International Journal of Advances in Mathematics
Volume 2017, Number 6, Pages 10-19, 2017
eISSN 2456-6098
© adv-math.com

A Two-Phase Heuristic Method for Solving Team Orienteering Problem

Bochar Laarabi* and Bouchaib Radi[1]
*,1 FST Settat,Morocco.
E-mail:h.bochar@hotmail.fr

ABSTRACT. The Team Orienteering Problem (TOP) is an catchy variant of the Vehicle Routing Problem (VRP). The aim at maximizing the total amount of profit collected by a fleet of vehicles while not exceeding a predefined travel cost limit on each vehicle. In this paper,we propose a two-phase heuristic method to find solutions for TOP. In phase 1,We determine the initial solution using the k-means clustering by not exceeding the constraints of the total cost. In phase 2, an improvement heuristic search for a better solution based on the initial solution in phase 1 is developed. The obtained results show the performance of the new proposed methodology. We evaluated our algorithm on the standard benchmark of TOP. The results show that the algorithm is competitive and is able to prove the optimality for the instances previously unsolved.

1 Introduction

The Team Orienteering Problem (TOP) [1] is a widely studied Vehicle Routing Problem (VRP) which can be described as follows: a set of vehicles is available to visit customers from a potential set and each vehicle is associated with a fixed travel time and two particular depots, the so-called departure and arrival. Each customer is associated with an amount of profit that can be collected at most once by the set of vehicles. The aim of TOP is to select customers and organize an itinerary of visits so as to maximize the total amount of collected profits. The applications of TOP include athlete recruiting [1], technician routing [2,3] and tourist trip planning [4]. Several exact methods have been proposed to solve TOP, such as, Butt and Ryan [5] described a set covering formulation and developed a column generation algorithm, Boussier et al [6] propose a Branch-and-price algorithm and a dynamic

* Corresponding Author.
Received August 23, 2017; revised October 18, 2017; accepted October 20, 2017.
2010 Mathematics Subject Classification: 90B06, 68T20, 90-02 .
Key words and phrases: Team Orienteering Problem; algorithm k-means; heuristic.
This is an open access article under the CC BY license http://creativecommons.org/licenses/by/3.0/.

programming approach to deal with the pricing problem, and more recently, a pseudo-polynomial linear model for TOP was introduced by Poggi de Arag ao et al [7] and a branch-cut and price algorithm was proposed.
Most studies on TOP focus on designing good heuristic methods. Among these, the most successful approaches recently proposed include a variable neighbourhood search (Archetti, Hertz, and Speranza 2007)[10], a memetic algorithm (Bouly, Dang, and Moukrim 2010)[11], particle swarm optimisation method.
More recently, [8] proposed a branch-and-price approach where the pricing subproblem is solved by a bounded bidirectional dynamic programming algorithm with decremental state space relaxation featuring a two-phase dominance rule relaxation. The authors were able to solve 301 of 387 instances, by closing 17 previously unsolved instances. The most recent exact approach is due to El-Hajj et al. [9] who proposed the effective use of a linear formulation with a polynomial number of variables. Cutting planes are the core component of the algorithm that closed to optimality 12 previously unsolved benchmark instances.

In this paper, we proposed new method based on two algorithms to solve the TOP. The first is to create the initial solution by using the k-mean algorithm of a new metric in terms of gain and cost. The second is a new heuristic proposed to develop the initial solution to resolve TOP.
The paper is organized as follows. In Section 2, we describe the mathematical formulation used, and in Section 3, Basic concepts and Principles,in section 4 we present two phase heuristic method. Computational results are presented in Section 5.

2 Problem statement

TOP is modeled with a complete graph G=(V,E), which $V = (\{1, ..., n\} \cup \{d, a\})$ the set of vertices representing customers and depots, and $E = \{(i, j)|i, j \in V\ i \neq j\}$ is the set of arcs. Vertices d and a are respectively the departure and arrival depot for the vehicles. For convenience, we use the three sets $\overline{V}$,$V^d$and V^a to denote respectively the sets of the customers only, of the customers with the departure depot and of the customers with the arrival one. Each vertex i is associated with a profit P_i and the travel cost c_{ij} is associated with each arc (i,j)∈E . The travel costs are assumed to satisfy the triangle inequality. A feet F is composed of m identical vehicles and available to visit customers. Each vehicle must start its route from d , visit a certain number of customers and return to a without exceeding its predefined travel cost C_{max}. The problem can be then formulated in Mixed Integer Programming (MIP) using a polynomial number of decision variables x_{ijr} and y_{ir} : x_{ijr} =1 if arc (i,j)is used by vehicle r to serve customer i then customer j and 0 otherwise y_{ir} = 1 if client i is served by vehicle r and 0:

$$\max \sum_{i \in \overline{V}} \sum_{r \in F} y_{ir} p_i \tag{2.1}$$

$$\sum_{r \in F} y_{ir} \leq 1 \quad \forall i \in \overline{V} \tag{2.2}$$

$$\sum_{j \in V^d} x_{jar} = \sum_{j \in V^a} x_{djr} = 1 \tag{2.3}$$

$$\sum_{i \neq k} x_{kir} = \sum_{j \neq k} x_{kjr} = y_{kr} \quad \forall r \in F, \forall k \in \overline{V} \tag{2.4}$$

$$\sum_{i \in V^d} \sum_{j \in V^a \setminus \{i\}} C_{ij} x_{ijr} \leq C_{max} \quad \forall r \in F \tag{2.5}$$

$$x_{ijr} \in \{0,1\} \quad \forall i \in V, \forall j \in V, \forall r \in F \tag{2.6}$$

$$y_{ir} \in \{0,1\} \quad \forall i \in \overline{V}, \forall r \in F \tag{2.7}$$

The objective function (2.1) is to maximize the sum of the collected profits. Constraints (2.2) guarantee that each customer is visited at most once. The connectivity of each tour is ensured by constraints (2.3) and (2.4). Constraints (2.5) describe the travel length restriction. Finally, constraints (2.6) and (2.7) sets the integral requirement on the variables.

3 Basic concepts and Principles

I*k*-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) priorly fixed. The main idea is to define k centers one for each cluster. These centers should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest center. When no point is pending, the first step is completed and the early group is done. At this point we need to re-calculate k new centroids as barycenter of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new center.

As a result of this clustring we notice that the k centers change their location step by step until no more changes are done or in other words centers do not move any more.

The instructions of the K means algorithm are presented as follow:

Algorithm 1 General scheme of the $k-$means algorithm

Step 1: Select k $\in$ O q_r centers for each class, however, with r $\in \{1, ..., K\}$.

Step 2: Assign each object o_i O to the class O_r that his qr center is closest to where o_i generating a partition P, P′= $\emptyset$.

Repeat steps 3 through 4 as long as P≠ P′

Step 3: If P =∅ make P = P′. Calculating the centers of each class O_r.

Step 4: Reassign new each object o_i the O to a class O_r, q_r center is closest to where o_i generating a new partition P′

We used *k*-means algorithm to generate classes vertices for the proposed method. In the following sections we detail the algorithm *k*-means adopted to the needs of the classification phase.

4 Basic concepts and Principles

The objective of this algorithm is to define a specific classification for TOP. This classification has performed on n vertices of a graph G characterized by p_i gains and Euclidean distances d_{ij}. For this classification, we fixed the k classes interfere at two deposits, which the compactness of a class must take account of distances and earnings. Moreover, we specified the initial choice of the nodes, the distance between the nodes, and class centers calculation for the proposed classification.

For obtaining a solution for TOP taking inventory control decisions into consideration, a two-phase heuristic method is proposed in this paper. Phase 1 finds the initial solution using k-mean clustring by not exceeding the constraints of the total cost. Phase 2 develops an improvement heuristic search for a better solution based on the initial solution in phase 1. The detailed procedure for this method is as follows :

Phase 1:finding the initial solution using the algorithm 2

Algorithm 2 steps of phase 1

Step 1: Select k distinct vertices $q_r \in$ V ∪ $\{a, d\}$ for each class, however, with Q = $\{q_1, .., q_k\}$.

Step 2: Assign each vertex a,d in class O_r permanently.

Step 3: Assign each vertex v_i of V-(Q ∪$\{a, d\}$) to the class O_r Don't q_r its center is the closest to v_i in terms of distance and gain, hence generating a partition P, P′=∅ Repeat steps 4-6 as P ≠ P′

Step 4: If P ≠ ∅ make P = P′.

Step 5: Update all Q centers

Step 6: Update Now class taking account of the nodes (Q ∪$\{a, d\}$), consequently generating a new partition P′.

---

phase 2:improving the initial solution

After clustering, We define for each cluster a new problem consists of a determining simple circuit of maximal total profit, and whose length does not exceed a preset constraints. For to solve it one uses the following algorithm 3:

Algorithm 3 proposed algorithm for phase 2

$I = d$

$r \leftarrow 1$

$m \leftarrow 1$

while $F = (V_k \notin I, C_{sk} + C_{kr} \leq \alpha_r C_{max})\} \neq \emptyset$ **do**

find v_i and v_j such that

$\frac{p_i}{c_{ir}}$ = max($\frac{p_k}{c_{kr}}$) and $\frac{p_j}{c_{js}}$ =max($\frac{p_k}{c_{kr}}$)

if $\frac{p_i}{c_{ir}} \geq \frac{p_j}{c_{js}}$ **then**

add v_i to I before r and set $r = i$

else

add v_j to I after s and set $s = j$

else if

close the current path by adding arc (v_n,v_m)

end while

This algorithm presents a new approach to solve the problem associated with phase 2, which is concerned with finding a path between given set of control points, among which a start and end point are specified, so as to maximize the total gain collected subject to a prescribed cost constraint. This approach based on maximization of $\frac{gain}{cost}$ ratio that is gradually constructs a path I by adding, at each iteration, an arc according to respected criterion.

5 Computational Results

The proposed algorithms described in above section were implemented on a core i7 processor and tested using Matlab and R software. The following figure summarized this implementation.

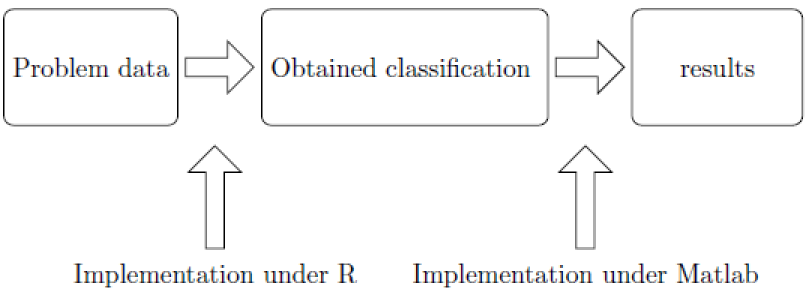

Figure 1: Scheme of our implementation.

Firstly, we implemented the K-means algorithm parameters using software R. Secondly, we used the obtained data to resolve the problem related to phase 2 under Matlab software.
In term to evaluate our method we use the benchmark instances proposed by Chao et al. (1996). This benchmark comprises 387 instances and divided into 7 data sets.
In each data set, the positions and the profits of the customers are identical for all instances. However, the number of vehicles varies from 2 to 4 and the cost length limit C_{max} is also different between instances. Table 1 reports the number of instances in the set (#), the number of customers in each instance n, the range values for the number of available vehicles m, and for the maximum cost C_{max}.

Table 1: Instances of Chao et al. (1996).

sets	# Inst	n	m	C_{max}
1	54	30	2-4	3.8-22.5
2	33	19	2-4	1.2-42.5
3	60	31	2-4	3.8-55
4	60	98	2-4	3.8-40
5	78	64	2-4	1.2-6
6	42	62	2-4	5-200
7	60	100	2-4	12.5-120

The results obtained give us the notice from applying all the proposed components together remarkably improves the number of instances being solved, reaching 292 of the 387 instances.

Table 2: Instances Solved by our method.

sets	m	#	# Inst solved	# Inst unsolved
1	2	18	18	0
1	3	18	18	0
1	4	18	18	0
2	2	11	11	0
2	3	11	11	0
2	4	11	11	0
3	2	20	20	0
3	3	20	19	1
3	4	20	12	8
4	2	20	5	15
4	3	20	11	9
4	4	20	9	11
5	2	26	22	4
5	3	26	21	5
5	4	26	15	11
6	2	14	14	0
6	3	14	14	0
6	4	14	12	2
7	2	20	11	9
7	3	20	9	11
7	4	20	11	9
Total	-	387	292	95

Our method shows a quite good results for sets 1,2,3 and 6. It was able to solve most of the instances in the set 5. We had only few difficulties with the sets 4,7 and more precisely on some instances with 4 available vehicles. Overall, our approach is clearly efficient and competitive with the existing methods in the literature.

These results confirm that our approach produces good quality solutions inding a number of solutions not already known in very short calculation times.

Table 3: Comparison with other exact methods.

set	Dang et al. (2013)	Boussier et al(2007)	our method
1	54/54	51/54	54/54
2	33/33	33/33	33/33
3	60/60	50/60	51/60
4	22/60	25/60	25/60
5	44/78	48/78	58/78
6	42/42	36/42	40/42
7	23/60	27/60	31/60
Total	278/387	270/387	292/387

Table 3 summarizes the results of the comparison between our method and the other state-of-the-art exact algorithms. For our algorithm we report, in addition to the number of solved instances. In terms of total number of instances solved, 14 instances more than Dang et al.(2013).

Table 4: Optimal solutions that remain unresolved by Dang et al. (2013) and Boussier et al (2007) .

Instance	N	m	C_{max}	UB	LB
p5.2.n	66	2	35	940	940
p5.2.o	66	2	37.5	1040	1040
p5.2.r	66	2	45	1280	1280
p5.2.s	66	2	47.5	1360	1360
p5.2.u	66	2	52.5	1480	1480
p5.2.v	66	2	55	1530	1530
p5.3.u	66	3	35	1400	1400
p5.3.t	66	3	33.3	1280	1280
p5.4.q	66	4	28.3	1100	1100
p5.4.z	66	4	32.5	1660	1620
p7.2.h	102	2	80	440	425
p7.3.j	102	3	66.7	577.5	572.9
p7.3.k	102	3	73.3	640	640
p7.4.o	102	4	75	790	790

In table 4,we list the instances solved to optimality for the first time by our method. Columns labeled with Instance, N, m, and C_{max} indicate the name of the instance solved, the number of customers, the number of vehicles, and the Columns UB,LB, report respectively the upper bound, lower bound.

Moreover, we can notice from Table 3 that our method was able to improve the upper bounds of respectively 32 and 27 instances more than the two algorithms of Keshtkarana et al. (2016). Overall, our approach is clearly efficient and competitive with the existing methods in the literature. We were able to prove the optimality of 14 instances than more Dang et al. (2013) and Boussier et al(2007. These instances are marked in Table 3.

6 conclusion

In this paper we focused to solve Team Orienteering Problem based two-phase heuristic method. Two algorithms has been proposed to create this goal, the first one is k-mean setted algorithm, the second is a new created algorithm proposed to resolve initial solution. The experiments were conducted on the standard benchmark of TOP to confirm the effectiveness of our approach. The obtained results show that our method is able to solve a large number and variety of benchmark instances.

References

[1] I.M. Chao B. Golden;E. Wasil. The team orienteering problem. *European Journal of Operational Research,88(1996)*, 464-474.

[2] H. Bouly; A. Moukrim; D. Chanteur;L. Simon. Un algorithme de destruction /construction iteratif pour la resolution d'un probleme de tournees de vehicules specique. *In MOSIM'08*, 2008.

[3] H. Tang ; E. Miller-Hooks. A tabu search heuristic for the team orienteering problem. *European Journal of Operational Research,32(2005)*, 1379-1407.

[4] P. Vansteenwegen; W. Souriau; G. Vanden Berghe; D. Van Oudheusden. Metaheuristics for tourist trip planning. *n Metaheuristics in the Service Industry , 624(2009)*, 15-31.

[5] S. E. Butt ; D. M. Ryan. An optimal solution procedure for the multiple tour maximum collection problem using column generation. *computers Operations Research*, *26(1999)*, 427-441.

[6] S. Boussier; D. Feillet; M. Gendreau. An exact algorithm for team orienteering problems. *4OR,5(3)(2007)*, 211-230.

[7] S.M. Poggi; H. Viana; E. Uchoa. team orienteering problem: Formulations and branch-cut and price. *In ATMOS*, *5(3)(2010)*, 142-155.

[8] Keshtkaran, M., Ziarati, K., Bettinelli, A., Vigo, D.,. Enhanced exact solution methods for the team orienteering problem. *International Journal of Production Research*, *54(2016)*,591-601.

[9] El-Hajj, R., Dang, D.-C., Moukrim, A.,. Solving the team orienteering problem with cutting planes. *Computers Operations Research*, *74(2016)*, 21-30.

[10] Archetti,Claudia,Alain Hertz,and Maria Grazia Speranza. Metaheuristics for the Team Orienteering Problem. *Journal of Heuristics*, *13 (1)(2007)*, 49-76.

[11] Bouly, Hermann, Duc Cuong Dang, and Aziz Moukrim. A Memetic Algorithm for the Team Orienteering Problem. *4OR*, *8(1)(2010)*, 49-70.