Global and reactive routing in urban context: first experiments / first difficulty assessment

Penélope A. Melgarejo
IBM, France

Thomas Baudel
IBM, France

Christine Solnon
INSA Lyon, France

September 15, 2012

1 Introduction

Optimod'Lyon (www.optimodlyon.com) is a project initiated by the Grand Lyon to improve mobility in Lyon's agglomeration. A first goal is to centralize all mobility data and provide a service of prediction of traffic conditions during the day. Another important goal (called Smart Deliveries) is to optimize planned tours of transporters in a global, time-dependent and reactive way. The optimization is global in the sense that all mobility demands are optimized at a whole; it is time-dependent as predicted travel times vary during the day according to predicted traffic conditions; and it is reactive as it dynamically adapts tours to unexpected events like car accidents.

In this paper, we compare different Constraint Programming (CP) models for addressing the Smart Deliveries routing problem. In this first study, we consider a simplified version where only one tour is to be found, and we focus on the ability to handle time-dependent data (wherein predicted travel times vary with time), and additional constraints specific to each individual transporter (such as time windows, ordering constraints to meet pickup-and-delivery types of moves, as well as various preferences).

2 Definition of the optimization problem

We basically consider the Time-Dependent Traveling Salesman Problem (TDTSP), which involves finding the shortest hamiltonian path in a complete asymmetric graph whose vertices correspond to visit points. In the TDTSP, the cost c_{ij} of an arc (i, j) depends on the time of departure from i. More precisely, $c_{ij}(t)$ is the expected duration of travelling from i to j when leaving i within the time window t. This time-dependent cost function is calculated during a preprocessing step from the prediction of traffic conditions: for every couple (i, j) of visit points and every time window t, we compute the shortest path from i to j when leaving i within t by using the Time-Dependent Dijkstra algorithm [DSSW09].

3 CP models

State-of-the-art complete approaches for solving the TSP are based on Integer Linear Programming (ILP) [App06]. However, some additional constraints are difficult to express, for example, not being allowed to change the position of some stops in the list of more than a given constant because of charging constraints like the weight of the packet to be delivered.

Recent work [BvHR+12] has shown us that CP is competitive with state-of-the-art special-purpose TSP solvers for medium size instances. Moreover, additional constraints are often very easily expressed in CP, and they usually strongly improve the solution process by reducing the search space in an a priori way.

We compare two different CP models for the TDTSP, $M1$ and $M2$. Input consists of n, the number of stops, m and s, the number of time steps and their size, and a time-dependent cost matrix.

In $M1$ we have two types of variables, pos_i gives the location of the i-th stop in the tour and t_i gives the time of departure from the i-th stop. Constraints are:

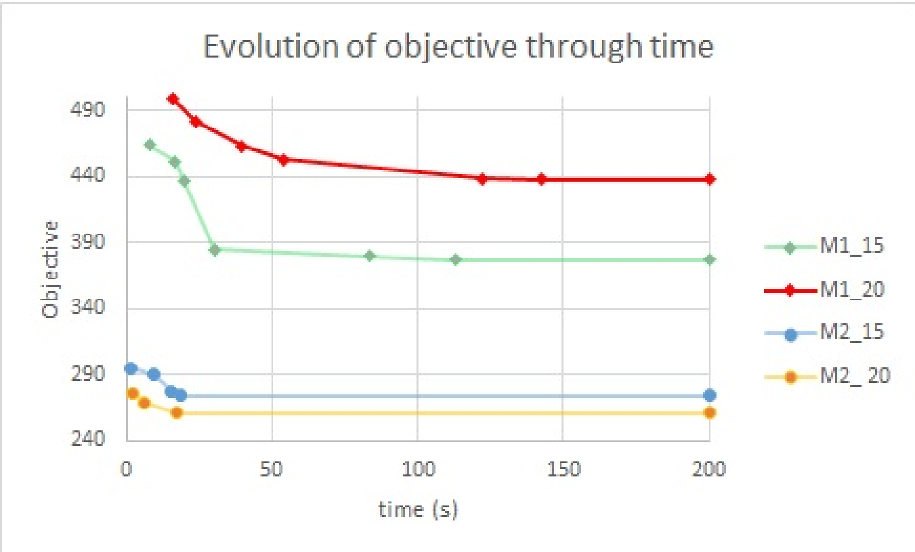
$$pos_1 = 1$$
$$pos_n = n$$
$$t_1 = 0$$
$$allDifferent(pos)$$
$$t_{pos_{i+1}} >= t_{pos_i} + c_{pos_i, pos_{i+1}, (t_{pos_i}/s)}, \forall i \in 1, ..., n-1$$

For $M2$ we used scheduling tools, the variables are: $stop_i$, time intervals representing the start and the end of the stop at the i-th location, and $tour$, a sequence of the interval variables $stop$. The three first constraints are the same, the other two are:

$$noOverlap(tour)$$
$$startOfNext(tour, stop_i) >= endOf(stop_i) + c_{i, next(tour, stop_i), endOf(stops[i])/s},$$
$$\forall i \in 1, ..., n-1$$

4 Experimental results

We tested both models with instances of 10, 15 and 20 stops randomly generated with a model which simulates rush hours when defining time-dependent costs. Tests were made using ILOG CPO 12.4 and CPU time was limited to 200 seconds per instance.



Specifying search phases to $M2$ enhanced a lot its performance but did not have the same effect on $M1$.

5 Conclusion

To choose the best approach to this problem we still have to test models with real data from the city and transporters and also to compare performances with ILP models [Bro12].

Some of our perspectives are to dynamically adapt tours to events interfering on their routes, to consider more than one tour at a time and to robustify initial route propositions through stochastic optimization.

References

[App06] Bixby R. E. Chvtal V. Cook W. J. Applegate, D. L. *The traveling salesman problem: A computational study.* Princeton: Princeton University Press, 2006.

[Bro12] Juan José Miranda Bront. *Integer Programming approaches to the Time Dependent Travelling Salesman Problem*. PhD thesis, Departamento de Computación, FCEyN, Universidad de Buenos Aires, 2012.

[BvHR+12] Pascal Benchimol, Willem Jan van Hoeve, Jean-Charles Régin, Louis-Martin Rousseau, and Michel Rueher. Improved filtering for weighted circuit constraints. *Constraints*, pages 205–233, 2012.

[DSSW09] Daniel Delling, Peter Sanders, Dominik Schultes, and Dorothea Wagner. Engineering route planning algorithms. In *ALGORITHMICS OF LARGE AND COMPLEX NETWORKS. LECTURE NOTES IN COMPUTER SCIENCE*. Springer, 2009.