Journal of Computer Science

Original Research Paper

A Randomized Population Constructive Heuristic for the Team Orienteering Problem

Manal Zettam

Informatics, Systems and Optimization Laboratory,
Department of computer Science, Faculty of Science, Ibn Tofail University Kenitra, Morocco

Article history
Received: 18-12-2016
Revised: 12-06-2017
Accepted: 16-06-2017

Email: zettammanal@gmail.com

Abstract: The NP-hard (complete) team orienteering problem is a particular vehicle routing problem with the aim of maximizing the profits gained from visiting control points without exceeding a travel cost limit. The team orienteering problem has a number of applications in several fields such as athlete recruiting, technician routing and tourist trip. Therefore, solving optimally the team orienteering problem would play a major role in logistic management. In this study, a novel randomized population constructive heuristic is introduced. This heuristic constructs a diversified initial population for population-based metaheuristics. The heuristics proved its efficiency. Indeed, experiments conducted on the well-known benchmarks of the team orienteering problem show that the initial population constructed by the presented heuristic wraps the best-known solution for 131 benchmarks and good solutions for a great number of benchmarks.

Keywords: Orienteering Problem, Team Orienteering Problem, Population-Based Meta-Heuristic

Introduction

The Orienteering Problem (OP) was first introduced by (Golden *et al*., 1987). It originates from orienteering sport. According to (Zettam and Elbenani, 2016) the Orienteering sport is defined as an outdoor sport, played in heavily forested and mountainous areas. A set of control points are located in the forest. Each control point is associated with a score. Assuming that the start control point and the end control point are fixed. Competitors equipped with compass and map are required to visit a subset of control points starting from the start control point and ending at the end control point with the aim of maximizing their total score within a predefined amount of time. Several variants of OP has been described in literature such as The Team Orienteering Problem, the Orienteering Problem with Time Windows, the team orienteering problem with time windows and the time-dependent orienteering Problem. OP and its variant were given a great interest as results of their applications (Vansteenwegen *et al*., 2011). Selective travelling salesperson problem (Tsiligirides, 1984), home fuel delivery problem (Golden *et al*., 1987), single-ring design problem (Thomadsen and Stidsen, 2003) and mobile tourist guide problem (Souffriau *et al*., 2008) are some of well known application of OP and its variants. For more details readers are referred to (Vansteenwegen *et al*., 2011).

The team orienteering problem TOP extends OP. It was first introduced by (Chao *et al*., 1996). Competitors are subdivided into teams. The competitors of a given team collaborate to maximize the score within a predefined amount of time or distance limit. Each control point is visited once by a member of a given team. Even though two exact methods have been proposed to solve the team orienteering problem, it still be considered as an NP-hard (complete) problem according to (Chao *et al*., 1996). Therefore, a number of heuristics and meta-heuristics have been developed to solve TOP. Lin (2013) proposed a multi-start simulated annealing algorithm that combines the simulated annealing algorithm and the multi-start hill climbing algorithm. Butt and Cavalier (1994) presented a mixed integer programming solved by a greedy method that adds the best pair of vertices to the solution tours at each iteration. They claims that their approach do well with relatively small number of control points within a tour. In the approach proposed by (Chao *et al*., 1996), the initial solution is constructed by inserting Control points into paths using the cheapest insertion rule. If unsigned control points remain new paths are constructed. The constructed initial solution is improved by using 1-point movement, 2-point exchange and 2-opt operator with record-to-record framework. Tang and Miller-Hook (2005) proposed a Tabu search embedded with adoptive memory. This method generates better solution with more computational time compared to (Chao *et al*., 1996) for a number of instances. (Archetti *et al*., 2007) addressed a variable neighborhood approach which outperforms (Tang and Miller-Hooks, 2005; Chao *et al*., 1996) approaches. Ke *et al*. (2008) proposed an Ant Colony Approach that embedded one of the three following methods: Sequential deterministic-concurrent method, random-concurrent method and simultaneous method. The performance of the Ant Colony Approach with the sequential method is comparable to (Archetti *et al*., 2007). Vansteenwegen *et al*. (2009) introduced the disturb method which combines a guided local search method and a diversifying mechanism. The disturb method find the best-known solutions in shorter computational time compared to the existing methods. (Bouly *et al*., 2009) proposed a PSO-inspired algorithm which updated the best-known solution for one instance use this style when you need to begin a new paragraph.

In this study, we propose a randomized population constructive heuristic for the Team orienteering problem. This heuristic constructs a diversified initial population for population-based metaheuristics. It constructs solutions based on randomly generated permutations of control points. The solutions within the initial population are diversified with different costs and tour lengths. A number of permutations are randomly generated. For each permutation of control points a given number of path beginning from the starting control point and ending at the arrival control point are constructed. The best solution of the population is then enhanced by a local search.

The rest of this paper is organized as follows. The first section addresses the team orienteering problem. The first section also introduces the mathematical model adopted in this study. The second section details the proposed heuristic. The fourth section involves the computational results followed by concluding remarks and proposal for future works in the fifth section.

Mathematical Formulation

Given a set of n control points and m competitors. The control points are usually named locations and competitors are usually named tours in the literature of TOP. The main goal of TOP is to construct m tours starting from the departure (location 1) and ending at the arrival (location n) that maximize the total score denoted s. A travel time or length of a tour cannot exceed $T_{\max}$. In the present paper, we employ the Euclidean distance to calculate the length tours. Table 1 shows an example of TOP instances where $T_{\max} = 20$, $m = 2$. Table 1 contains locations, XY coordinates and scores. The solution of the instance shown in Table 1 is represented in Fig. 1.

In this study, a mathematical model similar to the one used by Lin (2013) is described as follows:

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

Subject to:

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

$$\sum_{i=1}^{n-1} x_{ilk} = \sum_{j=2}^{n} x_{ljk} = y_{lp},\ \forall l = 2,...,n-1; \forall k = 1,...,m \quad (3)$$

$$\sum_{k=1}^{m} y_{ik} \le 1,\ \forall i = 2,...,n-1 \quad (4)$$

$$\sum_{i=1}^{n-1}\sum_{j=2}^{n} t_{ij} x_{ijk} \le T_{\max},\ \forall k = 1,...,m \quad (5)$$

$$\sum_{\substack{i,j\in U \\ i\prec j}} x_{ijk} \le |U| - 1 \quad (\mathrm{U} \subset \mathrm{V} \setminus \{1,n\} : 2 \le |U| \le \mathrm{n}-2;\ k = 1,...,m) \quad (6)$$

$$x_{ijk}, y_{ik} \in \{0,1\},\ \forall \mathrm{i},\mathrm{j} = 1,...,\mathrm{n};\ \forall k = 1,...,m \quad (7)$$

Where:

S_i	=	Is the score associated to the ith location
t_{ij}	=	The length of the path starting at the ith location and ending at the jth location. The travel length cannot exceed $T_{\max}$
V	=	Is the set of locations
U	=	Is a subset of V
$x_{ijm} = 1$	=	if, in tour m, a visit to location is followed by a visit to location, 0 otherwise
$y_{ik} = 1$	=	if location is i visited on tour k, 0 otherwise
s_{ik}	=	The start time of the service at location i on tour k

The objective function to maximize is represented by Equation 1. The constraint that all tours starts from location 1 and ends at location n is ensured by Equation 2. The connectivity of tours is maintained by Equation 3. Constraint (4) guarantees that every location is visited at most once. Constraint (5) guarantees that length limitation constraint is not violated for each tour. Constraint (6) excludes sub-tours. Constraint (7) states that the variables x and y are binary.

Table 1. An example of TOP instances

Location	X	Y	S
1	10.500	14.400	0
2	18.000	15.900	10
3	18.300	13.300	10
4	16.500	9.300	10
5	15.400	11.000	10
6	14.900	13.200	5
7	16.300	13.300	5
8	16.400	17.800	5
9	15.000	17.900	5
10	16.100	19.600	10
11	15.700	20.600	10
12	13.200	20.100	10
13	14.300	15.300	5
14	14.000	5.100	10
15	11.400	6.700	15
16	8.300	5.000	15
17	7.900	9.800	10
18	11.400	12.000	5
19	11.200	17.600	5
20	10.100	18.700	5
21	11.700	20.300	10
22	10.200	22.100	10
23	9.700	23.800	10
24	10.100	26.400	15
25	7.400	24.000	15
26	8.200	19.900	15
27	8.700	17.700	10
28	8.900	13.600	10
29	5.600	11.100	10
30	4.900	18.900	10
31	7.300	18.800	10
32	11.200	14.100	0

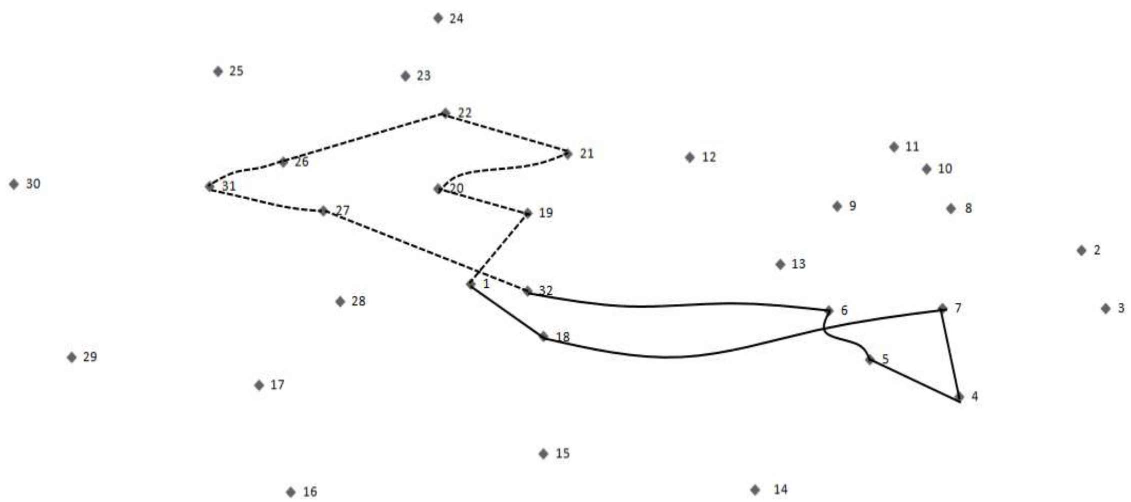

Fig. 1. A solution example

The Novel Randomized Heuristic

The heuristic randomly generates a predefined number of permutations. Solutions are then constructed on the basis of the generated permutations. Each solution is represented by m connected tours. Given a permutation of n elements. Each element represents a location. For each tour k, the heuristic seeks a permutation. If the ith location is not contained in other tours and if adding the ith location does not violate fifth constraint described in the mathematical model, the ith location is added to the tour kth. Otherwise, the heuristic goes to the next iteration and so forth. Finally the best solution of the population is enhanced via a local search. The local search used in this study employs the swap and the add/drop operators. The solution is enhanced 10 times with a neighbourhood size equal to the half of the number of customer in the first tour. Our heuristic is describes in details in the algorithm (1) below. *dist*(i,j) denotes the Euclidean distance between two location i and j.

```
Algorithm (1): The novel randomize heuristic
Inputs: The number of tours denoted m,
        The maximum length of a tour T_max,
        The set of locations V = {1,2,...,n},
        The size of the population l.
Start
        Generate randomly l permutations
        for each permutation do
            for k: = 1 to m do
            c: 0
               for each i∈V do
                  if (x_ij + c ≤ T_max) then
                     c: = c + dist(i,j)
                     j = i
                  end if
               end for
        end for
      end for
Apply the local search to enhance the best solution.
Output: population of feasible solutions
```

The randomized heuristic reaches best-known solutions for 131 of benchmarks in a minor computational time. The randomized heuristic also generates average solutions for the majority of the reminder benchmarks. A good initial population would allow population based metaheuristics to find good solutions in an optimistic computational time with less computational efforts.

Results and Discussion

The proposed heuristic was implemented using JAVA language and was run on a PC with an Intel Core i5-2540 M 2.60 GHz processor. To evaluate the performance of the proposed heuristic, each benchmark of TOP was tested over ten trial and compared to the best-known solution. The benchmarks we used in this study are available at (https://www.hds.utc.fr/~moukrim). Table 2 summarizes the characteristics of problem benchmarks sets. The first number of a problem indicates the set number, e.g., p2.2, p2.3 and p2.4 belong to set 2. The coordinates and score of each location are identical for the instances belonging to a given problem set. The second number in the problem set indicates the number of tours, e.g., p2.3 means that there are 3 tours.

We ran our heuristic 20 times on each instance in order to select the best obtained solution. Table 3 to 23 contain the problem nomination, the best-known solution in the literature and also the best solution in the population we obtained by applying the proposed randomized heuristic. The third and fifth column of each table contain the results obtained by generating a population of 10× T_{max} and 10 elements. The best-known solutions in bold are obtained by exacts methods. In this section, we use the best-known solutions provided by (Kim *et al*., 2013).

Table 2. Benchmarks characteristics

Problem set	Number of Locations	Number of sub-problem	T_{max}
p1.2	32	18	2.5-42.5
p1.3	32	18	1.7-28.3
p1.4	32	18	1.2-21.2
p2.2	21	11	7.5-22.5
p2.3	21	11	5.0-15.0
p2.4	21	11	3.8-11.2
p3.2	33	20	5.0-36.7
p3.3	33	20	3.8-27.5
p3.4	33	20	25.0-120.0
p4.2	100	20	16.7-80.0
p4.3	100	20	12.5-60.0
p4.4	100	20	2.5-65.0
p5.2	66	26	1.7-43.3
p5.3	66	26	1.2-32.5
p5.4	66	26	7.5-40.0
p6.2	64	14	1.2-32.5
p6.3	64	14	5.0-26.7
p6.4	64	14	3.8-20.0
p7.2	102	20	10.0-200.0
p7.3	102	20	6.7-133.3
p7.4	102	20	5.0-100.0

Table 3. The computational results for data set 1 with m = 2

Problem	Best-known solution	Best-obtained solution	Time (Second)	Best-obtained solution	Time (Second)	$T_{\max}$
p1.2.a	0	0	0.016	0	0.006	2.5
p1.2.b	15	15	0.013	15	0.008	5.0
p1.2.c	20	20	0.016	20	0.006	7.5
p1.2.d	30	30	0.012	30	0.007	10.0
p1.2.e	45	45	0.021	40	0.017	12.5
p1.2.f	80	80	0.065	60	0.008	15.0
p1.2.g	90	85	0.074	65	0.009	17.5
p1.2.h	110	100	0.093	75	0.008	20.0
p1.2.i	135	120	0.124	90	0.016	23.0
p1.2.j	155	130	0.112	120	0.011	25.0
p1.2.k	175	140	0.228	110	0.011	27.5
p1.2.l	195	155	0.134	155	0.012	30.0
p1.2.m	215	165	0.148	120	0.016	32.5
p1.2.n	235	165	0.172	125	0.02	35.0
p1.2.o	240	170	0.167	120	0.019	36.5
p1.2.p	250	180	0.169	130	0.013	37.5
p1.2.q	265	180	0.181	130	0.013	40.0
p1.2.r	280	195	0.21	145	0.023	42.5

Table 4. The computational results for data set 1 with m = 3

Problem	Best-known solution	Best-obtained solution	Time (Second)	Best-obtained solution	Time (Second)	$T_{\max}$
p1.3.a	0	0	0.013	0	0.007	1.7
p1.3.b	0	0	0.024	0	0.008	3.3
p1.3.c	15	15	0.025	15	0.01	5.0
p1.3.d	15	15	0.044	15	0.009	6.7
p1.3.e	30	30	0.03	30	0.01	8.3
p1.3.f	40	40	0.055	40	0.05	10.0
p1.3.g	50	50	0.054	50	0.01	11.7
p1.3.h	70	70	0.074	65	0.01	13.3
p1.3.i	105	100	0.091	80	0.01	15.3
p1.3.j	115	105	0.108	90	0.03	16.7
p1.3.k	135	120	0.022	105	0.06	18.3
p1.3.l	155	130	0.024	105	0.01	20.0
p1.3.m	175	140	0.023	120	0.04	21.7
p1.3.n	190	160	0.025	135	0.01	23.3
p1.3.o	205	165	0.021	145	0.01	24.3
p1.3.p	220	170	0.012	155	0.02	25.0
p1.3.q	230	180	0.035	150	0.02	26.7
p1.3.r	250	185	0.028	165	0.01	28.3

Table 5. The computational results for data set 1 with m = 4

Problem	Best-known solution	Best-obtained solution	Time (Second)	Best-obtained solution	Time (Second)	$T_{\max}$
p1.4.a	0	0	0.006	0	0.004	1.2
p1.4.b	0	0	0.006	0	0.005	2.5
p1.4.c	0	0	0.006	0	0.004	3.8
p1.4.d	15	15	0.007	15	0.008	5.0
p1.4.e	15	15	0.009	15	0.007	6.2
p1.4.f	25	25	0.009	25	0.006	7.5
p1.4.g	35	35	0.011	35	0.005	8.8
p1.4.h	45	45	0.013	45	0.007	10.0
p1.4.i	60	60	0.015	55	0.006	11.5
p1.4.j	75	75	0.015	70	0.007	12.5
p1.4.k	100	100	0.017	85	0.007	13.8
p1.4.l	120	120	0.024	110	0.008	15.0
p1.4.m	130	125	0.023	110	0.008	16.2
p1.4.n	155	140	0.035	120	0.012	17.5
p1.4.o	165	150	0.093	130	0.013	18.2
p1.4.p	175	155	0.095	130	0.01	18.8
p1.4.q	190	160	0.032	125	0.013	20.0
p1.4.r	210	165	0.023	145	0.019	21.2

Table 6. The computational results for data set 2 with m = 2

Problem	Best-known solution	Best-obtained solution	Time (Second)	Best-obtained solution	Time (Second)	T_{max}
p2.2.a	90	90	0.03	85	0.005	7.5
p2.2.b	120	120	0.009	120	0.005	10.0
p2.2.c	140	140	0.011	130	0.018	11.5
p2.2.d	160	160	0.048	145	0.012	12.5
p2.2.e	190	180	0.011	145	0.008	13.5
p2.2.f	200	200	0.013	180	0.009	15.0
p2.2.g	200	200	0.042	185	0.008	16.0
p2.2.h	230	230	0.065	185	0.009	17.5
p2.2.i	230	230	0.012	230	0.013	19.0
p2.2.j	260	230	0.023	210	0.011	20.0
p2.2.k	275	260	0.099	230	0.025	22.5

Table 7. The computational results for data set 2 with m = 3

Problem	Best-known solution	Best-obtained solution	Time (Second)	Best-obtained solution	Time (Second)	T_{max}
p2.3.a	70	70	0.028	70	0.013	5.0
p2.3.b	70	70	0.035	70	0.047	6.7
p2.3.c	105	105	0.036	105	0.007	7.7
p2.3.d	105	105	0.039	105	0.007	8.3
p2.3.e	120	120	0.045	120	0.01	9.0
p2.3.f	120	120	0.009	120	0.004	10.0
p2.3.g	145	145	0.01	140	0.01	10.7
p2.3.h	165	165	0.05	160	0.016	11.7
p2.3.i	200	200	0.012	180	0.019	12.7
p2.3.j	200	200	0.017	185	0.06	13.3
p2.3.k	200	200	0.012	200	0.008	15.0

Table 8. The computational results for data set 2 with m = 4

Problem	Best-known solution	Best-obtained solution	Time (Second)	Best-obtained solution	Time (Second)	T_{max}
p2.4.a	10	10	0.008	10	0.005	3.8
p2.4.b	70	70	0.009	70	0.006	5.0
p2.4.c	70	70	0.008	70	0.006	5.8
p2.4.d	70	70	0.007	70	0.006	6.2
p2.4.e	70	70	0.008	70	0.008	6.8
p2.4.f	105	105	0.01	105	0.019	7.5
p2.4.g	105	105	0.009	105	0.008	8.0
p2.4.h	120	120	0.011	120	0.009	8.8
p2.4.i	120	120	0.012	120	0.007	9.5
p2.4.j	120	120	0.011	120	0.013	10.0
p2.4.k	180	180	0.011	165	0.009	11.2

Table 9. The computational results for data set 3 with m = 2

Problem	Best-known solution	Best-obtained solution	Time (Second)	Best-obtained solution	Time (Second)	T_{max}
p3.2.a	90	90	0.006	80	0.006	7.5
p3.2.b	150	150	0.01	120	0.008	10.0
p3.2.c	180	180	0.014	160	0.015	12.5
p3.2.d	220	190	0.014	170	0.004	15.0
p3.2.e	260	230	0.018	210	0.008	17.5
p3.2.f	300	240	0.019	210	0.011	20.0
p3.2.g	360	270	0.023	190	0.015	22.5
p3.2.h	410	300	0.024	210	0.007	25.0
p3.2.i	460	330	0.028	270	0.019	27.5
p3.2.j	510	350	0.065	280	0.019	30.0
p3.2.k	550	370	0.028	270	0.019	32.5
p3.2.l	590	380	0.032	350	0.014	35.0
p3.2.m	620	390	0.034	320	0.015	37.5
p3.2.n	660	420	0.055	340	0.027	40.0
p3.2.o	690	430	0.031	370	0.066	42.5
p3.2.p	720	440	0.036	360	0.013	45.0
p3.2.q	760	460	0.076	380	0.015	47.5
p3.2.r	790	510	0.043	430	0.022	50.0
p3.2.s	800	520	0.045	450	0.072	52.5
p3.2.t	800	550	0.048	520	0.025	55.0

Table 10. The computational results for data set 3 with m = 3

Problem	Best-known solution	Best-obtained solution	Time (Second)	Best-obtained solution	Time (Second)	T_{max}
p3.3.a	30	30	0.008	30	0.006	5.0
p3.3.b	90	90	0.011	90	0.008	6.7
p3.3.c	120	120	0.012	120	0.008	8.3
p3.3.d	170	170	0.027	170	0.014	10.0
p3.3.e	200	200	0.014	200	0.04	11.7
p3.3.f	230	220	0.018	180	0.006	13.3
p3.3.g	270	240	0.032	200	0.05	15.0
p3.3.h	300	270	0.018	250	0.009	16.7
p3.3.i	330	300	0.021	280	0.012	18.3
p3.3.j	380	310	0.043	270	0.02	20.0
p3.3.k	440	370	0.026	310	0.017	21.7
p3.3.l	480	370	0.058	320	0.023	23.3
p3.3.m	520	370	0.031	320	0.024	25.0
p3.3.n	570	430	0.026	330	0.013	26.7
p3.3.o	590	420	0.033	380	0.014	28.3
p3.3.p	640	470	0.068	370	0.014	30.0
p3.3.q	680	500	0.036	480	0.014	31.7
p3.3.r	710	530	0.036	460	0.019	33.3
p3.3.s	720	510	0.037	460	0.015	35.0
p3.3.t	760	550	0.05	550	0.04	36.7

Table 11. The computational results for data set 3 with m = 4

Problem	Best-known solution	Best-obtained solution	Time (Second)	Best-obtained solution	Time (Second)	T_{max}
p3.4.a	20	20	0.021	20	0.015	3.8
p3.4.b	30	30	0.011	30	0.008	5.0
p3.4.c	90	90	0.011	90	0.006	6.2
p3.4.d	100	100	0.012	100	0.009	7.5
p3.4.e	140	140	0.023	140	0.009	8.8
p3.4.f	190	190	0.015	190	0.011	10.0
p3.4.g	220	220	0.018	220	0.013	11.2
p3.4.h	240	240	0.021	240	0.013	12.5
p3.4.i	270	240	0.019	230	0.012	13.8
p3.4.j	310	300	0.024	260	0.014	15.0
p3.4.k	350	310	0.023	270	0.016	16.2
p3.4.l	380	320	0.044	310	0.015	17.5
p3.4.m	390	350	0.056	290	0.013	18.8
p3.4.n	440	370	0.027	330	0.012	20.0
p3.4.o	500	410	0.055	380	0.02	21.2
p3.4.p	560	430	0.037	390	0.025	22.5
p3.4.q	560	440	0.031	380	0.017	23.8
p3.4.r	600	470	0.19	410	0.016	25.0
p3.4.s	670	500	0.038	430	0.026	26.2
p3.4.t	670	500	0.045	410	0.013	27.5

Table 12. The computational results for data set 4 with m = 2

Problem	Best-known solution	Best-obtained solution	Time (Second)	Best-obtained solution	Time (Second)	T_{max}
p4.2.a	206	152	0.041	131	0.046	25.0
p4.2.b	341	222	0.048	152	0.011	30.0
p4.2.c	452	231	0.063	169	0.011	35.0
p4.2.d	531	260	0.063	180	0.012	40.0
p4.2.e	618	253	0.067	200	0.017	45.0
p4.2.f	687	263	0.066	193	0.041	50.0
p4.2.g	757	299	0.086	216	0.027	55.0
p4.2.h	835	316	0.09	236	0.018	60.0
p4.2.i	918	310	0.093	232	0.02	65.0
p4.2.j	965	322	0.098	297	0.024	70.0
p4.2.k	1022	356	0.108	282	0.016	75.0
p4.2.l	1074	354	0.103	296	0.021	80.0
p4.2.m	1132	375	0.123	316	0.023	85.0
p4.2.n	1174	421	0.13	294	0.021	90.0
p4.2.o	1218	389	0.292	355	0.11	95.0
p4.2.p	1242	435	0.132	315	0.019	100.0
p4.2.q	1268	435	0.142	320	0.03	105.0
p4.2.r	1292	487	0.182	318	0.02	110.0
p4.2.s	1304	483	0.151	340	0.102	115.0
p4.2.t	1306	455	0.16	348	0.04	120.0

Table 13. The computational results for data set 4 with m = 3

Problem	Best-known solution	Best-obtained solution	Time (Second)	Best-obtained solution	Time (Second)	T_{max}
p4.3.a	0	0	0.011	0	0.002	16.7
p4.3.b	38	38	0.04	38	0.07	20.0
p4.3.c	193	173	0.05	123	0.02	23.3
p4.3.d	335	210	0.051	187	0.013	26.7
p4.3.e	468	255	0.06	212	0.01	30.0
p4.3.f	579	292	0.065	248	0.017	33.3
p4.3.g	653	298	0.071	265	0.018	36.7
p4.3.h	729	328	0.086	272	0.026	40.0
p4.3.i	809	314	0.091	274	0.023	43.3
p4.3.j	861	363	0.092	310	0.014	46.7
p4.3.k	919	353	0.251	305	0.099	50.0
p4.3.l	979	370	0.12	321	0.031	53.3
p4.3.m	1063	412	0.126	300	0.013	56.7
p4.3.n	1121	405	0.131	330	0.017	60.0
p4.3.o	1172	415	0.123	342	0.039	63.3
p4.3.p	1222	425	0.16	418	0.076	66.7
p4.3.q	1253	457	0.144	360	0.022	70.0
p4.3.r	1273	451	0.153	388	0.026	73.3
p4.3.s	1295	478	0.16	371	0.024	76.7
p4.3.t	1305	475	0.165	421	0.039	80.0

Table 14. The computational results for data set 4 with m = 4

Problem	Best-known solution	Best-obtained solution	Time (Second)	Best-obtained solution	Time (Second)	T_{max}
p4.4.a	0	0	0.034	0	0.008	12.5
p4.4.b	0	0	0.03	0	0.008	15.0
p4.4.c	0	0	0.05	0	0.01	17.5
p4.4.d	38	38	0.054	38	0.01	20.0
p4.4.e	183	183	0.223	172	0.013	22.5
p4.4.f	324	270	0.07	208	0.017	25.0
p4.4.g	461	277	0.08	244	0.036	27.5
p4.4.h	571	333	0.092	275	0.021	30.0
p4.4.i	657	332	0.088	292	0.033	32.5
p4.4.j	732	350	0.091	342	0.025	35.0
p4.4.k	821	403	0.093	379	0.022	37.5
p4.4.l	880	417	0.103	349	0.021	40.0
p4.4.m	919	420	0.12	356	0.023	42.5
p4.4.n	977	420	0.114	370	0.022	45.0
p4.4.o	1061	451	0.12	392	0.026	47.5
p4.4.p	1124	458	0.123	380	0.116	50.0
p4.4.q	1161	486	0.147	382	0.021	52.5

Table 15. The computational results for data set 5 with m = 2

Problem	Best-known solution	Best-obtained solution	Time (Second)	Best-obtained solution	Time (Second)	T_{max}
p5.2.a	0	0	0.035	0	0.009	2.5
p5.2.b	20	20	0.011	20	0.011	5.0
p5.2.c	50	50	0.02	40	0.01	7.5
p5.2.d	80	80	0.021	70	0.013	10.0
p5.2.e	180	170	0.035	135	0.025	12.5
p5.2.f	240	175	0.023	145	0.066	15.0
p5.2.g	320	205	0.137	190	0.015	17.5
p5.2.h	410	270	0.069	220	0.014	20.0
p5.2.i	480	320	0.041	275	0.025	22.5
p5.2.j	580	350	0.071	270	0.011	25.0
p5.2.k	670	375	0.083	375	0.04	27.5
p5.2.l	800	435	0.067	390	0.032	30.0
p5.2.m	860	430	0.072	375	0.024	32.5
p5.2.n	925	460	0.066	425	0.031	35.0
p5.2.o	1020	470	0.072	410	0.026	37.5
p5.2.p	1150	480	0.202	470	0.034	40.0
p5.2.q	1195	550	0.094	550	0.045	42.5
p5.2.r	1260	575	0.089	450	0.041	45.0
p5.2.s	1340	570	0.088	510	0.046	47.5
p5.2.t	1400	580	0.094	550	0.043	50.0
p5.2.u	1460	610	0.111	570	0.06	52.5
p5.2.v	1505	640	0.118	575	0.044	55.0
p5.2.w	1565	660	0.115	560	0.046	57.5
p5.2.x	1610	660	0.117	585	0.054	60.0
p5.2.y	1645	710	0.116	570	0.035	62.5
p5.2.z	1680	730	0.123	610	0.046	65.0

DOI: 10.3844/jcssp.2017.440.451

Table 16. The computational results for data set 5 with m = 3

Problem	Best-known solution	Best-obtained solution	Time (Second)	Best-obtained solution	Time (Second)	T_{max}
p5.3.a	0	0	0.005	0	0.009	1.7
p5.3.b	15	15	0.01	15	0.008	3.3
p5.3.c	20	20	0.02	20	0.01	5.0
p5.3.d	60	60	0.02	60	0.01	6.7
p5.3.e	95	95	0.02	95	0.016	8.3
p5.3.f	110	110	0.03	110	0.01	10.0
p5.3.g	185	150	0.02	140	0.01	11.7
p5.3.h	260	210	0.06	185	0.03	13.3
p5.3.i	335	240	0.04	220	0.07	15.0
p5.3.j	470	285	0.03	275	0.02	16.7
p5.3.k	495	335	0.08	305	0.01	18.3
p5.3.l	595	385	0.06	335	0.02	20.0
p5.3.m	650	425	0.06	390	0.02	21.7
p5.3.n	755	455	0.06	410	0.02	23.3
p5.3.o	870	510	0.081	345	0.03	25.0
p5.3.p	990	535	0.08	475	0.02	26.7
p5.3.q	1070	510	0.07	530	0.07	28.3
p5.3.r	1125	595	0.08	570	0.05	30.0
p5.3.s	1190	610	0.1	610	0.05	31.7
p5.3.t	1260	595	0.1	515	0.024	33.3
p5.3.u	1345	630	0.09	560	0.09	35.0
p5.3.v	1425	695	0.09	530	0.03	36.7
p5.3.w	1485	645	0.09	595	0.065	38.3
p5.3.x	1555	645	0.08	660	0.044	40.0
p5.3.y	1595	735	0.112	690	0.04	41.7
p5.3.z	1635	755	0.13	695	0.052	43.3

Table 17. The computational results for data set 5 with m = 4

Problem	Best-known solution	Best-obtained solution	Time (Second)	Best-obtained solution	Time (Second)	T_{max}
p5.4.a	0	0	0.01	0	0.012	1.2
p5.4.b	0	0	0.008	0	0.06	2.5
p5.4.c	20	20	0.02	20	0.001	3.8
p5.4.d	20	20	0.02	20	0.01	5.0
p5.4.e	20	20	0.019	20	0.01	6.2
p5.4.f	80	80	0.02	80	0.01	7.5
p5.4.g	140	140	0.035	125	0.01	8.8
p5.4.h	140	140	0.03	140	0.01	10.0
p5.4.i	240	200	0.24	185	0.021	11.2
p5.4.j	340	230	0.05	210	0.06	12.5
p5.4.k	340	285	0.06	270	0.02	13.8
p5.4.l	430	310	0.05	305	0.024	15.0
p5.4.m	555	360	0.059	345	0.02	16.2
p5.4.n	620	405	0.05	375	0.017	17.5
p5.4.o	690	435	0.06	405	0.032	18.8
p5.4.p	765	490	0.076	435	0.02	20.0
p5.4.q	860	500	0.07	470	0.03	21.2
p5.4.r	960	540	0.08	505	0.032	22.5
p5.4.s	1030	580	0.08	535	0.03	23.8
p5.4.t	1160	595	0.08	560	0.02	25.0
p5.4.u	1300	660	0.091	590	0.041	26.2
p5.4.v	1320	705	0.09	635	0.041	27.5
p5.4.w	1390	650	0.111	650	0.06	28.8
p5.4.x	1450	675	0.1	670	0.05	30.0
p5.4.y	1520	740	0.12	630	0.03	31.2
p5.4.z	1620	740	0.09	640	0.04	32.5

Table 18. The computational results for data set 6 with m = 2

Problem	Best-known solution	Best-obtained solution	Time(Second)	Best-obtained solution	Time(Second)	T_{max}
p6.2.a	0	0	0.01	0	0.009	7.5
p6.2.b	0	0	0.016	0	0.01	10.0
p6.2.c	0	0	0.014	0	0.009	12.5
p6.2.d	192	168	0.02	156	0.02	15.0
p6.2.e	360	264	0.03	204	0.039	17.5
p6.2.f	588	348	0.04	288	0.01	20.0
p6.2.g	660	432	0.041	330	0.024	22.5
p6.2.h	780	402	0.09	366	0.025	25.0
p6.2.i	888	468	0.059	354	0.021	27.5
p6.2.j	948	474	0.065	402	0.05	30.0
p6.2.k	1032	516	0.067	414	0.031	32.5
p6.2.l	1116	528	0.069	432	0.03	35.0
p6.2.m	1188	546	0.07	450	0.033	37.5
p6.2.n	1260	606	0.084	480	0.09	40.0

Table 19. The computational results for data set 6 with m = 3

Problem	Best-known solution	Best-obtained solution	Time (Second)	Best-obtained solution	Time (Second)	T_{max}
p6.3.a	0	0	0.011	0	0.009	5.0
p6.3.b	0	0	0.01	0	0.009	6.7
p6.3.c	0	0	0.05	0	0.01	8.3
p6.3.d	0	0	0.02	0	0.02	10.0
p6.3.e	0	0	0.02	0	0.01	11.7
p6.3.f	0	0	0.03	0	0.01	13.3
p6.3.g	282	234	0.044	216	0.042	15.0
p6.3.h	444	336	0.05	288	0.01	16.7
p6.3.i	642	420	0.05	336	0.02	18.3
p6.3.j	828	468	0.098	420	0.03	20.0
p6.3.k	894	534	0.06	444	0.03	21.7
p6.3.l	1002	534	0.07	486	0.036	23.3
p6.3.m	1080	636	0.07	546	0.045	25.0
p6.3.n	1170	654	0.071	528	0.045	26.7

Table 20. The computational results for data set 6 with m = 4

Problem	Best-known solution	Best-obtained solution	Time (Second)	Best-obtained solution	Time (Second)	T_{max}
p6.4.a	0	0	0.01	0	0.01	3.8
p6.4.b	0	0	0.014	0	0.008	5.0
p6.4.c	0	0	0.016	0	0.01	6.2
p6.4.d	0	0	0.018	0	0.008	7.5
p6.4.e	0	0	0.023	0	0.005	8.8
p6.4.f	0	0	0.025	0	0.009	10.0
p6.4.g	0	0	0.025	0	0.007	11.2
p6.4.h	0	0	0.03	0	0.009	12.5
p6.4.i	0	0	0.032	0	0.006	13.8
p6.4.j	366	294	0.047	264	0.015	15.0
p6.4.k	528	408	0.058	312	0.018	16.2
p6.4.l	696	450	0.058	426	0.02	17.5
p6.4.m	912	552	0.086	480	0.031	18.8
p6.4.n	1068	564	0.069	498	0.028	20.0

Table 3 contains the results obtained for data set 1 with 2 tours. Six best-known solutions are obtained for a population size equal to 10× T_{max}, while only four best-known solutions are obtained for a population size equal to 10. The results obtained for data set 1 with 3 tours are contained in Table 4. Eight best-known solutions are obtained for population size equal to 10× T_{max}, while seven best-known solutions are obtained for a population size equal to 10. The reminder tables contain the results of the rest sets. The obtained results by population size equal to 10× T_{max} are better than those obtained by a population size equal to 10.

The obtained results prove the efficiency of our heuristic. Indeed, a great number of solutions is near to the best-known solutions. Other obtained solutions reach the best-known solution in minor computational time. Those results represent a promising start for population-based metaheuristics.

Table 21. The computational results for data set 7 with m = 2

Problem	Best-known solution	Best-obtained solution	Time (Second)	Best-obtained solution	Time (Second)	T_{max}
p7.2.a	30	30	0.023	30	0.008	10.0
p7.2.b	64	64	0.008	64	0.008	20.0
p7.2.c	101	101	0.059	101	0.013	30.0
p7.2.d	190	156	0.083	131	0.074	40.0
p7.2.e	290	202	0.093	158	0.016	50.0
p7.2.f	387	255	0.105	191	0.009	60.0
p7.2.g	459	277	0.113	229	0.021	70.0
p7.2.h	521	269	0.127	232	0.014	80.0
p7.2.i	580	307	0.141	239	0.019	90.0
p7.2.j	646	325	0.15	257	0.027	100.0
p7.2.k	705	329	0.184	267	0.073	110.0
p7.2.l	767	348	0.173	289	0.018	120.0
p7.2.m	827	348	0.207	312	0.028	130.0
p7.2.n	888	389	0.212	257	0.01	140.0
p7.2.o	945	401	0.222	321	0.02	150.0
p7.2.p	1002	386	0.241	312	0.02	160.0
p7.2.q	1044	404	0.24	333	0.03	170.0
p7.2.r	1094	414	0.251	393	0.022	180.0
p7.2.s	1136	429	0.327	416	0.02	190.0
p7.2.t	1179	441	0.301	383	0.023	200.0

Table 22. The computational results for data set 7 with m = 3

Problem	Best-known solution	Best-obtained solution	Time (Second)	Best-Obtained Solution	Time (Second)	T_{max}
p7.3.a	0	0	0.02	0	0.004	6.7
p7.3.b	46	46	0.03	46	0.05	13.3
p7.3.c	79	79	0.05	79	0.01	20.0
p7.3.d	117	117	0.07	117	0.03	26.7
p7.3.e	175	175	0.23	148	0.009	33.3
p7.3.f	247	202	0.11	186	0.02	40.0
p7.3.g	344	253	0.1	214	0.13	46.7
p7.3.h	425	296	0.11	252	0.009	53.3
p7.3.i	487	311	0.13	269	0.01	60.0
p7.3.j	564	331	0.142	293	0.02	66.7
p7.3.k	633	350	0.17	253	0.02	73.3
p7.3.l	684	357	0.16	305	0.02	80.0
p7.3.m	762	388	0.18	304	0.025	86.7
p7.3.n	820	389	0.212	322	0.021	93.3
p7.3.o	874	403	0.21	346	0.02	100.0
p7.3.p	929	417	0.24	335	0.027	106.7
p7.3.q	987	475	0.241	388	0.018	113.3
p7.3.r	1026	450	0.25	389	0.02	120.0
p7.3.s	1081	446	0.26	382	0.02	126.7
p7.3.t	1120	473	0.281	401	0.022	133.3

Table 23. The computational results for data set 7 with m = 4

Problem	Best-known solution	Best-obtained solution	Time (Second)	Best-obtained solution	Time (Second)	T_{max}
p7.4.a	0	0	0.02	0	0.01	5.0
p7.4.b	30	30	0.03	30	0.012	10.0
p7.4.c	46	46	0.05	46	0.01	15.0
p7.4.d	79	79	0.07	79	0.01	20.0
p7.4.e	123	123	0.07	123	0.06	25.0
p7.4.f	164	164	0.08	155	0.01	30.0
p7.4.g	217	203	0.11	195	0.01	35.0
p7.4.h	285	247	0.121	222	0.02	40.0
p7.4.i	366	273	0.122	272	0.02	45.0
p7.4.j	462	326	0.15	287	0.028	50.0
p7.4.k	520	340	0.16	339	0.026	55.0
p7.4.l	590	376	0.16	351	0.02	60.0
p7.4.m	646	399	0.194	343	0.02	65.0
p7.4.n	730	422	0.226	370	0.021	70.0
p7.4.o	781	436	0.357	401	0.034	75.0
p7.4.p	846	456	0.223	398	0.026	80.0
p7.4.q	909	457	0.232	409	0.023	85.0
p7.4.r	970	490	0.254	388	0.024	90.0
p7.4.s	1022	498	0.254	399	0.028	95.0
p7.4.t	1077	506	0.271	449	0.03	100.0

Conclusion

This paper, introduces a randomized population constructive that generates the initial population for the team orienteering problem. The obtained results promote the integration of the heuristic to built initial population for population-based metaheuristics. The use of permutations for the proposed heuristic avoids infeasible solutions. Moreover, dealing with permutations during the search process would facilitate the application of operators and avoid infeasible solutions. Indeed, for each permutation a unique solution is associated by applying only the extern loop of the proposed heuristic. In this study, the best solution of the population is enhanced by a local search using the swap and the add/drop operators. This component allows finding better solutions.

Ethics

The author confirms that this manuscript has not been published elsewhere and that no ethical issues are involved.

References

Archetti, C., A. Hertz and M.G. Speranza, 2007. Metaheuristics for the team orienteering problem. J. Heurist., 13: 49-76. DOI: 10.1007/s10732-006-9004-0

Bouly, H., D.C. Dang and A. Moukrim, 2009. A memetic algorithm for the team orienteering problem. 4OR, 8: 49-70. DOI: 10.1007/s10288-008-0094-4

Butt, S.E. and T.M. Cavalier, 1994. A heuristic for the multiple tour maximum collection problem. Comput. Oper. Res., 21: 101-111. DOI: 10.1016/0305-0548(94)90065-5

Chao, I.M., B.L. Golden and E.A. Wasil, 1996. The team orienteering problem. Eur. J. Operat. Res., 88: 464-474. DOI: 10.1016/0377-2217(94)00289-4

Golden, B.L., L. Levy and R. Vohra, 1987. The orienteering problem. Nav. Res. Logist., 34: 307-318. DOI: 10.1002/1520-6750(198706)34:3<307::AID-NAV3220340302>3.0.CO;2-D

Ke, L., C. Archetti and Z. Feng, 2008. Ants can solve the team orienteering problem. Comput. Ind. Eng., 54: 648-665. DOI: 10.1016/j.cie.2007.10.001

Kim, B.I., H. Li and A.L. Johnson, 2013. An augmented large neighborhood search method for solving the team orienteering problem. Expert Syst. Applic., 40: 3065-3072. DOI: 10.1016/j.eswa.2012.12.022

Lin, S.W., 2013. Solving the team orienteering problem using effective multi-start simulated annealing. Applied Soft Comput. 13: 1064-1073. DOI: 10.1016/j.asoc.2012.09.022

Souffriau, W., P. Vansteenwegen, J. Vertommen, G.V. Berghe and D.V. Oudheusden, 2008. A personalized tourist trip design algorithm for mobile tourist guides. Applied Artif. Intell., 22: 964-985. DOI: 10.1080/08839510802379626

Tang, H. and E. Miller-Hooks, 2005. A TABU search heuristic for the team orienteering problem. Comput. Operat. Res., 32: 1379-1407. DOI: 10.1016/j.cor.2003.11.008

Thomadsen, T. and T. Stidsen, 2003. The quadratic selective travelling salesman problem. Informatics and Mathematical Modelling, Technical University of Denmark, DTU.

Tsiligirides, T., 1984. Heuristic methods applied to orienteering. J. Operat. Res. Society, 35: 797-809. DOI: 10.1057/jors.1984.162

Vansteenwegen, P., W. Souffriau, G.V. Berghe and D.V. Oudheusden, 2009. A guided local search metaheuristic for the team orienteering problem. Eur. J. Operat. Res., 196 : 118-127. DOI: 10.1016/j.ejor.2008.02.037

Vansteenwegen, P., W. Souffriau and D.V. Oudheusden, 2011. The orienteering problem: A survey. Eur. J. Operat. Res., 209 : 1-10. DOI: 10.1016/j.ejor.2010.03.045

Zettam, M. and B. Elbenani, 2016. A novel randomized heuristic for the team orienteering problem. Proceedings of the 3rd International Conference on Logistics Operations Management, May 23-25, IEEE Xplore Press, pp: 1-6. DOI: 10.1109/GOL.2016.7731671