ON A ROUTING PROBLEM*

By RICHARD BELLMAN (*The RAND Corporation*)

Summary. Given a set of N cities, with every two linked by a road, and the times required to traverse these roads, we wish to determine the path from one given city to another given city which minimizes the travel time. The times are not directly proportional to the distances due to varying quality of roads and varying quantities of traffic.

The functional equation technique of dynamic programming, combined with approximation in policy space, yields an iterative algorithm which converges after at most $(N - 1)$ iterations.

1. Introduction. The problem we wish to treat is a combinatorial one involving the determination of an optimal route from one point to another. These problems are usually difficult when we allow a continuum, and when we admit only a discrete set of paths, as we shall do below, they are notoriously so.

The purpose of this paper is to show that the functional equation technique of dynamic programming, [1, 2], combined with the concept of approximation in policy space, yields a method of successive approximations which is readily accessible to either hand or machine computation for problems of realistic magnitude. The method is distinguished by the fact that it is a method of exhaustion, i.e. it converges after a finite number of iterations, bounded in advance.

2. Formulation. Consider a set of N cities, numbered in some arbitrary fashion from 1 to N, with every two linked by a direct road. The time required to travel from i to j is not directly proportional to the distance between i and j, due to road conditions and traffic. Given the matrix $T = (t_{ij})$, not necessarily symmetric, where t_{ij} is the time required to travel from i to j, we wish to trace a path between 1 and N which consumes minimum time.

Since there are only a finite number of paths available, the problem reduces to choosing the smallest from a finite set of numbers. This direct, or enumerative, approach is impossible to execute, however, for values of N of the order of magnitude of 20.

We shall construct a search technique which greatly reduces the time required to find minimal paths.

3. Functional equation approach. Let us now introduce a dynamic programming approach. Let

$$f_i = \text{the time required to travel from } i \text{ to } N,\ i = 1, 2, \cdots, N-1, \text{ using an optimal policy,} \tag{3.1}$$

with $f_N = 0$.

Employing the principle of optimality, we see that the f_i satisfy the nonlinear system of equations

$$\begin{aligned} f_i &= \operatorname*{Min}_{j \neq i} [t_{ij} + f_j], \qquad i = 1, 2, \cdots, N-1, \\ f_N &= 0. \end{aligned} \tag{3.2}$$

*Received January 30, 1957.

This system differs from the usual systems encountered in dynamic programming in that we do not have a ready computational scheme.

4. Uniqueness. Let us show that there exists at most one solution of the system in (3.2).

Assume that $\{f_i\}$ and $\{F_i\}$ are two solutions, with $f_N = F_N = 0$, and let k be an index for which $f_k - F_k$ achieves it maximum. Then

$$\begin{aligned} f_k &= \operatorname*{Min}_{j \neq k} [t_{kj} + f_j] \\ F_k &= \operatorname*{Min}_{j \neq k} [t_{kj} + F_j]. \end{aligned} \tag{4.1}$$

Let the minimum in the first equation be assumed for $j = r$, and the second for $j = s$. It is clear, since $t_{ij} > 0$ for all i, j, that $r \neq k$, $s \neq k$. Then we have the equalities and inequalities:

$$\begin{aligned} f_k &= t_{kr} + f_r \leq t_{ks} + f_s , \\ F_k &= t_{ks} + F_s \leq t_{kr} + F_r . \end{aligned} \tag{4.2}$$

These lead to

$$\begin{aligned} f_k - F_k &\leq f_s - F_s \\ &\geq f_r - F_r . \end{aligned} \tag{4.3}$$

Since k was an index where $f_k - F_k$ achieved its maximum, we must have

$$f_k - F_k = f_s - F_s , \tag{4.4}$$

which can only be true if in (2) we have

$$f_k = t_{ks} + f_s . \tag{4.5}$$

Now repeat this procedure for the pair $\{f_s , F_s\}$. It follows, from the foregoing argument, that there must be another pair $\{f_p , F_p\}$ with $f_p - F_p = f_s - F_s = f_k - F_k$. Furthermore, $p \neq s$, and $p \neq k$, since we have

$$f_k = t_{ks} + t_{sp} + f_p . \tag{4.6}$$

Proceeding in this way, we exhaust the set of points $i = 1, 2, \cdots , N - 1$, with the result that one of the terms in the continued equality above must be $f_N - F_N = 0$. Hence $f_i = F_i$ for $i = 1, 2, \cdots , N - 1$.

5. Approximation in policy space. Let us now turn to the problem of determining an algorithm for obtaining the solution of the system in (3.2). The basic method is that of successive approximations. We choose an initial sequence $\{f_i^{(0)}\}$, and then proceed iteratively, setting

$$\begin{aligned} f_i^{(k+1)} &= \operatorname*{Min}_{j \neq i} (t_{ij} + f_j^{(k)}], \qquad i = 1, 2, \cdots , N - 1, \\ f_N^{(k+1)} &= 0, \end{aligned} \tag{5.1}$$

for $k = 0, 1, 2, \cdots ,$.

The choice of $\{f_i^{(0)}\}$ seems to require some care. Let us then invoke the concept of approximation in policy space in order to obtain a sequence which is monotone increasing.

Perhaps the simplest policy that we can employ is to proceed directly from i to N. Define

$$f_i^{(0)} = t_{iN}, \qquad i = 1, 2, \cdots, N. \tag{5.2}$$

It follows that $f_i^{(1)}$ as defined by

$$\begin{aligned} f_i^{(1)} &= \operatorname*{Min}_{j \neq i} [t_{ij} + f_j^{(0)}], \qquad i = 1, 2, \cdots, N-1, \\ f_N^{(1)} &= 0, \end{aligned} \tag{5.3}$$

satisfies the inequality

$$f_i^{(1)} \leq f_i^{(0)}, \qquad i = 1, 2, \cdots, N. \tag{5.4}$$

This inequality is immediate when we realize that $f_i^{(1)}$ represents the minimum time for a path with at most one stop. It follows then inductively that the sequences $\{f_i^{(k)}\}$ as defined by (5.1), with $f_i^{(0)}$ as in (5.2), satisfy the inequalities

$$f_i^{(k+1)} \leq f_i^{(k)}, \qquad i = 1, 2, \cdots, N, \qquad k = 0, 1, 2, \cdots \tag{5.5}$$

It is important to note that there are many other policies we could employ to obtain monotone convergence.*

It follows that

$$\lim_{k \to \infty} f_i^{(k)} = f_i, \qquad i = 1, 2, \cdots, N, \tag{5.6}$$

furnishing a solution to (3.2).

It is clear from the physical interpretation of this iterative scheme that at most $(N - 1)$ iterations are required for the sequence to converge to the solution.

6. Computational aspects. It is easily seen that the iterative scheme discussed above is a feasible method for either hand or machine computation for values of N of the order of magnitude of 50 or 100.†

For each i, we require only the column (t_{ij}), $j = 1, 2, \cdots, N$ of the matrix T. Hence, the memory requirement for a digital computer is small.

7. Monotone increasing convergence. Turning back to (3.2), let us consider the sequence of approximations defined by

$$\begin{aligned} f_i^{(0)} &= \operatorname*{Min}_{j \neq i} t_{ij}, \qquad i = 1, 2, \cdots, N = 1, \\ f_N^{(0)} &= 0, \\ f_i^{(k+1)} &= \operatorname*{Min}_{j \neq i} [t_{ij} + f_j^{(k)}], \qquad i = 1, 2, \cdots, N-1, \\ f_N^{(k+1)} &= 0. \end{aligned} \tag{7.1}$$

It is clear that $f_i^{(k+1)} \geq f_i^{(k)}$. It is not, however, obvious that this method yields a uniformly bounded sequence. To establish this, let us show, inductively, that

$$f_i^{(k)} \leq f_i, \qquad i = 1, 2, \cdots, N, \qquad k = 0, 1, 2, \cdots, \tag{7.2}$$

where $\{f_i\}$ is the solution of (3.2).

---

*I owe this choice of an initial policy to F. Haight.

†*Added in proof* (*December* 1957): After this paper was written, the author was informed by Max Woodbury and George Dantzig that the particular iterative scheme discussed in Sec. 5 had been obtained by them from first principles.

The inequality is certainly true for $k = 0$. Hence, assuming that it holds for k, we have

$$\begin{aligned} f_i^{(k+1)} &= \operatorname*{Min}_{j\neq i} [t_{ij} + f_j^{(k)}] \leq \operatorname*{Min}_{j\neq i} [t_{ij} + f_j] \\ &\leq f_i . \end{aligned} \tag{7.3}$$

It follows that the sequence $\{f_i^{(k)}\}$ converges to $\{f_i\}$ as $k \to \infty$, furnishing the desired monotone convergence. Once again, only a finite number of iterations will be required. It is to be expected that the first method will converge more rapidly.

References

1. R. Bellman, *Dynamic Programming,* Princeton University Press, 1957
2. R. Bellman, *The Theory of Dynamic Programming,* Bull. Amer. Math. Soc., **60,** 503-515 (1954)

ON THE DAMPED OSCILLATIONS EQUATION WITH VARIABLE COEFFICIENTS*

By E. V. Laitone (*University of California, Berkeley*)

A useful upper bound can be given for the oscillatory solutions of the second-order linear differential equation with continuous differentiable coefficients

$$u''(t) + p(t)u'(t) + q(t)u(t) = 0 \tag{1}$$

which commonly occurs in vibration studies and airplane or missile dynamics.

The solutions of Eq. (1) are oscillatory whenever

$$\phi(t) = (q - p^2/4 - p'/2) \geqq m^2 > 0 \tag{2}$$

see Bellman [1] or Kamke [2], and for this case it will be shown that as long as ϕ is monotonic the solutions are bounded in the following manner:

$$| u(t) | \leqq \frac{1}{m} \{\phi(0)u(0)^2 + [u'(0) + p(0)u(0)/2]^2\}^{1/2} \exp\left(-\int_0^t p/2 \, dt\right). \tag{3}$$

This result can be derived by introducing the unique transformation, see Bellman [1] or Kamke [2]

$$u = v \exp\left(-\int_0^t p/2 \, dt\right) \tag{4}$$

which preserves the same zeros in the oscillatory solutions. Then Eq. (1) is transformed to

$$v''(t) + \phi(t)v(t) = 0, \tag{5}$$

*Received February 5, 1957.