electronics

Article

A Critical Analysis of a Tourist Trip Design Problem with Time-Dependent Recommendation Factors and Waiting Times

Cynthia Porras [1,†], Boris Pérez-Cañedo [2,†], David A. Pelta [3,*] and José L. Verdegay [3]

1 Department of Software Engineering, Faculty of Informatics, Universidad Tecnológica de La Habana José Antonio Echeverría, La Habana 19390, Cuba; cporrasn@ceis.cujae.edu.cu
2 Department of Mathematics, Faculty of Economics and Business Sciences, University of Cienfuegos, Cienfuegos 55100, Cuba; bpcanedo@gmail.com
3 Department of Computer Science and AI, Universidad de Granada, 18071 Granada, Spain; verdegay@ugr.es
* Correspondence: dpelta@ugr.es
† These authors contributed equally to this work.

Abstract: The tourist trip design problem (TTDP) is a well-known extension of the orienteering problem, where the objective is to obtain an itinerary of points of interest for a tourist that maximizes his/her level of interest. In several situations, the interest of a point depends on when the point is visited, and the tourist may delay the arrival to a point in order to get a higher interest. In this paper, we present and discuss two variants of the TTDP with time-dependent recommendation factors (TTDP-TDRF), which may or may not take into account waiting times in order to have a better recommendation value. Using a mixed-integer linear programming solver, we provide solutions to 27 real-world instances. Although reasonable at first sight, we observed that including waiting times is not justified: in both cases (allowing or not waiting times) the quality of the solutions is almost the same, and the use of waiting times led to a model with higher solving times. This fact highlights the need to properly evaluate the benefits of making the problem model more complex than is actually needed.

Keywords: time-dependent recommendation factor; tourist trip design problem; waiting time

check for updates

Citation: Porras, C.; Pérez-Cañedo, B.; Pelta, D.A.;Verdegay, J.L. A Critical Analysis of a Tourist Trip Design Problem with Time-Dependent Recommendation Factors and Waiting Times. *Electronics* **2022**, *11*, 357. https://doi.org/10.3390/electronics11030357

Academic Editors: Juan M. Corchado, Stefanos Kollias and Javid Taheri

Received: 21 December 2021
Accepted: 22 January 2022
Published: 25 January 2022

1. Introduction

Tourism has been a key driver for the socio-economic progress of many countries. In 2020, due to the SARS-COV-2 pandemic, the sector experienced a USD 4.5 trillion drop in GDP and losses of 62 million jobs around the globe. However, in the November 2021 Report from the World Travel & Tourism Council one can read that "projections not only show promising growth opportunities in the domestic market for 2021; but a rise in international travel which will further accelerate in 2022 and beyond. Following a 49.1% decline in 2020 and a loss of almost US$4.5 trillion, Travel & Tourism GDP is projected to rise by 30.7% in 2021 and 31.7% in 2022." (https://research.wttc.org/trending-in-travel, accessed on 21 December 2021).

Within the tourism industry, an enormous set of artificial intelligence-based tools is available to solve a huge amount of underlying decision and optimization problems. For example, demand and price forecasting, recommendation systems, travel planning, personalization, sentiment analysis, and so on [1–3].

Here the focus is on one essential situation that should be addressed: a tourist arrives in a potentially unknown city, with several tourist places of interest to visit and some selection of them (and visiting order) needs to be suggested. Planning tourist itineraries is a challenging and time-consuming task due to the need to identify relevant places-of-interest (POIs) for tourists and plan visits to a subset of the POIs maximizing the overall interest of the route while having some time, budget, mobility, etc. constraints [4].

In this context, the development of an automated tool to assist decision-making is of utmost importance. However, before the implementation and deployment of such a tool, there is a need to properly understand the underlying problem and to decide which features should be included in the computational model of the problem.

From an optimization point of view, the tour planning is called "tourist trip design problem" (TTDP): given a set of POIs, each one with a level of interest (score), and certain time available, a route must be found that maximizes the overall interest subject to several constraints [5]. This problem is modeled as the so-called orienteering problem (OP) [6] and several variants have been considered [5,7].

The tour planning problem can be considered from other points of view, like consumer perceptions, consumer choice, cognitive attitudes and acquisition decisions [4]. In our contribution we focus on the area of optimization of tourist itineraries.

It should be remarked that the determination of a POI's level of interest is far from trivial as it may depend on several factors like personal preferences, weather conditions, accessibility, ticket price, visiting time, etc. Much of the needed data can be retrieved from social networks. These networks constitute a source of valuable information about tourist preferences that could be later used to derive scores and recommendation factors of POIs, as well as to promote different tourist destinations and specific itineraries [8]. The use of time-dependent information adds an additional layer of complexity to the problem that was previously considered in several works. For example, time-dependent travel times were considered in [9]. Subsequently, several papers on the subject have been presented, proposing extensions to the time-dependent OP (TDOP). Papers such as [10–12] manage time-dependent travel times which are associated with transportation modes available between POIs.

Li et al. [13] solved an OP in a time-dependent network where the travel times change with respect to the entry time to the POI. Gunawan et al. [14] proposed a variant of TDOP where the travel times change for each period due to congestion levels or characteristics of the area. Gavalas et al. [15] proposed a variant of TDOP where the travel times are dependent on the waiting times for the public transportation. Verbeeck et al. [16] proposed an algorithm based on an ant colony system for the TDOP, where the travel time between POIs depends on the departure time at the start node. Many times, the resulting problems belong to the NP-Hard class, thus paving the way to the application of metaheuristics like iterated local search, genetic algorithms, and so on [17].

Most of the available works assume that a POI's interest is fixed for a given user (or group of users), but here we depart from the following idea: *the level of interest of a POI should be related to when the POI is visited*. For example, visiting a water fountain is more interesting when the fountain is working than when it is not.

To the best of our knowledge, there are several works dealing with some kind of time-dependent interest. Murat and Labadie [18] proposed a team OP (TOP) with a decreasing function of the profits over time. Gündling and Witzel [19] presented a TOP with time windows and time-dependent profits based on the visit duration (the longer the visit, the higher the profit). Additionally, Isoda et al. [20] proposed three algorithms to obtain a tour's score in real time and to recommend an itinerary, where the POIs' scores can be static and dynamic depending on the time period. On the other hand, Yu et al. [21] proposed a TTDP variant considering several routes, time windows and POIs' levels of interest as a function of a time-dependent recommendation factor. Furthermore, Yu et al.'s [21] model introduces waiting times to allow reaching a POI when its level of interest is higher. Unfortunately, the proposed model (as appears in the publication) cannot be used *as is* and the test instances are not available.

The delay in the arrival of POIs has also been addressed in other variants of TTDP [11,12,14–16,18,19,21]. However, the inclusion of this feature in the model may (or may not) have an impact that should be properly analyzed.

In this paper, we focus on two features: (1) the previously mentioned time-dependent interest of a POI and (2) the opportunity to delay the arrival to a POI in order to obtain a higher interest.

We argue here that besides adding new features to existing problems, the usefulness and impact of such features should be properly evaluated. So, departing from a particular case of the model presented by Yu et al. [21] (with a single route and no time windows), the main objectives of this contribution are:

1. To propose a TTDP model with time-dependent recommendation factors, taking (and not taking) into account waiting times.
2. To solve a set of test instances, in order to gain insights regarding the empirical behavior of the models and to provide reference values for future research.
3. To evaluate whether or not considering waiting times provides any benefit in the tour's overall interest.

The rest of the paper is organized as follows: Section 2 describes the proposed mathematical model with time-dependent recommendation factors and waiting times (WT). In order to evaluate if allowing waiting times provides a benefit, a model without waiting times (NWT) is also presented. Section 3 describes a set of problem instances, using data from Granada city, Spain. Section 4 describes the computational experiment's settings while the results are presented in Section 5. Finally, Section 6 is devoted to conclusions and future research works.

2. Models for Time-Dependent TTDP

In this section, we propose a model for the tourist trip design problem with time-dependent recommendation factors (TTDP-TDRF) and waiting times (WT). It is a particular case of the model proposed in [21], with one itinerary and no time windows. Additionally, some constraints have been revised.

Let us consider a graph $G(V, A)$ where V is a set that includes all POIs and also the start and end nodes of any possible itinerary, and A is a set of arcs among them. The travel time between each pair of elements of V is known. Every POI in V has a visiting time and a fixed score, representing its level of interest. The itinerary has a time budget T_{max}. In order to model time-dependent recommendation factors, T time periods are included. For each time period, every POI has a recommendation factor to weigh its score.

When a tourist arrives at a POI and the visiting time is consumed, there are two alternatives: first, leaving immediately to the next POI in the itinerary, and second, waiting a certain time, thus arriving at the next POI in a period of time with a higher recommendation factor.

As one of the objectives of this contribution is to evaluate whether or not considering waiting times provides any benefit, a model without waiting times (NWT) is also proposed.

We start by describing the shared parameters and variables of both models.

Parameters:

- $V = \{1, 2, \ldots, n\}$: set of nodes (including all considered POIs) where 1 and n are the start and end nodes of the route, respectively. All nodes in between are POIs that may be included in the route.
- T: number of time periods.
- S_i: score of POI i.
- f_{it}: recommendation factor of POI i in period t.
- T_{max}: maximum duration time for the itinerary.
- t_{ij}: travel time (or distance) from POI i to POI j.
- v_i: visiting time at POI i.
- b_t: starting time of period t.
- e_t: ending time of period t.
- M: a very large constant.

Variables:

- $a_i \in [0, T_{max}]$: arrival time at node i.

- $X_{ij} \in \{0,1\}$: 1 if there is a path from POI i to POI j in the route, 0 otherwise.
- $Y_{it} \in \{0,1\}$: 1 if the visit to the POI i starts in period t, 0 otherwise.

Figure 1 shows an example of two solutions (tours) A and B under both models: the latter is an itinerary that allows waiting times, while the former does not. Both tours have $V = \{1, 2, \ldots, n\}$ nodes, where $n = 7$. The nodes $i = 1$ and $i = 7$ are the start and end nodes, respectively. The nodes from $i = 2$ to $i = 6$ are the POIs of the itinerary. The table displays, for each POI i, its score (S_i), visiting time (v_i) and recommendation factor for every time period t (f_{it}). So, for example, if POI $i = 2$ is visited in period $t = 2$, then it will have a level of interest $S_2 \times f_{2,2} = 30 \times 0.5$, while if it is visited in period $t = 4$, the interest will be lower $S_2 \times f_{2,4} = 30 \times 0.25$.

After node $i = 1$, POIs $i = 2$ and $i = 3$ are visited. We observe that the recommendation factor of POI $i = 4$ in the second period is 0.5, while it is 0.75 in the third one ($f_{4,2} < f_{4,3}$). If waiting times are allowed, a tourist that follows itinerary B will delay the arrival to POI $i = 4$ to gain a higher interest (he/she can spend more time visiting POI $i = 3$ or take extra time to travel from POI $i = 3$ to POI $i = 4$). Then, the visit continues to POIs $i = 5$ and $i = 6$, finishing at node $i = 7$. The interest of itinerary A is 110, while that of B is 112.5. In turn, itinerary A will be completed before B, precisely because no waiting times are allowed.

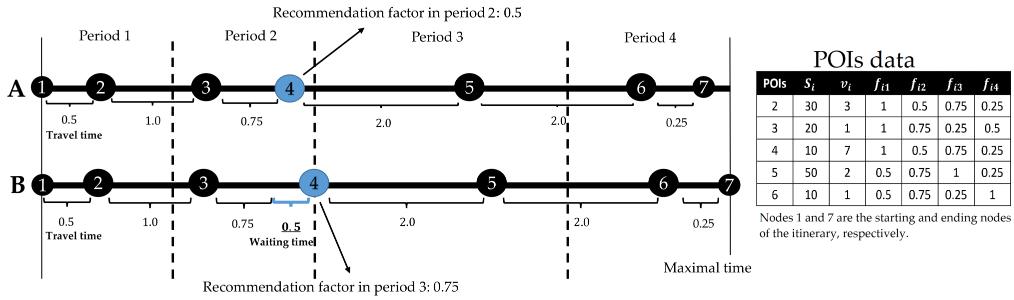

Figure 1. Illustrative example: itinerary (**A**) has no waiting time and itinerary (**B**) has waiting time.

In what follows, we present the specific features for each model.

2.1. TTDP-TDRF with Waiting Times

In addition to features explained before, in this model the tourist can wait a time in order to arrive at a POI in a period where its recommendation factor is higher. Consequently, we have

$$\max\ Z = \sum_{t \in T} \sum_{i \in V} S_i \times f_{it} \times Y_{it} \tag{1}$$

$$\text{s.t.} \sum_{t \in T} v_i \times Y_{it} + \sum_{i \in V} t_{ij} \times X_{ij} \leq T_{max} \quad \forall i \in V\ \forall j \in V\ \forall t \in T \tag{2}$$

$$a_{|V|} \leq T_{max} \tag{3}$$

$$\sum_{i \in V} X_{1i} = \sum_{i \in V} X_{i|V|} = 1 \tag{4}$$

$$\sum_{i \neq h \in V} X_{ih} = \sum_{j \neq h \in V} X_{hj} = \sum_{t \in T} Y_{ht} \quad \forall h \in \{2, \ldots, |V| - 1\} \tag{5}$$

$$\sum_{t \in T} Y_{it} \leq 1 \quad \forall i \in \{2, \ldots, |V| - 1\} \tag{6}$$

$$b_t \times Y_{it} \leq a_i \quad \forall i \in V \quad \forall t \in T \tag{7}$$

$$a_i \leq e_t \times (Y_{it} + ((1 - Y_{it}) \times M)) \quad \forall i,j \in V\ (i \neq j)\ \forall t \in T \tag{8}$$

$$a_i + t_{ij} + v_i \leq a_j + M \times (1 - X_{ij}) \quad \forall i,j \in V\ (i \neq j) \tag{9}$$

The objective function (1) is aimed at maximizing the total interest of the tour. Constraint (2) ensures that the total time of the tour does not exceed T_{max}. Constraint (3) guarantees that the arrival time $a_{|V|}$ at node $|V|$ (end node of the route) is less than or equal to the maximum travel time T_{max}. Constraint (4) establishes that the tour must begin at the start node and must finish at the end node, visiting them only once. Constraint (5) guarantees the tour connectivity. Constraint (6) ensures that POIs are visited only once. Constraints (7) and (8) establish that if POI i is visited in period t then its arrival time a_i has to be in the interval $[b_t, e_t]$.

Constraint (9) establishes that the arrival time at POI j (a_j) is greater than the arrival time at POI i (a_i), provided that POI j is visited after POI i, consuming the visiting time v_i and traveling t_{ij} time units, where $X_{ij} = 1$. This is the constraint that allows the use of waiting times, since $a_i + t_{ij} + v_i + w_i = a_j$, where w_i is the waiting time assigned to arrive at a POI in an instant of time.

2.2. TTDP-TDRF without Waiting Times

This second model also considers the objective function (1) and the constraints (2)–(8). In addition, to guarantee that tourists arrive at POI j at the exact time $a_j = a_i + t_{ij} + v_i$ (no waiting time), a new constraint (11) is added to the model.

So, a model for TTDP-TDRF without waiting times is:

$$
\begin{aligned}
\max\ Z = & \sum_{t \in T} \sum_{i \in V} S_i \times f_{it} \times Y_{it} \\
\text{s.t. } & \sum_{t \in T} v_i \times Y_{it} + \sum_{i \in V} t_{ij} \times X_{ij} \leq T_{max} \quad \forall i \in V\ \forall j \in V\ \forall t \in T \\
& a_{|V|} \leq T_{max} \\
& \sum_{i \in V} X_{1i} = \sum_{i \in V} X_{i|V|} = 1 \\
& \sum_{i \neq h \in V} X_{ih} = \sum_{j \neq h \in V} X_{hj} = \sum_{t \in T} Y_{ht} \quad \forall h \in \{2, \ldots, |V|-1\} \\
& \sum_{t \in T} Y_{it} \leq 1 \quad \forall i \in \{2, \ldots, |V|-1\} \\
& b_t \times Y_{it} \leq a_i \quad \forall i \in V \quad \forall t \in T \\
& a_i \leq e_t \times (Y_{it} + ((1 - Y_{it}) \times M)) \quad \forall i, j \in V\ (i \neq j)\ \forall t \in T \\
& a_i + t_{ij} + v_i \leq a_j + M \times (1 - X_{ij}) \quad \forall i, j \in V\ (i \neq j) \qquad (10) \\
& a_j \leq a_i + t_{ij} + v_i + M \times (1 - X_{ij}) \quad \forall i, j \in V\ (i \neq j) \qquad (11)
\end{aligned}
$$

Constraints (10) and (11) together guarantee that if POI j is visited right after POI i, i.e., when $X_{ij} = 1$, the arrival at POI j occurs exactly at $a_i + t_{ij} + v_i$; hence, no time is spent in waiting.

3. Description of the Test Instances (Samples)

This section explains the process followed to generate the tests instances. In the first place, a collection with 100 POIs from Granada city, Spain was constructed, using the Python package OSMnx (https://osmnx.readthedocs.io/en/stable/, accessed on 21 December 2021) and the function geometries _from_place('Granada, Spain') which returns a set that includes museums, bars, restaurants, religious sites, etc.

The following additional information of each POI is also gathered: identifier, name, service offered, node in the network, and latitude and longitude. Then, the travel times between every pair of POIs were calculated using the same package. Four time periods are considered ($T = 4$) and $T_{max} = 480$ min. Each time period has a duration of 120 min. The visit duration v_i of every POI i is a randomly generated integer in the interval $[20, 60]$. The score (interest) S_i is also a randomly generated integer in the interval $[1, 10]$.

In order to assign the recommendation factors f_{it} for every POI at each time period, we first define a set $RF = \{0.25, 0.50, 0.75, 1\}$. Then, every f_{it} is assigned a random value from the set RF. Each POI has $f_{it} = 1$ in some period t to ensure that there is an "ideal" period t to visit it.

Having this set of 100 POIs, we considered nine different sizes $|V| \in \{10, 20, 30, 40, 50, 60, 70, 80, 90\}$, and, for each size, three different test instances were constructed, randomly selecting the POIs. Two "dummy" nodes representing the start and end nodes of the tour were finally added.

4. Description of Computational Experiments

The computational experiments are oriented to the following goals:

1. To solve the 27 test instances under the two models considered (WT, NWT), thus obtaining the optimal solutions or reference values for future experimentation.
2. To evaluate the implications of including or not including waiting times in the model considering:
   - the quality of the solutions,
   - the effort needed to obtain them,
   - the similarity/differences between solutions.

In order to solve the problems (27 instances), CPLEX 20.1.0 [22] software was used, with a maximum computational time of one hour per instance. CPLEX has been successfully used to optimally solve and to compare different formulations of routing problems, such as TSP [23]. Variants of TTDP were solved using this solver as well [7,9,14,16,21].

The experiments were performed on a computer with an Intel(R) Core(TM) i7-5700HQ CPU 2.70GHz and 8GB RAM running Windows 10 Pro version 20H2. For each instance, we recorded the best value achieved by CPLEX and the time required to obtain it. This was done for the WT and NWT models. Thus, we ended up having a WT solution and an NWT solution for each instance. In order to analyze other features of the solutions, we also computed the number of POIs of the route, the total travel time as $\sum d_{ij} \times X_{ij}$, the total visiting time as $\sum v_i \times Y_{it}$ and the total time of the route (sum of total travel time and total visiting time).

We resorted to the Damerau–Levenshtein (edit) distance [24] (as implemented in the Python package available at https://pypi.org/project/fastDamerauLevenshtein/, accessed on 21 December 2021) for measuring the similarity between two solutions. Given two strings (or permutations as it is in our case) A and B, the Damerau–Levenshtein distance computes the minimum required number of edit operations needed to transform A into B. The edit operations are substitutions, insertions, deletions and transpositions of adjacent elements, and they are performed on either of the strings [24]. The similarity between two solutions is obtained upon normalization, i.e., we divide the number of operations by the length of the longest permutation and subtract the result from one. The procedure is clarified with the following example.

To make the calculations, solutions are interpreted as permutations of subsets of indexes corresponding to POIs. So, let us consider the following permutations (solutions to instance No. 9).

WT	**18**	<u>22</u>	**12**	**11**	**28**	20	9	**15**	<u>30</u>	**21**	4	3
NWT	12	18	<u>19</u>	28	11	20	9	21	15	4	3	

In order to find the minimum edit distance in the above example, we first delete the underlined numbers from both permutations; so 22, 19 and 30 are deleted. This makes three operations, and thus we get

WT	**18**	**12**	**11**	**28**	20	9	**15**	**21**	4	3
NWT	12	18	28	11	20	9	21	15	4	3

Next, the numbers in bold are swapped, i.e., swap 18 and 12, 11 and 28, and finally, 15 and 21; this also makes three operations. So we have performed six operations in total and ended up with the following permutations.

WT	**12**	**18**	**28**	**11**	20	9	**21**	**15**	4	3
NWT	12	18	28	11	20	9	21	15	4	3

We see that they are now equal. Consequently, the similarity value is $1 - 6/12 = 0.5$, where, for normalization, we have considered the number of operations performed (6) and the length of the longest (unmodified) permutation (12).

5. Analysis of Results

The results for every instance under both models are available in Appendix A, where Table A1 contains the results obtained for the WT model and Table A2 those obtained for the NWT model.

The first element to highlight is that CPLEX failed to reach the optimal values in the largest instances (those with 90 POIs) for both models. However, in the WT model, it failed to optimally solve those with 80 POIs. Figure 2 provides a comparison on the quality of the solutions obtained under each model, together with the effort (execution time) needed to reach them. The execution time was represented using a log-log plot.

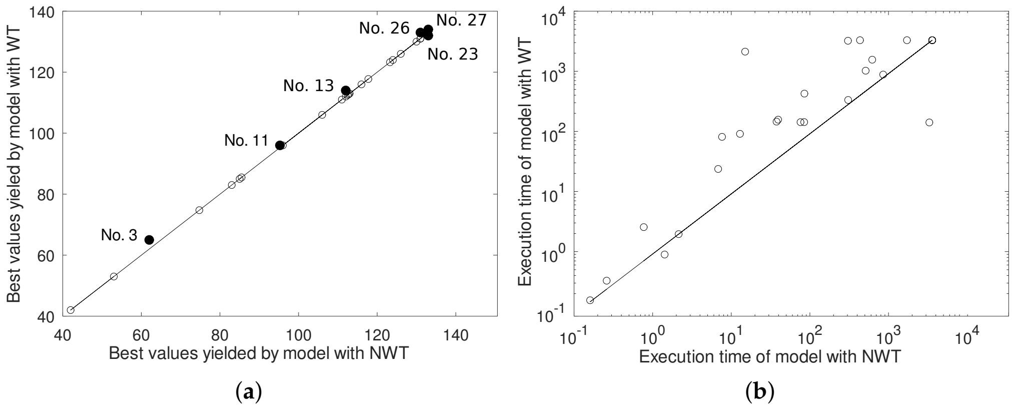

Figure 2. Comparison of best values and execution time for every instance under WT and NWT models. (**a**) Best values; (**b**) Execution time.

As Figure 2a shows, in general, no relevant differences can be observed in the values of the objective function for both models. Most of the points are in the straight line. Just five points are above the line (No. 3, 11, 13, 26 and 27), thus indicating that in those instances, the optimal values of the WT model are higher than those of the NWT model. In only one case the NWT model achieved a higher value (No. 23). However, in this case, CPLEX failed to reach the optimal value of the WT model within one hour. Additionally, in two cases (No. 26 and 27) where CPLEX did not obtain the optimal solution to any of the models, the WT model solution was better than the NWT model solution. Finally, just in 22% of the instances the best values of the models are different.

If we consider now the computational effort needed to obtain the optimal values, Figure 2b clearly shows that in 25 (out of 27) cases, it took CPLEX longer to solve the WT model. Just in two cases (No. 3 and 13) the NWT model was slower to solve. Finally, we applied the Wilcoxon non-parametric test between the execution times of the WT and NWT models with $\alpha = 0.05$. The results show a p-value ≤ 0.001, then there are significant differences between the execution times of both models. Moreover, CPLEX obtained solutions to the NWT model faster.

Regarding the characteristics of the solutions, we found that in three cases (No. 5, 14 and 15) the solutions to the WT and NWT models are exactly the same, and in two cases (No. 9 and 27) the routes have a different number of POIs. In 44% of the instances, the solutions to the WT model had a higher total travel time than those to the NWT model, but their average difference is merely about 2 min. The total visiting time was the same in 67% of the instances, and the average difference over all the instances is just 1.7 min.

Figure 3 displays the similarity values between solutions to the WT and NWT models for every test instance, calculated using the Damerau–Levenshtein (edit) distance. Recall that there are three instances per problem size. The plot shows that both models provide quite similar solutions: in 18 cases the similarity values were higher than 0.5 and in only one case, it was below 0.25. Another element to remark is that there does not seem to be a correlation between similarity and problem size, although in problems with 70 or more POIs the solutions never achieved similarity values higher than 0.75.

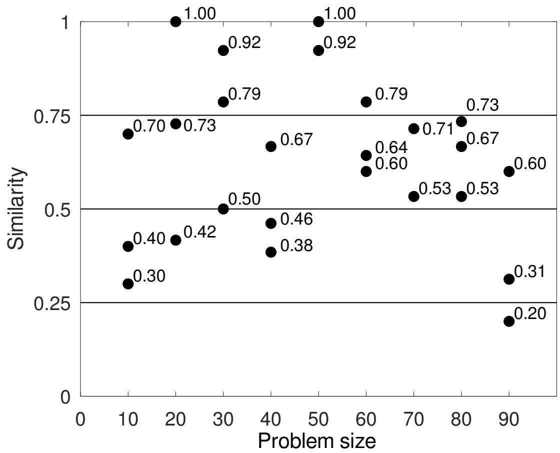

Figure 3. Similarity values between WT and NWT solutions.

6. Discussion and Conclusions

In this contribution, we presented an analysis of two variants of TTDP with time-dependent recommendation factors. The first one takes into account waiting times, while the second one does not. After constructing and solving 27 test instances, the following conclusions are drawn:

- Although solutions obtained with the WT model are theoretically better than those obtained with the NWT model, in 78% of the test instances the best scores from both models are the same.
- In almost all cases, the solver obtained the solutions to the NWT model faster than to the WT model.
- The similarity analysis revealed that the solutions to both models are quite similar.

In short, we claim that considering waiting times in the TTDP-TDRF is not justified, at least in the test instances selected. Although including waiting times is a quite specific feature, we also suggest that unneeded complexification should be avoided: the addition of new features to a model in order to make it more "realistic" is not always needed.

As we have seen, adding new features to a model in order to make it more realistic may complicate its solving process. However, in cases where those features are indeed justified, a feasible alternative would be to obtain a set of solutions to a simpler model and include such features a posteriori in the way proposed in [25].

Now, for most test instances the optimal overall interest is available in both models, thus paving the way to explore the use of metaheuristic algorithms to solve them. This is left as future work.

Author Contributions: Conceptualization, C.P., B.P.-C., D.A.P. and J.L.V.; methodology, C.P., B.P.-C., D.A.P. and J.L.V.; software, C.P., B.P.-C. and D.A.P.; validation, C.P., B.P.-C., D.A.P. and J.L.V.; formal analysis, C.P., B.P.-C. and D.A.P.; investigation, C.P., B.P.-C., D.A.P. and J.L.V.; resources, C.P., B.P.-C., D.A.P. and J.L.V.; data curation, C.P. and D.A.P.; writing—original draft preparation, C.P., B.P.-C., D.A.P. and J.L.V.; writing—review and editing, C.P., B.P.-C., D.A.P. and J.L.V.; visualization, C.P. and B.P.-C.; supervision, D.A.P. and J.L.V.; project administration, J.L.V. All authors have read and agreed to the published version of the manuscript.

Funding: Research supported by projects PID2020-112754GB-I0, MCIN/AEI/10.13039/501100011033 and FEDER/Junta de Andalucía, Consejería de Transformación Económica, Industria, Conocimiento y Universidades/ Proyecto (B-TIC-640-UGR20).

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: http://github.com/cporrasn/TTDP_TDRF_WT_NWT.git.

Acknowledgments: C.P. has been supported by a scholarship from AUIP Association coordinated with the Universidad de Granada. B.P.-C. was supported by the Erasmus+ programme of the European Union. The authors are grateful to the editors and the anonymous reviewers for their constructive comments and suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. Results of CPLEX for each instance of WT model. Optimal values are marked in **bold**.

No.	$\|V\|$	Best Value	Execution Time	No. POIs	Travel Time	Visiting Time	Total Time
1	10	**53**	0.36	10	41.02	371	412.02
2	10	**42**	0.17	10	38.53	410	448.53
3	10	**65**	0.98	10	32.15	357	389.15
4	20	**74.75**	2324.51	12	35.93	443	478.93
5	20	**83**	2.80	13	32.33	444	476.33
6	20	**85.5**	99.71	11	30.13	449	479.13
7	30	**85**	965.87	13	51.92	428	479.92
8	30	**106**	25.97	14	42.57	437	479.57
9	30	**96**	2.14	12	33.40	440	473.40
10	40	**111**	465.69	13	39.67	436	475.67
11	40	**96**	156.00	12	46.95	432	478.95
12	40	**112.75**	171.67	13	31.48	445	476.48
13	50	**114**	154.23	13	39.12	440	479.12
14	50	**112.5**	3524.52	13	40.67	439	479.67
15	50	117.75	3600	14	35.50	444	479.50
16	60	**113**	156.37	14	30.65	448	478.65
17	60	**112**	88.82	14	27.87	451	478.87
18	60	**116**	159.27	15	42.95	437	479.95
19	70	**124**	1706.51	14	41.85	438	479.85
20	70	**131**	364.93	15	43.05	434	477.10
21	70	**123.25**	1119.97	14	42.78	436	478.78
22	80	131	3600	15	42.68	436	478.68
23	80	132	3600	15	39.85	440	479.85
24	80	130	3600	15	45.72	434	479.72
25	90	126	3600	15	47.77	432	479.77
26	90	133	3600	15	44.63	426	470.63
27	90	134	3600	15	54.07	423	477.07

Table A2. Results of CPLEX for each instance of NWT model. Optimal values are marked in **bold**.

No.	$\lvert V \rvert$	Best Value	Execution Time	No. POIs	Travel Time	Visiting Time	Total Time
1	10	**53**	0.26	10	47.56	371	418.57
2	10	**42**	0.16	10	40.87	410	450.87
3	10	**62**	1.42	10	35.20	357	392.20
4	20	**74.75**	15.01	12	31.45	448	479.45
5	20	**83**	0.77	13	32.33	444	476.33
6	20	**85.5**	12.90	11	30.88	449	479.88
7	30	**85**	855.91	13	51.92	428	479.92
8	30	**106**	6.81	14	42.05	437	479.05
9	30	**96**	2.14	11	34.13	439	473.13
10	40	**111**	85.18	13	38.70	441	479.70
11	40	**95.25**	75.98	12	46.57	432	478.57
12	40	**112.75**	39.48	13	28.87	451	479.87
13	50	**112**	3308.19	13	39.52	440	479.52
14	50	**112.5**	305.14	13	40.67	439	479.67
15	50	117.75	3600	14	35.50	444	479.50
16	60	**113**	84.62	14	31.60	447	478.60
17	60	**112**	7.63	14	28.53	451	479.53
18	60	**116**	37.66	15	40.70	437	477.70
19	70	**124**	620.99	14	41.53	438	479.53
20	70	**131**	306.80	15	45.18	434	479.18
21	70	**123.25**	512.30	14	43.27	436	479.27
22	80	131	3600	15	42.08	437	479.08
23	80	**133**	434.04	15	34.70	445	479.70
24	80	**130**	1717.90	15	45.97	434	479.97
25	90	126	3600	15	46.70	432	478.70
26	90	131	3600	15	42.77	432	470.63
27	90	133	3600	16	40.98	439	479.98

Table A3. Similarity results.

No.	No. POIs of the Instance	No. POIs of the route (NWT)	No. POIs of the route (WT)	Similarity
1	10	10	10	0.40
2	10	10	10	0.70
3	10	10	10	0.30
4	20	12	12	0.42
5	20	13	13	1.00
6	20	11	11	0.73
7	30	13	13	0.92
8	30	14	14	0.79
9	30	11	12	0.50
10	40	13	13	0.38
11	40	12	12	0.67
12	40	13	13	0.46
13	50	13	13	0.92
14	50	13	13	1.00
15	50	14	14	1.00
16	60	14	14	0.64
17	60	14	14	0.79
18	60	15	15	0.60
19	70	14	14	0.71
20	70	15	15	0.53
21	70	14	14	0.71
22	80	15	15	0.67
23	80	15	15	0.53
24	80	15	15	0.73
25	90	15	15	0.60
26	90	15	15	0.20
27	90	16	15	0.31

References

1. Bulchand-Gidumal, J. Impact of Artificial Intelligence in Travel, Tourism, and Hospitality. In *Handbook of e-Tourism*; Springer: Cham, Switzerland, 2020; pp. 1–20. [CrossRef]
2. Pinheiro, A.B.; Pinto, A.S.; Abreu, A.; Costa, E.; Borges, I. The Impact of Artificial Intelligence on the Tourism Industry: A Systematic Review. In *Advances in Tourism, Technology and Systems*; Springer: Singapore, 2020; pp. 458–469._42. [CrossRef]
3. Tussyadiah, I. A review of research into automation in tourism: Launching the Annals of Tourism Research Curated Collection on Artificial Intelligence and Robotics in Tourism. *Ann. Tour. Res.* **2020**, *81*, 102883. [CrossRef]
4. Lim, K.H.; Chan, J.; Karunasekera, S.; Leckie, C. Tour recommendation and trip planning using location-based social media: A survey. *Knowl. Inf. Syst.* **2019**, *60*, 1247–1275. [CrossRef]
5. Vansteenwegen, P.; Gunawan, A. *Orienteering Problems*; Springer: Cham, Switzerland, 2019. [CrossRef]
6. Golden, B.L.; Levy, L.; Vohra, R. The orienteering problem. *Nav. Res. Logist. (NRL)* **1987**, *34*, 307–318. [CrossRef]
7. Vansteenwegen, P.; Souffriau, W.; Berghe, G.V.; Oudheusden, D.V. Iterated local search for the team orienteering problem with time windows. *Comput. Oper. Res.* **2009**, *36*, 3281–3290. [CrossRef]
8. Pop, R.A.; Săplăcan, Z.; Dabija, D.C.; Alt, M.A. The impact of social media influencers on travel decisions: The role of trust in consumer decision journey. *Curr. Issues Tour.* **2021**, 1–21. [CrossRef]
9. Verbeeck, C.; Sörensen, K.; Aghezzaf, E.H.; Vansteenwegen, P. A fast solution method for the time-dependent orienteering problem. *Eur. J. Oper. Res.* **2014**, *236*, 419–432. [CrossRef]
10. Garcia, A.; Arbelaitz, O.; Vansteenwegen, P.; Souffriau, W.; Linaza, M.T. Hybrid approach for the public transportation time dependent orienteering problem with time windows. In *International Conference on Hybrid Artificial Intelligence Systems*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 151–158. [CrossRef]
11. Abbaspour, R.A.; Samadzadegan, F. Time-dependent personal tour planning and scheduling in metropolises. *Expert Syst. Appl.* **2011**, *38*, 12439–12452. [CrossRef]
12. Garcia, A.; Vansteenwegen, P.; Arbelaitz, O.; Souffriau, W.; Linaza, M.T. Integrating public transportation in personalised electronic tourist guides. *Comput. Oper. Res.* **2013**, *40*, 758–774. [CrossRef]
13. Li, J.; Fu, P. A Label Correcting Algorithm for Dynamic Tourist Trip Planning. *J. Softw.* **2012**, *7*, 2899–2905. [CrossRef]
14. Gunawan, A.; Yuan, Z.; Lau, H.C. A mathematical model and metaheuristics for time dependent orienteering problem. In Proceedings of the 10th International Conference of the Practice and Theory of Automated Timetabling (PATAT), York, UK, 26–29 August 2014; pp. 202–217.
15. Gavalas, D.; Konstantopoulos, C.; Mastakas, K.; Pantziou, G.; Vathis, N. Heuristics for the time dependent team orienteering problem: Application to tourist route planning. *Comput. Oper. Res.* **2015**, *62*, 36–50. [CrossRef]
16. Verbeeck, C.; Vansteenwegen, P.; Aghezzaf, E.H. The time-dependent orienteering problem with time windows: A fast ant colony system. *Ann. Oper. Res.* **2017**, *254*, 481–505. [CrossRef]
17. Vansteenwegen, P.; Gunawan, A. Other Orienteering Problem Variants. In *Orienteering Problems: Models and Algorithms for Vehicle Routing Problems with Profits*; Springer International Publishing: Cham, Switzerland, 2019; pp. 95–112. [CrossRef]
18. Murat Afsar, H.; Labadie, N. Team Orienteering Problem with Decreasing Profits. *Electron. Notes Discret. Math.* **2013**, *41*, 285–293. [CrossRef]
19. Gündling, F.; Witzel, T. Time-Dependent Tourist Tour Planning with Adjustable Profits. In Proceedings of the 20th Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2020), Pisa, Italy, 7–8 September 2020; Volume 85, OpenAccess Series in Informatics (OASIcs); Huisman, D., Zaroliagis, C.D., Eds.; Schloss Dagstuhl–Leibniz-Zentrum für Informatik: Dagstuhl, Germany, 2020; pp. 14:1–14:14. [CrossRef]
20. Isoda, S.; Hidaka, M.; Matsuda, Y.; Suwa, H.; Yasumoto, K. Timeliness-Aware On-Site Planning Method for Tour Navigation. *Smart Cities* **2020**, *3*, 66. [CrossRef]
21. Yu, V.F.; Jewpanya, P.; Lin, S.W.; Redi, A.A. Team orienteering problem with time windows and time-dependent scores. *Comput. Ind. Eng.* **2019**, *127*, 213–224. [CrossRef]
22. Laborie, P.; Rogerie, J.; Shaw, P.; Vilím, P. IBM ILOG CP optimizer for scheduling. *Constraints* **2018**, *23*, 210–250. [CrossRef]
23. Bazrafshan, R.; Hashemkhani Zolfani, S.; Al-e hashem, S.M.J.M. Comparison of the Sub-Tour Elimination Methods for the Asymmetric Traveling Salesman Problem Applying the SECA Method. *Axioms* **2021**, *10*, 19. [CrossRef]
24. Zhao, C.; Sahni, S. String correction using the Damerau-Levenshtein distance. *BMC Bioinform.* **2019**, *20*, 1–28. [CrossRef] [PubMed]
25. Raoui, H.E.; Cabrera-Cuevas, M.; Pelta, D.A. The Role of Metaheuristics as Solutions Generators. *Symmetry* **2021**, *13*, 2034. [CrossRef]