Large Neighborhood Search and Structured Prediction for the Inventory Routing Problem

student: Louis Bouvier
advisors: Axel Parmentier and Guillaume Dalle

2021

Context of the internship The topic of my internship is the packaging return logistics of a large car manufacturer, representing dozens of millions of euros and hundreds of thousands of tons of CO_2 per year. This car manufacturer has a network of depots (factories) sending commodities (packaging) to customers (suppliers) by truck. This problem can be modelled as a multi-depot multi-commodity Inventory Routing Problem (IRP) [Coelho et al., 2013]. The main challenges we face are the size of the car manufacturer's network (with about 600 customers here compared with 50 maximum in the instance library of the literature [ORB]), the multi-depot aspect (leading to routing complexity), the multi-commodity aspect (with 30 concurrent commodities leading to binding constraints for the truck loading whereas most of the literature studies focus on the single-commodity framework with isolated cases of up to 5 commodities) and the large horizon (20 days compared with the $3-6$ days often considered in the previous studies). Our instances are thus one order of magnitude larger than the ones addressed in the IRP literature. We emphasize that no algorithm is known to properly scale to our context.

Benchmark algorithm We first implement a two-step heuristic inspired by the literature and use it as benchmark algorithm. An approximate flow formulation first decides who sends what to whom, when and which quantity. A local search based on the adaptation of neighborhoods of the literature then builds the truck routes and improves the solution of the first step.

Two new large neighborhoods If this benchmark algorithm performs well on instances from the literature, it reaches its limits on ours due to their large number of commodities and depots. We therefore introduce a Large Neighborhood Search (LNS) with two new neighborhoods. The first one solves a relaxation of the problem of the optimal reinsertion of a customer in an IRP solution. We formulate a MILP that relaxes the newly-created routes combinatorics. The second one solves a MILP that approximates the problem of the reinsertion of a commodity in the IRP solution, where the newly-created routes are restricted and their costs approximated. Both of them are solved in a few seconds on our large-scale instances. We combine them with 10 route neighborhoods to build a LNS.

Structured prediction For another use case, the car manufacturer needs to quickly find good solutions to the IRP. Our two-step greedy heuristic is a candidate for this requirement. Indeed, we observe that if we replace the flow solution by the one deduced from our LNS output, we almost reach the performance of our LNS with the greedy heuristic. We thus design a structured prediction algorithm to learn clever costs of the routing arcs. To make the parameter estimation problem differentiable, we use a Fenchel-Young loss [Blondel et al., 2020] and perturb the objective of the minimum cost flow problem with noise, as proposed by Berthet et al. [2020]. Our goal is to imitate the solutions provided by our LNS, while bypassing its heavy computations.

Results We extract real data from the car manufacturer's files, leading to 85 pre-processed instances at the European scale (see Table 1). Our LNS is able to significantly improve the solution of the two-step greedy heuristic within about one hour of computations. We reduce the gap by 30% on average over the 85 instances, compared with the greedy heuristic benchmark. Most of the gains are linked to the routing and customer shortage costs, and obtained with the two large reinsertion neighborhoods in our LNS. We emphasize no good lower bound is known for our IRP in the literature. Therefore, even though gaps enable us to compare our algorithms, their values are not sufficient to assess absolute performance.

Number of instances	85
Average number of depots	15
Average number of customers	602
Average number of commodities	30
Average horizon	21
Average cost after greedy-heuristic (benchmark)	2 421 050
Average cost after LNS	2 005 966
Average gap reduction after LNS	31%
Average LNS computation time [1]	73 minutes

Tableau 1: Results of the LNS compared with the greedy heuristic (benchmark).

We also get promising learning results. With a basic linear embedding, we manage to outperform the relaxation-based benchmark algorithm. Leveraging the expressiveness of graph neural networks or graph kernels, we hope to reach the performance of the LNS. We could also optimize the order and choice of the neighborhoods in the LNS, taking advantage of reinforcement learning techniques. Finally, computing stronger lower bounds is yet another challenge we intend to tackle during my PhD thesis.

Bibliography

OR-Brescia - Benchmark Instances. URL `https://or-brescia.unibs.it/instances`. Accessed on 2021-07-01.

Q. Berthet, M. Blondel, O. Teboul, M. Cuturi, J.-P. Vert, and F. Bach. Learning with Differentiable Perturbed Optimizers. *arXiv:2002.08676 [cs, math, stat]*, June 2020.

M. Blondel, A. F. T. Martins, and V. Niculae. Learning with Fenchel-Young Losses. *arXiv:1901.02324 [cs, stat]*, Mar. 2020.

L. C. Coelho, J.-F. Cordeau, and G. Laporte. Thirty Years of Inventory Routing. *Transportation Science*, 48(1):1–19, July 2013. ISSN 0041-1655. doi: 10.1287/trsc.2013.0472.

[1] the stopping criterion is a threshold on the cost gain after each LNS iteration