Plug-and-Play hyper-heuristics: an extended formulation

Patricia Ryser-Welch
Departments of Electronics
University of York
York
Email: patricia.ryser-welch@york.ac.uk

Julian F. Miller
Departments of Electronics
University of York
York
Email: julian.miller@york.ac.uk

I. Introdution

Hyper-heuristics is a very active field that is developing all the time. Defined as *"heuristics to choose heuristics"* [1], this area of bio-inspired intelligent systems covers a wide range of algorithms selection techniques. Most Hyper-heuristic frameworks focus on improving the quality of the solutions of the problems,but they retain very little knowledge of the algorithms behaviour. The development of visual diagnosis tools for hyper-heuristic systems aims at addressing this open issue. Many sophisticated state-of-the-art algorithms exist, yet currently it is unknown whether hyper-heuristic frameworks are capable of expressing such algorithms. [2], [3], [4]

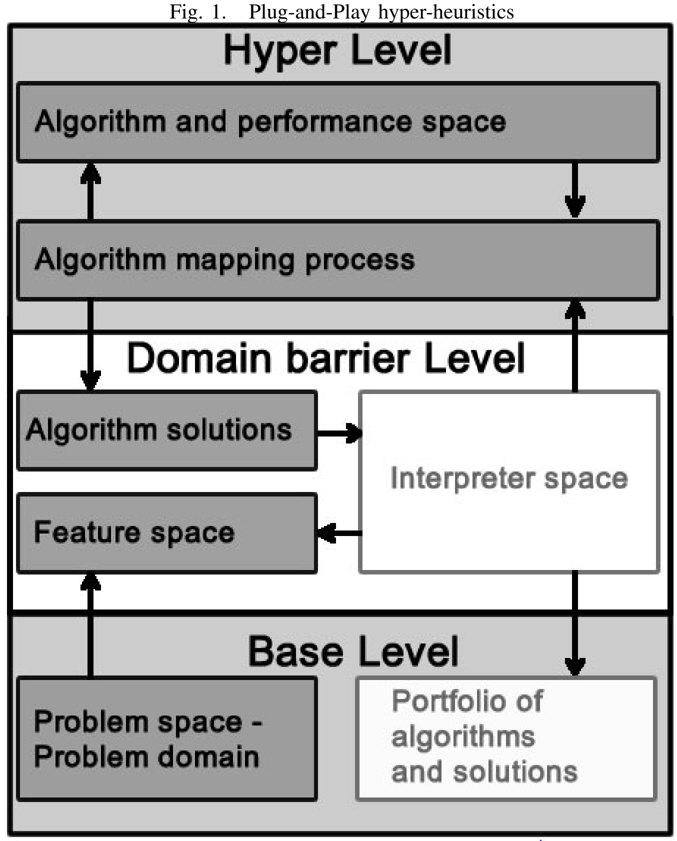
Our motivation is to improve not only the design of algorithms with hyper-heuristics, but also understanding better their behaviour. We are proposing to extend further the problem-domain independence of the current models, with the use of meta-statements. Our formulation applies the instruction mnemonics with the purpose of redefining hyper-heuristics as a method that uses heuristics to optimise heuristics. The search of an optimum algorithm should use effectively local interactions between sub-systems to spontaneously find the best orders of instructions without any direct intervention from any human-activities.

II. Proposed extended formulation

Hyper-heuristics seek to optimise algorithms to solve a class of problems. In the literature, two main models have been been adopted. First, *The "Algorithm Selection Problem"* describes explicitly an iterative learning mechanism taking place during an algorithms optimisation process. Second, *the ‘’ Two-level model"* is an easy-to-implement modular architecture, that separates the search method from a specific problem domain.

We believe these two models complement each other. Consequently, we are suggesting to merge them together to form a more explicit architecture. As a result, *Plug-and-Play hyper-heuristics* has included all the components of the Algorithm Selection Problem in three modules (in dark-grey in image 1). The architecture of the *two-level model* has been kept; the algorithm optimisation remains the focus of ‘’Hyper level" module and a specific problem domain continues to be provided by the ‘’Base level" module (see light-gray elements of image 1).

Fig. 1. Plug-and-Play hyper-heuristics



Our aim is to give greater functionality to the domain barrier; the interface between the problem domain and the algorithm search than is present in the current *Two-level* model. A level of indirection extends this interface into a new module, with its own full set of components. In our proposed extension, the algorithm search manipulates instructions independently of the problems solved. As a result, the algorithm optimisation mechanism is now fully separated from the problem domain. Also the problem domain is not aware of the process optimising the methods for finding solutions in the problem space.

Programmers use high-level programming languages to produce algorithms for a specific purpose. In a similar manner, hyper-heuristics automatically generate algorithms in the form of heuristics. The *Base Level* can then be compared to the hardware environment and its machine code, and the *Hyper Level* to the programmer coding programs. The new *Interpreter Space* has been added to the Domain Barrier Level, so that the algorithm solutions can be mapped explicitly to the operations of the problem-domain at the Base Level.

Another aspects of the new sub-system is the definition of instances of the chosen problem; this splits the problem space into smaller subsets that are defined by some common characteristics. The set of all solutions can be split into smaller subsets corresponding to the set of features using the function. For example, a subset S_n is a solution space defined by its unique features and the optimised algorithms searches only the solutions space S_n. Similarly, a subset S_m is a solution space defined by its unique features and the optimised algorithms searches only the solutions space S_m.

III. Implications and benefits

The new level of indirection brings more transparency and keeps the problem domain and the algorithm optimisation distinct from one another. The complexity added by the new level of indirection can be viewed as a criticism. Nonetheless, this more explicit formulation should make it easier to understand the hyper-heuristic frameworks' sub-systems with their components. Approaches based on perturbative low-level heuristics can continue to be applied over a wider range of problem domains. Each component of the Plug-and-Play hyper-heuristic formulation can continue implementing well-known combinatorial problem domains using their benchmarks and a greater range of heuristics. Equally real-life problems could be modelled with its own Problem, Features and Interpreter space, so that the hyper-heuristic search for suitable algorithms and problems solutions can be carried out independently from the problem domain. For a specific problem domain, the algorithm mapping process can be substituted with another one, without the need to write again the Domain Barrier and the Base levels.

In the previous section, the two sets of operations and encoding schemes contrasted completely with each other, and yet CGP should be able search for an optimum algorithm solutions without having any specific knowledge of the problem domain. It would be feasible to substitute CGP with another search method more easily.

Since the new Interpreter space is unique to each problem domain, each element needs to be customised to the operations defined in the problem space and these can be mapped to the instructions set of the hyper-level. The fitness evaluation of algorithms is now incorporated in the problem-specific Interpreter space. This problem-dependent function can still feedback important measure of fitness to the algorithm optimisation process, and should continue playing an important role in this process. Various algorithms' performance evaluations take into consideration different criteria. Some of them include the performance changes during the search and the execution times of the algorithms. Others have adopted the mean computed from the fitness evaluations of the problem solutions, obtained from executing the algorithms several times.

The ability to analyse the algorithms and the problems solutions has been extended by the portfolio. We hope a more formal understanding of the algorithms' behaviour can be developed and patterns of algorithms discovered. The portfolio is the first step in the direction of providing optimised algorithms, so that they can be fully compared against of the state-of-the-art at a later stage. One possible outcome would be to find out whether hyper-heuristics is capable of expressing such algorithms.

IV. Future work and conclusion

We have described a novel extended formulation that approaches hyper-heuristics with a three-level model. We called this formulation *Plug-and-play hyper-heuristics* to illustrate that algorithms should be optimised without the programmers intervention in resolving problem-dependency problems.

Components within each sub-system are expressed more explicitly. The Two-level model and the Algorithm Selection Problem have been merged together. At the Genetic and Evolutionary Computation Conference (GECCO) there has been an annual workshop on human-competitive results. It is hoped that *Plug-and-play hyper-heuristics* will in the future be able to contribute in improving state-of-the-art algorithms in various domains.

Future work will include investigating the ability of *Plug-and-play hyper-heuristics* to produce general algorithms that can be applied to a large classes of problems. In some fields of Artificial Intelligence, such as Machine Learning, there are extensive and widely used benchmark problems that originate in real-world problem domains. *Plug-and-play hyper-heuristics* could be used to find new and effective types of classification algorithms, this could assist in future automated diagnosis systems. In addition there are many logistics problems in the healthcare industry to which optimised algorithms may be extremely useful.

References

[1] P. Cowling, G. Kendall, and E. Soubeiga, "A hyperheuristic approach to scheduling a sales summit," in *Practice and Theory of Automated Timetabling III*. Springer, 2001, pp. 176–190.

[2] L. Cruz-Reyes, C. Gómez-Santillán, N. Castillo-García, M. Quiroz, A. Ochoa, and P. Hernández-Hernández, "A visualization tool for heuristic algorithms analysis," in *7th International Conference on Knowledge Management in Organizations: Service and Cloud Computing*. Springer, 2013, pp. 515–524.

[3] E. K. Burke, M. Gendreau, M. Hyde, G. Kendall, G. Ochoa, E. Özcan, and R. Qu, "Hyper-heuristics: A survey of the state of the art," *Journal of the Operational Research Society*, vol. 64, no. 12, pp. 1695–1724, 2013.

[4] P. Ryser-Welch and J. F. Miller, "A review of hyper-heuristic frameworks," 2014.