The Orienteering Problem: A Review of Variants and Solution Approaches

Ravi KANT
p20190020@pilani.bits-pilani.ac.in
Department of CSIS, Birla Institute of Technology and Science,
Pilani, 333031, India

Abhishek MISHRA
abhishek.mishra@pilani.bits-pilani.ac.in
Department of CSIS, Birla Institute of Technology and Science,
Pilani, 333031, India

ABSTRACT

Orienteering Problem (OP) fetched great attention in recent years because apart from the NP-hard routing problems, it is applicable in various applications like mobile crowd-sensing, manufacturing, etc. OP intends to maximize the overall price collected from the places covered in the itinerary within a time-bound. In this paper, the latest improvements in NP-hard routing problems are discussed. Some variations of the traveling salesman problem (TSP), OP, and their recent solutions based on nature-inspired algorithms are explored. Finally, we present the future scope of the OP and its variants.

Keywords: Traveling Salesman Problem (TSP), Orienteering Problem (OP), Set Orienteering Problem (SOP).

1. INTRODUCTION

Research on solving NP-hard problems has a long tradition. To find the solution of TSP [17] is a great deal among NP-hard problems. Consequently, the field of optimization problems for the TSP has been thoroughly explored. In the TSP, a complete weighted graph and a start vertex are given. The task is to find an itinerary starting and stopping at the given vertex with the minimum weight (the sum of edge-weights in the path). The TSP has applications in several fields, including logistics, planning, and microchip manufacturing. However, the stated formulation of the TSP forces us to visit every vertex, which may not always be required in other variants of TSP with the time-bound. OP is one of the most studied variants of TSP with some time constraints.

The OP [21] is a mixture of the TSP and knapsack problem [38]. The essence of the OP is to introduce a budget to the TSP and pick the vertices which maximize the profit. Numerous variants of the OP have been proposed for application in routing and networking problems like solving the best route for a tourist, mobile crowd-sensing, etc.

2. BACKGROUND

A lot of exertion is already done in solving the NP-hard routing problems mainly related to TSP and OP using different Graph-Based, Nature-Inspired based approaches, etc. Researchers are still searching for an efficient solution of TSP, OP, and their variants so that the best approximation time algorithm can be designed for these types of routing problems. Recently, some of the latest variants and their nature-inspired solution approaches have been suggested by researchers. A brief introduction is presented in sections 2 and 3.

Problem Description
The OP [21] aims to find a minimum-cost itinerary among the selected nodes from n nodes within a constraint; the constraint can differ according to the problem definition. i.e., time, energy consumption, etc. Fundamentally, it is a mixture of TSP and knapsack problems.
Mathematical Formulation of OP can be described on a complete graph $H = (A, B)$ where $A = \{1,2,3\ldots, n\}$ is the number of nodes, and B is the number of edges in the complete graph. Node 1 and node n is the origin and ending depots for the traveler respectively, where ϕ_x is the profit associated with node x and ψ_{xy} is 1 if the edge between node x and node y is selected; otherwise, it is 0. The order of the corresponding node i is defined by the node potential u_i (u_i is a positive integer variable, and the node potential of origin depot is 1) and T_{xy} defines the time taken on the edge between nodes x and y. Integer linear programming formulation [25] of OP can be given as:

$$Maximize \sum_{x=2}^{n-1}\sum_{y=2}^{n} \phi_x \, \psi_{xy} \qquad Eq.(1)$$

Subject to:

$$\sum_{x=2}^{n} \psi_{1x} = \sum_{x=1}^{n-1} \psi_{xn} = 1 \qquad Eq.(2)$$

$$\sum_{x=1}^{n-1} \psi_{xk} = \sum_{y=2}^{n} \psi_{ky} \leq 1 \quad \forall k = 2, \ldots, n-1 \qquad Eq.(3)$$

$$\sum_{x=1}^{n-1}\sum_{y=2}^{n} T_{xy} \, \psi_{xy} \leq T_{Max} \qquad Eq.(4)$$

Sub-tour Elimination Constraints:

$$2 \leq u_x \leq n \qquad \forall x = 2, \ldots, n \qquad Eq.(5)$$

$$u_x - u_y + 1 \leq (n-1)\left(1 - \psi_{xy}\right) \quad s.t.\ x \neq y,\ \forall x, y = 2, \ldots, n \qquad Eq.(6)$$

$$\psi_{xy} \in \{0,1\} \quad \forall x, y = 1, \ldots, n \qquad Eq.(7)$$

ISBN: 978-1-950492-64-0
ISSN: 2771-0947

https://doi.org/10.54808/WMSCI2022.01.41

In the above formulation, equations (2) ensures that travelers start and stop at the given depots, while equation (3) ensures that each node may not be visited more than once and it determines the flow bound also, and equation (4) ensures that the itinerary should be finished with the predefined time-bound (T_{Max}). Constraints (5) and (6) [31] determine that the traveler covers no sub-tour. Equation (7) enforces Integer constraints.

Some Variants of TSP and OP

The variations of TSP and OP has many real-life applications like mobile crowdsensing, arc routing problem, and tour trip design. In this section, we discuss some basic and some recent variants of TSP and OP to grasp the basic understanding of the objectives of these variants.

Multi Travelling Salesman Problem (mTSP): The mTSP [19] is a generalization of a famous TSP problem. In mTSP, a set of N nodes and distance between each pair of nodes are given. The objective is to find out m different paths for m different travelers (one path for each traveler). Still, there are some conditions we have to follow, like the itinerary should start and stop at the same vertex, each in-between node should not be covered more than once, and the overall cost should be minimum.

Multi Depot Multiple Travelling Salesman Problem (MmTSP): The MmTSP [6] is a generalization of the mTSP problem. N vertices and distance between each pair of vertices are given as input. Still, the constraints are: we have to find a minimum cost itinerary for each traveler who starts from a particular assigned depot and ends the itinerary at the same depot. Fixed destination Multi Depot Multiple Travelling Salesman Problem (FD-MmTSP) [4, 6], Non-Fixed destination Multi Depot Multiple Travelling Salesman Problem (NFD-MmTSP) [4, 6], and Open Path Multi Depot Multiple Travelling Salesman Problem (OP-MmTSP) [4, 6] are some of the other variants of MmTSP that follow all the constraints of MmTSP. Still, there are some liberties in each of the variants, like, in FD-MmTSP, each traveler has to go back to the same depot after completing the itinerary from which they started. In NFD-MmTSP, the traveler can return to any depot after finishing the tour. In OP-MmTSP, the traveler is not bound to return to any depot after completing the itinerary. FD-MmTSP and NFD-MmTSP are the variants of the Closed Path Multi Depot multiple Travelling Salesman Problem (CP-MmTSP) [4, 6]. As the name suggests, each traveler is bound to go back to any depot after completing the itinerary.

Multi Departure Single Destination mTSP (MDmTSP): In this variant [6] of mTSP, total numbers of vertices and the distance between each pair of vertices are provided as an input, and the essence of the problem is to find an optimal itinerary for m travelers who start the route from different depots and reach at a single depot after covering all the given vertices.

Single Departure Multiple Destination mTSP (SDmTSP): If m travelers start the journey from a single depot and end at multiple destinations covering all the given nodes, it is called SDmTSP [6]. This problem aims to find out such an optimal itinerary for all the travelers (one itinerary for each traveler) covering all the nodes (1, 2, ...,n).

Generalized TSP (GTSP): The aim of GTSP [33] is to find out the maximally profitable route within a given limit of time. The total number of nodes is divided into clusters which consist of a combination of nodes, and we have to select the nodes to find out the shortest path among the selected nodes by visiting any one node in the cluster.

Team Orienteering Problem (TOP): TOP [9] is a variation of OP where we have to discover out m itinerary for m teams (one itinerary for each team) to collect overall maximum profit (score), which fulfill all the constraints like time budget (traversing time between each pair of vertices is predefined in the problem) etc. all the vertices cannot be traversed because of the given time budget problem. Hence, vertex selection is a critical task in TOP.

Orienteering Problem with Time Window (OPTW) and Team Orienteering Problem with Time Windows (TOPTW): TOPTW [39] is an extension of OPTW [24], services at each node can be started only within a predefined time slot in OPTW, and the number of path m is assumed to be 1 in it, while in TOPTW, a team can start giving services within a time-bound only, and m is greater than 1. The problem intends to find out the optimal cost path for each team within a time budget etc.

Time-Dependent OP (TDOP): Traversing time between any two vertices is considered as a constant in the case of OP, but it is not necessary for all the possibilities, i.e., there may be congestion on that particular route. This problem is solved by TDOP [16], in which time is considered to be reliant on the exodus time of the prior vertex.

Clustered Orienteering Problem (COP): COP [2] is a simplification of OP. In OP, the vertices are grouped in clusters, and the profit is linked with each group rather than each vertex. To get the benefit from each cluster, a traveler needs to traverse through all the nodes in the cluster.

Stochastic Orienteering Problem (St-OP): In St-OP [23], the aim is to discover the itinerary between starting and ending nodes so that the overall profit from all the visited nodes is maximized within a budget and a given probability of failure with a given stochastic cost associated with edges.

Orienteering Problem with Stochastic Weights (OPSW): This variant [15] of St-OP focuses on collecting the maximum price after completing the itinerary. Here, weights are coupled with travel time and travel cost; weights may be affected by dynamic conditions of the path, like congestion.

Probabilistic Orienteering Problem (POP): In POP [1], travelers need to visit a place according to a certain predefined probability and get the service within the deadline so that total revenue is maximized within a given time-bound.

Synchronized team Orienteering Problem with Time Window (STOPTW): The objective of STOPTW [49] is to maximize the profit associated with an asset within a given time budget in an environment where customers and service points are synchronized. There are some other constraints to follow, like time window and compatibility between the service point and customers.

Orienteering Problem with Functional Profits (OPFP): OPFP [32] is another variant of OP where the location of the place in the itinerary and its characteristics are also essential to decide the price collected from it. The objective is to find an itinerary where the traveler can visit the maximum number of places within a bound of time.

Set Orienteering Problem (SOP): Set Orienteering Problem (SOP) [3] is a variant of OP, which is a mixture of TSP and Knapsack Problem. It is a simplification of the OP in which the vertices are partitioned into disjoint sets, such that their union is the set of all vertices. Revenue is associated with each set rather than each node, and visiting anyone vertex in the set gives us the revenue associated with the particular set. The objective of SOP is to collect the maximum revenue after completing the itinerary within the time budget.

Multi-objective Open Set Orienteering Problem (MOOSOP): MOOSOP [14] is an extension of SOP. The key objective of MOOSOP is to search for an itinerary that fulfills multiple predefined goals like maximizing the customer who gets the service within a time budget and collecting maximum profit from customers.

3. RECENT SOLUTION APPROACHES

In this section, we cover the most recent solution approaches and their outcomes. Tables 1 and 2 show the summary of the approaches used by the researchers, respectively.

An Algorithm to Solve the mTSP

The algorithm to solve mTSP [34] is based on Dynamic Programming (DP). It consists of 5 phases: initialization with a random instance, solving the instance using DP, clustering the edges based on a predefined threshold, solving the new instance using DP, and updating the original solution.

An Improved Partheno-Genetic Algorithm (IPGA) for the OPMDmTSP

The basic idea in the suggested IPGA approach [28] is to implement a new selection method with the features of roulette and elitist sections. In addition, a complete change activity that presents the engendering component of the invasive weed optimization algorithm [30] is introduced. Experiments results show that IPGA is better in terms of resolution quality and convergence ability.

Table 1. Recent algorithms to solve the variants of TSP.

Ref.	Problem	Proposed Algorithm	Methodology Used
[34]	mTSP	A Dynamic Algorithm	Dynamic Programming
[28]	OPMDmTSP	An Improved Partheno-Genetic Algorithm	The Invasive Weed Optimization & Roulette and Elitist Section
[47]	MDmTSP	An Extension of the Christodes Heuristic	Classical Christodes Heuristic
[35]	GTSP	A Reinforcing Ant Colony System	Traditional Ant Colony System

An Extension of the Christofides Heuristic for the Generalized Multiple Depot Multiple Traveling Salesmen Problem (MDmTSP)

The proposed heuristic [47] works on the classical Christofides heuristic with the changes only in the tree algorithm [37], where it adds the arcs of least weight perfect matching of nodes of the odd degree to MST. The tight approximation improves from 2-approximation to (2-1)/2k-approximation, where k is the total sum of depots.

A Reinforcing Ant Colony System (RACS) for Solving the GTSP

RACS algorithm [35] utilizes the features of an exact exponential-time solution approach and a traditional ant colony system (ACS) [13, 44]. Some modifications (pheromone rule, etc.) are done in ACS to improve the correction policy in the (RACS). Findings show that ACS improves computational time and solution quality.

An Evolutionary Algorithm for the OP

The population-based evolutionary technique to solve OP [27] is focused on the characteristic that maintains the unfeasible result throughout the searching procedure. The algorithm exploits the steady-state genetic algorithm schema [46], but the divergence is for some generations. It uses the Tour-Improvement function for conversion and Add-Node and Drop-Node function for path tightening.

A Similarity Hybrid Harmony Search (SHHS) Algorithm for the TOP

The SHHS algorithm [43] is an extension of the classical harmony search algorithm [18] with a new method called the similarity process. Similarity process is used to enhance the quality of the results. Seven different sets of instances were solved using the algorithm, and it turns out that the algorithm provides the optimal results 276 times out of 328 instances.

An Iterated Local Search (ILS) Algorithm for Solving the OPTW

The ILS algorithm [22] is divided into three phases: LocalSearch, Perturbation, and AcceptanceCriterionuses. Some methods like insert, replace, swap, and 2-opt for local searching. The Shake method is used in Perturbation while Diversification and Intensification of the search are controlled by a combination of Perturbation and AcceptanceCriterionuses phases of the algorithm.

An Evolution Strategy Approach to the TOPTW

The proposed solution technique [26] is a combination of evolution strategy (ES) [7, 40] and constructive heuristic; it uses the ruin and recreates procedure to generate an offspring result, where some nodes are removed from an exigent result and added back in the itinerary until the complete result is found.

A Fast solution method for the TDOP

Verbeeck et al. proposed an algorithm [45] to solve TDOP in which time duration between two vertices depends on the first vertex's leaving time. The basic idea of the algorithm is to use the properties of Ant-Colony optimization with a Time-Dependent local search algorithm operational with local evaluation matric.

A Hybrid Heuristic Algorithm to solve COP

The HHA algorithm [48] to solve COP is a blend of an Adaptive Large Neighborhood Search (ALNS) [36] and an efficient split method [5]. The features of these two components are helpful to explore an ample search space and direct representation to find an itinerary. In the case of only one traveler, the hybrid heuristic algorithm provides the 38 new best results, and it also ends the optimal results in all the tested cases.

An Adaptive Method for the St-OP
In the proposed strategy [42], a path tree is generated so that it has a higher skipping possibility if there are branches at vertex-time states; as a result, a new sequence of nodes is allowed into the solution. It is shown that the proposed solution approach is more effective in the case of collecting rewards even if the branches are limited to control evaluation time.

Table 2. Recent algorithms to solve the variants of OP.

Ref.	Problem	Proposed Algorithm	Methodology Used
[27]	OP	An Evolutionary Algorithm	Population-Based Evolutionary Algorithm
[43]	TDOP	A Similarity Hybrid Harmony Search Algorithm	Harmony Search Algorithm
[22]	OPTW	An Iterated Local Search (ILS) Algorithm	Local Search Algorithm
[26]	TOPTW	An Evolution Strategy Approach	Evolution Strategy & Constructive Heuristic
[45]	TDOP	A Fast Solution Method	Ant-Colony Optimization & Local Search Algorithm
[48]	COP	A Hybrid Heuristic Algorithm	An Adaptive Large Neighborhood Search & A Split Method
[42]	St-OP	An Adaptive Method	Constrained Markov Decision Process
[41]	OPSW	Two-stage Robust Optimization	Two Stage Optimization Model
[11]	POP	A Tabu Search Algorithm	Tabu Search & Monte Carlo Evaluator
[49]	STOPTW	GRASP-ILP and Set Cover Hybrid Heuristic	Local Search Algorithm & Randomized Adaptive Search Procedure
[32]	OPFP	Ant Colony Optimization	Ant-Colony Optimization
[8]	SOP	A Biased Random Key Generator Algorithm	Genetic Algorithm
[14]	MOOSOP	A Hybrid Algorithm	Pareto Evolutionary Algorithm & Genetic Algorithm

Two-stage Robust Optimization for the OPSW
In this approach, a two-stage optimization model [41] is proposed, with fewer variables and constraints than a one-stage optimization model, so it will be easier to solve using the MILP solver (IBM CPLEX, etc.). Results show that it is 900+ faster than a static approach to solve OPSW.

A Tabu Search Algorithm for the POP
The Tabu Search [11] to solve POP is a combination of traditional Tabu search [20] and Monte Carlo examining target objective evaluator [10]. The selection of nodes and evaluation of the objective function of an itinerary is determined by a Monte Carlo evaluator. At the same time, A 2-opt local search heuristic method is used to discover a new path by deleting some non-adjacent nodes to find out the final sequence of nodes.

GRASP-ILP and Set Cover Hybrid Heuristic for the STOPTW
The proposed hybrid algorithm [49] is an integration of Greedy Randomized Adaptive Search Procedure (GRASP) and Iterated Local Search (ILS) [29]. In contrast, set covering formulation is used for the post-optimization segment. The suggested algorithm has the features of variable neighborhood descent search method and adaptive candidate list-based insertion heuristic. Simulations show that the presented technique finds the optimal solution in the case of medium and large instances.

Ant Colony Optimization for the OPFP
ACO works on pheromone concentration; the path with a higher pheromone intensity is selected by the traveler. The suggested solution for OFPF [32] completes in two steps; in the first step, itinerary probing is done to choose the appropriate traveler. In the second step, ACO is used to improve the itinerary.

Biased Random key Generator Algorithm (BRKGA) for the SOP
The basic idea in BRKGA [8] is to evolve the fitness of chromosomes by applying insert, swap, and Mck local search operator. Three additional rules are taken into consideration so that size of the sample can be enhanced, and the resolution of the problem can be accelerated; also, a hash table is introduced to enhance the searching time during computation.

An Algorithm for the Multi-Objective Open SOP
The proposed solution approach [14] uses a combination of the Strength Pareto & Evolutionary Algorithm (SPEA2) [50] and the Nondominated Sorting Genetic Algorithm (NSGAII) [12]. An instance of the generalized traveling salesman is used to solve the Multi-Objective Open Set Orienteering Problem (MOOSOP). NSGAII is used for population updating, while SPEA2 is used to find alternative solutions for the MOOSOP model.

4. FUTURE WORK

The study of TSP and OP is a well-established area, and we have investigated a lot of research on their variants using different nature-inspired algorithms, i.e., ant colony optimization, bat optimization, etc., but we analyzed that only a few researchers have addressed the TDOP and solution of its variants using nature-inspired algorithms to date; therefore, the researchers can focus on designing and finding the solution of time-based variants of OP (TDOP, OPTW, etc.) and also, future examination endeavors should be dedicated to the improvement of fitting arrangement strategies for these problematic variations.

5. CONCLUSION

In this paper, some of the most recent solution approaches are presented for the variants of TSP and OP (mTSP, GTSP, SOP, MOOSOP, etc.). The results show that the nature-inspired approaches provide the most optimized results to date, and are better than benchmarks solutions in some cases; thus, we can conclude that nature-based meta-heuristic algorithms are one of the most suitable approaches to solve NP-hard routing problems.

6. REFERENCES

[1] E. Angelelli, C. Archetti, C. Filippi, and M. Vindigni, “The Probabilistic Orienteering Problem”, **Computers & Operations Research**, Vol. 81, May 2017, pp. 269–281.

[2] E. Angelelli, C. Archetti, and M. Vindigni, “The Clustered Orienteering Problem”, **European Journal of Operational Research**, Vol. 238, No. 2, Oct. 2014, pp. 404–414.

[3] C. Archetti, F. Carrabs, and R. Cerulli, “The Set Orienteering Problem”, **European Journal of Operational Research**, Vol. 267, No. 1, May 2018, pp. 264–272.

[4] M. Assaf and M. Ndiaye, “Multi Travelling Salesman Problem Formulation”, **International Conference on Industrial Engineering and Applications (ICIEA)**, 2017, pp. 292–295.

[5] J. Beasley, “Route First-Cluster Second Methods for Vehicle Routing”, **Omega**, Vol. 11, No. 4, Jan. 1983, pp. 403–408.

[6] T. Bektas, “The Multiple Traveling Salesman Problem: An Overview of Formulations and Solution Procedures”, **Omega**, Vol. 34, No. 3, Jun. 2006, pp. 209–219.

[7] H. G. Beyer and H. P. Schwefel, “Evolution Strategies – A Comprehensive Introduction”, **Natural Computing**, Vol. 1, No. 1, 2002, pp. 3–52.

[8] F. Carrabs, “A Biased Random-Key Genetic Algorithm for the Set Orienteering Problem,” **European Journal of Operational Research**, Vol. 292, No. 3, Aug. 2021, pp. 830–854.

[9] I. M. Chao, B. L. Golden, and E. A. Wasil, “The Team Orienteering Problem”, **European Journal of Operational Research**, Vol. 88, No. 3, Feb. 1996, pp. 464–474.

[10] X. Chou, L. M. Gambardella, and R. Montemanni, “Monte Carlo Sampling for the Tourist Trip Design Problem”, **Millenium**, Vol. 2, No. 10, 2019, pp. 83–90.

[11] X. Chou, L. M. Gambardella, and R. Montemanni, “A Tabu Search Algorithm for the Probabilistic Orienteering Problem”, **Computers & Operations Research**, Vol. 126, Feb. 2021, p. 105107.

[12] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II”, **IEEE Transactions on Evolutionary Computation**, Vol. 6, No. 2, Apr. 2002, pp. 182–197.

[13] M. Dorigo, **Optimization, Learning and Natural Algorithms**, Ph.D. Thesis, Politecnico di Milano, 1992.

[14] J. Dutta, P. S. Barma, A. Mukherjee, S. Kar, and T. De, “A Multi-objective Open Set Orienteering Problem”, **Neural Computing and Applications**, Vol. 32, No. 17, Mar. 2020, pp. 13953–13969.

[15] L. Evers, K. Glorie, S. Van Der Ster, A. I. Barros, and H. Monsuur, “A Two-stage Approach to the Orienteering Problem with Stochastic Weights”, **Computers & Operations Research**, Vol. 43, Mar. 2014, pp. 248–260.

[16] F. V. Fomin and A. Lingas, “Approximation Algorithms for Time-dependent Orienteering”, **Information Processing Letters**, Vol. 83, No. 2, Jul. 2002, pp. 57–62.

[17] B. Gavish and S. C. Graves, “The Travelling Salesman Problem and Related Problems”, **Operations Research Center**, MIT, Jul. 01, 1978.

[18] Z. W. Geem, J. H. Kim, and G. V. Loganathan, “A New Heuristic Optimization Algorithm: Harmony Search”, **Simulation**, Vol. 76, No. 2, Feb. 2001, pp. 60–68.

[19] R. Gerhard, “The Traveling Salesman: Computational Solutions for TSP Applications, Lecture Notes in Computer Science”, **Springer**, 1994.

[20] F. Glover, “Tabu Search-Part I”, **ORSA Journal on Computing**, Vol. 1, No. 3, Aug. 1989, pp. 190–206.

[21] B. L. Golden, L. Levy, and R. Vohra, “The Orienteering Problem”, **Naval Research Logistics**, Vol. 34, No. 3, Jun. 1987, pp. 307–318.

[22] A. Gunawan, H. C. Lau, and K. Lu, “An Iterated Local Search Algorithm for Solving the Orienteering Problem with Time Windows”, **Evolutionary Computation in Combinatorial Optimization**, Apr. 2015, pp. 61-73.

[23] T. İlhan, S. M. R. Iravani, and M. S. Daskin, “The Orienteering Problem with Stochastic Profits”, **IIE Transactions**, Vol. 40, No. 4, Feb. 2008, pp. 406–421.

[24] M. G. Kantor and M. B. Rosenwein, “The Orienteering Problem with Time Windows”, **Journal of The Operational Research Society**, Vol. 43, No. 6, Jun. 1992, pp. 629–635.

[25] P. Vansteenwegen, W. Souffriau, and D. Oudheusden, “The Orienteering Problem: A Survey”, **European Journal of Operational Research**, Vol. 209, No. 1, Feb. 2011, pp. 1–10.

[26] K. Karabulut and M. F. Tasgetiren, “An Evolution Strategy Approach to the Team Orienteering Problem with Time Windows,” **Computers & Industrial Engineering**, Vol. 139, Jan. 2020, p. 106109.

[27] G. Kobeaga, M. Merino, and J. A. Lozano, “An Efficient Evolutionary Algorithm for the Orienteering Problem”, **Computers & Operations Research**, Vol. 90, Feb. 2018, pp. 42–59.

[28] P. Lou, K. Xu, J. Yan, and Z. Xiao, “An Improved Partheno-Genetic Algorithm for Open Path Multi-Depot Multiple Traveling Salesmen Problem”, **Journal of Physics: Conference Series**, Vol. 1848, No. 1, Apr. 2021, p. 012002.

[29] H. R. Lourenço, O. C. Martin, and T. Stützle, “Iterated Local Search”, **Springer: Handbook of Metaheuristics**, Vol. 57, 2003, pp. 320–353.

[30] A. R. Mehrabian and C. Lucas, “A Novel Numerical Optimization Algorithm Inspired from Weed Colonization”, **Ecological Informatics**, Vol. 1, No. 4, Dec. 2006, pp. 355–366.

[31] C. E. Miller, A. W. Tucker, and R. A. Zemlin, “Integer Programming Formulation of Traveling Salesman Problems”, **Journal of the ACM**, Vol. 7, No. 4, Oct. 1960, pp. 326–329.

[32] K. D. Mukhina, A. A. Visheratin, and D. Nasonov, “Orienteering Problem with Functional Profits for Multi-Source Dynamic Path Construction,” **PLOS ONE**, Vol. 14, No. 4, Apr. 2019.

[33] C. E. NOON, **The Generalized Traveling Salesman Problem**, Ph.D. Thesis, University of Michigan, 1988.

[34] P. Pérez-Escalona, I. Rapaport, J. Soto, and I. Vidal, “The Multiple Traveling Salesman Problem on Spiders”, **International Conference on Current Trends in Theory and Practice of Informatics**, 2021, pp. 337–348.

[35] C. M. Pintea, P. C. Pop, and C. Chira, “The Generalized Traveling Salesman Problem Solved with Ant Algorithms”, **Complex Adaptive Systems Modeling**, Vol. 5, No. 1, Aug. 2017.

[36] S. Ropke and D. Pisinger, "An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows", **Transportation Science**, Vol. 40, No. 4, Nov. 2006, pp. 455–472.

[37] D. J. Rosenkrantz, R. E. Stearns, and P. M. Lewis, II, "An Analysis of Several Heuristics for the Traveling Salesman Problem", **SIAM Journal on Computing**, Vol. 6, No. 3, Sep. 1977, pp. 563–581.

[38] H. M. Salkin and C. A. De Kluyver, "The Knapsack Problem: A survey", **Naval Research Logistics Quarterly**, Vol. 22, No. 1, Mar. 1975, pp. 127–144.

[39] M. W. P. Savelsbergh, "The Vehicle Routing Problem with Time Windows: Minimizing Route Duration", **ORSA Journal on Computing**, Vol. 4, No. 2, May 1992, pp. 146–154.

[40] H. P. Schwefel, **Evolution Strategy and Numerical Optimization**, Ph.D. Dissertation, the Technical University of Berlin., 1975.

[41] K. Shang, F. T. S. Chan, S. Karungaru, K. Terada, Z. Feng, and L. Ke, "Two-Stage Robust Optimization for the Orienteering Problem with Stochastic Weights", **Complexity**, Nov 2020, pp. 1–15.

[42] T. C. Thayer and S. Carpin, "An Adaptive Method for the Stochastic Orienteering Problem", **IEEE Robotics and Automation Letters**, Vol. 6, No. 2, Apr. 2021, pp. 4185–4192.

[43] E. Tsakirakis, M. Marinaki, Y. Marinakis, and N. Matsatsinis, "A Similarity Hybrid Harmony Search Algorithm for the Team Orienteering Problem", **Applied Soft Computing**, Vol. 80, Jul. 2019, pp. 776–796.

[44] F. J. Varela and P. Bourgine, "Toward a Practice of Autonomous Systems", **Proceedings of the First European Conference on Artificial Life**, MIT Press, 1992.

[45] C. Verbeeck, K. Sörensen, E.H. Aghezzaf, and P. Vansteenwegen, "A Fast Solution Method for the Time-Dependent Orienteering Problem," **European Journal of Operational Research**, Vol. 236, No. 2, Jul. 2014, pp. 419–432.

[46] L. D. Whitley, **The GENITOR Algorithm and Selection Pressure: Why Rank-Based Allocation of Reproductive Trials is Best**", Colorado State University, 1989, pp. 116-121.

[47] Z. Xu and B. Rodrigues, "An Extension of The Christofides Heuristic for the Generalized Multiple Depot Multiple Traveling Salesmen Problem", **European Journal of Operational Research**, Vol. 257, No. 3, Mar. 2017, pp. 735–745.

[48] A. E. Yahiaoui, A. Moukrim, and M. Serairi, "The Clustered Team Orienteering Problem", **Computers & Operations Research**, Vol. 111, Nov. 2019, pp. 386–399.

[49] A. Yahiaoui, A. Moukrim, and M. Serairi, "GRASP-ILS and Set Cover Hybrid Heuristic for the Synchronized Team Orienteering Problem with Time Windows", **International Transactions in Operational Research**, Jan. 2022.

[50] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm SPEA2: Improving the Strength Pareto Evolutionary Algorithm", **TIK-Report**, Vol. 103, 2001.