MISTA 2013

A fast solution method for the Time-Dependent Orienteering Problem with time windows

C. Verbeeck ·
E.-H. Aghezzaf ·
P. Vansteenwegen

1 Introduction

This research focuses on developing an efficient solution method to solve the time-dependent orienteering problem with time windows (TD-OPTW), requiring only a few seconds of computation time.

In Orienteering problems an optimal combination of vertices (locations) needs to be selected and the routing between the vertices needs to be optimized. In the time-dependent version the travel time between two vertices depends on the departure time at the first vertex.

Orienteering problems [8] are typically used in logistic planning tools where each vertex resembles a customer and the score reflects the profit margin. Furthermore, they serve as the basic problem formulation of personalized touristic trip planners where a vertex stands for a point of interest and the score indicates the personal interest of the tourist. For a longer list of practical and real-life applications of the OP and its variants, we refer to the recent survey by Vansteenwegen et al [8]. The time-dependent problem formulation allows to tackle congestion related issues in routing problems such as morning and evening peaks on the highways or crowded city center traffic situations [5]. Secondly, multi-modal applications for logistic or touristic trip planners also rely on TD-OP solution methods [1, 3]. The most common example is the combination of walking and using public transport, where the waiting time at a bus station and the time table of the bus result in time-dependent travel times. Due to the rise in congestion problems on the one hand and the acceptance of mobile devices with GPS and internet connection on the other hand, the fast construction or update of routes

C.Verbeeck
Ghent University, Department of Industrial management
E-mail: Cedric.Verbeeck@ugent.be

E.-H. Aghezzaf
Ghent University, Department of Industrial management
E-mail: elhoussaine.aghezzaf@ugent.be

P. Vansteenwegen
KU Leuven, Centre for Industrial Management, Traffic and Infrastructure
E-mail: pieter.vansteenwegen@cib.kuleuven.be

based on congestion information increasingly becomes a necessity for a number of interesting business applications.

2 Methodology

Since high-quality solutions are required and the computation time should be limited to only a few seconds, the literature on vehicle routing suggests the implementation of a local search based metaheuristic [7]. These solution methods try to find better solutions combining efficient local search procedures with mechanisms to escape local optima. Three time-dependent local search procedures with a local evaluation metric have been successfully implemented. A local evaluation metric prevents a full and time-consuming evaluation of a solution after every modification attempt and enables an efficient checking and updating mechanism. Therefore, we store for every vertex in the current solution the maximum amount of time that a visit to it can be postponed before the solution becomes infeasible. Firstly, the problem-specific *insert* local search procedure iteratively attempts to insert non-included vertices into an existing solution. Secondly, a time-dependent *replace* procedure tries to replace a vertex from the current solution with a non included vertex with a higher score. Thirdly, the *swap* local search procedure tries to exchange two solution member vertices in order to save travel time.

The basic metaheuristic framework of an ant colony system (ACS) was used. This choice was motivated by the fact that generally very complex problems require simple solution frameworks. The ACS is based on the ant colony optimization algorithm (ACO) of Ke et al [4] and Schilde et al [6] for the OP and on the ACS of Donati et al [2] for the TD-VRP. The ACS starts by sequentially creating a number of initial solutions. The construction process starts from the first vertex and in succession adds a new vertex to the solution based on *greedy information* and *pheromone trails* of the arcs leading to that vertex. The greedy information is based on a ratio which consists of the score of the next vertex and the distance towards the next vertex. After the creation of a complete solution, the included arcs are made less attractive for the following solution construction procedures, in order to increase diversification. This is done by depreciating the pheromone values with an evaporation rate factor (*local pheromone update*). Once all solutions of one iteration are created, they are improved, using the following sequence of local search procedures in a first-improving manner until no more feasible improvements can be found: swap, insert and replace. The solution with the highest score is stored. Finally, the arcs that are used in this solution are made more attractive to be used in the solution construction procedure of succeeding iterations, by increasing their corresponding pheromone value. These steps are repeated during a specified number of iterations. To prevent that only a couple of arcs dominate this solution construction procedure, the pheromone values are reset during a *global pheromone update* when no improvement can be found during a certain number of iterations.

Realistic road information of the geographical region of Flanders in Belgium is used as provided by OpenStreetMap (OSM). Using this large amount of unstructured and partially redundant information would be too complicated to use in a metaheuristic so we filtered and efficiently restructured the information without sacrificing the quality of the road network. These steps resulted in a strongly connected graph representing a road network with 84 720 vertices and 116 683 arcs. Once the static road network is created, congestion information should be added. However, OSM offers no congestion information apart from the road types, so as preliminary solution we mapped each

OSM road type to a corresponding arc congestion category based on the authors insight. To lower the execution time of the travel time calculation, an adapted version of the shortest path algorithm of Dijkstra was implemented using binary heaps and the speed model of Donati et al [2] to calculate the time-dependent travel times on each arc. However, this implementation was still too costly to use during the execution of the ACS. Therefore, the travel time between the subset of vertices, representing the customers or depots on the graph, is calculated in a pre-processing phase. After these steps a new virtual network has been constructed. This network is implemented by a completed graph which consists of nodes representing the customers to visit and virtual arcs with time-independent and time-dependent travel time profiles which model the travel time behavior on a concatenation of arcs.

3 Results

To test the proposed algorithm, we created a set of 36 instances by randomly selecting respectively 20, 50, 100 and 250 vertices in our OSM road network and generating 3 variations in maximum allowed travel time and 3 variations in time window severity. Subsequently, all instances were solved first as time-*independent* OPs using the commercial solver CPLEX. During this optimization, the travel time is calculated using, on each arc, the maximum speed of the corresponding arc category. Following this optimization, the optimal time-*independent* solution is used to create a known optimal solution by slightly modifying the time-dependent instance. For each virtual arc included in the optimal time-independent solution, the travel speed is set to its maximal value, but only during the period in time this virtual arc is traversed. Since the travel speed is only modified in some of the time periods of these arcs, these arcs still have time-dependent travel times. The time-dependent travel times on all other arcs are not modified. Finally, when violations occur due to the time window restrictions, the time windows are altered to match the optimal time-independent solution. In this way, it is artificially ensured that the time-independent optimal solution is also an optimal solution to its time-dependent variant.

When comparing the results, displayed in Table 1, obtained by our approach on these 36 instances with the known optimal solution, an average gap of 0.2% is obtained, with an average computation time of 0.3 seconds. The maximal gap is 4.1% and the maximal computation time is 0.7 seconds. Furthermore, we conclude that the strength of this algorithm lies in the interaction effect of the repeated insert and replace local search procedures.

4 Conclusion

This paper presents the time-dependent orienteering problem with time windows, a relatively unstudied extension to the orienteering problem where time windows are considered together with travel times which depend on the departure time. In the considered tourist and logistical applications, this modification models multi-modal transport functionality, as well as congestion troubled vehicle routing planning. Most of these applications require solutions within a short amount of computation time.

The proposed algorithm obtains promising results on realistic time-dependent test instances, based on the open source database OpenStreetMap. On average, the algo-

Table 1 Results for the TD-OPTW

Amount vertices	best % gap	avg % gap	worst % gap	CPU s
20	0.0	0.0	0.0	0.1
50	0.0	0.2	0.8	0.3
100	0.0	0.1	0.5	0.4
250	0.9	1.0	1.1	0.6
overall maximum	3.6	4.1	6.4	0.7
overall average	0.1	0.2	0.5	0.3
% optimal	96.8	83.9	83.9	

rithm obtains solutions with an score gap of 0.2% using 0.3 seconds of computation time. The fast execution time of this algorithm enables some interesting business applications where it is necessary to update routes when new traffic information becomes available and to provide proper guidance to truck drivers or tourists on the road.

Further research will focus on fine-tuning and testing the proposed algorithm on a very large network like the Benelux road network with detailed historic travel time information per road segment. Subsequently, the time-dependent variant of the team orienteering problem (TD-TOPTW) will be researched. This rather interesting extension of the problem allows to optimize the routing of a fleet of vehicles, instead of only one vehicle, which is certainly useful for logistic companies.

References

1. Abbaspour A, Samadzadegan F (2011) Time-dependent personal tour planning and scheduling in metropolises. Expert Systems with Applications 38:12,439–12,452
2. Donati A, Montemanni R, Casagrande N, Rizzoli A, Gambardella L (2008) Time dependent vehicle routing problem with a multi ant colony system. European Journal of Operational Research 185:1174–1191
3. Garcia A, Vansteenwegen P, Arbelaitz O, Souffriau W, Linaza M (2013) Integrating Public Transportation in Personalised Electronic Tourist Guides. Computers & Operations Research 40:758–774
4. Ke L, Archetti C, Feng Z (2008) Ants can solve the team orienteering problem. Computers and Industrial Engineering 54:648–665
5. Li J, Wu Q, Li X, Zhu D (2010) Study on the time-dependent orienteering problem. In: International Conference on E-Product E-Service and E-Entertainment (ICEEE)
6. Schilde M, Doerner K, Hartl R, Kiechle G (2009) Metaheuristics for the biobjective orienteering problem. Swarm Intelligence 3:179–201
7. Sörensen K, Sevaux M, Schittekat P (2008) Adaptive, self-adaptive and multi-level metaheuristics, Lecture Notes in Economics and Mathematical Systems, vol 136, Springer, London, chap "Multiple neighbourhood search" in commercial VRP packages: evolving towards self-adaptive methods, pp 239–253
8. Vansteenwegen P, Souffriau W, Van Oudheusden D (2011) The orienteering problem: a survey. European Journal of Operational Research 209:1–10