Environment and Planning B: Planning and Design, 1988, volume 15, pages 447-460

The multiobjective vending problem: a generalization of the travelling salesman problem

C P Keller
Department of Geography, University of Victoria, Victoria, BC, Canada
M F Goodchild
Department of Geography, University of Western Ontario, London, Ont, Canada
Received 29 October 1987; in revised form 3 March 1988

Abstract. A generalization of the travelling salesman problem is introduced. Each node has an associated reward, and a penalty is incurred by travelling between nodes. In the multiobjective vending problem, the subset of nodes and associated tour which will minimize penalty and maximize reward is sought. The problem is placed within the context of multiobjective programming. A heuristic is proposed and evaluated, and it is found to give satisfactory performance when applied to a problem with twenty-five nodes. Further generalizations are suggested.

Introduction

The travelling salesman problem (TSP) is defined conceptually as follows: given a set of N nodes, determine the shortest complete circuit that connects all nodes, so that every node is visited once and once only.

The applications of the TSP to spatial analysis are broad and varied. This has led to the definition of numerous extensions and modifications to the basic TSP definition. These include, amongst others, the multiple travelling salesman problem (introduced by Dantzig et al, 1959), and a combination of TSP and depot location problems (Clarke and Wright, 1964; Orloff, 1974). A complete review of all TSP generalizations and their solutions is beyond this paper. A logical summary of generalizations of the TSP is discussed by Keller (1986).

An assumption underlying most of the above research is that the set of nodes to be visited is given and fixed. The problems are therefore generalized and are reduced to a single objective problem, that of finding one or a number of optimal travelling salesman routes that connect all specified nodes. This is not always realistic. Consider the hypothetical example demonstrated in figure 1, which shows a fourteen-node problem and the optimum (shortest) travelling salesman path.

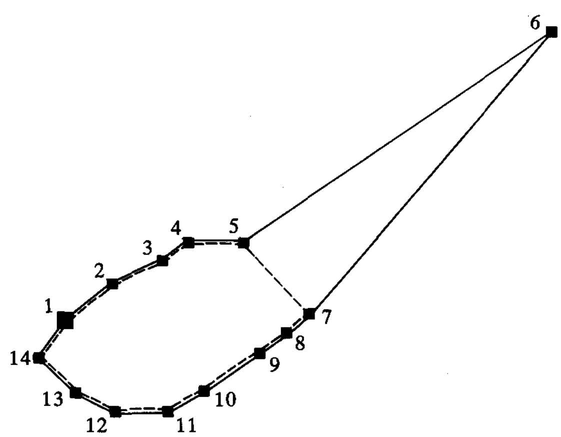

Figure 1. Hypothetical example of a travelling salesman route.

Assume that a penalty, which is proportional to the length of the route, must be accepted for travelling between the nodes, and that each node has associated with it some reward-potential that can be collected upon arrival at that node. Figure 1 shows node six to be located at quite some distance from all the other nodes. The route-penalty, that must be accepted to travel to and from node six, is therefore relatively large. The question may arise as to whether it is worthwhile travelling the extra distance to visit node six, or whether this node should be excluded from the route-sequence. The answer to this question will depend on the size of the reward offered at node six, and on the trade-off relationship established between reward and penalty.

Periodic marketing (Bromley et al, 1975; Ghosh, 1982; Hay, 1971; Webber and Symanski, 1973) provides a useful practical application. The problem faced by the vendors in the market is to find a tour which can be repeated on a regular cycle, and which combines the minimum of travel-penalty with the maximum of marketing-reward. Another conceptual application is the scheduling of a mobile vending service, such as a lunch truck, which must balance the time lost in travelling from site to site with the magnitude of the potential gains at each one (Keller, 1985). Similar problems arise in other areas of marketing, as well as in a number of public-sector applications where a mobile service cannot afford to visit every possible demand-site. Another application concerns the identification of the optimal route in score orienteering competitions, as identified by Tsiligirides (1984) and Golden et al (1985).

The problem raised contains two conceptual objectives, that of maximizing the reward to be collected by visiting as many nodes as possible, and that of keeping the total link-penalty to a minimum. If the two objectives can be defined in commensurable terms, say dollars, or if a trade-off relationship can be specified, then the problem can be solved as a single-objective problem. However, in many cases the two objectives will not be commensurable, and a study of the trade-off relationship between them may be of interest. The purpose of this paper is to discuss such a multiobjective definition, hereafter referred to as the MVP (multiobjective vending problem).

The MVP is defined conceptually as follows. A set of N nodes or demand points, each with a known reward-potential, is connected by links, each link with a known travel-penalty. The objectives are to find a circuit through a subset of the demand points in order simultaneously to maximize reward and minimize the accrued travel-penalty. No relationship is defined a priori between reward and penalty, and the two objectives must therefore be treated as noncommensurable.

The solution set

The trade-off curve between the two objectives can be represented graphically, as shown in figure 2. The axes represent the two objectives. The set of feasible solutions to the problem is confined to the shaded region in the lower right of the figure. A discrete subset of this feasible region can be identified as noninferior in the following sense:

> "A feasible solution to a multiobjective programming problem is noninferior if there exists no other feasible solution that will yield an improvement in one objective without causing a degradation in another objective" (Cohon, 1978, page 70).

The points representing the noninferior-solution set show how much of one objective must be traded off or sacrificed in order to obtain a specified gain in another objective.

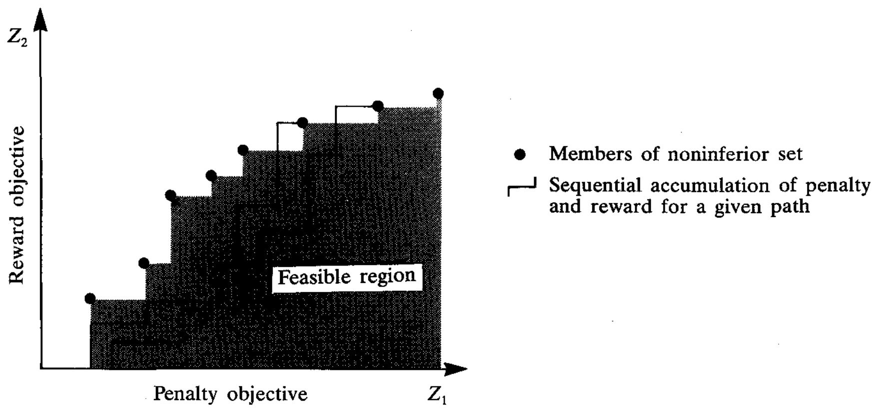

Figure 2. An example of an MVP noninferior-solution set.

Mathematical definition

The MVP problem can be expressed mathematically as follows.

$$\text{Maximize } Z = (Z_1, Z_2)\,, \tag{1}$$

$$Z_1 = \sum_{i=1}^{m} R_i\,, \qquad Z_2 = \sum_{i=1}^{m} P_{i,i+1}\,, \tag{2}$$

$$R_i > 0\,, \qquad P_{i,i+1} > 0\,, \tag{3}$$

where Z_1 is the total reward, Z_2 is the total penalty, R_i is the reward to be collected at node S_i, and $P_{i,i+1}$ is the penalty that must be accepted to travel from node S_i to node S_{i+1}. Also:

$$S_1 = S_{m+1} = S_\mathrm{d}\,, \tag{4}$$

$$S_i \in \{1, ..., N\}\,, \qquad \forall i\,, \tag{5}$$

$$S_i \neq S_j\,, \qquad \forall i, j\,, \quad i \neq j\,. \tag{6}$$

Equation (1) states that the overall objective, Z, consists of a combination of two single-objective functions, Z_1 and Z_2. The individual objective functions are merely listed; they are not added, multiplied, or combined in any other way as the relative importance of the objectives is not specified. S_i, the ith member of the route-sequence vector $\boldsymbol{S}$, stands for the ith node visited on the tour, and m represents the number of nodes actually visited. It will become obvious from the following discussion that the magnitude of m will be dependent on the solution obtained. Equation (6) ensures that no node is visited more than once, and equation (4) ensures that the tour begins and ends at the depot, enforcing a closed circuit. Subtours are not possible under this formalization. It is assumed that the underlying network is symmetrical.

Multiobjective methods and routing problems

A multiobjective approach to solving routing problems is not new. General discussions of multiobjective approaches to the analysis of spatial decisionmaking can be found in Nijkamp (1977) and Rietveld (1980). Multiobjective designs of transportation networks are discussed by Current and Min (1986). The maximum coverage/shortest path (MCSP) and maximum population/shortest path (MPSP) problems, discussed by Current (1981) and Current et al (1985b), are conceptually similar to the MVP. The problem in these cases is to find a path between specified origin and destination nodes. The first objective is to maximize the number of intervening nodes served or covered by the path, whereas the second is to minimize path length. It is only in exceptional circumstances that the optimal solution to the first objective is also the optimal solution to the second. In the general case, the objectives tend to be in conflict, and a number of optimal solutions will exist, each representing a specific trade-off relationship between the two objectives. The set of optimum solutions under all feasible trade-offs (the noninferior-solution set) is similar to that shown for the MVP in figure 2.

One of the first steps needed to solve MVPs is therefore to identify a method for generating the noninferior-solution set. Given the discrete nature of the problem of interest, the constraint method was adopted to solve the MVPs, the generation of the noninferior-solution set being broken into a number of single-objective problems. Preference for the constraint method is justified below.

Cohon (1978) discusses four alternative techniques for generating a noninferior-solution set, one of which, the 'multiobjective simplex' method, is a complex and intriguing mathematical problem, not entirely solved (Cohon, 1978; Keller, 1985). Current (1981) and Current et al (1985b) derive an approximation of the noninferior-solution set for MCSP and MPSP problems by utilizing the 'weighting' method. The two objectives are weighted by w and $(1-w)$, and are combined linearly as shown in equation (7), below:

$$Z = wZ_1 + (1-w)Z_2 \,. \tag{7}$$

The problem is solved repeatedly as a single-objective problem, with a selection of fixed values of w taken in the range 0 to 1 (see Cohon, 1978). The weighting method is most appropriately used where the weights themselves are of some importance in the interpretation of the results. However, it can give poor and inefficient coverage of the noninferior-solution set (Cohon, 1978), and its coverage may be incomplete in problems, such as the MVP, with discrete solution-spaces.

The 'noninferior-solution set estimation' (NISE) method, developed by Cohon (1978) relies on the assumption that the noninferior-solution set is convex. This clearly is not the case here, given the discrete nature of the problem addressed (see figure 2). The 'constraint' method (Cohon and Marks, 1975; Marglin, 1967) transforms a multiobjective problem into a finite number of single-objective problems by optimizing for one objective while constraining the other to a specified value. It is possible to examine the noninferior set by successively incrementing the constraining objective, in this case the maximum penalty, $P_{\max}$. With $P_{\max}$ set to zero, the circuit will be confined to the depot, and the reward will be equal to the reward from the depot. At the other extreme, an extremely large value of $P_{\max}$ will allow the circuit to include all demand-points, and the total reward available will be collected: the MVP in this case will always degenerate to the TSP. The number of noninferior solutions found will be: (a) equal to the number of values of $P_{\max}$ used, in the case of a problem with a continuous solution space; (b) less than or equal to the number of values of $P_{\max}$ used, in the case of a discrete problem, such as the MVP.

The MVP, which utilizes the constraint method, is a multiobjective generalization of the TSP, and can be broken into a number of single-objective problems. The next problem therefore concerns the search for a suitable procedure to find the single-objective solution. The search will commence with a discussion of the TSP literature.

Solution approaches

The TSP is a member of a class of difficult combinatorial problems, termed NP-hard (Papadimitriou, 1977), for which no algorithm is known that will execute in polynomial time, or in a time expressible as a polynomial function of the problem size (Papadimitriou, 1977; Garey and Johnson, 1979). Algorithmic solutions to the TSP have been reviewed by Bellmore and Nemhauser (1968), and Bodin et al (1983).

Exact-solution techniques for solving small TSPs include linear programming coupled with a branch and bound algorithm (Bellmore and Malone, 1971; Dantzig et al, 1954; 1959; Scott, 1971), and tree searching procedures (Crowder and Padberg, 1980; Held and Karp, 1971; Karp, 1977; Knuth, 1976). However, Bodin et al (1983) suggest that exact-solution techniques are impractical, given the current computing resources, for all but the smallest of TSPs. The MVP is a TSP with the added complexity that the set of nodes to be visited is not predetermined. Any attempt at utilizing an exact-solution procedure at present will therefore prove to be even less practical.

Heuristics offer a widely accepted alternative to exact-solution methods for the TSP, and a great variety have been developed, based on successive improvements to an initial, usually arbitrary, solution. One common process makes improvements on a working subset by adding or subtracting nodes from it, with the constraint that all nodes must be in the solution at the end of the process: an example is the 'steepest-ascent one-point-move' algorithm of Karg and Thompson (1964). A second common method of improvement is by shuffling the sequence of nodes in the working tour: examples of this include Bellmore and Nemhauser (1968), Cooper (1968), Golden et al (1980), Gupta (1978), and Lin and Kernigham (1973). A somewhat different 'divide and conquer' approach is used by Litke (1984). Points which are close together are gathered, by inspection, into clusters, each cluster containing no more than a specified number of points. An exhaustive search is then applied to identify the optimal path, both between and within clusters. Because the clustering is done by inspection, the algorithm can only be as good as the operator's ability to "see where the points are" (Litke, 1984, page 1229). Other heuristics have been developed for special cases of the TSP: for example, see Corpaneto et al, 1984; Cosmadakis and Papadimitriou, 1984; Garfinkel and Gilbert, 1978; Jongens and Volgenant, 1985. There are heuristics developed to solve the 'orienteering problem' (OP), a generalization of the TSP and conceptually similar to the MVP, by Tsiligirides (1984) and Golden et al (1985). Algorithms designed to solve OPs, namely the 'S-algorithm', 'D-algorithm' (Tsiligirides, 1984), and 'knapsack algorithm' (Golden et al, 1985) rely on the geometric nature of the OP, and do not enforce the condition of a Hamiltonian circuit. The MVP algorithm proposed in this paper was applied to a number of OP problems in order to evaluate its performance against the OP algorithms proposed by Tsiligirides (1984) and Golden et al (1985). The MVP algorithm was found to outperform heuristics written for the OP (Keller, 1988).

With this in mind, in the next section we describe a heuristic to solve the MVP. It incorporates the features of many of the approaches discussed above for the TSP, but it also has additional steps, designed to search for the optimal subset of nodes to be visited.

The MVP solution procedure

The MVP heuristic described below is the result of extensive experimentation, and it represents the most satisfactory combination of operations found. As with any heuristic, the particular sequence of operations which is most efficient for a given problem will depend on the specific attributes of the problem, thus it is always possible to fine tune the heuristic to a given data set. On the other hand, it is desirable that the heuristic can operate efficiently over as wide a set of problems as possible. As these two issues are necessarily in conflict, the approach we have taken is to construct a set of heuristic modules, or a 'toolbox', which can be controlled interactively by the user.

The sequence of logical operations underlying the heuristic is presented in the form of a flow chart in figure 3. Individual values of $P_{\max}$ are defined as follows:
(a) Identify the penalty received (P_{tot}) if all nodes are visited via some feasible route, but preferably by the route of the TSP solution.
(b) Specify the number of points (k) by which the noninferior-solution set is to be approximated.
(c) Commencing with $P_{\max} = 0$, increment $P_{\max}$ for each single-objective analysis, by an increment derived by dividing P_{tot} by k.
Once $P_{\max}$ is identified, the next step is to search for all demand-points, j, that can be visited directly from the depot without exceeding the penalty constraint:

$$P_{\mathrm{d}j} + P_{j\mathrm{d}} < P_{\max} , \tag{8}$$

where d denotes the depot. Every node that satisfies this condition is identified and placed in a vector, $\boldsymbol{M}$. This step excludes any nodes which cannot be reached within $P_{\max}$, and may therefore reduce the complexity of the problem, especially for small values of $P_{\max}$.

The next step is to generate some feasible starting-solution, $\boldsymbol{S}$, from the members in $\boldsymbol{M}$. Keller (1985) describes three possible approaches, an interactive approach, where the user intuitively identifies a starting-solution; and two random procedures, one with probabilities of selection that are proportional to the reward at each node, and the second with probabilities dependent on both reward and penalty. Any node that is now a member of the ordered-solution vector, $\boldsymbol{S}$, is removed from $\boldsymbol{M}$.

Once a feasible starting-solution has been generated, the heuristic implements two routines that attempt to reduce penalty by altering the route sequence while maintaining membership of the present route. The first routine (routine 1 in figure 3) searches for and eliminates self-crossing paths, which, by definition, cannot be contained in the optimal route. The path $\boldsymbol{S}$ is self-crossing if there exist two positions, i and j, within the sequence such that:

$$P_{i,i+1} + P_{j,j+1} > P_{i,j} + P_{i+1,j+1} . \tag{9}$$

Routine 2 sequentially drops every member in $\boldsymbol{S}$ out of its present position and inserts it at every alternative position within the route-sequence. Each cycle of the second routine identifies and implements that change which will result in the largest decrease in penalty.

Once these two routines detect no further possible reduction in penalty, the heuristic proceeds to search for possible increases in reward while remaining within the limit imposed by $P_{\max}$. This is achieved by attempting to alter the route-membership, by use of three strategies: 'one in—zero out' (routine 3); 'one in—one out' (routine 4); and 'one in—two out' (routine 5) in which the two nodes moved out are adjacent to each other in the route-sequence. In each case, all possible combinations of nodes in the 'in' (or $\boldsymbol{S}$) set and the 'out' (or $\boldsymbol{M}$) set, are tested. Higher levels of node exchange clearly exist, such as a 'two in—two out' exchange, but they become increasingly complex: to search all combinations in an 'x in—y out' exchange will require in the order of $m^y(N-m)^x$ steps.

If routines 3, 4, or 5 result in an improvement, the heuristic reverts to the first two penalty-reduction routines. If all five routines fail to make improvements, two further routines are called.

Experiments with the set of five routines, described thus far, showed that the heuristic tended to favour the inclusion of a remote node with high-reward potential over a cluster of low-reward nodes in close proximity to each other. This was found to be true even when the sum of all the rewards of the clustered nodes was larger than the reward-potential of the individual large node. This is because, in the absence of higher level node swaps as outlined earlier, a disproportionately large node, once it has entered the route-sequence, will not be dropped unless its

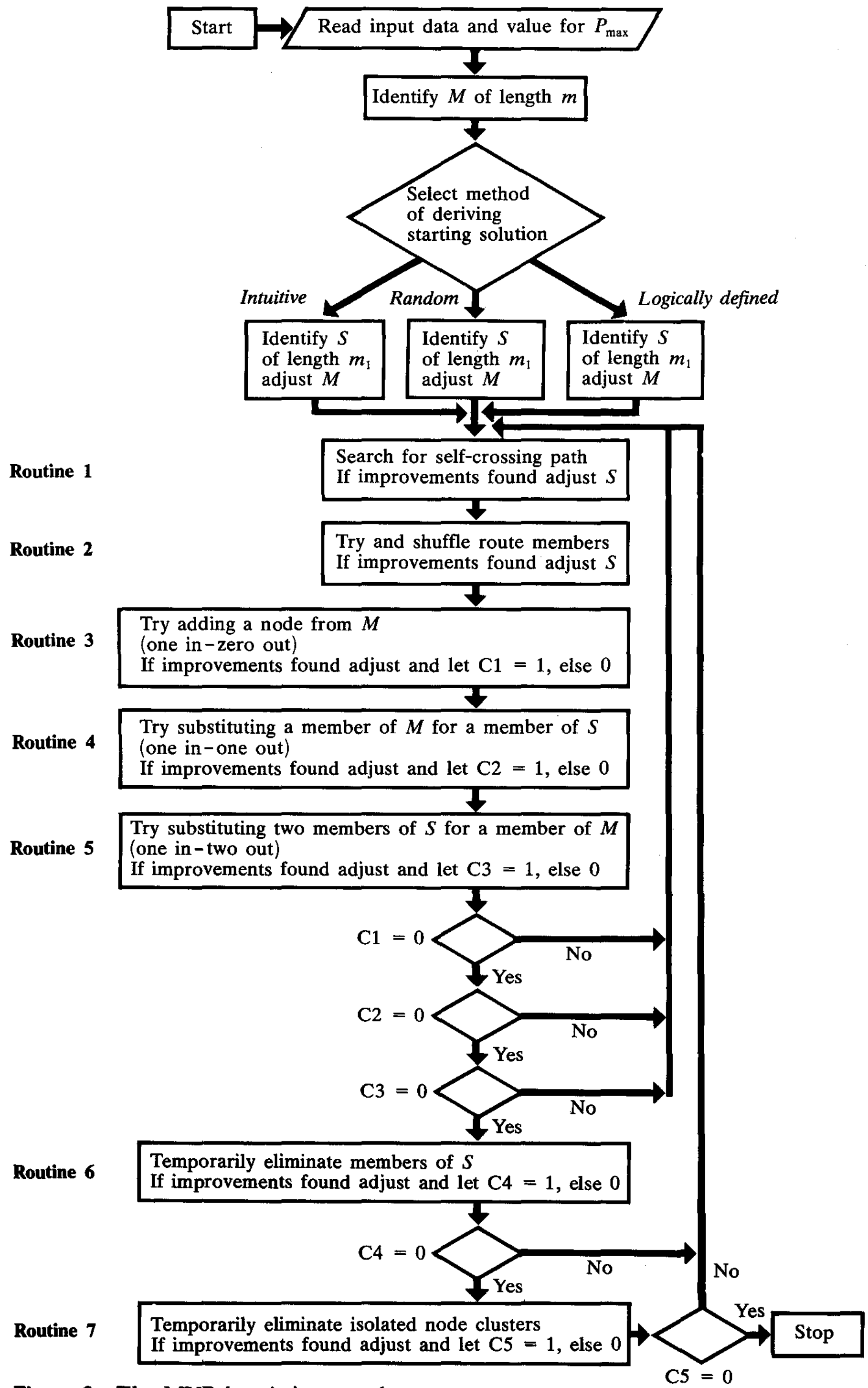

Figure 3. The MVP heuristic procedure.

reward-potential is less than the sum of the reward-potentials of at most two adjacent nodes. A further routine (routine 6) was therefore added. It operates by successively dropping, in turn, every node out of the present route-sequence, $\boldsymbol{S}$, but not returning it into $\boldsymbol{M}$ to be reconsidered for a swap. Each time a member of $\boldsymbol{S}$ is dropped, the remaining route is evaluated using the five routines described above. If an improvement is detected that yields a higher total reward-potential than the route that included the member just dropped, then the improvement is accepted as the new solution. The node temporarily removed from $\boldsymbol{S}$ is thereafter placed in $\boldsymbol{M}$. On the other hand, if no improvements can be detected, then the presently removed member is returned into its old place in the route, and the next member in the sequence is temporarily dropped.

This last routine is still incapable of detecting the type of situation illustrated in figure 4. Here, an isolated cluster of nodes located near a disproportionally large node (cluster of nodes A to E) can be replaced by another isolated cluster of nodes (cluster of nodes 6 to 8), a replacement resulting in an increase in total reward without exceeding the penalty constraint. None of the procedures described so far would detect this. Routine 7 deals with this situation by first finding the link associated with the highest penalty, called link 1. It subsequently finds the next highest penalty links in the route, before and after link 1—link 2 and link 3, respectively. Similarly, it identifies the next highest penalty links before and after link 2 and link 3—link 4 and link 5, respectively, and so on. The routine repeats this procedure until the pair of links found both connect into the depot. In the hypothetical example shown in figure 4, this occurs in the case of link 4 and link 5.

Based on the assumption that all nodes lying between link 2 and link 1 form a cluster of nodes removed far from the rest of the solution, the procedure continues by temporarily removing that cluster of nodes from $\boldsymbol{S}$. The first step is therefore to link the origin node of link 2 with the terminal node of link 1, temporarily omitting all nodes in between. A search is then made for possible improvement by means of the first five routines, as in the previous case, and if this is successful the omitted nodes are placed in $\boldsymbol{M}$; otherwise, it is returned to $\boldsymbol{S}$, and the nodes between link 1 and link 3 are temporarily removed, and the procedure is repeated as before. If no improvement results at any step, this routine will ultimately drop

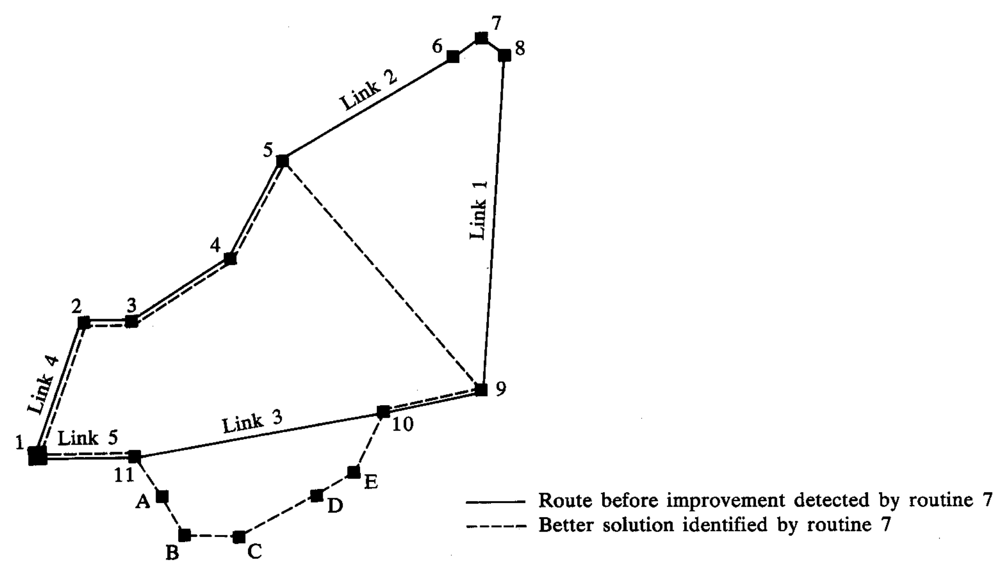

Figure 4. Route improvement undetected by node exchange.

all nodes of the last identified best route. Given a problem with a very large number of nodes constrained by a relatively small value of P_{max}, the result may be a completely new route.

Performance evaluation

An interactive program that solves the MVP by combining the constraint method and the MVP heuristic outlined above, was written in FORTRAN 77 and was tested on an IBM 4381 computer. To demonstrate and evaluate the performance of the heuristic, the program was applied to a set of twenty-five cities located in West Germany, as shown in figure 5(a). Bonn, the capital, was used as the depot and terminal node. The populations of the cities, shown in figure 6, were treated as surrogates for the reward that could be collected if the cities were visited. Intercity distances, also shown in figure 6, were used as surrogates for the penalties incurred for travelling between them.

Figure 7 shows a twenty-eight-solution approximation of the noninferior-solution set. Table 1 gives the total reward, total penalty, and exact route for each of the twenty-eight solutions.

Solutions (routes) 1 to 5 cover nodes in close proximity to the urban concentration of West Germany's Ruhr Valley. Solution 6 breaks from the Ruhr Valley cluster and includes the urban industrial concentration around Frankfurt. At solution 9, the maximum penalty has increased sufficiently to allow the inclusion of the large North Sea ports of Hamburg and Bremen, providing the Frankfurt cluster is dropped. Solution 13 includes the rather large but isolated city of West Berlin. At solution 18 the large city of Munich is included for the first time. Figure 5(b) shows the route of solution 28, which connects all twenty-five cities in a complete travelling salesman tour.

Thus far, it has been assumed that reward can be collected instantaneously upon arrival at a node. However, it may be more realistic in some applications to assume that penalties accrue both for travelling along the links and for collecting reward. The penalty for collecting reward might be taken to be a linear function, proportional to some constant, Q, multiplied by the size of the reward. Equation (2) can

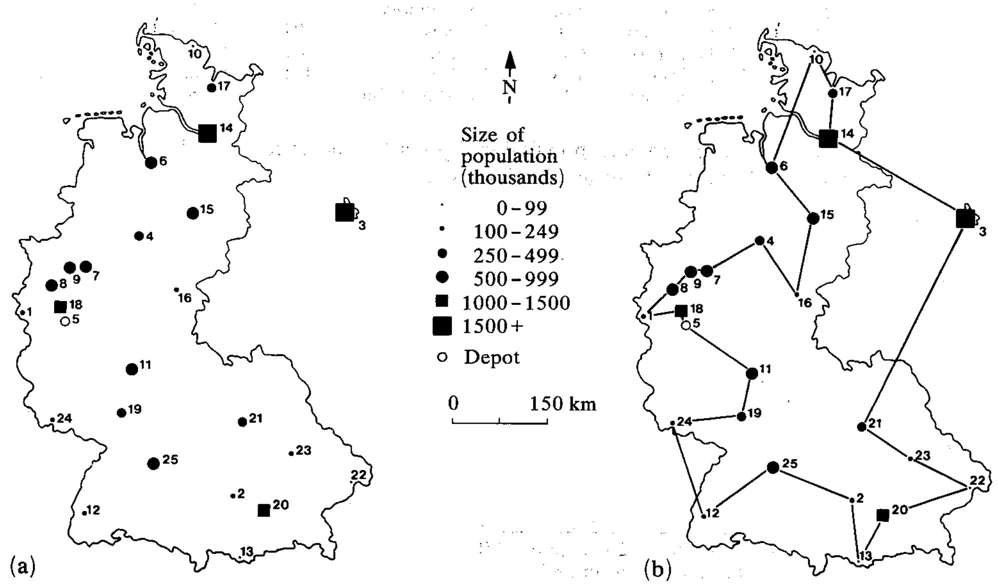

Figure 5. Twenty-five West German cities (a) their position, (b) and noninferior solution 28, which connects all twenty-five cities.

Distance matrix in km (distances are fastest, not necessarily the shortest)

Aachen 1	Augsburg 2	Berlin (West) 3	Bielefeld 4	Bonn 5	Bremen 6	Dortmund 7	Düsseldorf 8	Essen 9	Flensburg 10	Frankfurt 11	Freiburg 12	Garmisch 13	Hamburg 14	Hannover 15	Kassel 16	Kiel 17	Köln 18	Mannheim 19	München 20	Nürnberg 21	Passau 22	Regensburg 23	Saarbrücken 24	Stuttgart 25	City
																									1
597																									2
650	595																								3
267	557	394																							4
90	512	601	221																						5
386	693	393	153	360																					6
166	607	492	112	116	251																				7
75	571	569	189	76	316	66																			8
114	593	539	152	102	291	36	33																		9
691	886	457	438	645	236	530	595	570																	10
259	360	558	270	172	475	267	230	256	670																11
515	295	827	542	428	747	523	486	512	942	276															12
716	119	690	678	631	868	726	691	717	1060	479	348														13
524	719	290	271	478	119	363	428	403	167	503	775	893													14
367	578	285	114	321	113	212	277	252	323	351	628	748	156												15
347	434	381	123	265	284	175	236	210	477	201	473	558	310	165											16
630	825	388	377	584	225	469	534	509	83	609	881	999	115	262	416										17
69	530	572	190	28	329	85	48	74	608	190	446	649	441	290	250	547									18
319	285	636	348	234	553	329	292	318	746	82	201	404	579	434	279	685	252								19
634	66	592	619	571	770	666	629	655	963	419	354	98	795	650	496	901	589	344							20
469	138	441	449	380	600	475	438	464	793	215	383	257	626	481	326	735	398	258	165						21
694	251	653	674	605	825	700	663	689	1018	440	534	287	851	708	551	957	623	483	180	225					22
573	134	532	553	484	704	579	542	568	897	319	420	220	730	585	430	836	502	362	122	104	121				23
290	377	754	461	254	668	354	317	343	861	197	208	509	694	549	394	800	262	128	436	386	611	490			24
454	161	633	524	347	666	442	405	431	859	195	204	280	692	547	392	798	365	120	220	192	400	267	239		25

Population (thousands)

1 Aachen	242	8 Düsseldorf	659	14 Hamburg	1708	20 München	1312
2 Augsburg	248	9 Essen	674	15 Hannover	550	21 Nürnberg	496
3 Berlin (West)	1967	10 Flensburg	50	16 Kassel	204	22 Passau	40
4 Bielefeld	315	11 Frankfurt	631	17 Kiel	261	23 Regensburg	131
5 Bonn	284	12 Freiburg	174	18 Köln	1011	24 Saarbrücken	205
6 Bremen	571	13 Garmisch	30	19 Mannheim	312	25 Stuttgart	595
7 Dortmund	628						

Figure 6. Intercity distance matrix and population figures for the twenty-five West German cities studied. Source: Deutsche Centrale für Tourismus.

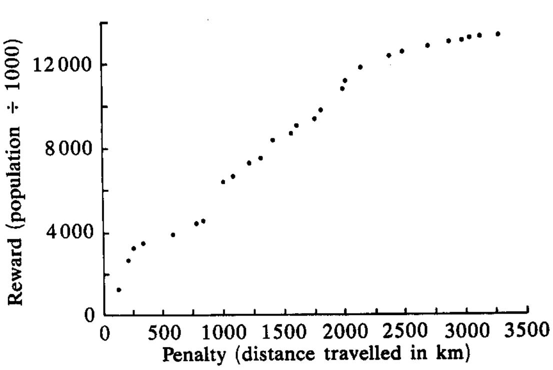

Figure 7. A 28-point approximation of the noninferior-solution set, for the twenty-five-node problem.

therefore be rewritten as follows:

$$Z_2 = \sum_{i=1}^{m} (P_{i,i+1} + QR_i) . \tag{10}$$

We can now evaluate the response of the noninferior-solution set to changes in the value of Q.

To evaluate the performance of the MVP heuristic, four additional noninferior-solution sets were derived for four different values of Q. Each of the sets, shown in figure 8, was approximated by sixteen solutions, obtained from sixteen values of P_{max}. To allow for an evaluation of the performance of the MVP heuristic, each of the 16×4, or 64, solutions was derived sixteen times, from independent starting-solutions. This resulted in $16 \times 16 \times 4$, or 1024, runs of the MVP heuristic. Details concerning the characteristics of the different starting-solutions can be found in Keller (1985). Of the sixteen runs made for every combination of P_{max} and Q, the one with the largest reward was treated as the optimal solution, and was allocated to the noninferior-solution set. The results obtained for each of the remaining fifteen runs were then evaluated against this best solution in order to obtain performance measures, which include the percentage of runs in which the best solution was found, the reward obtained in the worst solution encountered, and the ratio of the average penalty to the smallest penalty.

Table 1. Routes in the noninferior-solution set.

Route	Z_1	Z_2	Route sequence	Running time (s)
1	284	0	5	0.0
2	1295	56	5 18 5	0.1
3	2628	211	5 18 8 9 5	0.1
4	3256	258	5 18 7 9 8 5	0.1
5	3498	347	5 18 7 9 8 1 5	0.1
6	3887	584	5 18 8 9 7 11 5	0.1
7	4444	795	5 18 1 8 9 7 4 11 5	0.6
8	4692	841	5 18 6 15 4 7 9 8 5	0.8
9	6400	1003	5 18 4 15 14 6 7 9 8 5	1.2
10	6642	1092	5 18 4 15 14 6 7 9 8 1 5	0.7
11	7235	1223	5 18 8 9 7 4 6 14 15 16 11 5	1.3
12	7477	1319	5 18 1 8 9 7 4 6 14 15 16 11 5	1.5
13	8367	1422	5 18 4 15 3 14 6 7 9 8 5	1.8
14	8652	1573	5 18 8 9 7 4 6 14 3 16 11 5	2.3
15	8998	1623	5 18 8 9 7 4 15 6 14 3 11 5	3.2
16	9444	1738	5 18 1 8 9 7 4 6 14 3 15 16 11 5	2.9
17	9736	1817	5 18 1 8 9 7 4 15 6 14 3 21 11 5	5.6
18	10763	2003	5 18 8 9 7 15 14 3 21 20 2 25 11 5	6.9
19	11099	2019	5 18 8 9 7 4 6 14 3 21 20 2 25 11 5	7.5
20	11780	2154	5 18 8 9 7 4 15 6 14 3 21 23 20 2 25 11 5	3.7
21	12334	2394	5 18 1 8 9 7 4 15 6 14 3 21 23 20 2 25 11 19 5	2.6
22	12539	2504	5 18 1 8 9 7 4 15 6 14 3 21 23 20 2 25 24 19 11 5	5.3
23	12800	2717	5 18 1 8 9 7 4 15 6 14 17 3 21 23 20 2 25 24 19 11 5	7.8
24	13004	2891	5 18 1 8 9 7 4 16 15 6 14 17 3 21 23 20 2 25 24 19 11 5	3.3
25	13054	2993	5 18 1 8 9 7 4 16 15 6 10 17 14 3 21 23 20 2 25 24 19 11 5	2.9
26	13178	3217	5 18 1 8 9 7 4 16 15 6 14 17 3 21 23 20 2 25 12 24 19 11 5	3.2
27	13268	3345	5 18 1 8 9 7 4 16 15 6 10 17 14 3 21 23 22 20 2 25 12 24 19 11 5	2.2
28	13298	3496	5 18 1 8 9 7 4 16 15 6 10 17 14 3 21 23 22 20 13 2 25 12 24 19 11 5	0.9

Note: The total reward, Z_1, is the sum of the populations of the cities visited; the total penalty, Z_2, is expressed in kilometers travelled.
The route-sequence numbers identify the city and the order in which they were 'visited'. Refer to figure 5 for full identification.

The probability of obtaining the best answer, in any one run of the heuristic, averaged 0.74, reaching a low of 0.67 for one starting-technique, and a high of 0.89 for another. The worst departure from the best solution of all 1024 runs was 0.21, in terms of the reward objective. On average, the worst solutions encountered in each set of sixteen repeat runs, obtained rewards 9% lower than the best solutions. For all runs, the average decrease in reward compared with the best solution was 2%. Table 1 shows the running times for the heuristic on an IBM 4381 computer.

The dominant objective of this paper is to introduce a practical, but difficult, generalization of the TSP, and to show that it can be solved (even if not optimally). A thorough examination of the heuristic is beyond the length of this paper, but a single-sample problem is clearly not adequate to test in detail the performance of a heuristic. A comparison with other tested data-sets is required. A more detailed discussion of the performance of the heuristic against the above data set can be found in Keller (1985). A comparison of the heuristic to a number of score OPs has been discussed by Keller (1988).

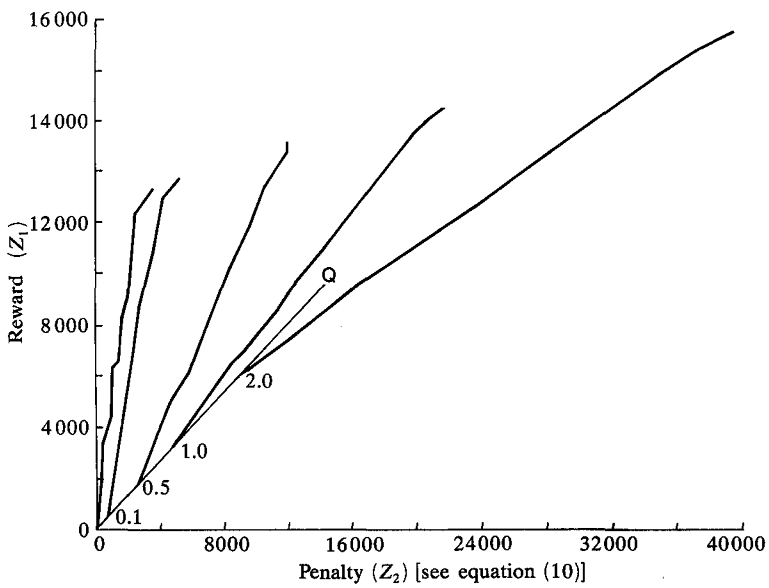

Figure 8. Generalized noninferior-solution sets, for various Q.

Discussion

Any heuristic offers a trade-off between computational complexity and performance. The performance of the MVP heuristic could be improved by including higher level exchanges, but only at the expense of substantial increase in computing costs. The decision to exclude them resulted in considerable loss of reward in a few cases, but in general the rewards associated with inferior solutions were very close to the best ones. The probability of reaching the best solution (found at least once in a number of runs by means of different starting techniques) was found to be high. The performance of the heuristic was therefore judged to be satisfactory.

The mathematical model used to represent and solve a real problem is always a simplification, because it excludes unspecified or unquantifiable, but nevertheless important, criteria. The solution derived may not represent the real-world ideal, but rather, it may function as an indicator of a good solution given certain assumptions. The MVP program was written in an interactive mode with this point in mind, and gives the user the opportunity to evaluate the sensitivity of solutions to the assumptions made, or to modify any solution in response to new criteria. The user can evaluate the effects of changes in the route-sequence or in the data from which it is derived.

The MVP was introduced as a form of generalization of the TSP, which is appropriate in many practical situations. Further generalizations are possible, and many can be incorporated into the solution procedure with little difficulty. For example, the reward potentially available at a demand-node is often time dependent. A vendor may find that the penalty of staying at one demand-point is a linear function of time, but that the reward obtained shows diminishing returns as the potential market is exhausted. It is also possible that the initial reward obtained on arrival, shows an increasing return as the market becomes aware of the presence of the vendor. Another form of time dependency occurs when reward varies both in response to the vendor's arrival and to the time of day. Both forms are present in the example of a mobile lunch-vendor, for whom certain times of day, namely conventional mealtimes, offer greater potential reward, and for whom the rate of reward first increases and then decreases following arrival at a site. Mobile-library branches, health clinics, and other public services appear to have similar characteristics. Another example concerns the planning of a political candidate's canvass tour, before an election. The politician has only a limited time-period in which to canvass; the candidate must canvass in as many locations as possible during the time when the impact will be greatest. Such a time dependent version of the MVP has been explored by Keller (1985).

Acknowledgement. This research was supported in part by NSERC, grant number A6533.

References

Bellmore M, Malone J, 1971, "Pathology of travelling salesman subtour-elimination algorithms" *Operations Research* **19** 278-307

Bellmore M, Nemhauser G L, 1968, "The travelling salesman problem" *Operations Research* **16** 538-558

Bodin L, Golden B, Assad A, Ball M, 1983, "Routing and scheduling of vehicles and crews: the state of the art" *Computers and Operations Research* special issue **10**(2) 63-211

Bromley R J, Symanski R, Good C M, 1975, "The rationale of periodic markets" *Annals of the Association of American Geographers* **65** 530-531

Clarke G, Wright J W, 1964, "Scheduling of vehicles from a central depot to a number of delivery points" *Operations Research* **12** 567-581

Cohon J L, 1978 *Mathematics in Science and Engineering, Volume 140: Multiobjective Programming and Planning* (Academic Press, London)

Cohon J L, Marks D, 1975, "A review and evaluation of multiobjective programming techniques" *Water Resources Research* **11** 208-220

Cooper L, 1968, "An extension of the generalised Weber problem" *Journal of Regional Science* **8** 181-197

Corpaneto G, Martello S, Toth P, 1984, "An algorithm for the bottleneck travelling salesman problem" *Operations Research* **32** 380-389

Cosmadakis S S, Papadimitriou C H, 1984, "The travelling salesman problem with many visits to few cities" *SIAM Journal of Computing* **13**(1) 99-108

Crowder H, Padberg M, 1980, "Solving large scale symmetric travelling salesman problems to optimality" *Management Science* **26** 495-509

Current J R, 1981 *Multiobjective Design of Transportation Networks* unpublished PhD dissertation, Department of Geography and Environmental Engineering, Johns Hopkins University, Baltimore, MD

Current J R, Cohon J, ReVelle C, 1984, "The shortest covering path problem: an application of location constraints to network design" *Journal of Regional Sciences* **24** 161-183

Current J R, Min H, 1986, "Multiobjective design of transportation networks: taxonomy and annotations" *European Journal of Operations Research* **26** 187-201

Current J R, ReVelle C, Cohon J, 1985a, "The application of location models to the multiobjective design of transportation network" *Regional Science Review* **14**

Current J R, ReVelle C, Cohon J, 1985b, "The maximum covering/shortest path problem: a multiobjective network design and routing formulation" *European Journal of Operations Research* **21** 189-199
Dantzig G B, Fulkerson D R, Johnson S M, 1954, "Solution of a large scale travelling salesman problem" *Operations Research* **2** 393-410
Dantzig G B, Fulkerson D R, Johnson S M, 1959, "On a linear programming combinatorial approach to the travelling salesman problem" *Operations Research* **7** 58-66
Garey M R, Johnson D S, 1979 *Computers and Intractability: A Guide to the Theory of NP Completeness* (W H Freeman, San Francisco, CA)
Garfinkel R S, Gilbert K C, 1978, "The bottleneck travelling salesman problem: algorithms and probabilistic analysis" *Journal of the Association for Computing Machinery* **27** 1227-1235
Ghosh A, 1982, "A model of periodic marketing" *Geographical Analysis* **14** 155-166
Golden B L, Bodin L, Doyle T, Stewart W Jr, 1980, "Approximate travelling salesman algorithms" *Operations Research* **28** 694-711
Golden B, Levy L, Vohra R, 1985, "The orienteering problem" WP series MS/S 85/041, College of Business and Management, University of Maryland at College Park, MD 20742, USA
Gupta J, 1978, "A search algorithm for the travelling salesman problem" *Computers and Operations Research* **5** 243
Hay A M, 1971, "Notes on the economic basis for periodic markets in developing countries" *Geographical Analysis* **3** 393-401
Held M, Karp R M, 1971, "The travelling salesman problem and minimum spanning trees: part II" *Mathematical Programming* **1**(1) 6-25
Jongens K, Volgenant T, 1985, "The symmetric clustered travelling salesman problem" *European Journal of Operations Research* **19** 68-75
Jonker R, Volgenant T, 1984, "Nonoptimal edges for the symmetric travelling salesman problem" *Operations Research* **32** 837-846
Karg L L, Thompson G L, 1964, "A heuristic approach to solving travelling salesman problems" *Management Science* **10** 225-248
Karp R M, 1977, "Probabilistic analysis of partitioning algorithms for the travelling salesman problem" *Mathematics and Operations Research* **2** 209-224
Keller C P, 1985 *Multiobjective Routing Through Space and Time: The MVP and TDVP Problems* unpublished PhD thesis, Department of Geography, The University of Western Ontario, London, Ont, Canada
Keller C P, 1986, "Applications of generalized travelling salesman problems to geography", paper presented at the Annual Meeting of the Association of American Geographers, Portland, OR, available from Department of Geography, University of Victoria, BC
Keller C P, 1988, "Algorithms to solve the orienteering problem: a comparison", occasional working paper, Department of Geography, University of Victoria, Victoria, BC (forthcoming *European Journal of Operational Research))*
Knuth D E, 1976, "Mathematics and computer science: coping with finiteness" *Science* **194**(4271) 1235-1242
Lin S, Kernigham B, 1973, "An effective heuristic algorithm for the travelling salesman problem" *Operations Research* **21** 498-516
Litke J D, 1984, "An improved solution to the travelling salesman problem with thousands of nodes" *Communications of the ACM* **27** 1227-1235
Marglin S, 1967 *Public Investment Criteria* (MIT Press, Cambridge, MA)
Nijkamp P, 1977 *Theory and Application of Environmental Economics* (North-Holland, Amsterdam)
Orloff C, 1974, "Routing a fleet of *M* vehicles to/from a central facility" *Networks* **4** 147-162
Papadimitriou C H, 1977, "The euclidean travelling salesman problem is NP-complete" *Theoretical Computer Science* **4** 237-244
Rietveld P, 1980 *Studies in Regional Science and Urban Economics, Volume 7: Multiple Objective Decision Methods and Regional Planning* (North-Holland, Amsterdam)
Scott A J, 1971 *Combinatorial Programming, Spatial Analysis and Planning* (Methuen, Andover, Hants)
Tsiligirides T, 1984, "Heuristic methods applied to orienteering" *Journal of the Operations Research Society* **35** 797-809
Webber M J, Symanski R, 1973, "Periodic markets: an economic location analysis" *Economic Geography* **49** 213-227