Applied Soft Computing 13 (2013) 1064–1073

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Solving the team orienteering problem using effective multi-start simulated annealing

Shih-Wei Lin

Department of Information Management, Chang Gung University, 259 Wen-Hwa 1st Road, Kwei-Shan, Taoyuan 333, Taiwan

ARTICLE INFO

Article history:
Received 4 February 2012
Received in revised form 18 August 2012
Accepted 13 September 2012
Available online 13 October 2012

Keywords:
Team orienteering problem
Meta-heuristics
Multi-start simulated annealing

ABSTRACT

The team orienteering problem (TOP) is known as an NP-complete problem. A set of locations is provided and a score is collected from the visit to each location. The objective is to maximize the total score given a fixed time limit for each available tour. Given the computational complexity of this problem, a multi-start simulated annealing (MSA) algorithm which combines a simulated annealing (SA) based meta-heuristic with a multi-start hill climbing strategy is proposed to solve TOP. To verify the developed MSA algorithm, computational experiments are performed on well-known benchmark problems involving numbers of locations ranging between 42 and 102. The experimental results demonstrate that the multi-start hill climbing strategy can significantly improve the performance of the traditional single-start SA. Meanwhile, the proposed MSA algorithm is highly effective compared to the state-of-the-art meta-heuristics on the same benchmark instances. The proposed MSA algorithm obtained 135 best solutions to the 157 benchmark problems, including five new best solutions. In terms of both solution quality and computational expense, this study successfully constructs a high-performance method for solving this challenging problem.

1. Introduction

The team orienteering problem (TOP) presents a set of n locations to be visited by m tours, each with a score S_i ($i \in 1, \ldots, n$). The starting point (location 1) and the end point (location n) are fixed. The time t_{ij} required to travel from location i to j is known in advance for all locations. Each location can be visited a maximum of once. Not all locations can be visited since the available time is limited to a given time budget T_{max} for all tours. TOP aims to determine m tours, and each tour visits some of the locations, to maximize total collected score.

The concept of TOP was first described by Butt and Cavalier [1], who called it the Multiple Tour Maximum Collection Problem, with Chao et al. [2] later introducing the term TOP. Many different problems may be treated as TOPs, including recruiting of football players from high schools for college teams [1], the multi-vehicle version of the home fuel delivery problem [2,3], and routing technicians for customer service [4].

Since the TOP is an NP-hard problem [2], only two exact algorithms have been proposed specifically for TOP. Butt and Ryan [5] proposed a column generation based algorithm, while Boussier et al. [6] proposed a branch and price based algorithm. However, these methods can only solve problems of limited size using a reasonable amount of computing time. Therefore, numerous research efforts have focused on heuristics and meta-heuristics. Butt and Cavalier [1] proposed a greedy method. The best pair of vertices is assigned to the solution tours in each iteration. The procedure stops following construction of all m tours and when no additional vertices can be added without exceeding T_{max}. Chao et al. [7] proposed a five-step method (initialization, main movement, clean up, local improvement and re-initialization) and a heuristic based on the stochastic algorithm of Tsiligirides [8]. Tang and Miller-Hooks [4] proposed a tabu search based approach embedded with an adaptive memory procedure [9] for TOP. Furthermore, Archetti et al. [10] proposed two tabu search algorithms and a variable neighborhood search algorithm. Their study proposed four approaches: tabu search with penalty strategy, tabu search with feasible strategy, fast variable neighborhood search, and slow variable neighborhood search. Ke et al. [11] proposed an ant colony optimization (ACO) framework for the TOP. Four methods, sequential, deterministic-concurrent, random-concurrent and simultaneous, were proposed to construct candidate solutions in the ACO framework. Vansteenwegen et al. [12,13] designed a guided local search (GLS) approach and a skewed variable neighborhood (SVN) approach for the TOP. Both approaches applied a combination of intensification and diversification procedure. Recently, Souffriau et al. [14] proposed two variants of a greedy randomized adaptive search procedure (GRASP) with path relinking (PR) for solving the TOP. The first variant is a slow version (SPR) which obtained excellent results, while the second variant

E-mail addresses: swlin@mail2000.com.tw, swlin@mail.cgu.edu.tw

1568-4946/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2012.09.022

is a fast version (FPR), which rapidly obtained an acceptable solution. Vansteenwegen et al. [15] presented a thorough survey of approaches for TOP.

Simulated annealing based heuristics have been successfully applied to various hard combinatorial optimization problems [16–23]. The search procedures of simulated-annealing (SA)-based algorithm that attempt to obtain (near) global optimal solutions typically require some form of diversification to escape from local optimality. Without such diversification, SA-based algorithms may become localized in a small area of the solution space, eliminating the possibility of finding a global optimum. Multi-start simulated-annealing algorithm (MSA) combines the advantages of SA algorithms in effectively achieving search convergence and the multi-start hill climbing strategy in escaping local optimality. MSA has been successfully applied to problems in a wide variety of fields [24–27]. This study proposes the MSA algorithm for solving TOP.

The remainder of this paper is structured as follows. After formulating the team orienteering problem in Section 2, the proposed MSA algorithm is described in Section 3. Using an existing benchmark problem set, Section 4 empirically assesses the effectiveness and efficiency of the proposed MSA algorithm by comparing its performance with the traditional single-start SA and state-of-the-art algorithms. Finally, conclusions are drawn and recommendations for future research are made in Section 5.

2. Mathematical formulation

The TOP can be defined on a directed network $G = (N, A)$, where $N = \{1, 2, \ldots, n\}$ is the set of locations, and $A = \{(i, j) : i \neq j \in N\}$ is the set of arcs connecting locations. Each location $i = 1,\ldots,n$ is associated with a score S_i. All paths must start at location 1 and end at location n. The time t_{ij} required to travel from location i to j is known for each pair of locations i and j. The travel time of a tour cannot exceed the given time budget T_{max}. Each location can be visited a maximum of once. The objective is to identify a set of m tours that maximizes the total score.

Define the following decision variables:

$x_{ijm} = 1$ if, in tour m, a visit to location i is followed by a visit to location j, 0 otherwise;

$y_{ik} = 1$ if location i is visited on tour k, 0 otherwise;

s_{ik} = the start time of the service at location i on tour k;

The TOP can then be formulated as the following mixed integer problem [4,11,13]:

$$\text{Maximize} \sum_{k=1}^{m}\sum_{i=2}^{n-1} S_i y_{ik}$$

Subject to

$$\sum_{k=1}^{m}\sum_{j=2}^{n-1} x_{1jk} = \sum_{k=1}^{m}\sum_{i=2}^{n-1} x_{ink} = m \tag{1}$$

$$\sum_{i=1}^{n-1} x_{ilk} = \sum_{j=2}^{n} x_{ljk} = y_{lp}, \forall l = 2, \ldots, n-1; \quad \forall k = 1, \ldots, m \tag{2}$$

$$\sum_{k=1}^{m} y_{ik} \leq 1, \quad \forall i = 2, \ldots, n-1 \tag{3}$$

$$\sum_{i=1}^{n-1}\sum_{j=2}^{n} t_{ij} x_{ijk} \leq T_{\max}, \quad \forall k = 1, \ldots, m \tag{4}$$

$$\sum_{\substack{i,j \in U \\ i<j}} x_{ijk} \leq |U| - 1 \quad (U \subset V \setminus \left\{1, n\right\}; \quad 2 \leq |U| \leq n-2; \quad k = 1, \ldots, m) \tag{5}$$

$$x_{ijk}, y_{ik} \in \{0, 1\}, \quad \forall i, j = 1, \ldots, n; \quad \forall k = 1, \ldots, m \tag{6}$$

The objective function maximizes the total collected score of all tours. Constraint (1) states that all tours must start from location 1 and terminate at location n. Constraint (2) maintains the connectivity of each tour. Constraint (3) ensures that every location is visited at most once. Constraint (4) guarantees that the time limitation of each tour is not violated. Constraint (5) excludes the sub-tours. Finally, constraint (6) specifies that the variables are binary.

3. Proposed multi-start simulated annealing algorithm

Metropolis et al. [28] first introduced the concept of simulated annealing, which was later popularized by Kirkpatrick et al. [29] and Cerny [30] in solving challenging combinatorial optimization problems. The SA typically begins with a randomly generated initial solution. At each iteration, the algorithm selects a new solution from the neighborhood of the current one. If the objective function value of the new solution is better than that of the current one, the new solution replaces the current solution from which the search process continues. Moreover, there is a small probability that a new solution with a worse objective function value may also be accepted as the new current solution.

The proposed multi-start simulated annealing (MSA) algorithm combines the advantages of the SA algorithm and the multi-start hill climbing strategy for solving the team orienteering problem. The following subsections further discuss the solution representation, initial solution, neighborhood, parameters, and MSA procedures.

3.1. Solution representation and initial solution

A TOP solution is represented by a string of numbers comprising a permutation of n locations, denoted by the set $\{1, 2, \ldots, n\}$, and $m-1$ zeros used to separate tours. The jth non-zero number indicates the jth location to be visited. Thus, the first non-zero element in a solution indicates the first location to be visited during the first tour. Other locations are added to the tour individually from left to right to represent the visiting sequence, provided the time limit for each tour is not exceeded. If adding a location to the tour violates the time limit (supposing that this location is visited and then the tour returns directly to deposit), the location is discarded, and the next location is considered. A zero in the solution representation indicates that the current tour will be terminated and a new tour constructed whenever feasible. This solution representation always yields a feasible TOP solution without violating the tour time limit.

Half of the initial solutions (at least one) are generated by the nearest neighbor algorithm, and the remainder is generated randomly.

3.2. Illustration of solution representation

Table 1 gives a small TOP instance with a start deposit, an end deposit, and 19 customers. The location (X, Y) and scores (S) of deposits and customers are listed in Table 1. Two routes are used and the time limit for each route is 15.5. A randomly generated sample solution for this instance is shown in Fig. 1. One dummy zero is present in the solution. Fig. 2 gives a visual illustration of the sample solution.

In this example, the first tour starts by servicing customer 9 and then servicing locations 10, 11, 6, 5, 4, 2, 1 and 7, followed by a zero. Thus, the route will terminate after finishing servicing location 7. Because servicing customers 18, 19, and 17 after customer 10 will

Table 1
Locations and scores of customers and deposits.

Location	X	Y	S
Start deposit	8.6	7.1	0
Customer 1	5.7	11.4	20
Customer 2	4.4	12.3	20
Customer 3	2.8	14.3	30
Customer 4	3.2	10.3	15
Customer 5	3.5	9.8	15
Customer 6	4.4	8.4	10
Customer 7	7.8	11.0	20
Customer 8	8.8	9.8	20
Customer 9	7.7	8.2	20
Customer 10	6.3	7.9	15
Customer 11	5.4	8.2	10
Customer 12	5.8	6.8	10
Customer 13	6.7	5.8	25
Customer 14	13.8	13.1	40
Customer 15	14.1	14.2	40
Customer 16	11.2	13.6	30
Customer 17	9.7	16.4	30
Customer 18	9.5	18.8	50
Customer 19	4.7	16.8	30
End deposit	8.0	12.6	0

violate the travel limit, these customers are excluded from the first tour.

The first customer to be serviced in the second tour is customer 14, followed by customers 15 and 16 (customer 8 cannot be served because of the time limit). After finishing servicing customer 16, this tour will return to the end depot since no other customers can be visited without violating the time limit. Location 8, 3, 13, and 12 are excluded from the second tour. The total score collected is 255.

3.3. Neighborhood

Let S denote the set of feasible solutions and let σ_k ($k = 1, \ldots, P_{size}$) represent the current solution, where $\sigma_k \in S$, and P_{size} denote the number of starting points in the MSA algorithm. The sets $N(\sigma_k)$ ($k = 1, \ldots, P_{size}$) are then the sets of solutions neighboring σ_k. $N(\sigma_k)$ can be generated through either swap or insertion operations. The swap move randomly selects the ith and the jth locations of σ_k and then exchanges their positions. The insertion move is carried out by randomly selecting the ith location of σ_k and then inserting it into the position immediately before another randomly selected jth location of σ_k.

The probabilities of performing the swap and insertion operations were fixed at 0.5 and 0.5, respectively.

9	10	18	19	17	11	6	5	4	2	1	7	0	14	8	15	16	3	13	12

Fig. 1. An example of solution representation.

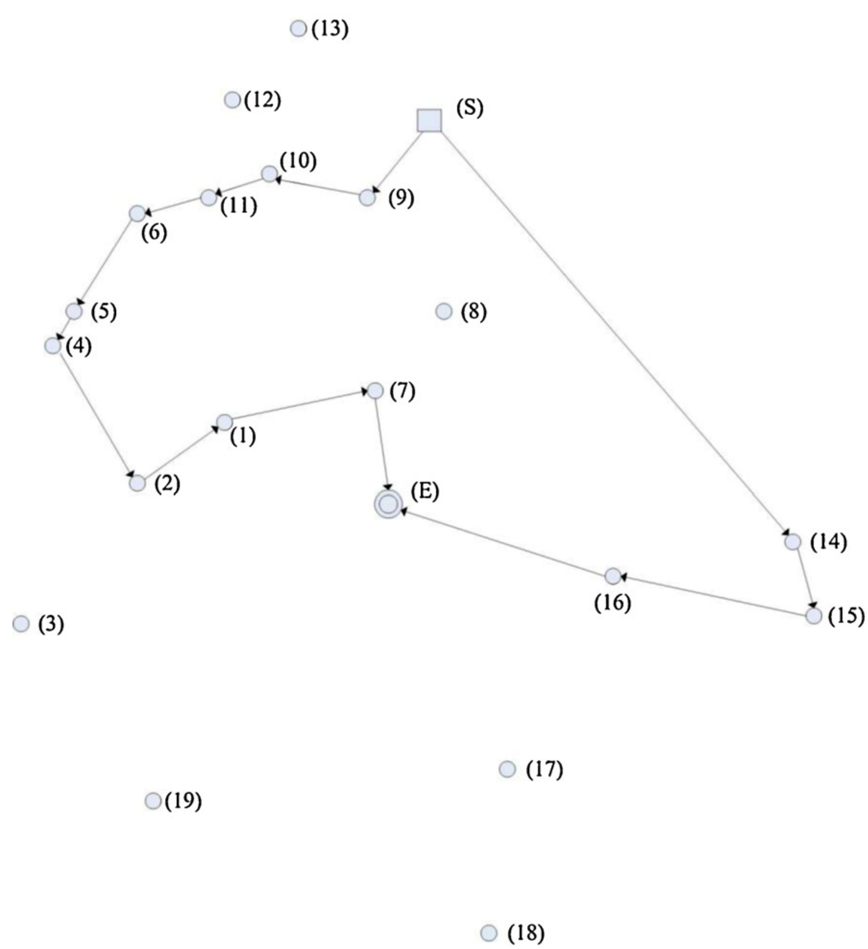

Fig. 2. A visual illustration of the example solution given in Fig. 1.

MSA (I_{iter}, T_0, $A_{threshold}$, $N_{non\ improving}$, P_{size} and α)
Step 1: Generating the initial solutions σ_k ($k = 1, 2, \ldots, P_{size}$)
Step 2: Let T T_0; R=0; N=0; σ_{best} = the best σ_k among the P_{size} solutions; $F_{best} = obj(\sigma_{best})$;
Step 3: N=N+1;
Step 4: For k 1 to P_{size} {
Step 4.1 Generating a solution σ'_k based on σ_k;
Step 4.2 If $\Delta_k = obj(\sigma'_k) - obj(\sigma_k) \geq 0$ {Let $\sigma_k = \sigma'_k$;}
Else {
Generate r~U(0,1);
If $r < Exp(\Delta_k / T)$ { Let $\sigma_k = \sigma'_k$;}
Else {Discard σ'_k;}
}
Step 4.3 If $obj(\sigma_k) > F_{best}$ {
$\sigma_{best} = \sigma_k$; $F_{best} = obj(\sigma_k)$; R=0;
Let $\sigma_j = \sigma_{best}$ ($j = 1, \ldots, P_{size}$)
}
Step 4.4 Calculate the accumulated probability of accepting worse solution $P_{acceptance}$;
}
Step 5: If $N = I_{iter}$ {
$T = T \times \alpha$; $N = 0$;
Perform local search for each σ_k;
For $k = 1$ to P_{size} {
If $obj(\sigma_k) > F_{best}$ {
$\sigma_{best} = \sigma_k$; $F_{best} = obj(\sigma_k)$; R=0;
Let $\sigma_j = \sigma_{best}$ ($j = 1, \ldots, P_{size}$)
}
}
σ_{temp} = the best σ_k among the P_{size} solutions;
If ($obj(\sigma_{temp}) < F_{best}$ and $P_{acceptance} < A_{threshold}$) { R= R +1;}
}
Else {Go to Step 3;}
Step 6: If $R = N_{non-improving}$ {Terminate the MSA procedure;}
Else {Go to Step 3;}

Fig. 3. The pseudo-code of proposed MSA algorithm.

3.4. Parameters

The proposed MSA algorithm begins with six parameters, namely I_{iter}, T_0, $N_{non\text{-}improving}$, $A_{threshold}$, P_{size}, and α, where I_{iter} denotes the number of iterations the search at a particular temperature, T_0 represents the initial temperature, and $N_{non\text{-}improving}$ is the permitted number of successive temperature reductions where the accumulated probability of accepting a worse solution is lower than threshold value $A_{threshold}$. P_{size} denotes the number of current solutions in each iteration. Finally, α denotes the coefficient controlling the cooling schedule.

3.5. MSA procedure

The procedures of the proposed MSA are depicted in Fig. 3. First, the current temperature T is set to T_0. Next, initial solutions σ_k ($k = 1, \ldots, P_{size}$) are generated by the nearest neighbor algorithm or randomly as multi-start points. For each iteration, the next solutions $\sigma\prime_k$ ($k = 1, \ldots, P_{size}$) are selected from their corresponding $N(\sigma_k)$. Furthermore, let $obj(\sigma_k)$ denote the calculation of the total score of σ_k, and let Δ_k represent the difference between $obj(\sigma_k)$ and $\mathrm{obj}(\sigma\prime_k)$; then $\Delta_k = \mathrm{obj}(\sigma_k) - \mathrm{obj}(\sigma\prime_k)$. The probability of replacing σ_k with $\sigma\prime_k$, given $\Delta_k < 0$, is $\mathrm{Exp}(\Delta_k/T)$. This can be achieved by generating a random number $r \in [0, 1]$ and replacing solution σ_k with $\sigma\prime_k$ if $r < \mathrm{Exp}(\Delta_k/T)$. Meanwhile, if $\Delta_k \geq 0$, the probability of replacing σ_k with $\sigma\prime_k$ is 1.

T is decreased after running I_{iter} iterations from the previous decrease, using the formula $T \leftarrow \alpha T$, where $0 < \alpha < 1$. When T is decreased, a local search procedure which sequentially performs swap and insertion operations is employed to enhance the best among the current solutions σ_k ($k = 1, \ldots, P_{size}$). If the incumbent best solution, σ_{best}, is not improved in $N_{non\text{-}improving}$ successive temperature reductions where the accumulated probability of accepting a worse solution is lower than threshold value $A_{threshold}$, the algorithm is terminated. If a new σ_{best} solution is obtained, all the current solutions σ_k ($k = 1, \ldots, P_{size}$) are set to be the same as the new σ_{best} solution. Following the termination of the MSA procedure, the (near) global optimal schedule can be derived using the σ_{best}.

4. Experimental results

This section describes the computational tests used to assess the performance of the proposed MSA algorithm for solving the team orienteering problem. The test problems, parameter selection, and computational results of the proposed MSA algorithm compared with the traditional single-start SA and state-of-the-art algorithms are discussed below.

4.1. Test problem

The proposed MSA approach is also applied to problem sets 4–7 [2] which contain 100, 66, 64 and 102(n) vertices, respectively. These sets were also adopted to evaluate the state-of-the-art metaheuristics in the literature. The number of tours m equals 2, 3, and 4 for each problem set. Instances of the same problem set with same number of tours only differ in the time limit T_{max}. The total number of instances from problem sets 4–7 is 157. The name of instance is notated as pa.b.c, where a represents the problem set which the instance belongs to, b is the number of tours, and c is a label that distinguishes between instances from the same problem set with same number of tours.

4.2. Algorithm parameters selection

Since the parameters of the proposed MSA have an important impact on the algorithm's performance, extensive computational testing was performed to determine the appropriate values for the experimental parameters. Two instances were randomly selected from each of the four problem sets for preliminary testing. The following combinations of the parameter values were tested on these eight test instances. $I_{iter} = (n + m - 1) \times \eta / P_{size}$, $N_{non\text{-}improving} = 3$, 5, 7, 9; $A_{threshold}$ = 0.005%, 0.01%, 0.02%, 0.03%, 0.05%; $T_0 = \beta \times TS$, and $\alpha = 0.93, 0.95, 0.97, 0.99$, where n denotes the number of locations; m represents the number of tours; η = 5000, 10,000, 20,000, 30,000, 40,000; P_{size} ranges from 1 to 8; β = 0.001, 0.002, 0.003; TS is the total scores of all locations.

Since the number of iterations (I_{iter}) is inversely proportional to the value of P_{size}, the total number of solutions evaluated for the same problem are almost identical for each different P_{size} value, ensuring the comparison is on a fair basis. The results indicated that the best solution quality was obtained using approximately η = 30,000, $N_{non\text{-}improving}$ = 5, $A_{threshold}$ = 0.01%, β = 0.001, $\alpha = 0.95$, $P_{size} = 3$ Therefore, the above parameter values were used for the experiments in this study.

It was observed that when I_{iter}, $N_{non\text{-}improving}$ and $A_{threshold}$ are increased, better solutions can be obtained at the expense of higher computational time. Increasing the value of P_{size} from one to three improves the solution quality. However, when the value of P_{size} exceeds three, larger value of P_{size} will result in worse solution quality. That is, when P_{size} equals three, the best solution quality is obtained. It is difficult to assess the effects of T_0 and α, separately. In general, if the initial temperature T_0 is too high, more computational time is needed to obtain good solutions. On the other hand, if T_0 is too low, the proposed approaches tend to converge prematurely because of the low probability of accepting worse solutions. However, combinations of parameters do have a significant impact on the solution quality. For example, setting $N_{non\text{-}improving}$ = 5, $I_{iter} = (n + m - 1) \times 30{,}000$, and P_{size} = 3 for the MSA, the difference in the objective values obtained by different combinations of T_0 and α can be as large as 14.534% of the best known solution value for the p5.2.h instance. The average percentage deviation of the solutions can be as large as 5.497%, compared to the solutions obtained using the best combination of parameters.

4.3. Results and discussion

The proposed MSA algorithm was implemented using C language and run on a PC with an Intel Core 2 2.5 GHz CPU.

To evaluate the proposed MSA algorithm, each of the problem instances was tested over ten trials and compared with other approaches, such as GTP, GTF, FVF, SVF [10], ASe, ADC, ARC, ASi [11], TMH [4], SVNS [12], GLS [13], PR [14], and the SA algorithm (P_{size} = 1).

GTP, GTF, FVF, and SVF denote the tabu search with penalty strategy, tabu search with feasible strategy, fast variable neighborhood search, and slow variable neighborhood search, respectively. GLS is a guided local search approach; ASe, ADC, ARC and ASi denote sequential ant colony optimization, deterministic concurrent ant colony optimization, random concurrent ant colony optimization, and simultaneous ant colony optimization, respectively. Furthermore, TMH is a tabu search based approach; SVNS denotes skewed variable neighborhood search. Finally, PR is a path relinking approach.

The best solution was selected as the experimental output and listed in Tables 2–5 for different combinations of number of locations and tours, providing a comparison with the traditional single-start SA and state-of-the-art algorithms from the literature. The results of other state-of-the-art approaches are taken from their original studies.

As shown in Tables 2–5, the proposed MSA algorithm obtains 135 solutions of benchmark instances that equal their BKS of benchmark problems, and obtains five solutions (p4.2.n, p4.2.6, p4.4.p, p4.4.r, and p7.3.t) that are new best solutions. These results clearly demonstrate that the proposed MSA algorithm can yield better solutions than all state-of-the-art algorithms. Notably, if $P_{size} = 1$, the proposed MSA algorithm is reduced to the traditional single-start SA algorithm which has been successfully applied to TOPTW [23]. In this case, 131 solutions of benchmark instances exist that have been obtained by the single-start SA algorithm and are equal to their BKS, while four solutions that are new best solutions. The results indicate that the proposed MSA algorithm is more effective in solving the team orienteering problem than the traditional single-start SA.

Table 6 summarizes the number of times the best known solution is found, the ARPD to the best known solution, and the average computational time in seconds required to obtain the best known solution in one run for each data set. The average relative percentage deviation (ARPD) over the 157 benchmark instances is then calculated according to $\mathrm{ARPD} = \sum_{l=1}^{157}((\mathrm{Obj}_l - \mathrm{BKS}_l/\mathrm{BKS}_l)/157) \times 100\%$, where Obj_l denotes the objective function value of problem instance l obtained using the algorithm being evaluated, and BKS_l is the best known solution of problem instance l obtained by existing algorithms, the SA algorithm (P_{size} = 1) and MSA (P_{size} = 3).

For all 157 problems, the 14 approaches (TMH, GTP, GTF, FVF, SVF, GLS, ASe, ADC, ARC, ASi, SVNS, PR, SA, and MSA) obtained the new best solution for 34 (34/157 = 21.66%), 69 (69/157 = 43.95%), 94 (94/157 = 59.87%), 97 (97/157 = 61.78%), 127 (127/157 = 80.89%), 21 (21/157 = 13.38%), 128 (128/157 = 81.53%), 80 (80/157 = 50.96%), 81 (81/157 = 51.60%), 84 (84/157 = 53.50%), 44 (44/157 = 28.03%), 127 (127/157 = 80.89%), 131 (131/157 = 83.44%) and 135 (135/157 = 85.99%) such instances, respectively. MSA obtained the largest number of new best solutions among the 14 approaches.

Table 2
Results for data set 4.

Instance	Best	TMH	GTP	GTF	FVF	SVF	GLS	ASe	ADC	ARC	ASi	SVNS	PR	SA	MSA
p4.2.*a*	**206**	202	**206**	**206**	**206**	**206**	**206**	**206**	**206**	**206**	**206**	202	**206**	**206**	**206**
p4.2.*b*	**341**	**341**	**341**	**341**	**341**	**341**	303	**341**	**341**	**341**	**341**	**341**	**341**	**341**	**341**
p4.2.*c*	**452**	438	**452**	**452**	**452**	**452**	447	**452**	**452**	**452**	**452**	**452**	**452**	**452**	**452**
p4.2.*d*	**531**	517	530	**531**	**531**	**531**	526	**531**	**531**	530	**531**	528	**531**	**531**	**531**
p4.2.*e*	**618**	593	**618**	613	**618**	**618**	602	**618**	600	600	613	593	**618**	**618**	**618**
p4.2.*f*	**687**	666	**687**	676	684	**687**	651	**687**	672	672	672	675	**687**	**687**	**687**
p4.2.*g*	**757**	749	751	756	750	753	734	**757**	756	756	756	750	**757**	756	**757**
p4.2.*h*	**835**	827	795	820	827	**835**	797	827	819	819	820	819	**835**	827	**835**
p4.2.*i*	**918**	915	882	899	916	**918**	826	**918**	900	**918**	**918**	916	**918**	**918**	**918**
p4.2.*j*	**965**	914	946	962	962	962	939	**965**	962	962	962	962	**965**	962	962
p4.2.*k*	**1022**	963	1013	1013	1019	**1022**	994	**1022**	1016	1016	1016	1007	**1022**	**1022**	**1022**
p4.2.*l*	**1074**	1022	1061	1058	1073	**1074**	1051	1071	1070	1071	1069	1051	**1074**	1073	1073
p4.2.*m*	**1132**	1089	1106	1098	**1132**	**1132**	1051	1130	1115	1119	1113	1051	**1132**	**1132**	**1132**
p4.2.*n*	**1174**	1150	1169	1171	1159	1171	1117	1168	1149	1158	1169	1124	1173	1170	**1174**
p4.2.*o*	**1218**	1175	1180	1192	1216	**1218**	1191	1215	1209	1198	1210	1195	**1218**	**1218**	1217
p4.2.*p*	**1242**	1208	1226	1239	1239	1241	1214	**1242**	1229	1233	1239	1237	**1242**	1241	1241
p4.2.*q*	**1265**	1255	1252	1255	**1265**	1263	1248	1263	1253	1252	1260	1239	1263	1255	1259
p4.2.*r*	**1290**	1277	1281	1283	1283	1286	1267	1288	1278	1278	1279	1279	1286	1282	**1290**
p4.2.*s*	**1304**	1294	1296	1299	1300	1301	1286	**1304**	**1304**	1303	**1304**	1295	1296	1298	1300
p4.2.*t*	**1306**	**1306**	**1306**	**1306**	**1306**	**1306**	1294	**1306**	**1306**	**1306**	**1306**	1305	**1306**	1304	**1306**
p4.3.*c*	**193**	192	**193**	**193**	**193**	**193**	**193**	**193**	**193**	**193**	**193**	**193**	**193**	**193**	**193**
p4.3.*d*	**335**	333	**335**	**335**	**335**	**335**	**335**	**335**	333	333	**335**	331	**335**	**335**	**335**
p4.3.*e*	**468**	465	**468**	**468**	**468**	**468**	444	**468**	**468**	**468**	**468**	460	**468**	**468**	**468**
p4.3.*f*	**579**	**579**	**579**	**579**	**579**	**579**	564	**579**	**579**	**579**	**579**	556	**579**	**579**	**579**
p4.3.*g*	**653**	646	651	652	**653**	**653**	644	**653**	652	**653**	652	651	**653**	**653**	**653**
p4.3.*h*	**729**	709	722	727	724	**729**	706	720	713	713	713	718	**729**	**729**	**729**
p4.3.*i*	**809**	785	806	806	806	807	806	796	793	793	786	807	**809**	**809**	**809**
p4.3.*j*	**861**	860	858	858	**861**	**861**	826	**861**	857	855	858	854	**861**	**861**	860
p4.3.*k*	**919**	906	**919**	918	**919**	**919**	864	918	913	910	910	902	918	**919**	**919**
p4.3.*l*	**979**	951	976	973	975	978	960	**979**	958	976	966	969	**979**	977	978
p4.3.*m*	**1063**	1005	1034	1049	1056	**1063**	1030	1053	1039	1028	1046	1047	**1063**	**1063**	**1063**
p4.3.*n*	**1121**	1119	1108	1115	1111	**1121**	1113	**1121**	1109	1112	1103	1106	1120	**1121**	**1121**
p4.3.*o*	**1172**	1151	1156	1157	**1172**	1170	1121	1170	1163	1167	1165	1136	1170	1168	1170
p4.3.*p*	**1222**	1218	1207	1221	1208	**1222**	1190	1221	1202	1207	1207	1200	1220	1219	**1222**
p4.3.*q*	**1253**	1249	1237	1241	1250	1251	1210	1252	1239	1239	1238	1236	**1253**	1251	1251
p4.3.*r*	**1272**	1265	1224	1269	**1272**	**1272**	1239	1267	1263	1263	1263	1250	**1272**	1268	1265
p4.3.*s*	**1295**	1282	1250	1294	1289	1293	1279	1293	1291	1289	1291	1280	1287	**1295**	1293
p4.3.*t*	**1305**	1288	1303	1304	1298	1304	1290	**1305**	1304	1303	1304	1299	1299	1300	1299
p4.4.*e*	**183**	182	**183**	**183**	**183**	**183**	**183**	**183**	**183**	**183**	**183**	**183**	**183**	**183**	**183**
p4.4.*f*	**324**	315	**324**	**324**	**324**	**324**	312	**324**	**324**	**324**	**324**	319	**324**	**324**	**324**
p4.4.*g*	**461**	453	**461**	**461**	**461**	**461**	**461**	**461**	**461**	**461**	460	**461**	**461**	**461**	**461**
p4.4.*h*	**571**	554	**571**	**571**	**571**	**571**	565	**571**	556	556	556	553	**571**	**571**	**571**
p4.4.*i*	**657**	627	655	**657**	**657**	**657**	**657**	**657**	653	652	653	**657**	**657**	**657**	**657**
p4.4.*j*	**732**	**732**	731	731	**732**	**732**	691	**732**	731	711	731	**723**	**732**	**732**	**732**
p4.4.*k*	**821**	819	**821**	816	**821**	**821**	815	**821**	820	818	818	**821**	**821**	**821**	**821**
p4.4.*l*	**880**	875	878	878	879	**880**	852	**880**	877	875	875	876	879	**880**	**880**
p4.4.*m*	**919**	910	916	918	916	**919**	910	918	911	906	911	903	**919**	917	**919**
p4.4.*n*	**977**	**977**	972	976	968	968	942	961	956	956	956	948	969	969	975
p4.4.*o*	**1061**	1014	1057	1057	1051	**1061**	937	1036	1030	1021	1029	1030	1057	**1061**	**1061**
p4.4.*p*	**1124**	1056	1120	1120	1120	1120	1091	1111	1108	1088	1110	1120	1122	**1124**	**1124**
p4.4.*q*	**1161**	1124	1148	1157	1160	**1161**	1106	1145	1150	1137	1148	1149	1160	**1161**	**1161**
p4.4.*r*	**1216**	1165	1203	1211	1207	1203	1148	1200	1195	1195	1194	1193	1213	**1216**	**1216**
p4.4.*s*	**1259**	1243	1245	1256	**1259**	1255	1242	1249	1256	1249	1252	1213	1250	1255	1256
p4.4.*t*	**1285**	1255	1279	**1285**	1282	1279	1250	1281	1281	1283	1281	1281	1280	1284	**1285**

Bold font means the best solution among various algorithms.

The ARPD obtained by MSA is only 0.025% from the best known solution, and is the smallest among 14 approaches, with the ARPD for TMH, GTP, GTF, FVF, SVF, GLS, ASe, ADC, ARC, ASi, SVNS, PR, and SA yielding values of 1.328%, 0.494%, 0.206%, 0.186%, 0.050%, 2.551%, 0.085%, 0.355%, 0.405%, 0.330%, 0.975%, 0.049% and 0.039%, respectively.

The run time depends on a variety of factors, such as the CPU of the machines, the operating system, the compiler, the computer program and the precision used during the execution of the run. Therefore, the relative efficiency of the algorithms is hard to determine. The heuristics of GTP, GTF, FVF, SVF [10] were run on a PC with an Intel Pentium 4 with 2.8 GHz and 1 GB of RAM. The heuristics of ASe, ADC, ARC, ASi [11] were coded in C++ and tested on a PC with 3.0 GHz Intel CPU. The TMH [4] were coded using C++ and run on DEC AlphaServer 1200/533 and DEC Alpha XP 1000 computers with 1 Gigabyte RAM and 1.5 Gigabyte swap. The SVNS [12] and the GLS [13] was run on a PC with an Intel Pentium 4 with 2.8 GHz and 1 GB of RAM. PR [14] was implemented using JAVA 1.6 and run on a PC with an Intel Xeon 2.5 GHz.

The proposed MSA yields better solutions than alternative algorithms for the team orienteering problem within a reasonable computing time. Since computational expenses may vary with hardware, software and programming skills, this study did not directly compare the computational efficiency of different methods. Nonetheless, the proposed MSA is highly effective from a solution quality perspective. Although the proposed MSA requires a longer CPU time than most approaches, this time is worthwhile provided the solution is competitive. Furthermore, the MSA can obtain better solutions than the traditional single-start SA and the computational time required for MSA is less than for SA.

Table 3
Results for data set 5.

Instance	Best	TMH	GTP	GTF	FVF	SVF	GLS	ASe	ADC	ARC	ASi	SVNS	PR	SA	MSA
p5.2.*h*	**410**	**410**	**410**	**410**	**410**	**410**	385	**410**	**410**	**410**	**410**	395	**410**	**410**	**410**
p5.2.*j*	**580**	560	**580**	**580**	**580**	**580**	**580**	**580**	**580**	**580**	**580**	**580**	**580**	**580**	**580**
p5.2.*k*	**670**	**670**	**670**	**670**	**670**	**670**	665	**670**	**670**	**670**	**670**	**670**	**670**	**670**	**670**
p5.2.*l*	**800**	770	**800**	**800**	**800**	**800**	760	**800**	**800**	**800**	**800**	770	**800**	**800**	**800**
p5.2.*m*	**860**	**860**	**860**	**860**	**860**	**860**	830	**860**	**860**	**860**	**860**	**860**	**860**	**860**	**860**
p5.2.*n*	**925**	920	**925**	**925**	**925**	**925**	920	**925**	920	920	**925**	920	**925**	**925**	**925**
p5.2.*o*	**1020**	975	**1020**	**1020**	**1020**	**1020**	1010	**1020**	**1020**	1010	1010	**1020**	**1020**	**1020**	**1020**
p5.2.*p*	**1150**	1090	1130	**1150**	**1150**	**1150**	1030	**1150**	**1150**	**1150**	**1150**	**1150**	**1150**	**1150**	**1150**
p5.2.*q*	**1195**	1185	**1195**	**1195**	**1195**	**1195**	1145	**1195**	**1195**	**1195**	**1195**	**1195**	**1195**	**1195**	**1195**
p5.2.*r*	**1260**	**1260**	**1260**	**1260**	**1260**	**1260**	1225	**1260**	**1260**	**1260**	**1260**	**1260**	**1260**	**1260**	**1260**
p5.2.*s*	**1340**	1310	1330	**1340**	**1340**	**1340**	1325	**1340**	1330	1330	1330	1325	**1340**	**1340**	**1340**
p5.2.*t*	**1400**	1380	1380	**1400**	**1400**	**1400**	1360	**1400**	**1400**	**1400**	**1400**	1380	**1400**	**1400**	**1400**
p5.2.*u*	**1460**	1445	1440	**1460**	**1460**	**1460**	**1460**	**1460**	**1460**	**1460**	**1460**	1450	**1460**	**1460**	**1460**
p5.2.*v*	**1505**	1500	1490	**1505**	1500	**1505**	1500	**1505**	1495	1500	1495	1500	**1505**	**1505**	**1505**
p5.2.*w*	**1565**	1560	1555	**1565**	1560	1560	1560	1560	1555	1555	1555	1560	1560	**1565**	**1565**
p5.2.*x*	**1610**	**1610**	1595	**1610**	1590	**1610**	**1610**	**1610**	**1610**	**1610**	**1610**	1600	**1610**	**1610**	**1610**
p5.2.*y*	**1645**	1630	1635	1635	1635	1635	1630	**1645**	**1645**	**1645**	**1645**	1630	**1645**	**1645**	**1645**
p5.2.*z*	**1680**	1665	1670	**1680**	1670	1670	**1680**	**1680**	**1680**	**1680**	**1680**	1665	**1680**	**1680**	**1680**
p5.3.*k*	**495**	**495**	**495**	**495**	**495**	**495**	470	**495**	**495**	**495**	**495**	**495**	**495**	**495**	**495**
p5.3.*l*	**595**	575	**595**	**595**	**595**	**595**	545	**595**	**595**	**595**	**595**	**595**	**595**	**595**	**595**
p5.3.*n*	**755**	**755**	**755**	**755**	**755**	**755**	720	**755**	**755**	**755**	**755**	**755**	**755**	**755**	**755**
p5.3.*o*	**870**	835	**870**	**870**	**870**	**870**	**870**	**870**	**870**	**870**	**870**	**870**	**870**	**870**	**870**
p5.3.*q*	**1070**	1065	**1070**	**1070**	**1070**	**1070**	1045	**1070**	1065	1065	1065	1065	**1070**	**1070**	**1070**
p5.3.*r*	**1125**	1115	1110	**1125**	**1125**	**1125**	1090	**1125**	1120	**1125**	**1125**	**1125**	**1125**	**1125**	**1125**
p5.3.*s*	**1190**	1175	1185	**1190**	**1190**	**1190**	1145	**1190**	**1190**	**1190**	1185	1185	**1190**	**1190**	**1190**
p5.3.*t*	**1260**	1240	1250	**1260**	**1260**	**1260**	1240	**1260**	1250	1255	**1260**	**1260**	**1260**	**1260**	**1260**
p5.3.*u*	**1345**	1330	1340	**1345**	**1345**	**1345**	1305	**1345**	1330	1335	1335	**1345**	**1345**	**1345**	**1345**
p5.3.*v*	**1425**	1410	1420	**1425**	**1425**	**1425**	**1425**	**1425**	**1425**	**1425**	1420	**1425**	**1425**	**1425**	**1425**
p5.3.*w*	**1485**	1465	**1485**	**1485**	**1485**	**1485**	1460	**1485**	1465	1465	1465	1475	**1485**	**1485**	**1485**
p5.3.*x*	**1555**	1530	**1555**	**1555**	**1555**	**1555**	1520	1540	1535	1540	1540	1535	1550	**1555**	**1555**
p5.3.*y*	**1595**	1580	1590	**1595**	**1595**	**1595**	1590	1590	1590	1590	1590	1580	1590	1590	1590
p5.3.*z*	**1635**	**1635**	1625	**1635**	**1635**	**1635**	**1635**	**1635**	**1635**	**1635**	**1635**	**1635**	**1635**	**1635**	**1635**
p5.4.*m*	**555**	**555**	**555**	**555**	**555**	**555**	550	**555**	**555**	**555**	**555**	550	**555**	**555**	**555**
p5.4.*o*	**690**	680	**690**	**690**	**690**	**690**	680	**690**	**690**	**690**	**690**	**690**	**690**	**690**	**690**
p5.4.*p*	**765**	760	**765**	**765**	**765**	**765**	760	**765**	760	760	760	760	760	**765**	**765**
p5.4.*q*	**860**	**860**	**860**	**860**	**860**	**860**	830	**860**	**860**	**860**	**860**	835	**860**	**860**	**860**
p5.4.*r*	**960**	**960**	**960**	**960**	**960**	**960**	890	**960**	**960**	**960**	**960**	**960**	**960**	**960**	**960**
p5.4.*s*	**1030**	1000	1025	**1030**	**1030**	**1030**	1020	**1030**	**1030**	**1030**	**1030**	1020	1025	**1030**	**1030**
p5.4.*t*	**1160**	1100	**1160**	**1160**	**1160**	**1160**	**1160**	**1160**	**1160**	**1160**	**1160**	**1160**	**1160**	**1160**	**1160**
p5.4.*u*	**1300**	1275	**1300**	**1300**	**1300**	**1300**	**1300**	**1300**	**1300**	**1300**	**1300**	**1300**	**1300**	**1300**	**1300**
p5.4.*v*	**1320**	1310	**1320**	**1320**	**1320**	**1320**	1245	**1320**	**1320**	**1320**	**1320**	**1320**	**1320**	**1320**	**1320**
p5.4.*w*	**1390**	1380	1375	**1390**	**1390**	**1390**	1330	**1390**	1380	**1390**	1380	1380	**1390**	**1390**	**1390**
p5.4.*x*	**1450**	1410	1440	**1450**	**1450**	**1450**	1410	**1450**	**1450**	**1450**	**1450**	1440	**1450**	**1450**	**1450**
p5.4.*y*	**1520**	**1520**	**1520**	**1520**	**1520**	**1520**	1485	**1520**	1510	1510	1500	1500	**1520**	**1520**	**1520**
p5.4.*z*	**1620**	1575	**1620**	**1620**	**1620**	**1620**	1590	**1620**	**1620**	1575	1580	1600	**1620**	**1620**	**1620**

Bold font means the best solution among various algorithms.

Table 4
Results for data set 6.

Instance	Best	TMH	GTP	GTF	FVF	SVF	GLS	ASe	ADC	ARC	ASi	SVNS	SPR	SA	MSA
p6.2.*d*	**192**	**192**	**192**	**192**	**192**	**192**	180	**192**	**192**	**192**	**192**	**192**	**192**	**192**	**192**
p6.2.*j*	**948**	936	**948**	**948**	**948**	**948**	**948**	**948**	**948**	**948**	**948**	**948**	**948**	**948**	**948**
p6.2.*l*	**1116**	**1116**	1098	1110	**1116**	**1116**	1104	**1116**	1110	**1116**	**1116**	**1116**	**1116**	**1116**	**1116**
p6.2.*m*	**1188**	**1188**	1164	**1188**	**1188**	**1188**	1164	**1188**	**1188**	**1188**	**1188**	**1188**	**1188**	**1188**	**1188**
p6.2.*n*	**1260**	**1260**	1242	**1260**	**1260**	**1260**	1254	**1260**	**1260**	1254	**1260**	1248	**1260**	**1260**	**1260**
p6.3.*g*	**282**	**282**	**282**	**282**	**282**	**282**	264	**282**	**282**	**282**	**282**	276	**282**	**282**	**282**
p6.3.*h*	**444**	**444**	**444**	**444**	**444**	**444**	**444**	**444**	**444**	438	438	**444**	**444**	**444**	**444**
p6.3.*i*	**642**	612	**642**	**642**	**642**	**642**	**642**	**642**	**642**	**642**	**642**	**642**	**642**	**642**	**642**
p6.3.*k*	**894**	876	**894**	**894**	**894**	**894**	882	**894**	888	888	**894**	**894**	**894**	**894**	**894**
p6.3.*l*	**1002**	990	**1002**	**1002**	**1002**	**1002**	990	**1002**	**1002**	**1002**	**1002**	996	**1002**	**1002**	**1002**
p6.3.*m*	**1080**	**1080**	**1080**	**1080**	**1080**	**1080**	1068	**1080**	1074	**1080**	**1080**	**1080**	**1080**	**1080**	**1080**
p6.3.*n*	**1170**	1152	**1170**	**1170**	**1170**	**1170**	1140	**1170**	1164	1164	1164	1152	**1170**	**1170**	**1170**
p6.4.*j*	**366**	**366**	**366**	**366**	**366**	**366**	360	**366**	**366**	**366**	**366**	**366**	**366**	**366**	**366**
p6.4.*k*	**528**	522	**528**	**528**	**528**	**528**	**528**	**528**	**528**	**528**	**528**	**528**	**528**	**528**	**528**
p6.4.*l*	**696**	**696**	**696**	**696**	**696**	**696**	678	**696**	**696**	**696**	**696**	678	**696**	**696**	**696**

Bold font means the best solution among various algorithms.

Table 5
Results for data set 7.

Instance	Best	TMH	GTP	GTF	FVF	SVF	GLS	ASe	ADC	ARC	ASi	SVNS	PR	SA	MSA
p7.2.*d*	**190**	**190**	**190**	**190**	**190**	**190**	**190**	**190**	**190**	**190**	**190**	182	**190**	**190**	**190**
p7.2.*e*	**290**	**290**	**290**	**290**	289	**290**	279	**290**	**290**	**290**	**290**	289	**290**	**290**	**290**
p7.2.*f*	**387**	382	**387**	**387**	**387**	**387**	340	**387**	**387**	**387**	**387**	**387**	**387**	**387**	**387**
p7.2.*g*	**459**	**459**	456	**459**	**459**	**459**	440	**459**	**459**	**459**	**459**	457	**459**	**459**	**459**
p7.2.*h*	**521**	**521**	520	520	**521**	**521**	517	**521**	**521**	**521**	**521**	**521**	**521**	**521**	**521**
p7.2.*i*	**580**	578	579	579	575	579	568	**580**	579	579	579	579	**580**	579	579
p7.2.*j*	**646**	638	643	644	643	644	633	**646**	**646**	**646**	**646**	632	**646**	**646**	**646**
p7.2.*k*	**705**	702	702	**705**	704	**705**	691	**705**	704	704	704	700	**705**	**705**	**705**
p7.2.*l*	**767**	**767**	758	**767**	759	**767**	748	**767**	**767**	**767**	**767**	758	**767**	**767**	**767**
p7.2.*m*	**827**	817	**827**	824	824	**827**	798	**827**	**827**	**827**	**827**	**827**	**827**	**827**	**827**
p7.2.*n*	**888**	864	884	**888**	883	**888**	861	**888**	878	878	878	866	**888**	**888**	**888**
p7.2.*o*	**945**	914	933	**945**	**945**	**945**	897	**945**	**945**	940	941	928	**945**	**945**	**945**
p7.2.*p*	**1002**	987	1000	**1002**	**1002**	**1002**	954	**1002**	991	993	993	955	**1002**	**1002**	**1002**
p7.2.*q*	**1044**	1017	1041	1043	1038	**1044**	1031	1043	1042	1043	1043	1029	**1044**	1043	1043
p7.2.*r*	**1094**	1067	1091	1088	**1094**	**1094**	1075	**1094**	1093	1088	**1094**	1069	**1094**	**1094**	1093
p7.2.*s*	**1136**	1116	1123	1128	**1136**	**1136**	1102	**1136**	**1136**	1134	1131	1118	**1136**	1135	1135
p7.2.*t*	**1179**	1165	1172	1174	1168	**1179**	1142	**1179**	**1179**	**1179**	**1179**	1154	1175	**1179**	1172
p7.3.*h*	**425**	416	**425**	**425**	**425**	**425**	418	**425**	**425**	**425**	**425**	**425**	**425**	**425**	**425**
p7.3.*i*	**487**	481	**487**	**487**	**487**	**487**	480	**487**	**487**	486	**487**	480	**487**	**487**	**487**
p7.3.*j*	**564**	563	**564**	**564**	562	**564**	539	**564**	**564**	**564**	**564**	543	**564**	**564**	**564**
p7.3.*k*	**633**	632	**633**	**633**	632	**633**	586	**633**	632	**633**	**633**	**633**	**633**	**633**	**633**
p7.3.*l*	**684**	681	683	679	681	681	668	**684**	683	**684**	**684**	681	**684**	**684**	**684**
p7.3.*m*	**762**	756	749	755	745	**762**	735	**762**	**762**	**762**	**762**	743	**762**	**762**	**762**
p7.3.*n*	**820**	789	810	811	814	**820**	789	**820**	819	819	**820**	804	**820**	**820**	**820**
p7.3.*o*	**874**	**874**	873	865	871	**874**	833	**874**	**874**	**874**	**874**	841	**874**	**874**	**874**
p7.3.*p*	**929**	922	917	923	926	927	912	**929**	925	926	925	918	927	927	927
p7.3.*q*	**987**	966	976	**987**	978	**987**	945	**987**	**987**	**987**	**987**	966	**987**	**987**	**987**
p7.3.*r*	**1026**	1011	1018	1022	1024	1022	1015	**1026**	1024	1021	1022	1009	1021	1023	**1026**
p7.3.*s*	**1081**	1061	**1081**	**1081**	1079	1079	1054	**1081**	**1081**	**1081**	1077	1070	**1081**	**1081**	**1081**
p7.3.*t*	**1119**	1098	1114	1116	1112	1115	1080	1118	1117	1103	1117	1109	1118	**1119**	**1119**
p7.4.*g*	**217**	**217**	**217**	**217**	**217**	**217**	209	**217**	**217**	**217**	**217**	**217**	**217**	**217**	**217**
p7.4.*h*	**285**	**285**	**285**	**285**	**285**	**285**	**285**	**285**	**285**	**285**	**285**	283	**285**	**285**	**285**
p7.4.*i*	**366**	359	**366**	**366**	**366**	**366**	359	**366**	**366**	**366**	**366**	364	**366**	**366**	**366**
p7.4.*k*	**520**	503	**520**	**520**	518	**520**	511	**520**	**520**	**520**	**520**	518	518	**520**	**520**
p7.4.*l*	**590**	576	**590**	588	588	**590**	573	**590**	**590**	**590**	**590**	575	**590**	**590**	**590**
p7.4.*m*	**646**	643	644	**646**	**646**	**646**	638	**646**	644	**646**	**646**	639	**646**	**646**	**646**
p7.4.*n*	**730**	726	723	721	715	**730**	698	**730**	725	725	726	723	**730**	**730**	**730**
p7.4.*o*	**781**	776	772	778	770	**781**	761	**781**	778	**781**	778	778	780	**781**	**781**
p7.4.*p*	**846**	832	841	839	**846**	**846**	803	**846**	**846**	838	842	841	**846**	**846**	**846**
p7.4.*q*	**909**	905	902	898	899	906	899	**909**	**909**	**909**	**909**	896	907	**909**	**909**
p7.4.*r*	**970**	966	**970**	969	**970**	**970**	937	**970**	**970**	**970**	**970**	964	**970**	**970**	**970**
p7.4.*s*	**1022**	1019	1021	1020	1021	**1022**	1005	**1022**	1019	1021	1019	1019	**1022**	**1022**	**1022**
p7.4.*t*	**1077**	1067	1071	1071	**1077**	**1077**	1020	**1077**	1072	**1077**	**1077**	1073	**1077**	**1077**	**1077**

Bold font means the best solution among various algorithms.

To make more rigorous comparisons, a group of one-sided paired-samples *t*-tests with respect to the relative percentage deviation (RPD = (Obj − BKS)/BKS × 100%) for each instance was performed to compare the proposed MSA algorithm with TMH, GTP, GTF, FVF, SVF, GLS, ASe, ADC, ARC, ASi, SVNS, PR, and SA. At confidence level $\alpha = 0.05$, the results listed in Table 7 demonstrate that the proposed MSA algorithm significantly outperformed the best existing approaches. Furthermore, these statistical results also clearly show that the multi-start hill climbing strategy significantly improved the performance of the single-start SA algorithm.

Table 6
Results summary.

	TMH	GTP	GTF	FVF	SVF	GLS	ASe	ADC	ARC	ASi	SVNS	PR	SA	MSA
BEST														
Set 4	5	16	15	25	36	6	30	12	12	13	7	36	34	39
Set 5	12	26	45	40	42	9	42	31	31	30	21	40	44	44
Set 6	9	12	14	15	15	4	15	11	11	13	10	15	15	15
Set 7	8	15	20	17	35	2	41	26	27	28	6	36	38	37
All	34	69	94	97	128	21	128	80	81	84	44	127	131	135
APRD (%)														
Set 4	1.989	0.864	0.477	0.276	0.110	2.959	0.302	0.893	0.978	0.765	1.462	0.106	0.130	0.065
Set 5	1.381	0.345	0.014	0.069	0.034	2.391	0.036	0.231	0.257	0.284	0.610	0.047	0.007	0.007
Set 6	0.788	0.337	0.036	0.000	0.000	1.776	0.000	0.152	0.201	0.124	0.520	0.000	0.000	0.000
Set 7	1.156	0.432	0.296	0.399	0.056	3.077	0.004	0.144	0.186	0.148	1.307	0.043	0.020	0.029
All	1.328	0.494	0.206	0.186	0.050	2.551	0.085	0.355	0.405	0.330	0.975	0.049	0.039	0.025
Avg. CPU (s)														
Set 4	796.7	105.3	282.9	22.5	457.9	11.4	37.1	32.0	30.7	32.0	7.4	36.7	97.3	81.0
Set 5	71.3	69.5	26.6	34.2	158.9	3.5	17.4	15.1	14.3	15.1	1.5	11.2	6.7	6.6
Set 6	45.7	66.3	20.2	8.7	147.9	4.3	16.1	14.1	13.5	14.2	1.9	9.0	1.7	1.4
Set 7	432.6	160.0	256.8	10.3	309.9	12.1	30.4	24.6	23.3	24.7	4.3	27.3	38.5	32.2
All	336.6	100.0	146.6	18.9	268.6	7.8	25.2	21.4	20.5	21.5	3.8	21.2	38.8	36.3

Table 7
Analytical results of the paired-samples *t*-tests with respect to RPD for each benchmark instance.

MSA V.S.	TMH	GTP	GTF	FVF	SVF	GLS	ASe	ADC	ARC	ASi	SVNS	PR	SA
Paired difference (RPD)	−1.4394	−0.5142	−0.2203	−0.1916	−0.0303	−2.683	−0.0830	−0.3949	−0.4479	−0.3647	−1.0531	−0.0292	−0.0199
t-value	−12.0222	−8.0529	−5.8049	−6.0212	−2.1310	−13.910	−2.8837	−7.3359	−7.1636	−7.0993	−10.876	−2.4959	−1.9272
Degree of freedom	156	156	156	156	156	156	156	156	156	156	156	156	156
P-value	<0.0001	<0.0001	<0.0001	<0.0001	0.0173	<0.0001	0.00224	<0.0001	<0.0001	<0.0001	<0.0001	0.0068	0.0279

5. Conclusions and future research

This study applied the multi-start simulated annealing to team orienteering problem. The main contributions of this study include:

(1) This study proposes a MSA heuristic which associates the main properties of the SAs (such as effective convergence, efficient use of memory, easy hardware implementation) with those of multi-start hill climbing strategies (such as sufficient diversification, excellent capability to escape local optimality, and efficient sampling of the solution space) for the team orienteering problem.

(2) Comparing the obtained results with the best known solutions in terms of a wide range of benchmark instances reveals that the proposed MSA algorithm is more effective in maximizing total score than the best existing algorithms and the traditional single-start SA. Given that the team orienteering problem is an extremely challenging NP-hard complete problem, the results obtained using the proposed MSA algorithm clearly indicate that this approach contributes considerably to the research on the optimum techniques for solving this problem. The proposed MSA algorithm obtained 135 best solutions to the 157 benchmark problems, including five new best solutions.

Several possible future directions exist for this research. One such possibility is for future studies to consider applying hybrid and meta-heuristic methods, such as artificial immune system (AIS) and artificial bee colony (ABC), to this problem. Another interesting potential research direction is to extend the proposed MSA algorithm to solve more complex scheduling problems, such as the team orienteering problem with time windows. Finally, extensions of the proposed MSA algorithm to solve the problem considered in this study with other performance criteria using multiple criteria also deserve further investigation.

Acknowledgement

The author would like to thank the National Science Council of the Republic of China (Taiwan), for financially supporting this research under Contract No. NSC101-2410-H-182-004-MY2.

References

[1] S.E. Butt, T.M. Cavalier, A heuristic for the multiple tour maximum collection problem, Computers & Operations Research 21 (1) (1994) 101–111.
[2] I. Chao, B. Golden, E. Wasil, The team orienteering problem, European Journal of Operational Research 88 (3) (1996) 464–474.
[3] B.L. Golden, L. Levy, R. Vohra, The orienteering problem, Naval Research Logistics 34 (3) (1987) 307–318.
[4] H. Tang, E. Miller-Hooks, A tabu search heuristic for the team orienteering problem, Computers & Operations Research 32 (6) (2005) 1379–1407.
[5] S. Butt, D. Ryan, An optimal solution procedure for the multiple tour maximum collection problem using column generation, Computers & Operations Research 26 (4) (1999) 427–441.
[6] S. Boussier, D. Feillet, M. Gendreau, An exact algorithm for the team orienteering problem, 4OR 5 (3) (2007) 211–230.
[7] I. Chao, B. Golden, E. Wasil, A fast and effective heuristic for the orienteering problem, European Journal of Operational Research 88 (3) (1996) 475–489.
[8] T. Tsiligirides, Heuristic methods applied to oritnteering, Journal of the Operational Research Society 35 (9) (1984) 797–809.
[9] Y. Rochat, E. Taillard, Probabilistic diversification and intensification in local search for vehicle routing, Journal of Heuristics 1 (1) (1995) 147–167.
[10] C. Archetti, A. Hertz, M.G. Speranza, Metaheuristics for the team orienteering problem, Journal of Heuristics 13 (1) (2007) 49–76.
[11] L. Ke, C. Archetti, Z. Feng, Ants can solve the team orienteering problem, Computers & Industrial Engineering 54 (3) (2008) 648–665.
[12] P. Vansteenwegen, W. Souffriau, G.V. Berghe, D. Van Oudheusden, Metaheuristics for tourist trip planning Metaheuristics in the Service Industry. Lecture Notes in Economics and Mathematical Systems, vol. 624, 2009, pp. 15–31.
[13] P. Vansteenwegen, W. Souffriau, G.V. Berghe, D. Van Oudheusden, A guided local search metaheuristic for the team orienteering problem, European Journal of Operational Research 196 (1) (2009) 118–127.

[14] W. Souffriau, P. Vansteenwegen, G.V. Berghe, D. Van Oudheusden, Path relinking approach for the team orienteering problem, Computers & Operations Research 37 (11) (2010) 1853–1859.
[15] P. Vansteenwegen, W. Souffriau, D. Van Oudheusden, The orienteering problem: a survey, European Journal of Operational Research 209 (1) (2011) 1–10.
[16] R. Zhang, C. Wu, A hybrid immune simulated annealing algorithm for the job shop scheduling problem, Applied Soft Computing 10 (1) (2010) 79–89.
[17] S.-W. Lin, V.F. Yu, S.-Y. Chou, Solving the truck and trailer routing problem based on a simulated annealing heuristic, Computers & Operations Research 36 (5) (2009) 1683–1692.
[18] N. Li, J. Cha, Y. Lu, A parallel simulated annealing algorithm based on functional feature tree modeling for 3D engineering layout design, Applied Soft Computing 10 (2) (2010) 592–601.
[19] K.-C. Ying, S.-W. Lin, C.-C. Lu, Cell formation using a simulated annealing algorithm with variable neighbourhood, European Journal of Industrial Engineering 5 (1) (2011) 22–42.
[20] X. Genga, Z. Chen, W. Yang, D. Shi, K. Zhao, Solving the traveling salesman problem based on an adaptive simulated annealing algorithm with greedy search, Applied Soft Computing 11 (4) (2011) 3680–3689.
[21] X. Zhao, Simulated annealing algorithm with adaptive neighborhood, Applied Soft Computing 11 (2) (2011) 1827–1836.
[22] P. Vasant, Hybrid simulated annealing and genetic algorithms for industrial production management problems, International Journal of Computational Methods 7 (2) (2010) 279–297.
[23] S.-W. Lin, V.F. Yu, A simulated annealing heuristic for the team orienteering problem with time windows, European Journal of Operational Research 217 (1) (2012) 94–107.
[24] S.-W. Lin, K.-C. Ying, C.-C. Lu, J.N.D. Gupta, Applying multi-start simulated annealing to schedule a flowline manufacturing cell with sequence dependent family setup times, International Journal of Production Economics 130 (2011) 246–254.
[25] S-W Lin, K-C. Ying, Minimizing makespan and total flowtime in permutation flowshops by a bi-objective multi-start simulated-annealing algorithm, Computers & Operations Research, http://dx.doi.org/10.1016/j.cor.2011.08.009, in press.
[26] S.-W. Lin, K.-C. Ying, Scheduling a bi-criteria flowshop manufacturing cell with sequence-dependent family setup times, European Journal of Industrial Engineering 6 (4) (2012) 474–496.
[27] S.-W. Lin, C.-Y. Huang, C.-C. Lu, K.-C. Ying, Minimizing total flow time in permutation flowshop environment, International Journal of Innovative Computing, Information and Control 8 (10 A) (2012) 6599–6612.
[28] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, Equations of state calculations by fast computing machines, Journal of Chemical Physics 21 (1953) 1087–1092.
[29] S. Kirkpatrick, C.D. Gelatt Jr., M.P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.
[30] V. Cerny, Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm, Journal of Optimization Theory and Applications 45 (1) (1985) 45–51.