An optimization approach for green tourist trip design

Ghazaleh Divsalar
Iran University of Science and Technology
Ali Divsalar (ali.divsalar@nit.ac.ir)
Babol Noshirvani University of Technology https://orcid.org/0000-0002-2461-3827
Armin Jabbarzadeh
Ecole de technologie superieure
Hadi Sahebi
Iran University of Science and Technology

Research Article

Keywords: Tourist Trip Planning, Environmental Aspects, Multi-modality, Multi-Objective Variable Neighborhood Search

Posted Date: June 14th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-315613/v1

Version of Record: A version of this preprint was published at Soft Computing on February 21st, 2022. See the published version at https://doi.org/10.1007/s00500-022-06834-1.

An optimization approach for green tourist trip design

Ghazaleh Divsalar[1], Ali Divsalar[2*], Armin Jabbarzadeh[3], Hadi Sahebi[1]

1 School of Industrial Engineering, Iran University of Science and Technology, Narmak, Tehran, Iran
2 Department of Industrial Engineering, Babol Noshirvani University of Technology, Babol, Iran
3 Systems Engineering Department, École de technologie supérieure (ETS), Montreal, Canada

Abstract

In this paper, the Multi-Objective Multi-Modal Green Tourist Trip Design Problem (MO-MM-GTTDP) as the multi-modal variant of the orienteering problem is investigated. For this problem, a Multi-Objective mixed-integer linear model is formulated, which maximizes the total score of the Trip, minimizes the total cost of the trip as well as the total emission produced in the trip. Various transportation modes are considered for the tourist to choose to move between points of interest (POIs). The tourist choice may be affected by the transportation time and cost. Moreover, choosing the transportation mode will have an impact on the amount of trip pollutants. The cost of visiting POIs, as well as the cost of transportation between POIs, is considered as the total cost of the tour. In addition, a Multi-Objective Variable Neighborhood Search (MOVNS) algorithm is designed to solve instances of this problem. Moreover, a $\varepsilon - \text{constraint}$ method is implemented in CPLEX and used to evaluate the performance of the presented MOVNS. New instances of the problem are generated based on the existed benchmark OP instances. The conclusion is the high quality of the proposed MOVNS algorithm solutions in practically acceptable computation time (few seconds). Finally, a small case study based on real data on several POIs in the city of Tehran is generated and used to demonstrate the performance of the proposed model and algorithm in practice. For this case study, by using the multi-attribute decision-making method of TOPSIS, the obtained non-dominated solutions are ranked, and the best ones are presented to the tourist.

Keywords: Tourist Trip Planning; Environmental Aspects; Multi-modality; Multi-Objective Variable Neighborhood Search

1. Introduction

Nowadays, Tourism has become as one of the significant and even crucial sources of foreign income in a lot of countries, particularly in the context of developing economies [1, 2]. This industry has a vital role in economic development, involves multiple sectors, and interlinks with several industries [3]. Imagine a tourist who wants to visit a touristic city with many attractions, in a couple of hours. According to the tourist's preferences, each POI has a degree of attractiveness and there is a specific visiting cost for each of them. In addition, there are different transportation facility modes with different transferring costs and time that can be selected by tourists to move between POIs. In this case, there are three aspects to consider: the potential utility as well as the cost of visiting each point, the travel cost and the duration of transporting between POIs, and the amount of carbon dioxide released when traversing between POIs [4].

In general, personalized tourist trip planning is a complicated and time consuming process which consists in selecting points of interest (POIs) and scheduling of trips [5], giving a suitable plan to tourists on the sequence of POIs to explore conveniently and cost-effectively [6,7]. However, tourists often do not have enough time to visit all of the POIs during their day tour [8]. This problem has been known as the "Tourist Trip Design Problem" (TTDP). TTDP includes tour route planning for tourists, and maximizing their

* Corresponding author. *E-mail address*: Ali.Divsalar@nit.ac.ir; ORCID: 0000-0002-2461-3827

desirability while considering existing constraints [9], such as selecting of POIs they feel are the most appropriate for them [10].

Over the past several years, extensive research has been generated on the content of TTDP [5, 8, 11-15]. Despite recent achievements, research on the domain of tourist personalized trip planning is still at early stages. By analyzing the existing studies, we find that most of researches have investigated the TTDP without considering the environmental aspects of the tour. However, the unplanned development of this section can have destructive effects on natural resources and local societies. Moreover, tourism is considered as a significant contributor to climate change by making greenhouse gas emission [16]. In 2003, the United Nations World Tourism Organization (UNWTO), mentioned tourism as an influential factor in climate change and global warming's phenomenon [17]. This subject requires a useful trade-off between maximizing tourist entertainment and minimizing the produced pollution of the tour. Thus, of the resulted personalized trip planning problem can be treated as a multi-objective (M-O) problem, which has attracted little attention to date.

knowing the restrictions of previous studies attend to the TTDP, the orienteering problem (OP) and its variants which have been applied to model various versions of the TTDP [18-19] are investigated in this research. The OP is a single-criterion variant of traveling salesman problem with profits (TSPP) in which a set of vertices with a determined score is given. For all existing pairs of vertices, there is a required travel time. Due to the time limitation, not all vertices can be visited. Besides, the first and final vertices have to be visited, and other vertices can be visited at most once [20].

Although Vincent et al. [21] made pioneering efforts on this issue from the view of the multi-modality of team orienteering problem, their study does not mention some relevant points: First, their research concentrate on attaining the benefits of tourists as uniformly as possible, but it neglects minimizing the cost as well as produced pollutants of the tour. Second, this study considered the choice of POIs, but dismissed sequencing the transportation facilities between POIs, which may depend on the cost and the amount of produced pollutant. Except for this study, the concern of tourism's roles in climate change is nearly neglected in the investigated variants of TTDP. Moreover, to our best understanding, in all the presented M-O variants of the OP there is a tradeoff for achieving the maximum utility of the tour by different categories of POIs (e.g., leisure, history, culture, shopping, etc), and different score for each category. In fact, the cost of the tour has been considered as a constraint.

To conceal the gaps, the present work aims to develop a personalized tourist tour by considering the tour cost as well as the environmental aspects in a multi-modal environment. This problem is complicated because of considering three potentially conflicting objectives and multiple limitations. To dominate the complexity involved, we handle three objectives by using the definition of Pareto front optimality and developing an efficient M-O meta-heuristic approach, namely the MOVNS. The performance of the presented MOVNS is evaluated according to the M-O evaluating criteria by comparing its results with the results obtained by the implemented epsilon-constraint method. In addition, a small real example over a number of POIs in the city of Tehran (IRAN) is generated and used to demonstrate the performance of the proposed model and algorithm in practice. For this case study, by using the multi criteria decision making method of TOPSIS, the obtained non-dominated solutions are ranked, and the best ones are presented to the tourist.

The rest of this study is structured as follows: Section 2 manages a review on the literature of TTDP and its variants. Section 3 introduces a mathematical modeling for the MO-MM-GTTDP. Section 4 describes the proposed solution method and presents all its phases in detail. The case study of Tehran is presented and discussed in Section 5. Finally, this study is summarized and conclusions and potential developments are provided in Section 6.

2. Literature review

According to Gavalas et al. [22], variants of tourist trip design problem can be categorized as single-tour and multiple-tour TTDP. The majority of published research in the literature of TTDP, model the problem based on OP and its variants [10]. The OP is categorized as a variant of TSPP which is a bi-objective problem. In the TSPP the time budget constraint of the OP is also considered as an objective function. Therefore, the goals in the TSPP are maximizing the collected profit while the travel cost is minimized [22]. Bérubé et al. [23] present the first exact Pareto solutions for TSPP by using the $\varepsilon - constraint$ method in 2009. At the same year, the M-O Orienteering Problem (MOOP) is introduced by Schilde et al. [24] as a variant of the OP, in which each POI can be in a different category (e.g., history, culture, shopping, leisure) and a different score has been considered for each category. They presented an adaptation of the Pareto Ant Colony Optimization (P-ACO) [25] metaheuristic and a M-O extension of Variable Neighborhood Search (P-VNS) introduced by chao et al. [26] for the first time. In 2015 Yu-Han Chen et al. [27] proposed an Ant Colony algorithm for MOOP with Time Windows (MOOPTW) to find a set of non-dominated solutions. In the MOOPTW, there is a different specified time interval for visiting each POI. An artificial bee colony algorithm is also presented for the bi-objective MOOP by Martin-Moreno & Vega-Rodrigues (2018) [28]. They compared their method with the P-ACO and P-VNS of Schilde et al. [24] and proved a higher quality performance over the benchmark and real-word instances. Recently Zhenga et al. [29] introduce a modified version of MOOP, which considers a compromise between the fairness of individual members, and the total obtained score of the group. In this problem a personalized trip is proposed for a heterogeneous group of tourists modeled as a common M-O problem. This study an ant colony optimization heuristic is combined with the differential evolution algorithm as the solution approach. The Time Windows version of the OP (OPTW) is introduced by Marisa et al. in 1992 [30] and for the first time is used to model the TTDP by Vansteenwegen et al. [31]. They argued that time windows can have a significant impact on the regular OP. Another variant of the OP is the Time Dependent OP (TDOP) proposed by Fomin and Lingas [32]. In the TDOP, the cost of moving between POIs depends on the time that this travel is started. Therefore, the TDOP has been suggested to model the Multi-Modal cases. In 2016, two meta-heuristic algorithms are presented for solving the M-O TDOP by Mei et al. [33]: A M-O Memetic Algorithm (MOMA) and a M-O Ant Colony System (MACS). Liao et al. in 2018 [5] introduce a time-dependent stochastic TDOP to design a personalized route where travel times or wait times are not static. This study designs a hybrid heuristic algorithm based on random simulation (RSH2A). They have evaluated their solution method by a case study in China. Another bi-objective variant of the OP is recently introduced by Dutta et al. [34], where they maximize the visited clusters' profits as well as the customers' satisfaction in the open Set Orienteering Problem (SOP). The SOP is a clustered version of the OP, which can be applied to model the TTDP [35].

An interesting multi-tour variant of the OP is the Orienteering Problem with Hotel Selection (OPHS), which is introduced by Divsalar et al. [20, 36]. In the OPHS they considered a set of attractions with an associated profit and a set of non-rated hotels, with the goal of determining a number of connected trips to visit a some of available attractions and maximizing the total collected profits. They proposed two methods to solve the OPHS, a VNS, and a memetic algorithm. Another multi-tour extension of the OP is the Team Orienteering Problem (TOP), in which the goal is to find K tours (or paths) in a limited time to maximize the total score given that each nonterminal node can be visited at most once in all K tours. The TOP is introduced by I-Ming et al. [37] and is used to model the TTDP variant by Vansteenwegen et al. [38].

Lately, Tsakirakis et al. [39] introduced the Similarity Hybrid Harmony Search (SHHS) algorithm as a new variant of Harmony Search (HS) for TOP. They proposed the static and dynamic versions of the method. In the static version, the values for the parameters are predefined and in the dynamic one, parameters are adjusted during the solve procedure. The results of applying the proposed method for benchmark instances of TOP show the superiority of the dynamic version.

In a research by Gunawan et al. [40] an Iterated Local Search as well as a hybrid Simulated Annealing and Iterated Local Search (SAILS) is presented for the TOP with Time Windows (TOPTW). They tested their proposed methods on benchmark TOPTW instances and concluded from the results that the proposed ILS and SAILS improves the solutions in terms of quality. In 2018, Hu et al. [41] introduced the M-O TOP with Time Windows (MO-TOPTW), where multiple profits for each point are considered. They developed a M-O evolutionary algorithm on the basis of a decomposition technique and constraint programming to solve this problem. The results show an improvement by comparing the results of applying benchmark instances with best-known solutions. Another version of MOTOPTW is introduced by Hapsari et al. [42], which is different in objective functions. In this research besides the maximizing of the scores, the available time of the tour is considered as a second objective instead of a constraint. The authors used an Adjusted ILS (AILS) to solve the proposed model on a real case and compared it with some other metaheuristics. The result demonstrates the higher performance of their method in terms of computational time and providing itineraries with higher total score. In 2020, Saeedvand et al. [43] proposed a new variant of MOTOPTW for rescue applications with five objectives. They present an efficient solution by combining M-O evolutionary algorithms (MOEAs) with learning algorithms. The results show that their algorithm decreases the gap with best known solutions of TOPTW by up to 42%

In 2017, the Multi-Modal Team Orienteering Problem (MM-TOPTW) as new functional variant of the TOPTW was introduced by Vincent et al. [21]. In this research, several transportation modes are considered for tourists in the TTDP. The goal is to achieve the maximum value of score without violating the travel cost and time limitations. In addition, time windows are considered for the opening and closing time of each POI. They proposed a two-level particle swarm optimization with multiple social learning terms for the MM-TOPTW. The authors implemented their proposed algorithm for solving 56 benchmark VRPTW instances. The results showed promising solutions within an acceptable computation time.

Due to the wideness of the existing OP variants and applications, for an all-inclusive literature review, we refer the readers to the fairy recent surveys on the OP variants by Gunawan et al. [19] and Vansteenwegen et al. [31].

With an overall look to the presented literature, very few number of research among the OP variants and its extensions, discuss the M-O and the multi-modal variant of the orienteering problem or, more specifically, the TTDP. In addition to the tourist's preferences and the total cost of the tour, that contains the visiting and transportation cost, giving a choice to the tourist to choose his/her transportation facility to move between POIs can be considered as an effective factor in the problem. Another factor, which has not been noticed in the literature, is the environmental aspect of the tour. Due to the major effect of tourism on air pollution, and climate change of the touristic cities, this factor can be considered in TTDP. According to this, we consider tourist preferences, the cost and the amount of co2 emission and introduce a new variant of TTDP, which we call the M-O Multi-Modal Green TTDP (MO-MM-GTTDP). As a result, because of the compromise between three objectives, the MO-MM-GTTDP is a complex problem to solve. A Pareto-based local search approach is proposed to achieve diversified and realistic routes for this problem based on M-O optimization and to handle conflicting objectives. Moreover, to rank and clarify the best route, the TOPSIS method is applied for the obtained non-dominated solutions from a real example made over real data from city of Tehran.

3. Problem description and mathematical formulation

In this section, the Multi-Objective Multi-Modal Green Tourist Trip Design Problem (MO-MM-GTTDP) is formally formulated as a mixed-integer linear problem.

Consider G = (V, E), as an undirected graph. In this graph, V is the set of N vertices, and E is the set of its edges. We consider the POIs as the vertices of the graph G from 1...N. Each POI is associated a score S_i based on the tourist's preferences as well as a visit cost showed by VC_i. K is the set of types of transportation facilities to move between POIs. Each transportation mode produces a certain amount of emission per distance measure. P_{ijk} represents the amount of carbon dioxide emission of traveling from i to j using the transportation mode k, and C_{ijk} is the cost of this transport. t_{ijk} is the travel time of passing edge $(i,j,k) \in E$. The start and the end node of the visit tour are fixed to nodes 1 and N, respectively. Each node is visited at most once. X_{ijk} is the binary variable set to 1 if edge (i,j) is included the solution, and traversed by transportation mode type $k \in K$; 0 otherwise. Then, the mathematical model is presented using equations (1) to (11).
$\delta^+(S) = \{(i.j) \in E: i \in S. j \notin S\}$ shows all the outgoing edges from set S and for the simplicity $\delta^+(\{i\})$ is presented as $\delta^+(i)$. In a similar way, $\delta^-(S) = \{(i,j) \in E: i \,!\in S, j \in S$ shows all the incoming edges to set S. Then, using the graph notation, the mixed integer model is presented as follows:

$$\boldsymbol{max} \sum_{(i,j)\in E} \sum_{k\in K} S_i X_{ijk} \tag{1}$$

$$\boldsymbol{min} \sum_{(i,j)\in E} \sum_{k\in K} (VC_i X_{ijk} + C_{ijk} X_{ijk}) \tag{2}$$

$$\boldsymbol{min} \sum_{(i,j)\in E} \sum_{k\in K} (P_{ijk} X_{ijk}) \tag{3}$$

$$\sum_{(i,j)\in E} \sum_{k\in K} t_{ijk} X_{ijk} \leq T_{max} \tag{4}$$

$$\sum_{k\in K} X_{ijk} \leq 1 \, , \forall (i,j) \in E \tag{5}$$

$$\sum_{k\in K} x(\delta^+(1))_k = 1 \tag{6}$$

$$\sum_{k\in K} x(\delta^-(N))_k = 1 \tag{7}$$

$$\sum_{k\in K} x(\delta^-(j))_k = \sum_{k\in K} x(\delta^+(j))_k \leq 1 \, , j \in V \tag{8}$$

$$U_i - U_j + 1 \leq (N-1)\left(1 - \sum_{k\in K} X_{ijk}\right), \forall (i,j) \in E \tag{9}$$

$$2 \leq U_i \leq N \, , \forall i \in V \tag{10}$$

$$X_{ijk} \in \{0,1\} \, , \forall (i,j) \in E, k \in K \tag{11}$$

Equation (1) maximizes the total profit of visited POIs. Equation (2) minimizes the total cost of the tour, including the cost of visiting POIs and the cost of moving between POIs. Equation (3) minimizes the carbon dioxide emissions produced by transportation facilities. Constraint (4) limits the total travel time within the time budget T_{max}. Constraints (5) ensure that in each path between i and j, only one transportation mode is selected. Constraints (6) & (7) ensure that node 1 is the start point, and node N is the end point of the tour. In a similar way, constraint (8) is the flow constraint and ensures that each node is visited at most once. Finally, constraints (9) and (10) ensure that there are no sub tours. The values of decision variables are determined by constraints (11).

4. Solution Strategies

In this section first, the epsilon-constraint method as one of the most reliable M-O methods is implemented to assess the effectiveness of the mathematical model. Moreover, the obtained results from this method are used to evaluate the proposed metaheuristic. Next to the epsilon-constraint method, an efficient meta-heuristic is developed to solve the MO-MM-GTTDP instances. This method is the M-O version of the VNS method (MOVNS). The method is based on combining several local moves within a systematic change procedure, avoiding to trap in a local optimum. Because of considering three objective functions simultaneously, designing special moves to focus on improving each objective and the transportation facility selection, the MO-MM-GTTDP is more challenging than the OP. Thus, significant computational efforts are assigned to evaluate combinations of vertices and the selected transportation facility to move between them.

4-1. Epsilon constraint

To have a benchmark for evaluating the performance of the proposed meta-heuristic, first an epsilon-constraint approach is implemented to solve the instances of the MO-MM-GTTDP. Therefore, in this section this method is briefly introduced.

Assume the following M-O problem: $Max\ (f_1(x), f_2(x), \dots, f_p(x))\ s.t.\ x \epsilon D$

Where x is the vector of decision variables, $f_1(x), \dots, f_p(x)$ are the p objective functions, and D is the feasible region. A M-O problem usually consists of a set of solutions called the Pareto-optimal front, where each Pareto optimal solution demonstrates a trade-off between different objectives. In fact, it is not possible to improve all objectives functions at the same time. In M-O optimization rather than the single-objective, comparing solutions is more complex. Two subjects have to be investigated in M-O optimizations: Pareto dominance and Pareto optimality [44].

In the ε-constraint method a part or the entire efficient set for a M-O problem is produced. This method can provide a representative subset of the Pareto set, which in most cases is efficient. In this method, the decision maker chooses only one objective as its main one to be optimized. The other objective functions are added as constraints to be less than or equal to given epsilon values [45]. Then, by variation in the epsilon values of the constrained objective functions, the efficient solutions of the problem are obtained. For a detail description of the ε-constraint method, one can refer to Mavrotas [46].

In the implementation of the ε-constraint method, the first objective (1 in the model) is used as the main objective function, and two other objectives (2 and 3 in the model) are placed in the set of constraints. Since the main objective is maximized and two others are minimization functions, in our implementation these constraints are first set to a large value and then decreased step by step in each iteration. This method is implemented in CPLEX 12.6 to solve the created instances of the MO-MM-GTTDP in 20 iterations as the stopping condition. Computational examination demonstrates that the exact method sometimes requires large computing times. Therefore, a meta-heuristic algorithm is proposed and implemented to solve real size instances of the MO-MM-GTTDP.

4-2. The multi-objective variable neighborhood search method

As mentioned in the literature section, the VNS has been successfully applied to many variants of the OP, such as TOP [47], TOP with time windows [48], and OP with Hotel Selection [20]. Therefore, in this paper, the M-O version of the VNS algorithm (MOVNS) is developed to tackle the MO-MM-GTTDP, which is in fact modeled as a M-O OP. Considering that the presented problem is M-O with three conflicting objectives; this is the first use of the MOVNS algorithm for solving an OP variant. In this algorithm, the optimizing concept is considering two main views: First, how to define the set of pre-established neighborhoods, and second, how to explore using these neighborhood structures [49]. A general overview of the applied MOVNS is presented in Figure 1.

Algorithm 1: Multi-Objective variable neighborhood search

Data preprocessing

Make initial Solutions(){$\text{Pareto} < -(\text{x1}, \text{x2}, \text{x3})$}

$\text{Pareto} \leftarrow \emptyset$

$\text{T} \leftarrow \text{Tmax}$

$\text{Iteration} \leftarrow 1$

While(Iteration $< maxIteration$){

| $\text{Pareto}(\text{x}'1, \text{x}'2, \text{x}'3) \leftarrow \text{Multi} - \text{Objective Shake}(\text{Pareto}(\text{x1}, \text{x2}, \text{x3}))$

| $\text{Pareto}(\text{x}'1, \text{x}'2, \text{x}'3) \leftarrow \text{Multi} - \text{Objective VND}(\text{Pareto}(\text{x}'1, \text{x}'2, \text{x}'3))$

| $\text{Pareto}(\text{x1}', \text{x2}', \text{x3}') \leftarrow \text{Update}(\text{Pareto}(\text{x}'1, \text{x}'2, \text{x}'3), \text{Pareto}(\text{x1}, \text{x2}, \text{x3}))$

| $\text{Iteration} \leftarrow \text{Iteration} + 1$

End While

Fig 1. General structure of the MOVNS algorithm

According to the Algorithm 1, after some preprocessing, at the first step of the algorithm, three initial solutions are generated. This initialization phase is explained in Section b. Then the improvement phase is started with the "shaking" phase followed by applying the local search part which is called the ''MOVND''[1]. Both shaking and the MOVND are presented in detail in Sections c and d, respectively. In the MOVND part of this algorithm, three main sets of neighborhoods are used, which correspond to three objective functions. In Section e, the ''update'' method as the third part of the improvement phase is described. In this method, the available Pareto solutions are updated so that finally dominated solutions are deleted, and the only effective (non-dominated) Pareto solutions are remained and moved to the next iteration. Obviously, these steps are repeated as long as the maximum iteration number is not exceeded.

a. Solution representation

To implement the proposed MOVNS, each solution of the MO-MM-GTTDP is represented as a two-dimensional array containing the list of selected nodes as well as the transportation mode that has been selected for moving between the selected nodes. The sequence of visiting nodes starts at the origin node (1) and ends at the destination node (n). An example of the solution representation of an instance of the problem with 20 nodes and 3 modes of transportation facilities is presented in Table 1.

Table 1. Solution representation for an instance with 20 nodes and 3 transportation facility modes

visNodes	**1**	5	11	7	2	3	15	17	8	**20**
SelectedTrpMode	2	1	2	2	0	0	2	1	0	-

[1] Multi-Objective variable neighborhood descent

b. Initialization

In the initialization part of the algorithm, due to the existence of three objective functions of the problem, three initial solutions are constructed; each of them corresponds to one objective function when creating a solution. In the preprocessing step, four matrices, including the distance between every pairs of nodes ($\mathrm{matrix_d}$), the travel time ($\mathrm{matrix_t}$) and the cost ($\mathrm{matrix_c}$) of moving between every pairs of nodes with any transportation mode, as well as the amount of co_2 emission that is produced by moving between every pairs of nodes using each transportation mode ($\mathrm{matrix_p}$) are generated. It should be mentioned that there are three types of constant values taken as the input and used in creating these matrices: constant-time, constant-cost, and constant-pollutant. Obviously, these constants depend on the transportation mode. Moreover, the Score and the visiting cost values for each node are used in the description of the algorithm as vectors of S(i) and VC(i).

All three initial solutions are made based on the famous nearest neighbor strategy. However, in each of them, a different selection measure is used. In the first initial solution, among the feasible non-included nodes, the selection priority for visiting node(i) is the maximum ratio of S(i) over the distance of the last visited node to node(i). After selection, node(i) is added to the end of the visited nodes list, and the process continues until no more node is feasible to visit. It should be noted that in this solution, it is assumed that the fastest transportation mode is selected for moving between POIs. In this case, the fastest facility is the one with the minimum constant-time. Considering the second objective of the model, for the second initial solution, the denominator of the mentioned ratio for node selection is cost (i). The cost of node(i) is equal to VC(i) plus the transportation cost of moving between the last visited node to node(i). For this solution, the cheapest transportation mode, the facility with the minimum constant-cost, is selected for moving between POIs. Creating the third initial solution is similar to the first except that the cleanest transportation mode is selected. An overview of the process of generating the first initial solution is presented in Figure 2.

Algorithm 2: Making initial Solution 1

Data preprocessing:
Making necessary matrices
$\mathrm{V} \leftarrow \{2, \ldots, \mathrm{n}-1\}$
$\mathrm{k} \leftarrow 1$
$\mathrm{visNodes[k]} \leftarrow 1$
$\mathrm{while(V \neq \emptyset)do\{}$
| find i , $\max\limits_{\forall i \in V} \frac{\mathrm{Score(i)}}{\mathrm{matrix_d(visNodes[k],i)}}$
| if(selecting i $\in$ V as next visiting point)is feasible{
| | $\mathrm{V} \leftarrow \mathrm{V}/\{\mathrm{i}\}$
| | $\mathrm{k} \leftarrow \mathrm{k}+1$
| | $\mathrm{visNodes[k]} \leftarrow \mathrm{i}$
| else "Break"
End while
$\mathrm{visNodes[k+1]} \leftarrow \mathrm{n}$

```
    x1 ← visNodes
```

Fig 2. Pseudocode for making initial solution 1

c. Shaking phase

According to the general structure of the algorithm (Algorithm 1), the Multi − Objective Shake is the first step in the improvement phase. The main goal of the shaking phase is to diversify the solution search space by skipping from local optimum. Considering the M-O nature of the problem, since both the sequence of visited nodes and transportation modes may affect the solution quality in terms of any of the objective functions, in the shaking phase of the proposed algorithm, two shaking moves are implemented: Node-Remove and Change-TrpMode. The first move is to remove some of the visited nodes randomly, and then to apply the local search to improve the solution. The number of nodes to remove for each solution is calculated by multiplying a constant parameter (α) by the number of visited nodes in the solution. At the same time when removing node (i) from the visited node list, the transportation mode for moving between node (i) to node (i+1) is removed, and the same transportation mode used to move between node $(i-1)$ and node (i) is used to move between node $(i-1)$ and node $(i+1)$. The second move in the shaking phase, Change-TrpMode, randomly changes the transportation modes for some of the arcs between visited nodes while keeping the solution feasible. The number of arcs for which the transportation modes are changed is again obtained by multiplying the constant parameter (α) by the number of visited nodes in the solution. It should be mentioned that the first shake move is applied on each solution in the current Pareto list. However, the second move is only applied on half of the solutions which are selected randomly. Algorithm 3 (Figure 3) represents the pseudo-code of the shaking phase.

```
Algorithm 3: Multi-Objective Shake
removeNum ← α × visNodes.size()
while(removeNum ≠ 0){
|    select a nodeNumber from visNodes randomely
|    removeNum ← removeNum − 1
|    NodeRemove(visNodes, nodeNumber)
End while
Generate a randomeValue in (0,1)
if(randomeValue < 0.5){
|    while(removeNum ≠ 0){
|    |    select a nodeNumber from visNodes randomely
|    |    select a TrpNumber from K randomely
|    |    while(TrpNumber = SelectedTrpMode(nodeNumber)){
|    |    |    select a TrpNumber from K randomely
|    |    if (replacing TrpNumber instead of SelectedTrpMode(nodeNumber))is feasible{
|    |    |    ChangeTrpMode(visNodes, nodeNumber, TrpNumber)
|    |    End if
|    End while
End
```

Fig 3. Pseudocode for multi-objective shake

a. MOVND

The local search part in the improving phase of the presented MOVNS algorithm is called the M-O variable neighborhood decent (MOVND). Again, according to the M-O nature of the problem, the MOVND contains three sub-parts that improve each solution according to each objective function. In fact, in this algorithm, three ''VND''[2] structures are used to improve the solution in terms of three objective functions, separately: VND-1, VND-2, and VND-3. As can be seen in Algorithm 4, in the overall structure of the MOVND, each VND includes local moves specifically designed to improve the solution in terms the corresponding objective function locally. More specifically, VND-1 (similarly VND-2 and VND-3) tries to improve the value of z_1 (symmetrically z_2 and z_3) and find the local optimum with respect to z_1 (uniformly z_2 and z_3) by using the implemented neighborhood structures. One main difference of the M-O over the single-objective optimization technique is that it is not enough only to check whether the final solution obtained by the VND is improved compared to the incumbent solution, but it needs to consider every explored solution during the search and give them a chance to be part of the Pareto front [49].

Algorithm 4: Multi-Objective Variable Neighborhood Descent

```
S ← ∅
for(each i on Pareto){
|   S ← VND − 1(Pareto[i])
Pareto ← Update(S, Pareto)
clear S
for(each i of Pareto){
|   S ← VND − 2(Pareto[i])
Pareto ← Update(S, Pareto)
clear S
for(each i of Pareto){
|   S ← VND − 3(Pareto[i])
Pareto ← Update(S, Pareto)
clear S
```

Fig 4. Pseudocode for multi-objective variable neighborhood descent

As shown in the Figure 4, all different solutions explored during a local move search are stored in the list S. After each VND, an Update function is applied to compare the solutions in S with the ones already exist in the Pareto list. This results in an updated Pareto list after each local move. The detail of the update process is presented in the next section. In the remainder of this section, each VND is explained separately.

As mentioned earlier, the VND-1 is designed to improve the first objective function of the model. So, concerning the time budget of the tour, two local search moves are used to raise the total score directly or

[2] Variable neighborhood descent

to reduction the total travel time with a hope to be able to visit more POIs and increase the total score: insert-Node and Two-Opt. A brief description of each move is given as follows:

Insert-Node: this move is increasing the total score of the tour by inserting nodes to the visited node list. Among non-visited nodes for which the insertion does not make the tour infeasible, the best insertion position of node (i) is found with the maximum ratio of the score over the length (i). Where, length (i) for the insertion of node (i) between node (j) and node $(j+1)$ is efficiently calculated using length $(i) = \mathrm{t}_{\mathrm{j,i}} + \mathrm{t}_{\mathrm{i,j+1}} - \mathrm{t}_{\mathrm{j,j+1}}$. It should be mention that the transportation mode for moving between node (j) and node (i) as well as between node (i) and node $(j+1)$ are assumed to stay the same as the former mode which is used between node (j) and node $(j+1)$ in the current solution.

Two-Opt: Due to its efficiency, this move is one of the most commonly used local neighborhood structures in routing problems. It reduces the travel time in the tour by replacing two existing edges with two new ones. In the implementation, for every possible pair of nodes, among the visited nodes it is checked whether the reversal in the order of nodes between them produce a travel time saving, and then the pair of nodes with the highest saving is chosen. After selection, the order of the nodes, as well as the transportation mode for moving between them, is reversed. To check out the feasibility of this move, the time matrix $(\mathrm{matrix}_{\mathrm{t}})$ is being used.

VND-2 is designed with respect to the second objective. The only local move used in this is called the Change-Node.

Change-Node: It tries to minimize the total cost of the tour by replacing a visited node by a none-visited node. As the first condition, node (j) is replaced with node (i) if this replacement decreases the total cost of the tour. This cost includes the visit cost, and transportation cost. If this change is feasible, then the difference between the scores is checked out, and with increase the total score, node (j) is replaced with node (i). If such a node is not found (i.e., the score of node (i) is smaller than the score of node (j)), then the replacement happens for the node with the maximum of the $\mathrm{diff}_{\mathrm{C}}/\mathrm{diff}_{\mathrm{S}}$ ratio. $\mathrm{diff}_{\mathrm{C}}$ Represents the difference of node (j) and node (i) cost (including the visit and the transport), and $\mathrm{diff}_{\mathrm{S}}$ equals to the difference of those nodes' score. Figure 5 represents the pseudo-code of Change-Node.

Algorithm 5: Change Node

Set a small number for max
for(each i on visNodes){
| for(each j on nodelist){
| if(j is not visited){
| | $\mathrm{diff}_{\mathrm{trp}} \leftarrow \mathrm{matrix}_{\mathrm{c}_{\mathrm{i-1,i,k}}} + \mathrm{matrix}_{\mathrm{c}_{\mathrm{i,i+1,k}}} - \mathrm{matrix}_{\mathrm{c}_{\mathrm{i-1,j,k}}} - \mathrm{matrix}_{\mathrm{c}_{\mathrm{j,i+1,k}}}$
| | $\mathrm{diff}_{\mathrm{VC}} \leftarrow \mathrm{vc}_{\mathrm{j}} - \mathrm{vc}_{\mathrm{i}}$
| | $\mathrm{D} \leftarrow \mathrm{diff}_{\mathrm{trp}} + \mathrm{diff}_{\mathrm{VC}}$
| | | if$(\mathrm{D} > 0)${
| | | | $\mathrm{diff} \leftarrow \mathrm{matrix}_{\mathrm{t}_{\mathrm{i-1,i,k}}} + \mathrm{matrix}_{\mathrm{t}_{\mathrm{i,i+1,k}}} - \mathrm{matrix}_{\mathrm{t}_{\mathrm{i-1,j,k}}} - \mathrm{matrix}_{\mathrm{t}_{\mathrm{j,i+1,k}}}$
| | | | if(replacing j inststead of i is feasible){

```
|   |   |   |   |   diff_S ← S_j − S_i
|   |   |   |   |   if( diff_S > 0){
|   |   |   |   |   |   j replace instead of i
|   |   |   |   |   else{
|   |   |   |   |   |   CS ← diff_C / diff_S
|   |   |   |   |   |   if(CS > max){
|   |   |   |   |   |   |   max ← CS
|   |   |   |   |   |   |   j replace instead of i
|   |   |   |   |   |   End if
|   |   |   |   |   End else
|   |   |   |   End if
|   |   |   End if
|   |   End if
|   End for
End
```

Fig 5. Pseudocode for change node

The third part of the MOVND is the VND-3. This part concentrates on the third objective function and includes one local move which tries to decrease the produced co_2 emissions over all the tour, according to z_3. This local move is called Change-TrpMode. It is designed to change the current transportation mode between POIs with a less pollutant transportation mode while respecting the time budget constraint. It starts from the first arc of the tour and the transportation mode used for moving between the first and second nodes of the visited node list. It greedily changes the transportation mode to reduce co_2 emissions. To be more specific, in each application of this local move, the transportation mode of an arc with the maximum '' $\mathrm{diff_p}$'' value is changed. $\mathrm{diff_p}$ Represents the difference between produced co2 emissions by using the new transportation mode instead of the current one for an arc. The pseudo-code of the Change-TrpMode is presented in Algorithm 6 (Figure 6).

```
Algorithm 6: Change TrpMode
Set a small number for max
for( all k ∈ K){
|   for( all i on visNodes){
|   |   if(k ≠ SelectedTrpMode[i]){
|   |   |   diff ← matrix_{t_{i,i+1,k}} − matrix_{t_{i,i+1,l}}
|   |   |   if(diff > max){
|   |   |   |   max ← diff
|   |   |   End if
```

```
|   |   End if
|   End for
if(max > 0){
|   k replace instead of l
End
```

Fig 6. Pseudocode for Change TrpMode

a. Update Pareto List

According to the presented M-O model, this section has an important role in our MOVNS algorithm. As shown in the Figure 1, in an iteration of the algorithm, the Update function is called in four places of the MOVNS. This function is designed to compare all the obtained solutions after application of each VND with the existing Pareto solutions. As a result of this phase, all none-dominated solutions are remained in the Pareto list and the duplicated solutions as well as the dominated ones are removed from the list. These comparisons are performed by using the corresponding objective values of each solution.

5. Experiments and results

The MOVNS algorithm is coded in Visual C++. The mathematical model and the $\varepsilon - \mathrm{constraint}$ method are implemented in CPLEX 12.6 to evaluate the performance of the presented meta-heuristic algorithm. First, new benchmark instances are generated to evaluate the performance of solution methods. These instances are described in Section 5-1. Generally, the performance of solution methods in solving the M-O models is evaluated by a set of standard criteria. The selected evaluating criteria in this research are described in Section 5-2. Before applying the MOVNS method on the benchmark instances, parameters of this algorithm should be tuned. In Section 5-3 it is described that how the Taguchi experimental design [50] is used for parameter tuning in this research. Finally, in Section 5-4 the results of both implemented methods over all generated benchmark instances are discussed and analyzed based on the described evaluation criteria.

5-1. Test instances

Because there exist no test instances for the MO-MM-GTTDP in the literature, a set of standard instances are generated to evaluate the work of the proposed algorithm. These instances are generated using the existing instances for the OP [51] by changing the number of nodes and available time for the tour. In addition, some new data are added to these instances, including the visit cost of POIs, travel time and cost for moving between POIs by any of available transportation modes, and also the amount of co2 emissions produced by any of available transportation modes. The visit cost of each POI is generated randomly with a uniform function in the interval of [10,150]. In all generated instances, three transportation modes are considered (e.g., Bicycle, Bus, and Taxi). The average travel time per distance unite for moving between POIs by using each transportation mode is assumed to 6, 4, and 2 units, respectively. Moreover, the cost of moving between POIs using each transportation mode is set as 0.3, 1, and 5 units per distance unite, respectively. The pollution constant per distance unit using each transportation mode is also assumed to be 0, 0.069, and 0.17, respectively.

The set of benchmark instances in this research is generated based on set1 of OP instances of Chao, et al., [26] which is available on [www.mech.kuleuven.be/cib/op]. In total, 30 instances are generated. Table 2 demonstrates the main characteristics of these instances. In general, Instances with 5 to 32 available POIs with a time budget of 10 to 80 are generated and used for evaluation of the algorithm. It should be mentioned that larger instances are unlikely to be solved using the implemented epsilon-constraint method in CPLEX.

Table 2. The set of generated test instances

Available time	Number of POIs
10	{5,10,15}
20	{5,10,15,20,25}
30	{10,15,20,25,30,32}
40	{10,15,20,25,30,32}
50	{15,20,25,30,32}
60	{20,25,30,32}
70	{20,25,30,32}
80	{25,30,32}

5-2. Evaluation criteria

Considering that the outcome of the M-O algorithms is usually an approximation of the Pareto-optimal set, an important issue is how to compare the performance of these algorithms with each other [52]. In this section, four criteria that are selected in this research for the performance evaluation of the proposed MOVNS are briefly described. Since the MO-MM-GTTDP has three objective functions, the set of effective Pareto solutions are assumed to be as follows: $(f_1^1, f_2^1, f_3^1), \dots, (f_1^n, f_2^n, f_3^n)$.

5-2-1. Maximum Spread Index (MSI)

Zitzler and Thiele [52] present the MSI for the first time in 1999. It shows the diversity of effective Pareto solutions. Obviously, the larger value of MSI indicates that solutions are scattered in a wider space, which shows the higher performance of the algorithm. This criterion is calculated using Eq (1).

$$D = \sqrt{\sum_{j=1}^{k}\left(f_j^{max} - f_j^{min}\right)^2} \qquad (1)$$

In this equation, K is the number of objective functions, f_j^{max} and f_j^{min} are the highest and lowest values obtained by the algorithm for the j_{th} objective function.

5-2-2. Spacing Index (SI)

The second evaluation criterion measures the uniformity of the non-dominated solutions in the solution space. To calculate this measure, first, the minimum value of the Euclidean distance between every pair adjoining Pareto solutions in the solution space is calculated by equation (2).

$$d_i = \min_{j=1,\dots,n\ j\neq i}\left(\sum_{k=1}^{k}\left|f_k^i - f_k^j\right|\right), \qquad \forall\ i = 1,\dots,n \qquad (2)$$

In equation (2), n is the number of Pareto solutions, K is the number of objective functions, and f_k^i is the amount of k_{th} objective function in the ith solution. Then, the SI value is obtained by Eq (3) as follows:

$$SI = \sqrt{\frac{1}{(n-1)}\left(\sum_{i=1}^{n}\left(\bar{d} - d_i\right)^2\right)} \qquad (3)$$

Where, $\bar{d}$ is the average of d_i's. The lower this criterion, the better the uniform distribution of the solutions.

5-2-3. Number of Pareto Solution (NPS)

The third criterion shows the number of approximate non-dominated Pareto solutions that evaluate the performance of the algorithm. Obviously, having a higher value for the NPS means the higher quality of the M-O algorithm.

5-2-4. Cpu-time

The last performance evaluation criterion is the computational time of the algorithm. For NP-hard problems, it is an essential criterion for evaluating the solution methodology, specifically for the desired application of the MO-MM-GTTDP in tourist trip planning.

5-3. Taguchi-based Parameters tuning

One of the important factors that may have an impact on the efficiency of the meta-heuristic algorithm is the predetermined value of parameters of the algorithm. Taguchi experimental design is a well-known method for achieving a good value for parameters of meta-heuristics without performing the full factorial possible combinations of experiments. This method is on the basis of two crucial factors: the orthogonal array (OA) and the signal to noise (S/N) ratio. The OA represents a matrix that is carrying the experimental scheme depend on different factors' levels [53]. In the S/N ratio, S and N represent the term ''signal'' as a useful value and ''noise'' as an unpleasant value, respectively. The Taguchi method tends to minimize the variation (S/N ratio) by minimizing the effect of noise and find the best level of the influential controllable parameter based on the robustness [54]. The first step in using this method is to determine the controllable parameters of the proposed algorithm. A positive side of the proposed MOVNS is that there are only two parameters to be set before running the algorithm: the iteration number and the alpha coefficient. As described in Section 3.d, the Alpha coefficient is a constant value that has a direct impact on the number of removable nodes as well as changeable transportation modes during the shaking phase of the algorithm. Four levels are considered for each of these two parameters, which are presented in Table 3. These values are selected based on some preliminary experiments.

Table 3. Different levels of MOVNS parameters used for turning

Factors (MOVNS parameters)	**Levels**			
	1	**2**	**3**	**4**
Iteration number	20	50	70	100
Alpha coefficient	0.2	0.4	0.5	0.7

To tune the parameters of the proposed method, above-described evaluating criteria (MSI, SI, and NPS) are used as responses of each experiment design to determine the most appropriate level for each parameter. To do these analyses, the Minitab software is used. It should be noted that all these experiments are implemented on three randomly selected instances with different problem sizes. Before this selection, all generated instances are classified in to three categories small, medium, and large size according to the number of POIs and the available time budget. Table 4 shows the classified instances and the ones selected for the parameter tuning. Results obtained from selected instances are normalized and averaged before applying the Taguchi method. Figures 7, 8 and 9 illustrate the results of the parameter tuning obtained from the Minitab. According to the S/N ratio plots (Figures 7-9), it is inferred that the best value of the iteration number and the alpha coefficient for small instances are 50 and 0.5, for average instances are 100 and 0.5, and for big instances are 100 and 0.7 respectively.

Table 4. Selected instances for parameters tuning

Instance's size	Instance number	Instance name
small	1	**(5,20)**
	2	**(5,30)**
	3	**(5,40)**
	4	**(10,20)**
	5	**(10,30)**
	6	**(10,40)**
	7	**(15,30)**
	8	**(15,40)**
Average	9	**(10,50)**
	10	**(15,50)**
	11	**(15,60)**
	12	**(15,70)**
	13	**(20,30)**
	14	**(20,40)**
	15	**(20,50)**
	16	**(20,60)**
	17	**(25,30)**
	18	**(25,40)**
	19	**(32,30)**
Big	20	**(20,70)**
	21	**(20,80)**
	22	**(25,50)**
	23	**(25,60)**
	24	**(25,70)**
	25	**(25,80)**
	26	**(32,40)**
	27	**(32,50)**
	28	**(32,60)**
	29	**(32,70)**
	30	**(32,80)**

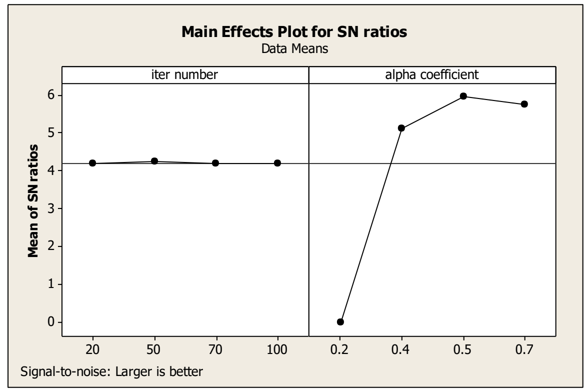

Fig 7. S/N ratio plot of Taguchi for **small** instances

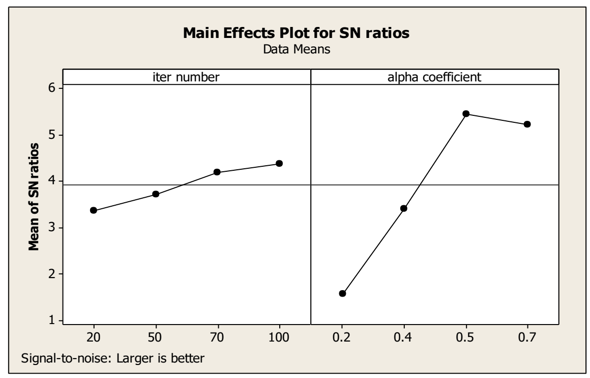

Fig 8. S/N ratio plot of Taguchi design for **medium** instances

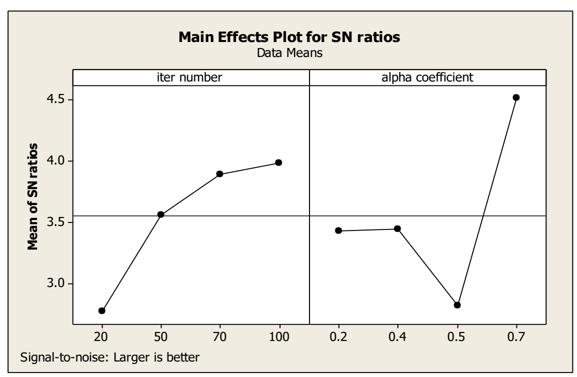

Fig 9. S/N ratio plot of Taguchi design for **large** instances

5-4.Results and discussion

After parameter tuning, all mentioned instances are solved using both implemented methods. All tests were conducted on an Intel core i7 PC with 3.1 GHz CPU and 8.00 GB RAM. The results of solving instances by presented the MOVNS algorithm as well as the ε –constraint method are summarized in Table 5. The first column in this table represents the instance's number. The second column gives the instance's name, which includes the number of POIs and the available time budget . Then the value of each evaluation criterion is presented for MOVNS and ε –constraint method, respectively.

Table 5. Results obtained from the MOVNS and the ε –constraint method

Instance number	Instance name	MOVNS				ε -Constraint			
		MSI	SI	Cpu-time(S)	NPS	MSI	SI	Cpu-time(S)	NPS
1	(5,20)	0	-	0.001	1	0	-	85.24	1
2	(5,30)	97.73	0	0.001	2	97.73	0	95.71	2
3	(5,40)	179.96	8.28	0.002	3	179.96	8.28	92.72	3
4	(10,20)	156.70	0	0.007	3	156.70	0	92.08	2
5	(10,30)	388.52	27.79	0.022	6	392.47	21.52	93.83	4
6	(10,40)	672.05	24.77	0.249	27	672.05	70.54	100.28	8
7	(15,30)	365.15	31.64	0.088	7	370.21	47.11	110.56	5
8	(15,40)	513.65	12.45	0.517	22	513.65	45.07	194.46	9
9	(10,50)	753.19	13.60	0.909	41	753.19	29.90	123.62	12
10	(15,50)	553.64	19.39	1.475	35	513.65	45.07	194.46	9
11	(15,60)	754.56	11.21	2.234	61	665.32	22.33	434.88	13
12	(15,70)	1041.5	20.51	4.332	64	847.36	32.52	1306.06	15
13	(20,30)	407.17	10.09	0.959	13	1120.2	37.88	2413.98	16
14	(20,40)	610.55	21.22	1.860	30	483.75	44.62	138.61	8
15	(20,50)	762.13	15.52	3.521	49	660.42	35.28	388.62	11
16	(20,60)	968.99	10.30	6.579	78	893.36	38.26	3011.53	16
17	(25,30)	244.95	8.877	1.671	14	297.28	29.19	176.48	8
18	(25,40)	533.32	11.45	4.465	41	591.16	23.34	1791.65	13
19	(32,30)	420.54	31.02	2.382	13	444.38	46.42	445.26	9
20	(20,70)	1223.1	14.62	10.194	94	1243.8	36.31	3612.24	16
21	(20,80)	1280.6	8.866	10.580	107	1296.9	40.69	2596.27	17
22	(25,50)	691.17	23.37	4.528	41	811.55	29.76	2833.21	14
23	(25,60)	967.35	8.44	13.512	87	976.88	23.30	3619.26	18
24	(25,70)	1201.6	7.864	19.558	121	1185.5	30.09	7201.46	17
25	(25,80)	1423.8	11.79	28.372	155	1304.3	35.51	7236.48	17
26	(32,40)	580.35	14.52	6.754	31	523.26	10.68	1799.14	12
27	(32,50)	668.61	9.775	12.891	58	637.25	21.64	3626.04	15
28	(32,60)	891.07	10.29	22.542	86	906.86	33.89	14410.4	19
29	(32,70)	1159.9	10.09	31.083	123	1050.8	34.01	18678.4	19
30	(32,80)	1196.3	9.184	39.390	153	1332.4	24.71	21698.8	20

A comparison of the results of both algorithms based on the MSI criterion is presented in Figure 10. According to this figure, the value of the MSI in the results of the MOVNS is approximately the same as the epsilon-constraint method for most of the instances. More precisely, as expected, in few instances with less time budget, the e-constraint method provides Pareto solutions with more extensive solution space, but in instances with 25 POIs or higher, the presented MOVNS algorithm represents a higher value of MSI. Generally, according to the obtained MSI criterion over the results of all instances, it can be concluded that our MOVNS algorithm has acceptable performance to provide Pareto solutions with wide solution space.

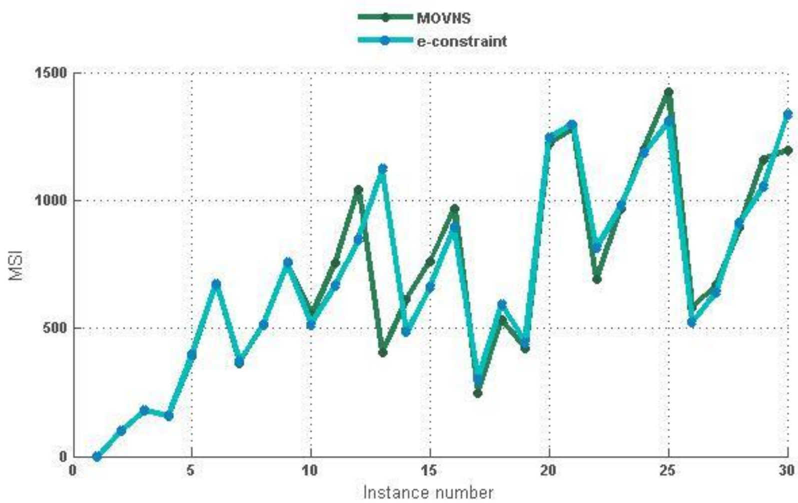

Fig 10. Comparing MOVNS algorithm with ε-constraint method by MSI criterion

Figure 11 illustrates the uniformity of the set of Pareto solutions obtained by each of the two methods. As can be seen, except for two instances, the MOVNS is superior in terms of spacing index criterion.

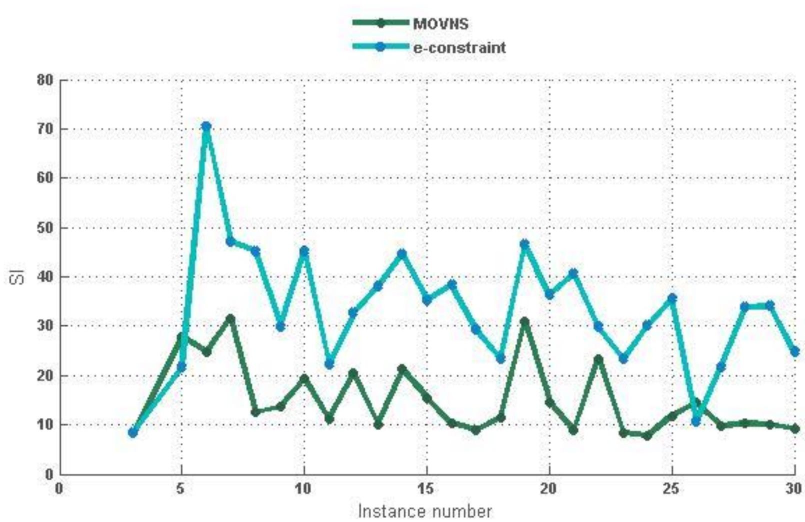

Fig 11. Comparing MOVNS algorithm with ε-constraint method by SI criterion.

Figure 12 shows the number of Pareto solutions obtained by each of the implemented algorithms. It is worth mentioning that this measure is relatively unfair because we fixed the maximum number of Pareto solutions that can be obtained by the $\varepsilon - \text{constraint}$ method on 20. However, the comparing procedure in some instances that have less than 20 Pareto solutions in the $\varepsilon - \text{constraint}$ method is considerable.

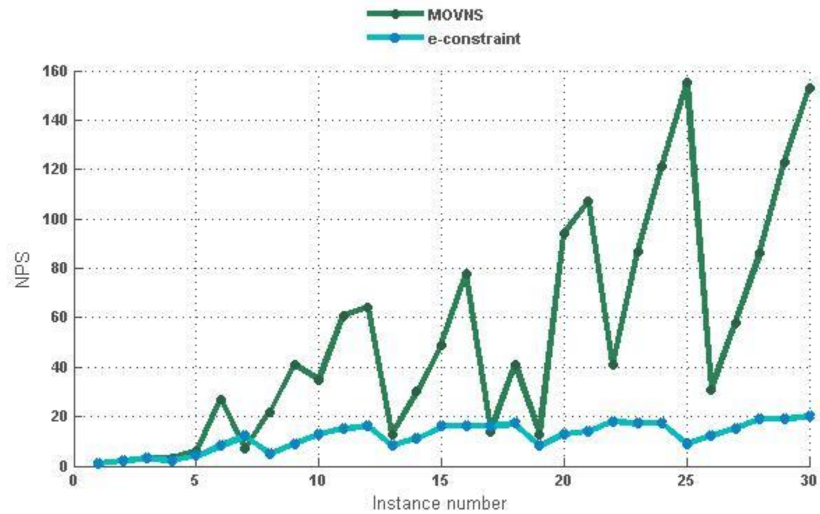

Fig 12. Comparing MOVNS algorithm with ε-constraint method by NPS criterion.

The computational time of both algorithms is compared over the generated instances in Figure 13. As the problem size increases, the Cpu-time of the $\varepsilon - \text{constraint}$ method has become more than 4 hours. Consequently, the exact algorithm is not applicable to real world problems.

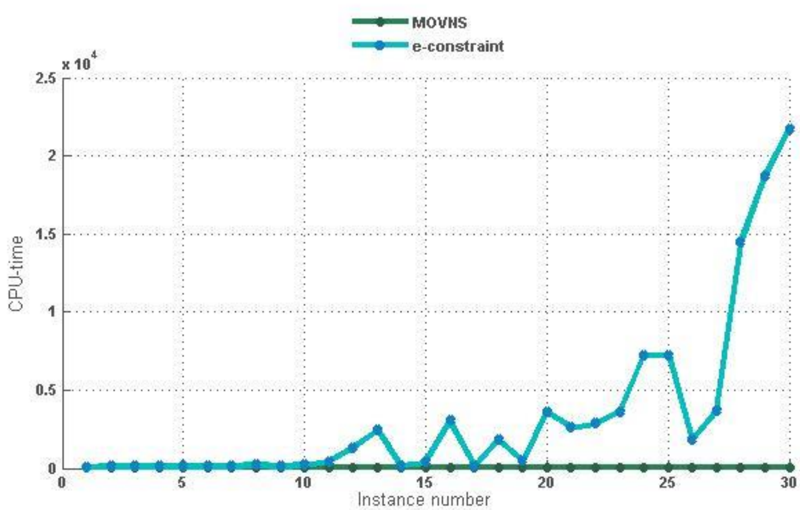

Fig 13. Comparing MOVNS algorithm with ε-constraint method by Cpu-time (S).

6- Case study

In this section, the proposed solution approach has been tested over a real dataset obtained from the city of Tehran. Tehran is the capital of Iran and one of the main important touristic places in the country. It has numerous famous touristic attractions. According to available data from 2018, annually, 1,200,000 foreign tourists visit Tehran. As a small test set, 10 POIs are selected from all the existing POIs in Tehran. These POIs are chosen from four different categories: park and green spaces, handicraft centers, museums, and shopping centers. Table 6 presents the name of each of these POIs as well as their visiting cost. The corresponding score for each POI and are generated randomly in the range of [0, 10]. The available time budget for the tour is considered 4 hours. Due to the available transportation facilities in Tehran, we proposed three transportation modes to the tourist for moving between each pair of POIs: the Subway, BRT[3], and Taxi. All the data related to the distance and available transportation modes between all pairs of nodes, the cost of traveling as well as the amount of co2 emission which is produced by each mode are shown in Table 7. Time values are the average value over different times of the day that obtained using Google Map to relatively considering the traffic. Moreover, the amount of produced co2 emission is based on the available information on the [www.co2nnect.org] that is 0.069 kg/km for BRT, 0.17 kg/km for Taxi, and 0 kg/km for Subway per person.

Table 6. Information table of POIs

PIOs number	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Name	Mellat park	Saei park	Qeytarieh park	Darakeh	Carpet museum	City theatre	Tajrish bazaar	Tabiat bridge	Saadabad complex	Baghe ferdows garden
Visiting Cost(T)	0	0	0	0	2500	0	0	0	16000	0
Score	0	3	5	8	10	3	7	6	10	0

Table 7. Information about distances and transportation facilities

Initial node	**Destination node**	**Distance(km)**	**Transportation facility number**	**Transportation facility mode**	**Moving time(Min)**	**Moving cost(T)**	**Co2 emission coefficient(Kg/Km)**
1	2	6,53	0	BRT	16	400	0.069
			1	Taxi	-	-	0
			2	Subway	17.5	19000	0.17
1	3	8.75	0	BRT	-	-	0.069
			1	Taxi	-	-	0
			2	Subway	17.5	12000	0.17
…	…	…	…	…	…	…	…
2	5	5.05	0	BRT	25	900	0.069
			1	Taxi	15	1000	0
			2	Subway	19	10000	0.17
2	6	6.83	0	BRT	11	400	0.069
			1	Taxi	9	1000	0
			2	Subway	21	10000	0.17
…	…		…	…	…	…	…
9	10	2.86	0	BRT	-	-	0.069
			1	Taxi	-	-	0
			2	Subway	10.5	7000	0.17

The presented MOVNS approach is applied to solve the real data instance. The results contain 46 non-dominated solutions. All of these non-dominated solutions are illustrated in Figure 14. For some of these non-dominated solutions, more details, including the proposed route, are presented in Table 8.

[3] Bus rapid transit

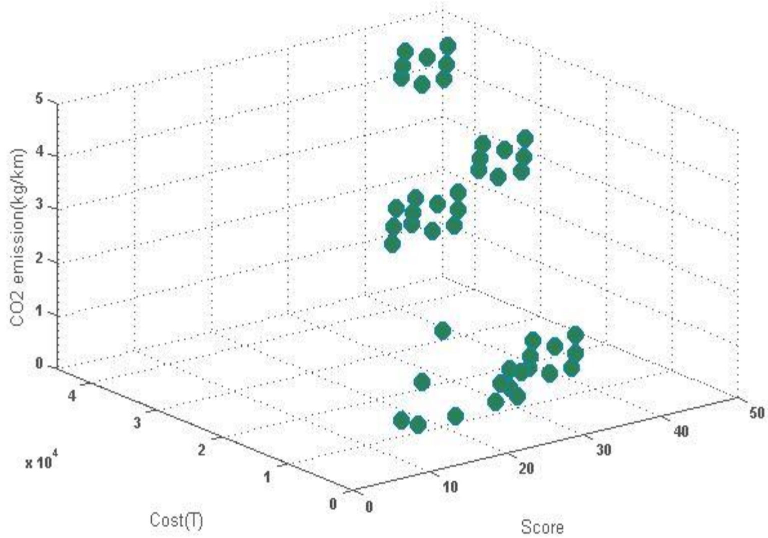

Fig 14. The position of 46 non-dominated solutions in the solution space

Table 8. Results of the proposed tour for some of the non-dominated solutions

Solution number	**Objective Value**			**Proposed order POIs**	**Proposed transportation facilities**
	Z_3	Z_2	Z_1		
1	0.79557	1900	15	1-2-3-7-10	0-1-1-0
2	0.79557	1800	10	1-2-7-10	0-1-0
6	3.55477	47100	52	1-2-5-6-8-3-7-9-10	0-1-1-1-1-2-2-2
7	1.19577	2000	21	1-6-8-3-7-10	0-1-1-1-0
14	1.98222	34900	35	1-2-5-3-7-9-10	0-0-1-1-2-2
17	3.80455	46500	46	1-5-8-3-7-9-4-10	0-1-1-1-2-2-2
21	1.88355	35500	38	1-5-8-3-7-9-10	0-1-1-1-2-2
22	1.44279	5900	34	1-2-5-6-8-3-7-9-10	0-0-0-1-1-1-0
30	1.93254	35600	44	1-2-5-6-8-3-7-9-10	0-1-0-1-1-1-2-2
36	1.09434	6000	34	1-2-5-6-8-3-7-9-10	0-1-0-1-1-1-0

As can be seen in Figure 14, all Pareto solutions are clustered in three groups displaying the impact of each objective function. This means that one group members have higher scores than the other groups. The second and third groups have lower travel costs as well as less pollution produced during the tour. Therefore, it can be concluded the proposed solution method (MOVND) has acceptable performance to provide Pareto solutions with considering all the aspects of the problem.

6-1. Selecting routes among the non-dominated solution

Considering the high number of obtained non-dominated solutions, choosing one best tour for the tourist may still be a difficult task. Selecting a solution among the non-dominated ones is itself a challenging problem in the area of M-O optimization (MOP), which is not in the scope of the current research. However, to make the procedure of tour recommendation more practical, in the rest of this section, a simple but practical multi-attribute decision-making (MADM) technique is used to indicate which solutions might be more interesting for the tourist among the 46 obtained solutions. As a result, at the end of this section, three routes are proposed to the tourist with considering his/her preferences.

In the first step, the existing criteria are weighed with the paired comparison method. This paired comparison method is introduced by Saaty [55] as one of the high performance multi-criteria decision-making techniques in 2008. Table 9 shows the paired comparison matrix between every two criteria, which is filled with the 9 scale standard range of the pairwise comparison. It should be noted that in this case study, the paired comparison matrix is filled artificially. In the second step, we rank all the obtained solutions by using the TOPSIS[4] method. TOPSIS is one of the MADM methods developed originally by Ching-Lai Hwang and Yoon [56] in 1981. This method is one of the most common multi-criteria decision-making techniques, which compares a set of alternatives by identifying weights for each criterion, normalizing scores for each criterion, and calculating the geometric distance between each alternative and the ideal alternative. The ideal alternative in this method is the one with the highest score in every criterion [57]. As can be imagined, this ideal alternative does not necessarily exist among the list of alternatives. For more details on the TOPSIS method, one may refer to the research by Triantaphyllou et al. [58].

Table 9. The paired comparison matrix of three proposed objective functions

criteria	Z_1	Z_2	Z_3
Z_1	1	3	1/5
Z_2	1/3	1	1/3
Z_3	5	3	1

In the current case study, all 46 non-dominated solutions are considered as alternatives in the TOPSIS method, numbered from A_1 to A_{46}. Each alternative A_i has three criteria value corresponding to each objective value of Z_1, Z_2, and Z_3. The final rank of each non-dominated solution depends on the paired comparison matrix. In practice, this matrix can be generated according to the tourist preferences over objective functions.

In the presented matrix, the priority is given to the tourist desirability by maximizing the total score. Then, the environmental aspect of the tour is considered with a lower weight. The minimum weight is given to the travel cost. Indeed, these decisions may have an impact on the selection of transportation modes. Applying the TOPSIS method, the normalized vector of weights for criteria is found as $[0.222518, 0.126834, 0.650648]$. The weighted normalized decision matrix (V) is then calculated by multiplying the vector W by the decision matrix. Equations 5 and 6 show the positive and the negative ideal alternatives obtained from the V matrix, respectively.

$$A^{+} = [0.047109,\ 0.000537,\ 0.029418] \tag{5}$$

[4] The Technique for Order of Preference by Similarity to Ideal Solution

$A^- = [0.006342, 0.031641, 0.159917]$ (6)

Variables (d_i^-) and (d_i^+) are then calculated as the measures of the distance of each alternative A_i (each element of V matrix) from the negative and the positive ideal solutions, respectively. The relative closeness from the negative ideal for each alternative is calculated by Eq(7).

$$CL_i = {d_i^-}/{d_i^- + d_i^+} \quad (7)$$

Then, all 46 alternatives are ranked based on the corresponding CL_i value. Table 10 represents the three best non-dominated solutions after this ranking. These three solutions are graphically presented in Figures 15 - 17.

Table 10. Three selected non-dominated solutions

Selected Solution	Objective Value			Proposed order of POIS	Proposed transportation facilities	CL_i	rank
	Z_3	Z_2	Z_1				
8	0.79557	3000	24	1-2-6-8-3-7-10	0-1-1-1-1-0	0.890013	1
27	0.79557	6400	31	1-2-5-3-7-10	0-1-1-1-0	0.874468	2
31	3.85354	46600	52	1-2-5-6-8-3-7-9-4-10	0-1-0-1-1-1-2-2-2	0.862061	3

Fig 15. The first non-dominated solution illustration

Fig 16. The second non-dominated solution illustration

According to the TOPSIS method, Figure 15 shows the route with the first rank. As we see in Table 10, this solution has the lowest value of traveling cost. Considering the Paired Comparison matrix of three objective functions, the Taxi is not proposed for moving between POIs in this route. Figure 16, 17 represents the second and third route, respectively. The second route has more score by suggesting a POI with more visiting cost as well as score. It should be mentioned that the total produced pollutant is the same in these two routes, Because of the facility mode, which produced more pollutants has a higher moving cost per unit of distance. In the third route, the value of total score (Z_1) is very different from two other routes. Obviously, as the value of the first objective function increases, the total cost and the produced pollutant are also increased. We can conclude from the obtained solutions that all the three aspects of the problem have been considered. Then, based on the tourist's priorities, the three best routes are presented.

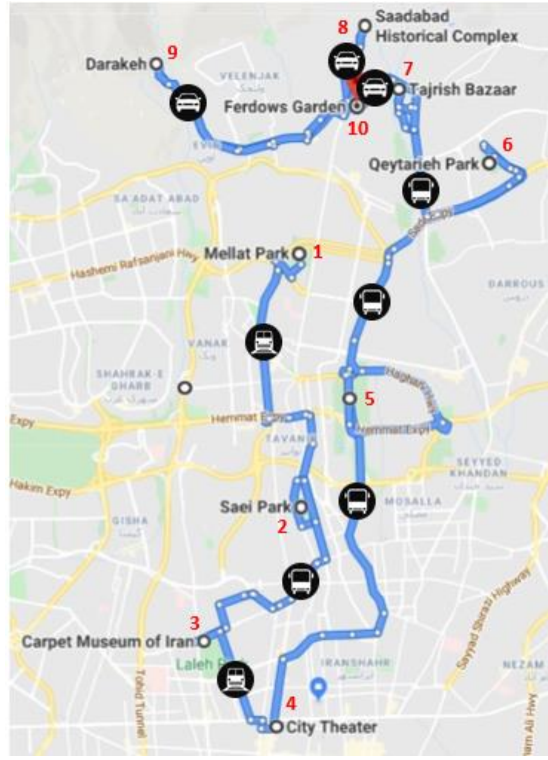

Fig 17. The third non-dominated solution illustration

7- Conclusion and future work

In this paper, the Multi Objective Multi Modal variant of the tourist trip design problem is introduced, which considers the environmental aspect in tour planning. The goal of the MO-MM-GTTDP is to determine a tour of the maximal score with minimal cost as well as produced co_2 emissions simultaneously. This tour presents a connected number of POIs and different types of transportation facilities to tourists for moving between them for one trip in one day. This problem has many practical applications as a sustainable model, but it is important even challenging to solve it to present an effective Pareto solution in the shortest time as possible.

We propose a MOVNS algorithm to cope with this problem. On the basis of the tradeoff between improving three objective functions, to generate combinations of the selected POIs and the transportation facility modes, high quality non-dominated solutions are obtained in short computation times. Also, we solve the problem by implementing it in CPLEX using the ε-constraint method as an exact solution approach. We use our 30 generated instances based on benchmark OP instances and compare two methods by using four M-O evaluating criteria. The results represent the efficiency of our proposed meta-heuristic algorithm.

Furthermore, we implemented the proposed MOVND algorithm on a small real case in Tehran to prove the applicability of the method. In addition, we used the TOPSIS method as an effective multi-criteria decision-making approach to summarize and suggest the three best tours to the tourist.

This model can be more realistic by considering time windows or the time dependent travel time using various transportation facilities in future works. A considerable challenge would be dealing with problems with multiple tours. Moreover, presenting a comprehensive algorithm which combines MOVNS approach with a multi-criteria decision-making method to simplify the selection for tourist can be considered in a future research.

Authorship contributions

Conceptualization: [Ali Divsalar, Armin Jabbarzadeh]; Methodology: [Ghazaleh Divsalar, Ali Divsalar]; Formal analysis and investigation: [Ghazaleh Divsalar, Ali Divsalar]; Writing - original draft preparation: [Ghazaleh Divsalar]; Writing - review and editing: [Ali Divsalar, Armin Jabbarzadeh, Hadi Sahebi]; Supervision: [Ali Divsalar, Armin Jabbarzadeh, Hadi Sahebi];

Declrations

- No funding was received to assist with the preparation of this manuscript.

- The authors have no conflicts of interest to declare that are relevant to the content of this article.

References

[1] Gautam V. An empirical investigation of consumers' preferences about tourism services in Indian context with special reference to state of Himachal Pradesh. Tourism Management. 2012 Dec 1;33(6):1591-2.

[2] Cucculelli M, Goffi G. Does sustainability enhance tourism destination competitiveness? Evidence from Italian Destinations of Excellence. Journal of Cleaner Production. 2016 Jan 16;111:370-82.

[3] Agaraj X, Murati M. Tourism an important sector of economy development. Annals-Economy Series. Constantin Brancusi University, Faculty of Economics, 2009;1:83-90. https://ideas.repec.org/a/cbu/jrnlec/y2009v1p83-90.html

[4] Divsalar GH, Jabbarzadeh A, Divsalar A, Sahebi H, A multi-objective approach for the multi-modal green tourist trip design problem, 14th Iranian International Industrial Engineering Conference, 2017.

[5] Liao Z, Zheng W. Using a heuristic algorithm to design a personalized day tour route in a time-dependent stochastic environment. Tourism Management. 2018 Oct 1;6:284-300.

[6] Wong UI. Buddhism and Tourism at Pu-Tuo-Shan, China (Doctoral dissertation, University of Waikato),2011.

[7] Leiper N. Tourist attraction systems. Annals of tourism research. 1990 Jan 1;17(3):367-84.

[8] Tsai CY, Chung SH. A personalized route recommendation service for theme parks using RFID information and tourist behavior. Decision Support Systems. 2012 Jan 1;52(2):514-27.

[9] Vansteenwegen P, Van Oudheusden D. The mobile tourist guide: an OR opportunity. OR insight. 2007 Jul 1;20(3):21-7.

[10] Vansteenwegen P, Souffriau W, Berghe GV, Van Oudheusden D. Metaheuristics for tourist trip planning. In Metaheuristics in the service industry. Springer, Berlin, Heidelberg. 2009 (pp. 15-31).

[11] Hsu FM, Lin YT, Ho TK. Design and implementation of an intelligent recommendation system for tourist attractions: The integration of EBM model, Bayesian network and Google Maps. Expert Systems with Applications. 2012 Feb 15;39(3):3257-64.

[12] Lee CS, Chang YC, Wang MH. Ontological recommendation multi-agent for Tainan City travel. Expert Systems with Applications. 2009 Apr 1;36(3):6740-53.

[13] Liu L, Xu J, Liao SS, Chen H. A real-time personalized route recommendation system for self-drive tourists based on vehicle to vehicle communication. Expert Systems with Applications. 2014 Jun 1;41(7):3409-17.

[14] Rodríguez B, Molina J, Pérez F, Caballero R. Interactive design of personalised tourism routes. Tourism Management. 2012 Aug 1;33(4):926-40.

[15] Zheng W, Liao Z, Qin J. Using a four-step heuristic algorithm to design personalized day tour route within a tourist attraction. Tourism Management. 2017 Oct 1;62:335-49.

[16] Becken S. How tourists and tourism experts perceive climate change and carbon-offsetting schemes. Journal of Sustainable Tourism. 2004 Jul 1;12(4):332-45.

[17] UNWTO., 2003. 1st Conference on Climate Change and Tourism. Available online: http://sdt.unwto.org/event/1st-conference-climate-changeand-tourism.

[18] Feillet D, Dejax P, Gendreau M. Traveling salesman problems with profits. Transportation science. 2005 May;39(2):188-205.

[19] Gunawan A, Lau HC, Vansteenwegen P. Orienteering problem: A survey of recent variants, solution approaches and applications. European Journal of Operational Research. 2016 Dec 1;255(2):315-32..

[20] Divsalar A, Vansteenwegen P, Cattrysse D. A variable neighborhood search method for the orienteering problem with hotel selection. International Journal of Production Economics. 2013 Sep 1;145(1):150-60.

[21] Vincent FY, Jewpanya P, Ting CJ, Redi AP. Two-level particle swarm optimization for the multi-modal team orienteering problem with time windows. Applied Soft Computing. 2017 Dec 1;61:1022-40.

[22] Gavalas D, Konstantopoulos C, Mastakas K, Pantziou G. A survey on algorithmic approaches for solving tourist trip design problems. Journal of Heuristics. 2014 Jun 1;20(3):291-328.

[23] Bérubé JF, Gendreau M, Potvin JY. An exact ϵ-constraint method for bi-objective combinatorial optimization problems: Application to the Traveling Salesman Problem with Profits. European journal of operational research. 2009 Apr 1;194(1):39-50.

[24] Schilde M, Doerner KF, Hartl RF, Kiechle G. Metaheuristics for the bi-objective orienteering problem. Swarm Intelligence. 2009 Sep 1;3(3):179-201.

[25] Doerner K, Gutjahr WJ, Hartl RF, Strauss C, Stummer C. Pareto ant colony optimization: A metaheuristic approach to multiobjective portfolio selection. Annals of operations research. 2004 Oct 1;131(1-4):79-99.

[26] Chao IM, Golden BL, Wasil EA. A fast and effective heuristic for the orienteering problem. European journal of operational research. 1996 Feb 8;88(3):475-89

[27] Chen YH, Sun WJ, Chiang TC. Multiobjective orienteering problem with time windows: An ant colony optimization algorithm. InTechnologies and Applications of Artificial Intelligence (TAAI), 2015 Conference on 2015 Nov 20 (pp. 128-135). IEEE.

[28] Martin-Moreno, R., & Vega-Rodriguez, M. A. (2018). Multi-objective artificial bee colony algorithm applied to the bi-objective orienteering problem. Knowledge-Based Systems, 154, 93-101.

[29] Zheng W, Liao Z. Using a heuristic approach to design personalized tour routes for heterogeneous tourist groups. Tourism Management. 2019 Jun 1;72:313-25.

[30] Kantor MG, Rosenwein MB. The orienteering problem with time windows. Journal of the Operational Research Society. 1992 Jun 1;43(6):629-35.

[31] Vansteenwegen P, Souffriau W, Van Oudheusden D. The orienteering problem: A survey. European Journal of Operational Research. 2011 Feb 16;209(1):1-0.

[32] Fomin FV, Lingas A. Approximation algorithms for time-dependent orienteering. Information Processing Letters. 2002 Jul 31;83(2):57-62.

[33] Mei Y, Salim FD, Li X. Efficient meta-heuristics for the multi-objective time-dependent orienteering problem. European Journal of Operational Research. 2016 Oct 16;254(2):443-57

[34] Dutta J, Barma PS, Mukherjee A, Kar S, De T. A multi-objective open set orienteering problem. Neural Computing and Applications. 2020 Mar 3:1-7

[35] Archetti C, Carrabs F, Cerulli R. The set orienteering problem. European Journal of Operational Research. 2018 May 16;267(1):264-72.

[36] Divsalar A, Vansteenwegen P, Sörensen K, Cattrysse D. A memetic algorithm for the orienteering problem with hotel selection. European Journal of Operational Research. 2014 Aug 16;237(1):29-49.

[37] Chao IM, Golden BL, Wasil EA. The team orienteering problem. European journal of operational research. 1996 Feb 8;88(3):464-74.

[38] Vansteenwegen, P., Souffriau, W., Berghe, G. V., & Van Oudheusden, D. (2009). Iterated local search for the team orienteering problem with time windows. Computers & Operations Research, 36(12), 3281-3290.

[39] Tsakirakis E, Marinaki M, Marinakis Y, Matsatsinis N. A similarity hybrid harmony search algorithm for the Team Orienteering Problem. Applied Soft Computing. 2019 Jul 1;80:776-96.

[40] Gunawan A, Lau HC, Vansteenwegen P, Lu K. Well-tuned algorithms for the team orienteering problem with time windows. Journal of the Operational Research Society. 2017 Aug 1;68(8):861-76.

[41] Hu W, Fathi M, Pardalos PM. A multi-objective evolutionary algorithm based on decomposition and constraint programming for the multi-objective team orienteering problem with time windows. Applied Soft Computing. 2018 Dec 1;73:383-93

[42] Hapsari I, Surjandari I, Komarudin K. Solving multi-objective team orienteering problem with time windows using adjustment iterated local search. Journal of Industrial Engineering International. 2019 Dec 1;15(4):679-93.

[43] Saeedvand S, Aghdasi HS, Baltes J. Novel hybrid algorithm for Team Orienteering Problem with Time Windows for rescue applications. Applied Soft Computing. 2020 Sep 5:106700.

[44] Abounacer R, Rekik M, Renaud J. An exact solution approach for multi-objective location–transportation problem for disaster response. Computers & Operations Research. 2014 Jan 1;41:83-93.

[45] Khalili-Damghani K, Abtahi AR, Tavana M. A new multi-objective particle swarm optimization method for solving reliability redundancy allocation problems. Reliability Engineering & System Safety. 2013 Mar 1;111:58-75.

[46] Mavrotas G. Effective implementation of the ε-constraint method in multi-objective mathematical programming problems. Applied mathematics and computation. 2009 Jul 15;213(2):455-65.

[47] Archetti C, Hertz A, Speranza MG. Metaheuristics for the team orienteering problem. Journal of Heuristics. 2007 Feb 1;13(1):49-76..

[48] Labadie N, Mansini R, Melechovský J, Calvo RW. The team orienteering problem with time windows: An lp-based granular variable neighborhood search. European Journal of Operational Research. 2012 Jul 1;220 (1):15-27.

[49] Duarte A, Pantrigo JJ, Pardo EG, Mladenovic N. Multi-objective variable neighborhood search: an application to combinatorial optimization problems. Journal of Global Optimization. 2015 Nov 1;63(3):515-36.

[50] Taguchi G, Chowdhury S, Wu Y. Taguchi's quality engineering handbook. Wiley; 2005.

[51] Tsiligirides T. Heuristic methods applied to orienteering. Journal of the Operational Research Society. 1984 Sep 1;35(9):797-809.

[52] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE transactions on Evolutionary Computation. 1999 Nov;3(4):257-71.

[53] Azadeh A, Elahi S, Farahani MH, Nasirian B. A genetic algorithm-Taguchi based approach to inventory routing problem of a single perishable product with transshipment. Computers & Industrial Engineering. 2017 Feb 1;104:124-33.

[54] Tsai JT, Ho WH, Liu TK, Chou JH. Improved immune algorithm for global numerical optimization and job-shop scheduling problems. Applied Mathematics and Computation. 2007 Dec 15;194(2):406-24.

[55] Saaty TL. Decision making with the analytic hierarchy process. International journal of services sciences. 2008 Jan 1;1(1):83-98.

[56] Hwang CL, Yoon K. Methods for multiple attribute decision making. In Multiple attribute decision making. Springer, Berlin, Heidelberg. 1981 (pp. 58-191).

[57] Rezki, H., & Aghezzaf, B. (2017, April). The bi-objective orienteering problem with budget constraint: GRASP_ILS. In 2017 International Colloquium on Logistics and Supply Chain Management (LOGISTIQUA) (pp. 25-30). IEEE.

[58] Triantaphyllou E. Multi-criteria decision making methods. InMulti-criteria decision making methods: A comparative study. Springer, Boston, MA. 2000 (pp. 5-21).