The orienteering problem

Article *in* Naval Research Logistics · June 1987

DOI: 10.1002/1520-6750(198706)34:3<307::AID-NAV3220340302>3.0.CO;2-D

3 authors, including:

Bruce Golden
University of Maryland, College Park

Rakesh V. Vohra
University of Pennsylvania

The Orienteering Problem

Bruce L. Golden, Larry Levy, and Rakesh Vohra*
College of Business and Management, University of Maryland, College Park, Maryland 20742

Orienteering is a sport in which start and end points are specified along with other locations. These other locations have associated scores. Competitors seek to visit, in a fixed amount of time, a subset of these locations on the way from the start point to the end point in order to maximize the total score. An effective center-of-gravity heuristic is presented that outperforms heuristics from the literature.

INTRODUCTION

Orienteering is an outdoor sport that is usually played in heavily forested areas. Located in the forest are a number of "control points" each with an associated score. Competitors armed with compass and map are required to visit a subset of the control points from the start point (node 1) so as to maximize their total score and return to the end point (node n) within a prescribed amount of time. Competitors who arrive late are either disqualified or charged a severe penalty. In this paper we shall assume that competitors who arrive late are disqualified.

As the distance and travel time between any pair of control points are determined by the geography, we will assume that these are known quantities. Thus, a simplified version of orienteering can be formulated in the following way. Given n nodes in the Euclidean plane each with a score $s(i) \geq 0$ [note that $s(1) = s(n) = 0$], find a route of maximum score through these nodes beginning at 1 and ending at n of length (or duration) no greater than TMAX. Tsiligirides [4] refers to this as the generalized traveling salesman problem (GTSP).

COMPLEXITY OF THE GTSP

A problem is *NP* hard (see Garey and Johnson [1]) if the existence of a polynomially bounded algorithm for it implies the existence of a polynomially bounded algorithm for all *NP*-complete problems. The GTSP clearly falls within the class of *NP*-hard problems, as it contains the well-known traveling-salesman problem as a special case. To see this, suppose we had a polynomial algorithm for solving the GTSP. That is, given TMAX and a collection of nodes and scores, our algorithm would determine in polynomial time an appropriate route that maximized total score. Consider any instance of the TSP—or at least its recognition version. The recognition version asks the question: Given a collection

*Current address: Ohio State University, Columbus, OH 43210.

Naval Research Logistics, Vol. 34, pp. 307–318 (1987)
 CCC 0028-1441/87/030307-12$04.00

of nodes, is there a tour of length less than T through all the nodes? Assign every node a score of 1 and set TMAX = T. Choose any node to be both the start and end point. Apply our algorithm for the GTSP to this problem. If the resulting score is equal to the number of nodes, then the answer to the TSP is yes, otherwise it is no.

In view of the above, one is led to consider heuristics for the GTSP. In this paper we consider several heuristics for solving this problem. The first two are due to Tsiligirides [4] and are described next. The third is one of our design. We have tested the new heuristic on a set of problems suggested by Tsiligirides and have compared its performance with the first two heuristics. The new heuristic seems to outperform the earlier ones.

HEURISTICS FROM THE LITERATURE

Two heuristic approaches for solving the orienteering problem have been proposed by Tsiligirides—a stochastic algorithm and a deterministic algorithm. In this section, we briefly describe these methods.

The stochastic algorithm relies on Monte Carlo techniques to build a large number of routes, and it selects the best from these. The driving force behind this approach is a measure $A(j)$ of "desirability" for all nodes j not currently on the route. In particular, Tsiligirides uses $A(j) = (s(j)/(t(\text{last},j)))^{4.0}$, where $s(j)$ is the score associated with node j and $t(\text{last},j)$ is the travel time from the last node chosen to node j. After determining the largest four values for $A(j)$ (fewer if four nodes are not eligible), the values are normalized so that they sum to one. A random number between 0 and 1 is then generated in order to select node j for inclusion. This procedure is repeated until no additional nodes can be included on the route. Since he is using random numbers, Tsiligirides is able to generate many routes for each value of TMAX and choose the one with the largest total score.

The deterministic algorithm creates routes using a variant of the Wren and Holliday [5] vehicle-routing procedure. This heuristic divides the geographic region into sectors determined by concentric circles. Routes are built up within sectors in an effort to minimize total travel time. By varying the radii of the circles and by rotating the axes, Tsiligirides examines 48 cases for each value of TMAX.

A NEW CENTER OF GRAVITY HEURISTIC

At the heart of the new heuristic is a systematic progression from one center of gravity to another. Each center of gravity has an associated node set and resulting route. The progression continues until a stopping rule is satisfied. At that point, the best route found is recorded as the solution.

The heuristic is made up of the following three steps:

(1) Route construction step,
(2) Route improvement step,
(3) Center-of-gravity step.

Steps 1 and 2 are used to generate a starting route from which the next portion of the heuristic begins. We now discuss the steps in more detail.

Route Construction Step

In this step the objective is to find a route that begins at 1 and ends at n which has a relatively high score while requiring less than TMAX units of time.

Essentially, our approach applies an insertion heuristic which examines "bang-for-buck" ratios at each step. In other words, nodes with high scores that do not add too much in duration to the already existing route are ideal candidates for insertion. The details are provided in the appendix.

Route Improvement Step

In this step one takes the route generated in the previous step and uses an interchange procedure such as 2-OPT to find a shorter route on the same set of nodes. This is followed by a cheapest insertion step in which as many nodes as possible are inserted into the route obtained thus far without violating TMAX. Call the route that results L.

Center of Gravity Step

Suppose now that node i has coordinates $(x(i),y(i))$. In this step, we calculate the center of gravity of L as $g = (\bar{x},\bar{y})$, where

$$\bar{x} = \sum_{i\in L} s(i)x(i) \Big/ \sum_{i\in L} s(i),$$

$$\bar{y} = \sum_{i\in L} s(i)y(i) \Big/ \sum_{i\in L} s(i).$$

Let $a(i) = t(i,g)$ for $i = 1,2, \ldots ,n$. Next a route including nodes 1 and n is formed as follows:

(a) Calculate the ratio of $s(i)/a(i)$ for all i.

(b) Add nodes to the route in decreasing order of this ratio, using cheapest insertion, until no additional nodes can be added without exceeding TMAX.

(c) Use the route improvement step to make adjustments to the resulting route.

We now have a route L'_1. This route's center of gravity gives rise to a repetition of (a)–(c). The resulting route is denoted by L'_2. This process is repeated until a cycle develops, that is, route L'_p and L'_q are identical for some $q > p$. Finally, we select the route that has the highest score among the routes $\{L,L'_1,L'_2, \ldots ,L'_q\}$. The current implementation requires that $q \leq 10$. In other words, within this step the center of gravity is computed at most a small number of times. In practice, we have never observed q in excess of 5.

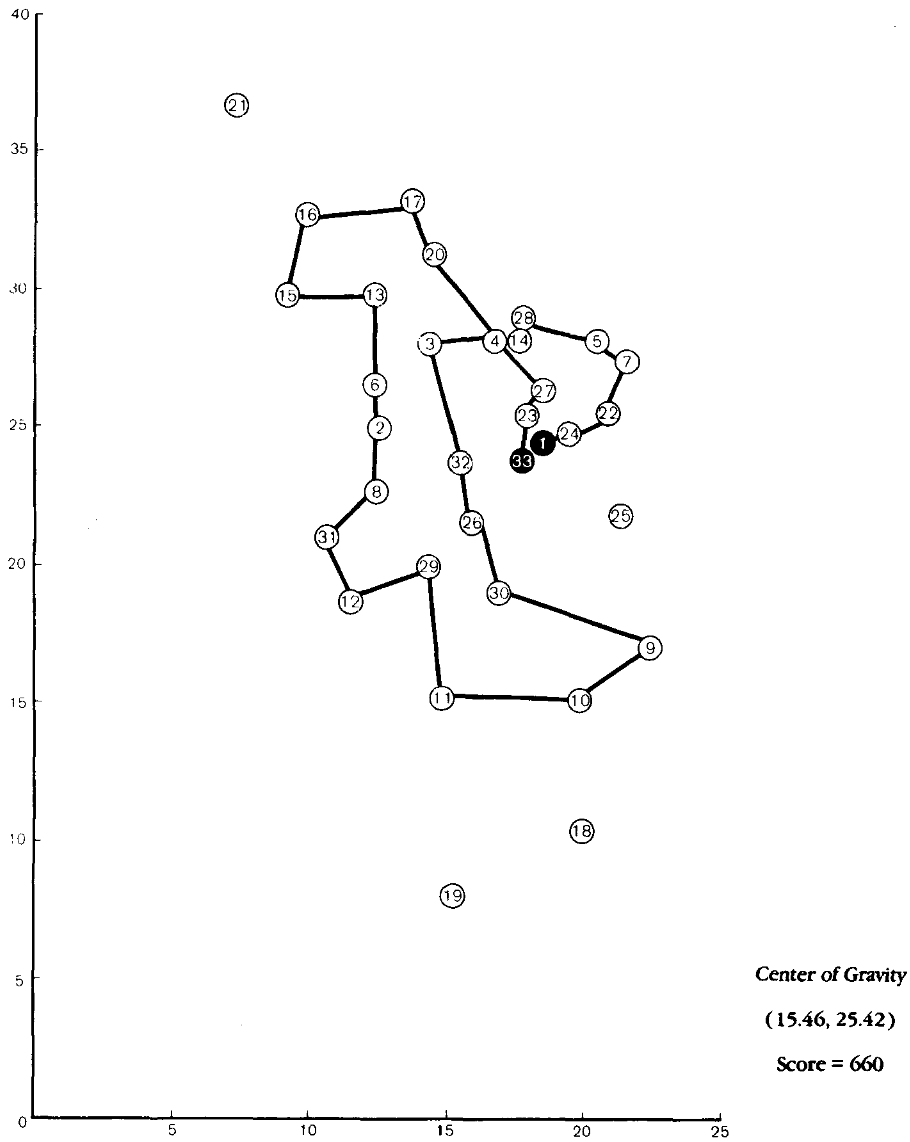

Figure 1. Route construction step (TMAX = 80).

The steps are illustrated in Figures 1–5. These figures represent a 33-node problem with TMAX set to 80. On each figure the center of gravity and the total score are displayed.

In going from Figure 1 to Figure 2, we notice that the center of gravity changes and the score improves slightly. The first center-of-gravity solution (Figure 3) provides a significantly better score. The second center-of-gravity solution (Figure 4) is not as good as the first. The third Center-of-Gravity solution is the same as the first.

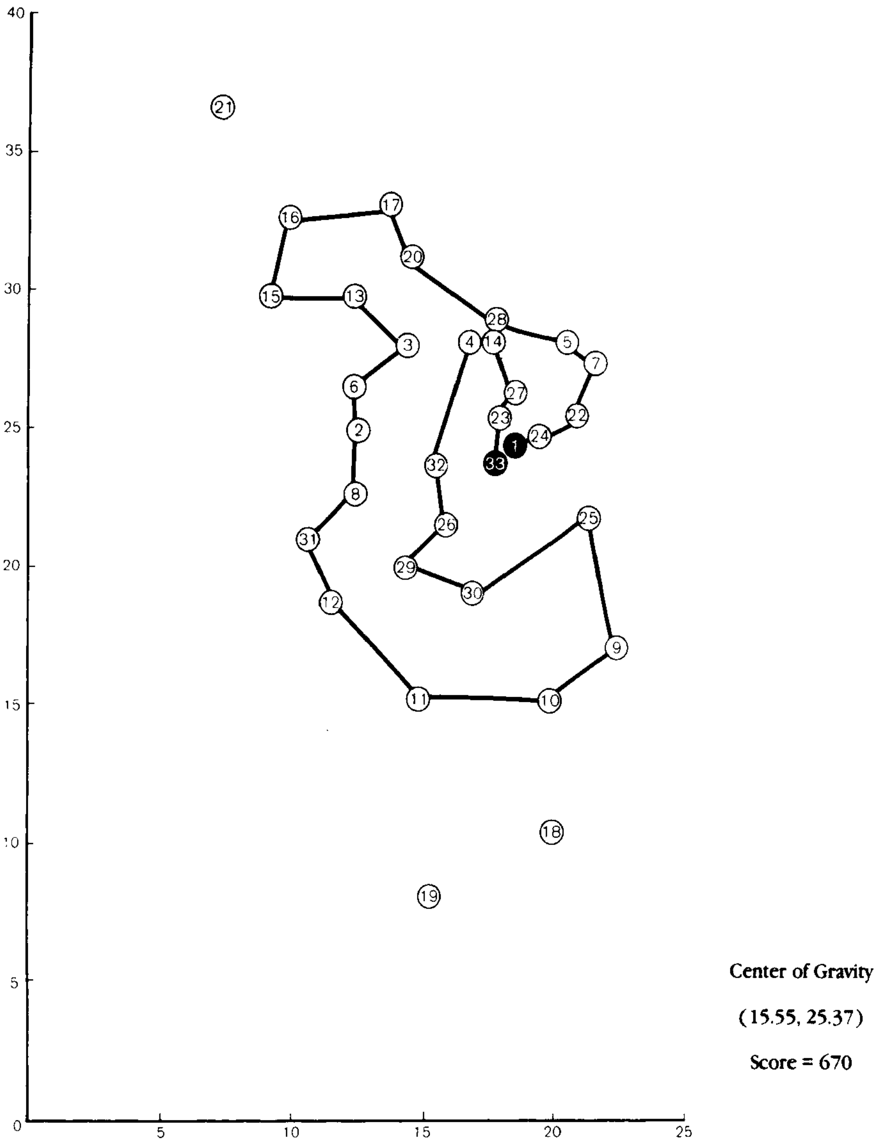

Figure 2. Route improvement step (TMAX = 80).

COMPUTATIONAL EXPERIENCE

In evaluating the performance of the center-of-gravity heuristic, results are compared with the deterministic (D) algorithm and the stochastic (S) algorithm, both due to Tsiligirides. Although running times are not reported in his paper, we estimate the running times to be similar to that of the proposed heuristic.

In Tables 1–3 we present Tsiligirides's three problems. The new heuristic results are presented in three stages—after route construction, route improve-

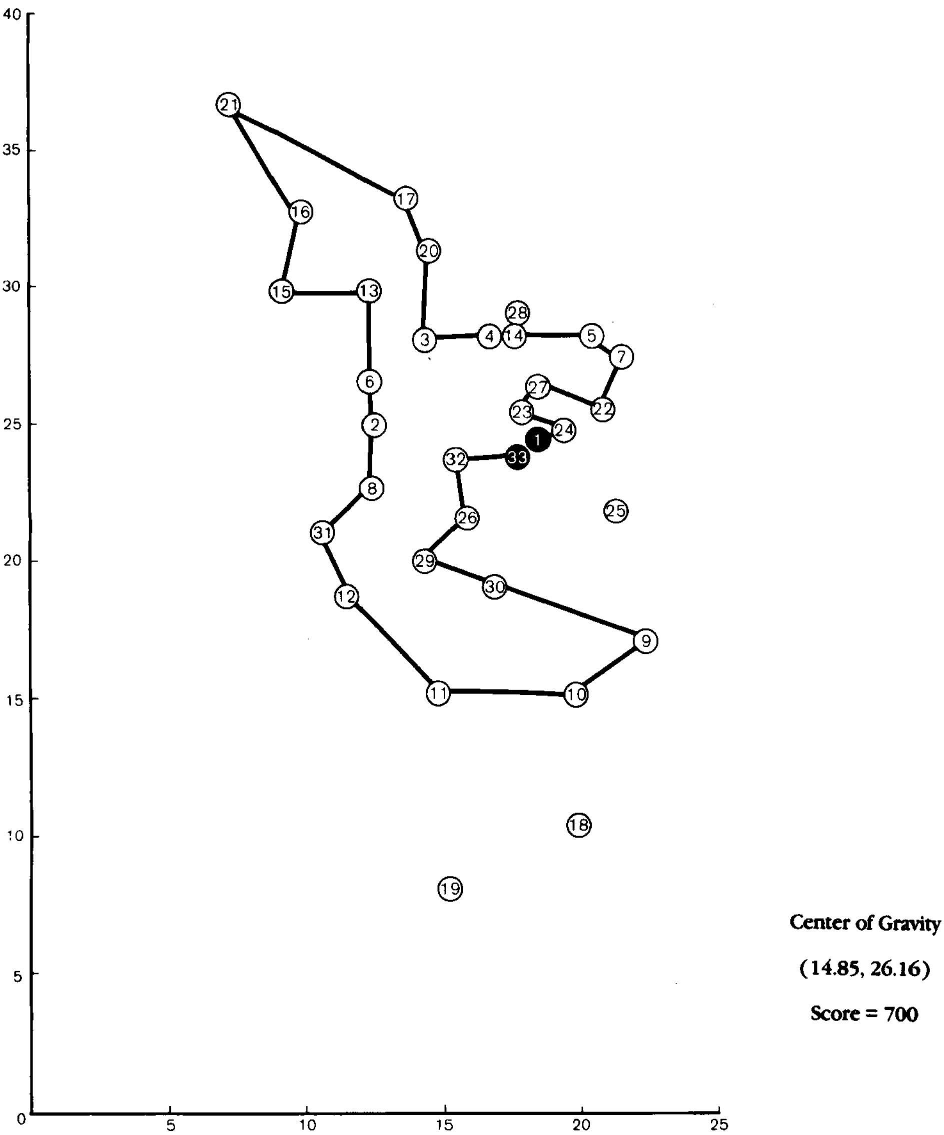

Figure 3. First center-of-gravity solution (TMAX = 80).

ment, and the center-of-gravity step. The final column represents the length of the best route found at each TMAX value. The proposed heuristic, after just the route construction step, is superior to the *D* algorithm for all values of TMAX across all three problems.

In making comparisons with the *S* algorithm, we note that, of the 49 problem instances examined in Tables 1–3, the center-of-gravity heuristic outperforms the *S* algorithm 25 times and ties 11 times. In specific cases where the *S* algorithm is superior, the center-of-gravity solution is not far off, whereas when the center of gravity procedure is better its score often exceeds that of the *S* algorithm by a wide margin.

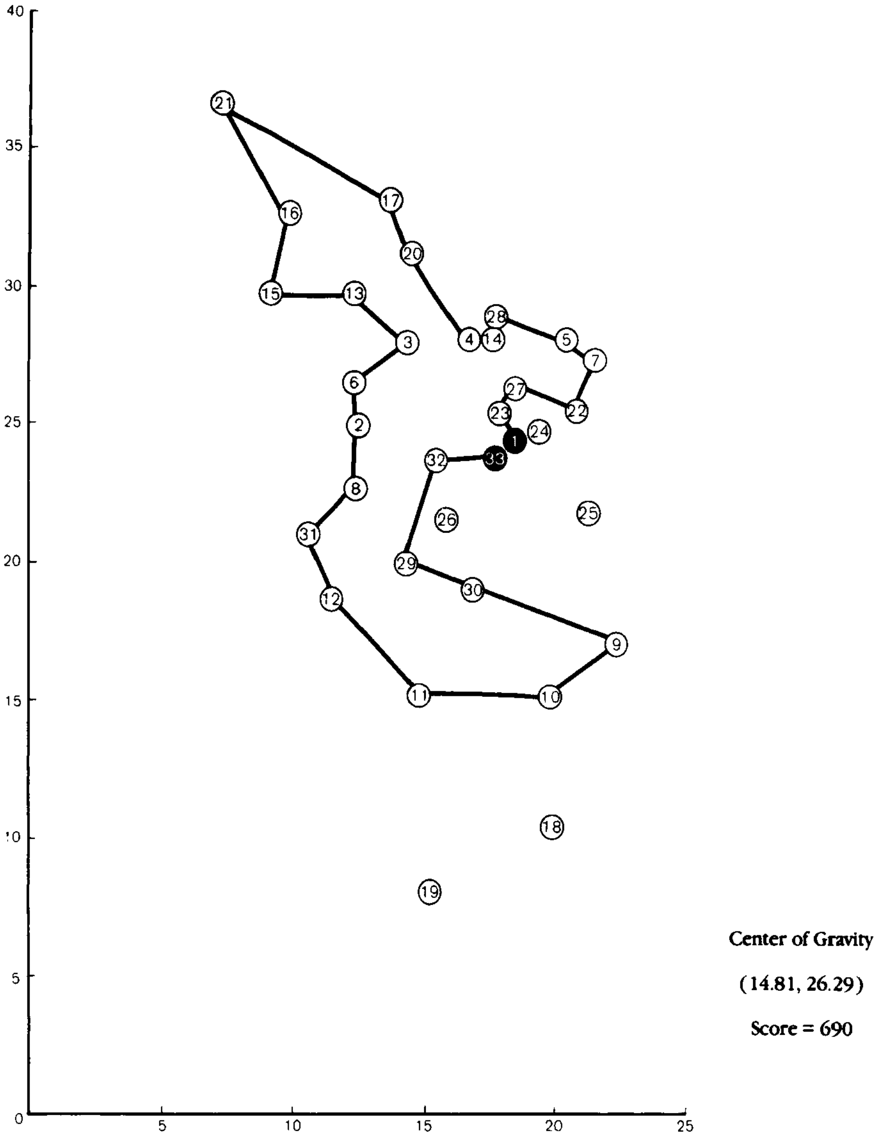

Figure 4. Second center-of-gravity solution (TMAX = 80).

The center-of-gravity heuristic was written in FORTRAN 77 and run on a UNIVAC 1190. Running times for the three problems and each value of TMAX are shown in Table 4. Also given are the average running times for each particular problem over the various levels of TMAX employed. Although the code was not implemented in the fastest possible way, the running times are quite reasonable. Rarely did any instance of a problem take longer than eight seconds to execute, and in the worst case running time was under 10 seconds.

Table 5 displays the percentages of total running time that each step of the algorithm accounts for. About two-thirds of the time is consumed in obtaining an initial solution. This step is extremely important, since it provides the initial

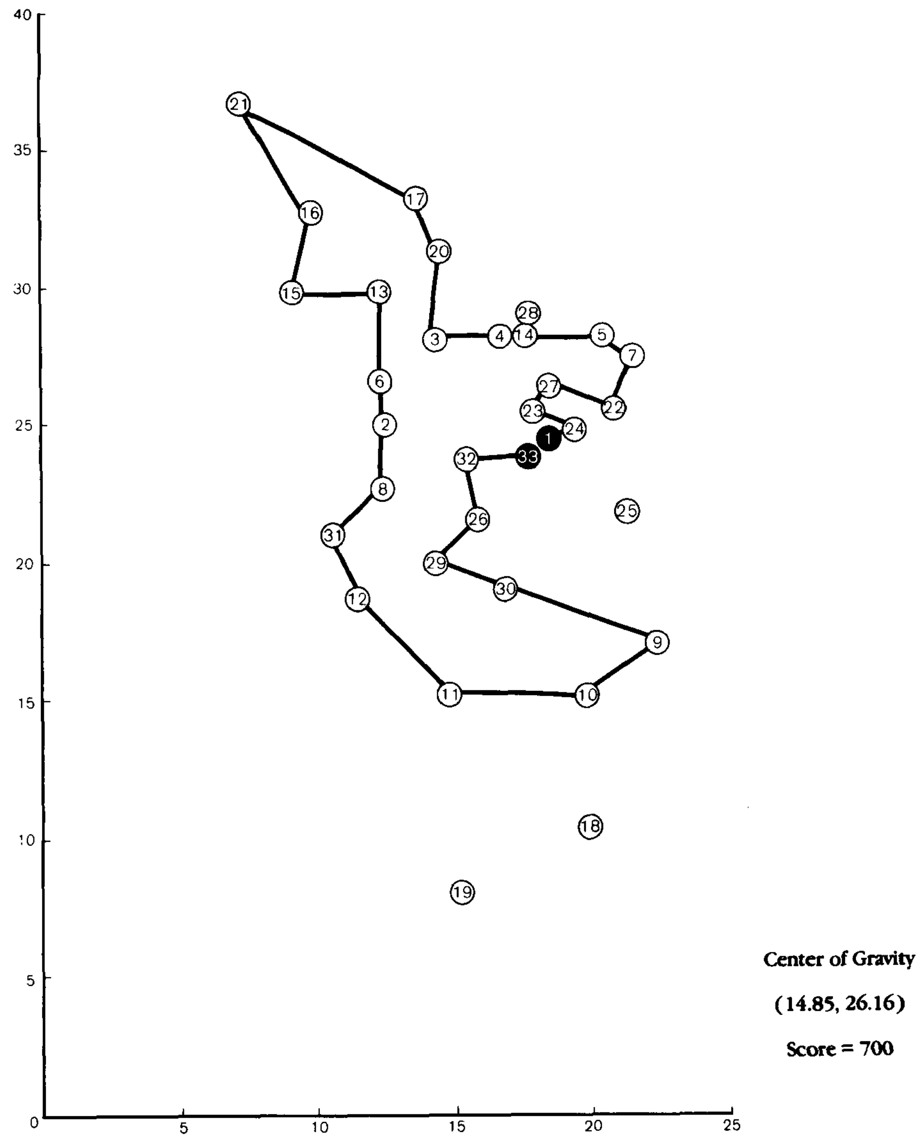

Figure 5. Third center-of-gravity solution (TMAX = 80).

center of gravity. A very small fraction of the time is devoted to route improvement. In addition, it appears that the center-of-gravity step is inexpensive computationally. From Tables 1–3, we see that the center-of-gravity step improves upon the second step in about 20% of the cases, sometimes significantly.

RELATED PROBLEMS

In previous work (see Golden *et al.* [2]) we have studied a large and complex inventory/routing problem in which a fleet of trucks must deliver fuel to a large number of customers on a daily basis. A key feature of this problem is that a customer's supply of fuel (inventory level) must be maintained at an adequate

Table 1. Problem 1 results.

TMAX	*D* algorithm	*S* algorithm	Route construction	Route improvement	Center of gravity	Length
5	...	10	10	10	10	4.14
10	...	15	15	15	15	6.87
15	...	45	45	45	45	14.26
20	40	65	65	65	65	19.85
25	65	90	90	90	90	24.88
30	80	110	110	110	110	29.88
35	105	135	125	125	125	33.60
40	105	150	140	140	140	39.87
46	130	175	165	165	165	45.85
50	140	190	175	180	180	49.92
55	160	205	195	200	200	54.38
60	175	220	205	205	205	59.40
65	200	240	210	220	220	64.69
70	200	255	230	230	240	69.89
73	205	260	240	240	255	72.27
75	210	270	250	250	260	74.42
80	220	275	260	260	275	79.57
85	235	280	275	275	285	81.78

level at all points in time. That is, each customer has a known tank capacity and its fuel level is expected to remain above a prespecified critical value which may be called the resupply point (e.g., 30% of tank capacity). The deliveries follow a "push system" in the sense that they are scheduled by the firm based on a forecast of the customers' tank levels. Stockouts are costly and are to be avoided whenever possible.

One can easily translate forecasted tank level into a measure of urgency for each customer. The higher the urgency measure, the more important it is to service that customer immediately. A primary goal is to select a subset of customers to visit each day that urgently require service and that are clustered geographically in such a way as to foster the construction of efficient truck trips. The orienteering problem can be used to solve the subset selection step in the larger inventory/routing problem (replace urgency with score). See Golden *et*

Table 2. Problem 2 results.

TMAX	*D* algorithm	*S* algorithm	Route construction	Route improvement	Center of gravity	Length
15	100	120	120	120	120	14.64
20	130	190	180	200	200	19.88
23	130	205	210	210	210	23.00
25	145	230	230	230	230	24.13
27	160	230	230	230	230	24.13
30	190	250	260	260	260	29.22
32	195	275	260	260	260	29.22
35	200	315	300	300	300	34.93
38	210	355	355	355	355	37.79
40	240	395	380	380	380	39.89
45	270	430	450	450	450	44.44

Table 3. Problem 3 results.

TMAX	D algorithm	S algorithm	Route construction	Route improvement	Center of gravity	Length
15	70	100	170	170	170	14.47
20	120	140	200	200	200	19.79
25	140	190	250	250	250	23.61
30	180	240	320	320	320	29.19
35	220	290	380	380	380	34.69
40	240	330	420	420	420	38.19
45	280	370	450	450	450	44.96
50	310	410	500	500	500	49.33
55	340	450	520	520	520	52.81
60	350	500	550	550	580	58.52
65	410	530	590	590	600	63.72
70	460	560	620	620	640	69.19
75	490	590	640	650	650	71.00
80	510	640	660	660	690	79.48
85	520	670	720	720	720	83.82
90	580	690	760	770	770	89.31
95	610	720	790	790	790	92.79
100	640	760	800	800	800	97.08
105	660	770	800	800	800	97.77
110	680	790	800	800	800	99.08

al. [2,3] for additional details. The other steps in the inventory/routing problem include (1) customer/vehicle assignment—assignment of customers selected for service on a particular day to one of the available trucks, and (2) routing—the construction of efficient trips (routes) for each truck over the set of its assigned customers.

The orienteering problem addressed in this paper assumes that TMAX cannot, under any circumstances, be violated. Let us consider a problem relaxation where TMAX can be exceeded at the expense of a per-minute penalty, say π. In other words, the score is reduced by the number of minutes in excess of TMAX multiplied by a per minute penalty for lateness. This relaxed problem actually comes closer to representing a real-world orienteering competition than the "strict" version.

One way of attacking the relaxed problem is to apply the center-of-gravity heuristic to the strict version for a number of different TMAX values. For example, suppose TMAX = 75 and $\pi = 8$ and consider Problem 3. From Table 3, one can easily generate Table 6. The adjusted score is the original score minus the lateness penalty. The best solution found requires 89.31 units of time. Clearly, one can use as many TMAX values as necessary in order to obtain a good solution to the relaxed problem. We point out that this relaxed problem can also be viewed as a more general problem in which an over-time decision needs to be made, i.e., the tradeoff between gain in productivity and cost of labor needs to be evaluated.

APPENDIX: THE ROUTE CONSTRUCTION HEURISTIC

The procedure used to generate an initial route is based on a weighted ranking. That is, insertions are made according to a convex combination of three ranks. The three ranks are (1) a score rank,

Table 4. Running time (in seconds).

	Problem 1	Problem 2	Problem 3
No. of nodes	32	21	33
TMAX			
5	0.57	···	···
10	0.83	···	···
15	1.18	0.97	1.87
20	1.54	1.18	2.38
23	···	1.26	···
25	1.84	1.20	2.43
27	···	1.40	···
30	2.15	1.64	2.70
32	···	1.64	···
35	2.44	1.67	2.97
38	···	1.99	···
40	2.95	1.86	3.47
45[a]	3.47	2.06	3.64
50	3.82	···	3.96
55	5.52	···	4.53
60	7.14	···	4.50
65	5.24	···	5.19
70	6.19	···	6.04
73	6.88	···	···
75	8.27	···	5.76
80	7.52	···	8.04
85	7.90	···	9.29
90	···	···	8.14
95	···	···	8.94
100	···	···	7.37
105	···	···	7.15
110	···	···	7.50
Average	4.18	1.53	5.29

[a]TMAX is 46 for Problem 1.

SR(I), (2) a distance to center-of-gravity rank, CR(I), and (3) a sum of distances to the two foci of an ellipse rank, ER(I). For any node k not yet on the route, its score, center-of-gravity distance, and ellipse distance are three important factors in determining how appropriate it is to insert k next. The weighted ranking is given by $WR(I) = \alpha * SR(I) + \beta * CR(I) + \gamma * ER(I)$, where $\alpha + \beta + \gamma = 1$.

The score rank assigns a one to the node with the largest score and increasing ranks, as the scores decrease. Any tie scores are assigned equivalent ranks. The ellipse rank uses the start and end nodes as foci of an ellipse. A rank of one is given to the node with the smallest sum of distances to the two foci. Increasing ranks are assigned to nodes in ascending order of distance. To begin, the center-of-gravity rank is based on a node's distance to the center of gravity of all the nodes in the network. The node closest to the center of gravity is given rank one, with ranks increasing as distances increase. To take advantage of the changing route structure, after each insertion, we recompute the

Table 5. Percent of total running time by step.

	Problem 1	Problem 2	Problem 3
Route construction	64.0	79.4	69.3
Route improvement	2.8	3.2	5.4
Center of gravity	33.2	17.4	25.3

Table 6. Orienteering problem relaxation.

TMAX	Score	Length	Adjusted score
75	650	71.00	650.00
80	690	79.48	654.16
85	720	83.82	649.44
90	770	89.31	655.52
95	790	92.79	647.68
100	800	97.08	623.36
105	800	97.77	617.84
110	800	99.08	607.36

center of gravity based on the nodes currently on the route. Thus a reranking of the center-of-gravity list is computed after each insertion.

Clearly, at each step we choose the node with the smallest WR(I) as the node to be inserted onto the route. We then apply a cheapest insertion procedure. We continue to insert nodes as long as TMAX is not violated. At termination, the route is passed to the route improvement step.

While α, β, and γ are held constant throughout a single pass of the procedure, by choosing different combinations of these three parameters we can produce a variety of routes. In practice, we employ 22 combinations and save the route with the highest score as the one to pass to the next step.

REFERENCES

[1] Garey, M., and Johnson, D., *Computers and Intractability,* Freeman, San Francisco, 1979.
[2] Golden, B., Assad, A., and Dahl, R., "Analysis of a Large-Scale Vehicle Routing Problem with an Inventory Component," *Large Scale Systems,* **7,** 181–190 (1984).
[3] Golden, B., Levy, L., and Dahl, R., "Two Generalizations of the Traveling Salesman Problem," *OMEGA,* **9**(4), 439–441 (1981).
[4] Tsiligirides, T., "Heuristic Methods Applied to Orienteering," *Journal of the Operational Research Society,* **35**(9), 797–809 (1984).
[5] Wren, A., and Holliday, A., "Computer Scheduling of Vehicles from One or More Depots to a Number of Delivery Points," *Operational Research Quarterly,* **23,** 333–344 (1972).

Manuscript received December 12, 1985
Revised manuscript received May 1, 1986
Accepted June 2, 1986