ELSEVIER

European Journal of Operational Research 88 (1996) 464–474

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

Theory and Methodology

The team orienteering problem

I-Ming Chao [a], Bruce L. Golden [b,*], Edward A. Wasil [c]

[a] *Department of Mathematics and Management Sciences, P.O. Box 90602, The Chinese Military Academy, Feng-Shen, Taiwan, ROC*
[b] *College of Business and Management, University of Maryland, College Park, MD 20742, USA*
[c] *Kogod College of Business Administration, American University, Washington, DC 20016, USA*

Received 1 September 1993; revised 1 September 1994

Abstract

In the team orienteering problem, start and end points are specified along with other locations which have associated scores. Given a fixed amount of time for each of the M members of the team, the goal is to determine M paths from the start point to the end point through a subset of locations in order to maximize the total score. In this paper, a fast and effective heuristic is presented and tested on 353 problems ranging in size from 21 to 102 points. The computational results are presented in detail.

Keywords: Vehicle routing problem; Heuristic search

1. Introduction

Orienteering is an outdoor sport usually played in a mountainous or heavily forested area. Armed with compass and map, a competitor starts at a specified control point, tries to visit as many other control points as possible within a prescribed time limit, and returns to a specified control point. Each control point has an associated score, so that the objective of orienteering is to maximize the total score. A competitor who arrives at the finish point after time has expired is disqualified, and the eligible competitor with the highest score is declared the winner. Since time is limited, a competitor may not be able to visit all control points. A competitor has to select a subset of control points to visit that will maximize total score subject to the time restriction. This is known as the Single-Competitor Orienteering Problem (OP).

Team orienteering extends the single-competitor version of the sport. A team consisting of several competitors (say, 2, 3, or 4 members) starts at the same point. Each member of the team tries to visit as many control points as possible within a prescribed time limit, and then ends at the finish point. Once a team member visits a point and is awarded the associated score, no other team member can be awarded a score for visiting the same point. Thus, each member of a team has to select a subset of control points to visit so that there is minimal overlap in the points visited by each member of the team, the time limit is not violated, and the total team score is maximized. We call this the Team Orienteering Problem and denote it by TOP.

* Corresponding author. E-mail: bgolden@umdacc.umd.edu

SSDI 0377-2217(94)00289-4

The TOP can be modeled as a multi-level optimization problem. At the first level, we need to select a subset of points for the team to visit. At the second level, we need to assign points to each member of the team. At the third level, we need to construct a path through the points assigned to each member of the team. We point out that the single-competitor version of this problem has been shown to be NP-hard (Golden, Levy, and Vohra, 1987), so the TOP is at least as difficult.

We now describe a network optimization representation of the TOP. Let V be the set of control points and E be the set of edges between points in V. Then $G = \{V, E\}$ is a complete graph. Each point i in V has a score $s_i \geq 0$ associated with it. The start point is vertex 1 and the finish point is vertex n, and these points each have a score of 0. Each edge in E has a symmetric, nonnegative cost c_{ij} associated with it, where c_{ij} is the distance between point i and point j, or the cost of traveling between the two points. For the M-member TOP, we need to find a set of M paths, where each path starts from point 1 and finishes at point n, that maximizes total team score. Each point's associated score is only awarded on the first visit by a team member and the total time taken to visit the points on each of the M paths cannot exceed the specified limit, denoted by $T_{\max}$. We point out that real-world orienteering competitions involve a variety of complications (e.g., stochastic travel times) not mentioned in this paper or in the cited literature. In the operations research literature, the orienteering problem denotes a class of routing problems related to the traveling salesman problem.

We note that the orienteering problem described by Chao, Golden, and Wasil (1996) can be considered a special case of the TOP, that is, the OP is a one-member TOP. In addition, many of the OP applications can be extended to team orienteering applications. For example, the home fuel delivery model used by Golden, Assad, and Dahl (1984) can be extended in the following way. Treating each customer's urgency for fuel as a score, each vehicle in the fleet would be assigned a subset of customers to service and a route would be constructed for each vehicle. The objective would be to maximize the total score amassed by the fleet. In a recent paper, Butt and Cavalier (1992) model the recruiting of college football players as a TOP. Suppose there are many high schools surrounding a college that a recruiter would like to visit in order to scout members of the football team. The recruiter needs to leave and return to the college campus within the same day and the recruiter can only meet with the high school students during their class time (this establishes the maximum time limit $T_{\max}$). There is a score associated with each high school that measures the potential 'benefit' to the college of visiting the high school. The number of high schools is so large that the recruiter cannot visit all schools within a limited time period (say, one day). If the recruiter has M days to visit the high schools, then the recruiter would like to find a set of M paths that maximizes the total potential for recruiting football players, where the total time taken by the recruiter to visit high schools on each of the M paths cannot exceed $T_{\max}$. The TOP can also be used to model a variety of vehicle routing problems in which only a subset of the customers can be visited on a given day.

In the next section, we mention an unpublished solution approach for the TOP developed by Butt and Cavalier. In the third section, we develop a new heuristic for the TOP. In the fourth section, we generate 353 test problems, present computational results produced by our new heuristic, and compare these results to those produced by a stochastic algorithm for the team orienteering problem. In the final section, we present our conclusions.

2. Review of solution approaches to the TOP

Although the original orienteering problem has attracted the attention of many researchers, the TOP has received no attention in the open literature. We are aware of only one unpublished paper – the recent work by Butt and Cavalier (1992) – in which the TOP is solved. In their version of the TOP, the start and finish points are the same. They apply their heuristic to small problems with less than 15 points and compare the results produced by their heuristic against the optimal results produced by a mathematical programming model that they formulated and solved. Based upon the performance of the heuristic on the small problems, Butt and Cavalier conclude that their method should work well on large problems, but they do not apply it to large-size TOPs. In addition, Butt and Cavalier do not publish test problem data in their paper.

In contrast to the TOP, the orienteering problem has been widely studied and a variety of heuristics have been developed and tested (see Chao (1993) and Chao, Golden, and Wasil (1996) for details). In order to evaluate the performance of our new TOP heuristic, we modify a standard OP heuristic so that it will solve the TOP. Since Tsiligirides's stochastic algorithm for the OP (Tsiligirides, 1984; Chao, Golden, and Wasil, 1996) can be easily modified to solve the TOP, we apply his algorithm and code two different versions. This is discussed in more detail in Section 4.

3. A new heuristic for solving the TOP

The TOP is more difficult to solve than the OP since we must take into account the performance of the entire team and not just one member as in the OP. In this section, we describe a new heuristic for the TOP that is easy to understand and easy to implement, and that produces high-quality solutions in a short amount of computation time. Our heuristic consists of two steps: initialization and improvement. We initialize the procedure by constructing an ellipse over the entire set of points by using the start and finish points as the two foci of the ellipse and the time limit T_{max} as the length of the major axis. We call this the T_{max} ellipse. In generating a path, we consider only the points that are within the ellipse, since any path that contains a point outside the ellipse will violate the T_{max} limit. We want to generate an initial solution quickly and then rely on the improvement step to find a solution with a large team score. In the improvement step, we allow the team score to decrease in the hope of ultimately finding a better solution.

3.1. *Initialization*

Initially, $L = \min(5, N)$ points, where N is the number of points within the ellipse, are chosen as candidate points to assign to each of the M paths (note that $M = 2$, 3, or 4 in our computational experiments). The L points are chosen to be the points furthest from the start and finish points. M of the L points are selected as the first points assigned to the M paths. Then, the remaining points are inserted in a greedy way (using cheapest insertion) onto the paths until each of the M paths is full (a path is full when inserting an additional point onto the path violates the time limit constraint). If unassigned points remain, we continue constructing new paths with these points until all points have been assigned. We then select the M paths with the highest scores as the initial solution and the sum of the scores of these paths is the team score.

For a problem, $\binom{L}{M}$ different solutions are possible when $L > M$. If $L \leq M$, an optimal solution can be obtained easily. (Note that for the values of M considered, $L \leq M$ implies $L = N$.) In this case, there are at most as many points in the T_{max} ellipse as there are team members, so that members can visit at most one point in an optimal solution. Among the $\binom{L}{M}$ solutions, the one with the highest team score is selected as our initial solution. We denote the set of M paths with the highest team score as $paths_{top}$ (these are the paths that form our initial solution) and the set of all remaining paths is denoted by $paths_{ntop}$.

The above initialization procedure assumes that points are located in two-dimensional Euclidean space. This is the case for all of the test problems considered in this paper. When this is not the case, starting solutions can be constructed in other ways and the improvement step may then be applied.

3.2. *Two-point exchange*

Using the starting solution produced in the initialization step, we try to improve this solution by performing a two-point exchange. A point i is moved from a path in $paths_{ntop}$ and inserted onto a path in $paths_{top}$, and a point j is moved from a path in $paths_{top}$ and inserted onto a path in $paths_{ntop}$. The two points are exchanged simultaneously and the insertions are performed in the cheapest way. If no feasible insertion is possible in $paths_{ntop}$ (that is, all insertions violate the T_{max} constraint), then a new path that contains point j is generated. This path contains point j along with the start and finish points.

Let $L(p)$ denote the length of path p. The feasibility of the path that results when point i is inserted onto path p and point j is removed from path p can be checked by examining the following expression

$$L(p) - (c_{j,fj} + c_{pj,j} - c_{pj,fj}) + \min_{\substack{k \text{ visited in } p,\\ k \neq 1,j}} \{c_{i,k} + c_{i,pk} - c_{k,pk}\}, \qquad (1)$$

where pj is the point that precedes point j on path p, fj is the point that follows point j on path p, and pk is the point that precedes point k on path p after point j has been removed from the path. In (1), the term after the first minus sign is the savings that results from removing point j and the term after the second plus sign is the cost incurred by inserting point i onto path p. If the distance that results from the calculation in (1) is less than or equal to T_{max}, then the insertion is feasible; otherwise, the insertion is infeasible.

For each point in $paths_{top}$, candidate exchanges are considered one at a time. Whenever a candidate exchange leads to a higher team score, the exchange is performed immediately, and all other exchanges are ignored. Whenever there is no candidate exchange for a point that increases the team score, we consider exchanges that decrease the team score by a small amount. If the decrease yields a score that is above a threshold value, then we perform it; otherwise, the point remains in its current position on the path, and we consider exchanging a different point. The score of the best solution obtained during this process is called the *record* and the amount of decrease below *record* that we allow during the process is called the *deviation*. This approach is referred to as record-to-record improvement and is due to Dueck (1990). It may be viewed as a deterministic variant of simulated annealing. (We might have applied simulated annealing or, alternatively, tabu search instead, but Dueck's approach seemed faster than simulated annealing and easier to implement than tabu search.) Our two-point exchange algorithm is given in Table 1. We point out that as two-point exchanges are performed it is possible for a path in $paths_{ntop}$ to replace a path in $paths_{top}$. We always keep the M paths with the highest team score in $paths_{top}$.

3.3. *One-point movement*

We now consider moving one point at a time between paths. In particular, we try to move a point i from its current location to another location in front of a point on another path. We make the move whenever it is feasible and increases the team score. If no movement increases the team score, then we consider the feasible movement that decreases the team score by the least amount. To obtain a candidate movement for a point and

Table 1
Outline of two-point exchange algorithm for the TOP

```
For j = the first to the last point in the first to the last path in paths_top          (A loop)
    For i = the first to the last point in the first to last path in paths_ntop         (B loop)
        If exchanging i and j is feasible and the team score increases, do the exchange and go to the A loop
        Else
        Set best exchange = feasible exchange with the highest team score
    End B loop
    If the team score of the best exchange ≥ record − deviation, make the best exchange
End A loop
```

then determine which move to make, we apply the steps in Table 2.

We point out that although only one point is moved at a time, this type of movement can still change $paths_{top}$. One point can be moved from $paths_{top}$ to a path in $paths_{ntop}$ and vice versa. We can also move a point between paths in $paths_{top}$ and between paths in $paths_{ntop}$. We consider all of these movements in an effort to find the set of paths with the largest team score.

3.4. Clean up

In order to shorten the length of each path in $paths_{top}$, we apply a 2-opt improvement procedure (Lin, 1965). The hope is that by decreasing the length of each path we have more opportunities to insert points from paths in $paths_{ntop}$ onto paths in $paths_{top}$.

3.5. Example

We illustrate two-point exchange, one-point movement, and clean up in Fig. 1. In Fig. 1(a), we provide the scores and locations of the points for a 2-member TOP, where the number above a point is the associated score. Fig. 1(b) shows an intermediate solution, where the thick, bold lines denote the paths in $paths_{top}$ (that is, the paths 1–6–9 and 1–7–5–9 with a team score of 19) and the thin lines denote the paths in $paths_{ntop}$ (that is, the paths 1–2–3–9 and 1–8–4–9). In moving from Fig. 1(b) to Fig. 1(c), we perform a one-point movement: point 3 moves from a path in $paths_{ntop}$ (that is, 1–2–3–9) to a path in $paths_{top}$ (that is, 1–6–9). In particular, point 3 is inserted in front of point 6. In Fig. 1(c), two paths (1–3–6–9 and 1–7–5–9) have a combined score of 21 and comprise $paths_{top}$. We now perform a 2-opt procedure on these two paths and obtain 1–6–3–9 and 1–7–5–9. Next, we move point 4 before point 9 on the first of these paths to obtain 1–6–3–4–9. This results in a team score of 23 in Fig. 1(d). We now perform a two-point exchange by inserting point 8 between points 1 and 7 and inserting point 5 between 1 and 9 at the same time, and this yields the solution shown in Fig. 1(e), which has the highest score of all solutions produced by our heuristic.

3.6. Reinitialization I

In the hope of finding a set of paths that yields a larger team score, we remove k points with the smallest scores on paths in $paths_{top}$ and insert them onto paths in $paths_{ntop}$. As the iteration count increases in our procedure, we increase the value of k and remove more points from paths in $paths_{top}$.

3.7. Reinitialization II

In this step, we remove k points from $paths_{top}$ in a slightly different way. The k points with the smallest ratio of score to insertion cost are removed from paths in $paths_{top}$ and inserted in the cheapest feasible way onto paths in $paths_{ntop}$. Note that p is reduced to 2.5 since we want to

Table 2
Outline of algorithm for one-point movement in the TOP

For i = the first to the last point in the T_{max} ellipse (say i is in path q)	(A loop)
For j = the first to the last point in the first to last path (p) in $paths_{top}$ and $paths_{ntop}$ ($p \neq q$)	(B loop)
If inserting i in the front of j on path p is feasible and the team score increases,	
then make the movement and go to the A loop	
Else	
Set best movement = feasible movement with the highest team score	
End B loop	
If the team score of the best movement ≥ *record* − *deviation*, then make the best movement	
End A loop	

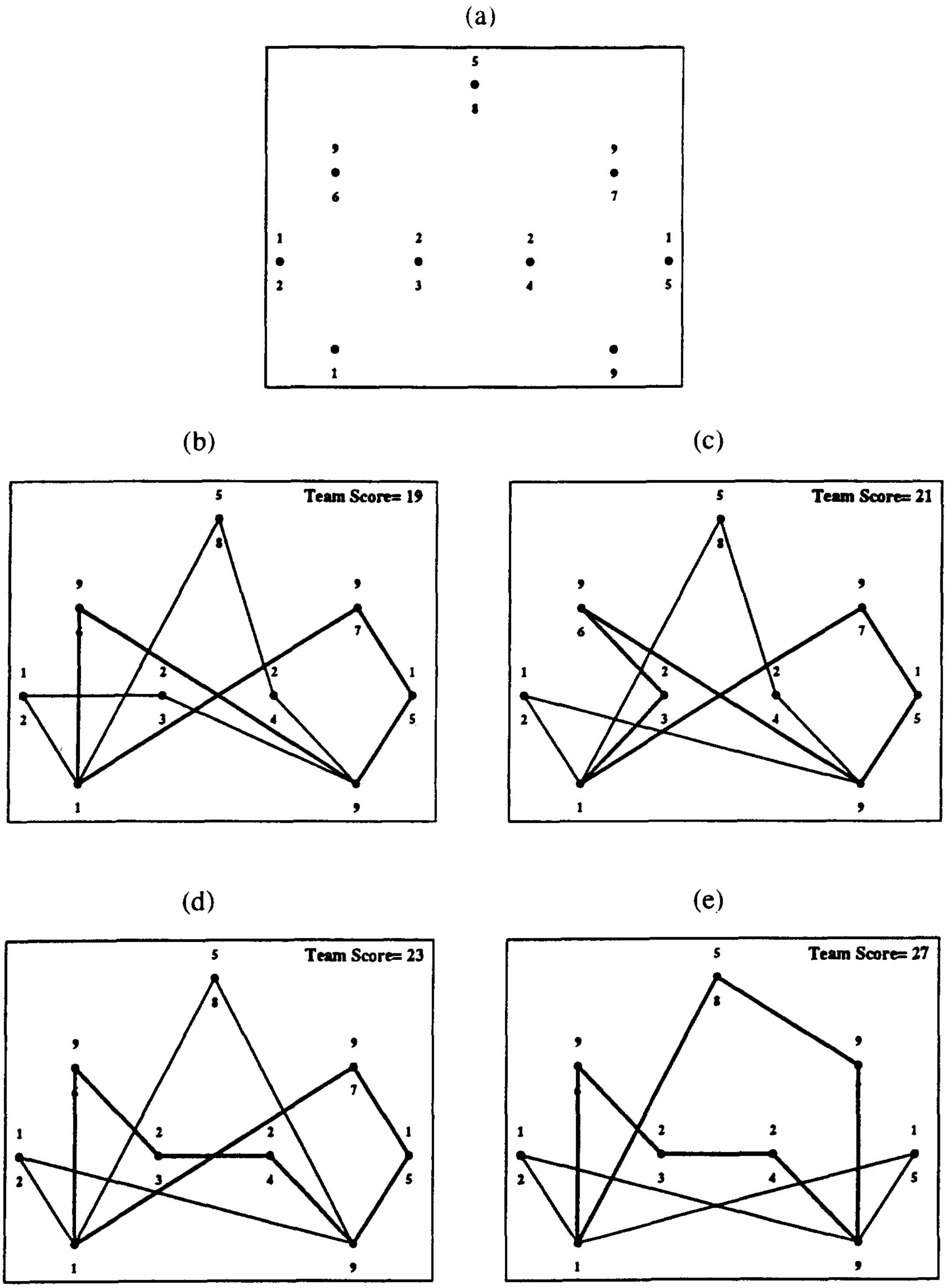

Fig. 1. Two-point exchange, one-point movement, and 2-opt improvement in the TOP heuristic.

perturb the solution only slightly at this point. Our complete, new heuristic for the TOP is shown in Table 3.

4. Computational testing

In this section, we apply our new heuristic for the TOP to a total of 353 test problems that we generate for 2-member, 3-member, and 4-member TOPs. We compare our results to those produced by a version of Tsiligirides's stochastic algorithm that we develop to solve the team orienteering problem. Both heuristics are coded in FORTRAN and executed on a SUN 4/370 workstation. We perform all computations using real precision (we do not round or truncate values) and round the length of the final path to one decimal place.

4.1. Generating test problems

As there are no test problems for the TOP that have been published in the literature, we need to generate a set of problems so that we can ascertain the effectiveness of our new heuristic. The easiest way to generate a test problem is to take a one-member OP and divide the T_{max} value by the number of team members. Thus, each team member has the same time limit, that is, T_{max}/M. In the three left-most columns of Table 4, we present information about the test problems that we generated for the 2-member, 3-member, and 4-member TOPs, respectively. All of these test problems are included in the dissertation by Chao (1993).

4.2. Generalizing Tsiligirides's stochastic algorithm to solve the TOP

We modify Tsiligirides's stochastic algorithm for the OP (Tsiligirides, 1984) so that it can solve the TOP. We develop a sequential version and a concurrent version of his algorithm. In our sequential version for the TOP, one path is constructed at a time, where each point i that is not yet included on the current path is assigned a desirability measure denoted by A_i. The desir-

Table 3
A New heuristic for the TOP

```
Step 1. Initialization
    Perform initialization
    Set record = team score of the initial solution
    Set p = 5
    Set deviation = p% × record
Step 2. Improvement
    For k = 1,2,...,K                                        (K loop)
        For i = 1,2,...,I                                    (I loop)
            Perform two-point exchange
            Perform one-point movement
            Perform clean up
            If no movement has been made above, end I loop
            If a new better solution has been obtained, then
                set record = score of new best solution
                set deviation = p% × record
        End I loop
        If no new record is obtained in 5 iterations, then
            go to Step 3
        Perform Reinitialization I (free k points)
    End K loop
Step 3. Reset p = 2.5, perform Reinitialization II (free k points, k is the stopping value in the K loop) and redo Step 2 once more
```

Table 4
Comparison of results

Test problem information			Computational results				Comparison	
Problem set	Number of points	Number of problems	Team Orienteering Heuristic TOH		Tsiligirides's Stochastic Algorithm TSA		TOH vs. TSA [a]	
			Average score	Maximum CPU (s)	Average score	Maximum CPU (s)	+	–
1.2	32	17	148.5	15.4	145.6	168.7	7	0
2.2	21	11	190.0	0.8	186.4	121.4	2	0
3.2	33	20	488.5	15.4	484.0	180.3	11	4
4.2	100	20	875.7	934.8	844.2	1748.2	17	3
5.2	66	25	890.6	193.7	885.6	566.9	12	4
6.2	64	11	814.9	150.1	746.2	420.2	11	0
7.2	102	20	633.9	844.4	586.6	1333.3	16	1
1.3	32	16	125.6	7.1	122.2	143.6	8	0
2.3	21	11	135.9	0.4	133.6	110.4	2	0
3.3	33	20	403.0	10.6	396.0	163.3	11	0
4.3	100	19	815.1	707.3	772.5	1563.0	17	1
5.3	66	25	776.2	157.8	761.6	516.1	15	0
6.3	64	8	787.5	135.7	707.3	336.7	8	0
7.3	102	19	585.5	557.6	554.1	1191.5	14	2
1.4	32	15	99.3	6.8	97.7	116.4	4	2
2.4	21	11	94.6	0.1	94.6	86.9	0	0
3.4	33	20	332.5	8.2	327.5	146.1	8	2
4.4	100	17	766.1	674.9	713.0	1358.0	16	0
5.4	66	24	696.0	141.2	674.8	454.4	16	1
6.4	64	5	716.4	109.6	624.0	295.0	5	0
7.4	102	19	497.4	482.9	480.6	1059.9	10	4
						Total	210	24

[a] +: TOH produced a higher score;
–: TOH produced a lower score.

ability measure is given by

$$A_i = (s_i/c_{i,\text{last}})^4, \qquad (2)$$

where s_i is the score associated with point i and $c_{i,\text{last}}$ is the distance from the last point on the current path to point i. The four points with the largest desirability measures are then selected and their measures are normalized according to

$$P_i = \frac{A_i}{\sum_{t=1}^{4} A_t}, \quad \text{for } i = 1,2,3,4. \qquad (3)$$

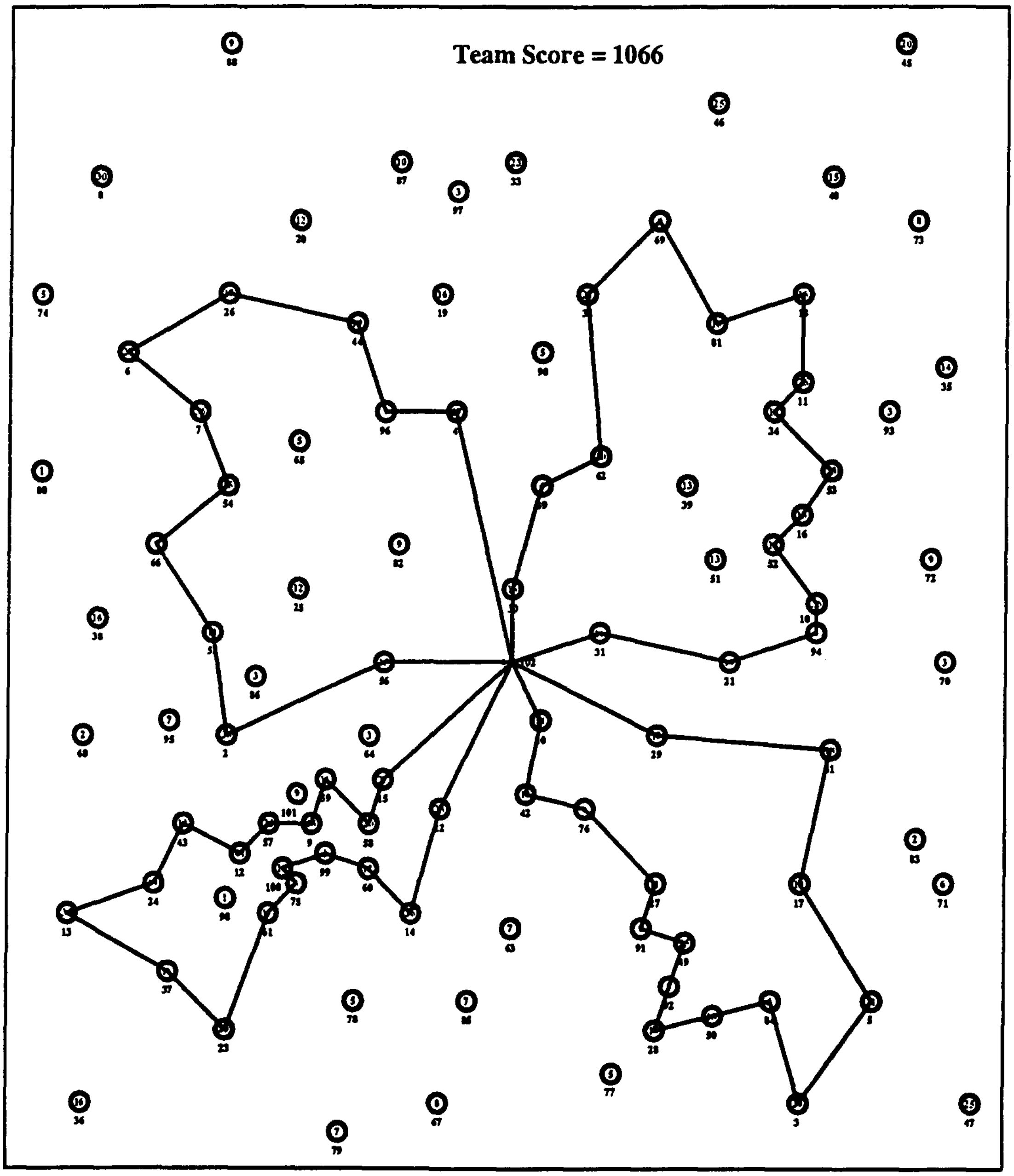

Fig. 2. Solution produced by TOH to a 4-member test problem.

Point i is randomly selected with probability P_i as the new last point on the current path. Points continue to be placed on the current path until the remaining time is so small that no points can be feasibly inserted onto the path. We continue constructing new paths using the remaining unassigned points in this way until M paths are obtained.

In the concurrent version of Tsiligirides's algorithm that we develop, the M paths are constructed simultaneously. In the k-th step, our procedure seeks the k-th point for each of the M paths, that is, the k-th point of the first path is found using the above stochastic version, then the k-th point of the second path is determined, and so on. When a path is full (that is, we cannot insert any more points onto the path), our procedure skips that path and searches over all paths that are not yet full. Our procedure stops when all M paths are full.

In Tsiligirides's heuristic for the OP (Tsiligirides, 1984), 3,000 solutions are generated for each problem. For the TOP, we generate 1,500 solutions using our sequential approach, and 1,500 solutions using our concurrent approach. The solution among all 3,000 solutions with the highest team score is chosen as the final solution. The final solution is then improved by applying a 2-opt procedure to each path and inserting as many points as possible onto the resulting paths.

4.3. Results on test problems

We now compare the results produced by our heuristic to the best results produced by the sequential or concurrent versions of Tsiligirides's stochastic algorithm to the 353 team orienteering problems that we developed. After experimenting with different values for K and I in our heuristic, we found that $K = 50$ and $I = 10$ produce good solutions in a reasonable amount of CPU time; these are the values we used to generate the results in Table 4. In Table 4, the first seven rows present results for 2-member problems, the next seven rows present results for 3-member problems, and the last seven rows present results for 4-member problems. In Fig. 2, we show the solution produced by our new heuristic to a 4-member test problem. Solutions to all 353 test problems are displayed in Chao (1993).

For the 2-member TOP, there are 124 problems and TOH produces scores for 76 problems that are better than the scores produced by TSA, and 12 scores that are worse than scores produced by TSA. For the 3-member TOP, there are 118 problems and TOH produces scores for 75 problems that are better than the scores produced by TSA, and 3 scores that are worse than the scores produced by TSA. For the 4-member TOP, there are 111 problems and TOH produces scores for 59 problems that are better than the scores produced by TSA, and 9 scores that are worse than the scores produced by TSA. Over all 353 feasible test problems, our new team orienteering heuristic produces better scores than our version of Tsiligirides's stochastic algorithm on 60% (that is, 210/353) of the test problems. Our new heuristic produces worse scores on 7% (that is, 24/353) of the test problems, and the same score on 33% (119/353) of the test problems.

5. Conclusions

In this paper, we have presented a heuristic for the team orienteering problem. The heuristic is based on the notion of record-to-record improvement. We have applied our heuristic and a competing heuristic to 353 problems ranging in size from 21 to 102 points. The new heuristic has been shown to be computationally efficient and it consistently outperforms its competition.

References

Butt, S.E., and Cavalier, T.M. (1992), "A heuristic for the multiple tour maximum collection problem", Working Paper, Department of Industrial and Management Systems Engineering, The Pennsylvania State University, University Park, PA.

Chao, I-M. (1993), "Algorithms and solutions to multi-level vehicle routing problems", Ph.D. Dissertation, Applied Mathematics Program, University of Maryland, College Park, MD.

Chao, I-M., Golden, B., and Wasil, E. (1996), "A fast and effective heuristic for the orienteering problem", *European Journal of Operational Research* 88, 475–489.

Dueck, G. (1990), "New optimization heuristics, the great deluge algorithm and the record-to-record travel", Scientific Center Technical Report, IBM Germany, Heidelberg Scientific Center, December.

Golden, B., Assad, A., and Dahl, R. (1984), "Analysis of a large-scale vehicle routing problem with an inventory component", *Large Scale Systems* 7, 181–190.

Golden, B., Levy, L., and Vohra, R. (1987), "The orienteering problem", *Naval Research Logistics* 34/3, 307–318.

Lin, S. (1965), "Computer solutions of the traveling salesman problem", *Bell System Technical Journal* 44, 2245–2269.

Tsiligirides, T. (1984), "Heuristic methods applied to orienteering", *Journal of the Operational Research Society* 35/9, 797–809.