

MODELS AND ALGORITHMS FOR MULTIPLE CRITERIA LINEAR COST NETWORK PROGRAMS

X. Q. YANG¹ and C. J. GOH¹

(Received 10 October 1995; revised 17 September 1996)

Abstract

In this paper, we discuss a general model for multiple criteria linear cost network flow problems. This model includes several classes of existing models in the operations research literature as special cases. Based on this model, a search algorithm for finding a feasible solution of the concurrent flow problem is suggested and illustrative numerical examples are given. This search algorithm is also extended to obtain a new algorithm for finding the efficient frontier of a multiple criteria linear program.

1. Introduction

It is well known that the multiple criteria concept is useful in solving a variety of optimization problems arising from economics, social sciences and engineering, see [9] and [12]. In general, multiple criteria optimization problems with general nonlinear cost are difficult to solve exactly [4, 5]. As a result more research effort has been directed at linear cost problems in view of their tractability. Since the invention of the simplex method [2] for solving linear programming problems, many extensions of the method have been studied. In particular, by exploiting the network structure, the network simplex method [7] often proves to be several orders of magnitude faster than the standard simplex method. In another important extension, Yu and Zeleny [13] developed a multiple criteria simplex method for solving linear programming problems with several objectives. Recently Ruhe [8] put these ideas together and developed another simplex method for solving a multiple criteria linear cost network flow problem.

A simple and effective algorithm for finding the efficient frontier for a bicriteria network program was developed in Aneja and Nair [1]. This procedure is also applicable for finding the efficient frontier of a bicriteria linear program.

¹Department of Mathematics, The University of Western Australia, Nedlands WA 6907, Australia.
© Australian Mathematical Society 1999, Serial-fee code 0334-2700/99

The aim of this paper is to consider a more general multiple criteria linear cost network flow model which includes the problem addressed by Ruhe [8] as a special case. In addition, this model also includes two other important network flow problems as special cases; namely (i) finding a feasible solution for the single criterion concurrent flow problems and (ii) solving the subproblem in a resource directive-decomposition method for multiple criteria multiple commodity flow problems. This general model will be formulated in Section 2, with the reduction of the general model to the special cases discussed in Section 3. A search algorithm for finding a feasible solution to the concurrent flow problem based on the multiple criteria model is presented in Section 4 with full justification. Using similar ideas as this search algorithm, an extension of Aneja and Nair’s algorithm for finding the efficient frontier of a multiple criteria linear program is presented in Section 5.

2. The general multiple criteria linear cost network flow problem

Let R^r be an r -dimensional space, $R^r_+ = \{\xi \in R^r : \xi_i \geq 0, i = 1, \dots, r\}$ be the (closed) positive octant, $R^r_- = -R^r_+$ be the (closed) negative octant and define $\text{int } R^r_+ = \{\xi \in R^r : \xi_i > 0, i = 1, \dots, r\}$.

DEFINITION 1. Let $\xi, \eta \in R^r$, and we define the following ordering relationships:

$$\begin{aligned} \xi \leq \eta &\iff \eta - \xi \in R^r_+; \\ \xi \leq \eta &\iff \eta - \xi \in R^r_+ \setminus \{0\}; \\ \xi \not\leq \eta &\iff \eta - \xi \notin R^r_+ \setminus \{0\}; \\ \xi < \eta &\iff \eta - \xi \in \text{int } R^r_+; \\ \xi \not< \eta &\iff \eta - \xi \notin \text{int } R^r_+. \end{aligned}$$

The ordering relationships $\geq, \geq, \not\leq, >, \not<$ are defined similarly. It is clear that $\xi \not\leq \eta \implies \xi \not< \eta$. If $r = 1$, then either $x \not\leq y$ or $x \not< y$ implies $x \geq y$.

DEFINITION 2. The convex hull of points $\xi_1, \dots, \xi_p \in R^r$ is defined by

$$\text{Co}\{\xi_1, \dots, \xi_p\} = \left\{ \xi = \sum_{i=1}^p \beta_i \xi_i : \beta_i \geq 0, \sum_{i=1}^p \beta_i = 1 \right\}.$$

Consider the following multiple criteria optimization problem:

$$\begin{aligned} \text{(MCOP)} \quad & \text{V-Minimize } f(x) \\ & \text{subject to } x \in C, \end{aligned}$$

where $f : R^n \rightarrow R^r$ is a vector-valued function, C is a subset of R^n and V-Minimize seeks to find all the efficient solutions of the MCOP. A solution x is said to be an *efficient solution* [9] of the MCOP if there exists no $y \in C$ such that

$$f(y) \leq f(x).$$

The corresponding function value $f(x)$ is said to be an *efficient point* of the MCOP.

Let $G = (N, A)$ be a directed graph, where N denotes a finite set of nodes and A denotes a finite set of directed arcs.

DEFINITION 3. The node-arc incidence matrix $E \in R^{|N| \times |A|}$ for G is defined by (see [7]):

$$E_{ia} = \begin{cases} +1, & \text{if the source node of arc } a \text{ is } i; \\ -1, & \text{if the sink node of arc } a \text{ is } i; \\ 0, & \text{otherwise.} \end{cases}$$

The concurrent flow problem (namely, the multiple commodity flow problem) is:

$$\begin{aligned} \text{(CCFP)} \quad & \text{Find a flow } x \text{ that satisfies the constraints:} \\ & Ex_k = b_k, \quad k = 1, \dots, K \\ & \sum_{k=1}^K x_{ka} \leq d_a, \quad \forall a \in A, \\ & 0 \leq x_k \leq u_k, \end{aligned}$$

where d_a is the *shared* capacity bound for arc a . The main purpose of the paper is to introduce a method solving the CCFP. This is done by transforming the CCFP into the problem of finding a negative efficient point of a multiple criteria linear cost network program.

We shall first focus our attention on an important special case of the MCOP referred to as the *multiple criteria linear cost network program* where f is linear and C is a polyhedral set of multiple commodity flows defined by

$$\begin{aligned} \text{(GMLN)} \quad & \text{V-Minimize } (c_1^\top x, \dots, c_r^\top x) \\ & \text{subject to } Ex_k = b_k, \quad k = 1, \dots, K, \\ & 0 \leq x \leq u, \end{aligned}$$

where $E \in R^{|N| \times |A|}$ is the usual node-arc incidence matrix for G and

$$\begin{aligned} x &= [x_1^\top, \dots, x_k^\top]^\top \in R^{K|A|}, \\ x_k &= [x_{k1}, \dots, x_{k|A|}]^\top \in R^{|A|}, \quad k = 1, \dots, K, \end{aligned}$$

is the flow of commodity k ,

$$\begin{aligned}
 \mathbf{b} &= [\mathbf{b}_1^\top, \dots, \mathbf{b}_K^\top]^\top \in \mathbf{R}^{K|N|}, \\
 \mathbf{b}_k &= [b_{1k}, \dots, b_{|N|k}]^\top \in \mathbf{R}^{|N|}, \quad k = 1, \dots, K, \text{ is the requirement of commodity } k, \\
 \mathbf{c}_i &= [c_{i1}^\top, \dots, c_{iK}^\top]^\top \in \mathbf{R}^{K|A|}, \quad i = 1, \dots, r, \text{ is the cost of criterion } i, \\
 \mathbf{c}_{ik} &= [c_{ik1}, \dots, c_{ik|A|}]^\top \in \mathbf{R}^{|A|}, \quad k = 1, \dots, K, \\
 \mathbf{u} &= [\mathbf{u}_1^\top, \dots, \mathbf{u}_K^\top]^\top \in \mathbf{R}^{K|A|} \text{ is the upper bound on } \mathbf{x} \text{ and} \\
 \mathbf{u}_k &= [u_{k1}, \dots, u_{k|A|}]^\top \in \mathbf{R}^{|A|}.
 \end{aligned}$$

Note that the multiple commodity flows \mathbf{x}_k in the GMLN are coupled by the multiple criteria but not by any shared capacity constraints.

3. Some special cases of network flow problems

Some important special cases of the GMLN are discussed as follows.

EXAMPLE 1. The special case of the GMLN with $K = 1$ is given by

$$\begin{aligned}
 \text{(MLNP)} \quad & \text{V-Minimize } (\mathbf{c}_1^\top \mathbf{x}, \dots, \mathbf{c}_r^\top \mathbf{x})^\top \\
 & \text{subject to } E\mathbf{x} = \mathbf{b}, \\
 & \quad 0 \leq \mathbf{x} \leq \mathbf{u},
 \end{aligned}$$

which reduces to the single commodity, multiple criteria, linear cost network flow problem previously considered by Ruhe [8]. However by taking the K copies of the network in the GMLN as a large network and rearranging the variables it is clear that the GMLN is reduced to the MLNP. A multiple criteria network simplex method for finding the set of all efficient extreme points has been developed in Ruhe [8] by specializing the multiple criteria simplex method of Yu and Zeleny [13] and Yu [12] to take advantage of the network structure. Further restriction of the MLNP to the single criterion case of $r = 1$ reduces to the simplest linear cost network flow problem considered in [7]. As a digression, we note also that the MLNP includes, as a special case, the multiple criteria efficient path problem which is of sufficient interest in its own right. Given a source node s and a sink node t for a strongly connected digraph, we seek a *pareto-optimal* or *efficient* path P joining s and t such that the vector cost

$$\phi(P) = \sum_{a \in P} \mathbf{c}^a$$

is not dominated by another path, that is,

$$\phi(P) \not\leq \phi(P'), \quad \forall P' \text{ joining } s \text{ and } t,$$

where $c^a \in R^r$ is a vector cost associated with each arc $a \in A$. Clearly, the efficient path problem is a special case of the MLNP if we restrict the parameters of the MLNP such that $[c_i]_a = [c^a]_i$, $u_a = \infty \forall a$, $b_s = 1$, $b_t = -1$, and $b_i = 0 \forall i \neq s, t$. In practice, the efficient path problem is solved by a specialized technique, see [10].

EXAMPLE 2. Consider the problem CCMP. Note that there is a shared capacity bound for each arc a . It is this shared capacity constraint that makes the problem difficult. In principle, this just amounts to finding a feasible solution to a large linear program and can be solved by the usual artificial variable method commonly used in the two phase simplex algorithm. In practice, however, one would like to be able to take advantage of the network structure of the problem and to reduce the problem to a sequence of decoupled problems of a much smaller size. With a slight twist, it is possible to get around the shared capacity constraint by posing the CCFP as a special case of the GMLN. Consider the following problem:

$$\begin{aligned} \text{(MCP)} \quad & \text{V-Minimize} \quad \left(\sum_{k=1}^K x_{k1} - d_1, \dots, \sum_{k=1}^K x_{k|A|} - d_r \right)^T \\ & \text{subject to} \quad Ex_k = b_k, \quad k = 1, \dots, K, \\ & \quad \quad \quad 0 \leq x_k \leq u_k. \end{aligned}$$

Note that here $r = |A|$. Let $f(x) = \left(\sum_{k=1}^K x_{k1} - d_1, \dots, \sum_{k=1}^K x_{kr} - d_r \right)$ and let \mathcal{E} and $f(\mathcal{E})$ be the set of all efficient solutions and efficient points of the MCP respectively. The following result allows us to determine if a feasible solution to the MCP exists and to construct it if it does.

THEOREM 1. *The problem CCFP has a feasible solution if and only if $f(\mathcal{E}) \cap R'_- \neq \emptyset$, where $f(\mathcal{E})$ is the set of all efficient points of the MCP. Furthermore, any efficient solution x satisfying $f(x) \in f(\mathcal{E}) \cap R'_-$ is a feasible solution of the CCFP.*

PROOF. Assume that the problem CCFP has a feasible solution x . Then x satisfies the constraints of the MCP and

$$\sum_{k=1}^K x_{ka} - d_a \leq 0, \quad \forall a \in A. \quad (1)$$

Thus $f(x) \in R'_-$. By a theorem of multiple criteria optimization (see [9]), there exists an efficient solution x^* of the MCP such that $f(x^*) \leq f(x)$. Thus $f(x^*) \in R'_-$. So $f(\mathcal{E}) \cap R'_- \neq \emptyset$. Conversely, if $f(\mathcal{E}) \cap R'_- \neq \emptyset$, then there exists an efficient solution x^* such that x^* satisfies (1) and the constraints of the MCP. Thus $f(x^*) \leq 0$. Therefore x^* is a feasible solution of the CCFP.

Since the MCP is a linear problem, the efficient frontier $f(\mathcal{E})$ is the union of simplices, where the $r = |A|$ corners of each of these simplices are images (under f) of the efficient solutions the MCP. A feasible solution to the CCFP exists if and only if this efficient frontier intersects the negative octant according to Theorem 1. This gives us clues to a conceptual algorithm which, at each intermediate iteration, solves a scalarized problem of the MCP and checks for the non-empty intersection of some simplex with the negative octant. Whenever the algorithm finds such a non-empty intersection, it stops and gives a feasible solution to the CCFP, otherwise it continues for a finite number of steps until it eventually declares that no feasible solution exists. We will describe this algorithm further in Section 4 and provide all the theoretical justification. The existence of a feasible solution of the problem the CCFP has also been investigated in [3, 6] *via* an alternative theorem.

EXAMPLE 3. Consider the multiple criteria multiple commodity linear cost flow problem with shared capacity constraints:

$$\begin{aligned} \text{V-Minimize} \quad & (c_1^T x, \dots, c_r^T x) \\ \text{subject to} \quad & Ex_k = b_k, \quad k = 1, \dots, K, \\ & \sum_{k=1}^K x_{ka} \leq d_a, \quad \forall a \in A, \\ & 0 \leq x \leq u. \end{aligned}$$

If the resource-directive decomposition method is used to solve this problem, a sequence of subproblems of the following the GMLN are required to be solved (see [7]):

$$\begin{aligned} \text{V-Minimize} \quad & (c_1^T x, \dots, c_r^T x) \\ \text{subject to} \quad & Ex_k = b_k, \quad k = 1, \dots, K, \\ & 0 \leq x \leq v, \end{aligned}$$

where v is some modified upper bound to x which is a decision variable for the master problem, but fixed for the subproblems.

4. A search algorithm to find a feasible solution for the CCFP

Instead of using the usual artificial variable method, we now propose an alternative which is based on the idea that a large number of decoupled subproblems is easier to solve than a single large coupled problem and that the decoupled problems can take advantage of the network structure much more readily than can the coupled problem.

We begin by defining the scalarized problem $P(\lambda)$ for the MCP, where $\lambda \in \text{int } \mathbf{R}_+^r$.

$$\begin{aligned} \text{Problem } P(\lambda) \quad & \text{Minimize} \quad \sum_{a=1}^r \lambda_a f_a(\mathbf{x}_k) = \sum_{a=1}^r \lambda_a \left(\sum_{k=1}^K x_{ka} - d_a \right) \\ & \text{subject to} \quad E\mathbf{x}_k = \mathbf{b}_k, \quad k = 1, \dots, K, \\ & \quad \quad \quad 0 \leq \mathbf{x}_k \leq \mathbf{u}_k. \end{aligned}$$

Let the solution to $P(\lambda)$ be denoted by $\mathbf{x}^*(\lambda)$ which is an efficient solution to the MCP; and let $\mathbf{f}(\mathbf{x}^*(\lambda))$ be the corresponding efficient point in the cost space. We shall summarize the algorithm as follows and justify the steps shortly. For convenience, a feasible solution to the CCFP, if it exists, is abbreviated as MFS (multiple commodity feasible solution).

Let $\xi^i \in \mathbf{R}^r, i = 1, \dots, r$ and $\Gamma = \text{Co}(\xi^1, \xi^2, \dots, \xi^r)$ be a $r - 1$ simplex, that is, $\{\xi^1, \xi^2, \dots, \xi^r\}$ are affinely independent. The focal point, denoted by $\text{Foc}(\Gamma)$, to Γ is defined to be the vector of components

$$\min\{\xi_i^j : j = 1, 2, \dots, r\}, \quad \text{where } i = 1, 2, \dots, r.$$

We now present the following algorithm.

Algorithm 1. The MFS Search Algorithm

Step 1: Let $\epsilon > 0$ be a small number (see Remark 1). Solve, for each $a = 1, 2, \dots, r$, the scalarized problem $P(\mathbf{e}^a)$, where \mathbf{e}^a is a vector in \mathbf{R}^r with $1 - \epsilon$ in the a^{th} entry and $\epsilon/(r - 1)$ everywhere else, to obtain the solutions $\mathbf{x}^*(\mathbf{e}^1), \mathbf{x}^*(\mathbf{e}^2), \dots, \mathbf{x}^*(\mathbf{e}^r)$.

- (i) If $\mathbf{f}(\mathbf{x}^*(\mathbf{e}^a)) \in \text{int } \mathbf{R}_+^r$ for any a , stop, no MFS exists. (See Proposition 1.)
- (ii) If $\mathbf{f}(\mathbf{x}^*(\mathbf{e}^a)) \in \mathbf{R}_-^r$ for any a , stop, $\mathbf{x}^*(\mathbf{e}^a)$ is an MFS. (See Proposition 2.)
- (iii) Otherwise let $\xi^a = \mathbf{f}(\mathbf{x}^*(\mathbf{e}^a)), a = 1, 2, \dots, r$ and $\Gamma = \text{Co}(\xi^1, \xi^2, \dots, \xi^r)$ be a simplex which is the convex hull of the efficient points $\xi^1, \xi^2, \dots, \xi^r$. Let the set of simplices to be searched be initialized as $\Omega \leftarrow \{\Gamma\}$.

Step 2: If Ω is empty, stop, no MFS exists. (See Proposition 3.) Otherwise pick any simplex $\Gamma = \text{Co}(\xi^1, \xi^2, \dots, \xi^r)$ from Ω to search for an MFS.

- (i) Check if $\Gamma \cap \mathbf{R}_-^r$ is nonempty (by solving for an auxiliary linear program, see Proposition 4). If so, stop, an MFS can be found from this nonempty intersection. (See Proposition 5.)
- (ii) If $\Gamma \cap \mathbf{R}_-^r = \emptyset$, compute the focal point to the simplex Γ . If $\text{Foc}(\Gamma) \notin \mathbf{R}_-^r$, no MFS can be found by searching from Γ (see Proposition 6), go to Step 4.
- (iii) Otherwise $\text{Foc}(\Gamma) \in \mathbf{R}_-^r$, go to Step 3 to continue search.

Step 3: Let the normal, $\lambda = \text{Nor}(\Gamma)$, to the simplex Γ be such that $[\lambda]_a > 0, \sum_{a=1}^r [\lambda]_a = 1$. Solve $P(\lambda)$ to get an efficient solution $\mathbf{x}^*(\lambda)$.

(i) If $f(x^*(\lambda))$ is the same as any of $\xi^1, \xi^2, \dots, \xi^r$, then Γ is a part of the efficient frontier and since $\Gamma \cap R'_- = \emptyset$, no MFS can be found on Γ . $\Omega \leftarrow \Omega \setminus \Gamma$ and return to Step 2.

(ii) Otherwise $f(x^*(\lambda))$ dominates some efficient point in Γ . If $f(x^*(\lambda)) \in R'_-$, stop, $x^*(\lambda)$ is an MFS. (See Proposition 8.) Otherwise $f(x^*(\lambda)) \notin R'_-$ and this new efficient point will generate r other simplices. Let

$$\begin{aligned} \Gamma_1 &= \text{Co}(f(x^*(\lambda)), \xi^2, \dots, \xi^r), \\ \Gamma_2 &= \text{Co}(\xi^1, f(x^*(\lambda)), \dots, \xi^r), \\ &\vdots \\ \Gamma_r &= \text{Co}(\xi^1, \xi^2, \dots, f(x^*(\lambda))). \end{aligned}$$

$\Omega \leftarrow \Omega \cup \{\Gamma_1, \Gamma_2, \dots, \Gamma_r\} \setminus \Gamma$ and return to Step 2.

Step 4: $\Omega \leftarrow \Omega \setminus \Gamma$, return to Step 2.

A formal justification to Algorithm 1 is provided by the following propositions.

PROPOSITION 1. *In Step 1 (i), if $f(x^*(e^a)) \in \text{int } R'_+$ for any a , then no MFS exists.*

PROOF. If an MFS x^{**} existed, it would satisfy that $f(x^{**}) \in R'_-$. Hence $f(x^{**}) < f(x^*(e^a))$ since $f(x^*(e^a)) \in \text{int } R'_+$, which contradicts the fact that $x^*(e^a)$ is an efficient solution of the MCP.

PROPOSITION 2. *In Step 1 (ii), if $f(x^*(e^a)) \in R'_-$ for some a , then $x^*(e^a)$ is an MFS.*

PROOF. $x^*(e^a)$ is an efficient solution for the MCP and lies in R'_- . By Theorem 1, it is also an MFS.

PROPOSITION 3. *If at the start of Step 2 Ω is empty, then no MFS exists.*

PROOF. If Ω is empty, the whole efficient frontier of the MCP would have been found as the union of a number of simplices. Since none of these simplices intersects R'_- , (we would have terminated with an MFS otherwise), $f(\mathcal{E}) \cap R'_- = \emptyset$ and from Theorem 1, no MFS exists.

PROPOSITION 4. *In checking for the nonempty intersection of $\Gamma \cap R'_-$ in Step 2 (i), we have: $\Gamma \cap R'_- \neq \emptyset$ if and only if $w_{\max} = 0$, which is the solution to the auxiliary linear program:*

$$\begin{aligned} \text{(LP)} \quad & \text{Maximize} \quad w = \sum_{a=1}^r q_a \\ & \text{subject to} \quad -Z^T y + q = 0, \\ & \quad \quad \quad y \geq 0, \quad q \geq 0, \end{aligned}$$

where $Z = [\xi^1; \xi^2; \dots; \xi^r] \in \mathbb{R}^{r \times r}$.

PROOF. Let $\Gamma = \text{Co}(\xi^1, \xi^2, \dots, \xi^r)$ and $Z = [\xi^1; \xi^2; \dots; \xi^r] \in \mathbb{R}^{r \times r}$. Then

- $\Gamma \cap \mathbb{R}_-^r \neq \emptyset$ iff $\exists \lambda$ such that $\sum_{a=1}^r \lambda_a \xi^a \leq 0, \sum_{a=1}^r \lambda_a = 1, \lambda_a \geq 0$;
- iff the system $\lambda > 0, Z\lambda \leq 0$ is feasible;
- iff the system $y_1 - Z^T y = 0, y_1 \geq 0, y \geq 0$ is infeasible
by the Motzkin theorem of alternative;
- iff the system $Z^T y \leq 0, y \geq 0$ is infeasible;
- iff $w_{\max} = 0$ solves the above auxiliary linear program.

PROPOSITION 5. In Step 2 (i), if $\Gamma \cap \mathbb{R}_-^r \neq \emptyset$, then an MFS can be found.

PROOF. If $\Gamma \cap \mathbb{R}_-^r \neq \emptyset$, then $\exists \eta \in \Gamma$ such that $\eta \leq 0$. If Γ is not quite part of the efficient frontier, there exists some other efficient point ξ on the efficient frontier such that $\xi \leq \eta \leq 0$. Hence ξ is an efficient point lying in \mathbb{R}_-^r . By Theorem 1, this also corresponds to an MFS. If Γ is already part of the efficient frontier, then the point in $\Gamma \cap \mathbb{R}_-^r$ is an MFS.

PROPOSITION 6. In Step 2 (ii), if $\text{Foc}(\Gamma) \notin \mathbb{R}_-^r$, then no MFS can be found by searching from Γ .

PROOF. Any efficient point which dominates points on Γ is contained in the convex hull with base Γ and vertex $\text{Foc}(\Gamma)$, that is, $\text{Co}(\xi^1, \xi^2, \dots, \xi^r, \text{Foc}(\Gamma))$, which is in turn dominated by the vertex $\text{Foc}(\Gamma)$ because of the way it is defined. If $\text{Foc}(\Gamma) \notin \mathbb{R}_-^r$, no efficient point which dominates points on Γ can be found in \mathbb{R}_-^r .

PROPOSITION 7. In Step 3, if the new efficient point $f(x^*(\lambda))$ is the same as one of the points $\xi^1, \xi^2, \dots, \xi^r$, then Γ is part of the efficient frontier of the MCP.

PROOF. If $\Gamma = \text{Co}(\xi^1, \xi^2, \dots, \xi^r)$ is not part of the efficient frontier, then there exists some point $\xi \in \Gamma$ and some efficient solution x^{**} of the MCP such that $f(x^{**}) \leq \xi$. Since $\xi \in \Gamma, \xi = \sum_{a=1}^r \mu_a \xi^a, \mu_a \geq 0, \sum_{a=1}^r \mu_a = 1$. For $\lambda = \text{Nor}(\Gamma) > 0$, then $\lambda^T \xi^{a_1} = \lambda^T \xi^{a_2}, \forall a_1, a_2$ and $\lambda^T f(x^{**}) < \lambda^T (\sum_{a=1}^r \mu_a \xi^a)$, or

$$\lambda^T f(x^{**}) < \sum_{a=1}^r \mu_a \lambda^T \xi^a = \lambda^T \xi^j, \text{ for any } j \in \{1, \dots, |A|\},$$

which contradicts the assumption that ξ^a solves $P(\lambda)$.

PROPOSITION 8. *If, in Step 3 (i), $f(x^*(\lambda)) \in R_-^r$, then $x^*(\lambda)$ is an MFS.*

PROOF. $x^*(\lambda)$ is an efficient solution of the MCP lying in R_-^r . By Theorem 1, it is an MFS.

REMARK 1. The use of ϵ in Step 1 ensures that solutions found are efficient solutions and not merely weakly efficient. The ϵ can be chosen to be small enough that the corresponding function value is the same as that obtained by solving $P(\bar{e}^\alpha)$, where \bar{e}^α is a vector in R^r with 1 in the α^{th} entry and 0 everywhere else.

REMARK 2. Note that solving the scalarized problem $P(\lambda)$ amounts to solving K decoupled single commodity linear cost flow problems:

$$\begin{array}{ll} \text{Problem } P_k(\lambda) & \text{Minimize } \sum_{a=1}^r \lambda_a f_a(x_k) = \sum_{a=1}^r \lambda_a x_{ka} \\ & \text{subject to } Ex_k = b_k, \\ & 0 \leq x_k \leq u_k, \end{array}$$

where $k = 1, \dots, K$. These can be solved very efficiently using existing scaled network simplex algorithms [7].

REMARK 3. In checking for the non-empty intersection of the simplex Γ with the negative octant (Step 2 (ii)), we need to solve an auxiliary linear program with $2r$ variables and r constraints. This is a much smaller problem than the original multiple commodity flow problem and can be solved quite efficiently. Alternatively this non-empty intersection can be checked recursively as shown in [11].

REMARK 4. In Step 3 (ii), the new efficient point generates r new simplices to be included in the set Ω for subsequent search. In practice, only a few of these new simplices need to be included since many of them would have their focal points lying outside the negative octant.

THEOREM 2. *Algorithm 1 terminates after a finite number of iterations and either gives an MFS or concludes that no MFS exists.*

PROOF. The efficient frontier for the MCP has only a finite number of simplicial faces. There is no possibility of cycling (that is, no simplex may enter the set Ω more than once) since all latter simplices dominate the one they replace. The algorithm either produces an MFS from one of the intermediate simplices, or terminates in Step 2, whence the whole efficient frontier would have been examined and Proposition 3 asserts that the efficient frontier does not intersect the negative octant.

The following examples are given to illustrate the MFS search algorithm.

EXAMPLE 4. Consider the network in Figure 1, where $K = 2$, d_a is the shared capacity of arc a , $b_1 = (5, 0, -5, 0)^T$, $b_2 = (0, 5, 0, -5)^T$ and each x_k has no upper bound, that is, each component of u_k is $+\infty$, $k = 1, 2$. Note that it is only easy to find an MFS just by inspection because of the way we have constructed the example. This is done deliberately on order to facilitate an understanding of the steps of the algorithm.

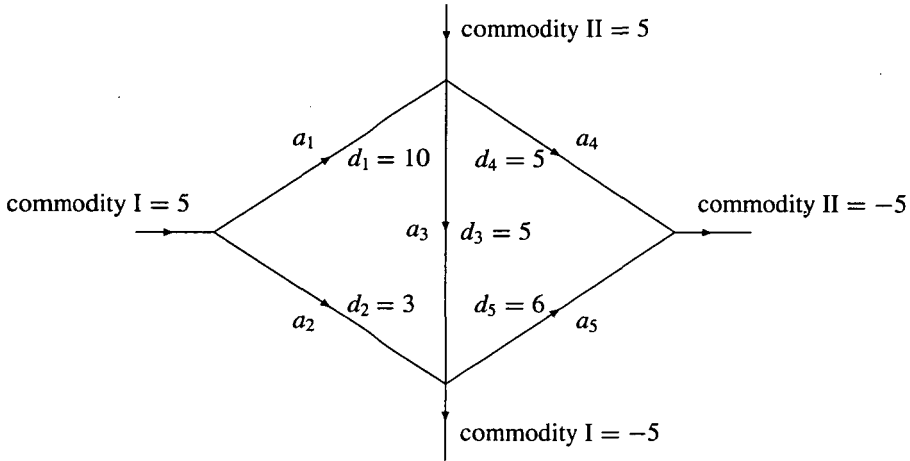


FIGURE 1. An example network.

Step 1. Solve $P(e^i)$ and let $x^*(e^i) = (x_1^i, x_2^i)$, $i = 1, \dots, 5$. We have:

$$\begin{aligned}
 x_1^1 &= (0, 5, 0, 0, 0)^T, & x_2^1 &= (0, 0, 0, 7, 0)^T, & f(x^*(e^1)) &= (-10, 2, -5, 2, -6)^T; \\
 x_1^2 &= (5, 0, 5, 0, 0)^T, & x_2^2 &= (0, 0, 0, 7, 0)^T, & f(x^*(e^2)) &= (-5, -3, 0, 2, -6)^T; \\
 x_1^3 &= (0, 5, 0, 0, 0)^T, & x_2^3 &= (0, 0, 0, 7, 0)^T, & f(x^*(e^3)) &= (-10, 2, -5, 2, -6)^T; \\
 x_1^4 &= (0, 5, 0, 0, 0)^T, & x_2^4 &= (0, 0, 7, 0, 7)^T, & f(x^*(e^4)) &= (-10, 2, 2, -5, 1)^T; \\
 x_1^5 &= (0, 5, 0, 0, 0)^T, & x_2^5 &= (0, 0, 0, 7, 0)^T, & f(x^*(e^5)) &= (-10, 2, -5, 2, -6)^T.
 \end{aligned}$$

Check (i) No $f(x^*(e^i)) \in \text{int } R_+^5$; (ii) no $f(x^*(e^i)) \in \text{int } R_-^5$; (iii) since $f(x^*(e^1)) = f(x^*(e^3)) = f(x^*(e^5))$, let

$$\begin{aligned}
 \xi^1 &= (-10, 2, -5, 2, -6)^T, & \xi^2 &= (-5, -3, 0, 2, -6)^T, & \xi^3 &= (-10, 2, 2, -5, 1), \\
 \Gamma &= \text{Co}\{\xi^1, \xi^2, \xi^3\}, & \Omega &\leftarrow \{\Gamma\}.
 \end{aligned}$$

Note that the simplex Γ is degenerate but that this does not create any problem for the algorithm.

Step 2. Check if $\Gamma \cap R_-^5 \neq \emptyset$ to find a point $(-8, 0, -1, 0, -4) = 0.3143\xi^1 + 0.4\xi^2 + 0.2857\xi^3$, with corresponding MFS flow $x_1 = (2, 3, 2, 0, 0)^T, x_2 = (0, 0, 2, 5, 2)^T$. Terminate Algorithm 1.

EXAMPLE 5. Consider the same network as in Example 4, but change d_3 to $d_3 = 3$.

Step 1. Solving $P(e^1), P(e^3), P(e^5)$ gives $f(x^*(e^1)) = (-10, 2, -3, 2, -6)^T$, solving $P(e^2)$ gives $f(x^*(e^2)) = (-5, -3, 2, 2, -6)^T$ and solving $P(e^4)$ gives $f(x^*(e^4)) = (-10, 2, 4, -5, 1)^T$. Check (i) No $f(x^*(e^i)) \in \text{int } R_+^5$; (ii) no $f(x^*(e^i)) \in \text{int } R_-^5$; (iii) let $\Gamma = \text{Co}\{\xi^1, \xi^2, \xi^3\}$.

Step 2. Check that $\Gamma \cap R_-^5 = \emptyset$ by solving the auxiliary linear program.

Step 3. A normal to Γ is $\lambda = (0.5, 0.5, 0, 0, 0)^T$. Solving $P(\lambda)$ gives

$$x_1 = (0, 5, 0, 0, 0)^T, x_2 = (0, 0, 0, 7, 0)^T, f(x^*(\lambda)) = (-10, 2, -3, 2, -6)^T = \xi^1.$$

So Γ is a part of the efficient frontier. Let $\Omega \leftarrow \{\Gamma\} \setminus \{\Gamma\}$. Return to Step 2.

Step 2. $\Omega = \emptyset$, stop. No MFS can be found.

5. An algorithm for solving the multiple criteria linear program

Consider the multiple criteria linear program with general linear constraints:

$$\begin{aligned} \text{(MCLP)} \quad & \text{V-Minimize} \quad (c_1^T x, \dots, c_p^T x) \\ & \text{subject to} \quad Ax = b, \\ & \quad \quad \quad 0 \leq x \leq u, \end{aligned}$$

where $A_{m \times n}$ is a general $m \times n$ matrix, $x, c_1, \dots, c_p, u \in R^n, b \in R^m$. We seek to find the whole efficient frontier to the problem.

Let $\lambda \in R^p$ and $\lambda \geq 0$. The scalarized problem of the MCLP is defined as follows:

$$\begin{aligned} \text{Problem } Q(\lambda) \quad & \text{Minimize} \quad \sum_{i=1}^p \lambda_i c_i^T x \\ & \text{subject to} \quad Ax = b, \\ & \quad \quad \quad 0 \leq x \leq u. \end{aligned}$$

Aneja and Nair [1] previously proposed a method for finding the efficient frontier of a bicriteria linear cost network flow problem. The idea is based on the fact that the efficient frontier for a bicriteria linear cost problem is a piecewise affine curve in two dimensional criteria space. To begin, the efficient frontier is first approximated by a single straight line joining two points obtained by solving for each of the criteria. Clearly this line is dominated by the efficient frontier, if it is not already the frontier itself. Hence by solving the scalarized problem $Q(\lambda)$ with λ corresponding to the

normal to this straight line, a point on the efficient frontier can be found. Thus an improved approximation to the efficient frontier is given by the two linear segments with this new point as the break point. Continuing in this manner, the efficient frontier can be determined after a finite number of iterations. It is quite clear that this method is also applicable to the case of a bicriteria linear program, which is a special case of the MCLP with $p = 2$. It is not so obvious, however, that this can also be extended to find the efficient frontier of the MCLP and this extension will be discussed shortly. This is a viable and practical alternative to the conceptual algorithm presented in Yu [12].

The idea here is based on the fact that the efficient frontier to the MCLP is a union of a number of simplicial faces, each being a $p - 1$ simplex in the p dimensional criteria space. We specialize the search algorithm presented in the previous section to find these simplicial faces iteratively.

Algorithm 2. Finding the efficient frontier of the MCLP

Step 1: Let $\epsilon > 0$ be a small number. Solve, for each $i = 1, 2, \dots, p$, the scalarized problem $P(e^i)$, where e^i is a vector in R^p with $1 - \epsilon$ in the i^{th} entry and $\epsilon/(p - 1)$ everywhere else, to obtain the solutions $x^*(e^1), x^*(e^2), \dots, x^*(e^p)$. Let $\xi^i = f(x^*(e^i)), i = 1, 2, \dots, p$ and $\Gamma = \text{Co}(\xi^1, \xi^2, \dots, \xi^p)$ be a simplex. Let the set of simplices to be searched be initialized as $\Omega \leftarrow \{\Gamma\}$ and the set of simplices whose union gives the efficient frontier be initialized as $\Pi \leftarrow \emptyset$.

Step 2: If Ω is empty, stop, the efficient frontier to the MCLP is given by the union of the simplices in Π . Otherwise pick any simplex $\Gamma = \text{Co}(\xi^1, \xi^2, \dots, \xi^p)$ from Ω .

Step 3: Let the normal, $\lambda = \text{Nor}(\Gamma)$, to the simplex Γ be such that $[\lambda]_a > 0, \sum_{a=1}^r [\lambda]_a = 1$. Solve $Q(\lambda)$ to get an efficient solution $x^*(\lambda)$.

(i) If $f(x^*(\lambda))$ is the same as any of $\xi^1, \xi^2, \dots, \xi^p$, then Γ is a simplicial face of the efficient frontier. $\Omega \leftarrow \Omega \setminus \Gamma$ and $\Pi \leftarrow \Pi \cup \Gamma$, return to Step 2.

(ii) Otherwise $f(x^*(\lambda))$ will generate p other simplices. Let

$$\begin{aligned} \Gamma_1 &= \text{Co}(f(x^*(\lambda)), \xi^2, \dots, \xi^p), \\ \Gamma_2 &= \text{Co}(\xi^1, f(x^*(\lambda)), \dots, \xi^p), \\ &\vdots \\ \Gamma_p &= \text{Co}(\xi^1, \xi^2, \dots, f(x^*(\lambda))). \end{aligned}$$

$\Omega \leftarrow \Omega \cup \{\Gamma_1, \Gamma_2, \dots, \Gamma_p\} \setminus \Gamma$ and return to Step 2.

The justification of Algorithm 2 is essentially the same as that for Algorithm 1 and the proof of the following theorem follows from that of Theorem 2.

THEOREM 3. *Algorithm 2 will generate the efficient frontier of the MCLP after a finite number of iterations.*

6. Concluding remarks

In this paper, it is shown that several special linear cost network problems can be reformulated as a general multiple criteria linear cost network flow problem. Based on this multiple criteria model, a search algorithm for finding a feasible solution to the multiple commodity flow problem is suggested. Numerical examples are given to justify the algorithm. The MFS search algorithm can also be modified to give a new algorithm for finding the efficient frontier of a multiple criteria linear program.

References

- [1] Y. P. Aneja and K. P. K. Nair, "Bicriteria transportation problem", *Management Science* **25** (1979) 73–78.
- [2] G. B. Dantzig, *Linear Programming and Extensions* (Princeton University Press, Princeton, NJ, 1963).
- [3] J. R. Evans and K. Martin, "A note on feasible flows in lower-bounded multicommodity networks", *J. Oper. Res. Soc.* **29** (1978) 923–927.
- [4] A. M. Geoffrion, "Strictly concave parametric programming, Part I: Basic theory", *Management Science* **13** (1966) 244–253.
- [5] C. J. Goh and X. Q. Yang, "Analytic efficient solution set for multi-criteria quadratic program", *European J. Oper. Res.* **89** (1996) 483–491.
- [6] M. Gondran and M. Minoux, *Graphs and Algorithms* (Wiley-Interscience, New York, 1984).
- [7] J. L. Kennington and R. V. Helgason, *Algorithms for Network Programming* (Wiley, New York, 1980).
- [8] G. Ruhe, *Algorithmic Aspects of Flows in Networks* (Kluwer Academic Publishers, London, 1991).
- [9] Y. Sawaragi, H. Nakayama and T. Tanino, *Theory of Multi-Objective Optimization* (Academic Press, New York, 1985).
- [10] C. T. Tung and K. L. Chew, "A multicriteria pareto-optimal path algorithm", *European J. Oper. Res.* **62** (1992) 203–209.
- [11] X. Q. Yang and C. J. Goh, "On intersections of two particular convex sets", *J. Optimiz. Theory Appl.* **89** (1996) 483–491.
- [12] P. L. Yu, *Multiple-Criteria Decision Making - Concepts, Techniques, and Extensions* (Plenum Press, 1985).
- [13] P. L. Yu and M. Zeleny, "Linear multiparametric programming by multicriteria simplex methods", *Management Science* **23** (1976) 159–170.