Exact and Heuristic Approaches to the Double TSP with Multiple Stacks

Hanne L. Petersen

The Double Travelling Salesman Problem with Multiple Stacks (DTSPMS) is concerned with finding the shortest route performing pickups and deliveries in two separated networks using only one container. Repacking is not allowed, instead each item can be positioned in one of several *rows* in the container, such that each row can be considered a LIFO (Last-In-First-Out) stack, while no mutual constraints exist between the rows. Additionally no stacking is allowed and all items are required to be uniform.

In the DTSPMS a set of orders is given, each one requiring transportation of one item from a given location in one region to a given location in another region. The two regions are far apart, and thus some long-haul transportation is required between the depots in each of the regions. This long-haul transportation is not part of the problem. All pickups and deliveries must be carried out with the same container, which cannot be repacked along the way. Hence the problem to be solved consists of the two geographically separated problems of picking up and delivering the items in a feasible way with regard to container loading. Items in the container can only be accessed from the opening in one end of the container, and the objective is to find a pair of tours with the shortest total length. No time windows are considered in this problem.

In practice this can occur when an empty container is loaded onto a truck that performs the pickups, then returned by that truck to a local depot where it is transferred onto e.g. a train or another truck, which then performs the long-haul transportation. Upon arrival at a depot in the delivery region, the container is again transferred to a truck which carries out the actual deliveries.

The connection between the two tours to be found is given by the loading of the container. Since no repacking is allowed, the only items that can be delivered next at any time during delivery are the ones that are accessible from the opening of the container. This implies that the loading is subject to LIFO constraints.

However, in the DTSPMS there is no LIFO ordering for the container as a whole. Rather, it contains several loading rows, each of which can be considered a LIFO stack, but all rows are independently accessible.

In a real life situation the items to be transported would typically be Euro Pallets, which fit 3 by 11 on the floor area of a 40-foot pallet container, thus providing three independent loading rows.

A solution to a given problem consists of three elements: a *pickup route*, a *delivery route*, and a *row assignment*, which for each item tells which loading row it should be placed in. A row assignment only gives the row that each item should be placed in, and does not indicate which position the item will occupy in that row. Given a route (pickup or delivery) and a row assignment, one can construct the *loading plan*, which gives the exact position of each item in the container.

The problem may at first glance seem purely theoretical, since the extra mileage incurred by not being able to repack can seem prohibitive. However the problem has been encountered in real life applications, where this extra mileage is justified by the wages stemming from handling and requirements to comply with union restrictions (the driver is not allowed to handle the goods).

Special cases of the problem occur when the number of loading rows is equal to one or to the number of orders n. In both cases the problem of finding a row assignment for the solution becomes irrelevant.

In the single row case the pickup route will strictly dictate the delivery route (or vice versa), and the two routes need to be exact opposites. In this case the problem can be solved by adding the distance matrices of the two graphs, and solving a regular TSP for the resulting distance matrix.

Additionally a solution to the single row problem will always be a feasible solution to any multiple row problem, and thus provides an easily obtainable upper bound.

Conversely, when the number of loading rows equals the number of orders n, the two routes do not impose any restrictions on each other and the optimal solution to the n-row DTSPMS consists of the optimal solutions to the two independent TSPs. This solution provides an easy lower bound to any problem with fewer than n rows.

In the setting described here the problem size is naturally limited by the size of a 40-foot container, thus the typical limit would be 33 items, positioned in a 3 by 11 grid.

To increase the problem size beyond this the problem would most likely have to be reformulated as a DVRPMS, Double Vehicle Routing Problem with Multiple Stacks, for any application currently known.

Solutions to the DTSPMS

Different solution approaches have been applied to the DTSPMS. Initially various heuristic approaches were implemented, including Tabu Search (TS), Simulated Annealing (SA) and Steepest Descent (SD) with restarts. In particular Simulated Annealing provided promising results. Using running times of three minutes on problems of size 3 by 11, solution values were obtained that were around 10% above the best solution value known at the time.

During the development of the metaheuristic solution methods, it was necessary to decide on a operator/neighbourhood structure that would maintain solution feasibility. Due to the composite nature of a solution to a DTSPMS it has not been possible to construct a single operator that is able to cover the entire solution space while maintaining feasibility. Instead two different operators were developed, that in combination achieve this task. Each focuses on a particular part of the solution - one focuses on the routing, while not affecting the loading plan, and the other focuses on the loading, while only performing changes to the routing where necessary.

These two operators were then used in combination for each of the three above-mentioned heuristic approaches in accordance with the nature of each algorithm. Thus the operator was chosen randomly for each iteration of SA and SD, and following a fixed pattern for TS since the latter is a deterministic algorithm.

As part of the regular parameter tuning that is needed for the heuristic algorithms (length of tabu list for TS, and temperatures for SA) some parameters were included to describe how the operators were combined. One interesting observation was that when there was no difference in the resource consumption of the two operators (i.e. when a random neighbour is chosen in SA, but not when the entire neighbourhood is searched in TS) the proportion of all iterations that used one particular operator had a surprisingly small effect on the final objective value, as long as both probabilities were positive. For TS where the operator type was not chosen randomly for each operation, changing rather frequently between the two operators was additionally found to be beneficial.

The work concerning heuristic solutions to the DTSPMS has been described in further detail in [1].

Subsequently, some different exact approaches have been implemented.

An initial approach was based on repeatedly solving the two TSPs, adding cuts as violated loading constraints were discovered in the resulting solution. The TSPs could then be resolved until a pair of solutions were found where a feasible row assignment existed.

Such violated loading constraints could be identified by considering the k last customers of the pickup route and the ($k-1$ or) k first customers of the delivery route and solving the loading feasibility problem for these customers for increasing values of k.

As this subproblem considers a partial solution to the original problem, it will often include single customers who appear only in one of the routes, say the pickup route. In order for the feasibility problem to make sense this customer must then be inserted at some position in the delivery route. Since we know that the customer is not among the k first customers in the delivery route it can be placed *after* the k "real" delivery customers. Additionally it must be ensured that when several customers appear in only one of the routes, they are inserted in such a way, as to not obstruct any otherwise feasible solutions. Hence all customers that appear in the pickup route but not in the delivery route are assigned "artificial" positions in the delivery route that are reversed compared to their pickup positions. The argument can obviously be reversed when assigning artificial positions to customers that appear only in the delivery route.

References

[1] H.L. Petersen, "Exact and Heuristic Approaches to the Double TSP with Multiple Stacks", Technical Report, CTT, DTU, 2006