Routing Problems and Markovian Decision Processes

N. G. F. Sancho*

Department of Mathematics and Statistics, McGill University, Montreal, Quebec H3A 2K6, Canada

Submitted by E. Stanley Lee

1. INTRODUCTION

The problem of finding a routing problem through a network has been solved by dynamic programming (Bellman [1] and Bellman and Dreyfus [2]). A stochastic view of the routing problem for finding the maximum reliability has been studied by Roosta [5]. A different version of the problem is now introduced, where at each city there is a transition probability $p_{ij}$ of going from city $i$ to city $j$. The expected time of going from city $i$ to $N$ (end of destination) is then calculated. If, however, at each city there is a choice of policies (one might decide to travel from city $i$ to city $j$ by a different method of transport) then a new Markovian decision process is formulated. The method is easily extended for finding the maximum reliability from city $i$ to city $N$ using an optimal policy.

2. THE ROUTING PROBLEM

Consider the typical routing problem described by Bellman [1]. We are given $N$ cities, numbering $i = 1, 2, ..., N$, in some order and a set of numbers, $t_{ij}$, where $t_{ij}$ is the time required to travel from city $i$ to city $j$. Let $f_i$ be the time required to travel from the $i$th to the $N$th city using an optimal routing policy. The principle of optimality leads to the following recurrence equation:

$$f_i = \min_{j \neq i} [t_{ij} + f_j], \quad i = 1, 2, ..., N - 1, f_N = 0. \quad (1)$$

Consider the deterministic problem now as a Markov process, i.e., at city $i$ there is a transition probability, $p_{ij}$, of going from city $i$ to city $j$. Hence

*This research was supported in part by NSERC under Contract A-4051.
ROUTING PROBLEMS

\[ \sum_{j=1}^{N} P_{ij} = 1, \forall i \text{'s and } P = \{ P_{ij} \} \text{ is the transition } (N \times N) \text{ matrix. } \]

Let \( v_i \) be the expected time required to travel from the \( i \)-th to the \( N \)-th city:

\[ v_i = \sum_{j=1}^{N} P_{ij} (t_{ij} + v_j), \quad i = 1, 2, \ldots, N - 1, v_N = 0 \quad (2) \]

\[ \text{where } k_i = \sum_{j=1}^{N} p_{ij} t_{ij}. \quad (3) \]

Consider the vector-matrix notation

\[ \mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_{N-1} \end{pmatrix}, \quad \mathbf{k} = \begin{pmatrix} k_1 \\ k_2 \\ \vdots \\ k_{N-1} \end{pmatrix}, \quad \mathbf{A} = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1,N-1} \\ p_{21} & p_{22} & \cdots & p_{2,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ p_{N-1,1} & p_{N-1,2} & \cdots & p_{N-1,N-1} \end{pmatrix}. \]

(Note that the sum of rows of \( A \) are \( \leq 1 \). \( A \) is irreducible with at least one row sum \( < 1 \), as there must be at least one route to get to the \( N \)-th city. It is assumed that the system is completely ergodic with city \( N \) being the only trapping state.) Now Eq. (3) can be written

\[ \mathbf{v} = \mathbf{k} + \mathbf{A} \mathbf{v}. \quad (4) \]

Note \( v_N = 0 \) so it need not be written into the vector-matrix form of Eq. (4). This Eq. (4) turns out to be the same Leontief input–output model of mathematical economics described in Lancaster [4]. Its solution is given by

\[ \mathbf{v} = (I - \mathbf{A})^{-1} \mathbf{k}, \quad (5) \]

\( (I - \mathbf{A})^{-1} \) exists with non-negative coefficients as \( \mathbf{A} \geq 0 \) with row sums \( \leq 1 \) with at least one row sum \( < 1 \), see [4].

Let us now assume that at each city \( i \) the transition matrix can be chosen from a set of such matrices and denote the matrix corresponding to policy decision by \( P(q) = \{ p_{ij}(q) \} \), where \( \sum_{j=1}^{N} P_{ij}(q) = 1, \forall i \text{'s. Let } T(q) \text{ be the corresponding time matrix using policy } q \text{ given by } \{ t_{ij}(q) \} \text{ and let } f(i) \text{ be the expected time required to travel from the } i\text{-th to the } N\text{-th city using an optimal routing policy. Hence using the principle of optimality we obtain}

\[ f(i) = \min_q \left\{ \sum_{j=1}^{N} p_{ij}(q) (t_{ij}(q) + f(j)) \right\}, \quad i = 1, 2, \ldots, N - 1; f(N) = 0 \quad (6) \]

\[ = \min_q \left\{ k_i(q) + \sum_{j=1}^{N} p_{ij}(q) f(j) \right\}, \quad \text{where } k_i(q) = \sum_{j=1}^{N} p_{ij}(q) t_{ij}(q). \quad (7) \]
Equation (7) can be solved using Howard’s policy iteration method \([2, 3]\). This is described below:

1. We first guess a particular policy, \(q\), for each \(f(i), i = 1, 2, \ldots, N - 1\).
2. Next we write out the transition matrix \(\{p_{ij}(q)\}\) associated with each particular policy guessed.
3. Solve for \(v_i, i = 1, 2, \ldots, N - 1\), using Eq. (5), with \(v_i\) substituted for \(f(i)\).
4. Substitute the values of \(v_j\) for \(f(j), j = 1, 2, \ldots, N - 1, v_N = 0\) into the r.h.s. of (7) and find the minimum for different values of \(q\). This is called the policy improvement routine.
5. Repeat step (1) for this new policy and stop when there is no change in policy.

It can be shown that this will eventually lead to an optimal policy. The solution to this problem therefore tells you, if you are in a particular city \(i\), what is your immediate optimal policy in order to get to city \(N\) in the minimum expected time.

The above method works only when city \(N\) is the only trapping state. A method is described below for finding the maximum reliability of non-ergodic systems for travelling from city \(i\) to city \(N\) when there are more than one trapping states besides city \(N\). Let \(g_i\) denote the maximum reliability of getting from city \(i\) to city \(N\) using an optimal policy. From the principle of optimality we obtain

\[
g_i = \max_q \left[ \sum_{j=1}^{N} p_{ij}(q) g_j \right], \quad i = 1, 2, \ldots, N; g_N = 1
\]  

or

\[
g_i = \max_q \left[ \sum_{j=1}^{N-1} p_{ij}(q) g_j + p_{iN}(q) \right], \quad i = 1, 2, \ldots, N - 1; \text{ as } g_N = 1.
\]  

Let \(w_i\) be defined now as the reliability of getting from city \(i\) to city \(N\) using a certain policy \(q\), i.e.,

\[
w_i = \sum_{j=1}^{N-1} p_{ij} w_j + p_{iN}, \quad w_N = 1, i = 1, 2, \ldots, N - 1.
\]

In vector-matrix form this equation can be written

\[
w = Aw + d
\]
where \( \mathbf{w} \) is an \( N - 1 \) vector, \( \mathbf{d} \) is an \( N - 1 \) vector with positive elements and \( \mathbf{A} \) is an \( (N - 1) \) square matrix defined as before whose row sums are \( \leq 1 \) with at least one row sum \( < 1 \). Its solution is again given by \( \mathbf{w} = (I - \mathbf{A})^{-1} \mathbf{d} \) and we can again use Howard’s policy iteration method [3] to solve for \( g_i \) in (9).

3. Proof of Policy Iteration Method

Assume we have evaluated a policy \( \alpha \) in the operation of the system and the policy improvement routine has produced a policy \( \beta \) that is different from \( \alpha \). We therefore seek to prove that \( v_i(\beta) > v_i(\alpha) \), see also Howard [3].

From the P.I.R. since \( \beta \) was chosen over \( \alpha \) we have

\[
\sum_{j=1}^{N} p_{ij}(\beta) v_j(\alpha) \geq k_i(\alpha) + \sum_{j=1}^{N} p_{ij}(\alpha) v_j(\alpha)
\]

\[
\therefore k_i(\beta) - k_i(\alpha) \geq \sum_{j=1}^{N} p_{ij}(\alpha) v_j(\alpha) - \sum_{j=1}^{N} p_{ij}(\beta) v_j(\alpha). \tag{11}
\]

For policies \( \alpha \) and \( \beta \) individually we have from (3)

\[
v_i(\beta) = k_i(\beta) + \sum_{j=1}^{N} p_{ij}(\beta) v_j(\beta) \tag{12}
\]

\[
v_i(\alpha) = k_i(\alpha) + \sum_{j=1}^{N} p_{ij}(\alpha) v_j(\alpha), \tag{13}
\]

giving

\[
k_i(\beta) - k_i(\alpha) = v_i(\beta) - v_i(\alpha) + \sum_{j=1}^{N} p_{ij}(\alpha) v_j(\alpha) - \sum_{j=1}^{N} p_{ij}(\beta) v_j(\beta). \tag{14}
\]

Substituting (14) into (11) we obtain

\[
v_i(\beta) - v_i(\alpha) \geq \sum_{j=1}^{N} p_{ij}(\beta)(v_j(\beta) - v_j(\alpha)). \tag{15}
\]

Let

\[
\Delta v_i = v_i(\beta) - v_i(\alpha).
\]

Hence (15) can be written

\[
\Delta v_i \geq \sum_{j=1}^{N} p_{ij}(\beta) \Delta v_j
\]
\[ \Delta v_i = \gamma_i + \sum_{j=1}^{N} p_{ij}(\beta) \Delta v_j, \quad \gamma_i \geq 0 \] (16)

i.e.,

\[ \Delta v = \gamma + A(\beta) \Delta v \]

where \( \Delta v \) is a vector of order \( N - 1 \) and \( A(\beta) \) is an \( N - 1 \) square matrix as described before and \( v_N = 0 \) for any policy. The solution to this Leontief equation is given by

\[ \Delta v = (I - A(\beta))^{-1} \gamma \]

where \( (I - A(\beta))^{-1} \) exists with non-negative coefficients. Hence \( \Delta v \) will have non-negative elements and \( v_i(\beta) - v_i(\alpha) \geq 0 \).

Assume now that the P.I.R. has converged on policy \( \alpha \) and there exists a policy \( \beta \) such that \( v_i(\beta) - v_i(\alpha) \geq 0 \). Since it has converged on \( \alpha \) then following the same arguments given above with \( \beta \) interchanged for \( \alpha \) and \( \alpha \) for \( \beta \) we again obtain \( v_i(\alpha) - v_i(\beta) \geq 0 \). This contradicts the assumption that \( v_i(\beta) - v_i(\alpha) \geq 0 \), hence there is no policy \( \beta \) such that \( v_i(\beta) - v_i(\alpha) \geq 0 \).

4. WORKED EXAMPLE

Consider a \( P(1) \) matrix of policy 1 given by

\[
P(1) = \begin{pmatrix}
0 & 0.5 & 0.5 & 0 \\
0 & 0 & 0.2 & 0.8 \\
0 & 0.3 & 0 & 0.7 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]

and let the corresponding \( \{k_i(1)\} \) vector associated with the time matrix and policy 1 be given by

\[
k(1) = \begin{pmatrix}
4 \\
5 \\
6
\end{pmatrix}.
\]
Let the new $P(2)$ matrix of policy 2 be given by

$$P(2) = \begin{pmatrix} 0 & 0.6 & 0.4 & 0 \\ 0 & 0 & 0.5 & 0.5 \\ 0 & 0.4 & 0 & 0.6 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

and the corresponding $\{k_i(2)\}$ vector associated with the time matrix and policy 2 be given by

$$k(2) = \begin{pmatrix} 3 \\ 4 \\ 5 \end{pmatrix}.$$

Starting with policy

$$\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix},$$

i.e., policy 1 for $f(1)$, $f(2)$ and $f(3)$, we solve for $v$ in Eq. (5) ($v$ replaces $f$) with

$$A = \begin{pmatrix} 0 & 0.5 & 0.5 \\ 0 & 0 & 0.2 \\ 0 & 0.3 & 0 \end{pmatrix},$$

giving

$$v = \begin{pmatrix} 11.2873 \\ 6.5958 \\ 7.9783 \end{pmatrix}.$$
Continuing on with step (4) of the policy improvement routine we obtain the new values for Eq. (7):

\[
\begin{array}{c|c|c}
  i & \text{policy} & k_1(q) + \sum_{j=1}^{N} p_{ij}(q) v_j \\
  1 & 1 & 11.287 \\
  & 2 & 10.149 \\
  2 & 1 & 6.596 \\
  & 2 & 7.989 \\
  3 & 1 & 7.978 \\
  & 2 & 7.638 \\
\end{array}
\]

We see that for \( i = 1 \), the quantity in the right-hand column is minimized when \( q = 2 \). For \( i = 2 \) and 3, it is minimized when \( q = 1 \) and 2, respectively. Hence our new policy is

\[
\begin{pmatrix}
  2 \\
  1 \\
  2 
\end{pmatrix}
\]

Using this new policy we solve for \( v \) using Eq. (5) with

\[
A = \begin{pmatrix}
  0 & 0.6 & 0.4 \\
  0 & 0 & 0.2 \\
  0 & 0.4 & 0
\end{pmatrix}
\quad \text{and} \quad
k = \begin{pmatrix}
  3 \\
  5 \\
  5
\end{pmatrix}
\]

to obtain

\[
v = \begin{pmatrix}
  9.9565 \\
  6.5217 \\
  7.6087
\end{pmatrix}.
\]

Substituting for this value of \( v \) into (7) we again obtain the same policy

\[
\begin{pmatrix}
  2 \\
  1 \\
  2
\end{pmatrix}
\]
and we can conclude that the system has now converged with

\[
\mathbf{f} = \begin{pmatrix} 9.9565 \\ 6.5217 \\ 7.6087 \end{pmatrix}.
\]

5. Discussion

We have studied a stochastic version of the routing problem, when starting at city \(i\) there is probability \(p_{ij}\) of going from city \(i\) to city \(j\). The results obtained are very similar to the Markovian decision process of Howard [3] and Bellman and Dreyfus [2]. It is conceivable to extend this problem to the travelling salesman problem, where we can obtain the minimum expected time to tour all the cities and return home if there is a probability associated with going from city \(i\) to city \(j\).

References