Y. P. ANEJA

K. P. K. NAIR

Two problems in multicommodity networks

RAIRO. Recherche opérationnelle, tome 13, nº 2 (1979), p. 135-142

<http://www.numdam.org/item?id=RO_1979__13_2_135_0>

R.A.I.R.O. Recherche opérationnelle/Operations Research
(vol. 13, n° 2, mai 1979, p. 135 à 142)

TWO PROBLEMS IN MULTICOMMODITY NETWORKS (*)

par Y. P. ANEJA (1) and K. P. K. NAIR (1)

Abstract. — *The problem of maximizing the sum of flows in a multicommodity network has been studied by Ford and Fulkerson, and they provided an algorithm, involving column generation, for computing the maximal flows. In this paper, considering a finite traversal time for each commodity along each arc, two problems of interest are studied. In the first problem the objective is one of maximizing the sum of flows but subject to additional constraints of the form that the maximal traversal time for each commodity must not exceed a specified limit. In the second problem the objective is to minimize the maximum of the traversal times for the commodities subject to meeting the specified requirement of flows. Introducing the concept of constrained shortest path generation, algorithms are presented for computing the solution in each of the problems.*

Résumé. — *Le problème de trouver le maximum de la somme des écoulements dans un réseau de plusieurs produits a été étudié par Ford et Fulkerson. Ils ont trouvé un algorithme, pour calculer les écoulements maximaux, qui emploie une génération de colonnes. Dans cet article nous considérons un temps fini de traversée pour chaque produit le long de chaque arc. Deux problèmes sont étudiés. Dans le premier, l'objectif est de maximiser la somme des écoulements sous les contraintes de telle sorte que le temps de traversée maximal pour chaque produit n'excède pas une limite spécifiée. Dans le second problème, on minimise le maximum des temps de traversée pour les produits sous les contraintes de telle sorte que les flots de chaque produit soient supérieurs ou égaux à des quantités données. Pour appliquer les algorithmes de résolution de ces problèmes il convient de résoudre le problème suivant (plus courts chemins contraints) : aux arcs d'un réseau numérotés $i=1, 2, \ldots, m$ on associe 2 nombres h_i et g_i. Il s'agit de trouver un plus court chemin d'un sommet s à un sommet p (les longueurs des arcs étant les h_i) sous la contrainte que la longueur de ce chemin (si on prend pour longueur des arcs les g_i) soit inférieure ou égale à un nombre B donné.*

1. INTRODUCTION

The study of multicommodity flows in networks is of special interest, particularly because of its application in the areas of transportation and communication. Ford and Fulkerson [2] provided an efficient algorithm for computing the maximal sum of the flows in a network. Their algorithm is based on the idea of a column generation, and indeed as a consequence the algorithm is suitable even in large problems. In this paper we consider a finite traversal time

(*) Reçu janvier 1977.
(1) School of Administration, University of New Brunswick, Fredericton, N.B., E3B 5A3, Canada.

for each commodity along each arc, and in general, along an arc these could be different for the various commodities. Associated with each commodity we have a set of chains that can be used for implementing its flow from source to sink. For a commodity the traversal time along a chain is the sum of the traversal times for this commodity along those arcs constituting this chain. In certain situations such as in the transportation of perishable or deteriorating goods, the maximum traversal time for each commodity in the network may have to be restricted. In this context we consider two problems. The first is one of maximizing the sum of flows subject to restrictions on the allowable traversal time for each commodity in the network. In the second problem we minimize the maximum of the traversal times for the commodities subject to specified requirement of the flows. Developing a technique for generating a constrained column both the problems are solved, and the methods are applicable even in large problems.

2. FORMULATION OF THE PROBLEMS

Consider a multicommodity network whose arcs are numbered $1, 2, \ldots, i, \ldots, m$ and let the commodities be denoted by the set $Q = \{1, 2, \ldots, q, \ldots, K\}$. A chain j is associated with one of the commodities and let $C^q\,(q \in Q)$ be the set of all the chains associated with the commodity q. The set of all possible chains in the network is denoted by C so that

$$C = \bigcup_{q \in Q} C^q \tag{1}$$

Setting $A = \{a_{ij}\}$ to be the incidence matrix of the network

$$a_{ij} = \left\{ \begin{array}{ll} 1 & \text{if chain } j \text{ includes arc } i, \\ & 0 \quad \text{otherwise.} \end{array} \right\} \tag{2}$$

Let the traversal time for commodity q on arc i be $t_i^q\,(i=1, 2, \ldots, m;$ $q=1, 2, \ldots, K)$. Define $F^q\,(q=1, 2, \ldots, K)$ to be a subset of C^q such that

$$F^q = \left\{ j \,\middle|\, j \in C^q \text{ and } \sum_{i=1}^{m} a_{ij}\, t_i^q \leqq d^q,\ q=1, 2, \ldots, K \right\}, \tag{3}$$

where d^q is the maximum allowable traversal time for commodity q from its source to sink. The set of all chains in the network given by (3) is denoted by $F\,(F \subset C)$ such that

$$F = \bigcup_{q \in Q} F^q. \tag{4}$$

PROBLEM I: Maximize

$$\sum_{j\in F} x_j. \tag{5}$$

Subject to

$$\sum_{j\in F} a_{ij} x_j \leqq b_i, \qquad i=1, 2, \ldots, m. \tag{6}$$

$$x_j \geqq 0, \qquad j\in F, \tag{7}$$

where x_j = the amount of flow in chain $j\in F$ and b_i = capacity of arc $i(i=1, 2, \ldots, m)$.

The above formulation differs from that of Ford and Fulkerson [2] with respect to the admissibility of the chains. While their formulation admits all the chains contained in the set C, the present formulation allows only those chains in the set $F\,(F\subset C)$. The constraint on the admissible set of chains results from the fact the maximum allowable transversal time from source to sink for each commodity is specified to be $d^q\,(q\in Q)$. Therefore, the column generation technique of Ford and Fulkerson [2] requires some modifications for solving this problem. A constrained shortest chain generation as detailed in Section 4 allows this problem to be solved efficiently. An algorithm based on this is presented in the next section.

PROBLEM II: Minimize

$$T=\operatorname*{Max}_{q\in Q}\ \operatorname*{Max}_{\substack{j\in C^q\\ x_j>0}}\ \sum_{i=1}^{m} a_{ij}\, t_i^q. \tag{8}$$

Subject to

$$\sum_{j\in C} a_{ij} x_j \leqq b_i, \qquad i=1, 2, \ldots, m, \tag{9}$$

$$\sum_{j\in C^q} x_j \geqq \sigma^q, \qquad q=1, 2, \ldots, K, \tag{10}$$

$$x_j \geqq 0, \qquad j\in C, \tag{11}$$

where x_j = amount of flow in chain $j\,(j\in C)$ and σ^q = required flow of commodity $q\,(q\in Q)$.

The objective function (8) is one of bottleneck type as considered by Frieze [3] in the case of general linear programming. However, his method is not directly applicable to the present problem since it is not pratical to enumerate all the chains given by the set C. But the technique of constrained column generation detailed in Section 4 allows also this problem to be solved without enumerating all the chains, and an algorithm is given in the next section.

3. ALGORITHMS

Algorithms I and II, respectively, presented in this section are applicable to Problems I and II above. Algorithm I is similar to that of Ford and Fulkerson [2] except in Step 3. Algorithm II is based on the one given by Frieze [3] but incorporates a constrained column generation. Essentially this algorithm performs phase *I* of the simplex method, in such a way that, at each step, the chain generated for entering in to the basis has the minimum time among all the chains which can reduce infeasibility.

Algorithm I

STEP (0): Start the simplex method with an initial basis having columns which correspond to all the slacks introduced to convert the problem into an equality.

STEP (1): For a given basis B at the current iteration determine the simplex multipliers $\pi_1, \pi_2, \ldots, \pi_m$.

STEP (2): If $\pi_i < 0$ for some i, introduce the corresponding slack in to the basis, and go to step (1). Otherwise go to step (3).

STEP (3): For each commodity q, $q = 1, 2, \ldots, K$, determine a 0-1 vector $(y_1^q, y_2^q, \ldots, y_m^q)$ representing a chain such that

$$z_q = \text{Min} \sum_{i=1}^{m} \pi_i y_i^q,$$

$$\text{s.t.} \sum_{i=1}^{m} t_i^q y_i^q \leq d^q.$$

Let $z_{q^*} = \text{Min}\{z_1, z_2, \ldots, z_K\}$.

STEP (4): If $z_{q^*} < 1$, then introduce the chain $(y_1^{q^*}, y_2^{q^*}, \ldots, y_m^{q^*})$ in to the basis and go to step (1). Otherwise terminate. The current basis yields an optimal solution to the problem.

Algorithm II

STEP (0): Start phase *I* of the simplex algorithm with an initial basis consisting of columns corresponding to slacks introduced in to the first m constraints and artificial variable introduced in the last K constraints.

STEP (1): For the basis B at the current iteration determine the simplex multipliers $(\pi_1, \pi_2, \ldots, \pi_m, \alpha_1, \alpha_2, \ldots, \alpha_K)$. If all the artifical variables in the solution corresponding to the current basis B are equal to zero, go to step (5). Otherwise go to step (2).

STEP (2): If any $\pi_i < 0$ for $i = 1, 2, \ldots, m$ or $\alpha_q > 0$ for $q = 1, \ldots, k$; introduce the corresponding slack in to the basis, and go to step (1). Otherwise go to step (3).

STEP (3): For each commodity q, $q = 1, 2, \ldots, K$ determine a 0-1 vector $(y_1^q, y_2^q, \ldots, y_m^q)$ representing a chain such that

$$z_q = \text{Min} \sum_{i=1}^{m} t_i^q \cdot y_i^q$$

$$\text{s.t.} \sum_{i=1}^{m} \pi_i \cdot y_i^q < 1 - \alpha_q.$$

Let $z_q = \infty$ if the above problem does not have a feasible solution. Let $z_{q^*} = \text{Min} \{ z_1, z_2, \ldots, z_K \}$ and go to step (4).

STEP (4): If $z_{q^*} = \infty$, terminate since the problem does not have a feasible solution. Otherwise introduce $(y_1^{q^*}, y_2^{q^*}, \ldots, y_m^{q^*})$ in to the basis and go to step (1).

STEP (5): Let $T^* = \underset{x_j > 0}{\text{Max}} \sum_i a_{ij} t_i^q$. Then T^* is the value of the optimal solution to problem II.

The validity of Algorithm I and II follows from the works of Ford and Fulkerson [2] and Frieze [3] respectively. It is of interest to note that step 3 in each of the algorithms is similar, and involves, basically, generating a constrained shortest chain from source to sink.

4. CONSTRAINED SHORTEST CHAIN

Let $G[N; A]$ represent a general network with node set N with two distinguished nodes s and t, $s \neq t$, and arc set A. Let $a(x, y) \geqq 0$ and $b(x, y) \geqq 0$ be two numbers associated with each arc $(x, y) \in A$. The constrained shortest chain problem can be defined as:

Minimize

$$\sum_{(x,\, y) \in A} a(x, y) \cdot f(x, y)$$

$$\text{s.t.} \sum_{y \in N} f(x, y) - \sum_{y \in N} f(y, x) = \begin{cases} 1 & \text{if } x = s, \\ -1 & \text{if } x = t, \\ 0 & \text{otherwise,} \end{cases}$$

$$\text{(P)} \qquad \sum_{(x,\, y) \in A} b(x, y) . f(x, y) \leqq B,$$

$$f(x, y) = 0 \quad \text{or} \quad 1,$$

for some specified positive number B.

For any chain from source to any given node y, we can define two numbers $\alpha_1(y)$ and $\alpha_2(y)$ where $\alpha_1(y)$ is the a-value of the chain and $\alpha_2(y)$ is the b-value of the chain. Consider a chain from s to y with values $\alpha_1^0(y)$ and $\alpha_2^0(y)$ such that $\alpha_2^0(y) \leqq B$. Then such a chain is said to be non-dominated if for any other feasible chain (satisfying the constraint) with values $\alpha_1(y)$ and $\alpha_2(y)$:

$$\alpha_1(y) < \alpha_1^0(y) \quad \Rightarrow \quad \alpha_2(y) > \alpha_2^0(y)$$

and

$$\alpha_2(y) < \alpha_2^0(y) \quad \Rightarrow \quad \alpha_1(y) > \alpha_1^0(y).$$

The algorithm presented below develops a multiple labelling scheme for determining a non-dominated chain from s to t, if one such exists, with minimum a-value. During the steps of the algorithm, a node may acquire none, one or more labels. The source, however, would have only one label.

The r-th label at node y, denoted by $\theta^r(y)$ has the structure: $\theta^r(y) = [x_l, \alpha^r(y)]$ where $\alpha^r(y) = (\alpha_1^r(y), \alpha_2^r(y))$ and x_l is the index which tells that $\alpha^r(y)$ is obtained from l-th label at node x, using the relation $\alpha^r(y) = \alpha^l(x) + \gamma(x, y)$. Here $\gamma(x, y) = (a(x, y), b(x, y))$.

To begin with there is no label at any node. Every label assigned is initially tentative, but may subsequently become permanent. Let $L(x)$ denote the set of tentative labels at node x and set $L = \bigcup_{x \in N} L(x)$. Similarly $P = \bigcup_{x \in N} P(x)$ where $P(x)$ is the set of permanent labels at node x.

Algorithm III

STEP (0): Assign $\theta^1(s) = [-;\, \alpha^1(s)]$ where $\alpha^1(s) = (0, 0)$. Let $L(s) = \{\theta^1(s)\}$, $L(x) = \emptyset$ for $x \neq s$. Set $P = \emptyset$.

STEP (1): If $L = \emptyset$, the problem is infeasible and therefore stop; otherwise go to step (2).

STEP (2): In the set L, find a label $\theta^l(k)$ such that $\alpha_1^l(k) \leqq \alpha_1^r(x)$ for all $\theta^r(x) \in L$. If there are more than one label satisfying this, choose $\theta^l(k)$ to be the one with least value of $\alpha_2(\quad)$. Set $L(k) = L(k) - \{\theta^l(k)\}$ and $P(k) = P(k) \cup \{\theta^l(k)\}$. If $k = t$, the optimal solution is obtained and therefore stop; otherwise go to step (3).

STEP (3): Identify the set of arcs $A^l(k)=\{(k, y) \mid \alpha_2^l(k)+b(k, y) \leqq B\}$. If $A^l(k)=\emptyset$, go to step (1), otherwise go to step (4).

STEP (4): for $(k, y) \in A^l(k)$, compute the vector $v=\alpha^l(k)+\gamma(k, y)$. If there is no label at node y, then insert the label $\theta^1(y)=[k_l, v]$ and set $L(y)=\{\theta^1(y)\}$. Otherwise check whether or not $v \geqq \alpha^r(y)$ for any r. If "yes" discard the vector computed; otherwise insert the label $\theta^q(y)$ [where q is different from any r given $\theta^r(y) \in L(y)$] so that $\theta^q(y)=[k_l, v]$ and set $L(y)=L(y) \cup \{\theta^q(y)\}$. Identify the set $D(y)=\{\theta^r(y) \in L(y) \mid \alpha^r(y) \geqq \alpha^q(y)\}$. Set $L(y)=L(y)-D(y)$. Repeat this step for each $(k, y) \in A^l(k)$ and go to step (1).

5. VALIDITY OF THE ALGORITHM

The proof of the validity of the algorithm is shown by the following two lemmas.

LEMMA 1: *Each permanent label at a node represents a non-dominated chain from source to that node.*

Proof: The proof is by induction. Clearly the first permanent label does represent a non-dominated chain. Suppose that at some stage the labels are divided into two sets – Set A containing all permanent labels and set B containing all tentative labels. Assume that labels of set A represent non-dominated chains from source to their respective nodes. By construction, any label, say at node y, in the set B denotes a chain from s to y and this chain is revealed by a sequence of labels from source to node y; and further all these labels, except at y, are contained in set A. Let $S(y)$ be the set of such chains; one chain for each label of $L(y)$. Every chain in the set $S(y)$ is non-dominated among chains of $S(y)$. Then the algorithm makes the tentative label $\theta^r(y)$, at node y, permanent if it has the minimum a-value among all tentative labels. This label must represent a non-dominated chain. For, if there exists a chain which dominates this chain to y, then it must be realized by a sequence of labels from s to y where at least the last two labels are from set B. Since all a's and b's are non-negative, and $\theta^r(y)$ is the label with minimum $\alpha_1(\quad)$ value, this can not happen.

LEMMA 2: *The first permanent label at any node represents a non-dominated chain from source to that node with minimum a-value.*

Proof: The proof is by induction again. At some iteration of the algorithm, let the nodes of the sets be divided in two sets A and B. Set A containing all nodes which have at least one permanent label and set B containing nodes which have only tentative labels associated with them. By lemma 1, permanent labels represent non-dominated chains. Assume that the first permanent label at any node in set A represents a non-dominated chain with minimum a-value. Assume also that at this stage a node y of set B will be labeled permanent, with the label $\theta^r(y)$. By construction, for any node in set B, the label with minimum a-value represents a shortest feasible chain from source to that node among all chains where all except the terminal node are from set A. Since a's and b's are non-negative, $\theta^r(y)$ must represent a non-dominated chain with minimum a-value.

The solution obtained by algorithm III satisfies two properties. It is an optimal solution to (P) and no other optimal solution to (P) has lesser constraint values.

For Algorithm I it is important to note that in step (3) one does not require determining a shortest constrained chain. Any feasible chain (satisfying the constraint) with length less than 1 would be a candidate for entering into the basis. Algorithm III can be modified to achieve this leading to computational advantage.

REFERENCES

1. E. W. Dijkstra, *A Note on Two Problems in Connection with Graphs*, Numerische Mathematik, Vol. 1, 1959, pp. 269-271.
2. L. R. Ford and D. R. Fulkerson, *A Suggested Computation for Maximal Multicommodity Network Flows*, Management Science, Vol. 5, 1958, pp. 97-101.
3. A. M. Frieze, *Bottleneck Linear Programming*, Operational Research Quaterly, Vol. 26, 1975, pp. 871-874.