A new hyper-heuristic implementation in HyFlex: a study on generality

Mustafa Misir [ab] Katja Verbeeck [ab] Patrick De Causmaecker [b]

Greet Vanden Berghe [ab]

[a] *CODeS, KAHO Sint-Lieven, Gebroeders Desmetstraat 1, 9000 Gent, Belgium*
[b] *CODeS, K.U.Leuven Campus Kortrijk, Etienne Sabbelaan 53, 8500 Kortrijk, Belgium*

Selection hyper-heuristics concentrate on using the strength of multiple low-level search mechanisms for solving instances from various problem domains [2, 3, 4]. A traditional selection hyper-heuristic is composed of 1) a heuristic selection mechanism for choosing heuristics at each decision step and 2) a move acceptance strategy for deciding about whether or not to use the explored solutions by the selected heuristics. These mechanisms work together in a problem-independent manner to raise the level of generality on the one hand and to ease their applicability on the other hand. The present study provides a new selection hyper-heuristic equipped with various adaptive features [5].

The first adaptive component, an adaptive dynamic heuristic set (ADHS) strategy, determines heuristic subsets with respect to the varying performance of the heuristics. The motivation behind this approach is to decrease the effect of the heuristic selection process by increasing the quality level of the heuristic set. A group of basic features are determined for measuring the performance of the heuristics. For each performance check, the heuristics' behaviour are monitored during a number of iterations referring to a phase. The performance outcomes are used to exclude worse performing heuristics. The second adaptive component chooses heuristics from the heuristic subsets. A learning automaton is used to indicate the probability of each heuristic being selected. Another adaptive feature which is added determines which heuristics should be applied in pairs to yield new best solutions. During a run, a heuristic residing in the heuristic subset is occasionally applied as a second heuristic to explore effective heuristic pairs. If a new best solution is found after applying the second heuristic, it is added to the heuristic list belonging to the first heuristic. For deciding to choose a single heuristic or heuristics in pairs, a linear probability schedule is deployed. Typically, schedules prefer applying heuristics individually during the early iterations of a phase. Pairwise heuristics are chosen in later iterations.

The acceptance duty is handled by a threshold accepting mechanism, namely adaptive iteration-limited list-based threshold accepting (AILLA). This method maintains a list of values regarding the quality of the previously found best solutions. This list is updated whenever a new best solution is found. These list values are utilised as threshold values for accepting worsening solutions if the search process gets stuck. The threshold level is increased over time by using a higher value from the list if a new best solution could not be found. In addition to that, the acceptance mechanism waits for a number of iterations before accepting a worsening solution. This iteration value simply reflects the required number of steps for discovering a new best solution. This value is also updated with the condition that a new best solution is found.

Our hyper-heuristic is applied to 40 instances from four different problem domains, namely Bin Packing, Max SAT, Flowshop Scheduling and Personnel Scheduling. These problem instances together with corresponding heuristic sets are provided in HyFlex, i.e. a hyper-heuristic software framework [1]. The experimental results show that the proposed hyper-heuristic performs superior across the given problem domains compared to the default hyper-heuristics announced at the CHeSC website[1].

[1] *http://www.asap.cs.nott.ac.uk/chesc2011/*

References

[1] E.K. Burke, T. Curtois, M. Hyde, G. Kendall, G. Ochoa, S. Petrovic, and J.A. Vazquez-Rodriguez. HyFlex: A flexible framework for the design and analysis of hyper-heuristics. In *Proceedings of the 4th Multidisciplinary International Scheduling Conference: Theory & Applications (MISTA'09)*, pages 790–797, Dublin, Ireland, August 10–12 2009.

[2] E.K. Burke, E. Hart, G. Kendall, J. Newall, P. Ross, and S. Schulenburg. *Handbook of Meta-Heuristics*, chapter Hyper-Heuristics: An Emerging Direction in Modern Search Technology, pages 457–474. Kluwer Academic Publishers, 2003.

[3] E.K. Burke, M. Hyde, G. Kendall, G. Ochoa, E. Ozcan, and R. Qu. A survey of hyper-heuristics. CS Technical Report No: NOTTCS-TR-SUB-0906241418-2747, Univeristy of Nottingham, 2009.

[4] E.K. Burke, M. Hyde, G. Kendall, G. Ochoa, E. Ozcan, and J.R. Woodward. A classification of hyper-heuristic approaches. *Handbook of Metaheuristics*, pages 449–468, 2010.

[5] M. Misir, K. Verbeeck, P. De Causmaecker, and G. Vanden Berghe. A new hyper-heuristic implementation in hyflex: a study on generality. In *the 5th Multidisciplinary International Scheduling Conference: Theory & Applications (MISTA'11)*, pages 374–393, Phoenix/Arizona, USA, August 10–12 2011.