



## Exact and heuristic methods for the Split Delivery Vehicle Routing Problem

Gamst, Mette; Lusby, Richard Martin ; Røpke, Stefan

*Publication date:*  
2022

*Document Version*  
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

*Citation (APA):*  
Gamst, M., Lusby, R. M., & Røpke, S. (2022). *Exact and heuristic methods for the Split Delivery Vehicle Routing Problem*. Abstract from Route 2022, Snekkersten, Denmark.

---

### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

# Route 2022

May 22 – May 25

Comwell Borupgaard, Snekkersten, Denmark



# Exact and heuristic methods for the Split Delivery Vehicle Routing Problem

Mette Gamst<sup>1</sup> and Richard Lusby<sup>1,\*</sup> and Stefan Ropke<sup>1</sup>

\*Speaker

<sup>1</sup>mega@dtu.dk, rmlu@dtu.dk, ropke@dtu.dk, Department of Technology, Management and Economics, Technical University of Denmark, Denmark

We propose a heuristic method and a branch-and-cut algorithm for the Split Delivery Vehicle Routing Problem (SDVRP). The SDVRP is a variant of the Vehicle Routing Problem, where customer demand can be satisfied through multiple visits and where each vehicle has a known capacity. The objective is to minimize the total travelled distance.

The heuristic method serves as an initial upper bound to the branch-and-cut algorithm and is based on an Adaptive Large Neighborhood Search (ALNS). The ALNS is initialized by a constructive solution based on the work of Wilck and Cavalier. For each vehicle, one of the three farthest customers with remaining demand is assigned to a subset and the closest pair of customers to the subset are iteratively considered to also add. When vehicle capacity has reached a certain level, the optimal route for the subset is found by solving a Travelling Salesman Problem. When the ratio of customers to the minimum number of vehicles needed is low (i.e.,  $< 2$ ), vehicle routes are likely to be short and connect two to three closely located customers. For instances with a low customer to vehicle ratio, we generate 50 candidate solutions, select the best, and then attempt to improve on this by solving the Route Based Formulation (RBF) of Archetti et al comprised of a partially enumerated set of two-customer routes and a set of at most 500 promising 3-customer routes. The best solution found within 300 seconds is then used in the ALNS.

The ALNS repeatedly destroys and repairs solutions and applies record-to-record as the acceptance method. Three destroy methods are implemented: *Random destroy*: removes  $\alpha$  random customers from their routes. *History destroy*: removes  $\alpha$  customers from positions that in the past have led to inferior solutions. *Route destroy*: removes randomly selected routes until  $\alpha$  customers have been removed. The first two destroy methods remove split customers fully from all routes, while the third destroy method only partially removes split customers. Three repair methods are implemented: *Parallel insertion* attempts to greedily insert unassigned customers in all routes. Customers may be split. *Regret insertion* and *Regret split insertion* which differ in that the former does not split customers and may thus be unable to insert certain customers, while the latter handles split customers by calculating the cost of inserting them into several routes.

The ALNS saves generated routes together with their best solution value. The ALNS is restarted a number of times; after half the restarts, we spend 10 seconds on solving the RBF on the  $2n$  most promising routes in an attempt to improve the initial solution.  $n$  is the number of customers. After the ALNS, the RBF is again solved on the  $4n$  most promising routes with a time limit of 240 seconds.

The branch and cut algorithm is based on the two-index vehicle flow formulation (2IDX) by Archetti et al. Two families of valid inequalities are included in the branch and cut algorithm: capacity cuts and connectivity cuts. Separation of capacity cuts is done using a greedy heuristic and using an exact method based on solving an MIP. The exact capacity cut separation is time consuming, so it is only used in the root node and when an integer solution to the 2-index model is discovered. Separation of connectivity cuts is done by solving maximum flow problems.

The 2IDX formulation is a relaxation of the SDVRP. The two-edge SD-CFRP formulation by Munari and Savelsbergh is solved to check SDVRP feasibility of 2IDX integer solutions. The SD-CFRP is solved on a reduced graph, where edge usage is limited to be exactly that of the 2IDX solution. This makes the SD-CFRP computationally tractable. In case of infeasibility, we add no-good cuts forbidding the current solution to the 2IDX model. In case of feasibility, we can easily derive the routes from the SD-CFRP solution because of its two-edge variables. Customer deliveries are assigned to routes by solving the RBF on the derived routes.

The proposed method is implemented in Julia using Gurobi 9.5. It is tested on four sets of benchmark instances from the 12th DIMACS Implementation Challenge on Vehicle Routing<sup>1</sup>.

The results reveal that all parts of the proposed method contribute to finding best solutions. Generally, the ALNS is very good at improving the initial solution. The branch-and-cut mainly comes into play when proving optimality. Given a runtime of 7200 seconds, the proposed method is able to solve five instances to proven optimality for the first time (to the best of our knowledge) when the vehicle fleet size is limited. In case of an unlimited fleet, the method solves six instances to proven optimality for the first time (to the best of our knowledge). The results are preliminary and could possibly be further improved by better parameter settings and algorithm tuning.

---

<sup>1</sup><http://dimacs.rutgers.edu/programs/challenge/vrp/>