JOURNAL OF MATHEMATICAL ANALYSIS AND APPLICATIONS 14, 191-197 (1966)

The Shortest Route Problem with Constraints*

H. C. JOKSCH

The Mitre Corporation, Bedford, Massachusetts

Submitted by Richard Bellman

The shortest route problem has drawn much attention and several effective methods for its solution are known [1]. A natural extension, the kth shortest route problem, has also been studied and solved [2]. Another just as natural extension with many practical applications has apparently not been studied in the literature.[1] It is the following problem: Given a set of nodes, pairs of which are connected by one or several arcs which are characterized by numbers a_k and b_k: Find a route from a given origin to a given terminal through the net such that the sum of the a_k along the route is minimal, under the constraint that the sum of the b_k along the route is not less (or more) than a certain t.

The arcs might be directed or undirected; one can always assume a directed network by replacing undirected arcs by two directed ones. We will also assume that the a_k and b_k are nonnegative. Even then the optimal route might contain loops, or there might be no routes satisfying the constraint. If loops are possible, additional constraints might be given, such as not to use an arc repeatedly, not to go from one node to another or to pass a node twice or more than a specified number of times

1. THE LINEAR PROGRAMMING APPROACH

If one uses variables x_k which shall be nonnegative integers to indicate how many times the arc k is used by a route, the problem is to minimize

$$z = \sum a_k x_k \tag{1}$$

[1] After this study was completed, the author learned of the work of Witzgall and Goldman [3], which was presented at the 27th National Meeting of ORSA, Boston, May 7, 1965. They consider a slightly different problem, but arrive at essentially the method presented in Section 2.

* This work was sponsored by the Air Force Systems Command, Electronics System Division, under Contract AF 19(628)2390. Further reproduction is authorized for purposes of the United States Government.

under the constraint

$$\sum b_k x_k \geqslant t \tag{2}$$

and the conditions which ensure that the x_k describe a route, namely,

$$\sum_{k \in L(0)} x_k = 1 \tag{3}$$

$$\sum_{k \in L(j)} x_k = \sum_{k \in E(j)} x_k \qquad \text{for all} \qquad j = 1, \cdots, N-1 \tag{4}$$

and

$$\sum_{k \in E(N)} x_k = 1, \tag{5}$$

where $L(j)$ and $E(j)$ are the are the sets of arcs which are leaving and entering respectively the node j.

Equations (1)-(5) form an integer linear programming problem which can be solved by Gomory's method. However, the simplicity of the linear programming solution for the classical shortest route problem is due to the fact that the basic solutions are automatically integer, and therefore the solution of the dual problem can be used. This suggests the method of Land and Doig for solving the problem (1)-(5), because it leads to a problem with a simple dual. The basic idea is to add a constraint

$$\sum a_k x_k \geqslant \lambda \tag{6}$$

to the problem and solve it parametrically with respect to λ. A systematic search shows for which smallest value of λ a solution has integer values x_k. However, for each λ which is larger than the $z_{\min}$ of the problem (1)-(5), the solution is multiple and careful bookkeeping is necessary to find the integer solutions.

In our case, however, the method becomes useless. Assume that in a diagram like Fig. 1 the $\sum a_k x_k$ and $\sum b_k x_k$ for all routes through the net are plotted. Each of these routes corresponds to a basic solution of the system (3)-(5). Any combination of basic solutions with positive weights of sum 1 is a solution of (3)-(5) and vice versa. Therefore, the optimal noninteger solution of the system (1)-(5) is usually a linear combination of two basic solutions, as indicated by point A in Fig. 1: the optimal solution to the noninteger problem gives two routes. However, none of these routes needs to be *the* optimal route, which is Route 3 in the case shown in Fig. 1. If the constraint (6) with $\lambda > z_{\min}$ is added, the optimal noninteger solution (B in Fig. 1) is usually a linear combination of at least three basic solutions; in fact,

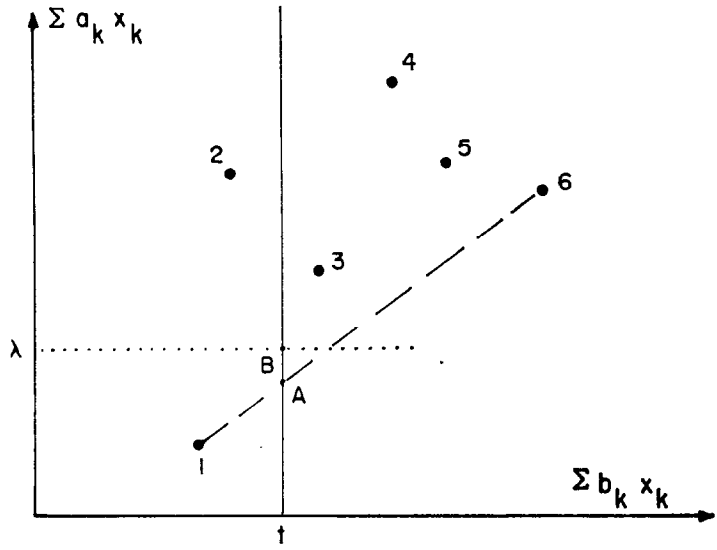

FIG. 1

it is a combination of *all* basic solutions of (2)-(5) with weights in a certain linear space. Therefore, in order to apply the search technique used in Land and Doig's method one has to know all routes through the net. One would not have any advantage over a naive exhaustive search.

However, the linear programming approach has the advantage that additional constraints such as those mentioned above concerning the number of times an arc, a group of arcs or a node might be passed can easily be incorporated into the model. Therefore, a search for more sophisticated solution techniques along these lines seems to be worthwhile.

2. THE DYNAMIC PROGRAMMING APPROACH

The dynamic programming solution of the shortest route problem is based on the fact that any part of a shortest route from its origin to its terminal is also the shortest route between its endpoints. Similarly, the solution of the kth shortest route problem uses the fact that the kth shortest path is defined between any two points of a net. In our problem the difficulty arises that satisfaction or violation of condition (2) is defined only for an entire route from its origin to its terminal, but not its parts. The obvious way to overcome this difficulty is to treat t in (2) as a parameter and solve the problem for all—within a reasonable range—values of t.

Let $z_i(t)$ be defined as the length of the shortest route from the origin to node i, for which condition (2) holds. Then the recursion formula

$$z_j(t) = \min_{k \in S(i,j)} \{z_i(t - b_k) + a_k\} \tag{7}$$

has to hold, where $S(i, j)$ is the set of all arcs going from i to j. For the origin we have

$$z_0(t) = 0 \tag{8}$$

In order to show that there exists, at most, one set of functions $z_j(t)$ satisfying (7) and (8), we assume that there are two different sets, $z_i(t)$ and $z_i^*(t)$. We chose a node j and value t such that $z_j(t) > z_j^*(t)$. Since both functions satisfy (7), the following relations hold:

$$z_j(t) = z_p(t - b_k) + a_k \leqslant z_i(t - b_l) + a_l \qquad l \in S(i, j) \tag{9}$$

$$z_j^*(t) = z_q^*(t - b_m) + a_m \leqslant z_i^*(t - b_l) + a_l \qquad l \in S(i, j), \tag{10}$$

where p, q, k, and m are the numbers for which the minima in (7) are obtained and i and l can be any numbers for which $S(i, j) \neq \emptyset$. From this it follows that

$$z_j(t) \leqslant z_q(t - b) + a_m \tag{11}$$

and

$$z_j(t) - z_j^*(t) \leqslant z_q(t - b) - z_q^*(t - b_m). \tag{12}$$

Therefore,

$$z_q(t - b_m) - z_q^*(t - b_m) > 0. \tag{13}$$

We can continue in this manner and arrive finally, possibly after several loops, at

$$z_0(t) - z_0^*(t) > 0, \tag{14}$$

which contradicts our assumption about $z_0(t)$.

That there exists a solution is shown by the following construction, which is patterned after the successive approximation procedure for the shortest route problem [4]. One defines

$$z_j^{r+1}(t) = \min_{k \in S(i,j)} \{z_i^r(t - b_k) + a_k\} \tag{15}$$

$$z_j^0(t) = \infty \qquad j \neq 0 \tag{16}$$

$$z_0^r(t) = 0. \tag{17}$$

This allows recursive calculation of $z_j^r(t)$. Since $0 \leqslant z_j^{r+1}(t) \leqslant z_j^r(t)$, and since the differences, if any, are larger than a certain amount, it follows that the sequences $t_j^r(t)$ converge in finite numbers of steps to functions $z_j(t)$.

It is possible to calculate $z_j(t)$ from the recursion formula (15) for sufficiently many values of t. However, this is an ineffective method since $z_j(t)$ is a step function. It is sufficient to know the locations of these steps. Each step corresponds to a route; using an economic term one might call them "efficient routes." Thus one can describe the function $z_j(t)$ by a listing of the efficient routes

$$R_j = \{z_j^p, t_j^p, n_j^p\}, \tag{18}$$

where p is the number of the route in the listing and n_j^p the number of the last arc in the route.

To construct R_j^{r+1} according to (15) we proceed in the following manner: assume an R_j^* to be given (at the beginning of the procedure for this node it will be the empty set). Choose an arc k leading into j. The origin i of this arc has the set of efficient routes R_i^r. We then "merge" R_j^* with the extensions of R_i^r, retaining only the efficient routes and obtaining R_j^{**}. Then we drop one asterisk, proceed to the next k and so on until none is left. The last R_j^{**} is R_j^{r+1}. Then we go on to the next j, and after all nodes are exhausted, to the next iteration.

The "merging" of the two sets of routes is performed in the following manner:

$$R_j^* = \{z_j^{*p}, t_j^{*p}, n_j^{*p}\} \quad \text{and} \quad R_j^r = \{z_i^{rq}, t_i^{rq}, n_i^{rq}\} \tag{19}$$

are known. Assume that we already know the first u efficient routes of R_j^{**}. Then we determine

$$z_j^{**u+1} = \min_{t_j^{*p} > t_j^{**u}} \{z_j^{*p}, z_i^{rq} + a_k\} \qquad t_i^{rq} + b_k > t_j^{**u}. \tag{20}$$

If

$$z_j^{**u+1} = z_j^{*g} \tag{21}$$

we have

$$t_j^{**u+1} = t_j^{*g} \quad \text{and} \quad n_j^{**u+1} = n_j^{*g}, \tag{22}$$

but if

$$z_j^{**u+1} = z_j^{rh} + a_k \tag{23}$$

we have

$$t_j^{**u+1} = t_i^{rh} + b_k \quad \text{and} \quad n_j^{**u+1} = k. \tag{24}$$

Thus one finally obtains the set of all efficient paths (in a net with loops one can only obtain all paths up to an arbitrary value of t) to the terminal point.

This is more than the original formulation of the problem asked for. However, in many practical problems the knowledge of all or more than one efficient route with values of t near the desired one is more important than *the* solution of the problem (1)-(5). On the other hand, if only *the* solution is wanted, the question arises whether this dynamic programming solution is computationally worthwhile, because it is easy to find examples where all routes from the origin to the terminal are efficient routes; therefore the method would not offer an advantage over complete enumeration.

The special conditions mentioned above, which can easily be incorporated into a linear programming formulation, cannot be incorporated into a simple dynamic programming model, since they imply knowledge of the "past history" of a route. There seems to be no computationally feasible way to overcome this difficulty in a dynamic programming model.

3. Problems with Several Constraints

The dynamic programming approach can easily be extended to problems where the sums of several parameters b_k, c_k, $\cdots$ of the arcs are restricted.

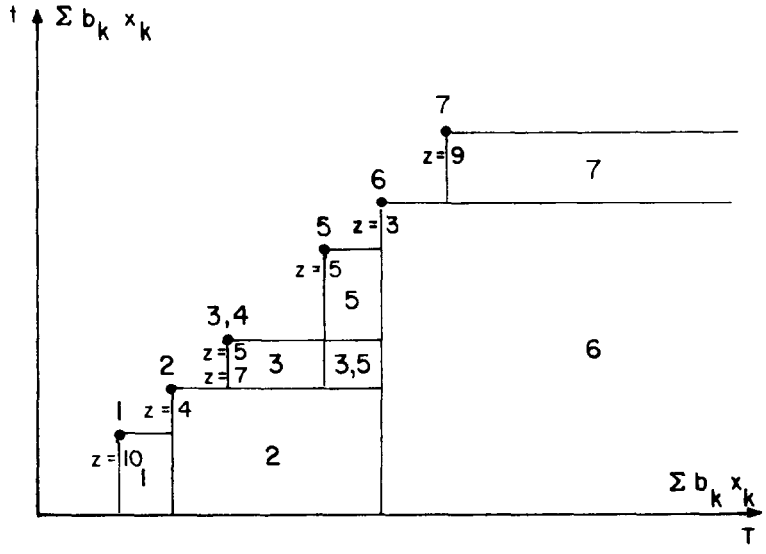

Fig. 2

Only the computational aspects become more cricital. Of special interest for some applications is the case where two constrains

$$\sum b_k x_k \geqslant t \tag{25}$$

and

$$\sum b_k x_k \leqslant T \tag{26}$$

are given. Though in some special cases the optimal solution can be found by examining the efficient solutions of the problem with either condition (25) or (26), generally one has to determine the set of efficient solutions for all combinations of t and T. This set consists of all routes through the net except when some routes have the same $\sum b_k x_k$, but different $\sum a_k x_k$. This is illustrated in Fig. 2, where for all routes through the net the $\sum b_k x_k$ are plotted as coordinates and the points labelled with the number of the route and $z = \sum a_k x_k$. It also shows which route is the optimal solution for any combination of t and T. The only nonefficient route is 4 which has the same $\sum b_k x_k$ as 3, but a larger z. This demonstrates that any method determining all efficient routes through the net will usually not be computationally worthwhile, except when there are many routes with the same values of $\sum b_k x_k$.

References

1. M. Pollack and W. Wiebenson. Solutions of the shortest route problem—A review. *Operations Res.* **8** (1960), 224-230.
2. M. Pollack. Solutions of the k-th best route through a network—A review. *J. Math. Anal. Appl.* **3** (1961), 547-559.
3. C. Witzgall and A. J. Goldman. Most profitable routing before maintenance. *Bull. Operations Res. Soc. Am.* (1965), B82.
4. R. Bellman. On a routing problem. *Quart. Appl. Math.* **16** (1958), 87-90.