ABSTRACT

The main purpose of using hyper-heuristics for optimisation problems is to develop flexible efficient approaches that may be applied with minimal tailoring within and between different problem domains to produce solutions of reasonable quality with acceptable demands on resources. Compared to simple heuristic and meta-heuristic methods hyper-heuristics are a more recent development and there is a need for continuing exploration and development of their potential.

Hyper-heuristic approaches operate within a general framework where the hyper-heuristic controller has access to a set of low-level heuristics and selects one of these to use at the next stage of the search based on its performance in previous stages of the search. The hyper-heuristic controller and the set of low-level heuristics are detached from the problem domain where solutions are generated and a cost (objective) function is evaluated for the solution obtained using the selected low-level heuristic. Information about the change in the value of the cost function and the time taken to generate a solution are stored and communicated to the hyper-heuristic controller which uses this accumulated information to evaluate the past performance of the low-level heuristic. The hyper-heuristic controller typically uses a choice function for this evaluation and the low-level heuristics are normally swap based heuristics that exchange elements of the current solution in order to produce a new solution. The hyper-heuristic controller, the set of low-level heuristics and the choice function are independent of specific knowledge about the problem and the expectation is that this detachment will enable the approach to be used within and across different problem domains with minimal tailoring. In this study the general framework for hyper-heuristics is investigated and three modifications are developed and tested using experiments and comparisons with published results.

The first modification introduces a self learning mechanism into the choice function. This allows the hyper-heuristic controller to automatically modify the values of parameters in the choice function as the search progresses in order to intensify or diversify appropriately the selection of low-level heuristics. Experimental evidence shows that this modification improves the performance of the choice function.

The second modification enables the hyper-heuristic controller to dynamically configure low-level heuristics and it is developed in two stages. First a procedure is developed for configuring low-level heuristics at each stage of the search and this addresses the limitation of the hyper-heuristic controller only having access to a fixed set of pre-defined low-level heuristics. However, the configuration process produces many low-level heuristics and so in the second stage in order to improve the effectiveness of the hyper-heuristic controller it is redesigned as a hierarchy of sub hyper-heuristic controllers working together each employing a choice function to select suitable configuration options and combinations of configuration options. Experiments with this new dynamic approach produce promising results when it is compared with the non-dynamic approach and other algorithms specifically designed for the experimental problems.

The third modification investigates centralised and peer-to-peer schemes for distributing the search using multiple processors. The objective is to allow the hyper-heuristic controller to be able to work on more than one set of low-level heuristics simultaneously. The dynamic and non-dynamic approaches are compared using both schemes and there is evidence of definite benefits from distributing the search.

These three modifications and the extensive experimental evidence presented in this study represent important contributions to growth in the understanding of the operation of hyper-heuristic approaches and open the way for subsequent studies.