynamo.

1 Introduction

In the late 1950's, the single source–single terminal maximum flow problem was popularized by the resolution of Ford and Fulkerson [7]. They specially showed the connection between the maximum flow and the min cut problems.

In the setting of an undirected and connected edge-weighted graph, the multi-terminal flow 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 [8] delivered an ingenious method to solve this maximum flow analysis problem using only $n-1$ maximum flow computations. Gomory and Hu summarized their 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 [9] provided a simpler procedure to obtain the GH cut tree using also $n-1$ maximum flow computations.

In 1964, Elmaghraby [6] was the first to extend the former multi-terminal flow problem to the case where a single capacity is varying. He analyzed the effect of the capacity variation on the all pairs maximum flows in the network concluding that *a priori* some maximum flows will be affected by the variation and others not, but, as far as the variation is concerned, some maximum flows may change their behavior. To provide the values of the parametric capacity (referred to as critical capacity) for which some flow behavior change may occur, Elmaghraby computed as many GH cut trees as the number of possible critical capacities. Recently, Diallo and Hamacher [4] provided a more efficient way to perform the Elmaghraby analysis.

The multi-terminal flow problem with constant capacity has many known applications in the fields of transports, energy and telecommunications (see for example [6][3] and references therein). The parametric multi-terminal flow problem also reflects in these fields problems including link breakdown, capacity improvement, bandwidth reservation, network expansion.

In this paper, we present the generalized parametric multi-terminal flow problem. This extends the literature to the case where more than one capacity vary in the network. We provide algorithms that give the all pairs maximum flows in the network for any instance of the varying capacities. The main result of this paper is that if the capacity of k edges vary independently, 2^k GH cut trees are enough to complete the analysis. This widely improves and extends the previous studies. Especially, the critical capacities become not a central key but a simple consequence of our result 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.

In the remainder, we present in Sec. 2 some basic definitions and briefly describe the main ideas of Elmaghraby's method, and of Diallo and Hamacher's method. Sec. 3 is devoted to the details of our method in the case of a single edge-capacity variation. In Sec. 4, we provide the generalization to several parametric edge-capacities. We close the paper with a conclusion and some perspectives. An appendix gives some examples that illustrate our purpose.

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 [7][10][1].

2.1 Definitions

Let G be an undirected and connected graph. We denote by $V(G)$ the vertex-set of G and by $E(G)$ its edge-set. In our setting, a *network* is intended to be G associated to an edge-weight function $c : E(G) \to R^+$ called *edge-capacity* function. A maximum flow from a source vertex s to a terminal vertex t in G is given by a function $f : E(G) \to R^+$. The flow has to be conserved in each vertex, excepted in vertices s and t, *i.e*, $\forall v \neq s, t$, $\forall u \in N_G(v)$, $\sum f(u, v) - \sum f(v, u) = 0$. Moreover, for each $[u, v] \in E(G)$, $f(u, v) \leq c(u, v), \forall (u, v) \in E$.

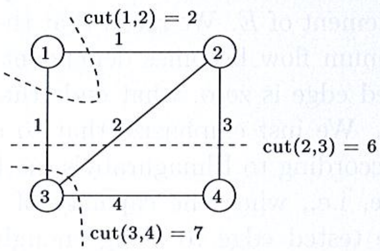
For simplicity, we denote the maximum flow from a source s to a terminal t as f_{st}.

Definition 1 GH cut tree *Given a network G, a GH cut tree T obtained from G is a weighted tree of G; a tree with the same set of vertices $V(G)$ connected by $n - 1$ weighted edges. We denote the set of the edges of the GH cut tree T by $E(T)$.*

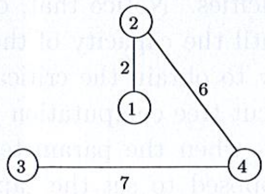
In the GH cut tree:

- *each edge of $E(T)$ represents a minimum cut in the original network G between its end vertices;*
- *each edge of $E(T)$ is labeled by the value of the minimum cut that it represents;*
- *a minimum cut in G between any two vertices x and y is represented in GH by an edge in $E(T)$ labeled with a minimum weight on the path from x to y;*
- *the maximum flow value in G between x and y is given in the GH by the minimum edge weight on the path from x to y.*

In Fig. 1(b), we illustrate a GH cut tree T obtained from the given network G with the illustrated minimum cuts in Fig. 1(a). As shown in [8], $n - 1$ 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 [8] and in [9]. An experimental study of minimum cut algorithms and a comparison of algorithms producing GH cut trees is provided in [2].



(a) G



(b) GH tree

Figure 1: A graph and one of its Gomory-Hu cut tree

2.2 Sensitivity analysis and critical capacities

A natural extension of the multi-terminal flow problem with constant edge-capacities is to parameterize exactly one of the capacities and analyze its effect on the all pairs maximum flows. This was effectively the aim of Elmaghraby in [6]. He mainly described a procedure to analyze the effect of a capacity decreasing linearly and gave a sketch of the procedure to solve the capacity increasing case.

The main problem solved in [6] can be stated as follows. Given the above network description, and an edge $e = (i, j)$ of $E(G)$ with a capacity given by $c(e) = \bar{c} - \epsilon$, $0 \leq \epsilon \leq \bar{c}$, it is to determine the set of all pairs maximum flows at each instance of ϵ.

With the decrease on the capacity $c(e)$ of 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 before constant may begin to decrease linearly with respect to ϵ. The value of $c(e)$ for which a flow changes of behavior is called *critical capacity*. Precisely, in this setting a *critical capacity* is the value of the parametric capacity for which a previously constant flow begins to decrease with ϵ. 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 problem to the one of finding the critical capacities since in each such interval a GH cut tree computation provides the desired maximum flows.

Next, we turn to briefly describe the general idea of the Elmaghraby's procedure and its improvement by Diallo and Hamacher. In an earlier work, Elmaghraby proved Theorem 1 below that was used as kernel of his sensitivity analysis.

Theorem 1 ([5]) *Let $G = (V, E)$ be a network, T a GH cut tree of G, and $e = (i, j)$ an edge of G. Then, edge e only belongs to minimum cuts in G that separate its end vertices i and j in the cut tree T.*

In the first step, a GH cut tree is computed with the parameter ϵ set to zero. Based on Theorem 1, an $E(G) \times (n-1)$ matrix is constructed which defines some minimum cuts $\{r_k\}$ in terms of the original edges e in G with entries $+1$ if edge $e \in r_k$ and 0 otherwise. With the matrix at hand, in the cut tree, the value of the minimum cuts containing the edge e are updated by adding $-\epsilon$ to their initial values, and thus are identified the set of parametric minimum cut values E, and its complement set $\bar{E}$ in the cut tree.

From this minimum cut classification, one can determine the values of the parametric capacity for which some flow previously in $\bar{E}$ becomes an element of E. We recall that these values represent the critical capacities. Notice that, once a maximum flow becomes dependent on the parameter, it remains so until the capacity of the investigated edge is zero, what ends the analysis. In [6], it is described how to obtain the critical capacities. We just emphasize that to obtain each critical capacity, a GH cut tree computation is needed, according to Elmaghraby's method.

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.

In [4], Diallo and Hamacher showed that the $E(G) \times (n-1)$ matrix used by Elmaghraby 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 test whether or not a minimum cut between a given pair of vertices contains the investigated edge. Their algorithm works as follows. Given two vertices s and t, and the investigated edge $e = (i, j)$, a minimum cut between s and t is computed. If it does not contain the edge, the capacity $c(e)$ of e is decreased by one and a minimum cut between s and t is recomputed. If the value of the new minimum cut is the older value decreased by one then there is a minimum cut between these vertices containing the edge e; otherwise there is no minimum cut between s and t that contains e.

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. Thus, we solve with two GH cut tree computations what would need the double complexity of performing a single Elmaghraby sensitivity analysis.

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 Sec. 3.2, we explore this result to obtain the critical capacities for the all pairs maximum flows. A numerical example is developed in Sec. 3.3.

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 $f^{\infty}_{s,t}$ (or simply f^{∞}) $\lim_{\lambda \to \infty} f_{s,t}(\lambda)$, i.e., the maximum flow when there is no constraint on the edge e. This latter value is finite for all pairs, except when $(s,t) = (i,j)$. It can be simply computed by setting the capacity of the tested edge to the sum of the capacities of the adjacent edges to e.

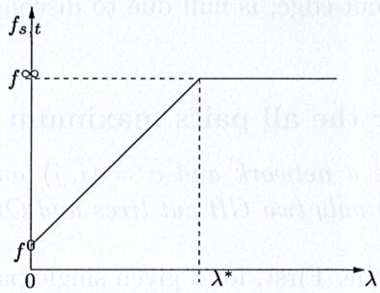


Figure 2: behavior of the maximum flow function $f_{s,t}(\lambda)$

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 parametrization. As shown in Fig. 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;
- at some value λ^* of λ, namely, critical capacity, the maximum flow becomes saturated (except if $\{s,t\} = \{i,j\}$).

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 $f^{\lambda}_{i,i} = \infty$.

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

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

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 2 *Let $G = (V, E)$ be a network and $e = (i, j)$ an edge of G. The set of all critical capacities can be computed using only two GH cut trees and $O(n^2)$ operations, where $n = |V|$.*

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

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

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

The algorithm can directly be obtained from this proof.

3.3 Numerical example: a single parametric capacity

In this section, we provide an example in order to compare the original methods to ours.

Given the network of Fig. 3, let us consider the investigated edge to be $e = (2, 5)$ with a parametric capacity $c(e) = 13 \pm \epsilon$ (the $\pm$ is to avoid two figures, one for each of the decreasing and increasing cases).

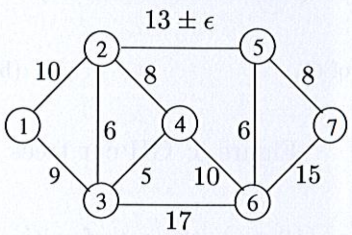


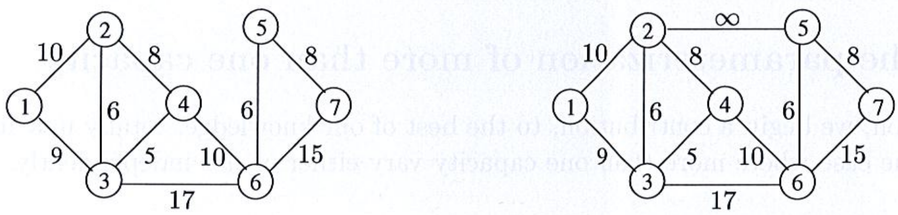
Figure 3: Initial Network with investigated edge $e = (2, 5)$

With these hypotheses, the Elmaghraby method and all the others based on it, would need to compute *eight GH cut trees* in order to determine the *five critical capacities* in the case $13 - \epsilon$ and *three* others for the case $13 + \epsilon$. In the following we illustrate that our procedure needs just *two GH cut trees and* $O(n^2)$ *to solve* simultaneously the increasing and decreasing cases. Let us make the following variable change:

$$\lambda = 13 \pm \epsilon, \quad where\ \lambda \geq 0.$$

Thus we only study a capacity increasing case and deliver with two GH cut trees all critical capacities for both $c(e) = 13 - \epsilon$ and $c(e) = 13 + \epsilon$ cases.

In Fig.s 4(a) and 4(b), we respectively show the network in the absence of the investigated edge $e = (2, 5)$, i.e., $\lambda = 0$ and when the capacity of the investigated edge is set to ∞, i.e., $\lambda = \infty$.



(a) Graph G_0 (b) Graph G_∞

Figure 4: Extremal networks for the previous case.

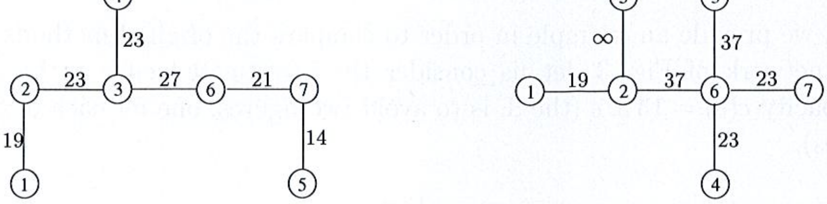
In Fig. 5, we deliver the only two needed GH cut trees respectively from the networks of Fig. 4(a) and Fig. 4(b).

As illustrated in the following tables, all the critical capacities are obtained with

$$\lambda^*_{s,t} = f^{\infty}_{s,t} - f^{0}_{s,t}.$$

For the decreasing case $(13 - \epsilon)$ the set of critical capacities is:

$$\{\lambda^*_{s,t}; \lambda^*_{s,t} \leq 13\} = \{0, 2, 5, 9, 10\}.$$



(a) One cut tree of G_0 (b) One cut tree of G_∞

Figure 5: GH cut trees

As regard to the increasing case $(13+\epsilon)$, the set of critical capacities is

$$\{\lambda^*_{s,t}; \lambda^*_{s,t} \geq 13\} = \{14, 23, \infty\}.$$

s	t	$f^0_{s,t}$	$f^\infty_{s,t}$	$\lambda^*_{s,t}$
1	2	19	19	0
1	3	19	19	0
1	4	19	19	0
1	5	14	19	5
1	6	19	19	0
1	7	19	19	0
2	3	23	37	14

s	t	$f^0_{s,t}$	$f^\infty_{s,t}$	$\lambda^*_{s,t}$
2	4	23	23	0
2	5	14	∞	∞
2	6	23	37	14
2	7	21	23	2
3	4	23	23	0
3	5	14	37	13
3	6	27	37	10

s	t	$f^0_{s,t}$	$f^\infty_{s,t}$	$\lambda^*_{s,t}$
3	7	21	23	2
4	5	14	23	9
4	6	23	23	0
4	7	21	23	2
5	6	14	37	23
5	7	14	23	9
6	7	21	23	2

If the initial capacity was just $\epsilon \geq 0$, then taking $\lambda = \epsilon$ would give as critical capacities the elements of the set $\{0, 2, 5, 9, 10, 13, 14, 23, \infty\}$.

4 On the parameterization of more than one capacity

In this section, we begin a contribution, to the best of our knowledge, totally new in the literature. We study the case where more than one capacity vary either or not independently.

4.1 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 only 2^2 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 are concerned to determine the maximum flow between any pair of vertices when the capacity of e_1 is λ and the capacity of e_2 is μ, where $\lambda, \mu \geq 0$.

4.1.1 On the effect of two parametric capacities on a single maximum flow.

In this section, 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 $\lambda \mapsto f_{s,t}(\lambda, \mu_0)$ and $\mu \mapsto f_{s,t}(\lambda_0, \mu)$ (with λ_0 and μ_0 fixed) have the same profile as illustrated in Fig. 2: the maximum flow, first, increases up to a saturation step (critical capacity), then, stagnates from there on.

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 ∞. 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 3 *Let $G = (V, E)$ be a network, e_1 and e_2 two different edges of E, and s and t two distinct vertices. 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 first one, 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, this latter equation 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:

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

□

As previously, the proof outlines the algorithm to compute the resulting maximum flows. Fig. 6 shows the behavior of the maximum flows for a specific assignment of the extreme flows.

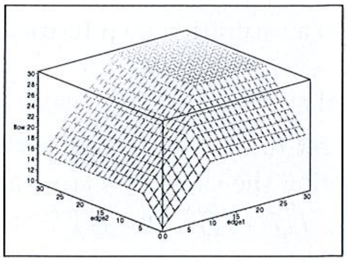


Figure 6: Resulting maximum flows $f_{s,t}$ for two parametric capacities. The four extreme flows are: $f_{s,t}^{0,0} = 10$, $f_{s,t}^{0,\infty} = 15$, $f_{s,t}^{\infty,0} = 20$ and $f_{s,t}^{\infty,\infty} = 30$.

4.1.2 Effect on the all pairs maximum flows.

From the property given by Theorem 3, we can deduce the all pairs maximum flows theorem for two parametric capacities.

Theorem 4 *Let $G = (V, E)$ be a network, and e_1 and e_2 two different edges of E. Then, the all pairs parametric maximum flow problem can be solved using the computation of 2^2 GH cut trees if the capacity of both edges e_1 and e_2 vary.*

Proof: This is a direct consequence of Theorem 3. 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 existence or the removal of each tested edge with the infinite capacity.

When the four GH cut-trees are computed any value of maximum flow can be obtained using Theorem 3. □

4.1.3 Application: dependent capacities.

One main important application of this theorem is the computation of any maximum flow when the capacities of the tested edges are dependent. In this case, critical capacities can be stated as in the case when only one edge has a parametric capacity. For example, assuming that the capacity of both edges is set to λ, then two critical capacities may exist. First, when the capacity of the edges increases, both edges are influent in the remaining maximum flow, thus the slope of the function is 2. At the first critical point, one of the edges may lose its influence on the remaining flow, then the slope becomes only 1. Finally, for large capacities both edges are non-influent on the resulting maximum flow, equal to $f_{s,t}^{\infty,\infty}$. This latter break corresponds to the second critical capacity. As example, Fig. 7 shows the resulting maximum flow function in the case described in Fig. 6 when both edges have their capacities equal to λ. This result is formalized in the theorem below. In order to simplify, we note in this paragraph: $f_{s,t}(\lambda) = f_{s,t}(\lambda, \lambda)$.

Corollary 3 *Let $G = (V, E)$ be a network, e_1 and e_2 two different edges of E, and s and t two distinct vertices. Then, the maximum flow $(f_{s,t}(\lambda))$ between s and t when both capacities of e_1 and e_2 are set to λ can be directly obtained from the four maximum flow computations $f_{s,t}^{0,0}$, $f_{s,t}^{0,\infty}$, $f_{s,t}^{\infty,0}$ and $f_{s,t}^{\infty,\infty}$. The behavior of this function is:*

$$f_{s,t}(\lambda) = \min(f_{s,t}^{0,0} + 2\lambda, f_{s,t}^{0,\infty} + \lambda, f_{s,t}^{\infty,0} + \lambda, f_{s,t}^{\infty,\infty}) \tag{11}$$

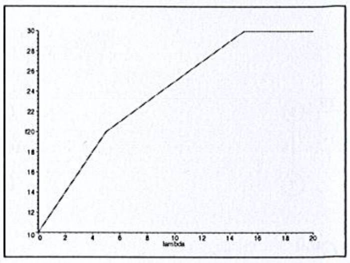


Figure 7: Resulting flow function when the capacities of both tested edges are equal to λ. The values of the extreme flows are given in Fig. 6

Proof: This is direct consequence of Theorem 3 by setting $\lambda = \mu$. $\square$

Corollary 4 *Let $G = (V, E)$ be a network, e_1 and e_2 two different edges of E, and s and t two distinct vertices. If the capacities of both edges e_1 and e_2 are λ, then there exists one or two breakpoints in the behavior of the maximum flow function between s and t.*

Proof: Assume that $f_{s,t}^{\infty,0} > f_{s,t}^{0,\infty}$. Then, the behavior of the function can be simplified into the minimum of 3 linear functions:

$$f_{s,t}(\lambda) = \min(f_{s,t}^{0,0} + 2\lambda, f_{s,t}^{0,\infty} + \lambda, f_{s,t}^{\infty,\infty}) \tag{12}$$

Consider now the three intersection points of these functions: $\lambda_1^* = f_{s,t}^{0,\infty} - f_{s,t}^{0,0}$; $\lambda_2^* = 1/2(f_{s,t}^{\infty,\infty} - f_{s,t}^{0,0})$ and $\lambda_3^* = f_{s,t}^{\infty,\infty} - f_{s,t}^{0,\infty}$.

Then, it is easy to verify that $\lambda_1^* < \lambda_3^*$ and $\lambda_2^* < \lambda_3^*$. If $\lambda_1^* < \lambda_2^*$ then, there exists two breakpoints: λ_1^* and λ_3^*; otherwise there exists only one breakpoint: at point λ_2^*. $\square$

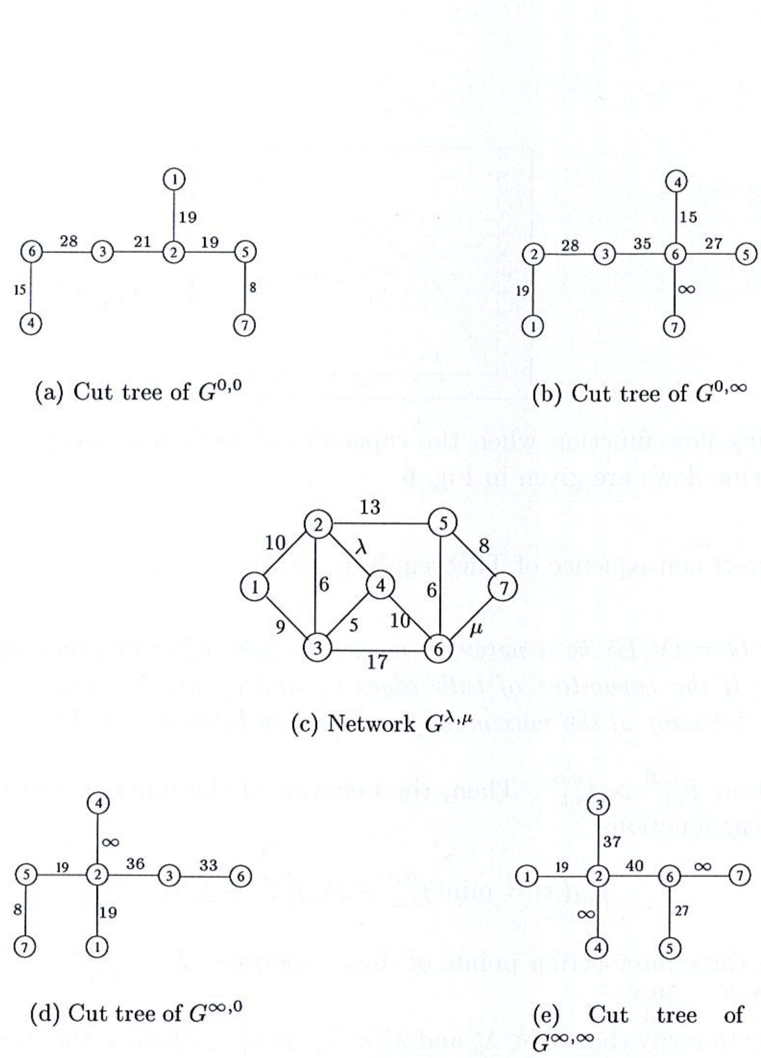
From this latter corollary, we can compute all the breakpoints, simply by knowing the extremum flows.

4.2 Numerical Example: two parameterized edge-capacities

In Fig. 8, we illustrate this analysis with a simple example where two capacities vary independently with the parameters λ and μ as shown in Fig. 8(c). We need to compute a GH cut tree for the four graphs obtained from the initial graph (called $G^{a,b}$, where a and b stand for either 0 or ∞ corresponding to the effective capacity on both tested edges); these GH cut trees are shown in Fig.s 8(a), 8(b), 8(d) and 8(e). For any value of λ and μ, and, any two vertices s and t, it is possible to obtain the exact value of the maximum flow between the two vertices using Equation 4. For instance, consider $\lambda = 5$, $\mu = 9$ and the pair $(3, 7)$, thus, the respective maximum flow value is $f_{3,7}(5, 9) = \min((f_{3,7}^{0,0} + 5 + 9, f_{3,7}^{0,\infty} + 5, f_{3,7}^{\infty,0} + 9, f_{3,7}^{\infty,\infty}) = \min(8 + 5 + 9, 35 + 5, 8 + 9, 37) = 17$.

Assume now that the capacities are dependent. More precisely, we have the further relation $\lambda = \mu$. We only consider here examples of pairs where there is two, one or no breakpoints in the behavior of the maximum flow function.

No breakpoint: consider for example the pair of nodes $(1, 6)$. We note that the flow between these two points remain unchanged in the whole parameterization process (by simply reading the minimum weight on the path between nodes 1 and 6 in the GH cut trees).



(a) Cut tree of $G^{0,0}$

(b) Cut tree of $G^{0,\infty}$

(c) Network $G^{\lambda,\mu}$

(d) Cut tree of $G^{\infty,0}$

(e) Cut tree of $G^{\infty,\infty}$

Figure 8: GH cut trees of a graph with two parametric capacities

One breakpoint: it suffices to have either $f_{s,t}^{0,\infty}$ or $f_{s,t}^{\infty,0}$ equal to $f_{s,t}^{0,0}$, and, $f_{s,t}^{\infty,\infty}$ different of $f_{s,t}^{0,0}$. This is the case of the pair $(5,7)$ for which the breakpoint value is 19.

Another interesting case is when the flow always increases with a slope 2. This occurs when the virtual breakpoints induced by $f_{s,t}^{\infty,0}$ and $f_{s,t}^{0,\infty}$ (that are respectively equal to $f_{s,t}^{\infty,0} - f_{s,t}^{0,0}$ and $f_{s,t}^{0,\infty} - f_{s,t}^{0,0}$) are smaller than the breakpoint induced by $f_{s,t}^{\infty,\infty}$ (equals to $1/2(f_{s,t}^{\infty,\infty} - f_{s,t}^{0,0})$. For instace, the pair of vertices $(3,6)$ illustrate this phenomenon where the only breakpoint is 4.5.

Two breakpoints: this is the case for some pairs in the network. For example, let us consider the pair $(2,3)$. In this case, we find that the first breakpoint is for $\lambda = 7$ and the second one is for $\lambda = 9$.

4.3 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, $\dots$, e_k be the selected edge capacities that will be parametrized by λ_1, λ_2, $\dots$ and λ_k.

Theorem 5 *Let $G = (V, E)$ be a network, k be an integer, and e_1, e_2, $\dots$, e_k be k different edges. The parametric all pairs maximum flow 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 Sec. 3.2. The sketch of the general case strictly follows the proof of Theorem 4. 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. If we want to compute the parametric all pairs maximum flows, we have to compute the maximum flows for all pairs. The GH cut trees delivers this.

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. □

Notice that we assumed the capacities of the selected edges to vary independently. When there exists some dependency between these capacities, the number of required GH cut tree computations does not decrease as shown in Corollary 3.

5 Conclusion

In this paper, we have shown how to compute efficiently the all pairs parametric maximum flow when the capacities of several edges vary. In the older methods, to solve the single parametric capacity case, as many GH cut tree computations as the number of critical capacities are needed, whereas in our approach, only two GH cut trees are required. Our work provides on the one hand a major improvement on all previous studies on the single parametric capacity case and on the other hand, provides an efficient algorithm to solve the several parametric capacities case. To the best of our knowledge, no work did tackle the multiple parametric capacities case.

This work has many applications when the parametric capacities are dependent. In this case, we have put in evidence the existence of critical capacities. The determination of these critical capacities is widely simplified with the present analysis.

An improvement on our paper would be to provide a more efficient way to compute all the GH cut trees needed to find the critical capacities. We know that $n-1$ maximum flow computations are required to compute one GH cut tree. However, since there exists an important connection between both graphs for which we compute the GH cut trees, there should be a way to compute less than $2(n-1)$ 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] C.S. Chekuri, A.V. Goldberg, D.R. Karger, M.S. Levine, and C. Stein. Experimental study of minimum cut algorithms. In *Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 324–333, New Orleans, Louisiana, 5–7 January 1997.

[3] M. Diallo. Parametric multi-terminal networks flow analysis. Master's thesis, Dpt of Mathematics, University of Kaiserslautern, Germany, October 2000.

[4] M. Diallo and H.W. Hamacher. Parametric capacity and multi-terminal network flows. Technical report, Laboratoire PR*i*SM, Université de Versailles, France, 2003.

[5] 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.

[6] S.E. Elmaghraby. Sensitivity analysis of multi-terminal network flows. *J. ORSA*, 12:680–688, 1964.

[7] L.R. Ford and D.R. Fulkerson. *Flows in Networks.* Princeton Univ. Press, Princeton, NJ, 1973.

[8] 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.

[9] D. Gusfield. Very simple methods for all pairs network flow analysis. *SIAM Journal of Computing*, 19:143–155, 1990.

[10] T. C. Hu. *Combinatorial Algorithms.* Addison-Wesley Publ. Comp., Reading, Mass., 1982.

RAPPORTS INTERNES AU LRI - ANNEE 2003

N°	Nom	Titre	Nbre de pages	Date parution
1345	FLANDRIN E LI H WEI B	A SUFFICIENT CONDITION FOR PANCYCLABILITY OF GRAPHS	16 PAGES	01/2003
1346	BARTH D BERTHOME P LAFOREST C VIAL S	SOME EULERIAN PARAMETERS ABOUT PERFORMANCES OF A CONVERGENCE ROUTING IN A 2D-MESH NETWORK	30 PAGES	01/2003
1347	FLANDRIN E LI H MARCZYK A WOZNIAK M	A CHVATAL-ERDOS TYPE CONDITION FOR PANCYCLABILITY	12 PAGES	01/2003
1348	AMAR D FLANDRIN E GANCARZEWICZ G WOJDA A P	BIPARTITE GRAPHS WITH EVERY MATCHING IN A CYCLE	26 PAGES	01/2003
1349	FRAIGNIAUD P GAURON P	THE CONTENT-ADDRESSABLE NETWORK D2B	26 PAGES	01/2003
1350	FAIK T SACLE J F	SOME b-CONTINUOUS CLASSES OF GRAPH	14 PAGES	01/2003
1351	FAVARON O HENNING M A	TOTAL DOMINATION IN CLAW-FREE GRAPHS WITH MINIMUM DEGREE TWO	14 PAGES	01/2003
1352	HU Z LI H	WEAK CYCLE PARTITION INVOLVING DEGREE SUM CONDITIONS	14 PAGES	02/2003
1353	JOHNEN C TIXEUIL S	ROUTE PRESERVING STABILIZATION	28 PAGES	03/2003