

# An Evolutionary Programming Hyper-heuristic with Co-evolution for CHeSC'11

David Meignan

Department of Mathematical and Industrial Engineering  
Ecole Polytechnique de Montréal, Montréal, Canada  
[david.meignan@polymtl.ca](mailto:david.meignan@polymtl.ca), <http://lalea.fr/>

**Abstract.** We present an Evolutionary Programming Hyper-heuristic (EPH) implemented for the Cross-Domain Heuristic Search Challenge 2011. The proposed method combines an evolutionary programming approach and co-evolution. The solving process of EPH consists in evolving a population of solutions by applying heuristics sequences. The heuristics sequences are also in a population and evolve according to their performances.

## 1 Method principles

EPH (Evolutionary Programming Hyper-heuristic) is a hyper-heuristic of selection with online learning mechanisms [2], implemented using HyFlex framework [1]. EPH is based on evolutionary programming approach and co-evolution. Two populations evolve during the solving process; a population of solutions, and a population of heuristics sequences. The set of heuristics sequences corresponds to the search strategy which evolves through mutations and selections. The population of solutions and the population of heuristics sequences co-evolve in the sense that evolution of solutions is done by the application of heuristics sequences, and evolution of heuristics sequences is related to their efficiency on current solutions.

### 1.1 Solutions population

The structure and process related to the population of solutions is quite simple. The population is composed of a set of  $N$  solutions. These solutions are used to evaluate heuristic sequences. This evaluation of heuristics sequences is the main process to produce new solutions. To evaluate a heuristics sequence, one or two solutions are chosen in the population. Then, the sequence of heuristics is applied on these solutions. Finally, the resulting solution is inserted in the population if it meets acceptance criteria.

The initial set of solutions is generated using the default heuristic of generation. The choice of the solutions for evaluating heuristic sequences is “random” (following the order of indexes in the population of solutions). After the application of a heuristics sequence, the resulting solution is inserted in the population if its cost is better than at least one cost in the population of solutions, and

its cost is different to all costs in the population of solutions. When a solution is inserted in the population it replaces the worst solution and the size of the population remain the same.

## 1.2 Population of heuristics sequences

The set of heuristics sequences corresponds to the search strategy that evolves during the solving process. The population is composed of a set of  $S$  heuristics sequences based on the same pattern. A sequence consists of a set of perturbation heuristics and a set of local search heuristics with an intensity value associated to each perturbation and local search heuristics. These two phases are similar to the “intensification/diversification” scheme in coalition-based metaheuristic [4]. The perturbation part contains one or two heuristics of perturbation: mutation, crossover or ruin-recreate. If this part contains two heuristics, they must be of different types, and crossover must be in first position. The local search phase contains all local search heuristics available for the problem. Application of a sequence starts by running once the perturbation heuristics with their respective intensity parameters. When a heuristic is applied, the resulting solution serves as initial solution for the next heuristic in the sequence. Then, local search heuristics are either applied once, or applied using a Variable Neighborhood Descent (VND) strategy [3]. Both local search strategies use the intensity parameters and the order of local search heuristics specified by the sequence. The choice between VND and unique application of local search heuristics is determined in a initialization phase of the hyper-heuristic.

The initial population of heuristics sequences is randomly generated. This population evolves until the end of the solving process by mutation and selection. To obtain a new generation, a mutation is applied on each heuristics sequences and doubles the size of the population. Then, the heuristics sequences are selected using tournament selection that evaluates heuristics sequences by applying it. The new generation consists of the heuristics sequences that win the tournaments.

Four mutation operators are defined for heuristics sequences. They have the same probability of being applied. The first mutation consists in modifying all values of intensity parameter for the perturbation part of the sequence. The second mutation randomly adds, replaces or removes a perturbation heuristic while limiting the size of the perturbation part to one or two heuristics. The third mutation modifies all values of intensity parameter for the local search phase of the sequence. The last mutation randomly changes the order of local search heuristics.

The selection of heuristics sequences is performed on the population of parent sequences added to the mutated sequences. Each tournament confronts two randomly chosen sequences. The competing sequences are removed from the initial population and the winning sequences form the new generation. The winner of a tournament is the sequence that scores most on  $r$  rounds. A round consists in applying the two sequences on a same initial solution (and a second solution for crossover). The criteria for winning a round are; first the insertion of the

resulting solutions in the population of solutions; then the value of the resulting solutions. If sequences are tied after the rounds, the oldest heuristics sequence is selected and added to the new generation.

## 2 Parameter setting

The parameters of EPH are the following:

- $N$ , the size of the population of solutions,
- $S$ , the size of the population of heuristics sequences,
- $r$ , the number of rounds for the selection of sequences,
- the type of strategy for applying local search heuristics (VND—single application).

The parameter  $r$  is fixed to 3 regardless to the solved instance. The size  $S$  of the population of sequences is set to the value of the number of heuristics available for the problem. The size of the population of solutions and the type of strategy for local search is determined from a “profile” of the problem. The profile is a set of values that characterize the heuristics operation on the solved instance. The objective of this profile is not to “recognise” the instance but to analyze the behavior of heuristics on the instance.

The profile values are computed in a preliminary phase of the hyper-heuristic. This profiling phase records the average time of each heuristics on five runs, and analyses five VND local searches. These sample VND local searches and the average times allow determining the following values:

- Estimated number of heuristic applications in the time limit,
- average number of solution improvements during VND,
- estimated number of VND applications in the time limit,
- ratio of VND that failed in producing a better solution than a randomly generated solution.

The size of the population and the type of local search is determined from these parameters. If the estimated number of heuristic applications is very low, a small size of solutions population is set and local search strategy based on single application of heuristics is used. If the VND failure ratio is low, a small population of solutions and the VND local search strategy are preferred. A high value for the average number of solution improvements and a high value of estimated number of VND applications in the time limit favor a large population of solutions.

## 3 Tricks for the competition

In addition to the global solving process described above, the following elements have been added to improve the results of EPH:

- From 75% of elapsed time, the best solution of the population is regularly intensified with local search heuristics.
- The VND allows several attempts for improving a solution using a local search heuristic. The number of attempts is increased if it conducts to improvements. This mechanism is disabled when it does not conduct to any additional solution improvement after the first sequences generation.
- Solutions generated during the profiling phase are used in the initial population of solutions.

## References

1. Burke, E.K., Curtois, T., Hyde, M., Kendall, G., Ochoa, G., Petrovic, S., Vazquez-Rodriguez, J.A., Gendreau, M.: Iterated local search vs. hyper-heuristics: Towards general-purpose search algorithms. In: *IEEE World Congress on Computational Intelligence (CEC 2010)*. pp. 3073–3080 (2010)
2. Burke, E.K., Hyde, M., Kendall, G., Ochoa, G., Ozcan, E., Woodward, J.: *Handbook of Metaheuristics*, chap. A Classification of Hyper-heuristics Approaches, pp. 449–468. *International Series in Operations Research & Management Science*, Springer (2010)
3. Hansen, P., Mladenović, N.: *Handbook of Metaheuristics*, chap. Variable Neighborhood Search, pp. 145–184. Kluwer Academic (2003)
4. Meignan, D., Créput, J.C., Koukam, A.: Coalition-based metaheuristic: a self-adaptive metaheuristic using reinforcement learning and mimetism. *Journal of Heuristics* 16, 859–879 (2010)