A Hyper-heuristic for the CHeSC 2011

Mathieu Larose

Department of Computer Science and Operations Research
Université de Montréal, Canada

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

larosma@iro.umontreal.ca

Abstract. This extended abstract describes a hyper-heuristic called *ML* for the Cross-domain Heuristic Search Challenge (CHeSC 2011).

1 Introduction

ML is based on the self-adaptive meta-heuristic of Meignan et al. [1] called *CBM*. CBM uses multiple agents that cooperate throughout the optimization process. The search behavior of an agent is adapted during the optimization process by reinforcement learning and mimetism learning.

ML consists of a single agent that use reinforcement learning in a similar way as CBM does, but with a few modifications. Thus, *ML* will be described in terms of *CBM*[1].

ML iterates through Diversification-Intensification cycle (D-I cycle). A D-I cycle can be described as follows:

1. Apply a diversification heuristic (Diversification)
2. Apply several intensification heuristics until the current solution cannot be improved (Intensification)
3. Accept or discard the new solution (Move Acceptance)

The selection of heuristics and reinforcement learning is similar as Meignan et al. [1]. (We refer the reader to this paper.) For this implementation, the weight values in the matrix are initialized with parameter $\alpha = 1$ and the learning factor is $\sigma = 1$.

[1] *ML* can also be viewed as a modified *iterated local search* (ILS) with reinforcement learning.

2 *ML*

2.1 Stages

Diversification The diversification heuristics are the mutation and ruin-recreate heuristics as well as a no-op heuristic, that is a heuristic that does not modify the solution. Thus, the diversification phase will sometime not perturb the current solution. This has the effect that the intensification heuristics have a "second chance" to improve the current solution.

Intensification The intensification heuristics are the local search heuristics.

Move acceptance Accept the new solution if it improves the current solution or if the current solution has not been improved for the last `max_iter` = 120 iterations. Otherwise, discard the new solution.

References

1. D. Meignan, A. Koukam and J.-C. Creput. *Coalition-based metaheuristic: A self-adaptive metaheuristic using reinforcement learning and mimetism*, Journal of Heuristics, vol. 16 pp. 859-879, 2010.