RAIRO. RECHERCHE OPÉRATIONNELLE

V. AGGARWAL
Y. P. ANEJA
K. P. K. NAIR

Ratio rewards in networks

RAIRO. Recherche opérationnelle, tome 15, nº 2 (1981), p. 129-138
<http://www.numdam.org/item?id=RO_1981__15_2_129_0>

R.A.I.R.O. Recherche opérationnelle/Operations Research
(vol. 15, n° 2, mai 1981, p. 129 à 138)

RATIO REWARDS IN NETWORKS (*)

by V. AGGARWAL, Y. P. ANEJA and K. P. K. NAIR (1)

Abstract. — *An infinite horizon network flow problem is considered where the objective is to maximize the ratio of two types of discounted rewards which are proportional to the flows over the corresponding arcs. Equivalence of this problem is shown to a special case of Markov ratio decision problem and to another network problem with constant gains in which ratio of rewards is being maximized. The simplex method is specialized to exploit the special structure of the later problem.*

Keywords: Network flows, ratio rewards, Markov ratio decision problems, simplex methods.

Résumé. — *Cet article porte sur un problème de réseau avec flot à horizon infini où la fonction objective est de maximiser le rapport de deux types de récompenses escomptées. Ces récompenses sont proportionnelles aux flots sur les arcs correspondants. On démontre que ce problème est équivalent à une classe spéciale de problèmes de décision proportionnelle de Markov dans lesquels des matrices de transition à chaque étape doivent être déterminées, et à un problème de réseau avec gains constants où le rapport des récompenses est maximisé. La méthode du simplexe est spécialisée afin d'exploiter la structure particulière de ce dernier problème équivalent.*

Mots clés : Réseau avec flot, rapport récompenses, problèmes de décision proportionnelle de Markov, méthode du simplexe.

1. INTRODUCTION

There are several reasons for examining special structures in mathematical programming. Extensions to more general optimization problems at a marginal expense of computational work, if can be accomplished, can lead to much wider use of the algorithms. Also the investigation of problems with special structures provides deeper insight into the problem itself.

This paper presents an extension of the special class of network with gains problems arising in discounted deterministic Markov decision models, studied by Dirickx and Rao [3] in the ratio reward context. It is seen that this problem is actually an infinite-horizon finite-state network flow problem in which the flow from the sources has to be distributed so as to maximize the ratio of two types of discounted rewards which are proportional to the flows over the corresponding arcs.

(*) Received in June 1979.
This work was supported in part under N.S.E.R.C. Grant No. A 3368.
(1) School of Administration, University of New Brunswick, Fredericton, N. B. Canada E3B 5A3.

R.A.I.R.O. Recherche opérationnelle/Operations Research, 0399-0559/1981/129/$ 5.00

This problem, in turn, is shown to be equivalent to a special case of Markov ratio decision problem of Aggarwal, Chandrasekran and Nair [1] in which for each alternative action in a state the process moves to a state at the next step deterministically. The policy iteration scheme developed in [1] could be specialised to derive a solution of the above program. However this paper employs the simplex method to exploit the one-tree structure of this problem.

2. STATEMENT OF THE PROBLEM

Consider a directed network (N, A) with node set N and arc set A. This network is viewed over an infinite horizon at time periods $n=0, 1, 2, \ldots$ For each directed arc (i, j) there are two numbers $c_{ij}, d_{ij}>0$ and given the flow at time n is $f_{ij}(n)$, the associated reward and cost, respectively, are given by $c_{ij} f_{ij}(n)$ and $d_{ij} f_{ij}(n)$. Both the rewards and costs occurring in future periods are discounted by a factor α, $0<\alpha<1$ per period. Let each of the starting nodes $i \in N$ be sources so that at the time period $n=0$, the flows generated through the respective sources are $a_i>0$ $(i=1, \ldots, N)$.

Letting $f_{ij}(n)$ be the flow in arc (i, j) at time period n, the decision problem can be stated as follows:

$$\text{Max} \frac{\sum_{n=0}^{\infty} \sum_{i \in N} \sum_{j \in N} \alpha^n c_{ij} f_{ij}(n)}{\sum_{n=0}^{\infty} \sum_{i \in N} \sum_{j \in N} \alpha^n d_{ij} f_{ij}(n)}, \tag{1}$$

$$\text{S.t.} \sum_{j=1}^{N} f_{ij}(0)=a_i, \qquad i \in N, \tag{2}$$

$$\sum_{j=1}^{N} f_{ij}(n)=\sum_{j=1}^{N} f_{ji}(n-1), \qquad n=1, 2, \ldots \tag{3}$$

$$f_{ij}(n) \geqq 0, \qquad i, j \in N, \quad n=0, 1, 2, \ldots \tag{4}$$

The above network problem as stated has a close resemblance to the Markov Ratio Decision problem [1] of a special kind in which for each alternative action in a state the process makes a transition deterministically. The node set N then corresponds to the state space and an element (i, j) of the arc set A corresponds to being in state i and selecting an alternative that causes a transition to state j at the next epoch. The decision variables, $f_{ij}(n)$ $(i, j \in N, n=0, 1, \ldots)$ represent the joint probabilities of being in state i and making a transition to state j at epoch n. The constants a_i $(i=1, \ldots, N)$ normalized as $a_i = a_i / \sum_{i=1}^{N} a'_i \, (i=1, \ldots, N)$ correspond to the initial probability of being in the respective starting states $i \in N$.

The problem (1)-(4) can be reduced to a program with a finite number of variables using the following transformation:

$$x_{ij} = \sum_{n=0}^{\infty} \alpha^n f_{ij}(n), \qquad i, j \in N. \tag{5}$$

The transformation is valid since $f_{ij}(n)$ are finite being bounded by $\sum_{i \in N} a_i$ and $0<\alpha<1$. x_{ij} could be considered to represent an equivalent discounted flow between nodes i and j over the reduced network. The program, therefore, becomes:

$$\text{Max} \frac{\sum_i \sum_j c_{ij} x_{ij}}{\sum_i \sum_j d_{ij} x_{ij}}, \tag{6}$$

$$\text{S.t.} \sum_j x_{ij} - \alpha \sum_j x_{ji} = a_i, \qquad i \in N, \tag{7}$$

$$x_{ij} \geqq 0, \qquad i, j \in N. \tag{8}$$

In the formulation (6) through (8) the feasible set is indentical to that in a network flow problem with gains [4]. But the objective in the current formulation is a ratio of two linear functions while that in all gain problems treated in the literature [4] is a linear function. Therefore available algorithms [4] for the gain problem cannot be directly applied to the present problem. Thus the approach considered here consists of transforming (6) through (8) to a linear program using the method of Charnes and Cooper [2] and exploiting the one-tree structure of the solution. Considering the relationship of the problem to the special case of the Markov ratio decision process, one has an obvious lemma as stated below.

LEMMA 1: *If x^*_{ij} is a solution to* (6)-(8), *then there is also a stationary solution to* (1)-(4) *given by* $f^*_{ij} = (1-\alpha) x^*_{ij}$.

3. THE STRUCTURE OF THE SOLUTION

The fractional L. P. (6)-(8) can be reduced to a standard L. P. using the method of Charnes and Cooper [2]. Define the variables,

$$y=\frac{1}{\sum_i \sum_j d_{ij} x_{ij}}>0, \tag{9}$$

and:

$$y_{ij}=x_{ij}.y \geqq 0, \qquad i, j \in N. \tag{10}$$

Hence follows the following problem denoted by (P):

$$\operatorname{Max} \sum_i \sum_j c_{ij} y_{ij}, \tag{11}$$

$$\text{S.t.} \sum_{j=1}^{N} y_{ij}-\alpha \sum_{j=1}^{N} y_{ji}-a_i y=0, \qquad \forall i, \tag{12}$$

$$\sum_{i=1}^{N} \sum_{j=1}^{N} d_{ij} y_{ij}=1, \tag{13}$$

$$y,\ y_{ij} \geqq 0, \qquad i, j \in N. \tag{14}$$

LEMMA 2: *In any basic feasible solution to* (P), *y appears at a positive level.*

Proof: Suppose $y=0$ in a basic feasible solution. Then constraints (12) become:

$$\sum_{j=1}^{N} y_{ij}-\alpha \sum_{j=1}^{N} y_{ji}=0, \qquad \forall i.$$

Summing up over all i, we get:

$$(1-\alpha) \sum_{i=1}^{N} \sum_{j=1}^{N} y_{ij}=0,$$

i.e. $\sum_i \sum_j y_{ij}=0$, contradicting constraint (13).

LEMMA 3: *In any basic feasible solution to* (P) *exactly one* $y_{ij}>0$ *for each* $i \in N$.

Proof: We first show that at least one $y_{ij}>0$ for each $i \in N$. Suppose $y_{i_0 j}=0, \forall j$ for some i_0. Then constraint (12) for $i=i_0$ becomes:

$$-\alpha \sum_{j=1}^{N} y_{j i_0}=a_{i_0}.y>0, \quad \text{a contradiction.}$$

Since there are total $N+1$ constraints, we have at most $N+1$ variables at positive level in any basic feasible solution. Thus there is exactly one $y_{ij}>0$ for each $i \in N$.

COROLLARY: *Any basic feasible solution to* (P) *is nondegenerate.*

The interpretation of an optimal basic feasible solution is clear. If $y_{ij}>0$ in an optimal basic solution, then arc (i, j) has to be chosen whenever state i occurs.

DEFINITION: A *one-tree* of a finite directed network (N, A) is defined as $U=(C, T, E)$ with:

$$C=\{i_1, i_2, \ldots, i_m\}, \qquad T=\{j_1, j_2, \ldots, j_n\}$$

$C, T \subseteqq N$, $C \cap T = \emptyset$ and:

$$E \subseteqq \{(i, j) \mid (i, j) \in A \text{ and } i, j \in C \cup T\}$$

such that:

(i) $E_c=\{(i, j) \in E \mid (i, j) \in C\}$ is a circuit, and U contains no other circuit;

(ii) for each $j_q \in T$ there exists a unique path in U to each $i_k \in C$.

With the aforementioned definition of one-tree, the following lemma can then be shown easily.

LEMMA 4: *Each basic feasible solution of* (P) *defines a collection of disjoint one-trees.*

4. THE SIMPLEX METHOD APPLIED TO (P)

It follows from lemmas 2 and 3 that an initial basic feasible solution is found easily and at any iteration of the simplex method, once an entering variable is known, the variable to leave the basis is immediately determined. Hence the values of the primal variables need not be computed and only the values of the dual variables are required at each iteration. To see that, let us first look at the *dual* of (P):

$$\operatorname{Min} \pi, \tag{15}$$

$$\text{S.t. } \pi_i - \alpha \pi_j + d_{ij} \pi \geqq c_{ij}, \qquad \forall (i, j) \in A, \tag{16}$$

and:

$$\sum_{i=1}^{N} \pi_i a_i = 0, \tag{17}$$

$$\pi \text{ and } \pi_i\text{'s unrestricted in sign.} \tag{18}$$

Let $\{U_1, U_2, \ldots, U_Q\}$ be the one-trees associated with a basic feasible solution to (P). Let $U_q=\{C_q, T_q, E_q\}$ where $C_q=\{i_1, i_2, \ldots, i_m\}$ and $T_q=\{j_1, j_2, \ldots, j_n\}$ and $E_{C_q}=\{(i_1, i_2), \ldots, (i_m, i_1)\}$.

Define:

$$l_q^* = \sum_{k=1}^{m} [\alpha^{k-1} c_{i_k, i_{k+1}}]/(1-\alpha^m),$$

and:

$$m_q^* = \sum_{k=1}^{m} [\alpha^{k-1} d_{i_k, i_{k+1}}]/(1-\alpha^m) \qquad \text{where} \quad i_{m+1} \equiv i_1.$$

For any node $i_s \in U_q$, let $\{(i_s, i_{s+1}) \ldots (i_{s+t_s}, i_1)\}$ be the unique path from i_s to i_1. Define:

$$l_s = \sum_{k=s}^{s+t_s} \alpha^{k-s} c_{i_k, i_{k+1}},$$

and:

$$m_s = \sum_{k=s}^{s+t_s} \alpha^{k-s} d_{i_k, i_{k+1}} \qquad \text{where} \quad i_{s+t_s+1} \equiv i_1,$$

and:

$$l_{i_1} \equiv m_{i_1} \equiv 0.$$

Define, also:

$$\alpha_{(s)} = \alpha^{t_s+1} \qquad \text{and} \qquad \alpha_{(i_1)} \equiv 1.$$

By making use of complementary slackness the following lemma shows how the dual variables in (D) can be determined.

LEMMA 5: *Given a collection of disjoint one-trees $\{U_1, \ldots, U_Q\}$ associated with a basic feasible solution, one can show that:*

$$\pi = \frac{\sum_{i=1}^{N} a_i l_i + \sum_{q=1}^{Q} l_q^* \sum_{i \in U_q} a_i \alpha_{(i)}}{\sum_{i=1}^{N} a_i m_i + \sum_{q=1}^{Q} m_q^* \sum_{i \in U_q} a_i \alpha_{(i)}},$$

and for a node $i \in U_q$:

$$\pi_{i_1} = l_q^* - \pi m_q^*,$$

$$\pi_i = (l_i + \alpha_{(i)} l_q^*) - \pi (m_i + \alpha_{(i)} . m_q^*), \qquad i \neq i_1.$$

Thus once for each U_q, l_q^*, m_q^*, l_i's and m_i's are determined, the dual variables π and π_i's can be obtained by the expressions stated above.

The variable to enter the basis is determined by calculating the following:

$$z^* = \underset{(i,\,j)}{\text{minimum}} \{ \pi_i - \alpha\pi_j + d_{ij}\pi - c_{ij},\ 0 \}.$$

If $z^* = 0$, then the current basic feasible solution yields an optimal solution. If $z^* < 0$, i. e., for some r and s, $\pi_r - \alpha\pi_s + d_{rs}\pi - c_{rs} < 0$, then arc (r, s) can enter the basis. According to lemma 2, we know the variable which leaves the basis. Lemma 4 can be used again to determine the new dual variables, and the simplex method can proceed in this manner from iteration to iteration until an optimal solution is obtained.

Certain observations can be made which lead to some computational savings in updating the dual variables. Since the dual variables are linked by a coupling constraint in (D), a change in one causes change in others. However certain l_i's and m_i's need not be recomputed. Suppose (r, s) is an arc which enters the basis. Denoting the new value of l's and m's by primes, the changes can be accomplished as follows: five cases have to be considered. Define, first:

$$T_q^r = \{ j \mid j \in T_q \text{ such that there is a path from } j \text{ to } r \text{ in } T_q \}.$$

If $r, s \in U_q$, the following three cases arise:

(i) $r \in T_q$ and $s \in T_q^r$;

(ii) $r \in T_q$ and $s \notin T_q^r$;

(iii) $r \in C_q$.

If nodes r, s belong to different one-trees, say $r \in U_q$ and $s \in U_h$, then two cases arise:

(iv) $r \in T_q$;

(v) $r \in C_q$.

In case (i), a new one-tree is created, say $U'_p = (C'_p, T'_p, E'_p)$ with $C'_p = \{ r, s$ and all nodes along the path between r and $s \}$, $T'_p = T_q^i - C'_p$ and E'_p defined accordingly. All l's, m's associated with U'_p have to be recomputed. The one-tree U_q changes since the nodes in T_q^i arc deleted from it; the l's and m's associated with the remaining nodes are left unchanged however.

In case (ii), $l'_r = c_{rs} + \alpha\, l_s$ and $m'_r = d_{rs} + \alpha\, m_s$. For a node $i \in T_q^r$, if (i, j) is an arc in the one tree, $l'_i = c_{ij} + \alpha\, l'_j$ and $m'_i = d_{ij} + \alpha\, m'_j$. For all other nodes l's and m's remain unchanged.

In case (iii), circuit C_q is broken up, the one tree U_q is updated with $C'_q = \{ s, r$ and all the nodes along the path between s and $r \}$, and $T'_q = (T_q \cup C_q) - C'_q$. E'_q changes correspondingly. All l's, m's, l^* and m^*, and $\alpha_{(i)}$'s associated with this updated one-tree have to be recomputed.

In case (iv), we have $T'_q = T_q - T_q^r$ and $T'_h = T_h \cup T_q^r$. Here $l'_r = c_{rs} + \alpha\, l_s$, $m'_r = d_{rs} + \alpha\, m_s$, $\alpha'_{(r)} = \alpha\alpha_{(s)}$. Similar to case (ii), for a node $i \in T_q^r$, if (i, j) is an arc in the one-tree, then $l'_i = c_{ij} + \alpha\, l'_j$, $m'_j = d_{ij} + \alpha\, m'_j$ and $\alpha'_{(i)} = \alpha . \alpha'_{(j)}$. For all other nodes l's and m's remain unchanged.

In case (v) the number of one-trees decreases as U_q disappears and $T'_h = T_h \cup T_q \cup C_q$. Only the quantities associated with this one updated tree have to be recomputed.

5. A NUMERICAL EXAMPLE

A five node problem with c_{ij}/d_{ij} values for the alternative actions as shown in table is considered. Arcs of the form (i, i) are not included; however, the algorithm can handle a network with this type of arcs also. Given are also $a_i = 100$ for $i = 1, 2, \ldots, 5$ and $\alpha = 0.8$.

Iteration 0: Arbitrarily start with two 1-trees:

$$y_{12} = y_{25} = y_{51} = 1, \qquad y_{34} = y_{43} = 1,$$

and other $y_{ij} = 0$.

Iteration 1: Set $l_1 = 0$, $m_1 = 0$ and obtain:

$$l_2 = 5.2, \qquad m_2 = 5.4, \qquad l_5 = 4, \qquad m_5 = 3,$$
$$l^*_{(1)} = 29.0164, \qquad m^*_{(1)} = 17.0492.$$

Set $l_3 = 0$, $m_3 = 0$ and obtain:

$$l_4 = 10, \qquad m_4 = 4, \qquad l^*_{(2)} = 47.2222, \qquad m^*_{(2)} = 25.5556.$$

Hence obtain, $\pi = 1.75$ and get:

$$\pi_1 = -0.8197, \qquad \pi_2 = -4.7746, \qquad \pi_5 = -1.9057,$$
$$\pi_3 = 2.5, \qquad \pi_4 = 5.$$

Calculation of z^* leads to $z^* < 0$ for arc (1, 4).

Hence arc (1, 2) leaves and (1, 4) enters.

Iteration 2: Nodes 1 and 4 corresponding to arc (1, 4) belong to two different 1-trees respectively. The updating of the tree falls under the case (v), and there is only a single 1-tree now:

$$l_3, \quad m_3, \quad l_4, \quad m_4, \qquad l^* = l^*_{(2)} = 47.2222, \qquad m^* = m^*_{(2)} = 25.5556,$$

remain the same as in iteration 1.

TABLE

Data of the Problem

i \ j	1	2	3	4	5
1		10 / 4	5 / 3	2 / 2	5 / 2
2	4 / 5		4 / 5	6 / 7	2 / 3
3	3 / 2	8 / 5		9 / 6	3 / 5
4	5 / 6	7 / 9	10 / 4		5 / 10
5	4 / 3	10 / 5	9 / 8	7 / 6	

Obtain new values of:

$$l_1=10, \quad m_1=5.2; \quad l_5=12, \quad m_5=7.16;$$
$$l_2=11.6, \quad m_2=8.728.$$

Hence obtain $\pi=1.8230$ and get:

$$\pi_3=0.6351, \quad \pi_4=3.2161:$$
$$\pi_1=0.9270, \quad \pi_5=-0.7274, \quad \pi_2=-4.0508.$$

Calculation of z^* leads to $z^* \geqq 0$.

Hence the current solution is optimal.

REFERENCES

1. V. AGGARWAL, R. CHANDRASEKARAN and K. P. K. NAIR, *Markov Ratio Decision Processes*, J. Opt. Th. and Appli., Vol. 21, No. 1, 1977, pp. 27-37.
2. A. CHARNES and W. W. COOPER, *Programming with Linear Fractional Functionals*, Naval Research Logistics Quarterly, Vol. 9, 1974, pp. 1428-1431.
3. Y. M. I. DIRICKX and M. R. RAO, *Networks with Gains in Discrete Dynamic Programming*, Management Science, Vol. 20, No. 11, 1974, pp. 1428-1431.
4. J. F. MAURRAS, *Optimization of the Flow Through Networks with Gains*, Mathematical Programming, Vol. 3, 1972, pp. 135-144.