Research Journal of Applied Sciences 8 (3): 181-190, 2013
ISSN: 1815-932X

Scatter Search for Solving Team Orienteering Problem

[1]Hamzah Ali Alkhazaleh, [1]Masri Ayob and [2]Zulkifli Ahmad
[1]Center for Artificial Intelligence, Faculty of Information Science and Technology,
[2]School of Linguistics and Language Studies, Faculty of Social Sciences and Humanities,
Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia

Abstract: This research proposes a scatter search metaheuristic approach for solving Team Orienteering Problem. The goal is to build a particular number of routes that visit some points to maximize the sum of the score while the route's length does not exceeding the time budget. The approach is compared to other state-of-art approaches and tested using a large set of test instances from the literature. The obtained results are competitive comparing the best known results of these heuristics but the computational time is reduced significantly.

Key words: Scatter search, team orienteering problem, heuristic search, computational time, metaheuristic approach

INTRODUCTION

Orienteering Problem (OP) originates from sport skiing game of orienteering (Chao *et al*., 1996) where the players, start at one point and end at other point in a certain amount of time. The player needs to collect as many as possible the scores from the visited points where each point has its own score. The goal is to maximize the total collected scores before reaching the end points within a given amount of timeframe. The Team Orienteering Problem (TOP) (Chao *et al*., 1996; Tang and Miller-Hooks, 2005) (or called as multiple tour maximum collection problem (Butt and Cavalier, 1994)) is an OP where the goal is to determine the path, P that gives total high scores, limited to T_{max}. The TOP consists of a team of several players each collecting scores during the same time span.

The OP is also applicable for several problems such as selective travelling salesperson problem (Gendreau *et al*., 1998), maximum collection problem and intelligent tourist travel guide system (Vansteenwegen *et al*., 2009b). These problems were studied and many exact and heuristic solutions were used over the past years and were modeled as orienteering problem.

There are several exact methods that have been proposed to solve OP such as branch and cut (Fischetti *et al*., 1998) and branch and bound (Ramesh *et al*., 1992). Based on Fischetti *et al*. (1998), instances up to 500 location can be solved by branch and cut procedure. Golden *et al*. (1988) proved that OP is a Non-deterministic Polynomial-time (NP)-hard problem. Therefore, the exact methods could not solve this problem in a reasonable computation time.

Recently, many researchers focused on applying metaheuristic approaches to solve TOP. For instance, Ke *et al*. (2008) proposed Ant Colony Optimization (ACO) with four methods such as the sequential, deterministic-concurrent, random-concurrent and simultaneous methods to construct candidate solutions. These methods are used to obtain best known solution quality within less than one minute of calculation time. Vansteenwegen *et al*. (2009a) proposed an algorithm that combines Guided Local Search (GLS) and a skewed Variable Neighborhood Search (VNS) algorithm to solve the TOP. More recently, a memetic algorithm was proposed by Bouly *et al*. (2010). They utilize an optimal split procedure for chromosome evaluation and local search techniques for mutation. This approach had shown that it could obtain good solution within less than one minute of computation time.

As far as researchers are concerned, none of the researches were reported to apply scatter search in solving TOP. Scatter search has been proposed by Glover (1998) as a population-based meta-heuristic in which solutions are intelligently combined to yield better solutions (Glover *et al*., 2003). It has been successfully used to solve a variety of scheduling problem such as exam timetabling (Sabar and Ayob, 2009), course timetabling (Jaradat and Ayob, 2011), traveling

Corresponding Author: Masri Ayob, Center for Artificial Intelligence, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia

salesman problems (Liu, 2008), vehicle routing problem (Maquera *et al*., 2011), flow shop scheduling problem (Engin *et al*., 2009) and nurse rostering problem (Burke *et al*., 2010). Therefore, this research investigates the performance of applying scatter search algorithm to solve TOP.

PROBLEM FORMULATION

The TOP is formulated as in Eq. 1 (Vansteenwegen *et al*., 2009a) where the decision variables: x_{ijr} = 1 if in route r, a visit to point i is followed by visit to point j or 0 otherwise; y_{ir} = 1 if point i is visited in route r or 0 otherwise; u_{ir} = the position of point i in route r, t_{ij} is the distance between points i and j, S_i is the score at the ith point:

$$\text{Max}\sum_{r=1}^{R}\sum_{i=2}^{N-1} S_i y_{ir} \quad (1)$$

$$\sum_{r=1}^{R}\sum_{j=2}^{N} x_{1jr} = \sum_{r=1}^{R}\sum_{i=2}^{N-1} x_{iNr} = R \quad (2)$$

$$\sum_{r=1}^{R} y_{mr} \leq 1; \forall m = 2,\ldots,N-1 \quad (3)$$

$$\sum_{i=1}^{N-1} x_{ikr} = \sum_{j=2}^{N} x_{kjr} = y_{kr}; \forall k=2,\ldots,N-1; \forall r=1,\ldots,R \quad (4)$$

$$\sum_{i=1}^{N-1}\sum_{j=2}^{N} t_{ij}x_{ijr} \leq T_{max}; \forall r=1,\ldots,R \quad (5)$$

$$2 \leq \mu_{ir} \leq N; \forall i=2,\ldots,N; \forall r=1,\ldots,R \quad (6)$$

$$\mu_{ir}-\mu_{jr}+1 \leq (N-1)(1-x_{ijr}); \forall i,j=2,\ldots,N; \forall r=1,\ldots,R \quad (7)$$

$$x_{ijr}, y_{ir} \in \{0,1\}; \forall i,j=1,\ldots,N; \forall r=1,\ldots,R \quad (8)$$

The Eq. 1 is to maximize the total collected score, where R is the total number of route and N is the total points to be visited. Constraint (2) is to ensure that each route start at point 1 and ends at point N. Constraint (3) ensures that each point is only visited one time at most. Constraint (4) ensures that if a point is visited in a given route, it is outstripped and followed by an exact one other visit in the same route. Constraint (5) guarantees that each route does not exceed the time budget T_{max}. Constraint (6) and (7) are important to prohibit sub-routes.

These sub-routes elimination constraints are formulated according to Miller-Tucker-Zemlin (MTZ) formulation of the TSP (Miller *et al*., 1960).

SCATTER SEARCH ALGORITHM

Scatter Search (SS) Algorithm was introduced by Glover (1977). Scatter search is an evolutionary algorithm which is very flexible in incorporating strategies for search intensification and diversification. Each of its elements can be implemented in a variety of ways according to the specific domain problem (Laguna and Armentano, 2005). Similarly to other evolutionary methods, SS operates with a population of solution rather than with a single solution at a time and employs procedure to combine these solutions to obtain new trial solution. SS explores solutions in the search space and selects a set of reference points. The reference set that consist of elite solutions (with good solution quality pool and diverse pool) is evolved and updated (Laguna *et al*., 2003; Resende *et al*., 2010). SS consists of five components (Resende *et al*., 2010):

- A Diversification Generation Method is used to generate a collection of diverse trial solutions using one or more arbitrary trial solutions as an input
- Improvement method is used to transforms a trial solution into one or more enhanced trial solutions using any S-metaheuristic. In general, a candidate local search algorithm is applied and then a local optimum is generated
- A Reference Set Update Method is used to construct and maintain a reference set consisting of the b best solutions found (where the value of b is typically small, e.g., not >20), organized to provide efficient access by other component procedure. Several alternative criteria may be used to add solutions to the reference set and delete solutions from the reference set
- A Subset Generation Method is used to operate on the reference set to produce a subset of its solutions as a basis for creating combined solutions. The most common subset generation method is to generate all pairs of reference solutions
- A Solution Combination Method is used to transform a given subset of solutions produced by the Subset Generation Method into one or more combined solutions

Researchers applied Scatter Search algorithm to tackle TOP based on Resende *et al*. (2010). The pseudo code of SS is shown:

The pseudo code of scatter search for TOP:

1. Start with P = Ø, |P| = 50. Use the Diversification Generation Method to construct a solution and apply the improvement Method. Let χ be the resulting solution. If $\chi \notin P$ then add χ to P, otherwise, discard χ.
   Repeat step 1 until |P| = PSize.
2. Build reference set |RefSet| = 20, RefSet = [X^1-, ..., X^b], [Y^1, ..., Y^d] with the best b1 and divers b_2 solutions in P. Order the solutions in b_1 according to their objective function value such that χ^1 is the best solution and χ^b the worst.

while (StoppingCriterion)

3. Generate NewSubsets with the subset generation Method.
   Select the subset of solutions randomly that contain 2 or 3 or 4 solutions.
4. Apply the solution combination Method to NewSubsets to obtain one solutionχ'.
   Select one the combination point randomly.
5. Apply the improvement method to the χ'.
6. Update the reference set. // if χ' better than the worse solution in b_1 replace with χ if replace χ' with the worse divers in b_2

End While

Diversification Generation Method: SS contains a diversification generation method to generate m of diverse trial solution (m = 50 in this study) by employing controlled randomization and frequency memory. A large number of diverse solutions have been generated randomly to obtain diverse region of the solution space. All the generated solutions considered in this research are feasible.

Improvement method: This component is used to enhance the solution generated via local search procedure which drives the solution to local optima. Researchers randomly generate a neighbor from five neighborhood structures (Vansteenwegen *et al*., 2009a) at each iteration. Researchers use hill climbing as the local search to search for a better quality solution. Researchers use consecutive non improvement iterations as a termination criterion.

A Reference Set Update Method: After building the population (researchers use 50 diverse feasible solutions), researchers apply References Set Update Method to build RefSet which consists of two pools of elites solutions that were selected from the population based on best quality and best diversity. The RefSet was built in the research as follow: selects the best 10 quality solution from the population and store that in b1 and eliminated the solutions from population. Measure the diversity of the rest of solutions in population with the solution in b1 by calculating the distance. The distance between two solutions is calculated based on Maquera *et al*. (2011). Researchers calculate the average distance between each solution in the population with the solutions in bland then select 10 solutions that have the highest average distance and add it to b2 as best diversity in the population.

Subset Generation Method: Subset Generation Method plays an important role in selecting pair for combination method to generate new solution. In the research, researchers follow Glover's Method (Glover, 1999) and generate the subsets as 2-4 solution for the following combination.

Solution Combination Method: The main goal of this component is to generate new trial solution in another region of the solution space. In this study, the highest quality solution in the subset is selected and then shares 25% from other solutions with the highest quality solution. The new solution has an attributes of quality solution and perhaps infeasible. Repair method is applied on the infeasible solution to restore feasibility by removing the repeated points and the lowest score to fix the exceeding time budget.

The improvement method is employed in new solution to gain better solution quality by exploring its neighborhood. Then, researchers employ the references set update method to update the RefSet to the new solution. If the new solution's quality is better than worse solution in best quality list (b_1) the new solution will replace it. If not we will update the best divers list (b_2) by measuring the differences with the solutions in best quality list (b_1) and compare it with existing solution in best divers list (b_2) then researchers replace the worse divers by new solutions.

If there is no more new good solution, researchers will pick up one solution randomly from the population which contain 30 solutions. The operations of subset generation method until references update method are repeated until there is no more updating process on the references set.

EXPERIMENTAL SETUP AND RESULTS

In this study, scatter search were examined on large numbers of test problem. There are four data sets used to benchmark different approaches from (Chao *et al*., 1996) each data sets contain different numbers of locations: n = 100 (date set 4), n = 66 (data set 5), n = 64 (data set 6) and n = 102 (date set 7) including start and end location. Each set contains instances with r equal to 2, 3 and 4 routes. The time budget T_{max} differs for each data set. The Scatter Search is coded in Java 1.7 and performed on Intel Pentium (R) Daul-Core 3.0 GHz CPU personal computer with 2 gigabyte RAM, running on Windows Vista operating system (32-bit).

The parameters for SS is set based on the preliminary experiment on the population size, reference set size and the number of non-improvement iteration. These are 50 diverse solutions for the population size, the reference set size = 20 (i.e., 10 good quality and 10 diverse solutions) and the non-improvement iteration = 200 iterations. The experimental results from the SS are compared with the best known results and results of the following research:

- A1: Tabu search by Tang and Miller-Hooks (2005)
- A2: Tabu search with penalty strategy by Archetti *et al.* (2007)
- A3: Tabu search with feasible strategy by Archetti *et al.* (2007)
- A4: Fast Variable Neighborhood by Archetti *et al.* (2007)
- A5: Slow Variable Neighborhood by Archetti *et al.* (2007)
- A6: Guided Local Search by Vansteenwegen *et al.* (2009a)
- A7: Sequential ant colony optimization by Ke *et al.* (2008)
- A8: Deterministic concurrent ant colony optimization by Ke *et al.* (2008)
- A9: Random concurrent ant colony optimization by Ke *et al.* (2008)
- A10: Simultaneous ant colony optimization by Ke *et al.* (2008)
- A11: Skewed Variable Neighborhood search by Vansteenwegen *et al.* (2009b)
- A12: Memetic Algorithm by Bouly *et al.* (2010)
- A13: Fast Path relinking by Souffriau *et al.* (2010)
- A14: Slow Path relinking by Souffriau *et al.* (2010)
- A15: Discrete Particle Swarm Optimization by Muthuswamy and Lam (2011)

Researchers exclude the instance which it the same result was obtained by above algorithms from the comparison. The comparison will be in 158 instances (Archetti *et al*., 2007).

Scatter search had been runned 31 times for each instance. Table 1-4 summarized the statistical analysis of the results which include minimum and maximum results researchers then calculate the average and standard deviation for each instance. In Table 5-8 each test set will be compared to the best score for the algorithm with the best scores found by algorithms from the past studies.

Table 1: Test set 4

Instances	Max.	Min.	Avg.	SD
p4.2.a	206	206	206.0	0.000
p4.2.b	341	340	340.5	0.506
p4.2.c	452	441	451.3	2.250
p4.2.d	531	515	524.6	3.630
p4.2.e	618	589	609.1	9.946
p4.2.f	678	643	664.2	6.107
p4.2.g	750	723	737.2	6.715
p4.2.h	821	767	791.0	12.870
p4.2.i	892	825	842.7	14.170
p4.2.j	937	869	898.7	19.310
p4.2.k	979	922	950.5	18.060
p4.2.l	1032	955	993.9	18.390
p4.2.m	1094	1007	1043.0	23.790
p4.2.n	1119	1041	1088.0	18.040
p4.2.o	1173	1082	1127.0	21.140
p4.2.p	1195	1148	1174.0	12.510
p4.2.q	1230	1178	1202.0	12.250
p4.2.r	1249	1199	1226.0	14.390
p4.2.s	1267	1218	1247.0	11.640
p4.2.t	1295	1250	1275.0	8.954
p4.3.c	193	187	188.0	2.243
p4.3.d	335	315	324.3	5.307
p4.3.e	467	438	450.5	8.437
p4.3.f	579	531	562.6	13.110
p4.3.g	653	619	642.3	7.603
p4.3.h	726	698	716.8	6.927
p4.3.i	788	745	771.7	13.010
p4.3.j	839	769	812.6	17.160
p4.3.k	894	775	862.5	28.160
p4.3.l	949	835	899.0	23.470
p4.3.m	1009	905	977.5	25.240
p4.3.n	1078	995	1044.0	22.760
p4.3.o	1145	1063	1110.0	19.740
p4.3.p	1190	1113	1160.0	19.610
p4.3.q	1212	1161	1190.0	11.810
p4.3.r	1244	1210	1223.0	8.990
p4.3.s	1268	1228	1243.0	9.201
p4.3.t	1290	1249	1268.0	9.514
p4.4.e	183	182	182.6	0.486
p4.4.f	324	300	321.5	6.244
p4.4.g	461	440	457.1	6.946
p4.4.h	564	520	543.7	10.980
p4.4.i	654	607	640.6	13.100
p4.4.j	719	676	702.6	12.550
p4.4.k	813	758	793.4	19.630
p4.4.l	872	795	858.6	16.360
p4.4.m	907	870	892.1	9.103
p4.4.n	947	887	927.2	12.520
p4.4.o	1025	975	996.5	14.350
p4.4.p	1090	1047	1072.0	12.810
p4.4.q	1130	1075	1109.0	13.540
p4.4.r	1185	1134	1164.0	13.050
p4.4.s	1232	1171	1205.0	15.670
p4.4.t	1271	1218	1243.0	13.680

Table 2: Test set 5

Instances	Max.	Min.	Avg.	SD
p5.2.h	410	345	378.23	18.51
p5.2.i	480	410	451.94	19.26
p5.2.j	580	550	567.26	11.75
p5.2.k	670	625	653.71	13.41
p5.2.l	795	735	762.26	19.36
p5.2.m	860	810	839.52	13.93
p5.2.n	920	895	910.32	8.56
p5.2.o	1015	945	976.29	16.17
p5.2.p	1135	1050	1087.10	24.96
p5.2.q	1175	1135	1155.81	11.04
p5.2.r	1255	1205	1225.81	11.84
p5.2.s	1315	1275	1288.06	10.06
p5.2.t	1375	1335	1357.74	9.30
p5.2.u	1450	1395	1417.74	12.77
p5.2.v	1490	1450	1474.03	9.17
p5.2.w	1540	1510	1525.32	10.08
p5.2.x	1590	1565	1575.16	6.52
p5.2.y	1625	1605	1613.39	5.23
p5.2.z	1675	1635	1652.90	8.04
p5.3.k	495	470	487.90	9.55
p5.3.l	595	535	569.52	13.62
p5.3.m	650	635	647.58	3.62
p5.3.n	755	705	734.03	15.41
p5.3.o	870	810	857.58	17.31
p5.3.p	990	930	979.68	12.31
p5.3.q	1070	1020	1048.06	12.95
p5.3.r	1120	1090	1099.35	6.02
p5.3.s	1190	1145	1173.39	12.93
p5.3.t	1255	1210	1236.45	9.85
p5.3.u	1330	1285	1314.84	9.96
p5.3.v	1400	1360	1376.61	10.98
p5.3.w	1455	1410	1433.71	9.83
p5.3.x	1520	1485	1500.48	8.00
p5.3.y	1590	1545	1568.55	12.79
p5.3.z	1635	1590	1615.65	11.16
p5.4.m	555	495	539.03	16.70
p5.4.n	620	620	620.00	0.00
p5.4.o	690	655	672.58	9.99
p5.4.p	760	710	746.13	15.31
p5.4.q	860	820	846.29	11.76
p5.4.r	960	875	904.84	17.58
p5.4.s	1025	995	1007.74	8.84
p5.4.t	1160	1095	1133.87	19.65
p5.4.u	1285	1165	1245.97	25.90
p5.4.v	1320	1235	1294.84	22.34
p5.4.w	1370	1345	1354.35	6.29
p5.4.x	1430	1350	1420.00	16.02
p5.4.y	1500	1415	1478.23	15.73
p5.4.z	1595	1520	1556.45	19.76

Table 3: Test set 6

Instances	Max.	Min.	Avg.	SD
p6.2.d	192	192	192.00	0.00
p6.2.e	360	360	360.00	0.00
p6.2.f	588	588	588.00	0.00
p6.2.g	660	660	660.00	0.00
p6.2.h	780	780	780.00	0.00
p6.2.i	888	888	888.00	0.00
p6.2.j	948	936	939.10	3.42
p6.2.k	1032	1020	1028.32	4.29
p6.2.l	1116	1092	1102.06	5.23
p6.2.m	1182	1158	1167.68	7.70
p6.2.n	1260	1230	1246.45	8.05
p6.3.g	282	276	281.23	2.04

Table 3: Continue

Instances	Max.	Min.	Avg.	SD
p6.3.h	444	438	442.45	2.67
p6.3.i	642	618	630.39	7.89
p6.3.j	828	816	824.90	4.06
p6.3.k	894	858	882.39	7.89
p6.3.l	990	960	973.16	7.81
p6.3.m	1074	1038	1059.87	8.13
p6.3.n	1164	1134	1151.42	9.21
p6.4.j	366	366	366.00	0.00
p6.4.k	528	516	521.03	3.83
p6.4.l	696	678	692.52	4.59
p6.4.m	912	882	899.61	9.42
p6.4.n	1068	1050	1064.32	4.82

Table 4: Test set 7

Instances	Max.	Min.	Avg.	SD
p7.2.d	190	179	184.68	5.59
p7.2.e	290	289	289.58	0.50
p7.2.f	387	382	385.87	1.59
p7.2.g	459	456	456.81	1.17
p7.2.h	521	517	517.90	0.98
p7.2.i	579	530	569.16	15.56
p7.2.j	641	597	625.10	9.05
p7.2.k	699	682	691.90	4.16
p7.2.l	758	714	740.84	10.69
p7.2.m	827	776	801.90	13.76
p7.2.n	888	840	866.45	10.75
p7.2.o	939	892	919.19	13.13
p7.2.p	988	938	971.35	13.67
p7.2.q	1029	990	1009.16	12.33
p7.2.r	1078	1041	1058.16	10.71
p7.2.s	1122	1095	1103.71	6.66
p7.2.t	1159	1131	1145.94	7.06
p7.3.h	425	422	424.68	0.79
p7.3.i	487	482	486.16	1.29
p7.3.j	564	547	558.87	4.51
p7.3.k	632	609	627.74	6.23
p7.3.l	684	668	680.81	3.70
p7.3.m	762	739	754.00	5.95
p7.3.n	820	783	799.55	10.26
p7.3.o	874	832	846.39	8.14
p7.3.p	929	908	917.26	3.71
p7.3.q	977	956	965.03	4.49
p7.3.r	1015	996	1005.55	4.86
p7.3.s	1072	1038	1053.71	6.66
p7.3.t	1116	1096	1103.39	3.39
p7.4.g	217	209	212.23	2.60
p7.4.h	285	283	283.97	1.02
p7.4.i	366	366	366.00	0.00
p7.4.j	462	459	461.90	0.54
p7.4.k	518	514	516.90	1.66
p7.4.l	590	575	581.52	4.79
p7.4.m	646	627	641.10	4.99
p7.4.n	726	706	720.29	6.38
p7.4.o	777	744	767.10	8.17
p7.4.p	840	808	824.87	7.33
p7.4.q	899	877	888.52	5.70
p7.4.r	960	931	950.10	6.36
p7.4.s	1006	977	987.87	7.96
p7.4.t	1046	1006	1031.74	7.97

The number of times the best results for each algorithm equals the best known result will be summarized in Table 9. It includes also the average gaps

Table 5: Results test set 4

Instances	Best	SS	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15
p4.2.a	206	206	202	206	206	206	206	206	206	206	206	206	202	206	206	206	206
p4.2.b	341	341	341	341	341	341	341	303	341	341	341	341	341	341	341	341	341
p4.2.c	452	452	438	452	452	452	452	447	452	452	452	452	452	452	452	452	452
p4.2.d	531	531	517	530	531	531	531	526	531	531	530	531	528	531	531	531	527
p4.2.e	618	618	593	618	613	618	618	602	618	600	600	613	593	618	612	618	600
p4.2.f	687	678	666	687	676	684	687	651	687	672	672	672	675	687	687	687	685
p4.2.g	757	750	749	751	756	750	753	734	757	756	756	756	750	757	757	757	737
p4.2.h	835	821	827	795	820	827	835	797	827	819	819	820	819	835	835	835	820
p4.2.i	918	892	915	882	899	916	918	826	918	900	918	918	916	918	918	918	901
p4.2.j	965	937	914	946	962	962	962	939	965	962	962	962	962	964	962	965	953
p4.2.k	1022	979	963	1013	1013	1019	1022	994	1022	1016	1016	1016	1007	1022	1013	1022	1009
p4.2.l	1074	1032	1022	1061	1058	1073	1074	1051	1071	1070	1071	1069	1051	1071	1064	1074	1058
p4.2.m	1132	1094	1089	1106	1098	1132	1132	1051	1130	1115	1119	1113	1051	1132	1130	1132	1095
p4.2.n	1174	1119	1150	1169	1171	1159	1171	1117	1168	1149	1158	1169	1124	1174	1161	1173	1118
p4.2.o	1218	1173	1175	1180	1192	1216	1218	1191	1215	1209	1198	1210	1195	1217	1206	1218	1184
p4.2.p	1242	1195	1208	1226	1239	1239	1241	1214	1242	1229	1233	1239	1237	1242	1240	1242	1200
p4.2.q	1267	1230	1255	1252	1255	1265	1263	1248	1263	1253	1252	1260	1239	1267	1257	1263	1237
p4.2.r	1292	1249	1277	1281	1283	1283	1286	1267	1288	1278	1278	1279	1279	1292	1278	1286	1251
p4.2.s	1304	1267	1294	1296	1299	1300	1301	1286	1304	1304	1303	1304	1295	1304	1293	1296	1275
p4.2.t	1306	1295	1306	1306	1306	1306	1306	1294	1306	1306	1306	1306	1305	1306	1299	1306	1293
p4.3.c	193	193	192	193	193	193	193	193	193	193	193	193	193	193	193	193	191
p4.3.d	335	335	333	335	335	335	335	335	335	333	333	335	331	335	333	335	333
p4.3.e	468	467	465	468	468	468	468	444	468	468	468	468	460	468	468	468	458
p4.3.f	579	579	579	579	579	579	579	564	579	579	579	579	556	579	579	579	560
p4.3.g	653	653	646	651	652	653	653	644	653	652	653	652	651	653	653	653	646
p4.3.h	729	726	709	722	727	724	729	706	720	713	713	713	718	725	725	729	718
p4.3.i	809	788	785	806	806	806	807	806	796	793	793	786	807	809	797	809	795
p4.3.j	861	839	860	858	858	861	861	826	861	857	855	858	854	861	858	861	845
p4.3.k	919	894	906	919	918	919	919	864	918	913	910	910	902	919	918	918	896
p4.3.l	979	949	951	976	973	975	978	960	979	958	976	966	969	974	968	979	955
p4.3.m	1063	1009	1005	1034	1049	1056	1063	1030	1053	1039	1028	1046	1047	1063	1043	1063	1015
p4.3.n	1121	1078	1119	1108	1115	1111	1121	1113	1121	1109	1112	1103	1106	1121	1108	1120	1077
p4.3.o	1172	1145	1151	1156	1157	1172	1170	1121	1170	1163	1167	1165	1136	1172	1165	1170	1147
p4.3.p	1222	1190	1218	1207	1221	1208	1222	1190	1221	1202	1207	1207	1200	1222	1209	1220	1190
p4.3.q	1253	1212	1249	1237	1241	1250	1251	1210	1252	123	1239	1238	1236	1252	1246	1253	1234
p4.3.r	1273	1244	1265	1224	1269	1272	1272	1239	1267	1263	1263	1263	1250	1273	1257	1272	1264
p4.3.s	1295	1268	1282	1250	1294	1289	1293	1279	1293	1291	1289	1291	1280	1295	1276	1287	1275
p4.3.t	1305	1290	1288	1303	1304	1298	1304	1290	1305	1304	1303	1304	1299	1304	1294	1299	1291
p4.4.e	183	183	182	183	183	183	183	183	183	183	183	183	183	183	183	183	182
p4.4.f	324	324	315	324	324	324	324	312	324	324	324	324	324	324	324	324	324
p4.4.g	461	461	453	461	461	461	461	461	461	461	460	460	461	461	461	461	460
p4.4.h	571	564	554	571	571	571	571	565	571	556	556	556	571	571	571	571	555
p4.4.i	657	654	627	655	657	657	657	657	657	653	653	653	653	657	653	657	641
p4.4.j	732	719	732	731	731	732	732	691	732	731	731	731	732	732	732	732	724
p4.4.k	821	813	819	821	816	821	821	815	821	820	818	818	820	821	820	821	804
p4.4.l	880	872	875	878	878	879	880	852	880	877	875	875	875	879	875	879	843
p4.4.m	919	907	910	916	918	916	919	910	918	911	911	911	914	918	914	919	898
p4.4.n	977	947	977	972	976	968	968	942	961	956	956	956	953	969	953	969	957
p4.4.o	1061	1025	1014	1057	1057	1051	1061	937	1036	1030	1029	1029	1033	1061	1033	1057	998
p4.4.p	1124	1090	1056	1120	1120	1120	1120	1091	1111	1108	1110	1110	1098	1124	1098	1122	1086
p4.4.q	1161	1130	1124	1148	1157	1160	1161	1106	1145	1150	1148	1148	1149	1161	1139	1160	1131
p4.4.r	1216	1185	1165	1203	1211	1207	1203	1148	1200	1195	1194	1194	1193	1216	1196	1213	1156
p4.4.s	1260	1232	1243	1245	1256	1259	1255	1242	1249	1256	1252	1252	1213	1259	1231	1250	1230
p4.4.t	1285	1271	1255	1279	1285	1282	1279	1250	1281	1281	1281	1281	1281	1284	1256	1280	1253

Table 6: Results test set 5

Instances	Best	SS	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15
p5.2.h	410	410	410	410	410	410	410	385	410	410	410	410	395	410	410	410	410
p5.2.j	580	580	560	580	580	580	580	580	580	580	580	580	580	580	580	580	580
p5.2.k	670	670	670	670	670	670	670	665	670	670	670	670	670	670	670	670	670
p5.2.l	800	795	770	800	800	800	800	760	800	800	800	800	770	800	800	800	770
p5.2.m	860	860	860	860	860	860	860	830	860	860	860	860	860	860	860	860	855
p5.2.n	925	920	920	925	925	925	925	920	925	920	920	925	920	925	925	925	920
p5.2.o	1020	1015	975	1020	1020	1020	1020	1010	1020	1020	1010	1010	1020	1020	1020	1020	1010
p5.2.p	1150	1135	1090	1130	1150	1150	1150	1030	1150	1150	1150	1150	1150	1150	1150	1150	1140
p5.2.q	1195	1175	1185	1195	1195	1195	1195	1145	1195	1195	1195	1195	1195	1195	1195	1195	1190
p5.2.r	1260	1255	1260	1260	1260	1260	1260	1225	1260	1260	1260	1260	1260	1260	1260	1260	1255
p5.2.s	1340	1315	1310	1330	1340	1340	1340	1325	1340	1330	1330	1330	1325	1330	1340	1340	1315
p5.2.t	1400	1375	1380	1380	1400	1400	1400	1360	1400	1400	1400	1400	1380	1400	1400	1400	1380
p5.2.u	1460	1450	1445	1440	1460	1460	1460	1460	1460	1460	1460	1460	1450	1460	1460	1460	1450

Table 6: Continue

Instances	Best	SS	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15
p5.2.v	1505	1490	1500	1490	1505	1500	1505	1500	1505	1495	1500	1495	1500	1505	1505	1505	1495
p5.2.w	1565	1540	1560	1555	1565	1560	1560	1560	1560	1555	1555	1555	1560	1560	1560	1560	1560
p5.2.x	1610	1590	1610	1595	1610	1590	1610	1610	1610	1610	1610	1610	1600	1610	1610	1610	1580
p5.2.y	1645	1625	1630	1635	1635	1635	1635	1630	1645	1645	1645	1645	1630	1645	1645	1645	1645
p5.2.z	1680	1675	1680	1670	1680	1670	1670	1680	1680	1680	1680	1680	1665	1680	1670	1680	1660
p5.3.k	495	495	495	495	495	495	495	470	495	495	495	495	495	495	495	495	495
p5.3.l	595	595	575	595	595	595	595	545	595	595	595	595	595	595	595	595	595
p5.3.n	755	755	755	755	755	755	755	720	755	755	755	755	755	755	755	755	755
p5.3.o	870	870	835	870	870	870	870	870	870	870	870	870	870	870	870	870	870
p5.3.q	1070	1070	1065	1070	1070	1070	1070	1045	1070	1065	1065	1065	1065	1070	1070	1070	1070
p5.3.r	1125	1120	1115	1110	1125	1125	1125	1090	1125	1120	1125	1125	1125	1125	1125	1125	1115
p5.3.s	1190	1190	1175	1185	1190	1190	1190	1145	1190	1190	1190	1185	1185	1190	1185	1190	1175
p5.3.t	1260	1255	1240	1250	1260	1260	1260	1240	1260	1250	1255	1260	1260	1260	1260	1260	1260
p5.3.u	1345	1330	1330	1340	1345	1345	1345	1305	1345	1330	1335	1335	1345	1345	1335	1345	1340
p5.3.v	1425	1400	1410	1420	1425	1425	1425	1425	1425	1425	1425	1420	1425	1425	1420	1425	1405
p5.3.w	1485	1455	1465	1485	1485	1485	1485	1460	1485	1465	1465	1465	1475	1485	1465	1485	1455
p5.3.x	1555	1520	1530	1555	1555	1555	1555	1520	1540	1535	1540	1540	1535	1555	1540	1550	1515
p5.3.y	1595	1590	1580	1590	1595	1595	1595	1590	1590	1590	1590	1590	1580	1590	1590	1590	1540
p5.3.z	1635	1635	1635	1625	1635	1635	1635	1635	1635	1635	1635	1635	1635	1635	1635	1635	1620
p5.4.m	555	555	555	555	555	555	555	550	555	555	555	555	550	555	555	555	555
p5.4.o	690	690	680	690	690	690	690	680	690	690	690	690	690	690	690	690	690
p5.4.p	765	760	760	765	765	765	765	760	765	760	760	760	760	760	760	760	765
p5.4.q	860	860	860	860	860	860	860	830	860	860	860	860	835	860	860	860	860
p5.4.r	960	960	960	960	960	960	960	890	960	960	960	960	960	960	960	960	960
p5.4.s	1030	1025	1000	1025	1030	1030	1030	1020	1030	1030	1030	1030	1020	1030	1005	1025	1030
p5.4.t	1160	1160	1100	1160	1160	1160	1160	1160	1160	1160	1160	1160	1160	1160	1160	1160	1160
p5.4.u	1300	1285	1275	1300	1300	1300	1300	1300	1300	1300	1300	1300	1300	1300	1300	1300	1300
p5.4.v	1320	1320	1310	1320	1320	1320	1320	1245	1320	1320	1320	1320	1320	1320	1320	1320	1290
p5.4.w	1390	1370	1380	1375	1390	1390	1390	1330	1390	1380	1390	1380	1380	1380	1380	1390	1385
p5.4.x	1450	1430	1410	1440	1450	1450	1450	1410	1450	1450	1450	1450	1440	1450	1430	1450	1430
p5.4.y	1520	1500	1520	1520	1520	1520	1520	1485	1520	1510	1510	1500	1500	1520	1520	1520	1520
p5.4.z	1620	1595	1575	1620	1620	1620	1620	1590	1620	1620	1575	1580	1600	1620	1620	1620	1575

Table 7: Results test set 6

Instances	Best	SS	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15
p6.2.d	192	192	192	192	192	192	192	180	192	192	192	192	192	192	192	192	192
p6.2.j	948	948	936	948	948	948	948	948	948	948	948	948	948	948	942	948	948
p6.2.l	1116	1116	1116	1098	1116	1116	1116	1104	1116	1110	1116	1116	1116	1116	1110	1116	1116
p6.2.m	1188	1182	1188	1164	1188	1188	1188	1164	1188	1188	1188	1188	1188	1188	1188	1188	1188
p6.2.n	1260	1260	1260	1242	1260	1260	1260	1254	1260	1260	1254	1260	1248	1260	1260	1260	1242
p6.3.g	282	282	282	282	282	282	282	264	282	282	282	282	276	282	282	282	282
p6.3.h	444	444	444	444	444	444	444	444	444	444	438	438	444	444	444	444	444
p6.3.i	642	642	612	642	642	642	642	642	642	642	642	642	642	642	642	642	636
p6.3.k	894	894	876	894	894	894	894	882	894	888	888	894	894	894	894	894	894
p6.3.l	1002	990	990	1002	1002	1002	1002	990	1002	1002	1002	1002	996	1002	1002	1002	1002
p6.3.m	1080	1074	1080	1080	1080	1080	1080	1068	1080	1074	1080	1080	1080	1080	1080	1080	1080
p6.3.n	1170	1164	1152	1170	1170	1170	1170	1140	1170	1164	1164	1164	1152	1170	1164	1170	1158
p6.4.j	366	366	366	366	366	366	366	360	366	366	366	366	366	366	366	366	366
p6.4.k	528	528	522	528	528	528	528	528	528	528	528	528	528	528	528	528	522
p6.4.l	696	696	696	696	696	696	696	678	696	696	696	696	678	696	696	696	696

to the best known result and the average CPU time in seconds. Table 10 includes the execution time needed for each algorithm to solve each data set.

The basic scatter search is capable of gaining 60 times of best known solution and achieves solutions that depart on average 0.69% from the best known solution within 15.08 sec. Finally, Table 11 expresses the number of results that were achieved better, equals and worse using scatter search compared to the algorithms in the earlier studies. Based on this comparison, the quality of results A2, A4 and A5 proposed by Archetti *et al*. (2007); A7, A8, A9 and A10 proposed by Ke *et al*. (2008); A13 and A14 proposed by Souffriau *et al*. (2010) and A11 proposed by Bouly *et al*. (2010) is better than the SS. On the other hand, the average execution time of scatter search is 15.08 sec which appears to be much better.

The contribution of this study is to show that basic scatter search is able to tackle TOP within competitive

Table 8: Results test set 7

Instances	Best	SS	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15
p7.2.d	190	190	190	190	190	190	190	190	190	190	190	190	182	190	190	190	190
p7.2.e	290	290	290	290	290	289	290	279	290	290	290	290	289	290	290	290	290
p7.2.f	387	387	382	387	387	387	387	340	387	387	387	387	387	387	387	387	387
p7.2.g	459	459	459	456	459	459	459	440	459	459	459	459	457	459	459	459	459
p7.2.h	521	521	521	520	520	521	521	517	521	521	521	521	521	521	521	521	521
p7.2.i	580	579	578	579	579	575	579	568	580	579	579	579	579	579	578	580	578
p7.2.j	646	641	638	643	644	643	644	633	646	646	646	646	632	646	646	646	623
p7.2.k	705	699	702	702	705	704	705	691	705	704	704	704	700	704	702	705	698
p7.2.l	767	758	767	758	767	759	767	748	767	767	767	767	758	767	759	767	755
p7.2.m	827	827	817	827	824	824	827	798	827	827	827	827	827	827	816	827	807
p7.2.n	888	888	864	884	888	883	888	861	888	878	878	878	866	888	888	888	868
p7.2.o	945	939	914	933	945	945	945	897	945	945	940	941	928	945	932	945	908
p7.2.p	1002	988	987	1000	1002	1002	1002	954	1002	991	993	993	955	1002	993	1002	970
p7.2.q	1044	1029	1017	1041	1043	1038	1044	1031	1043	1042	1043	1043	1029	1044	1043	1044	1020
p7.2.r	1094	1078	1067	1091	1088	1094	1094	1075	1094	1093	1088	1094	1069	1094	1076	1094	1066
p7.2.s	1136	1122	1116	1123	1128	1136	1136	1102	1136	1136	1134	1131	1118	1136	1125	1136	1097
p7.2.t	1179	1159	1165	1172	1174	1168	1179	1142	1179	1179	1179	1179	1154	1179	1168	1175	1129
p7.3.f	247	247	247	247	247	247	247	247	247	247	247	247	247	247	247	247	240
p7.3.h	425	425	416	425	425	425	425	418	425	425	425	425	425	425	425	425	419
p7.3.i	487	487	481	487	487	487	487	480	487	487	486	487	480	487	485	487	485
p7.3.j	564	564	563	564	564	562	564	539	564	564	564	564	543	563	560	564	557
p7.3.k	633	632	632	633	633	632	633	586	633	632	633	633	633	633	633	633	633
p7.3.l	684	684	681	683	679	681	681	668	684	683	684	684	681	683	684	684	681
p7.3.m	762	762	756	749	755	745	762	735	762	762	762	762	743	762	762	762	754
p7.3.n	820	820	789	810	811	814	820	789	820	819	819	820	804	820	813	820	813
p7.3.o	874	874	874	873	865	871	874	833	874	874	874	874	841	874	859	874	848
p7.3.p	929	929	922	917	923	926	927	912	929	925	926	925	918	927	925	927	919
p7.3.q	987	977	966	976	987	978	987	945	987	987	987	987	966	987	970	987	960
p7.3.r	1026	1015	1011	1018	1022	1024	1022	1015	1026	1024	1021	1022	1009	1024	1017	1021	1017
p7.3.s	1081	1072	1061	1081	1081	1079	1079	1054	1081	1081	1081	1077	1070	1081	1076	1081	1064
p7.3.t	1120	1116	1098	1114	1116	1112	1115	1080	1118	1117	1103	1117	1109	1120	1111	1118	1093
p7.4.g	217	217	217	217	217	217	217	209	217	217	217	217	217	217	217	217	211
p7.4.h	285	285	285	285	285	285	285	285	285	285	285	285	283	285	285	285	283
p7.4.i	366	366	359	366	366	366	366	359	366	366	366	366	364	366	366	366	349
p7.4.k	520	518	503	520	520	518	520	511	520	520	520	520	518	518	518	518	516
p7.4.l	590	590	576	590	588	588	590	573	590	590	590	590	575	590	581	590	578
p7.4.m	646	646	643	644	646	646	646	638	646	644	646	646	639	646	646	646	633
p7.4.n	730	726	726	723	721	715	730	698	730	725	725	726	723	726	723	730	712
p7.4.o	781	777	776	772	778	770	781	761	781	778	781	778	778	779	780	780	776
p7.4.p	846	840	832	841	839	846	846	803	846	846	838	842	841	846	842	846	819
p7.4.q	909	899	905	902	898	899	906	899	909	909	909	909	896	907	902	907	890
p7.4.r	970	960	966	970	969	970	970	937	970	970	970	970	964	970	961	970	948
p7.4.s	1022	1006	1019	1021	1020	1021	1022	1005	1022	1019	1021	1019	1019	1022	1022	1022	988
p7.4.t	1077	1046	1067	1071	1071	1077	1077	1020	1077	1072	1077	1077	1073	1077	1066	1077	1066

Table 9: Summary of results

Algorithm	# best	Avg. gap (%)	Avg. CPU (s)
SS	61	0.69	15.08
A1	36	1.03	336.58
A2	70	0.40	100.00
A3	95	0.17	146.61
A4	95	0.15	18.94
A5	128	0.04	268.64
A6	22	1.93	7.83
A7	129	0.08	252.28
A8	81	0.70	213.80
A9	81	0.31	204.75
A10	85	0.28	214.98
A10	46	0.75	3.78
A12	131	0.03	63.76
A13	79	0.33	4.98
A14	127	0.04	212.43
A15	40	1.01	-

Table 10: Execution time (sec)

Algorithm	Set 4	Set 5	Set 6	Set 7	Avg.
SS	13.89	15.79	12.36	18.29	15.08
A1	796.70	71.30	45.70	432.60	336.58
A2	105.29	69.45	66.29	158.97	100.00
A3	282.92	26.55	20.11	256.76	146.61
A4	22.52	34.17	8.74	10.34	18.94
A5	457.89	158.93	147.88	309.87	268.64
A6	11.40	3.50	4.30	12.10	7.83
A7	370.90	173.60	161.10	303.50	252.28
A8	317.90	150.60	140.80	245.90	213.80
A9	307.40	143.30	135.20	233.10	204.75
A10	320.40	151.30	141.70	246.50	214.98
A11	7.40	1.50	1.90	4.30	3.78
A12	125.26	23.96	15.53	90.30	63.76
A13	8.60	2.90	2.10	6.30	4.98
A14	367.40	119.90	89.60	272.80	212.43
A15	-	-	-	-	-

Table 11: SS compared to literature

Algorithm	Better	Equal	Worse
A1	80	32	46
A2	31	51	76
A3	15	53	90
A4	16	56	86
A5	4	60	94
A6	103	25	30
A7	1	62	95
A8	19	59	80
A9	22	55	81
A10	16	59	83
A11	54	41	63
A12	4	63	91
A13	24	54	80
A14	1	64	93
A15	77	37	44

time. Scatter search has a reference set component that contain a set of elite solution and a set of divers solution which were combined in systematic way to reduce the probability of getting stuck in local optimas.

CONCLUSION

This study presents scatter search algorithm to tackle TOP. Applying scatter search to solve TOP is obviously a very promising method. Scatter search can achieves good solutions during small amount of calculation time that depart on average 0.69% from the best known solution within 15.08 sec average time. The proposed algorithm is compared with several promising algorithms from the literature. The results prove that this algorithm can rival any other popular algorithms. The advantages of the proposed algorithm are efficient, effective and simple. Scatter search algorithm can be hybridized with other local search to increase intensification search in each region to get the best solution in the neighborhood and to reduce execution time.

ACKNOWLEDGEMENT

The researchers wish to thank Ministry of Higher Education for supporting this research under the FRGS Research Grant Scheme (FRGS/1/2012/SG05/UKM/02/11).

REFERENCES

Archetti, C., A. Hertz and M.G. Speranza, 2007. Metaheuristics for the team orienteering problem. J. Heurist., 13: 49-76.

Bouly, H., D.C. Dang and A. Moukrim, 2010. A memetic algorithm for the team orienteering problem. 4OR, 8: 49-70.

Burke, E.K., T. Curtois, R. Qu and G.V. Berghe, 2010. A scatter search methodology for the nurse rostering problem. J. Oper. Res. Soc., 61: 1667-1679.

Butt, S.E. and T.M. Cavalier, 1994. A heuristic for the multiple tour maximum collection problem. Comput. Oper. Res., 21: 101-111.

Chao, I., B.L. Golden and E.A. Wasil, 1996. A fast and effective heuristic for the orienteering problem. Eur. J. Oper. Res., 88: 475-489.

Engin, O., C. Kahraman and M.K. Yilmaz, 2009. A scatter search method for multiobjective fuzzy permutation flow shop scheduling problem: A real world application. Comput. Intel. Flow Shop Job Shop Scheduling, 230: 169-189.

Fischetti, M., J.J.S. Gonzalez and P. Toth, 1998. Solving the orienteering problem through branch-and-cut. INFORMS J. Comput., 10: 133-148.

Gendreau, M., G. Laporte and F. Semet, 1998. A tabu search heuristic for the undirected selective travelling salesman problem. Eur. J. Oper. Res., 106: 539-545.

Glover, F., 1977. Heuristics for integer programming using surrogate constraints. Decis. Sci., 8: 156-166.

Glover, F., 1998. A template for scatter search and path relinking. http://leeds-faculty.colorado.edu/glover/SS-PR%20Template.pdf.

Glover, F., 1999. Scatter Search and Path Relinking. In: New Ideas in Optimization, Corne, D., M. Dorigo and F. Glover (Eds.). McGraw-Hill Ltd., UK., pp: 297-316.

Glover, F., M. Laguna and R. Marti, 2003. Scatter Search. Springer-Verlag, USA., pp: 519-537.

Golden, B.L., Q. Wang and L. Liu, 1988. A multifaceted heuristic for the orienteering problem. Naval Res. Logistics, 35: 359-366.

Jaradat, G.M. and M. Ayob, 2011. Scatter search for solving the course timetabling problem. Proceedings of the 3rd Conference on Data Mining and Optimization (DMO), June, 28-29, 2011, Malaysia, pp: 213-218.

Ke, L., C. Archetti and Z. Feng, 2008. Ants can solve the team orienteering problem. Comput. Ind. Eng., 54: 648-665.

Laguna, M. and V. Armentano, 2005. Lessons from Applying and Experimenting with Scatter Search. In: Metaheuristic Optimization via Memory and Evolution, Sharda, R., S. Vob, C. Rego and B. Alidaee (Eds.). Vol. 30, Springer, USA., pp: 229-246.

Laguna, M., R. Marti and R.C. Marti, 2003. Scatter Search: Methodology and Implementations. Springer, Netherlands.

Liu, Y.H., 2008. Diversified local search strategy under scatter search framework for the probabilistic traveling salesman problem. Eur. J. Oper. Res., 191: 332-346.

Maquera, G., M. Laguna, D.A. Gandelman and A.P. Sant'Anna, 2011. Scatter search applied to the vehicle routing problem with simultaneous delivery and pickup. Int. J. Applied Metaheuristic Comput., 2: 1-20.

Miller, C., A.W. Tucker and R. Zemlin, 1960. Integer programming formulation of traveling salesman problems. J. ACM, 7: 326-329.

Muthuswamy, S. and S.S. Lam, 2011. Discrete particle swarm optimization for the team orienteering problem. Memetic Comput., 3: 287-303.

Ramesh, R., Y.S. Yoon and M.H. Karwan, 1992. An optimal algorithm for the orienteering tour problem. ORSA J. Comput., 4: 155-165.

Resende, M.G.C., C.C. Ribeiro, F. Glover and R. Marti, 2010. Scatter Search and Path-Relinking: Fundamentals, Advances and Applications. In: Handbook of Metaheuristics, Gendreau, M. and J.Y. Potvin (Eds.). Springer, USA., pp: 87-107.

Sabar, N.R. and M. Ayob, 2009. Examination timetabling using scatter search hyper-heuristic. Proceedings of the 2nd Conference on Data Mining and Optimization, October 27-28, 2009, Kajand, pp: 127-131.

Souffriau, W., P. Vansteenwegen, G.V. Berghe and D. van Oudheusden, 2010. A path relinking approach for the team orienteering problem. Comput. Oper. Res., 37: 1853-1859.

Tang, H. and E. Miller-Hooks, 2005. A tabu search heuristic for the team orienteering problem. Comput. Oper. Res., 32: 1379-1407.

Vansteenwegen, P., W. Souffriau, G.V. Berghe and D.V. Oudheusden, 2009a. Metaheuristics for tourist trip planning. Metaheuristics Serv. Ind., 624: 15-31.

Vansteenwegen, P., W. Souffriau, G.V. Berghe and D.V. Oudheusden, 2009b. A guided local search metaheuristic for the team orienteering problem. Eur. J. Oper. Res., 196: 118-127.