A novel Randomized Heuristic for the Team Orienteering Problem

Manal Zettam
Department of computer science
Mohammed V University
Rabat, Morocco
zettammanal@gmail.com

Bouazza Elbenani
Department of computer science
Mohammed V University
Rabat, Morocco
elbenani@fsr.ac.ma

Abstract— **During the last decades, researchers have shown a great interest in studying the team orienteering problem due to its applications in several fields such as tourist trip planning, technician routing and athlete recruiting. Therefore, solving optimally the team orienteering problem would play a major role in logistics management. In this paper, an unprecedented heuristic is applied to the team orienteering problem. The heuristic is used to generate the initial population for population-based metaheuristics. The heuristic proved its efficiency. Indeed, experiments conducted on the team orienteering problem show that the initial population generated by our heuristic contains the best-known solutions for 67 benchmarks and good solutions for a great number of benchmarks. A good initial population will reduce the computational time and the number of iterations necessary to reaches the best-known solutions.**

Keywords—Logistics Management; Team Orienteering Problem; Orienteering Problem; Randomized Heuristic.

I. INTRODUCTION

The term Orienteering Problem (OP) first introduced by Bruce et al., 1987 [1], originates from a sport entitled Orienteering. Orienteering is an outdoor sport generally played in a heavily forested or mountainous area. A number of control points are located in the forest. Each of these control points is associated with a score. Competitors equipped with compass and map, are required to visit a subset of control points from the starts point to the end point in such a manner to maximize their total score within a predefined amount of time. According to Bruce et al., 1987 [1] OP is an NP-hard problem. OP and its variants were given a great interest as a result of their applications. Vansteenwegen et al., 2011 [2] gives a recent survey of OP and its variants applications. Tourist trip planning, technician routing and athlete recruiting are three of well known application of OP.

The Team Orienteering Problem (TOP) extends OP. It was first introduced by Chao et al., 1996 [3]. Competitors are partitioned into teams. The competitors of a given team collaborate to maximize the score within a predefined amount of time. Each control point is visited once by a member of a given team. According to Chao et al [3] TOP is an NP-hard problem. Therefore, research efforts focuses on heuristics and metaheuristics to solve TOP.

Dang et al., 2013 [4] highlighted three methods, described as the most significant methods within the state-of- art of TOP. The first method is a slow variable neighborhood search (SVNS) presented by Archetti et al., 2015 [5]. The second method is a slow Path-Relinking (SPR) introduced by Souffriau et al., 2010 [6]. The third one and the most interesting is the memetic algorithm presented by Bouly et al., 2009 [7] that proposed an efficient technique to represent TOP solutions known as giant tours. Heuristics using giant tours tend to reach easily global optimum as stated by Dang et al., 2013 [4].

In this paper, we introduce a heuristic that generates the initial population. This heuristic takes a permutation of control points, a seed and the size of the initial population l as incomes. For each permutation the heuristic generates randomly permutations of control points. Then m paths beginning from the starting control point 1 and ending at the ending control point n are constructed.

The reminder of this paper is organized as follows. The first section introduces the team orienteering problem and the mathematical model adopted in this work. The second section introduces our heuristic. Further, the computational results are shown in the fourth section followed by concluding remarks and proposal for future works in the fifth section.

II. THE TEAM ORIENTEERING PROBLEM

A. Team Orienteering Problem

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 aim of TOP is finding m tours starting from the location 1 and ending at the location n in such a manner to maximize the total score S. A travel time or length of a tour cannot exceed $T_{\max}$. In our work, we considered the Euclidean distance as a measure to calculate the length of a given tour. Table I shows an example of TOP instances where For $T_{\max}=10$ and $m=2$ the departure location is one and the arrival location is thirty-two. Table I contains locations, XY coordinates and scores. The solution of the instance shown in Table I is represented on figure 1.

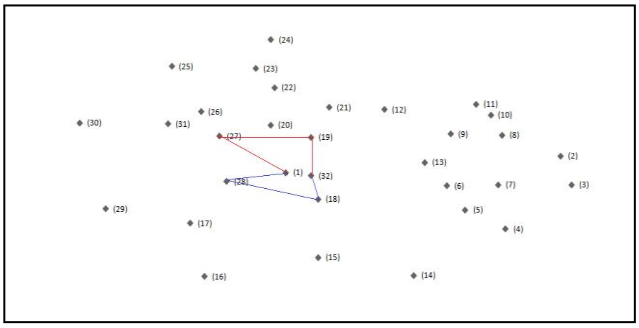

Fig. 1. Example of a solution.

TABLE I. 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

B. The Mathematical Model

In this paper, a mathematical model similar to the one used by Lin, 2013 [8] is described as follows:

$$Maximize \sum_{k=1}^{m}\sum_{i=2}^{n-1} S_i y_i^k \tag{1}$$

subject to

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

$$\sum_{i=1}^{n-1} x_{il}^k = \sum_{j=2}^{n} x_{lj}^k = y_l^k,\ \forall l = 2,...,\mathrm{n}-1; \forall k = 1,...,\mathrm{m} \tag{3}$$

$$\sum_{k=1}^{m} y_i^k \le 1,\ \forall i = 2,...,n-1 \tag{4}$$

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

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

$$x_{ij}^k, y_i^k \in \{0,1\},\ \forall \mathrm{i,j} = 1,...,\mathrm{n};\ \forall k = 1,...,m \tag{7}$$

Where S_i is the score associated to the i^{th} location. t_{ij} is the length of the path starting at the i^{th} location and ending at the j^{th} location. The travel length cannot exceed $T_{\max}$. V is the set of locations. U is a subset of V.

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

$y_i^k = 1$ if location i is visited in tour k, 0 otherwise.

The equation (1) represents the objective function to maximize. The equation (2) ensures that all tours starts from location 1 and ends at location n. The equation (3) maintains the connectivity of tours. 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 x and y variables are binary.

III. THE NOVEL RANDOMIZED HEURISTIC

The randomized heuristic consists in randomly generating a given number of permutations. On the basis of these permutations solutions are constructed. A solution in our case is represented by m connected tours. Given a permutation of n elements. Each element represents a location. For each tour k, the heuristic browses a permutation. If the i^{th} location is not contained in other tours and if adding the i^{th} location does not violate fifth constraint described in the mathematical model, the i^{th} location is added to the k^{th} tour. Otherwise, the heuristic passes to the next iteration and so on. 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 denoted $T_{\max}$,
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 $i \in V$
if($x_{ij} + c \leq T_{\max}$ ***)***
$c := c + dist(i, j)$
$j := i$
end if
end for
end for
end for
Output: *A feasible solution*

The randomized heuristic reaches best-known solutions for a great number of benchmarks in a reasonable computational time. It 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.

IV. THE COMPUTATIONAL RESULTS

This section contains the computational results we obtained by applying the randomized heuristic on TOP benchmarks. The proposed heuristic was coded in JAVA language, and was run on a personal computer with an Intel Core i5-2540M 2.60GHz processor.

The benchmarks we used in this paper are available at (https://www.hds.utc.fr/~moukrim). Table II summarizes the characteristics of problem benchmarks sets.

TABLE II. 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
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

Problem set	*Number of Locations*	*Number of sub-problem*	$T_{\max}$
p4.3	100	20	12.5-60.0

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. For each set of problem and number of tour the maximum length denoted $T_{\max}$ varies.

The size of the initial population is set at 10 for all instances. We ran our heuristic 10 times on each instance in order to select the best obtained solution. Table III contains the problem, the best-known solution in the literature and also the best solution in the population we obtained by applying the proposed randomized heuristic. 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[9].

TABLE III. THE COMPUTATIONAL RESULTS

Problem	*Best-known solution*	*Best obtained solution*	*Time (Second)*
p1.2.a	**0**	0	0.016
p1.2.b	**15**	15	0.013
p1.2.c	**20**	20	0.016
p1.2.d	**30**	30	0.012
p1.2.e	**45**	40	0.016
p1.2.f	**80**	45	0.017
p1.2.g	**90**	55	0.015
p1.2.h	110	55	0.013
p1.2.i	**135**	60	0.018
p1.2.j	155	60	0.015
p1.2.k	175	55	0.019
p1.2.l	**195**	65	0.011
p1.2.m	**215**	70	0.014
p1.2.n	**235**	70	0.02
p1.2.o	**240**	75	0.016
p1.2.p	250	75	0.006
p1.2.q	265	75	0.016
p1.2.r	280	80	0.007
p1.3.a	**0**	0	0.009
p1.3.b	**0**	0	0.007
p1.3.c	**15**	15	0.008
p1.3.d	**15**	15	0.007
p1.3.e	**30**	30	0.004
p1.3.f	**40**	40	0.006
p1.3.g	**50**	50	0.007
p1.3.h	70	60	0.005
p1.3.i	**105**	65	0.007
p1.3.j	**115**	70	0.007
p1.3.k	**135**	70	0.006
p1.3.l	**155**	75	0.005
p1.3.m	175	80	0.007
p1.3.n	190	80	0.007
p1.3.o	205	85	0.006
p1.3.p	**220**	80	0.008
p1.3.q	230	85	0.008
p1.3.r	250	85	0.006
p1.4.a	**0**	0	0.007
p1.4.b	**0**	0	0.005
p1.4.c	**0**	0	0.006
p1.4.d	**15**	15	0.005
p1.4.e	**15**	15	0.006
p1.4.f	**25**	25	0.007
p1.4.g	**35**	35	0.005
p1.4.h	**45**	45	0.005
p1.4.i	**60**	55	0.013
p1.4.j	**75**	70	0.007
p1.4.k	**100**	75	0.045
p1.4.l	**120**	75	0.007
p1.4.m	130	85	0.039
p1.4.n	**155**	85	0.008
p1.4.o	**165**	90	0.007
p1.4.p	**175**	90	0.007
p1.4.q	**190**	90	0.006
p1.4.r	**210**	100	0.044
p2.2.a	**90**	70	0.005
p2.2.b	**120**	90	0.038
p2.2.c	**140**	90	0.036
p2.2.d	**160**	100	0.004
p2.2.e	**190**	110	0.004
p2.2.f	**200**	110	0.004
p2.2.g	**200**	110	0.004
p2.2.h	**230**	120	0.004
p2.2.i	**230**	120	0.004
p2.2.j	**260**	120	0.039
p2.2.k	275	135	0.004
p2.3.a	**70**	70	0.006
p2.3.b	**70**	70	0.005
p2.3.c	**105**	105	0.039
p2.3.d	**105**	105	0.005
p2.3.e	**120**	120	0.003
p2.3.f	**120**	120	0.005
p2.3.g	**145**	125	0.004
p2.3.h	**165**	130	0.039
p2.3.i	**200**	135	0.004
p2.3.j	**200**	135	0.005
p2.3.k	**200**	150	0.004
p2.4.a	**10**	10	0.004
p2.4.b	**70**	70	0.008
p2.4.c	**70**	70	0.005
p2.4.d	**70**	70	0.004
p2.4.e	**70**	70	0.006
p2.4.f	**105**	105	0.007
p2.4.g	**105**	105	0.006
p2.4.h	**120**	120	0.004
p2.4.i	**120**	120	0.007
p2.4.j	**120**	120	0.006
p2.4.k	**180**	150	0.004
p3.2.a	**90**	80	0.004
p3.2.b	**150**	110	0.008
p3.2.c	**180**	130	0.025
p3.2.d	220	140	0.012
p3.2.e	260	150	0.103
p3.2.f	300	150	0.124
p3.2.g	360	150	0.014
p3.2.h	410	150	0.012
p3.2.i	460	180	0.148
p3.2.j	510	190	0.023
p3.2.k	550	220	0.127
p3.2.l	590	200	0.029
p3.2.m	620	200	0.014
p3.2.n	660	240	0.014
p3.2.o	690	240	0.023
p3.2.p	720	240	0.021
p3.2.q	760	240	0.025
p3.2.r	790	240	0.137
p3.2.s	800	260	0.135
p3.2.t	**800**	260	0.118
p3.3.a	**30**	30	0.024
p3.3.b	**90**	90	0.016
p3.3.c	**120**	120	0.016
p3.3.d	**170**	150	0.015
p3.3.e	**200**	170	0.018
p3.3.f	**230**	160	0.02
p3.3.g	270	180	0.023
p3.3.h	**300**	190	0.123
p3.3.i	330	200	0.022
p3.3.j	380	210	0.022
p3.3.k	440	220	0.127
p3.3.l	480	240	0.023
p3.3.m	520	260	0.022
p3.3.n	570	230	0.121
p3.3.o	590	250	0.022
p3.3.p	640	250	0.026
p3.3.q	680	280	0.08
p3.3.r	**710**	270	0.123
p3.3.s	720	270	0.019
p3.3.t	760	260	0.014
p3.4.a	**20**	20	0.019
p3.4.b	**30**	30	0.018
p3.4.c	**90**	90	0.017
p3.4.d	**100**	100	0.016
p3.4.e	**140**	140	0.023
p3.4.f	**190**	170	0.019
p3.4.g	**220**	190	0.018
p3.4.h	**240**	190	0.022
p3.4.i	**270**	200	0.124
p3.4.j	310	220	0.023
p3.4.k	**350**	220	0.159
p3.4.l	**380**	250	0.028
p3.4.m	**390**	250	0.024
p3.4.n	440	250	0.228
p3.4.o	**500**	300	0.027
p3.4.p	**560**	280	0.136
p3.4.q	560	310	0.023
p3.4.r	600	270	0.163
p3.4.s	**670**	280	0.024

Problem	*Best-known solution*	*Best obtained solution*	*Time (Second)*
p3.4.t	**670**	300	0.035
p4.2.a	**206**	83	0.042
p4.2.b	**341**	91	0.031
p4.2.c	452	97	0.04
p4.2.d	531	94	0.025
p4.2.e	618	102	0.037
p4.2.f	687	105	0.021
p4.2.g	757	114	0.026
p4.2.h	835	130	0.03
p4.2.i	918	130	0.031
p4.2.j	965	131	0.036
p4.2.k	1022	131	0.022
p4.2.l	1074	120	0.039
p4.2.m	1132	130	0.028
p4.2.n	1174	145	0.034
p4.2.o	1218	145	0.034
p4.2.p	1242	143	0.028
p4.2.q	1268	142	0.019
p4.2.r	1292	146	0.013
p4.2.s	1304	146	0.026
p4.2.t	1306	146	0.029
p4.3.a	**0**	0	0.018
p4.3.b	**38**	38	0.022
p4.3.c	**193**	110	0.029
p4.3.d	335	134	0.053
p4.3.e	**468**	133	0.039
p4.3.f	579	134	0.054
p4.3.g	653	127	0.048
p4.3.h	729	143	0.042
p4.3.i	809	133	0.03
p4.3.j	861	144	0.027
p4.3.k	919	137	0.049
p4.3.l	979	149	0.042
p4.3.m	1063	159	0.055
p4.3.n	1121	171	0.064
p4.3.o	1172	166	0.032
p4.3.p	1222	172	0.062
p4.3.q	1253	177	0.032
p4.3.r	1273	177	0.043
p4.3.s	1295	185	0.055
p4.3.t	1305	181	0.073
p4.4.a	**0**	0	0.02
p4.4.b	**0**	0	0.034
p4.4.c	**0**	0	0.023
p4.4.d	**38**	38	0.028
p4.4.e	**183**	136	0.03
p4.4.f	**324**	144	0.039
p4.4.g	461	155	0.041
p4.4.h	**571**	164	0.039
p4.4.i	657	168	0.045
p4.4.j	732	169	0.019
p4.4.k	821	177	0.033
p4.4.l	880	177	0.07
p4.4.m	919	170	0.029
p4.4.n	977	180	0.029
p4.4.o	1061	180	0.033
p4.4.p	1124	180	0.029
p4.4.q	1161	185	0.022
p4.4.r	1216	190	0.024
p4.4.s	1260	191	0.027
p4.4.t	1285	204	0.027
p5.2.a	**0**	0	0.025
p5.2.b	**20**	20	0.027
p5.2.c	**50**	40	0.018
p5.2.d	**80**	65	0.025
p5.2.e	**180**	90	0.029
p5.2.f	**240**	90	0.131
p5.2.g	**320**	140	0.023
p5.2.h	**410**	145	0.033
p5.2.i	**480**	155	0.03
p5.2.j	**580**	165	0.038
p5.2.k	670	180	0.023
p5.2.l	800	190	0.012
p5.2.m	860	190	0.012
p5.2.n	925	215	0.029
p5.2.o	1020	200	0.036
p5.2.p	1150	210	0.02
p5.2.q	1195	215	0.02
p5.2.r	1260	230	0.021
p5.2.s	1340	235	0.019
p5.2.t	1400	260	0.024
p5.2.u	1460	250	0.02
p5.2.v	1505	250	0.02
p5.2.w	1565	260	0.02
p5.2.x	1610	260	0.18
p5.2.y	1645	270	0.031
p5.2.z	1680	270	0.02
p5.3.a	**0**	0	0.007
p5.3.b	**15**	15	0.016
p5.3.c	**20**	20	0.010
p5.3.d	**60**	60	0.001

Problem	*Best-known solution*	*Best obtained solution*	*Time (Second)*
p5.3.e	**95**	80	0.01
p5.3.f	**110**	80	0.009
p5.3.g	**185**	110	0.031
p5.3.h	**260**	140	0.016
p5.3.i	**335**	160	0.032
p5.3.j	**470**	195	0.024
p5.3.k	**495**	210	0.019
p5.3.l	**595**	210	0.014
p5.3.m	**650**	210	0.014
p5.3.n	**755**	225	0.016
p5.3.o	**870**	225	0.019
p5.3.p	**990**	240	0.007
p5.3.q	1070	250	0.023
p5.3.r	1125	275	0.059
p5.3.s	1190	265	0.024
p5.3.t	1260	275	0.016
p5.3.u	1345	305	0.015
p5.3.v	1425	305	0.021
p5.3.w	1485	305	0.014
p5.3.x	1555	295	0.028
p5.3.y	1595	305	0.017
p5.3.z	1635	305	0.022
p5.4.a	**0**	0	0.405
p5.4.b	**0**	0	0.119
p5.4.c	**20**	20	0.019
p5.4.d	**20**	20	0.018
p5.4.e	**20**	20	0.01
p5.4.f	**80**	80	0.011
p5.4.g	**140**	110	0.049
p5.4.h	**140**	120	0.05
p5.4.i	**240**	145	0.082
p5.4.j	**340**	170	0.015
p5.4.k	**340**	180	0.01
p5.4.l	**430**	210	0.07
p5.4.m	**555**	230	0.013
p5.4.n	**620**	250	0.01
p5.4.o	**690**	260	0.009
p5.4.p	765	270	0.008
p5.4.q	**860**	270	0.011
p5.4.r	**960**	285	0.013
p5.4.s	1030	300	0.007
p5.4.t	**1160**	310	0.01
p5.4.u	**1300**	335	0.007
p5.4.v	**1320**	335	0.01
p5.4.w	1390	350	0.016
p5.4.x	1450	360	0.009
p5.4.y	1520	360	0.011
p5.4.z	1620	350	0.008
p6.2.a	**0**	0	0.008
p6.2.b	**0**	0	0.006
p6.2.c	**0**	0	0.009
p6.2.d	**192**	90	0.01
p6.2.e	**360**	114	0.007
p6.2.f	**588**	144	0.004
p6.2.g	**660**	138	0.044
p6.2.h	**780**	156	0.006
p6.2.i	**888**	168	0.008
p6.2.j	948	168	0.007
p6.2.k	1032	186	0.01
p6.2.l	1116	174	0.006
p6.2.m	1188	180	0.019
p6.2.n	1260	186	0.043
p6.3.a	**0**	0	0.015
p6.3.b	**0**	0	0.016
p6.3.c	**0**	0	0.047
p6.3.d	**0**	0	0.025
p6.3.e	**0**	0	0.024
p6.3.f	**0**	0	0.02
p6.3.g	**282**	126	0.043
p6.3.h	**444**	150	0.037
p6.3.i	**642**	168	0.027
p6.3.j	**828**	186	0.146
p6.3.k	894	210	0.013
p6.3.l	1002	210	0.012
p6.3.m	1080	222	0.042
p6.3.n	**1170**	210	0.019
p6.4.a	**0**	0	0.014
p6.4.b	**0**	0	0.02
p6.4.c	**0**	0	0.038
p6.4.d	**0**	0	0.012
p6.4.e	**0**	0	0.026
p6.4.f	**0**	0	0.019
p6.4.g	**0**	0	0.026
p6.4.h	**0**	0	0.025
p6.4.i	**0**	0	0.016
p6.4.j	**366**	162	0.025
p6.4.k	**528**	198	0.024
p6.4.l	696	204	0.015
p6.4.m	912	222	0.028
p6.4.n	**1068**	252	0.038
p7.2.a	**30**	30	0.014

Problem	*Best-known solution*	*Best obtained solution*	*Time (Second)*
p7.2.b	**64**	64	0.018
p7.2.c	**101**	91	0.013
p7.2.d	**190**	94	0.03
p7.2.e	**290**	100	0.161
p7.2.f	**387**	115	0.032
p7.2.g	459	112	0027
p7.2.h	521	137	0.05
p7.2.i	580	121	0.158
p7.2.j	646	127	0.034
p7.2.k	705	127	0.026
p7.2.l	767	143	0.023
p7.2.m	827	162	0.019
p7.2.n	888	153	0.022
p7.2.o	945	156	0.012
p7.2.p	1002	160	0.015
p7.2.q	1044	160	0.02
p7.2.r	1094	160	0.024
p7.2.s	1136	160	0.015
p7.2.t	1179	164	0.022
p7.3.a	**0**	0	0.026
p7.3.b	**46**	46	0.019
p7.3.c	**79**	79	0.026
p7.3.d	**117**	104	0.155
p7.3.e	**175**	128	0.025
p7.3.f	**247**	130	0.031
p7.3.g	**344**	137	0.023
p7.3.h	425	148	0.023
p7.3.i	**487**	165	0.018
p7.3.j	564	166	0.031
p7.3.k	633	169	0.024
p7.3.l	684	169	0.023
p7.3.m	762	163	0.022
p7.3.n	820	164	0.017
p7.3.o	874	164	0.036
p7.3.p	929	188	0.033
p7.3.q	987	186	0.038
p7.3.r	1026	185	0.025
p7.3.s	1081	197	0.038
p7.3.t	1120	206	0.017
p7.4.a	**0**	0	0.036
p7.4.b	**30**	30	0.021
p7.4.c	**46**	46	0.02
p7.4.d	**79**	79	0.038
p7.4.e	**123**	123	0.043
p7.4.f	**164**	144	0.016
p7.4.g	**217**	146	0.036
p7.4.h	**285**	158	0.033
p7.4.i	**366**	185	0.03
p7.4.j	**462**	191	0.032
p7.4.k	520	200	0.027
p7.4.l	590	202	0.121
p7.4.m	646	187	0.146
p7.4.n	730	177	0.018
p7.4.o	781	195	0.017
p7.4.p	846	209	0.022
p7.4.q	909	205	0.032
p7.4.r	970	204	0.013
p7.4.s	1022	213	0.022
p7.4.t	1077	215	0.025

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.

V. CONCLUSION

This paper, introduces a randomize heuristic to generate the initial population for the team orienteering problem. The obtained results were promising and promote the integration of our heuristic as a start for population-based metaheuristics. The use of permutations for the proposed heuristic avoid 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.

REFERENCES

[1] "The orienteering problem - Golden - 2006 - Naval Research Logistics (NRL) - Wiley Online Library." [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1002/1520-6750(198706)34:3%3C307::AID-NAV3220340302%3E3.0.CO;2-D/abstract.

[2] P. Vansteenwegen, W. Souffriau, and D. V. Oudheusden, "The orienteering problem: A survey," Eur. J. Oper. Res., vol. 209, no. 1, pp. 1–10, février 2011.

[3] I.-M. Chao, B. L. Golden, and E. A. Wasil, "The team orienteering problem," Eur. J. Oper. Res., vol. 88, no. 3, pp. 464 – 474, 1996.

[4] D.-C. Dang, R. N. Guibadj, and A. Moukrim, "An effective PSO-inspired algorithm for the team orienteering problem," Eur. J. Oper. Res., vol. 229, no. 2, pp. 332 – 344, 2013.

[5] C. Archetti, Á. Corberán, I. Plana, J. M. Sanchis, and M. G. Speranza, "A matheuristic for the Team Orienteering Arc Routing Problem," Eur. J. Oper. Res., vol. 245, no. 2, pp. 392 – 401, 2015.

[6] W. Souffriau, P. Vansteenwegen, G. V. Berghe, and D. V. Oudheusden, "A Path Relinking approach for the Team Orienteering Problem," Comput. Oper. Res., vol. 37, no. 11, pp. 1853 – 1859, 2010.

[7] H. Bouly, D.-C. Dang, and A. Moukrim, "A memetic algorithm for the team orienteering problem," 4OR, vol. 8, no. 1, pp. 49–70, 2009.

[8] S.-W. Lin, "Solving the team orienteering problem using effective multi-start simulated annealing," Appl. Soft Comput., vol. 13, no. 2, pp. 1064 – 1073, 2013.

[9] B.-I. Kim, H. Li, and A. L. Johnson, "An augmented large neighborhood search method for solving the team orienteering problem," Expert Syst. Appl., vol. 40, no. 8, pp. 3065 – 3072, 2013.