Hybrid approach for the Public Transportation Time Dependent Orienteering Problem with Time Windows

Ander Garcia[1], Olatz Arbelaitz[2], Pieter Vansteenwegen[3], Wouter Souffriau[3] and Maria Teresa Linaza[1]

[1] Vicomtech, Spain
agarcia@vicomtech.org
[2] University of the Basque Country, Spain
[3] Katholieke Universiteit Leuven, Belgium

Abstract. The Time Dependent Orienteering Problem with Time Windows (TDOPTW) consists of a set of locations with associated time windows and scores. Visiting a location allows to collect its score as a reward. Traveling time between locations varies depending on the leave time. The objective is to obtain a route that maximizes the obtained score within a limited amount of time. In this paper we target the use of public transportation in a city, where users may move on foot or by public transportation. The approach can also be applied to the logistic sector, for example to the multimodal freight transportation. We apply an hybrid approach to tackle the problem. Experimental results for the city of San Sebastian show we are able to obtain valid routes in real-time.

1 Introduction

In the Orienteering Problem with Time Windows (OPTW), several locations with an associated score and a time window can be visited in order to obtain a total trip score. Each location can be visited only once. The objective is to maximize the total trip score without violating a given time restriction.

The OPTW extension we present in this paper, the Time Dependent Orienteering Problem with Time Windows (TDOPTW), integrates public transportation. The TDOPTW consists of a number of Point Of Interests (POIs) with a fixed location, opening hours (time windows) and a given score. The public transportation network is composed by a number of fixed stops and different lines between these stops, each with a given frequency. Movements between POIs can be made by public transportation and on foot as well.

We are interested in Personalized Electronic Tourist Guides (PETs) [1], which are mobile hand-held devices. A PET should create personalized routes that maximize the tourists' satisfaction in near real-time, taking into account several restrictions, such as opening hours, duration of the visits, entrance fees and travel distances. The OPTW has been successfully used to model the optimization problem of a PET, without the integration of public transportation [2, 3].

Apart from tourism, the system presented in this paper can also be applied to the logistic sector. For example, Yamada et al. [4] applied a genetic algorithm based heuristic to model the strategic level of multimodal transport planning, particularly in freight terminal development and freight transport network design. In the context of multimodal freight transportation, each transportation destination would be equivalent to a tourist POI, having an opening and closing time and a benefit (score). A company should select the optimal combinations of customers, it would not be feasible to visit all of them. It would always be possible to transport goods directly by road. Another possibility would be to use the existing multimodal transportation network (train, ship). Nodes of this multimodal network would be equivalent to the public transportation network, with several service frequencies and costs. The objective would be to maximize the benefit.

The travel time is the main difference between the OPTW and the TDOPTW. While the travel time between the POIs is fixed in the OPTW, in the TDOPTW the travel time between POIs varies according to the leave time of the first POI and the transportation mode: a tourist can choose between going by public transportation or on foot. Thus, the difficulty of the problem increases. Moreover, we require near real-time calculations.

We present an hybrid approach to solve the TDOPTW. First we deal with the time dependent travel times using a Time Dependent Shortest Path (TDSP) algorithm. Then, we apply an Iterated Local Search (ILS) heuristic to solve the the TDOPTW. Finally we test the approach on a set of test instances based on real data for the city of San Sebastian.

This paper proceeds in Section 2 reviewing existing literature. In Section 3 we describe our approach. In Section 4 we present our experimental results. Conclusions and further work are discussed in Section 5.

2 State of the art

The TDOPTW is based on extensions of the Orienteering Problem (OP). The generalization of the OP to a multiple-day trip is known as the Team Orienteering Problem (TOP). When POIs have an associated time window, the problem is called TOPTW. Different approaches to solve the (T)OPTW have been proposed [5, 6, 3, 7].

Formin and Lingas [8] presented the Time Dependent OP (TDOP), an extension of the OP where the time needed to travel from a POI i to a POI j depends on the leave time from i. Although their approach could be adapted to deal with different transportation modes, they don't present an algorithm that can be used in real-time applications and no time windows are considered.

The model we propose is the TDOPTW. Combining previous models, we are able to integrate one or more public transportation networks to travel between POIs. The travel time required to move from one POI to the next, varies according to the transportation mode (on foot or by public transportation) and the leave time. Moreover, when a tourist chooses the public transportation the waiting time depends on the moment the tourist arrives at the boarding point. We are not aware of approaches able to solve the TDOPTW in near real-time.

The available examples about finding the shortest path either in public transportation networks or time dependent networks, focus on solving individual queries [9–11]. Moreover, there are also examples of more advanced applications [12–15] but they focus on POI-to-POI cheapest cost problems. On the other hand, we focus on a higher level selection and routing problem between multiple POIs. We explain our hybrid approach in Section 3.

3 Hybrid approach

In order to solve the TDOPTW, we need to evaluate several insertions of POIs before obtaining the route that maximizes the obtained score. Each time we evaluate the insertion of a candidate POI in a route, we need to find the shortest path between at least two POIs, i.e. we need to solve a Time Dependent Shortest Path (TDSP) problem. Solving this TDSP problem directly inside the TDOPTW algorithm each time a a query distance between POIs is executed, would violate the real-time constraint.

For example, in a random OPTW test instance of 50 locations, the OPTW algorithm executes the distance calculation operation more than 200,000 times. If instead of simple Euclidean distances (fixed travel times), each of these calculations involves solving a TDSP problem, each TDSDP calculation should not take more than 0.025 milliseconds, in order to obtain a real time calculation. This time is beyond the fastest examples available in the literature.

Thus, in order to handle the difficulty of the TDOPTW, we apply an hybrid approach combining two different heuristics. Each of them is focused on a different aspect of the problem. First we explain how we handle the time dependency introduced by the inclusion of the public transportation. Next we focus on the Iterated Local Search (ILS) heuristic tackling the OPTW. We finalize explaining the integration of both heuristics.

First, in order to avoid calculating the required TDSP in real-time, we precalculate the average travel time for each pair of POIs. This way, we handle the extra difficulty added by the public transportation without the real-time requirement.

We solve in batch the TDSP problems between all the possible pairs of POIs with leave time steps of 1 minute. Then, we obtain an average travel time for each pair of POIs. As test results show, in case of high frequency public transportation, these average travel times are accurate in practice. We store these travel times on a database for their later retrieval.

Eliminating the real time requirement, any suitable algorithm can be used to solve the TDSP problems. We have implemented a time dependent Dijkstra's Shortest Path algorithm modified to cope with public transportation. Delling [11] details the latest algorithms fulfilling this task.

Based on the conclusions of Pyrga et al.[9], we have applied a time dependent approach with simple transfer times due to its better performance. We assign to each node (POI/stop) of the system a group of labels with its arrival time, its penalized arrival time (both of them initialized to infinite) and the path followed to reach it. The algorithm has two sets of nodes: the set of settled nodes, S (nodes whose shortest distances from the source have been found), and the set of unsettled nodes, Q. The algorithm has four different steps, summarized in Algorithm 1, which are executed until the shortest distance to the destination node is found. N represents the set of nodes, d the shortest distance vector and u the actual node where the step of the algorithm is focused.

Algorithm 1: Diagram of the time dependent algorithm

```
d = ∞;
u = startLocation;
Q = (N − u);
S = (u);
while u not equals destination do
    find shortest distance to neighbors (u);
    u = extractMinimum (Q);
    S = S + u;
    Q = Q − u;
```

If the transportation mean changes, a time penalization is added to the shortest distance time. If this penalized time improves the previously existing shortest distance, the shortest distance and the path to arrive to the node are updated. The inclusion of a time penalization of 3 minutes avoids choosing routes which make users change transportation for a time saving of only some minutes. In order to limit the number of edges of the system, we have limited the maximum walking distance between bus stops to 300 meters, the maximum walking distance between POIs to 2 km and the maximum distance between POIs and stops to 1 kilometer. These are reasonable assumptions for the algorithm and scenario we consider.

The complete calculation of the average travel time matrix (50x50) takes around 90 minutes on a PC Intel Core 2 Quad with 2.40 GHz processors and 2 GB Ram. The average travel times are stored on the database for a later fast retrieval (They are available from authors upon request). Due to the off-line precalculation, this approach is highly scalable. For scenarios with a higher number of POIs, only the precalculation should take longer. Therefore, the scalability of the proposed algorithm only depends on the scalability of the OPTW algorithm. Test instances having up to 288 POIs have been solved in near real-time (less than 5 seconds [3]).

Once we have calculated these average travel times, we solve the TDOPTW in real-time as a regular OPTW. Our design and implementation is based on the algorithm proposed by Vansteenwegen et al. [3] for the (T)OPTW. In this section, we give a general description of the OPTW algorithm. For more details we refer to the (T)OPTW article.

The heuristic is based on Iterated Local Search (ILS) [16]. ILS is a metaheuristic method based on iteratively building sequences of solutions generated by an embedded heuristic called local search. This leads to much better solutions than repeating random trials of the same heuristic. The heuristic perturbs the solution found by the local search to create a new solution. Then, it takes the best solution as the new starting solution for the local search. The process is repeated until a termination criterion is met.

The local search heuristic inserts new visits to a route, one by one. For each visit i that can be inserted, the cheapest insertion time ($Shift_i$) is determined. For each of these visits the heuristic calculates a ratio, which relates the score of the POI to the time required to visit it. Among them, the heuristic selects the one with the highest ratio for insertion. This process is repeated until no more POIs can be inserted. The perturbation phase removes consecutive POIs from a route. After the removal, the heuristic shifts all visits following the removed visits towards the beginning of the route as much as possible, in order to avoid unnecessary waiting. The perturbation procedure and the local search heuristic are executed until a termination criterion is met. The heuristic returns the incumbent solution as the result.

Finally, once we find a final solution, we run a repair procedure (Algorithm 2) introducing the real travel times between the POIs. Starting from the first POI of the route, we compare the average and the real travel time, taking into account the real leave time. If the real travel time is smaller than the average one, the travel time is adapted by advancing the arrive time to the visit towards the beginning of the route as much as possible. The waiting time and the leave time of the visit are also updated.

Otherwise, if the real travel time is larger than the average one, we arrive later to the visit. If this causes a visit to become unfeasible, we remove it from the route and we update the route, moving the rest of the visits forward.

Algorithm 2: Diagram of the repair procedure

```
for i = 1 to routeLength do
    averageDistance = averageDistance(i − 1, i);
    realDistance = realDistance(i − 1, i, l_{i−1});
    a_i = l_{i−1} + realDistance;
    if (averageDistance < realDistance) and (visit i unfeasible)) then
        remove i from route;
    else
        Wait_i = max(0, Open − a_i);
        l_i = a_i + Wait_i + T_i;
```

We have used this hybrid approach to calculate benchmark results for TDOPTW instances. One way to evaluate our approach is to verify how many POIs are removed to restore feasibility. If we need to remove POIs in only a few cases (or none), the approach will work well.

4 Results

In order to test the strength of our approach, we have generated 32 test instances using real data from San Sebastian. San Sebastian is a beautiful city located at the North of Spain, just 20 kilometers away from France. San Sebastian has around 200,000 inhabitants and it is best visited combining public transportation with short walks. The city has around 50 POIs, 26 public transportation lines and 467 stops. We have varied two different criterions:

- Starting POI of each day: We have chosen 8 different starting POIs spread over the city.
- Length of each day: We have established four values for the maximum length: 2, 4, 6 and 8 hours.

Table 1 summarizes the results of the tests. The first column indicates the identification of the start POI. Then, we present the results according to the length of each day in hours (2, 4, 6 and 8). Each group of results includes the score, the number of POIs visited during the route (excluding starting and ending POIs) and the calculation time in seconds.

For the number of POIs of San Sebastian, our approach is valid to calculate trips in real-time (worst calculation time is less than 0.25 seconds). Calculation times have the same magnitude as those obtained by Vansteenwegen et al. [3] solving regular OPTW problems. Analyzing current calculation times, we can expect that our approach can handle instances with a higher number of POIs.

Regarding the obtained scores, a tourist would need around 20 hours to visit all the POIs, excluding travel times. Taking into account one day routes tend to include main POIs, which have a higher score, and skip secondary POIs, it seems reasonable that an 8 hour route collects around half of the maximum possible score (3195).

An analysis of the detailed results confirms that the integration of public transport in the routes works correctly. Besides, the average travel time based

Table 1. Summary of the results

startId	2 hours			4 hours			6 hours			8 hours		
	score	#	CPU(s)	score	#	CPU(s)	score	#	CPU(s)	score	#	CPU(s)
1	235	3	0.0	1035	14	0.1	1415	19	0.1	1745	23	0.2
2	530	7	0.0	1070	14	0.1	1485	20	0.1	1795	24	0.2
3	600	8	0.0	1115	15	0.1	1485	20	0.1	1715	23	0.2
4	605	8	0.0	1145	15	0.1	1485	20	0.1	1800	24	0.2
5	530	7	0.0	1070	14	0.1	1470	20	0.1	1810	24	0.2
6	750	10	0.0	1195	16	0.1	1485	20	0.1	1790	24	0.2
7	500	7	0.0	1035	14	0.1	1455	20	0.1	1770	24	0.2
8	650	9	0.0	1115	15	0.1	1485	20	0.1	1775	24	0.2

approach we propose to solve the time dependent problem works correctly, since in the San Sebastian's test cases, no removals are required. Due to the proximity of most POIs, public transportation is mainly worth to arrive to or leave from the city center, and also to reach distant POIs. Detailed results, including POIs' details and average travel times, are available from the authors upon request.

5 Conclusions and future work

In this paper we propose an hybrid approach to tackle the Time Dependent Orienteering Problem with Time Windows (TDOPTW). Our objective is to integrate public transportation in personalized tourist routes' planning. The approach can also be applied to the logistic sector, for example to the multimodal freight transportation. Our approach is based on two heuristics. The first one involves a precalculation step, where we calculate the average travel times between all pairs of POIs. With these average travel times, the second heuristic solves the TDOPTW as a regular OPTW. Then, we adapt the arrival and leave times of the visits according to the differences between the average travel times and the (time dependent) real travel times. If due to the adaptation some visits become unfeasible, we remove them from the final route proposed to the tourist.

We have applied and tested the approach for a set of 32 test instances based on San Sebastian, a medium size city with around 50 POIs, 26 public transportation lines and 467 stops. We have been able to solve these TDOPTW instances in real-time. Our hybrid approach can also be applied to other scenarios with time dependent travel times.

The next step is to implement the algorithm for multiple days. We also plan to test the routes with real tourists, in order to check the tourist's quality perception of the routes. Regarding the algorithm, our intention is to apply it in other cities, with different public transportation network topologies and POI distributions. For bigger cities, the Dijkstra's Shortest Path algorithm should be replaced by more optimized algorithms.

Acknowledgments

The authors would like to thank the Basque Government for partially funding this work through the neurebide and etourgune projects and to the Centre for Industrial Management of the Katholieke Universiteit Leuven for hosting Ander Garcia as a guest researcher during 2009. Pieter Vansteenwegen is a postdoctoral research fellow of the "Fonds Wetenschappelijk Onderzoek - Vlaanderen (FWO)".

References

1. Vansteenwegen, P.: Planning in tourism and public transportation. PhD thesis, Centre for Industrial Management, Katholieke Universiteit Leuven, Belgium (2008)

2. Garcia, A., Linaza, M., Arbelaitz, O., Vansteenwegen, P.: Intelligent routing system for a personalised electronic tourist guide. In: Information and Communication Technologies in Tourism 2009, Amsterdam, The Netherlands, SpringerWienNewYork (2009) 185–197
3. Vansteenwegen, P., Souffriau, W., Vanden Berghe, G., Van Oudheusden, D.: Iterated local search for the team orienteering problem with time windows. Computers & Operations Research **36**(12) (2009) 3281–3290
4. Yamada, T., Russ, B., Castro, J., Eiichi, T.: Designing multimodal freight transport networks: A heuristic approach and applications. Transportation Science **43**(2) (2009) 129–143
5. Righini, G., Salani, M.: Decremental state space relaxation strategies and initialization heuristics for solving the orienteering problem with time windows with dynamic programming. Computers & Operations Research **36**(4) (2009) 1191–1203
6. Tricoire, F., Romauch, M., Doerner, K., Hartl, R.: Heuristics for the multi-period orienteering problem with multiple time windows. Computers & Operations Research **37**(2) (2010) 351–367
7. Montemanni, R., Gambardella, L.: Ant colony system for team orienteering problems with time windows. Foundations of Computing and Decision Sciences: accepted (2009)
8. Fomin, F., Lingas, A.: Approximation algorithms for time-dependent orienteering. Information Processing Letters **83**(2) (2002) 57–62
9. Pyrga, E., Schulz, F., Wagner, D., Zaroliagis, C.: Efficient models for timetable information in public transportation systems. Journal of Experimental Algorithmics **12** (2008) 1–39
10. Ding, B., Yu, J., Qin, L.: Finding time-dependent shortest paths over large graphs. In: EDBT '08: Proceedings of the 11th international conference on Extending database technology, New York, NY, USA, ACM (2008) 205–216
11. Delling, P.: Engineering and Augmenting Route Planning Algorithms. PhD thesis, Fakultat fur Informatik Universitat Fridericiana zu Karlsruhe (2009)
12. Zografos, K.G., Androutsopoulos, K.N.: Algorithms for itinerary planning in multimodal transportation networks. IEEE Transactions on Intelligent Transportation Systems **9**(1) (2008) 175–184
13. Chiu, D.K.W., Lee, O.K.F., Leung, H.F., Au, E.W.K., Wong, M.C.W.: A multimodal agent based mobile route advisory system for public transport network. In: System Sciences, 2005. HICSS '05. Proceedings of the 38th Annual Hawaii International Conference on. (2005) 92.2
14. Tumas, G., Ricci, F.: Personalized mobile city transport advisory system. In: Information and Communication Technologies in Tourism 2009, Amsterdam, The Netherlands, SpringerWienNewYork (2009) 173–183
15. Zenker, B., Ludwig, B., Schrader, J.: Rose assisting pedestrians to find preferred events and comfortable public transport connections. In: ACM Mobility Conference 2009: accepted, ACM New York, NY, USA (2009)
16. Lourenço, H.R., Martin, O., Stutzle, T.: Iterated local search. In Kochenberger, F.G., G., eds.: Handbook of Metaheuristics. Kluwer Academic Publishers (2003) 321–353