OPSEARCH (2023) 60:1703–1730
https://doi.org/10.1007/s12597-023-00678-5

APPLICATION ARTICLE

A new mathematical model and a heuristic algorithm for the tourist trip design problem under new constraints: a real-world application

Gulcin Dinc Yalcin[1] · Hilal Malta[1] · Seher Saylik[1]

Accepted: 27 June 2023 / Published online: 25 July 2023

Abstract
The tourist trip design problem (TTDP) is about to generate routes for tourists to maximize the points of interest (POIs) visited within specific time windows. In this study, new constraints: budget, weather and break are considered. First, the budget is required for entrance fees and the distance between two points where a taxi has to be used. Additionally, the expense of the break was taken into account. Then, the weather was considered for summer and for other seasons. On a summer day, tourists are likely to prefer visiting POIs, which are indoor areas, between specific times e.g. 11 a.m. to 3 p.m. to protect against the side effects of the sun. Furthermore, tourists need to take a break to relax during the trip. A mathematical model of the TTDP with these new constraints (TTDP-BWB) was developed. Then, a heuristic algorithm was developed with a new defined function that took the new constraints into account. The algorithm was codded using Android Studio and developed a mobile application for the case of Eskisehir in Türkiye. Problems are generated on the small and medium scale for the case of Eskisehir and used large-scale problems from published literature. The results of the algorithm were compared with the results of the mathematical model for the small scale problems. Additional, large-scale problems from literature were solved to see the performance of the heuristic algorithm. Computational results showed that the algorithm is promising.

Keywords Tourist trip design problem · Heuristic algorithm · Time windows · Mobile application · Mixed-integer programming

JEL Classification C61 · C63 · C88 · Z30

✉ Gulcin Dinc Yalcin
gdinc@eskisehir.edu.tr

[1] Department of Industrial Engineering, Faculty of Engineering, Eskisehir Technical University, Iki Eylul Campus, 26555 Eskisehir, Turkey

1 Introduction

Tourism plays an important role in the economy of countries. Approximately 1460 million international tourists contributed to tourism in 2019, and within 10 years, real growth in international tourism receipts is 54% while the growth in world gross domestic product (GDP) is 44% [46].

When tourists explore a city, firstly they should decide which points of interest (POIs) to visit. After that, they should plan their daily routes for the remaining days of their tour. This plan requires to learn some information, such as visiting times, entrance fees and time windows for each POI, and the time between POIs, etc. Additionally, they should take into consideration their personal limits, e.g. budget and time. This problem is called the tourist trip design problem (TTDP) in published literature (see e.g. [5, 13, 14, 16, 17, 26, 32, 33, 37]. and is an extension of the team orienteering problem (TOP) (see e.g. [4, 7, 8, 10, 36] that applies in the travel industry. The aim of the problem is to construct routes which start and end at a particular point, and the objective is to maximize the total score which are based on interest of a tourist visiting POIs. Also, the score is sum of points given by the tourist that shows how much the tourist keens on visiting the each POI.

In this study, TTDP was handled with three new constraints: budget, weather and break, and abbreviated the new problem to TTDP-BWB. Budget was considered to be only the entrance fee to POIs. However, tourists pay for transportation if the travel distance to a POI is too far to walk. It was assumed that tourists use taxis when it was impossible to walk. Thus, in this study, the standing taxi charge and taxi fare per kilometer were taken into consideration besides the entrance fee as individual parameters of the problem dealing with budget. Additionally, the expense of taking a coffee break, as explained later, was also taken into consideration.

Choosing the time to visit each POI was considered, especially those that are outdoors and affected by the weather. During the summer season, tourist may not prefer to be outside between specific times, e.g. between 11 a.m. and 3 p.m. due to heat. Thus, outdoor POIs should be visited outside of these times and inside POIs should be preferred instead. However, cold and rainy days are not handled in this study since it is the choice of the tourist whether to walk in the rain or not, and some of them may not care about being outside on these days.

After visiting a certain number of POIs, tourists need a break. In this study, it was focused on a coffee break (see e.g. [34], not a main meal, since the expense, time and duration resulting from the choice of a main meal may differ greatly from person to person. Thus, one coffee break within a limited duration was considered under a time window within a tolerance. Furthermore, an average expense for this break was added to the budget constraint.

Firstly, the current mathematical model of TTDP (see e.g. [23, 36]. was improved by adding new constraints or modifying some of the existing constraints and proposed a mathematical model. Additionally, new decision variables were defined to construct the constraints about weather and breakpoints, and defined new parameters for all new constraints.

Then, a heuristic algorithm was proposed with a newly defined function to solve the problem given the computational complexity of the TTDP-BWB mathematical model. Budget, time and the scores for POIs was put into the heuristic function to balance the objective function as well as the main constraints on time and budget in the other constraints: weather and breakpoints were also affected by them. The main advantages of the heuristic algorithm is to have an easy structure and to solve the problem within a reasonable time, in that any tourist does not want to wait too long to generate a route.

Finally, the algorithm was codded using Android Studio to develop a mobile application for tourists. This mobile application was mainly designed for the case of Eskisehir, Türkiye. Furthermore, small and medium problems were generated using the case of Eskisehir and large-scale problems from published literature. The results were compared with the mathematical model solutions where it was found within a reasonable time.

The main contributions of the paper can be summarized as follows:

- A new mathematical model of TTDP was proposed by considering new conditions: budget, weather, and break time.
- A heuristic algorithm where the function used for the selection of the next POIs consists of two significant parameters: the cost and the time needed to visit the next POI and scores of POIs given by tourists was proposed.

Addition, in this paper, the followings are applied for computational results of both real-life cases of Eskisehir, Türkiye and test problems from published literature.

- A mobile application was developed for the case of Eskisehir, Türkiye.
- Different small-scale test problems from the case of Eskisehir, Türkiye was proposed, and then the results of the heuristic algorithm were compared with the results of the mathematical model.
- The test problems from literature were solved to show the performance of the heuristic algorithm for large-scale problems.

The rest of the study is organized as follows: Sect. 2 reviews the literature on TTDP. The proposed mathematical model is given in Sect. 3, while Sect. 4 explains the heuristic algorithm. The computational results are then analyzed in Sect. 5. Finally, some implications of the study is given for Practice and Academicsin Sect. 6, and Sect. 7 draws some conclusions from the study.

2 Literature review

2.1 Team orienting problem

This section was started considering state-of-the-art problems with TOP – the team orienteering problem – which is the most related topic, since TTDP is an extension of that problem. TOP [8] itself is an extension of the orienteering problem (OP) for multiple tours, while OP is a special type of traveling salesman problem for the situation where all points are not visited and where the aim is to maximize profit [9, 24]. Recently, several methods such as a tabu search ([7, 19], a guided local search metaheuristic [38], a memetic algorithm [6], particle swarm optimization [10, 28] have been presented to solve the TOP. Moreover, Archetti et al. [3] proposed a new version of TOP in which the nodes are located on an arc (TOARP). Then, several methods have been proposed in published literature: a metaheuristic method [4] and a column generation approach [29].

Addition, TOP has been studied with considering time windows and named as TOP with time windows which is the most related version of TOP related to TTDP. Metaheuristic approaches such as iterated local search [39], an ant colony system [27], an lp-based granular variable neighborhood search [23], a simulated annealing heuristic [25] and an iterative three-component heuristic [18] were developed in the literature. Furthermore, we refer the reader to the survey about different types of OP and TOP by Vansteenwegen et al. [41] and Gunawan et al. [17].

2.2 Tourist trip design problem (TTDP)

TTDP is considered as an application of TOP and Vansteenwegen and Van Oudheusden [37] firstly used operations research (OR) techniques to model the TTDP, and proposed decision models. Then, Souffriau et al. [32] proposed using a combination of artificial intelligence and metaheuristic (guided local search) to solve a simple version of TTDP by only considering the maximum allowed distance, and the algorithm was applied to the case of the city of Ghent. After that, Souffriau et al. [33] took the opening hours of POIs and model TTDP as an extension of TOP with time windows. Additionally, an iterated local search algorithm (ILS) on a mobile phone was proposed. Also, we refer the reader to the survey of the metaheuristics for TTDP by Vansteenwegen et al. [40]. Similarly, Gavalas et al. [13] proposed a novel heuristic and develop a mobile application for TTDP. Brito et al. [5] proposed a fuzzy GRASP algorithm to solve TTDP. Mei [26] developed an ant colony algorithm under the Android platform.

Furthermore, TTDP was improved with additional new constraints, new algorithms, and various applications. Vansteenwegen et al. [42], developed a web application for five cities in Belgium. A greedy adaptive search algorithm (GRASP) was proposed to solve TTDP with the new constraint: lunch time, while a mathematical model was not included. Abbaspour and Samadzadegan [1] dealed TTDP with several models of transportation system: bus, subway, walking, and propose a mathematical model and two adapted genetic algorithms. The data was obtained from the city of Tehran. Garcia et al. [12] taken into account the transportation as walking and public transport and develop an iterated local search algorithm to solve TTDP in real-real time. Computational results were obtained by using the data from the city of San Sebastian. In addition, we refer the reader to the survey about TTDP comparing with OP and TOP by Gavalas et al. [14].

Moreover, Gavalas et al. [15] considered time dependency between POIs and proposed two novel randomized metaheuristic approaches based on ILS. Zheng et al. [43] proposed a four-step heuristic algorithm to solve TTDP with aesthetic fatigue and variable sightseeing value, and applied data from the Jiuzhali Valley in Sichuan, China. Kotiloglu et al. [21] developed a novel filter-first, tour-second framework with an iterated tabu search algorithm for TTDP that had a set of mandatory points selected by the tourist. Sylejmani et al. [35] extend TTDP for tourist groups by considering individual preferences and mutual social relationship. Then a mathematical model was developed and an algorithm based on tabu search. Wörndl et al. [47] recommended TTDP for a walking route and modified Dijkstra's algorithm to find the shortest route and maximize the entertainment. Expósito et al. [11] clustered POIs to represent different types of attraction sites and then a fuzzy GRASP algorithm was proposed using distance- and score-based evaluation criteria. Ko et al. [20] considered tourist fatigue in the objective function and developed an ILS.

Recently, Zheng et al. [44] improved TTDP by adding hotel selection and proposed a two-level heuristic approach that combined a genetic algorithm and a variable neighborhood search. Also, the case of Xiamen—a coastal city in south eastern China—was studied to evaluate the performance of the algorithm. Zhao and Alfandari [45] introduced a diversified package tour to balance the tradeoff between the tourist's choice and the economy of scale for tourism agencies. Then, branch-and-price and branch-cut-and-price methods were developed and the instances from both published literature and the case of Chinese online travel agencies (OTA) were solved to test the methods. Ruiz-Meza and Montoya-Torres [30] considered heterogenous preference of tourist group and the level of CO2 emission. Then the mathematical model was developed and results by using data sets were obtained. Addition, we refer the reader to the systematic review about TTDP [31].

It can be concluded that the research gap is to deal with a model that takes into consideration all of the constraints- budget, weather, and break time- at once. Thus, a comparative review of the literature is summarized in Table 1 that compares studies with some features: budget, weather, and break time constraints, mathematical model, algorithm and real-world application. It can be seen from table, this study contains all of these feature.

3 Mathematical model

The TTDP-BWL was formulated with the new constraints: budget, weather and break time. New parameters and new decision variables are defined to handle these constraints. The definition of the sets, parameters and decision variables were as follows:

Table 1 Literature review

Study	Problem	Budget in terms of money	Weather	Break Time	Mathematical Model	Algorithm	Real-world Application
Vansteenwegen and Van Oudheusden [37]	TTDP	–	–	–	+	+	–
Souffriau et al. [32]	TTDP	–	–	–	–	+	–
Souffriau et al. [33]	TTDP	–	–	–	+	+	+
Gavalas et al. [13]	TTDP	–	–	–	–	+	+
Brito et al. [5]	TTDP	–	–	–	–	+	+
Mei [26]	TTDP	–	–	–	–	+	+
Vansteenwegen et al., [42]	TTDP with lunch time	–	–	–	–	+	+
Abbaspour and Samadzadegan [1]	TTDP with several models of transportation system	–	–	–	+	+	+
Garcia et al. [12]	TTDP with transportation as walking and public transport	–	–	–	–	+	+
Gavalas et al. [15]	TTDP with time dependency between POIs	–	–	–	–	+	-
Zheng et al. [43]	TTDP with aesthetic fatigue and variable sight-seeing value	–	–	–	–	+	+
Kotiloglu et al. [21]	TTDP with mandatory POIs	–	–	–	–	+	-
Sylejmani et al. [35]	TTDP with tourist groups	–	–	–	+	+	–
Wörndl et al. [47]	TTDP with a walking route	–	–	–	–	+	–
Expósito et al. [11]	TTDP with clustered POIs	–	–	–	–	+	–
Ko et al. [20]	TTDP with tourist fatigue	–	–	–	–	+	–
Zheng et al. [44]	TTDP with hotel selection	–	–	–	–	+	+
Zhao and Alfandari [45]	TTDP with trade-off between tourist selection and the economy of scale for tourism agencies	–	–	–	–	+	+
Ruiz-Meza and Montoya-Torres [30]	TTDP with heterogenous preference of tourist group and the level of CO2 emission	–	–	–	+	+	–
This study	TTDP with budget, weather and break time	+	+	+	+	+	+

Sets:

K	Set of days
I	Set of POIs
I_0	Set of POIs and the starting point ($I_0 = I \cup \{0\}$)

Parameters:

p_i	The score of the i th POI by the tourist,$\forall i \in I$
t_{ij}	Travel time between the i th and j th POIs,$\forall i,j \in I_0$
v_i	Visit time for the i th POI,$\forall i \in I$
T_{max}	Maximum route duration,$\forall k \in K$
$[e_i, l_i]$	Time windows for the i th POI,$\forall i \in I$
$\left[e_i, Oe_i\right]$ $\cup$ $[Ol_i, l_i]$	Time windows of a POI if it is outdoors and the summer season, $\exists i \in I$ where Oe_i is the latest time to visit in the morning e.g. 11 a.m. and Ol_i is the earliest time to visit afternoon e.g. 3 p.m.
T_{start}	The starting time of the routes
B	The budget for the routes
c_i	The entrance fee of the i th POI,$\forall i \in I$
tu_{ij}	$\begin{cases} 1, \text{if a taxi is used between the ith and the jth POIs} \\ 0, \text{ otherwise} \end{cases} ,\forall i,j \in I_0$
km_{ij}	The distance in km between the i th and the j th POIs
tf	Taxi fare per kilometer
to	Taxi standing charge
b_i	$\begin{cases} 1, \textit{if the ith POI is suitable for a break} \\ 0, \quad \textit{otherwise} \end{cases} , \forall i \in I$
BT	The break times for the routes
BT_{tol}	The break time tolerances for the routes
DB	Duration of the break time
c_b	Cost of break
O_i	$\begin{cases} 1, \text{if the ith POI is outdoor} \\ 0, \text{otherwise} \end{cases} , \forall i \in I$
S	$\begin{cases} 1, \text{if it is the summer season} \\ 0, \text{otherwise} \end{cases}$
M	A big number

Decision variables:

x_{ijk}	$\begin{cases} 1, \text{if} \frac{\text{route}}{\text{day}} k \text{ goes from } i \text{ to } j \\ 0, \text{otherwise} \end{cases} , \forall i,j \in I_0, i \neq j, k \in K$
y_{ik}	$\begin{cases} 1, \text{if the ith POI is included in route/day } k \\ 0, \text{otherwise} \end{cases} ,$ $\forall i \in I, k \in K$

T_i	The start time of the visit to the ith POI, $\forall i \in I_0$
w_i	$\begin{cases} 1, \text{if the ith POI is outdoors and it is summer season} \\ \text{and the ith POI should be visited after } \mathrm{Ol}_i \\ 0, \text{otherwise} \end{cases}$ $,\forall i \in$ I.
BP_i	$\begin{cases} 1, \text{if the ith POI is the breakpoint} \\ 0, \text{otherwise} \end{cases}$ $, \forall i \in I$
v_k	$\begin{cases} 1, \text{if on the kth day there can be a break} \\ 0, \text{otherwise} \end{cases}$ $, \forall k \in K$

The mathematical model was formulated as follows:

$$\max z = \sum_{k\in K} \sum_{i\in I} p_i y_{ik} \tag{1}$$

subject to

$$\sum_{j\in I_0} x_{0jk} = \sum_{j\in I_0} x_{j0k} \leq 1 \quad \forall k \in K \tag{2}$$

$$\sum_{\substack{j \in I_0 \\ j \neq i}} x_{ijk} = \sum_{\substack{j \in I_0 \\ j \neq i}} x_{jik} \quad \forall i \in I, k \in K \tag{3}$$

$$\sum_{\substack{j \in I_0 \\ j \neq i}} x_{ijk} = y_{ik} \quad \forall i \in I, k \in K \tag{4}$$

$$\sum_{k\in K} y_{ik} \leq 1 \quad \forall i \in I \tag{5}$$

$$\sum_{i\in I_0} \sum_{k\in K} c_i y_{ik} + \sum_{i\in I_0} \sum_{\substack{j \in I_0 \\ j \neq i}} \sum_{k\in K} \left(to + tf.km_{ij}\right).tu_{ij}.x_{ijk} + \sum_{i} c_b.BP_i \leq B \tag{6}$$

$$T_0 = T_{start} \tag{7}$$

$$T_j \geq T_i + v_i + BP_i * DB + t_{ij} - M.\left(1 - \sum_{k\in K} x_{ijk}\right) \quad \forall i \in I_0, j \in I \tag{8}$$

$$T_i + v_i + DB.BP_i + \sum\nolimits_{k\in K} t_{i0}.x_{i0k} \leq \left(T_{max} + T_{start}\right) + M.\left(1 - \sum\nolimits_{k\in K} y_{ik}\right) \quad \forall i \in I \tag{9}$$

$$w_i.O_i.S.Ol_i.\sum_{k\in K} y_{ik} + e_i.\left(1 - O_i.S\right).\sum_{k\in K} y_{ik} \le T_i \quad \forall i \in I \tag{10}$$

$$T_i \le (1 - w_i).O_i.S.Oe_i.\sum_{k\in K} y_{ik} + M.w_i.S.O_i + (l_i - v_i - DB.BP_i).(1 - O_i.S).\sum_{k\in K} y_{ik} \quad \forall i \in I \tag{11}$$

$$v_k.\left(BT - B_{tol}\right).\sum_{j\in I} x_{0jk} \le \sum_{i\in I} b_i.BP_i.T_i.y_{ik} \le v_k.\left(BT + B_{tol}\right).\sum_{j\in I} x_{0jk} \quad \forall k \in K \tag{12}$$

$$\sum_{j\in I} T_j x_{j0k} \le M * v_k + \left(BT + B_{tol}\right) \quad \forall k \in K \tag{13}$$

$$\left(BT + B_{tol}\right) \le M * \mathrm{v_k} + \sum_{j\in I} T_j x_{j0k} \quad \forall k \in K \tag{14}$$

$$x_{ijk} \in \{0, 1\} \quad \forall i,j \in I_0, k \in K \tag{15}$$

$$y_{ij} \in \{0, 1\} \quad \forall i,j \in I_0 \tag{16}$$

$$BP_i \in \{0, 1\} \quad \forall i \in I \tag{17}$$

$$w_i \in \{0, 1\} \quad \forall i \in I \tag{18}$$

$$T_i \ge 0 \quad \forall i \in I_0 \tag{19}$$

Firstly, the objective function (1) of the problem is to maximize the total score which is awarded to POIs by the tourists. Constraint (2) ensures that each route starts and ends at the starting point if the k th route is used. The right hand side of the constraints is smaller than 1 since some days may not be used to visit POIs due to other constraints, e.g. budget may not be enough. Constraint (3) states that if a route arrives at a POI, then it also departs from there. Constraint (4) ensures that if a POI is assigned to a route then the POI is visited by that route. Constraint (5) ensures that each POI is assigned one route at most.

Next, the budget is controlled by constraint (6). The total expenses include the entrance fee and the taxi fee if a taxi is necessary between two consecutive POIs. Constraint (7) states that the starting time of the route is equal to the preferred starting time.

Then, constraint (8) requires that the starting time of a POI on a route should be later than the previous POI on the same route. The visiting time and the travelling time between these two consecutive POIs are also considered. Furthermore, if the previous POI is a breakpoint, then the duration of the break time is taken into consideration. Constraint (9) ensures that the maximum time for each route takes the time to visit a POI into consideration. If returning to the starting point, then the travel time between the relevant POI and the starting point is considered. In addition, the duration of any break time is also taken into consideration.

After that, constraints (10) and (11) state that the start time of the visit to a POI matches the opening time windows of the POI, and these constraint are significant to prevent the starting time before the opening time and thus the waiting time does not occur. If the POI is outdoors and it is the summer season, then the start time of the visit can be before Oe_i or after Ol_i. In this situation, the decision parameter w_i controls the "or" statement between constraints (10) and (11). If w_i is equal to 1, then it means that the start time of the visit is after Ol_i, otherwise it is earlier than Oe_i. If the POI is not outdoors and it is not summer, whether the value of w_i is 1 or 0, the start time of the visit is between the original time windows, since the product of parameter $O_i = 0$ and $S = 0$. Note that w_i should be a decision variable instead of a parameter since w_i decides the start time of the visit before Oe_i or after Ol_i.

Finally, the start time of the breakpoint is chosen for each route with a break time within the tolerances of constraints (12), (13) and (14). The breakpoint for each route is selected from POIs that are assigned to the route and are suitable for a break. The sum $\sum_{j\in I} x_{0jk}$ is equal to 1 if the k th route is used, otherwise the value is 0. Thus, multiplying both sides of the inequality by $\sum_{j\in I} x_{0jk}$ ensures that if the k th route is not used, a breakpoint is not chosen. Additional, v_k takes the value 1 together with constraint (13) and (14) if a break is suitable for the k th route before the break time has ended, otherwise it takes the value 0. In this case, constraint (12) is forced to 0 for both sides and the route is ended without a break right after the break time, plus its tolerances, has ended. Furthermore, constraints (15) to (19) specify the variable domains. Additional, the number of decision variables is $O(I_0^2K + IK + I_0 + 2I + K)$ while the number of constraints is $O\left(I_0I + 2IK + 4I + 2K + 3\right)$.

Constraints (2), (4) and (8) to (11) are modified from published literature (se e.g. [37]. In constraint (2), all days are not allowed to be constructed because of some constraints e.g. all the budget is consumed before assigning POIs to all routes/days, since the tourist may not have enough budget for all routes/days she/he selects. Constraint (4) is formulated with an equality because if y_{ik} is equal to 1, then the departure from i must occur. Significant modifications are established for constraints (8) to (11) to add the breaktime and the season effect into the time windows.

Constraints (6), (7), (12), (13) and (14) are newly formed in this study. First, while tourists travel, they may have limited money. The generated routes should fit the budget of tourists under constraint (6). Then, the costs of the route are the entrance fees and the distance between two points where a taxi has to be used. In general, T_0 is assumed to be 0. However, there is a start time defined by tourists. Thus, constraint (7) is written. Constraint (12) is formed by selecting the breakpoint for each route used. Lastly, constraint (13) and (14) are for the situation where a break may not be possible.

4 The Heuristic algorithm for TTDP-BWB

A heuristic algorithm was proposed to solve TTDP-BWB. The algorithm can be classified as greedy algorithm since it uses a function to select the next POIs under the problem constraints and selects the POI that has the maximum function value if the POI is suitable for selecting under the problem constraints. The function covers three parts: budget, time and the POI score as follows:

$$f_{ij} = \left(\frac{1}{\left(\left(to + tf.km_{ij} \right).tu_{ij} + c_j \right).f_{B_i}} \right) . \left(\frac{1}{(\mathrm{t_{ij}} + \mathrm{v_j}).f_{T_i}} \right) .p_j \quad \forall i \in I_0, j \in I \tag{20}$$

where$f_{B_i} = \max_{j \in I} \frac{1}{(to+tf.km_{ij}).tu_{ij}+c_j}, f_{T_i} = \max_{j \in I} \frac{1}{t_{ij}+v_j} \quad \forall i \in I_0.$

The first part of the function considered the proportion of the expense budget used going from the i th POI to the j th POI. The $\left(to + tf.km_{ij} \right).tu_{ij}$ and c_j show the cost of taxi usage, and the entrance fee, respectively. The term f_{B_i} gives the maximum value of this proportion. Thus, a normalized value between 0 and 1 was obtained. The value 1 means that going from the i th POI to the j th POI is the most suitable choice, according to the budget. The second part is about the time as a proportion of the total time spent going from the i th POI to the j th POI and the visit time for the j th POI. $\mathrm{t_{ij}}$ and $\mathrm{v_j}$ represent the travel time and the visiting time. Similarly, f_{T_i} is the maximum value and this part is also normalized. The last part of the function is the score for the j th POI, which is a significant parameter of the TTDP-BWB. This score is between 0 and 5. The score may decrease with the value of the first and second part of the function. In this way, the expense and time needed to go from the i th POI to the j th POI is taken into consideration as well as the scores given by tourists.

The proposed heuristic algorithm is explained using the algorithm pseudocode. The same notations are used as the proposed mathematical model. The algorithm starts at the starting point for each route k, then chooses the next POI with the function (20) under the constraint of time windows, budget, break time and weather. This procedure continues until the budget runs out or the time is up, or there are no suitable POIs left.

Algorithm: Pseudocode of the heuristic algorithm for TTDP-BWB (Heuristic-TTDP-BWB)

```
1:  Initialization:  Input: Get parameters: e.g. $T_{max}$, $T_{start}$, $B$. \\ See parameters of the proposed mathematical model.
                     Set $route(k) \leftarrow 0 \ \forall k \in K$. \\ $route(k)$ list represents the decision variable $x_{ijk}$ of the proposed mathematical model, is a list and starting point is added all routes/days
                     Set $List(k) \leftarrow i \ \forall i \in I, k \in K$. \\ $List(k)$ holds suitable POIs for each route/day and is a list. All POIs is set as suitable for all routes/days.
                     Set $UnassignList(k) \leftarrow \emptyset$. \\ $UnassignList(k)$ holds the removed POIs in lines 7 and 8.
                     Set $Break_{point}(k) \leftarrow \emptyset$ \\ $Break_{point}(k)$ holds POI also visited for break for all routes/days.
                     Set $Break_{time}(k) \leftarrow \emptyset \ \forall k \in K$ \\ $Break_{time}(k)$ holds the  break time for all routes/days.
                     Compute $f_{ij}$ (20) for all $i,j$ POIs.
                     Set $B_{rest} \leftarrow B$. \\ Set the residual of budget as the tourist's budget.
2:  for k:1 to $|K|$ do
3:      $T_{rest} = T_{max}$ , $i = 0$, $T_i = T_{start}$ . \\ Set the residual of time as the tourist's time and the starting time of the starting point.
        $Current_{Time} = T_{start}$ \\ Define the current time as the starting time. It is used to check the break point time.
4:      while ($B_{rest} > 0$ AND $T_{rest} > 0$ AND $List(k) \neq \emptyset$, ) do
          Set $adding - POI \leftarrow 0$ .\\ Set the parameter 0 to keep track whether a new POI is added the route/day k.
          Set $adding - break - POI \leftarrow 0$ .\\ Set the parameter 0 to keep track whether a POI is defined as break pint for the route/day k.
5:        while ($List(k) \neq \emptyset$ OR $adding - POI = 0$) do \\ Procedure goes on until there is not any available POIs to assign or one POI is assigned to route/day k.
6:          Choose $j$ which gives $\boldsymbol{max}_{j \in List(k)} f_{ij}$  where $f_{ij} \neq 0$.
7:          If $BT - BT_{tol} \leq Current_{Time} < BT + BT_{tol}$ and $Break_{point}(k) = \emptyset$,
              while ($Break_{point}(k) = \emptyset$ OR $List(k) \neq \emptyset$)
                  If time windows, budget, break point, and weather are suitable for assigning $j$,
                      remove $j$ from the $List(k) \ \forall k \in K$,
                      add $j$ to $route(k)$ ,
                      add $j$ to $Break_{point}(k)$,
                      set $B_{rest} \leftarrow B_{rest} - B_{ij}$, \\$B_{ij}$ is the sum of the costs of travelling from $i$th  POI to $j$th POI, entrance fee of  $j$th POI and the cost coffee break.
                      set  $T_{rest} \leftarrow T_{rest} - Time_{ij}$, \\ $Time_{ij}$ is the time needed for travelling from $i$th POI to $j$th POI and visiting time of $j$th POI.
                      set $Current_{Time} \leftarrow Current_{Time} + Time_{ij}$
                      set  $i \leftarrow j$,
                      calculate $T_i$,
                      set $Break_{time}(k) \leftarrow T_i$
                      set $adding - break - POI \leftarrow 1$,
                      set $adding - POI \leftarrow 1$.
                  Otherwise,
                      remove $j$ from the $List(k)$
                      add $j$ to the  $UnassignList(k)$
                  Choose $j$ which gives $\boldsymbol{max}_{j \in List(k)} f_{ij}$  where $f_{ij} \neq 0$.
              If $adding - break - POI = 0$,
                  add 0 to $route(k)$ \\ If break point is not available, then return the starting point.
                    set $List(k) = \emptyset$
          Else If time windows, budget, break time, and weather are suitable for assigning $j$,
              remove $j$ from the $List(k) \ \forall k \in K$,
              add $j$ to  $route(k)$ ,
              set $B_{rest} \leftarrow B_{rest} - B_{ij}$,
              set  $T_{rest} \leftarrow T_{rest} - Time_{ij}$,
              set $Current_{Time} \leftarrow Current_{Time} - Time_{ij}$
              calculate $T_i$,
              set  $i \leftarrow j$,
              set adding-POI ← 1.
8:          Otherwise,
              remove $j$ from the $List(k)$
              add $j$ to the  $UnassignList(k)$
9:        end while
10:       Set adding-POI ← 0.
          If $List(k) \neq \emptyset$,
                          add $j$, $\forall j \in UnassignList(k)$ to the $List(k)$,
                          set $UnassignList(k) \leftarrow \emptyset$.
11:     end while
        Add 0 to $route(k)$ \\ Return the starting point.
12: end for
13: Output: $route(k)$, $Break_{point}(k)$, $Break_{time}(k) \ \forall k \in K$, $T_i \ \forall i \in I$.
```

Line 1 is about initialization of the algorithm. The parameters (see the parameters of the mathematical model) are read, the list of routes/days *route*(k) and suitable POIs for each route/day *list*(k) is set. The starting point is assigned to each route. Furthermore, all POIs are added to the suitable list for each route and then the function (20) is calculated. Lastly, the remaining budget is equal to the tourist budget. The routes are constructed between the lines 2 and 12. In line 3, the remaining time for each route is defined as the tourist's maximum route duration at the beginning of assigning any POI to each route/day after the starting point. Additionally, the starting time for each route/day is defined as the tourist's start time. In line 6, the next POI is chosen according to the function (20). Then, in line 7, the constraints are checked to decide whether the chosen POI is suitable to assign as the next POI for that route/day. If all the constraints are satisfied, then the selected POI is removed from the list of all routes/days, since each POI can be visited one time at most, and is added to the route list of the relevant route/day. Additional, the break time and the break point are defined if the current time is suitable for break. If there is not any available POI for break, then the route is ended by returning the starting point. Otherwise, in line 8, the selected POI is removed from the suitable list of relevant routes/days. In line 10, if the suitable list of the relevant k th route is not empty, then all discarded j s from lines 5 to 9 are added to the list again, given that from the current point, it may not be suitable to go to these j s. However, from the next point, it may be feasible to travel to those POIs.

5 Computational experiments

There are two parts in this section. In the first part, instances were generated using the case of Eskisehir and in the second part, large-scale problems were generated using test problems from published literature. The experiments were run on a computer with an Intel® Core™ i5 9300HF CPU at 2.40 GHz with 8 GB of RAM, using simulation mobile devices which means the CPU times will be similar for a mobile phone.

5.1 The mobile application for the case of Eskisehir

The heuristic algorithm was coded using Android Studio and a mobile application was developed for the case of Eskisehir. In Fig. 1, the screenshots of the mobile application are shown. The picture on the left shows the main screen where the users write the budget, starting time, travel date, travel time (in hours) and travel days, and then choose the starting point for their trip. The picture on the right shows the scoring screen where the users award a score for each POI of Eskisehir from 0 to 5. A score 0 for a POI means that the user will not visit this POI, while a score 5 of a POI means that the user absolutely wants to visit this POI. The other scores show the desire to visit a POI between never and absolutely.

5 points was defined as possible starting points for Eskisehir where tourists choose to stay nearby. The users may choose the closest points among the starting points available. Since Eskisehir is a small town and these starting points are chosen to cover the whole city, these starting points are enough. A tourist will most probably be staying at a place near one of those starting points. 43 POIs is defined including parks, museums, shopping centers and famous streets. All the parameters needed, such as opening and closing times, entrance fees etc., are listed.[1] Additionally, the break time BT is defined in the middle of the route with a tolerance of BT_{tol} ±30 minutes, and the duration of the break DB is given as 30 minutes. The locations of the starting points and POIs are given in Fig. 2.

5.2 The results for Eskisehir

A problem containing 28 POIs was generated for Eskisehir.[2] New problems were generated with different parameters: the season S, the number of days K, the budget B and the maximum route duration T_{max}. The season S is shown as 1 or 0, which means that the season is summer or not summer, respectively. In total, 18 problems were produced. The locations are shown in Fig. 3.

All the problems were solved by both mathematical model, and the proposed heuristic algorithm (Heuristic-TTDP-BWB). The mathematical model is solved by using a GAMS (General Algebraic Modeling System) optimization software program using SCIP solver (see e.g. [22]. Additional, a time limit of 12 h is put and it should be noted that the Heuristic-TTDP-BWB finds solutions within seconds. The results for GAMS-SCIP z_{model} and the Heuristic-TTDP-BWB $z_{heuristic}$, the gap between the solutions, and the solving time needed for both GAMS-SCIP and the Heuristic -TTDP-BWB are shown in Table 2. The gap between solutions were calculated as follows:

$$GAP\% = \frac{z_{model-best-bound} - z_{model}}{z_{model-best-bound}}.100 \tag{21a}$$

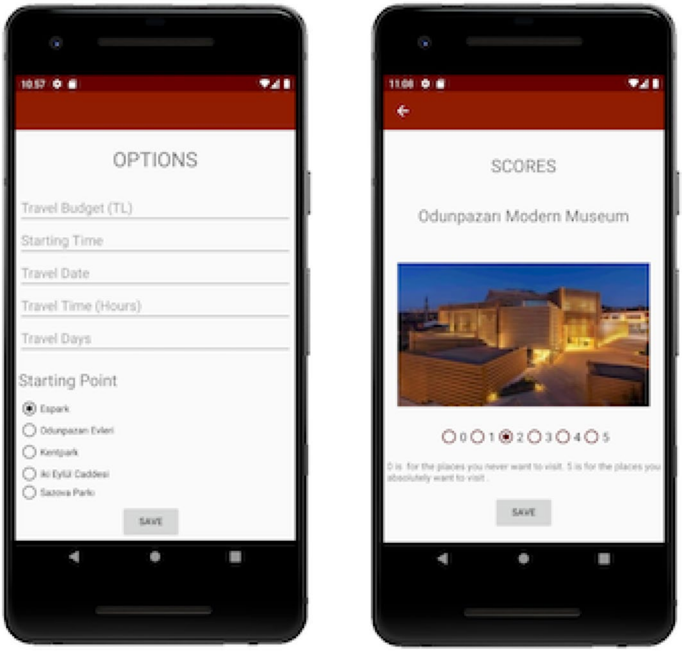
$$GAP\% = \frac{z_{model} - z_{heuristic}}{z_{model}}.100 \tag{21b}$$

$$GAP\% = \frac{z_{model-best-bound} - z_{heuristic}}{z_{model-best-bound}}.100 \tag{21c}$$

Compared to GAMS-SCIP, the Heuristic-TTDP-BWB found better solutions for 14 of the 18 problems, worse solutions for 2 of the 18 problems, and the same solutions for 2 of the 18 problems. The average gap for better solutions was – 15% while the average gap for worse solutions was 14%. Additionally,

[1] Details of the all parameters are shown in https://docs.google.com/spreadsheets/d/1Hr08j6yttG9lYDaZ4u_1foFe0zYM6BUj/edit?usp=sharing&ouid=117375956105700927487&rtpof=true&sd=true. Note that new problems for the case of Eskisehir may be generated using this form.

[2] Details of the problem are shown in. https://drive.google.com/file/d/1PhQPGgbVnEsZs8KsTDVb5eBjHQ0FkmVk/view?usp=sharing



OPTIONS

Travel Budget (TL)

Starting Time

Travel Date

Travel Time (Hours)

Travel Days

Starting Point

● Espark
○ Odunpazarı Evleri
○ Kentpark
○ İki Eylül Caddesi
○ Sazova Parkı

SCORES

Odunpazarı Modern Museum

○ 0 ○ 1 ● 2 ○ 3 ○ 4 ○ 5

0 is for the places you never want to visit. 5 is for the places you absolutely want to visit .

Fig. 1 The screenshots from the mobile application for the case of Eskisehir, Türkiye

the best gap was – 38%, while the worst gap was 23%. Furthermore, the overall average gap was – 10%. In addition, a comparison of GAMS-SCIP and the Heuristic-TTDP-BWB is shown in Fig. 4. These results demonstrated that the performance of the Heuristic -TTDP-BWB is remarkable with respect to the model results. Furthermore, the results are compared with best bounds (which may be hard to reach these values since the dual gap is possible), and GAMS-SCIP with 12 h' time limit had 28% of overall average gap while the Heuristic-TTDP-BWB had 22% of overall average gap. Addition, the Heuristic-TTDP-BWB reached the best bound in 4 problems. Since the mathematical formulation of the TTDP-BWB is integer, exact solvers such that GAMS-SCIP need so much time (the needed time may be days to find the exact solution) than heuristic algorithms which is not generate the exact solution however good solution with less time. So, it is the reason that there is a time limit for GAMS-SCIP and the Heuristic-TTDP-BWB is able to find better solutions than GAMS-SCIP. Moreover, in Fig. 5, the routes for the problem PE28_13_5 are shown on the map of Eskisehir and In Fig. 6, the screen view of the application is represented. A tourist would probably feel pleased with the route obtained by the Heuristic -TTDP-BWB.

5.2.1 Sensitivity *analysis*

In this section, sensitivity analysis of weather, number of days, budget and time (S, K, B and T_{max}) are made for both GAMS-SCIP and the Heuristic-TTDP-BWB,

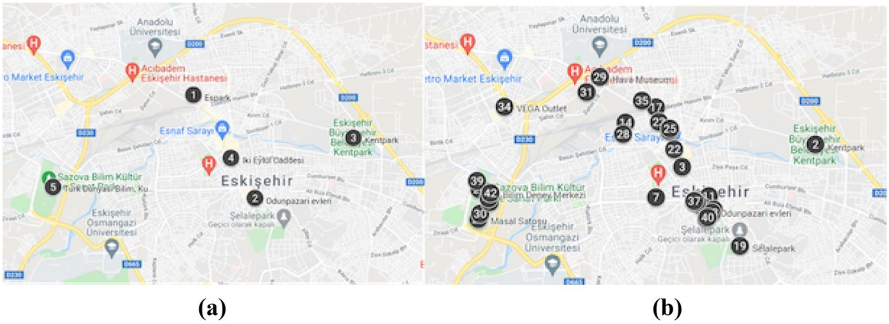
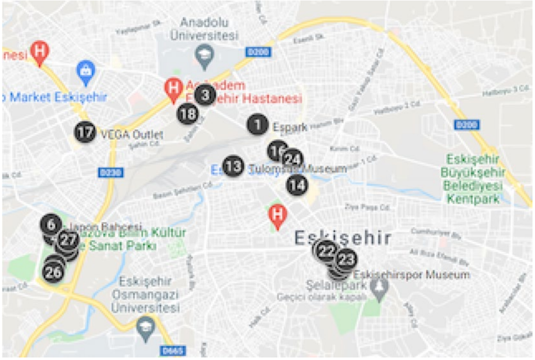


Fig. 2 The locations of the starting points (**a**) and all POIs (**b**) for the case of Eskisehir, Türkiye

Fig. 3 The locations of POIs for the problems generated


Anadolu Üniversitesi
Acıbadem Eskişehir Hastanesi
VEGA Outlet
Espark
Tulomsaş Museum
Japon Bahçesi
Sazova Bilim Kültür ve Sanat Parkı
Eskişehir
Eskişehirspor Museum
Şelalepark
Eskişehir Osmangazi Üniversitesi
Eskişehir Büyükşehir Belediyesi Kentpark
1
3
6
13
14
17
18
22
23
24
26
27

and the results are given in column graphics in Fig. 7. When non-summer case (S = 0) as summer case (S = 1) is analyzed, GAMS-SCIP finds solutions that have better value of objective function for non-summer case than summer case since the number of constraints increases for summer case because of visiting times of POIs. However, the heuristic-TTDP-BWB is able to generate similar solutions for both cases. Moreover, number of days and budget are investigated together in that when the number of days increases, the budget should be increase either to afford the expense of additional days. As expected, these parameters have positive effect on the value of objective function, and both GAMS-SCIP and the heuristic-TTDP-BWB get better solutions while the value of these parameters are getting greater. Finally, time has the similar impact as number of days and budget on the value of objective function.

5.3 Results of the large scale problems from published literature

The large problems[3] with 100 POIs were generated from a data set by Montemanni and Gambardella [27] and used to investigate the performance of the Heuristic

[3] All data are shown in https://drive.google.com/drive/folders/1fCUI2cz6nEgY1v7evBU32caZK4TiCkQG?usp=sharing

Table 2 Results for the medium problem for Eskisehir with 28 POIs

Problem Name	S	K	B	T_{max}	z_{model}	Best bound (BB) of z_{model}	GAP% (21a)	$z_{heuristic}$	Solving time of heuristic (sec.)	*GAP%*(21b)	*GAP%*(21c)
PE28_11_5	1	1	100 ₺	5 h	20	60	67%	**27**	0.008409	−35%	55%
PE28_01_5	0				24	61	61%	24	0.0136396	0%	60%
PE28_12_5	1	2	300 ₺		34	66	48%	**44**	0.0133022	−29%	33%
PE28_02_5	0				*40*	66	39%	**46**	0.0118584	−15%	30%
PE28_13_5	1	3	500 ₺		52	67	22%	**61**	0.0238167	−17%	10%
PE28_03_5	0				*50*	67	25%	**64**	0.0136989	−28%	4%
PE28_11_7.5	1	1	100 ₺	7.5 h	24	62	61%	**33**	0.0098189	−38%	47%
PE28_01_7.5	0				33	63	48%	31	0.0074511	6%	49%
PE28_12_7.5	1	2	300 ₺		55	66	17%	**57**	0.0123928	−4%	14%
PE28_02_7.5	0				57	66	14%	**59**	0.0150051	−4%	11%
PE28_13_7.5	1	3	500 ₺		63	69	9%	**69**	0.0293836	−10%	0%
PE28_03_7.5	0				66	69	4%	**69**	0.022952	−5%	0%
PE28_11_10	1	1	100 ₺	10 h	41	62	34%	**45**	0.012985	−10%	27%
PE28_01_10	0				48	62	23%	37	0.0071423	23%	40%
PE28_12_10	1	2	300 ₺		63	66	5%	63	0.0149367	0%	5%
PE28_02_10	0				61	66	8%	**63**	0.0141799	−3%	5%
PE28_13_10	1	3	500 ₺		66	71	7%	**71**	0.0196944	−8%	0%
PE28_03_10	0				66	71	7%	**70**	0.016418	−6%	0%
						Average gap%	28%		**Average gap%**	−10%	22%
						Best gap%	4%		**Best gap%**	−38%	0%
						Worst gap%	67%		**Worst gap%**	23%	60%

The results of the Heuristic-TTDP-BWB that are better than z_{model} are shown as bold

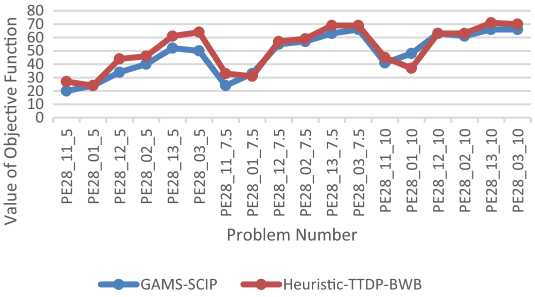
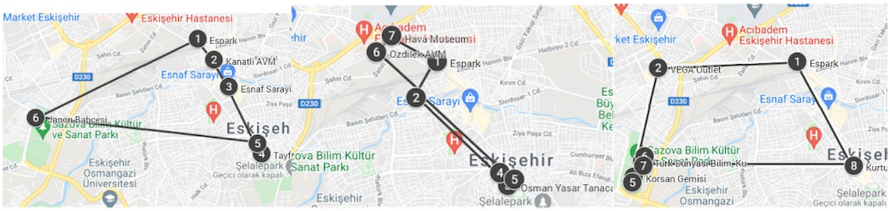


Fig. 4 A comparison of GAMS-SCIP z_{model} and the Heuristic -TTDP-BWB $z_{heuristic}$ for the case of Eskisehir, Türkiye

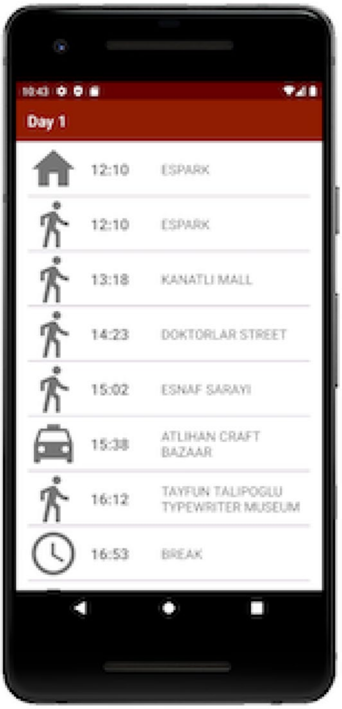


1st day 2nd day 3rd day

Fig. 5 The routes for the problem PE28_13_5 obtained by GAMS-SCIP on the map of Eskisehir

-TTDP-BWB on a large scale. The score of each POIs, location of POIs, the starting point, time windows of each POI are taken from the data set and the new parameters (entrance fee, the suitability of a POI for a break, whether a POI is an open area) needed for TTDP-BWB were obtained randomly. The distance between two POIs was represented by the Euclidian distance even though real distances instead of Euclidian distance is used in the case study, Eskisehir, Türkiye. Because, the locations are defined as x- and y-axis for the large problems. It should be noted that the real distances should be used in real world applications. Also, the unit of the distance is taken as km. Then, the taxi usage between two POIs was defined as 1 if the distance was greater than 5 km. The travel duration between two POIs was calculated by multiplying the distance by 2 min/km if the taxi was not used. Otherwise, the duration was equal to the distance. The other parameters used in the Heuristic -TTDP-BWB are shown in Table 3.

The results are shown in Table 4. All solutions were obtained within a second. These results demonstrated that the speed of the Heuristic-TTDP-BWB was good enough even when the size of the problem was large, e.g. 100 POIs. Additional, the results are given in Fig. 8 as non-summer case (S = 0) as summer case (S = 1). It may be concluded that the Heuristic-TTDP-BWB is able to produce the solutions where the values of the objective functions are close to each other when the season selection is summer or not. However, for non-summer cases, the value of objective



Day 1

12:10	ESPARK
12:10	ESPARK
13:18	KANATLI MALL
14:23	DOKTORLAR STREET
15:02	ESNAF SARAYI
15:38	ATLIHAN CRAFT BAZAAR
16:12	TAYFUN TALIPOGLU TYPEWRITER MUSEUM
16:53	BREAK

Fig. 6 The screen view of the mobile application for the routes

function may certainly be greater than summer cases since summer cases has one more constraint that is not to visit open area POI during some time period.

Moreover, since TTDP-BWB which is a new version of TTDP is handled in the paper and thus the results are original, to compare the results of large problems, a gap is also proposed as following:

$$\widetilde{Gap}\% = \frac{\widetilde{z}^{*} - z_{heuristic}}{\widetilde{z}^{*}} * 100 \tag{22}$$

where $\widetilde{z}^{*}$ is the sum of all POIs without considering any constraint and is the best possible value which is likely unfeasible under the problem constraints. The gaps are given in Table 5. While the gap is higher for the problems c101-rc108, the gap is relatively small for the problems c201-rc208. It is certainly impossible to reach the value of $\widetilde{z}^{*}$ because of the problem constraints. Additional, the time range of the problems c101-rc108 are too short while the visit times are too large and it is natural that the parameters of the problems are also effective on the quality of the solutions besides of algorithm. Thus, to make an interpretation on the results, it should be underlined the results that are smaller than 55%, and 14% (average gaps) for the

Fig. 7 Sensitivity analysis of parameters S for GAMS-SCIP (**a**) and the Heuristic-TTDP-BWB (**b**), K, B for GAMS-SCIP (**c**) and the Heuristic-TTDP-BWB (**d**), and T for GAMS-SCIP (**e**) and the Heuristic-TTDP-BWB (**f**)

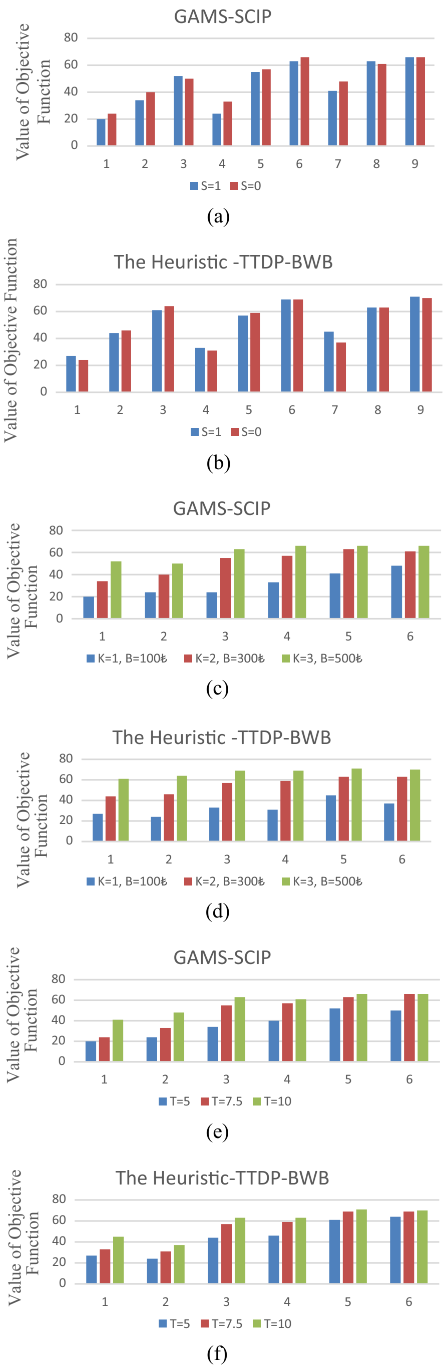


Table 3 The other parameters of the large scale problems

Problem name	K	B	T_{max}(minutes)	BT_{tol}(minutes)	DB(minutes)
c101-c109	5	5000₺	1236	90	30
r101-r112	5	5000₺	230	30	5
rc101-rc108	5	5000₺	240	30	5
c201-c208	3	8000₺	3390	90	30
r201-r211	3	8000₺	1000	60	30
rc201-rc208	3	8000₺	960	60	30

problems c101-rc108 and c201-rc208, respectively. According to these gaps, 28 out of 58 and 35 out of 54 results are lower average gaps for the problems c101-rc108 and c201-rc208, respectively.

6 Implications for practice and academics

TTDP dealt in in this paper includes novel constraints which are related to practical conditions: weather, budget and break time. From the viewpoint of tourists, it can be remarked that tourists can plan their trips by deciding only three issues: which POIs are desired to visit with a degree, their budget, and their time of the trip. These three issues are basic information that each tourist can inform. The rest of other parameters of the problem is obtained from POIs such as opening and closing time and from weather forecasting as sunny day or not etc. After that, the algorithm called the heuristic-TTDP-BWB in this paper is readily applied to obtain a solution under problem constraints with the aim of maximizing tourists' satisfaction (maximize the total score which is awarded to POIs by the tourists). Moreover, since the computation time to generate a solution by the Heuristic-TTDP-BWB is relatively less even if the number of POIs is large-scale, it can be concluded that the Heuristic-TTDP-BWB is an algorithm that is quite appropriate to implemented in practice. Addition, from the managerial implications, as the number of tourists who are grateful with the trip heightens, the number of tourists who are willing to come in the future is going to be ascended. Furthermore, the Heuristic-TTDP-BWB is easy to apply for different touristic regions, cities etc. and code as a mobile application which are functional for tourists, since every tourist has most probably a smart phone and internet connection within it and is used to take advantage of mobile applications.

Finally, from implications for academics, the mathematical model of TTDP is improved by adding the new constraints and as a consequence of new constraints, new parameters and decision variables are defined. Thus, the mathematical model of TTDP-BWB is a novel model in literature. In the meantime, a novel heuristic is proposed based on newly defined function that covers scores of POIs given by tourists, and also both budget and time. The other constraints are examined within the heuristic. Thus, the Heuristic-TTDP-BWB is capable of using budget and time efficiently

Table 4 Results of the large problems from published literature

Name	S	$z_{heuristic}$	CPU (sec.)	S	$z_{heuristic}$	CPU (sec.)	Name	S	$z_{heuristic}$	CPU (sec.)	S	$z_{heuristic}$	CPU (sec.)
c101	0	850	0.1020037	1	820	0.0685834	c201	0	1660	0.0750062	1	920	0.0581916
c102	0	970	0.0469071	1	980	0.0647326	c202	0	1400	0.1017837	1	1330	0.0903669
c103	0	1030	0.0795258	1	1070	0.069018	c203	0	1470	0.0664428	1	1380	0.0691699
c104	0	1090	0.1043911	1	1120	0.1028935	c204	0	1570	0.091733	1	1600	0.084917
c105	0	1010	0.070653	1	900	0.079322	c205	0	1650	0.0667085	1	1380	0.0771457
c106	0	780	0.0916054	1	710	0.065816	c206	0	1620	0.0552369	1	1430	0.0658846
c107	0	920	0.050164	1	890	0.0721762	c207	0	1560	0.0831319	1	1500	0.0883237
c108	0	1020	0.0626781	1	970	0.0639327	c208	0	1560	0.091099	1	1680	0.0804491
c109	0	1100	0.0869442	1	1110	0.0698421	r201	0	1046	0.070073	1	1011	0.0628797
r101	0	368	0.0575279	1	368	0.0430215	r202	0	1258	0.0814566	1	1207	0.0740131
r102	0	519	0.0445514	1	519	0.0499065	r203	0	1267	0.0788199	1	1276	0.0950372
r103	0	541	0.0659722	1	541	0.0536929	r204	0	1386	0.0807589	1	1385	0.0840095
r104	0	666	0.0546125	1	666	0.0967636	r205	0	1336	0.0854746	1	1318	0.0704613
r105	0	379	0.0552238	1	379	0.0445182	r206	0	1393	0.0827014	1	1308	0.0801113
r106	0	564	0.0503568	1	564	0.0552046	r207	0	1402	0.1036942	1	1393	0.0801778
r107	0	565	0.0515373	1	565	0.0509951	r208	0	1441	0.0969656	1	1418	0.0893643
r108	0	625	0.068201	1	625	0.0537164	r209	0	1301	0.0993529	1	1218	0.0750133
r109	0	532	0.0538932	1	532	0.0480105	r210	0	1347	0.0955851	1	1282	0.0826521
r110	0	650	0.0583264	1	650	0.052206	r211	0	1447	0.0936425	1	1422	0.0963969
r111	0	494	0.0490846	1	494	0.0449878	rc201	0	1192	0.062729	1	1140	0.0513765
r112	0	602	0.0502571	1	602	0.0513961	rc202	0	1288	0.0797065	1	1299	0.0685797
rc101	0	462	0.04293314	1	462	0.0508562	rc203	0	1498	0.0857505	1	1452	0.0791079
rc102	0	614	0.1087705	1	614	0.0542304	rc204	0	1695	0.0992266	1	1547	0.0795065
rc103	0	833	0.0625683	1	833	0.0641697	rc205	0	1335	0.0667718	1	1281	0.0663876

Table 4 (continued)

Name	S	$z_{heuristic}$	CPU (sec.)	S	$z_{heuristic}$	CPU (sec.)	Name	S	$z_{heuristic}$	CPU (sec.)	S	$z_{heuristic}$	CPU (sec.)
rc104	0	777	0.064333	1	777	0.0715937	rc206	0	1618	0.0898794	1	1600	0.0793494
rc105	0	538	0.0456593	1	538	0.0594384	rc207	0	1523	0.0671701	1	1464	0.0914641
rc106	0	541	0.0470683	1	541	0.042373	rc208	0	1664	0.0893569	1	1633	0.0826219
rc107	0	650	0.0599812	1	650	0.0694231							
rc108	0	696	0.0596498	1	696	0.0566193							

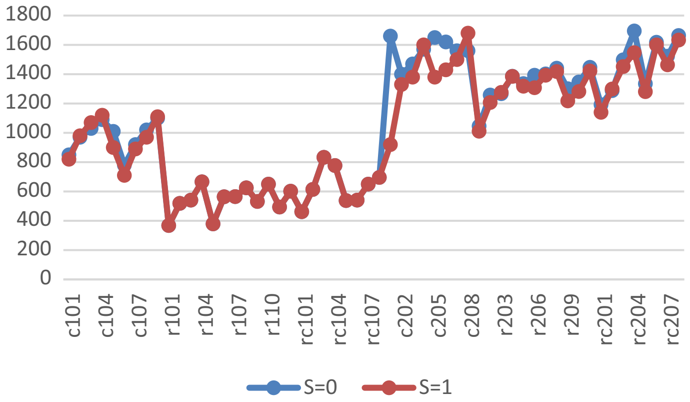


Fig. 8 The results of large problems where non summer case (S = 0) as summer case (S = 1). y-axis shows the value of objective function

to maximize the total scores of POIs visited under the other constraints of problem. Addition, small scale test problems which are based on real-world data from Eskisehir, Türkiye are generated and large-scale test problems are modified from published literature, thus if another algorithms are developed to solve the TTDP-BWB, then the performance of the related algorithms can be compared by using these test problems. In this way, new test problems for both small-scale and large-scale in the literature exist for the TTDP-BWB which is also a new variation problem of TTDP in literature.

7 Conclusions

In this study, TTDP was investigated with new constraints: budget, weather and break times. The budget is used to travel between two POIs if a taxi is needed, for the POI entrance fee – if not free, and for a coffee at break time. Consideration was given to what season it was. If the season was summertime, then the visiting time of an open area POI is limited e.g. before 11:00 a.m. and after 16:00 p.m. to offer protection from the effects of the sun. Break time was to take a rest during the route. The problem is abbreviated as TTDP-BWB.

Mathematical model of TTDP-BWB was constructed, and then a heuristic algorithm called Heuristic -TTDP-BWB was proposed. A case study was created for Eskisehir in Türkiye and all needed the data was formed. Then, a mobile application for Eskisehir was developed by coding the Heuristic -TTDP-BWB. problems containing 28 POIs for the case of Eskisehir was generated and solved by both mathematical model and the Heuristic-TTDP-BWB to see the performance of the Heuristic -TTDP-BWB. The results of these problems for the Heuristic -TTDP-BWB were promising when the results were compared with the solutions of the mathematical model. Additionally, the time needed to find a solution with the Heuristic

Table 5 Gaps between the results of large problems and the best possibleresult (may be infeasible) for comparison

Name	S	$\widetilde{Gap}$%(22)	S	$\widetilde{Gap}$% (22)	Name	S	$\widetilde{Gap}$%(22)	**S**	$\widetilde{Gap}$%(22)
c101	0	53%	1	55%	c201	0	8%	1	49%
c102	0	46%	1	46%	c202	0	23%	1	27%
c103	0	43%	1	41%	c203	0	19%	1	24%
c104	0	4%	1	38%	c204	0	13%	1	12%
c105	0	44%	1	5%	c205	0	9%	1	24%
c106	0	57%	1	61%	c206	0	1%	1	21%
c107	0	49%	1	51%	c207	0	14%	1	17%
c108	0	44%	1	46%	c208	0	14%	1	7%
c109	0	39%	1	39%	r201	0	28%	1	31%
r101	0	75%	1	75%	r202	0	14%	1	17%
r102	0	64%	1	64%	r203	0	13%	1	12%
r103	0	63%	1	63%	r204	0	5%	1	5%
r104	0	54%	1	54%	r205	0	8%	1	1%
r105	0	74%	1	74%	r206	0	4%	1	1%
r106	0	61%	1	61%	r207	0	4%	1	4%
r107	0	61%	1	61%	r208	0	1%	1	3%
r108	0	57%	1	57%	r209	0	11%	1	16%
r109	0	64%	1	64%	r210	0	8%	1	12%
r110	0	55%	1	55%	r211	0	1%	1	2%
r111	0	66%	1	66%	rc201	0	31%	1	34%
r112	0	59%	1	59%	rc202	0	25%	1	25%
rc101	0	68%	1	68%	rc203	0	13%	1	16%
rc102	0	58%	1	58%	rc204	0	2%	1	1%
rc103	0	43%	1	43%	rc205	0	23%	1	26%
rc104	0	47%	1	47%	rc206	0	6%	1	7%
rc105	0	63%	1	63%	rc207	0	12%	1	15%
rc106	0	63%	1	63%	rc208	0	3%	1	5%
rc107	0	55%	1	55%					
rc108	0	52%	1	52%					

Thus, to make an interpretation on the results, it should be underlined the results that are smaller than 55%, and 14% (average gaps) for the problems c101-rc108 and c201-rc208, respectively

-TTDP-BWB was a second, which is really practical for a tourist who wants to use the mobile application. Furthermore, the Heuristic -TTDP-BWB was able to quickly find solutions to even large scale problems with 100 POIs, which demonstrated its real-life applicability.

For future research, a mobile application could be developed with the Heuristic -TTDP-BWB for other tourist cities in Türkiye, other countries, and for the IOS operating system. A metaheuristic could be developed to see if it made a difference in terms of solution quality and solution time compared with the Heuristic -TTDP-BWB.

Declarations

Conflict of interest The authors have submitted this work to the OPSEARCH for review and publication with full consent. The authors declare that they have no competing interests and have not been supported by any funding source. Also, all datasets used in this paper are available. The sources of data are given as footnote in the sections.

References

1. Abbaspour, R., Samadzadegan, F.: Time-dependent personal tour planning and scheduling in metropolises. Expert Syst. Appl. **38**(10), 12439–12452 (2011)
2. Archetti, C., Herts, A., Speranza, M.G.: Metaheuristics for the team orienteering problem. J. Heuristics **13**(1), 49–76 (2007)
3. Archetti, C., Speranza, M.G., Corberán, Á., Sanchis, J.M., Plana, I.: The team orienteering arc routing problem. Transp. Sci. **48**(3), 442–457 (2013)
4. Archetti, C., Corberán, Á., Plana, I., Sanchis, J.M., Speranza, M.G.: A matheuristic for the team orienteering arc routing problem. Eur. J. Oper. Res. **245**(2), 392–401 (2015)
5. Brito, J., Expíósito-Márquez, A., Moreno, J.A.: A fuzzy GRASP algorithm for solving a tourist trip tesign problem. 2017 IEEE international conference on fuzzy systems (FUZZ IEEE), Naples, pp. 1–6. (2017)
6. Bouly, H., Dang, D., Moukrim, A.: A memetic algorithm for team orienteering problems. 4OR, 8, 49-70 (2010)
7. Boussier, S., Feiller, D., Gendreau, M.: An exact algorithm for team orienteering problems. 4OR, 5(3), 211–230 (2007)
8. Chao, I., Golden, B., Wasil, E.: The team orienteering problem. Eur. J. Oper. Res. **88**(3), 464–474 (1996)
9. Chao, I., Golden, B., Wasil, E.: A fast and effective heuristic for the orienteering problem. Eur. J. Oper. Res. **88**(3), 475–489 (1996)
10. Dang, D.-C., Guibadj, R.N., Moukrim, A.: An effective PSO-inspired algorithm for the team orienteering problem. Eur. J. Oper. Res. **229**, 332–344 (2013)
11. Expósito, A., Mancini, S., Brito, J., Moreno, J.: A fuzzy GRASP for the tourist trip design with clustered POIs. Expert Syst. Appl. **127**, 210–227 (2019)
12. Garcia, A., Vansteenwegen, P., Arbelaitz, O., Souffriau, W., Linaza, M.T.: Integrating public transportation in personalised electronic tourist guides. Comput. Oper. Res. **40**(3), 758–774 (2013)
13. Gavalas, D., Kenteris, M., Konstantopoulos, C., Pantziou, G.: Web application for recommending personalised mobile tourist routes. IET Softw. **6**(4), 313–322 (2012)
14. Gavalas, D., Konstantopoulos, C., Mastakas, K., Pantziou, G.: A survey on algorithmic approaches for solving trip design problems. J. Heuristics **20**, 291–328 (2014)
15. Gavalas, D., Konstantopoulos, C., Mastakas, K., Pantziou, G., Vathis, N.: Heuristics for the time dependent team orienteering problem: an application to tourist route planning. Comput. Oper. Res. **62**, 36–50 (2015)
16. Godart, J.-M.: Combinatorial optimization based decision support system for trip planning. In information and communication technologies in tourism 1999, 318–327, Springer (1999)
17. Gunawan, A., Lau, H.C., Vanteenwegen, P.: Orientering problem: a survey of recent variants, solution approaches and applications. Eur. J. Oper. Res. **255**(2), 315–332 (2016)
18. Hu, Q., Lim, A.: An iterative three-component heuristic for the team orienteering problem with time windows. Eur. J. Oper. Res. **232**(2), 276–286 (2014)
19. Keshtkaran, M., Ziarati, K., Bettinelli, A., Vigo, D.: Enhanced exact solution methods for the team orienteering problem. Int. J. Prod. Res. **54**(2), 591–601 (2016)

20. Ko, T., Qureshi, A.G., Schmöcker, J.-D., Fujı, S.: Tourist trip design problem considering fatigue. J. Eastern Asia Soc. Transp. Stud. **13**, 1233–1248 (2019)
21. Kotiloglu, S., Lappas, T., Pelechrinis, K., Repoussis, P.: Personalized multi-period tour recommendations. Tour. Manag. **62**, 76–88 (2017)
22. Kronqvist, J., Bernal, D.E., Lundell, A., Grossmann, I.E.: A revive and comparison of solvers for convex MINLP. Optim. Eng. **20**, 397–455 (2019)
23. Labadie, N., Mansini, R., Calvo, W.R., Melechovský, J.: The team orienteering problem with time windows: an lp-based granular variable neighborhood search. Eur. J. Oper. Res. **220**(1), 15–27 (2012)
24. Laporte, G., Martello, S.: The selective travelling salesman problem. Discret. Appl. Math. **26**, 193–207 (1990)
25. Lin, S.-W., Vincent, F.Y.: A simulated annealing heuristic for the team orienteering problem with time windows. Eur. J. Oper. Res. **217**(1), 94–107 (2012)
26. Mei, Y.: Study on the application and improvement of ant colony algorithm in terminal tour route planning under Android platform. J. Intell. Fuzzy Syst. **35**(3), 2761–2768 (2018)
27. Montemanni, R., Gambardella, L.: An ant colony system for team orienteering problem with time windows. Found. Comput. Decis. Sci. **34**(4), 287–306 (2009)
28. Muthuswamy, S., Lam, S.: Discreate particle swarm optimization for the team orienteering problem. Memet. Comput. **3**, 287–303 (2011)
29. Riera-Ladesma, J., Salazar-Gonzáles, J.J.: Solving the team orienteering arch routing problem with a column generation approach. Eur. J. Oper. Res. **262**(1), 14–27 (2017)
30. Ruiz-Meza, J., Montoya-Torres, J.R.: Tourist trip design with heterogeneous preferences, transport mode selection and environmental considerations. Ann. Oper. Res. **305**(1), 227–249 (2021)
31. Ruiz-Meza, J., Montoya-Torres, J.R.: A systematic literature review for the tourist trip design problem: extensions, solution techniques and future research lines. Oper. Res. Perspect. **9**, 100228 (2022)
32. Souffriau, W., Vansteenwegen, P., Vertommen, J., Vanden Berghe, G., Van Oudheusden, D.: A personalized tourist trip design algorithm for mobile tourist guides. Appl. Artif. Intell. **22**(10), 964–985 (2008)
33. Souffriau, W., Maervoet, J., Vansteenwegen, P., Vanden Berghe, G., Van Oudheusden, D.: A mobile tourist decision support system for small footprint devices. In Bio-Inspired Systems: Computational and Ambient Intelligence. In: Proceedings of 10th international work-conference on artificial neural networks (IWANN2009), Special session on advances in AI models adaptable to mobile devices, number 5517 in Lecture Notes in Computer Science (pp. 1248–1255), Salamanca, Spain. ISBN 978–3–642–02477–1 (2009)
34. Suarez, A.N., KoelenLacay, J., Villanueva, V., Ashley Velasquez, R., Reyes, C., Serrano, V., Borbon, C.D.: Impacts of coffee shop business to tourism industry in three cities of Batangas, Philippines. J. Tour. Hosp. Res. **14**(1), 131–145 (2017)
35. Sylejmani, K., Dorn, J., Musliu, N.: Planning the trip itinerary for tourist groups. Inform. Technol. Tour. **17**(3), 275–314 (2017)
36. Tang, H., Miller-Hooks, E.: A tabu search heuristic for the team orienteering problem. Comput. Oper. Res. **32**(6), 1379–1407 (2005)
37. Vansteenwegen, P., Van Oudheusden, D.: The mobile tourist guide: An OR opportunity. OR Insight **20**, 21–27 (2007)
38. Vansteenwegen, P., Souffriau, W., Vanden Berghe, G., Van Oudheusden, D.: A guided local search metaheuristic for the team orienteering problem. Eur. J. Oper. Res. **196**(1), 118–127 (2009)
39. Vansteenwegen, P., Souffriau, W., Vanden Berghe, G., Van Oudheusden, D.: Iterated local search for the team orienteering problem with time windows. Comput. Oper. Res. **36**(12), 3281–3290 (2009)
40. Vansteenwegen, P., Souffriau, W., Vanden Berghe, G., Van Oudheusden, D.: Metaheuristics for tourist trip planning. Chapter in lecture notes in economics and mathematical systems, May. (2009c)
41. Vansteenwegen, P., Souffriau, W., Van Oudheusden, D.: The orienteering problem: a survey. Eur. J. Oper. Res. **209**, 1–10 (2011)
42. Vansteenwegen, P., Souffriau, W., Vanden Berghe, G., Van Oudheusden, D.: The city trip planner: an expert system for tourist. Expert Syst. Appl. **38**, 6540–6546 (2011)

43. Zheng, W., Liao, Z., Qin, J.: Using a four-step heuristic algorithm to design personalized day tour route within a tourist attraction. Tour. Manag. **62**, 335–349 (2017)
44. Zheng, W., Ji, H., Lin, C., Wang, W., Yu, B.: Using a heuristic approach to design personalized urban tourism itineraries with hotel selection. Tour. Manag. **76**(1), 1–14 (2020)
45. Zhao, Y., Alfandari, L.: Design of diversified package tours for the digital travel industry: a branch-cut-and-price approach. Eur. J. Oper. Res. **285**(3), 825–843 (2020)
46. W. T. Organization, Ed., UNWTO Tourism Highlights. (2020)
47. Wörndl, W., Hefele, A., Herzog, D.: Recommending a sequence of interesting places for tourist trips. Inform. Technol. Tour. **17**(1), 31–54 (2017)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.