Using Machine Learning to Enhance Clarke and Wright Heuristic

Clément Legrand[1,2], Luca Accorsi[4], Diego Cattaruzza[3], Laetitia Jourdan[2], Marie-Eléonore Kessaci[2], and Daniele Vigo[4]

[1]Univ. Rennes, ENS Rennes, France
[2]Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France
[3]Univ. Lille, CNRS, Inria, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France
[4]Department of Electrical, Electronic and Information Engineering "G. Marconi," University of Bologna, 40136 Bologna, Italy

1 Context

The *Capacitated Vehicle Routing Problem* (CVRP) is a variant of routing problems widely studied in the literature, and despite its complexity (NP-complete), it is now possible to reach near-optimal solutions even for large instances. The purpose is to design routes of vehicles, starting and finishing at a same location, called *depot*, to serve all the n customers, while minimizing the total length of the fleet. Moreover the sum of the demands d_i of the customers on a route can not exceed the capacity Q of the vehicle.

We can find in the literature many algorithms and heuristics to solve the CVRP (see the book of Toth and Vigo [4]). Since optimal solutions are computationally expensive to find, most of the recent works focus on approximation algorithms. Moreover recent trends try to integrate learning mechanisms into metaheuristics to improve both the quality of the solutions obtained and the speed of the heuristic. In the case of the CVRP, Arnold in his thesis [1] trained a neural network to predict if a given solution was near-optimal or not. Thanks to the knowledge generated he was able to find relevant metrics to characterise the badness of an edge in a solution. Following this work, we propose to predict edges that appear in good solutions with a neural network and then we use these predictions to improve the well known Clarke & Wright [3] heuristic.

2 Learning Good Edges and Application

The choice of relevant features to characterise an edge is a key step to obtain a robust and efficient neural network. These features have to be *route*-independent. As we only consider symmetric instances in the following, we will only study edges (i, j) where $j > i$. According to the thesis of Arnold [1], for an edge (i, j) we simply consider the following features that appear the most relevant: the cost (euclidean distance between i and j), the demand of i (resp. j), the distance between i (resp. j) and its nearest neighbor, the demand of the nearest neighbor of i (resp. j), the distance between i (resp. j) and the depot, and finally the angle $\widehat{iv_0j}$ (v_0 is the depot). Features like distance between a customer and the depot or the angle, have been showed relevant [1] to distinguish non-optimal and near-optimal solutions. We also normalize these features to have a robust neural network applicable on different instances.

In routing problems the *Clarke and Wright* (CW) heuristic is a famous constructive heuristic. Although this heuristic does not generate very good solutions, several enhancements have been provided over the years. We present here a variant of CW that uses the trained neural network, called *SubLists*. First we use the basic saving formula to compute all the savings. Then, we sort the saving list and we divide it into sublists of size l, but we keep the order of the savings. In each sublist we associate to a saving, the probability of the corresponding edge (given by the neural network), and we sort the sublist by decreasing probability. Finally, we concatenate all the sublists by conserving their initial macro order. It gives the order in which we have to consider the savings. Note that with $l = 1$, we retrieve the *Basic saving* algorithm.

By experimenting the *SubLists* variant, we noticed that we cannot choose a value of l such that we obtain the best initial solution for all instances. To fix this problem, we consider the variant *Best SubLists*, in which we define a list L_l of value for l, and we keep the best solution among the $|L_l|$ solutions obtained.

3 Experimental Results

We use the Uchoa set of instances [5] to train the neural network. The Uchoa set is very challenging, mixes various instances, and is commonly used to test new algorithms for the CVRP. One half of the instances are randomly chosen for the training set while the remaining ones are for the validation set. For each instance of the set, we generate a set of solutions by perturbating the best-known solution, then we compute the weight of each solution, a real number between 0 and 1, which represents the quality of the solution (a weight near 1 represents a promising solution). The weight of each edge is then the normalized sum of the weights of the solutions in which it appears. Finally the features of each edge of the instance are computed, and the label *good* is assigned if the weight of the edge is over 0.5 contrary to the label *bad* if the weight is below 0.2. The edges with a weight between 0.2 and 0.5 are ignored. Since there are much more *bad* edges than *good* ones we have decided to downsample the *bad* edges with a probability $p \in \{0.005, 0.01, 0.02, 0.05\}$. This step was necessary to obtain interesting results. The learning phase is performed with the scikit-learn module of python and with a MLP classifier as neural network. Parameters are set following the recommendation of Chollet [2], to perform a binary classification. One can remark that a high ratio (i.e. a small value of p) decreases the accuracy, but statistics (recall and precision) of the two classes are more balanced.

Moreover, in order to fairly evaluate the performance, a hill climbing using the 2-opt operator was performed starting from each solution returned by CW or the proposed variants. With the variant *SubLists*, we obtain in average slightly better solutions than the *Basic* CW, and considering the preprocessing of the training of the neural network and of the predicted probability of each edge, we also get a similar time to the *Basic* CW. However the most promising results are given by the variant *Best SubLists*, which improves in average the solutions by almost 1.5%. It is also important to note that this latter variant needs a more important computational effort.

The results are promising, but the Clarke and Wright heuristic is still a naive heuristic and thus generally returns bad solutions for big instances. In future work, we will analyse the integration of the knowledge into more efficient metaheuristics (during a perturbation phase for example).

References

[1] Florian Arnold. *Efficient heuristics for routing and integrated logistics.* PhD thesis, University of Antwerp, 2018.

[2] Francois Chollet et al. *Deep learning with Python*, volume 361. Manning New York, 2018.

[3] Geoff Clarke and John W Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, 12(4):568–581, 1964.

[4] Paolo Toth and Daniele Vigo. *Vehicle routing: problems, methods, and applications.* SIAM, 2014.

[5] Eduardo Uchoa et al. New benchmark instances for the capacitated vehicle routing problem. *European Journal of Operational Research*, 257(3), 2017.