A multi-policy hyper heuristic for multiobjective optimization

Michele Urbani [1] Francesco Pilati [1]

[1]Department of Industrial Engineering, University of Trento (Italy)

Motivations

- Selection hyper-heuristics (SSHH) are search strategies that can be successfully applied to multi-objective optimization problems [2].
- Learning a single selection rule in a SSHH might be limiting because the information from the multiple objectives might be unexploited [4].
- Algorithms based on sequences of problem-specific low-level heuristics can efficiently solve complex combinatorial problem [3].

Research goals

1. A novel approach named the **multi-policy** approach is proposed to further enhance the searching ability of sequence-based selection hyper-heuristics.
2. The multi-policy approach performs **online** learning of the select policies. One selection policy per objective is learned using objective-wise information.
3. The proposed algorithm is tested on a **real-world** instance of the vehicle routing problem with pickup and delivery (VRPPD).

Methodology

A **low-level heuristic** (LLH) is a rule that modifies the decision variables $\mathbf{z}$ of the problem under analysis. A set H of m LLHs is assumed to be available to modify the solutions $\mathbf{z} \in Pop$.

A **selection policy** is an ensemble of Markov decision models that alternates the selection of LLHs h and sequence-termination signals AS (see box 1 in Fig. 1). There are **as many selection policies as the number N of objectives**.

At each iteration, a selection policy produces a sequence of low-level heuristics SEQ (see box 2 in Fig. 1) to be applied to a solution $\mathbf{z}$.

For each new solution $\mathbf{z}' \in Pop'$ that was improved by a sequence of heuristics SEQ, the **reward rule** (see box 3 in Fig. 1) attributes a score to all the couples of subsequent heuristics in SEQ.

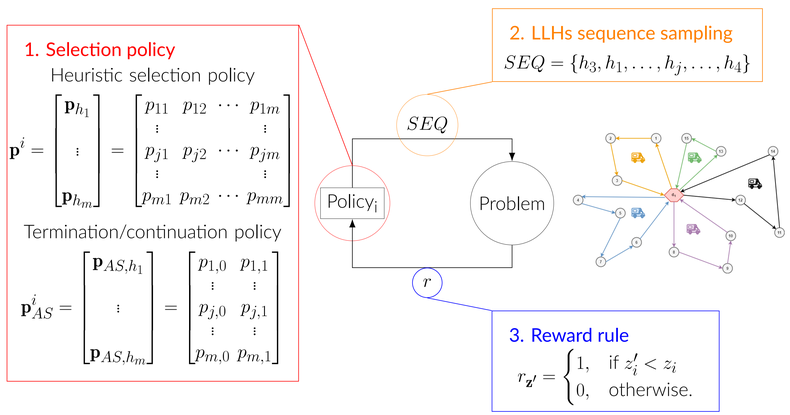

Figure 1. The architecture of the learning system.

Algorithm architecture

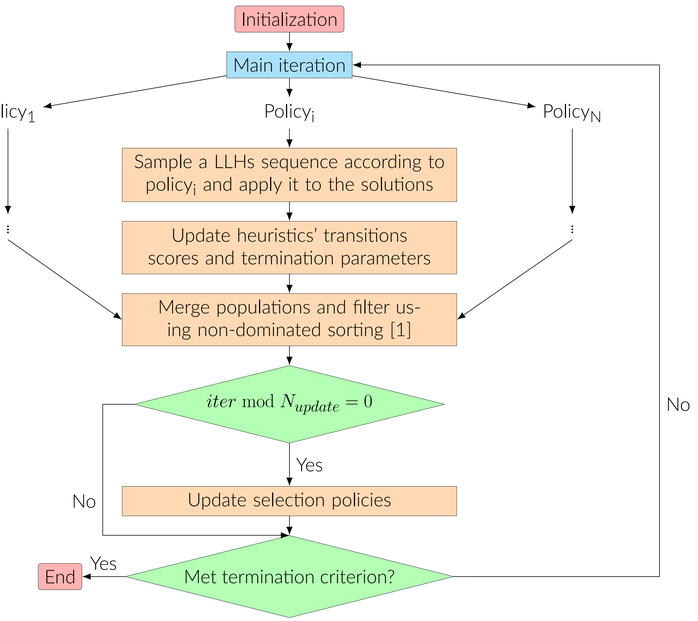

Figure 2. The architecture of the hyper heuristic algorithm.

Experimental setting

Numerical experiments were carried out to solve a three-objective version of the vehicle routing problem with pickup and delivery (VRPPD).

The multi-policy algorithm was tested on a single instance of the VRPPD with **60 deliveries** and **4 pickup points**.

Data are inspired to a **real-world case study**: a geography with non-Euclidean distances was used, and goods to be delivered presented different weights and volumes.

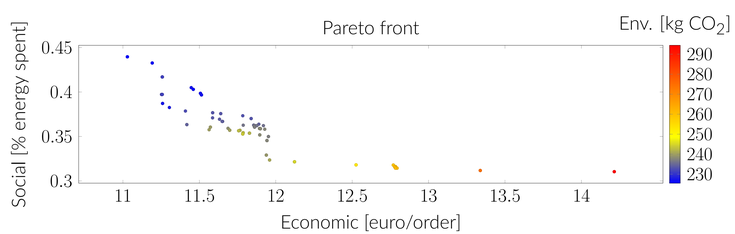

Figure 3. An example of Pareto set of solutions found by the multi-policy hyper heuristic.

Selected results

- Selection policies evolve over time: for example, comparing $\mathbf{p}^{(eco)}$ at iteration 25 and 50 in Table 1, the selection probability distribution p_{h_4} changes significantly.
- Selection policies evolve differently from each other: for instance, $\mathbf{p}^{(eco)}$ and $\mathbf{p}^{(env)}$ at iteration 50 in Table 1 present significantly different values for p_{h_0} and p_{h_1}.

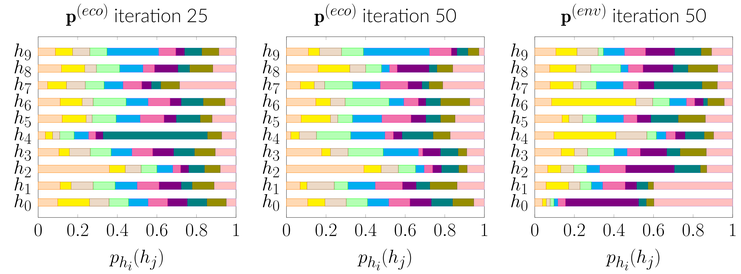

Table 1. A graphical representation of different heuristic selection policies at different stages of the algorithm execution.

- The learned policies are sensitive to: (1) the **reward value** r that is used, (2) the number of iterations between **policy updates** N_{update}, and (3) the weight that is used when updating the probabilities in the Markov decision model.

Conclusions and future research

Learning selection policies is not straightforward. Extensive testing of the learning rule is required over a large and diverse set of instances of the problem studied.

Online learning in multiobjective combinatorial problems is **time expensive**. Either a high-performance implementation of the algorithm exists, or offline learning should be considered.

Comparative analysis of the proposed multi-policy hyper heuristic is required. Algorithms such as multiobjective local search (MOLS) and choice function hyper heuristic (CFHH) will be considered for testing over **large** (e.g., up to 500 nodes) and **diverse** (e.g. geography, load, time windows) **problem instances**.

Sampling of LLH sequences yield widely variable execution times when the termination criterion is the number of iterations, which might be undesirable in the production phase.

References

[1] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[2] J. H. Drake, A. Kheiri, E. Özcan, and E. K. Burke. Recent advances in selection hyper-heuristics. *European Journal of Operational Research*, 285(2):405–428, 2020.

[3] A. Kheiri. Heuristic sequence selection for inventory routing problem. *Transportation Science*, 54(2):302–312, 2020.

[4] F. Tricoire. Multi-directional local search. *Computers & Operations Research*, 39(12):3089–3101, 2012.