Solution paradigms that involve single or multiple heuristics and single or multiple problem representations

Abdellah Salhi [a], Eric S. Fraga [b], and El-Ghazali Talbi [c]
a. Department of Mathematical Sciences, University of Essex, Colchester CO4 3SQ, as@essex.ac.uk
b. Centre for Process Systems Engineering, Department of Chemical Engineering, UCL, Torrington Place, London WC1E 7JE
c. University of Lille-INRIA-CNRS, Lille, France

It is well known that some algorithms perform better than others on a given problem. The performance of algorithms may even be affected by the instance of the problem solved. For instance, an unstable algorithm which otherwise is suitable for some problem, may perform poorly on an instance that involves ill-conditioned data. What is less known is that different mathematical problem *formulations* of the same problem affect differently the solution process, too. Hall and McKinnon (2004) present some examples for the Simplex method. It is, therefore, important to choose not only the most suitable algorithm for a given problem, but one that is most suitable for a given formulation of the problem and the particular instance being solved. The issue of different instances can be dealt with via tailoring, as illustrated recently for the flexible flow-shop problem by Salhi and Vazquez-Rodriguez (2014). In the presence of alternate formulations of the same problem and a variety of possible algorithms, the match making problem is as hard as the original optimization problem. Nevertheless, it must be addressed and the solution of it is beneficial.

As well as algorithms and formulations, there is also the issue of *representation*: how the solutions to a problem are expressed on the computer for a particular algorithm. For instance, in genetic algorithm, a particular optimisation variable x may be represented using a real-valued allele or by a binary representation which discretises the domain of x more coarsely. This representation may have a significant impact on the efficacy and the efficiency of the algorithm for the specific problem as formulated. Often, the difference in computational performance between two representations for a genetic algorithm, for instance, will depend on the closeness of match between the space defined by the representation and that defined by the formulation. Also, for the particular combination of an algorithm and a representation, there may be associated a number of operations used by the algorithm that manipulate representations. A given representation will encourage certain mathematical operations but not others.

Let us, for a moment, insist that an algorithm is a process that guarantees the optimum solution when it exists, unlike a heuristic. In the same way, a formulation defines a solution space whereas a representation, together with any operators used to manipulate those representations, may or may not cover the same solution space. Let n_a be the number of algorithms and n_f be the number of formulations. When two or more exact algorithms and/or two or more closed form formulations of a given problem are available, the Multiple Algorithms, Multiple Formulations (MAMF) paradigm implemented on a distributed platform will simultaneously solve the original problem and the *match making* problem of identifying the best formulation and the best algorithm. The MAMF reduces to the Single Algorithm, Multiple Formulations (SAMF) problem if $n_a = 1$ and to the Multiple Algorithms and Single Formulation (MASF) problem when $n_f = 1$. If both $n_a = 1$ and $n_f = 1$, we end up with the approach that we currently use, one which does not involve any match making: SASF, a Single Algorithm and a Single Formulation as presented by Salhi (2010).

This is all reasonable and promising until one realizes that what we really want are paradigms that capture the different combinations of solution approaches and search spaces when heuristics are involved. Heuristics are the methods of choice when it comes to solving intractable optimization problems; see Talbi (2009). It is therefore necessary to extend the above solution paradigms to include heuristics and solution representations. This is the aim of the present work.

First, we need to define more precisely what we mean by representation. Given an optimisation or search problem, a representation is a finite description of an element of the solution space of the problem and a data structure to hold it. An element could be complex and consist of sub-elements in a hierarchical structure. With this in mind, and since a heuristic can be seen as an algorithm that does not guarantee optimality in finite time, it is easy to see how equivalent solution paradigms to the above can be introduced; MHMR, for instance, would be the most general and powerful paradigm which will take all available heuristics and all available representations and solve both the original problem and the match making problem since, here also, a given representation may not be ideal for a given algorithm. This assumes a single problem formulation but the problem formulation is defined implicitly by the combination of heuristic and representation for these optimisation problems.

The issue is that in most cases, it is very difficult to distinguish between the representation and solution approach. It is also not easy to think of many different representations for a given problem. We will address this issues, provide examples and construct a generalization of the paradigms introduced by Salhi (2010).

References

Hall, J. A. J., and K. I. M. McKinnon. 2004. “The Simplex Examples Where the Simplex Method Cycles and Conditions Where Expand Fails to Prevent Cycling.” *Mathematical Programming* 100: 133–50.

Salhi, A. 2010. “The Ultimate Solution Approach to Intractable Problems.” In *Proceedings of the 6th IMT-GT Conference on Mathematics, Statistics and Its Applications (ICMSA2010)*, 84–93. Universiti Tunku Abdul Rahman.

Salhi, A., and J. A. Vazquez-Rodriguez. 2014. “Tailoring Hyper-Heuristics to Specific Instances of a Scheduling Problem Using Affinity and Competence Functions.” *Journal of Memetic Computing*.

Talbi, E. G. 2009. *Metaheuristics: from Design to Implementation*. Wiley.