Discrete Applied Mathematics 1 (1979) 287–300

THE MAX-FLOW PROBLEM WITH PARAMETRIC CAPACITIES

Janet SOMERS*

Bell Laboratories, Holmdel, NJ 07733, U.S.A.

Ilan ADLER

Department of Industrial Engineering and Operations Research, University of California, Berkeley, CA 94720, U.S.A.

Received 19 December 1978
Revised 14 August 1979

In this paper we present a comprehensive analysis of the max-flow problem with n parametric capacities, and give the basis for an algorithm to solve it. In particular we give a method for finding the max-flow value as a function of the parameters, and max-flows for all parameter points, in terms of max-flow values to problems at certain key parameter points. In the problem with nonzero lower bounds on the arc flows, we derive a set of linear constraints whose solution set is identical to the set of all feasible parameter points.

The intrinsic difficulty of the problem is compared with that of the general multiparametric linear programming problem, and thus light is shed on the difficulty of the latter problem, whose complexity is currently unknown.

1. Introduction

In this paper we present a comprehensive analysis of the structure of the max-flow problem with one or more parametric capacities, and give the basis for an algorithm for its solution. Let $G=\{N, \mathcal{A}\}$ be a directed graph (where N is the nodes set and $\mathcal{A}$ the arcs set). Corresponding to each arc (i, j) in $\mathcal{A}$ is a pair of real numbers (possibly $-\infty, \infty$) $l(i, j)$, $c(i, j)$ which we call respectively *lower* and *upper capacities*. We define a *circulation* in G to be a real-valued function f on $\mathcal{A}$ satisfying

(i) $\quad l(i, j) \leq f(i, j) \leq c(i, j) \quad \forall (i, j) \in \mathcal{A}$,

(ii) $\quad f(i, N) - f(N, i) = 0 \quad \forall i \in N$,

where

$$f(i, N) = \sum_{(i, j) \in \mathcal{A}} f(i, j), \; f(N, i) = \sum_{(k, i) \in \mathcal{A}} f(k, i).$$

We may distinguish two nodes s and t as a *source* and *sink*. Then an s-t *flow* is a real-valued function f on $\mathcal{A}$ satisfying (i) and (ii) $\forall i \neq s, t$. The *value* of f is

* Work done while at University of California at Berkeley Mathematics Department and Tel Aviv University Statistics Department.

defined as $f(N, t)-f(t, N)$ $(=f(s, N)-f(N, s))$; we denote it $V(f)$. In the *max-flow problem* we seek an s-t flow f of maximum value. Note that every s-t flow f in $G=\{N, \mathcal{A}\}$ defines a circulation in $G'=\{N, \mathcal{A}'\}$, $\mathcal{A}'=\mathcal{A}\cup\{(t, s)\}$, with $V(f)=f(t, s)$, and vice-versa. Thus the max-flow problem is equivalent to the problem of finding a circulation in G' with maximum $f(t, s)$.

In this study we focus specifically on parametric upper capacities, and indicate how the results can be modified to apply to lower capacities. We define the problem as follows.

Let $\mathcal{A}_1=\{A^1,\ldots, A^n\}$ be some subset of arcs. $\lambda=(\lambda_1,\ldots,\lambda_n)$ will denote a vector of parameters. Let $\mathcal{A}_2=\mathcal{A}'\setminus\mathcal{A}_1$ be the nonparametrized arcs. The problem can then be written as follows:

$$P(\lambda)\colon \max f(t, s),$$

subject to

$$f(i, N)-f(N, i)=0 \quad \forall i\in N,$$
$$l(A^k)\leqslant f(A^k)\leqslant\lambda_k, \quad k=1,\ldots,n,$$
$$l(i, j)\leqslant f(i, j)\leqslant c(i, j) \quad \forall(i, j)\in\mathcal{A}_2.$$

We denote by $V(\lambda)$ the optimal solution value in $P(\lambda)$. We make the assumptions that $l(i, j)\leqslant c(i, j)$ $\forall(i, j)\in\mathcal{A}_2$ and that $P(\lambda)$ is bounded for all possible values of λ. Our objective is to determine for which λ the problem $(P\lambda)$ is feasible, and to construct a function enabling us to explicitly calculate for every feasible λ the max-flow value and a max-flow itself.

We make use of the (generalized) *max-flow min-cut theorem*, which says that in any network with fixed capacities the max-flow value is equal to the minimum value among all s-t cuts. An *s-t cut* is a set of arcs

$$(X, \bar{X})=\{(i, j)\mid i\in X, j\in\bar{X}\}$$

where X, $\bar{X}$ is a partition of N such that $s\in X$, $t\in\bar{X}$; its *value* is defined to be $c(X, \bar{X})-l(\bar{X}, X)$ where

$$c(X, \bar{X})=\sum_{(i, j)\in(X, \bar{X})} c(i, j) \quad\text{and}\quad l(\bar{X}, X)=\sum_{(i, j)\in(\bar{X}, X)} l(i, j).$$

We also find as a result of our work a minimal cut (i.e., a cut of minimum value) for every value of λ such that $P(\lambda)$ is feasible.

Note that if we wish to consider the problem in which arc (i, j) has fixed upper capacity $c(i, j)$ and parametric lower capacity λ, we may replace (i, j) by an arc (j, i) with fixed lower capacity $-c(i, j)$ and parametric upper capacity $-\lambda$ and obtain an equivalent problem. Thus our results may be applied to the case in which there also exist parametric lower capacities. We discuss how our feasibility results may be modified to solve the feasibility question directly for the case of parametric lower capacities only, without making this transformation. We do not concern ourselves with the case in which both upper and lower capacities on the same arc are parametrized.

We begin the analysis in Section 2 with the special case in which all lower capacities are equal to zero. In this relatively simple case $P(\lambda)$ is always feasible. In Section 3 we extend the results to the general case, first combining the results of Section 2 with known feasibility results in order to determine the set of all λ for which $P(\lambda)$ is feasible, and then modifying results of Section 2 to show how $P(\lambda)$ may be solved. Finally, since $P(\lambda)$ is actually a special case multidimensional parametric right-hand-side linear program, we discuss in Section 4 how our approach to this special case differs from general parametric linear program methodology (specifically, applied to our problem), and we point out how our results can be instrumental in evaluating the practical limitations of multi-dimensional parametric analysis.

We shall denote by Δ the set of all n-dimensional vectors δ with components equal to 0 or 1. Given $\delta \in \Delta$ we define $\eta(\Delta)$ by $\eta_k(\delta) = 0 \cdot \delta_k^{-1}$ (where $0/0 \equiv \infty$). (For example, $\eta(1, 0, 1) = (0, \infty, 0)$.)

2. The zero lower capacities case

In this section we analyze $P(\lambda)$ under the assumption that $l(i, j) = 0\ \forall (i, j) \in \mathscr{A}$. We consider $\lambda \geqslant 0$ only. Our first result concerns the structure of $V(\lambda)$. For $\delta \in \Delta$ we denote by $M(\delta)$ the minimum value of all s-t cuts $(X, \bar{X})$ in G for which $(i_k, j_k) \in (X, \bar{X})$ iff $\delta_k = 1$, where in evaluating the cut we set $\lambda_k = 0\ \forall k$ such that $\delta_k = 1$. (The inclusion of arcs from $\mathscr{A}_2$ is determined only by the minimum cut value.) We take $M(\delta) \equiv \infty$ if the set of s-t cuts containing exactly arcs (i_k, j_k) with $\delta_k = 1$ is empty.

Theorem 1.

$$V(\lambda) = \min\{M(\delta) + \sum_{k=1}^{n} \delta_k \lambda_k \mid \delta \in \Delta\} \quad (\lambda \leqslant 0).$$

Proof. Follows directly from the max-flow min-cut theorem. (See [1].) □

Note that the max-flow value $V(\lambda)$ is a continuous piecewise linear function of λ over the n-dimensional nonnegative ortant. Specifically, let

$$\Gamma(\delta) = \{\lambda \mid V(\lambda) = M(\delta) + \sum_{k=1}^{n} \delta_k \lambda_k\};$$

then $V(\lambda)$ is linear (with coefficients zero or one) over each of the nonempty sets $\Gamma(\delta)$. The union of the sets $\Gamma(\delta)$ over all $\delta \in \Delta$ covers the n-dimensional nonnegative ortant, and it is easy to see that each is a d-dimensional polyhedral

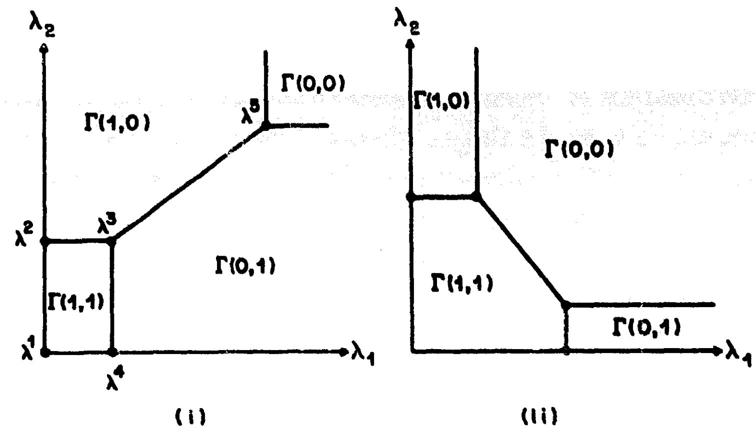

Fig. 1.

set for some $d \leqslant n$. Note that there are at most 2^n such sets (since this is the cardinality of Δ). Obviously for $n=1$, $\Gamma(0)=[0, M(1)-M(0)]$ and $\Gamma(1)=[M(1)-M(0), \infty)$.

Two typical cases for $n=2$ are shown in Fig. 1. (For a reference see [2].)

In order to obtain an explicit expression for $V(\lambda)$ it is necessary to compute all $M(\delta)$ which are relevant in the expression presented in Theorem 1. We next show how this can be accomplished by solving a sequence of max-flow problems in G, each with a special set of parameters λ. In the following we adopt the convention that $\infty \cdot 0 = 0$.

Theorem 2. *Given $\bar{\delta} \in \Delta$, let $(X, \bar{X})$ be a minimal cut in the network for problem $P(\eta(\bar{\delta}))$. (Thus its value is $V(\eta(\bar{\delta}))$.)*

(a) *If $A^k \in (X, \bar{X})$ $\forall k$ for which $\bar{\delta}_k = 1$, then $M(\bar{\delta}) = V(\eta(\bar{\delta}))$.*

(b) *If $\exists k$ such that $\bar{\delta}_k = 1$ and $A^k \notin (X, \bar{X})$, then*

$$M(\bar{\delta}) + \sum_{k=1}^{n} \bar{\delta}_k \lambda_k \geqslant \min \left\{ M(\delta) + \sum_{k=1}^{n} \delta_k \lambda_k \mid \delta \in \Delta \backslash \bar{\delta} \right\} \quad \forall \lambda \geqslant 0.$$

Proof. (a) Since $\bar{\delta}_k = 0 \Rightarrow \eta_k(\delta) = \infty$ it is clear that $A^k \notin (X, \bar{X})$ $\forall k$ such that $\bar{\delta}_k = 0$ (by our assumption that $P(\lambda)$ is always bounded). Thus $A^k \in (X, \bar{X})$ iff $\bar{\delta}_k = 1$, so since $(X, \bar{X})$ is minimal its value is $M(\bar{\delta})$.

(b) Define $\hat{\delta} \in \Delta$ by $\hat{\delta}_k = 1$ if $A^k \in (X, \bar{X})$ and $\hat{\delta}_k = 0$ otherwise. Obviously (since $(X, \bar{X})$ is minimal) $M(\bar{\delta}) \geqslant M(\hat{\delta})$ and $\hat{\delta} < \bar{\delta}$, so

$$M(\bar{\delta}) + \sum_{k=1}^{n} \lambda_k \bar{\delta}_k \geqslant M(\hat{\delta}) + \sum_{k=1}^{n} \lambda_k \hat{\delta}_k \quad \forall \lambda \geqslant 0$$

and the result follows.

Corollary.

$$V(\lambda) = \min \left\{ V(\eta(\delta)) + \sum_{k=1}^{n} \delta_k \lambda_k \mid \delta \in \Delta \right\} \quad \forall \lambda \geqslant 0.$$

Proof. Follows directly from Theorems 1 and 2. □

Note. Theorem 2 is proved for the case $n=2$ in [2].

The Corollary to Theorem 2 provides us with a method for computing $V(\lambda)$ for any $\lambda \geq 0$ by calculating $V(\eta(\delta))$ for each $\delta \in \Delta$. It should be noted that any method of computing $V(\eta(\delta))$ can be used.

Theorem 3. *Given $\bar{\lambda} \geq 0$, suppose that $\bar{\lambda} \in \Gamma(\bar{\delta})$ for a certain $\bar{\delta} \in \Delta$ and let $\lambda^1, \ldots, \lambda^l$ be the extreme points and $\zeta^1, \ldots, \zeta^p$ the extreme rays (if any) of $\Gamma(\bar{\delta})$. If $\bar{\lambda} = \sum_{i=1}^{l} \alpha_i \lambda^i + \sum_{j=1}^{p} \beta_j \zeta^j$ (with $\sum_{i=1}^{l} \alpha_i = 1, \alpha_i \geq 0, i=1, \ldots, l, \beta_j \geq 0, j=1, \ldots, p$), then the flow $f(\bar{\lambda})$ defined by $f(\bar{\lambda}) = \sum_{i=1}^{l} \alpha_i f(\lambda^i)$, where each $f(\lambda^i)$ is a max-flow for $P(\lambda^i)$, is a max-flow for $P(\bar{\lambda})$.*

In order to prove Theorem 3 we shall need the following lemma, in which we show that $\Gamma(\bar{\delta})$ is bounded in the direction of the λ_k axis for every k such that $\bar{\delta}_k = 1$.

Lemma 1. *Suppose $\Gamma(\bar{\delta})$ is unbounded and let $\bar{\zeta}$ be a ray of $\Gamma(\bar{\delta})$ (i.e., $\hat{\lambda} + \gamma\bar{\zeta} \in \Gamma(\bar{\delta})$ $\forall \lambda \in \Gamma(\bar{\delta})$ and $\gamma \geq 0$). Then $\sum_{k=1}^{n} \bar{\delta}_k \bar{\zeta}_k = 0$.*

Proof. By definition, $\lambda \in \Gamma(\bar{\delta})$ implies that

$$M(\bar{\delta}) + \sum_{k=1}^{n} \bar{\delta}_k \lambda_k = \min \left\{ M(\delta) + \sum_{k=1}^{n} \delta_k \lambda_k \mid \delta \in \Delta \right\} \leq M(0) < \infty.$$

But since $\bar{\zeta}$ is a ray of $\Gamma(\bar{\delta})$ we have that for $\hat{\lambda} \in \Gamma(\bar{\delta})$,

$$M(\bar{\delta}) + \sum_{k=1}^{n} \bar{\delta}_k (\hat{\lambda}_k + \gamma\bar{\zeta}_k) \leq M(0) \quad \forall \gamma \geq 0.$$

Therefore $\sum_{k=1}^{n} \bar{\delta}_k \bar{\gamma}_k = 0$. $\square$

Proof of Theorem 3. It is easy to verify the feasibility of $f(\bar{\lambda})$ (because $f(\bar{\lambda})$ is a convex combination of $f(\lambda^i)$ which are feasible for $P(\lambda^i)$, $i=1, \ldots, l$). Moreover, the value of $f(\bar{\lambda}) = V(f(\bar{\lambda}))$ is equal to $\sum_{i=1}^{n} \alpha_i V(\lambda^i)$, and since $\lambda^i \in \Gamma(\bar{\delta})$, we have by Theorem 2

$$V(f(\bar{\lambda})) = \sum_{i=1}^{l} \alpha_i V(\lambda^i) = \sum_{i=1}^{l} \alpha_i \left(M(\bar{\delta}) + \sum_{k=1}^{n} \bar{\delta}_k \lambda_k^i \right)$$
$$= M(\bar{\delta}) + \sum_{k=1}^{n} \bar{\delta}_k \left(\sum_{i=1}^{l} \alpha_i \lambda_k^i \right).$$

But by Lemma 1,

$$\sum_{k=1}^{n} \bar{\delta}_k \bar{\lambda}_k = \sum_{k=1}^{n} \bar{\delta}_k \left(\sum_{i=1}^{l} \alpha_i \lambda_k^i + \sum_{j=1}^{p} \beta_j \zeta_k^j \right)$$
$$= \sum_{k=1}^{n} \bar{\delta}_k \left(\sum_{i=1}^{l} \alpha_i \lambda_k^i \right).$$

Thus, $V(f(\bar{\lambda}))=M(\bar{\delta})+\sum_{k=1}^{n}\bar{\delta}_k\bar{\lambda}_k=V(\bar{\lambda})$, since $\bar{\lambda}\in\Gamma(\bar{\delta})$; and $f(\bar{\lambda})$ is a max-flow for $P(\bar{\lambda})$. □

Note that Theorem 3 enables us to find a max-flow for $P(\bar{\lambda})$ after we enumerate all the extreme points and extreme rays of $\Gamma(\bar{\delta})$. The following corollary shows that we actually need only the extreme points.

Corollary. *Given $\bar{\lambda}\geq 0$, suppose that $\bar{\lambda}\in\Gamma(\bar{\delta})$ for a certain $\bar{\delta}\in\Delta$, and let $\lambda^1,\ldots,\lambda^l$ be the extreme points of $\Gamma(\bar{\delta})$. If there exist $\alpha_1,\ldots,\alpha_p$ with $\bar{\lambda}_k=\sum_{i=1}^{l}\alpha_i\lambda_k^i$ for all k such that $\bar{\delta}_k=1$ and $\bar{\lambda}_k\geq\sum_{i=1}^{l}\alpha_i\lambda_k^i$ for all k such that $\bar{\delta}_k=0$, where $\alpha_i\geq 0$, $i=1,\ldots,l$, and $\sum_{i=1}^{l}\alpha_i=1$, then $f(\bar{\lambda})=\sum_{i=1}^{l}\alpha_i f(\lambda^i)$ is a max-flow for $P(\bar{\lambda})$.*

Proof. Define $\tilde{\lambda}=\sum_{i=1}^{l}\alpha_i\lambda^i$. Then $\bar{\lambda}-\tilde{\lambda}$ is clearly a ray of $\Gamma(\bar{\delta})$; thus it is a nonnegative combination of $\zeta^1,\ldots,\zeta^p$, and since by the Lemma, $\zeta_k^i=0\,\forall k$ such that $\bar{\delta}_k=1$, $i=1,\ldots,p$, the conditions of Theorem 3 are satisfied with the α_i's assumed to exist here and some β_i's, and the result follows. □

The preceding results provide us with the following method for computing $V(\lambda)$ and a max-flow $f(\lambda)\,\forall\lambda\geq 0$.

(i) For every $\delta\in\Delta$ find $V(\eta(\delta))$ by solving $P(\eta(\delta))$.

(ii) Enumerate all of the extreme points of $\Gamma(\delta)$ for each $\delta\in\Delta$ and compute max-flow at each of these points. (Note that by Theorem 2,

$$\Gamma(\delta)=\left\{\lambda\geq 0\mid V(\lambda)=V(\eta(\delta))+\sum_{k=1}^{n}\delta_k\lambda_k\right\}.\Big)$$

Once these computations are performed and recorded, $P(\lambda)$ can be solved for any $\lambda\geq 0$ as follows:

(a) Find $V(\lambda)$ using the expression in the Corollary to Theorem 2 and thus at the same time identify $\delta\in\Delta$ for which $\lambda\in\Gamma(\delta)$.

(b) Let λ^i, $i=1,\ldots,l$ be the extreme points of $\Gamma(\delta)$ and $f(\lambda^i)$, $i=1,\ldots,l$ associated max-flows (available from (ii) above). Find coefficients $\alpha_1,\ldots,\alpha_l\geq 0$ with $\sum_{i=1}^{l}\alpha_i=1$ such that $\lambda_k=\sum_{i=1}^{l}\alpha_i\lambda_k^i\,\forall k$ such that $\delta_k=1$, $\lambda_k\geq\sum_{i=1}^{l}\alpha_i\lambda_k^i\,\forall k$ such that $\delta_k=0$. (If necessary these can be found by solving a linear program.) Then a max-flow for $P(\lambda)$ is $f(\lambda)=\sum_{i=1}^{l}\alpha_i f(\lambda^i)$.

Example. Consider Fig. 2.

(i) Solve $P(0,0)$, $P(0,\infty)$, $P(\infty,0)$, and $P(\infty,\infty)$. In this example, $M(\delta)=V(\eta(\delta))\,\forall\delta\in\Delta$.

(ii) Enumerate points $\lambda^1,\lambda^2,\lambda^3,\lambda^4,\lambda^5$ and compute max-flows $f(\lambda^i)$, $i=1,\ldots,5$. (E.g., $\lambda^4=(V(\infty,0)-V(0,0),0)$.)

Given $\bar{\lambda}\geq 0$, we have (we use the particular $\bar{\lambda}$ in Fig. 2 as our example):

(a) $\bar{\delta}=(0,1)$, $V(\bar{\lambda})=M(0,1)+\bar{\lambda}_2$.

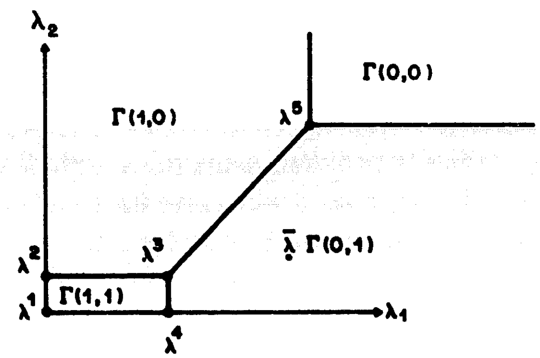

Fig. 2.

(b) The extreme points of $\Gamma(0, 1)$ are $\lambda^3, \lambda^4, \lambda^5$; thus (in this case) we find $\alpha \geq 0$ such that $\bar{\lambda}_2 = \alpha\lambda_2^4 + (1-\alpha)\lambda_2^3$ and take $f(\bar{\lambda}) = \alpha f(\lambda^4) + (1-\alpha)f(\lambda^3)$.

Remarks. (1) It is not efficient to solve $P(\eta(\delta))$ separately for every $\delta \in \Delta$. Depending on the method used it is probably possible to use the solution for one δ as a starting point for the next. The determination of a sequence by which the $P(\eta(\delta))$ are solved could increase the efficiency of an algorithm based on the results presented here.

(2) It would be inefficient to require enumeration of the extreme points of the $\Gamma(\delta)$ separately for each $\delta \in \Delta$, since most extreme points are common to more than one set. An algorithm exploiting the neighboring relationships of the sets $\Gamma(\delta)$ would be preferable.

(3) It is well-known that if all capacities are integer (in our case, this would have to include the parameters) then there exists an integer max-flow. However, the convex combinations of flows advocated here will not in general be integer even if the fixed and parametric capacities are integer. (Obviously, if the max-flow is unique the unique integer solution would be obtained.)

(4) It can be seen from the structure of $V(\lambda)$ that the parametric max-flow problem is intrinsically one requiring effort that increases exponentially with the number of parameters. In fact the computational effort of any good algorithm based along the lines suggested here would be proportional to the cardinality of Δ, which is equal to 2^n where n is the number of parameters. We discuss the ramifications of this fact in terms of the general multi-parametric linear program in the last section.

3. The nonzero lower capacities case

In this section we extend the preceding results to the case in which the lower capacities are not necessarily all zero.

The introduction of nonzero lower capacities raises the question of feasibility of

$P(\lambda)$ for a given λ. Therefore we proceed first to determine the feasibility set

$$\mathscr{F} = \{\lambda \geqslant O \mid P(\lambda) \text{ has a feasible solution}\}.$$

We shall deal with the feasibility problem using the standard technique of defining a new network $\tilde{G} = \{\tilde{\mathcal{N}}, \tilde{\mathscr{A}}\}$ with zero lower capacities and new upper capacities. The existence or nonexistence of a feasible flow for $P(\lambda)$ is determined by solving a max-flow problem in $\tilde{G}$. We can apply the results of Section 2 to obtain $\mathscr{F}$.

We make the following definitions:

$$S = \{i \in N \mid l(N, i) - l(i, N) \equiv a(i) > 0\},$$

$$T = \{i \in N \mid l(i, N) - l(N, i) \equiv b(i) > 0\},$$

$$I = \{i \in N \mid l(i, N) - l(N, i) = 0\},$$

$$a(S) = \sum_{i \in S} a(i), \qquad b(T) = \sum_{i \in T} b(i).$$

The network $\tilde{G} = \{\tilde{N}, \tilde{\mathscr{A}}\}$ is defined as follows:

$$\tilde{N} = N \cup \{\tilde{s}, \tilde{t}\},$$

$$\tilde{\mathscr{A}} = \mathscr{A} \cup \{(\tilde{s}, j) \mid j \in S\} \cup \{(i, \tilde{t}) \mid t \in T\}.$$

We define upper capacities $\tilde{c}(i, j)$ on arcs of $\tilde{\mathscr{A}}$ by:

$$\tilde{c}(i, j) = \begin{cases} a(j) & \text{if } i = \tilde{s}, j \in S, \\ b(i) & \text{if } j = \tilde{t},\ i \in T, \\ c(i, j) - l(i, j) & \text{if } (i, j) \in \mathscr{A}_2, \\ \lambda_k - l(i, j) \equiv \tilde{\lambda}_k & \text{if } (i, j) = A^k \in \mathscr{A}_1. \end{cases}$$

We define the corresponding max-flow problem $\tilde{P}(\lambda)$ in $\tilde{G}$ as follows:

$$\tilde{P}(\lambda)\colon \max \tilde{f}(\tilde{t}, \tilde{s}),$$

subject to

$$\tilde{f}(\tilde{N}, i) - f(i, \tilde{N}) = 0 \quad \forall i \in N,$$

$$0 \leqslant \tilde{f}(i, j) \leqslant \tilde{c}(i, j) \quad \forall (i, j) \in \tilde{\mathscr{A}} \setminus \mathscr{A}_2,$$

$$0 \leqslant \tilde{f}(i_k, j_k) \leqslant \tilde{\lambda}_k, \quad k = 1, \ldots, n.$$

We shall denote a max-flow for $\tilde{P}(\lambda)$ by $\tilde{f}(\lambda)$ and its value by $\tilde{V}(\lambda)$. The relationship between the max-flow value of $\tilde{P}(\lambda)$ and feasibility of $P(\lambda)$ is given in the following theorem.

Theorem 4. (i) $\tilde{V}(\lambda) \leqslant a(S)\ \forall \lambda$.

(ii) *$P(\lambda)$ is feasible iff $\tilde{V}(\lambda) = a(S)$.* (*Remark.* By construction, $a(S) = b(T)$.)

Proof. (i) Since $(\{\bar{s}\}, \bar{N}\backslash\{\bar{s}\})$ is an $\bar{s}$-$\bar{t}$ cut of capacity $a(S)$.

(ii) (a) Suppose $P(\lambda)$ is feasible and let f be a feasible flow. Define a flow $\bar{f}$ on $\bar{G}$ as follows:

$$\bar{f}(x,y)=f(x,y)-l(x,y), \quad (x,y)\in\mathcal{A},$$

$$\bar{f}(\bar{s},x)=a(x), \quad x\in S,$$

$$\bar{f}(x,\bar{t})=b(x), \quad x\in T.$$

It is easy to verify that f satisfies conservation. In addition, it clearly satisfies capacity constraints (by definition of $\bar{c}$, since f did) and has value $a(S)$.

(b) Let $\bar{f}$ be a flow in $\bar{G}$ of value $a(S)$. Define f on $\mathcal{A}$ by $f(x,y)=\bar{f}(x,y)+l(x,y)$. Again it is easy to verify that f is a feasible flow on G. □

In the next theorem we apply the results of Section 2 to Theorem 4 to obtain $\mathcal{F}$.

Theorem 5. (i) $\mathcal{F}\neq\emptyset$ *iff* $\bar{V}(\eta(0))=a(S)$.

(ii) *If* $\mathcal{F}\neq\emptyset$, *then*

$$\mathcal{F}=\{\lambda\geq 0 \,\Big|\, \sum_{k=1}^{n}\delta_k\lambda_k\geq\bar{V}(\eta(0))-\bar{V}(\eta(\delta))+\sum_{k=1}^{n}\delta_k l(A^k)\ \forall\delta\in\Delta\}.$$

Proof. By the corollary to Theorem 2, since $\bar{\lambda}_k=\lambda_k-l(A^k)$ we have

$$\bar{V}(\lambda)=\min\left\{\bar{V}(\eta(\delta))+\sum_{k=1}^{n}\delta_k\lambda_k-\sum_{k=1}^{n}\delta_k l(A^k)\,\Big|\,\delta\in\Delta\right\},$$

and since $\bar{V}(\eta(0))\geq\bar{V}(\lambda)$ for any λ, $P(\lambda)$ is feasible iff this quantity is equal to $a(S)$. Thus:

(i) Clearly if $\bar{V}(\eta(0))<a(S)$, then $\mathcal{F}=\emptyset$ (since $\bar{V}(\eta(0))\geq\bar{V}(\lambda)$ for any λ). If $\bar{V}(\eta(0))=a(S)$ then for λ sufficiently large $\bar{V}(\eta(0))=\bar{V}(\lambda)$ so $\mathcal{F}\neq\emptyset$.

(ii) Assuming $\bar{V}(\eta(0))=a(S)$, then since $P(\lambda)$ is feasible iff the above minimum is equal to $a(S)$, we have that $P(\lambda)$ is feasible iff the above minimum is equal to $\bar{V}(\eta(0))$, i.e., iff

$$\bar{V}(\eta(0))\leq\bar{V}(\eta(\delta))+\sum_{k=1}^{n}\delta_k\lambda_k-\sum_{k=1}^{n}\delta_k l(A^k)\quad\forall\delta\in\Delta,$$

and the result follows by rearranging terms. □

Thus, the explicit determination of $\mathcal{F}$ can be accomplished by a method similar to that of obtaining $V(\lambda)$ in the previous section (here the flow problems are solved in $\bar{G}$ instead of G).

The set $\mathcal{F}$, if nonempty, is a polyhedral set imbedded in the n-dimensional nonnegative ortant with up to 2^n-1 facets; that is, $\mathcal{F}$ can be expressed as the solutions to a set of at most 2^n-1 nonredundant linear inequalities in n variables.

The structure of $\mathscr{F}$ is rather simple and redundant inequalities can be easily detected and removed if desired. In fact, the inequality associated with a given $\bar{\delta}\in\Delta$ is redundant iff

$$\sum_{\delta\in S}\left[\sum_{k=1}^{n}\delta_k l(A^k)-\tilde{V}(\eta(\delta))\right]\geq\sum_{k=1}^{n}\bar{\delta}_k l(A^k)-\tilde{V}(\eta(\bar{\delta})),$$

where S is any subset of $L(\bar{\delta})\equiv\{\delta\in\Delta\mid\delta<\bar{\delta}\}$ such that $\sum_{\delta\in S}\delta=\bar{\delta}$.

To simplify the presentation we shall write $\mathscr{F}$ as

$$\mathscr{F}=\left\{\lambda\geq 0\mid\sum_{k=1}^{n}\delta_k\lambda_k\geq d(\delta),\ \delta\in\Delta\right\}$$

where

$$d(\bar{\delta})\equiv\max\left[\left\{\sum_{\delta\in S}\left[\sum_{k=1}^{n}\delta_k l(A^k)-\tilde{V}(\eta(\delta))\mid S\subseteq L(\bar{\delta}),\ \sum_{\delta\in S}\delta=\bar{\delta}\right\},\right.\right.$$
$$\left.\left.\sum_{k=1}^{n}\bar{\delta}_k l(A^k)-\tilde{V}(\eta(\bar{d}))\right]\right].$$

We remark that to find the feasibility set for a problem with parametric lower capacities only, we may define a network similar to $\bar{G}$ but depending on the fixed upper capacities, prove a theorem analogous to Theorem 4, and use it in an analogous way to prove a Theorem analogous to Theorem 5. The development from here on can then be appropriately adapted to the parametric lower capacity case. However, if we wish to consider problems with both parametric lower and upper capacities, the transformation described in Section 1 for changing all parametric lower capacities to parametric upper capacities must be made.

Our next objective is analysis of the function $V(\lambda)$. This task is not as simple as in the zero lower capacity case; although Theorem 1 is easily extended, the explicit determination of $V(\lambda)$ is more complicated and requires, besides the solution of max-flow problems, the solutions of as many linear programs whose complexities increase with n. Theorem 6 is a generalization of Theorem 1.

Theorem 6.

$$V(\lambda)=\min\left\{M(\delta)+\sum_{k=1}^{n}\delta_k\lambda_k\mid\delta\in\Delta\right\}\quad\forall\lambda\in\mathscr{F}.$$

($M(\delta)$ is defined as in Section 1.)

Proof. Same proof as Theorem 1. □

As before, the $M(\delta)$ for $\delta\in\Delta$ are not readily available. Theorem 7 provides the basis for a method of computing $M(\delta)$ for every $\delta\in\Delta$ or showing that the particular quantity

$$M(\delta)+\sum_{k=1}^{n} \delta_k\lambda_k$$

is redundant in the expression above for $V(\lambda)$. We first prove a Lemma.

Lemma 2. *Given $\bar{\delta}\in\Delta$, consider the following linear program:*

$$Q(\bar{\delta}):\max \mu$$

subject to

$$M(\delta)+\sum_{k=1}^{n} (\delta_k-\bar{\delta}_k)\lambda_k-\mu\geq 0 \quad \forall\delta\in L(\bar{\delta}), \qquad \lambda\in\mathcal{F}.$$

(i) *If $\mathcal{F}\neq\emptyset$, then $Q(\bar{\delta})$ has an optimal solution $\forall\bar{\delta}\in\Delta$.*

(ii) *If $(\bar{\mu},\bar{\lambda})$ is an optimal solution, then*

$$\bar{\mu}=\min\left\{M(\delta)+\sum_{k=1}^{n}\delta_k\bar{\lambda}_k \mid \delta\in L(\delta)\right\}-\sum_{k=1}^{n}\bar{\delta}_k\bar{\lambda}_k.$$

(iii) *If $(\bar{\mu},\bar{\lambda})$ is an optimal solution, then*

$$\bar{\mu}\geq\min\left\{M(\delta)+\sum_{k=1}^{n}\delta_k\lambda_k \mid \delta\in L(\delta)\right\}-\sum_{k=1}^{n}\bar{\delta}_k\lambda_k \quad \forall\lambda\in\mathcal{F}.$$

Proof. Follows directly from the formulation. □

Theorem 7. *Let $(X,\bar{X})$ be a minimum cut associated with $P(\eta(\bar{\delta})+\lambda)$, where $(\bar{\mu},\bar{\lambda})$ is an optimal solution of $Q(\bar{\delta})$.*

(a) *If $A^k\in(X,\bar{X})$ for all k such that $\bar{\delta}_k=1$, then*

$$M(\bar{\delta})=V(\eta(\bar{\delta})+\bar{\lambda})-\sum_{k=1}^{n}\bar{\delta}_k\bar{\lambda}_k.$$

(b) *If $\exists k$ such that $\bar{\delta}_k=1$ and $A^k\notin(X,\bar{X})$, then*

$$M(\bar{\delta})+\sum_{k=1}^{n}\bar{\delta}_k\lambda_k\geq\min\left\{M(\delta)+\sum_{k=1}^{n}\delta_k\lambda_k \mid \delta\in L(\delta)\right\} \quad \forall\lambda\in\mathcal{F}.$$

Proof. (a) The proof is analogous to the proof of Theorem 2(a).

(b) Define $\hat{\delta}\in\Delta$ by $\delta_k=1$ iff $A^k\in(X,\bar{X})$. Then the value of $(X,\bar{X})$ in the network for $P(\eta(\bar{\delta})+\lambda)$ is

$$M(\hat{\delta})+\sum_{k=1}^{n}\hat{\delta}_k\bar{\lambda}_k,$$

and by minimality

$$M(\hat{\delta})+\sum_{k=1}^{n}\hat{\delta}_k\bar{\lambda}_k \leq M(\bar{\delta})+\sum_{k=1}^{n}\bar{\delta}_k\bar{\lambda}_k.$$

Thus by Lemma 2(ii),

$$\bar{\mu}=M(\hat{\delta})+\sum_{k=1}^{n}\hat{\delta}_k\bar{\lambda}_k-\sum_{k=1}^{n}\bar{\delta}_k\bar{\lambda}_k \leq M(\bar{\delta}).$$

Hence by 2(iii),

$$\min\left\{M(\delta)+\sum_{k=1}^{n}\delta_k\lambda_k \mid \delta\in L(\bar{\delta})\right\}\leq \bar{\mu}+\sum_{k=1}^{n}\bar{\delta}_k\bar{\lambda}_k \leq M(\bar{\delta}) + \sum_{k=1}^{n}\bar{\delta}_k\lambda_k \quad \forall\lambda\in\mathscr{F}$$

which proves the claim. □

The foregoing results suggest an algorithm designed along the following lines. We present not an algorithm in all detail but rather only the basis for one. We assume that $\mathscr{F}$ is already given (that is, $d(\delta)$ has been computed $\forall\delta\in\Delta$). We denote by $F(i)$ the set $\{\delta\in\Delta \mid |\delta|=i\}$, $i=1,\dots,n$ where $|\delta|\equiv|\{i \mid \delta_i=1\}|$.

(i) Compute $M(\delta)\ \forall\delta\in F(1)$. (Note that no linear programs are needed at this stage, since $\bar{\lambda}_k=d(\delta)$ for $\delta_k=1$ can be used.)

(ii) Given all the relevant $M(\hat{\delta})$ for $\hat{\delta}\in\bigcup_{i=1}^{m-1}F(i)$, then for every $\bar{\delta}\in F(m)$:

(ii.a) Solve the linear program $Q(\bar{\delta})$; let $(\bar{\mu},\bar{\lambda})$ be an optimal solution.

(ii.b) Solve the max-flow problem $P(\eta(\bar{\delta})+\lambda)$; let $(X,\bar{X})$ be an associated minimum cut.

(ii.c) If $A^k\in(X,\bar{X})$ for every k such that $\bar{\delta}_k=1$, let $M(\bar{\delta})=V(\eta(\bar{\delta})+\bar{\lambda})-\sum_{k=1}^{n}\bar{\delta}_k\bar{\lambda}_k$; otherwise $M(\bar{\delta})$ is always redundant in the expression for the function $V(\lambda)$.

(iii) Set $m=m+1$ and go to (ii) unless $m=n$.

The construction of min-cuts and max-flows as functions of the parameter vector λ would be analogous to the construction presented in Section 2. In particular, the min-cut associated with $P(\eta(\bar{\delta})+\bar{\lambda})$ is also a min-cut for all $\lambda\in\Gamma(\bar{\delta})=\{\lambda\in\mathscr{F} \mid M(\bar{\delta})+\sum_{k=1}^{n}\bar{\delta}_k\lambda_k=V(\lambda)\}$ and the extreme points of $\Gamma(\delta)\ \forall\delta\in\Delta$ can be enumerated and used as in Section 2.

4. Comparisons and ramifications concerning general multi-parametric methods

The problem $P(\lambda)$ is a special case of the multidimensional parametric right-hand-side linear program, a solution method for which is given in [3]. In this section we compare our method with this general method (which can be adapted slightly to network analysis and applied to our problem), and indicate how our results are of theoretical interest since they imply ramifications concerning the difficulty of the general problem. For convenience we refer to "our method" instead of "an algorithm based on our results." (Note: See also the remarks at the end of Section 2.)

In parametric methods where we wish to solve a problem for all values of a parameter vector λ, we actually first solve a number of problems for certain specific values of λ and record the information; we will refer to this as phase 1. Afterwards, given any specific λ we must use the information from phase 1 to obtain a solution for λ; we will call this phase 2. Phase 1 for the general method in [3] consists of solving the problem for a particular starting value of λ and performing sensitivity analysis; specifically, the region of parameters for which the starting basis is optimal is found as the solution set to a system of linear inequalities, and after solving a certain linear program one facet of the set is determined. An alternate solution for the facet is found by performing a pivot (if adapted to network methods, by performing a flow augmention–see[1]), a new region is found, and the process continues. Our phase 1 consists of (i) and (ii) of the method given in Section 2.

We wish to discuss similarities and distinctions between the two methods in both phases 1 and 2.

In phase 1 both methods produce a partition of the parameter space in which the optimal solution value is an affine function over each region of the partition. The most important distinction is in these partitions; in fact, our partition contains the partition found in the general method. In the one-dimensional case, for example, the general method would generate a sequence of intervals, each corresponding to an augmenting path (see [1]); the endpoint of the last interval found would be the λ beyond which the optimal solution value remains constant (i.e., there are no more augmenting paths). Our method produces exactly this point (namely, $V(\infty)-V(0)$) and only the two intervals $[0, V(\infty)-V(0)]$ and $[V(\infty)-V(0), \infty)$, which are in some sense intrinsic to the problem. (We refer in this instance to the zero lower capacity case.) A similar situation would result in higher dimensions. Furthermore, in our method we know that the number of regions in the partition is at most 2^n, whereas in the general method nothing is known about this a priori. In fact, since our solution may have as many as 2^n "intrinsic" regions and the general method applied to our problem in general produces even more than this, and since our problem is simpler than the general multiparametric linear program, we see that the general method applied to the general problem must be very complicated.

We next discuss differences in the effort in actually generating the partition. In the zero lower capacity case we obtain our partition easily after solving at most 2^n max-flow problems in any order. (Some order of solving the problems can increase the efficiency by using one solution as the starting point for the next.) The general method, as we saw above, not only involves solving a linear program to find each facet of every region, but also requires extremely complex logistics in order to keep track of the facets and regions covered. In the nonzero lower capacity case we must solve also up to 2^n linear programs (one for each region); but this is still much less effort than in the general case where one linear program must be solved for each facet of a larger number of regions. Also, notice that our method saves work if it is known that all lower capacities are zero; the general method would proceed exactly the same way in both cases.

We now compare the form of the information obtained after phase 1 in both methods. In the general method a set of inequalities must be stored for each region; our method requires storage of only $\leq 2^n$ numbers (one for each region). If one is interested only in the max-flow value, in the general method we need also store one number for each region; our method contains this information in the $\leq 2^n$ constants referred to above. If one is interested in min-cuts, we need only record one for each region; this is true in both methods but in a true sense our method produces the minimum number of such regions. Note, however, that if we wish to produce max-flows, our method also requires storage of the inequalities defining each region, in order to find the extreme points, whereas the general method records just one max-flow (and an augmenting path) per region.

We now turn to phase 2, the computation of the solution for a given $\bar{\lambda}$. We first discuss the determination of feasibility or infeasibility (if lower capacities are not identically zero). In the general method we must substitute $\bar{\lambda}$ into a system of inequalities for each region, whereas in our method we substitute $\bar{\lambda}$ into only $\leq 2^n - 1$ terms and find the minimum.

After feasibility is determined, in order to find the max-flow value our method requires additionally the substitution of $\bar{\lambda}$ into $\leq 2^n$ more terms in order to find which region contains $\bar{\lambda}$; in the general method the region is found simultaneously as feasibility is determined (however, this is still in general more work than in our method).

To find a max-flow our method requires additionally the solution of one linear program for each region (still less work than the general method).

We mention again (see Remark (3) at the end of Section 2) that our method does not guarantee integer flows; the general method does.

Finally, we remark that our approach is global, and does not provide for the reduction of effort if solutions are desired only for some small subset of the parameter space. In this case the general method presented in [3], which is based on sensitivity analysis, would be more efficient.

References

[1] L.R. Ford, Jr., and D.R. Fulkerson, Flows in Networks (Princeton University Press, Princeton, NJ, 1962).

[2] L.S. Shapley, On network flow functions, in: Notes on Linear Programming and Extensions, Part 50, Rand Research Memorandum RM-2338, (The Rand Corporation, 1959).

[3] D.M. Walters, Multiparametric Mathematical Programming Problems, Ph.D. Thesis, Department of Industrial Engineering and Operations Research, University of California, Berkeley, 1976.