The Single Depot Multiple Set Orienteering Problem

Ravi Kant, Abhishek Mishra and Siddharth Sharma
Department of CSIS, Birla Institute of Technology and Science, Pilani, 333031, India

Keywords: The Orienteering Problem, The Set Orienteering Problem.

Abstract: In this article, we present the single Depot multiple Set Orienteering Problem (sDmSOP), a new variant of the classical Set Orienteering Problem (SOP). A significant feature of sDmSOP is the presence of many travelers who set off from the same depot and return there at the end of their journey. The objective of the problem is to maximize the profit while remaining within the budget; hence the challenge at hand involves searching multiple paths among the mutually clustered sets for travelers. A set's profit can be collected with a single node visit only. Supply-chain management, the bus delivery problem, etc., are just a few examples where the sDmSOP has proven useful. By simulating the instances of the Generalized Traveling Salesman Problem (GTSP) using GAMS 37.1.0, we determine the optimal profit for GTSP instances for some small and medium instances which follow the triangular and symmetric properties. We find that the use of multiple travelers is beneficial for both service providers and customers, as it allows service providers to offer their services to customers at a lower cost because the service provider gets a significant amount of profit using multiple travelers.

1 INTRODUCTION

Due to their practical implications, NP-hard routing problems with profits have gained interest. Specifically, (Archetti and Speranza, 2015) studied arc routing problems, whereas (Archetti et al., 2014) and (Gunawan et al., 2016) focused on node routing problems. Profit in Arc routing problems is associated with arcs, while in node routing problems, it is associated with nodes. Since the node routing problem has real-world applications in supply chain management, the bus delivery problem, and smart city waste management, it is more important to research. (Golden et al., 1987) proposed the Orienteering Problem (OP), which is now widely known as a classical example of a node routing problem. The goal of the OP is to maximize profit by making as many node visits as possible utilizing a fixed budget and a single traveler. There is a profit to be earned from each node in the OP, but it can only be earned once.

A new variant of the OP, termed the Set Orienteering Problem (SOP), was recently suggested by (Archetti et al., 2018), in which the nodes are grouped into mutually exclusive clusters, and the profit associated with a cluster can only be achieved if at least one node is visited by the traveler. The problem is based on the revenue generated by clusters as opposed to individual nodes. (Archetti et al., 2018), (Pěnička et al., 2019), and (Carrabs, 2021) proposed the solutions for the SOP by combining the Lin-Kernighan heuristic given by (Lin and Kernighan, 1973) and the Tabu-search meta-heuristic given by (Glover and Laguna, 1998), the Variable Neighbourhood Search, and the Biased random-key genetic algorithm (BRKGA), respectively. It is shown that BRKGA achieves better results in terms of time.

This article focuses on a generalized version of the Set Orienteering Problem, which we refer to as the single Depot multiple Set Orienteering Problem (sDmSOP), in which customers (vertices) are grouped in mutually disjoint sets, and the profit is associated with each set. In sDmSOP, multiple travelers begin and end their journey at a fixed depot in order to collect the maximum profit by visiting as many sets as possible exactly once within the given budget. While the research into the SOP is significant because of the practical applications proposed by the aforementioned authors, many real-world situations cannot be simulated with a single traveler, such as the well-known bus delivery problem and reliable supply where more than one distributor is required. It was discovered that the SOP could be applied to the supply chain by grouping customers from different chains together; however, in sDmSOP, we must also account for the situation in which more than one traveler is required to provide the services to the supply

Kant, R., Mishra, A. and Sharma, S.
The Single Depot Multiple Set Orienteering Problem.
DOI: 10.5220/0011681800003396
In *Proceedings of the 12th International Conference on Operations Research and Enterprise Systems (ICORES 2023)*, pages 175-179
ISBN: 978-989-758-627-9; ISSN: 2184-4372

chain, and distributors can maximize their profits by offering these services to the chains that are most convenient for their customers to access.

The sDmSOP also has potential use in the well-known school bus problem, in which the individuals typically travel in clusters. Through the use of sDmSOP, the best route to pick up as many passengers as possible from a given starting place can be determined.

This paper is categorized as follows: In section 2, We define and give a mathematical formulation of the sDmSOP, the comparative results are discussed in section 3, and section 4 contains the conclusion of the paper.

2 PROBLEM DEFINITION

The sDmSOP is a generalization of SOP (Set Orienteering problem), so first, we give a formal definition of SOP.

The SOP can be formalized on an undirected complete graph $G(V,E)$ where $V = \{v_1, ..., v_n\}$ is the set of vertices and $E = \{e_{ij}\}$ is the set of edges, e_{ij} defined as an edge between vertices v_i and v_j, moreover a cost $c_{ij} \geq 0$, is associated with each edge. The vertices are partitioned into disjoint sets $S = \{s_1, ..., s_q\}$ such that their union contains all vertices of the graph. The objective of the problem is to gain maximum profit by visiting the possible number of sets within a distance constraint B with the predefined starting and ending depot. The profit from a set can be collected if only one vertex of a set is visited by a traveler.

In this paper, we propose the single Depot multiple Set Orienteering Problem (sDmSOP), a general variant of the SOP.

2.1 Mathematical Formulation

To represent an integer linear programming formulation for sDmSOP, we use some notations, which are given as follows:

- Let v represent vertices and s represent sets
- $t : \{1, ..., m\}$ (the different salesman).
- $i, j : \{1, ..., n\}$ (the list of vertices).
- Assume the depot at vertex 1.
- Let c_{ij} represent the edge weights.
- Let P_q represent the profit associated with a set S_q.
- Let B represent a Budget.

We can then construct an Integer Linear Program (ILP) formulation using the decision variables:

- x_{tij} : 1 if traveler t uses the edge $(i, j) \in E$ and 0 otherwise.
- y_{ti} : 1 if vertex i is visited by traveler t and 0 otherwise.
- z_{tq} : 1 if any vertex in set q is visited by traveler t and 0 otherwise.
- u_{ij} : flow variable for the Sub-tour Elimination Constraints (SECs).

The proposed mathematical formulation of sDmSOP:

$$maximize \sum_t \sum_q P_q z_{tq}, \tag{1}$$

subject to:

$$x_{tij}, y_{ti}, z_{tq} \in \{0, 1\}, \tag{2}$$

$$\sum_t \sum_i \sum_j x_{tij} c_{ij} \leq B, \tag{3}$$

$$\sum_t \sum_j x_{t1j} = m = \sum_t \sum_j x_{tj1}, \quad \forall t, \forall j, \tag{4}$$

$$\sum_{v_i \in V - \{v_j\}} x_{tij} = y_{tj} \quad \forall t, \forall j, \tag{5}$$

$$\sum_{v_i \in V - \{v_j\}} x_{tji} = y_{tj} \quad \forall t, \forall j, \tag{6}$$

$$\sum_{v_i \in S_q} y_{ti} = z_{tq} \quad \forall t, \forall q, \tag{7}$$

$$\sum_t z_{tq} \leq 1 \quad \forall q, \tag{8}$$

$$0 \leq u_{ij} \leq (n - m) \sum_{t=1}^{m} x_{tij}, \quad \forall t, \forall i, \tag{9}$$

$$\textstyle\sum_{j \in V} u_{ij} - \sum_{j \in V-\{1\}} u_{ji} = \sum_{t=1}^{m} y_{ti}, \quad \forall t, \forall i \,|\, i \neq j, i \in V - \{1\}. \tag{10}$$

The objective function (1) maximizes collected profits from the sets visited, constraints (2) define the domain of the variables x_{tij}, y_{ti}, and z_{tq}. Constraint (3) ensures that budget B is not exceeded. Constraint (4) ensures that exactly m travelers start and end at depot 1. Equations (5) and (6) imply that the in-degree is equal to the out-degree of a vertex except for the depot. Constraint (7) ensures that a set S is visited by a traveler t if any vertex in the set is visited and at most one vertex can be visited per set. Constraint (8) implies that no set can be visited by more than one travelers while equations (9) and (10) are used to remove the Sub-tours in the path. The Sub-tour Elimination Constraints (SECs) are based on the Traveling Salesman Problem (TSP) model proposed by (Gavish and

Table 1: Comparison with optimal solutions on small instances with $w < 0.5$

Instance	n	t	Pg	w	Opt.	ILP		
						Sol.	Time	Gap (%)
11berlin52	52	2	g1	0.2	37	37	28.875	0.00
11berlin52	52	2	g1	0.3	43	43	51.046	0.00
11berlin52	52	2	g1	0.4	47	47	36.172	0.00
11berlin52	52	2	g2	0.2	1729	1729	36.188	0.00
11berlin52	52	2	g2	0.3	2090	2090	105.047	0.00
11berlin52	52	2	g2	0.4	2284	2284	96.828	0.00
11berlin52	52	3	g1	0.2	43	43	50.094	0.00
11berlin52	52	3	g1	0.3	48	48	8452.922	0.00
11berlin52	52	3	g1	0.4	51	51	1.578	0.00
11berlin52	52	3	g2	0.2	2090	2090	117.297	0.00
11berlin52	52	3	g2	0.3	2338	2338	627.5	0.00
11berlin52	52	3	g2	0.4	2508	2508	6.516	0.00
11eil51	51	2	g1	0.2	28	28	34.313	0.00
11eil51	51	2	g1	0.3	39	39	85.703	0.00
11eil51	51	2	g1	0.4	46	46	433.641	0.00
11eil51	51	2	g2	0.2	1376	1376	29.797	0.00
11eil51	51	2	g2	0.3	1911	1911	97.297	0.00
11eil51	51	2	g2	0.4	2272	2272	238.563	0.00
11eil51	51	3	g1	0.2	39	39	85.25	0.00
11eil51	51	3	g1	0.3	48	48	203.828	0.00
11eil51	51	3	g1	0.4	50	50	5.437	0.00
11eil51	51	3	g2	0.2	1911	1911	98.5	0.00
11eil51	51	3	g2	0.3	2421	2421	213.594	0.00
11eil51	51	3	g2	0.4	2475	2475	2.187	0.00
16eil76	76	2	g1	0.2	40	40	1909.547	0.00
16eil76	76	2	g1	0.3	59	59	3125.797	0.00
16eil76	76	2	g1	0.4	70	70	36372.547	0.00
16eil76	76	2	g2	0.2	2144	2144	10217.375	0.00
16eil76	76	2	g2	0.3	3090	3090	3397.734	0.00
16eil76	76	2	g2	0.4	3550	3550	43317.578	0.00
16eil76	76	3	g1	0.2	59	59	3089.171	0.00
16eil76	76	3	g1	0.3	72	72	1916444.594	0.00
16eil76	76	3	g1	0.4	75	75	8.468	0.00
16eil76	76	3	g2	0.2	3090	3090	3318.11	0.00
16eil76	76	3	g2	0.3	3632	3632	383533.094	0.00
16eil76	76	3	g2	0.4	3700	3700	52.172	0.00

Graves, 1978) and assessed for the Asymmetric Traveling Salesman Problem by (Öncan et al., 2009). In the above ILP formulation equations (1)-(10) attempt to find out the optimal path with maximization of the profit using permutation of the sets and the vertices which are to be visited in the specific set.

3 COMPARATIVE RESULTS

In this section, we simulate the mathematical model using GAMS 37.1.0 on GTSP instances and give the performance of comparative results using multiple travelers. The simulation is done on the windows 10 platform with an i7-6400 CPU @3.4Ghz processor with 32GB of RAM.

In section 3.1, we describe how the instances are generated for sDmSOP, and the simulation results are shown in section 3.2.

3.1 Test Instances

To analyze the comparative results of the above formulation, the Generalized Traveling Salesman Problem (GTSP) instances suggested by (Noon, 1988) are used. The branch and cut method proposed by (Fis-

Table 2: Comparison with optimal solutions on medium instances with $w < 0.5$

Instance	n	t	Pg	w	Best Possible Sol.	ILP		
						Sol.	Time	Gap (%)
21lin105	105	2	g1	0.2	82.817211	49	7200.828	40.83
21lin105	105	2	g1	0.4	104	84	7204.031	19.23
21lin105	105	3	g1	0.2	104	104	306.531	0.00
21lin105	105	3	g1	0.4	93	75	7200.812	19.35
30ch150	150	2	g1	0.2	94.106272	43	7201.218	54.3
30ch150	150	2	g1	0.4	139	110	7201.36	26.16
30ch150	150	3	g1	0.2	125.32	80	7201.687	36.16
30ch150	150	3	g1	0.4	149	140	7202.359	6.04
40d198	198	2	g1	0.2	162.32	50	7202.766	69.19
40d198	198	2	g1	0.4	197	169	7203.609	14.21
40d198	198	3	g1	0.2	197	130	7201.719	34.01
40d198	198	3	g1	0.4	19	171	7206.047	13.19

chetti et al., 1997) for the Symmetric Generalized Traveling Salesman Problem was used for executing the formulation. We modified the GTSP for the single depots for the need of our formulation as follows:

1. Transfer the depot vertex from the non-depot set to the depot set.
2. Sort the non-depot sets in ascending order of the number of vertices in the set.
3. Iterate over the list, and if there is an empty set, find the first vertex from a non-empty set with a size greater than one and put it into the empty set found.

This algorithm generates sets that satisfy the constraints of our problem.

The profit is assigned to the sets using g_1 and g_2 schemes used by (Pěnička et al., 2019). In the g_1 scheme, the profit of a set is equal to the number of vertices. Whereas in the g_2 scheme for each vertex numbered k, the profit assigned is $(1+714 \times k)$mod 100, and the profit assigned to a set is the sum of profits of the vertices in the set. In each case, the depot set is assigned a profit of 0.

3.2 Computational Results

It is not feasible to solve the large instances of GTSP using GAMS 37.1.0, so we put the following criteria for Table 1 and Table 2 for the simulation:

1. Table 1: threads = 1 on small GTSP instances
2. Table 2: threads = 1 with two hours of CPU time on large instances.

here threads=1 means the system is using only one thread to solve the instance. Results are presented in Table 1 and Table 2. The organization of Table 1 and Table 2 are as follows: The first five columns represent the GTSP instance name, number of vertices (n), number of travelers (t), the rule to generate the profit (P_g) and the value of w. Budget (B) is calculated as $w \times t \times T_{\max}$, where $T_{\max}$ is the solution of the GTSP instance, Opt. column represents the best possible solution given by GAMS, and the last three columns represent the solution, time, and relative gap for ILP respectively. There is no CPU time restriction for Table 1; we run all the instances of table 1 till we find the optimal solution, while each instance of Table 2 is run for two hours because if we try to solve the large instances till we find the optimal solution, GAMS does run out of memory. Only one instance of $21lin105$ is solved optimally with three travelers by GAMS in 306.531 seconds. The results presented in Table 1 and Table 2 show that the service provider gets the better profit for multiple travelers for the same instance.

4 CONCLUSION

In this paper, we introduced a new variant of the Set Orienteering Problem, which has a single depot and multiple travelers with all the travelers associated with that particular depot; the objective of this problem is to gain maximum profit out of mutually exclusive sets by using multiple travelers with a fixed starting and ending depot within a fixed budget B. A fixed profit is associated with each cluster, calculated using two rules named g_1 and g_2, and the profit can only be gained if any traveler visits exactly one node of a set. The sDmSOP is an extension of the single depot multiple traveling salesman problem; it has an application in the supply chain, where a distributor has one service point from which the distributor can supply the products to the retailers and can gain the maximum profit out of it while giving a better price of the product to the retailers within a given budget.

REFERENCES

Archetti, C., Carrabs, F., and Cerulli, R. (2018). The set orienteering problem. *European Journal of Operational Research*, 267(1):264–272.

Archetti, C. and Speranza, M. G. (2015). Chapter 12: Arc routing problems with profits. In *Arc routing: problems, methods, and applications*, pages 281–299. SIAM.

Archetti, C., Speranza, M. G., and Vigo, D. (2014). Chapter 10: Vehicle routing problems with profits. In *Vehicle routing: Problems, methods, and applications, second edition*, pages 273–297. SIAM.

Carrabs, F. (2021). A biased random-key genetic algorithm for the set orienteering problem. *European Journal of Operational Research*, 292(3):830–854.

Fischetti, M., Salazar González, J. J., and Toth, P. (1997). A branch-and-cut algorithm for the symmetric generalized traveling salesman problem. *Operations Research*, 45(3):378–394.

Gavish, B. and Graves, S. C. (1978). The travelling salesman problem and related problems.

Glover, F. and Laguna, M. (1998). Tabu search. In *Handbook of combinatorial optimization*, pages 2093–2229. Springer.

Golden, B. L., Levy, L., and Vohra, R. (1987). The orienteering problem. *Naval Research Logistics (NRL)*, 34(3):307–318.

Gunawan, A., Lau, H. C., and Vansteenwegen, P. (2016). Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, 255(2):315–332.

Lin, S. and Kernighan, B. W. (1973). An effective heuristic algorithm for the traveling-salesman problem. *Operations research*, 21(2):498–516.

Noon, C. E. (1988). *The generalized traveling salesman problem*. PhD thesis, University of Michigan.

Öncan, T., Altınel, I. K., and Laporte, G. (2009). A comparative analysis of several asymmetric traveling salesman problem formulations. *Computers & Operations Research*, 36(3):637–654.

Pěnička, R., Faigl, J., and Saska, M. (2019). Variable neighborhood search for the set orienteering problem and its application to other orienteering problem variants. *European Journal of Operational Research*, 276(3):816–825.