Operations Research B

Lars Burghardt, Alexander Wördekemper, Ahmad Hashemi

22.02.2017

Contents

1 Initial solution

For our initial solution we decided to use a sequential construction algorithm with a hill climbing algorithm introduced in [$HoMu$12]. We have chosen these algorithms because they seemed to produce good and fast initial solutions in the literature. We had to do some little adjustments because we could not open as many routes as we want to because we have a fix number of vehicles. In line 1 we chose a random customer from a list of customers which have no route yet and insert it into the route. In line 2 the hill climbing algorithm improves the route with the new customer. In line 3 we check if the new route is feasible. If it is feasible the customer can stay in the route, else the customer gets removed again. Line 5 says that if no of the left customers can be added to the route without making it infeasible the algorithm starts again with a new route. After all vehicles have their route we check if the solution is feasible. If it is feasible we chose it as our initial solution. Otherwise we restart the whole sequential construction algorithm.

In line 1 of the hill climbing algorithm we have a for loop for every pair of locations in the route. In line 2 we switch the locations if the latter location has a later upper time window. In line 3 we calculate the new cost function. The cost function is dependant of the route duration, the number of time window violations and the number of capacity violations. They are weighted with w1, w2 and w3 equal to 1. If the new route is better we keep the route in line 4. Else we switch it back (line 5).

Algorithm 1 Sequential construction

Chose random customer which is not yet in a route and insert it into the route
Let hill climbing algorithm improve the route
if (route is feasible) **then** customer stays in the route
else customer gets removed again
if (no feasible customer can be added to the route) **then** algorithm starts again for the next route

Algorithm 2 Hill climbing

1: **for** each possible pair of locations **do**
2:     Switch the locations if the latter location has a later upper time window
3:     Calculate the cost function of the new route
4:     **if** (new root is better) **then** keep the new route
5:     **else** switch it back

1.1 Different approaches

As we found out that we did not find initial solutions for larger instances we tried some different approaches to solve the problem

1.1.1 Removing the customer with the largest distance

The sequential construction algorithm adds random customers to the route and after the root gets improved by the hill climbing algorithm we remove the customer we had just added if the new route is infeasible. Instead of removing the just added customer we tried to remove the customer which needed the most time on the route. For every customer and it's destination we added the distance from the last node to the customer/destination and from the customer/destination to the next node. We removed the customer which needed the most time. In experiments we found out that this approach was no improvement compared to the one we had before.

2 Large neighborhood search

ToDo

3 Relevant work in the literature

ToDo

4 How to compile and run the code

ToDo

5 Experimental investigation of our approach's components

ToDo

6 Experimental investigation of our approach's performance

ToDo

7 Literature

[$HoMu$12] M. I. Hosny and C. L.Mumford, “Constructing initial solutions for the multiple vehicle pickup and delivery problem with time windows”, Journal of King Saud University, Computer and Information Sciences, vol. 24, no. 1, pp. 59–69, 2012.

[$JaHe$11] S. Jain , P. Van Hentenryck, “Large neighborhood search for dial-a-ride problems”, In: Principles and practice of constraint programming, Notes in computer science, vol. 6876. Springer, 2011.