Two algorithms for a variant of Capacitated Pickup and Delivery Problem with Time Windows, and Transfers

José-L. Figueroa[1], Alain Quilliot[1]

[1] Université Clermont-Auvergne, LIMOS, Clermont-Ferrand, France
{jofigueroa, alain.quilliot}@ismima.fr

Key-words *: pickup and delivery with transfers, time-expanded graph, dynamic programming.*

Introduction

The family of pickup and delivery problems covers a broad range of optimization problems and has been studied extensively due to its relation with real-world transportation tasks. A preliminary survey of these problems can be found in [1]. Like other families of vehicle routing problems, the pickup and delivery problems can be classified in several subfamilies according with the nature of the requests involved (e.g. static or dynamic requests), the location and number of depots (many-to-many, one-to-many-to-one, and one-to-one problems; see [2]), or the type of constraints involved (time windows, capacities, transfers, etc.). In this work we address a variant of Capacitated Pickup and Delivering Problem with Time Windows and Transfers; we are particularly interested in the solutions when we allow new transfers between fleet vehicles. A related reference can be found in the work of Berbeglia, Cordeau, and Laporte [3].

Description of the Problem

In this variant of the problem, a set of vehicles routes with transfers has already been established, and a collection of additional one-to-one requests arrive in a sequence. For each request, in the order given by the sequence, we need to decide either if we accept the request and modify the last tours configuration to incorporate the new request, or if we reject the request. The vehicles routes evolve with each accepted request in an iterative way, this is, if we accept a request r_i obtaining a new set of vehicles routes T_i, the next accepted request must be incorporated into T_i. The objective is to determine, for each request in the sequence, if it can be incorporated in the current set of vehicles routes, and if so, find a good quality or optimal solution, minimizing the path followed by the vehicles to transport the request.

Model and Algorithms Proposed

This work proposes a novel time-expanded graph formulation and two algorithms for address this problem, treating each request independently.

The graph model consists of an acyclic time-expanded digraph, including the original arcs in the routes of vehicles, and collections of special arcs representing previous transfers, deviations and the

new allowed transfers between vehicles. Although, the size of this graph can be quadratic (with respect of the total number of vertices in the tours), we can reduce the number of special arcs by using a threshold parameter over the length of the arcs only to maintain promising movements.
The first algorithm, is a Dijkstra based procedure to search for a good quality solution for a particular request in the sequence of the problem. Tuning the digraph threshold parameter, this algorithm can be turned into an algorithm with polynomial space-time requirements.
The second algorithm handles also one request at time. It searches for a solution, but using a dynamic programming approach to grow a rooted tree of feasible ``states" that eventually can become complete solutions. Each feasible state in this tree, correspond to a partial path allowing to pick up the current request. We use a special structure to store some useful information related with the paths, and we use this information to discard quickly some unfeasible nodes. Also, to reduce the combinatorial explosion, we use a partial ordering over the nodes and we delete redundant nodes. It's also possible, during the exploration of the solutions tree, to limit the number of authorized transfers.

The heart of both algorithms is a consistence test that permits determine if the collection of arcs corresponding to a partial solution respects the given time window. The complexity of this consistence test can be proved polynomial $O(|E|^3)$ in the size of the number of arcs $|E|$ in the vehicles routes, but in practice this algorithm exhibits a linear complexity $O(|E|)$.

Finally, when these algorithms find a feasible solution, they return as solution a new time-expanded digraph allowing to transport the given request. The new time-expanded digraph is used as input of the algorithms for the next request.

Results

We have tested these algorithms on two sets of instances. The first test consists of 100 pseudorandom instances with one request. They are grouped together in collections of 10 instances with the same number of tours (varying from 3 to 12 tours, up to 8 arcs on average).

The second set of instances consists of 20 pseudorandom instances with 10 requests (taking 2 instances from each group of the previous set of instances). The goal of the experiments on this set is to understand the evolution of the type of solutions when the instance become more constrained due to new transfers and the saturation of some arcs.

We have verified that the algorithm based on Dijkstra can find good quality solutions in most of the instances. For the second algorithm, we obtained interesting running times, especially when we limit the number of authorized transfers to a number less than 3.

References

[1] Maria Batarra, J. F. Cordeau, and Manuel Iori. *"Pickup-and-Delivery Problems for Goods Transportation"*. Vehicle routing: problems, methods, and applications, pp. 161-191, 2014.

[2] Gerardo Berbeglia, Jean-François Cordeau, and Gilbert Laporte. *“Dynamic pickup and delivery problems”*. European Journal of Operational Research, Volume 202, Issue 1, pp. 8-15, 2010.

[3] P. Bouros, D. Sacharidis, T. Dalamagas, and T. Sellis. *“Dynamic Pickup and Delivery with Transfers”*. Advances in Spatial and Temporal Databases. SSTD 2011, Lecture Notes in Computer Science, vol 6849. Springer, Berlin, Heidelberg, pp. 161-191, 2011.