Machine Learning for Multi-Objective Problems

Clément Legrand[1], Diego Cattaruzza[1], Laetitia Jourdan[1], Marie-Eléonone Kessaci[1]

Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France
{clement.legrand4.etu, diego.cattaruzza, laetitia.jourdan,
marie-eleonore.kessaci}@univ-lille.fr

Keywords : *Operational Research, Knowledge Extraction, Routing Problems, Multi-Objective.*

1 Context

The *Vehicle Routing Problem with Time Windows* (VRPTW) is an extension of the well-known Vehicle Routing Problem, where time is considered as an important resource. The reader is referred to [3] for a broad overview of routing problems and their resolutions. In VRPTW, the customers are only available during specific time windows. The objective of this problem is to design routes of vehicles, starting at a same location, called *depot*, to serve n customers, while minimizing the total distance covered by the vehicles. Moreover the sum of the demands of the customers on a route can not exceed the capacity of the vehicle, and all time windows have to be respected. This problem is known to be NP-complete, and thus, it is hard to solve for large instances.

However, in real life problems, a decision maker may not be interested in optimizing only one objective but in several ones. Among the objectives studied in the literature [2] we find the total traveled distance and the total waiting time of drivers. Considering these two objectives turns the VRPTW into the *Multi-Objective* VRPTW (MO-VRPTW).

Multi-Objective Problems (MOP) can be solved using a Pareto approach where the objective is to find the best set of non dominated solutions, called Pareto-front. The *Multi-Objective Evolutionary Algorithm based on Decomposition* (MOEA/D) [4] is known to efficiently solve MOP. Over the last years, with the increasing success of machine learning, many studies integrate machine learning tools to solve combinatorial problems. If it starts to be quite common in the field of single-objective optimization, it remains uncommon to solve multi-objective combinatorial problems.

In particular the *Pattern Injection Local Search* (PILS) [1] has been designed to solve the well-known *Capcitated VRP* by using data mining techniques. In our work we have designed a new mechanism called MOPILS, based on PILS, and hybridized it with MOEA/D to solve MO-VRP.

2 The MOPILS Mechanism

Briefly, the PILS mechanism is built over two main concepts: *pattern extraction* (PE) and *pattern injection* (PI). Note that, in routing problems, a pattern is a sequence of consecutive customers inside a route. First, the PE mechanism identifies frequent patterns and stores them (incrementing the frequency if a pattern has already been found), so that, they can be used during the PI step. Then, during the PI process, a few frequent patterns are tentatively inserted in the current solution to define high-order local search moves.

The idea behind MOPILS is to generalize the two mechanisms of PILS, so that, they can handle multi-objective problems. Thus we propose MOPE, a *multi-objective pattern extraction*, and MOPI, a *multi-objective pattern injection*, which form together the MOPILS mechanism.

	Classic	*Hybrid*	$Hybrid^{RG}$
Rank (size 50)	3	**1.3**	1.7
Rank (size 100)	3	1.63	**1.37**

TAB. 1: Mean rank of each variant for each size of instance considered.

The idea behind the MOPE procedure is to regroup frequent patterns that are relevant for a given portion of the Pareto-Front. That is why we start by defining groups which contain close solutions on the Pareto-front. Hence each group refers to a set of patterns which can be used during MOPI. Patterns extracted from a solution s during MOPE are only stored in groups which contain the solution s. Note that several groups can contain the same solution. Then, patterns stored in a group will improve solutions of this group only. On the other hand MOPI aims to diversify a set of solutions by using PI with different sets of patterns. Each set of patterns used for injection is a subset of a group defined in MOPE.

3 Hybridization with MOEA/D and Experimental Results

MOEA/D approximates the Pareto-front by decomposing the problem into a number of single objective problems (where the fitness is an aggregation of all the objectives). Here we consider a weighted sum of the two objectives. Thus each subproblem is characterized by a unique weight vector $w = (w_1, w_2)$ with $w_i \in [0, 1]$. Hence given a subproblem i and its weight vector w^i it is possible to compute the *neighborhood* (e.g. the k nearest neighbors) of i, by computing the euclidean distance between w^i and w^j for all subproblems $j \neq i$. For MOPE, we associate a group to each subproblem, so that, we have as many groups as subproblems. To define the i-th group, we propose to consider the neighborhood of the subproblem i to extract patterns relatively close in the objective space. Finally, our hybridization works as follows: first an initialisation phase is performed to generate an initial population and define the groups for MOPE. Then for each subproblem i two solutions are randomly selected in the neighborhood of i and a crossover is performed. MOPI is performed just after the crossover, then a local search is performed to reach a local optimum, and finally MOPE is applied. A mutation can also occur after the crossover.

For our tests we use the Solomon's instances of size 50 and 100. For each size three types of generation are available (R, C and RC). We compare three variants of MOEA/D: the classic MOEA/D (*Classic*), the hybridization (*Hybrid*), and a variant of our hybridization where the group of patterns used in MOPI is selected randomly. Both *Classic* and *Hybrid* variants are tuned with iRace. Then for each instance we apply each algorithm 20 times, and we compare the mean of the hypervolumes obtained with a common reference solution (we recall that the higher the hypervolume, the better the pareto-front). Table 1 compares the mean ranks of the variants proposed. Experiments show that both hybrid algorithms give better performance than the classic MOEA/D. Moreover, the patterns have to be extracted using neighboring solutions.

References

[1] F. Arnold, Í. Santana, K. Sörensen, and T. Vidal. Pils: Exploring high-order neighborhoods by pattern mining and injection. *arXiv preprint arXiv:1912.11462*, 2019.

[2] N. Jozefowiez, F. Semet, and E. Talbi. Multi-objective vehicle routing problems. *European journal of operational research*, 2008.

[3] P. Toth and D. Vigo. *Vehicle routing: problems, methods, and applications*. SIAM, 2014.

[4] Q. Zhang and H. Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 11(6):712–731, 2007.