J Heuristics (2010) 16: 745–748
DOI 10.1007/s10732-010-9147-x

Special issue on hyper-heuristics in search and optimization

Gabriela Ochoa · Ender Özcan

Published online: 5 October 2010

A hyper-heuristic is an automated methodology for selecting or generating heuristics to solve hard computational search problems. The main feature distinguishing these methods is that they explore a search space of heuristics (rather than a search space of potential solutions to a problem). The goal is that hyper-heuristics will lead to more general systems that are able to automatically operate over a wider range of problem domains than is possible today. The term hyper-heuristic was first used in 1997 to describe a protocol that combines several artificial intelligence methods in the context of automated theorem proving. The term was independently used in 2000 to describe 'heuristics to choose heuristics' in the context of combinatorial optimization. The idea of automating the design of heuristics, however, can be traced back to the early 60s. A more recent research trend in hyper-heuristics attempts to automatically generate new heuristics suited to a given problem or class of problems. This is typically done by combining, through the use of genetic programming for example, components or building-blocks of human designed heuristics.

The first workshop devoted to hyper-heuristics was held in 2008 as part of the 10^{th} International Conference on *Parallel Problem Solving from Nature* (PPSN X) in Dortmund, Germany. The event was lively and well attended; it was seeded by three prominent invited speakers: Prof. Edmund Burke, University of Nottingham, UK; Prof. Roberto Battiti, University of Trento, Italy; and Prof. Riccardo Poli, University of Essex, UK. Over twenty submissions were received and carefully reviewed to select a dozen that were presented during the workshop. In connection with this workshop, we are glad to present this special issue of the *Journal of Heuristics*, devoted to *Hyper-heuristics in Search and Optimization*. Besides inviting the workshop

G. Ochoa (✉) · E. Özcan
Automated Scheduling, Optimisation and Planning (ASAP) Research Group, School of Computer Science, University of Nottingham, Nottingham, UK
e-mail: gxo@cs.nott.ac.uk

presenters to submit extended versions of their work, we sought submissions from the entire research community in automated heuristic design. Twenty-eight manuscripts were received and went through a strict review process, at the end of which seven papers covering a broad range of hyper-heuristic methodologies and application domains were selected for publication in this special issue.

The first three papers present approaches that explore a search space of sequences of domain specific (mainly) constructive heuristics. Constructive heuristics build a complete candidate solution by iteratively extending partial candidate solutions. In "A Scatter Search based Hyper-heuristic for Sequencing a Mixed-Model Assembly Line", Cano-Belmán, Ríos-Mercado and Bautista propose a hyper-heuristic embedded within a scatter search framework. The approach is applied to the problem of sequencing products (mixed-model) on a paced assembly line, and considers a set of 20 priority (or dispatching) rules as low level heuristics. These priority rules are used to select a product among a set of candidates, and are based on product and work station features such as demand, processing time, idle time, work overload, etc. Following the scatter search methodology, the so called *reference set* contains sequences of priority rules, whose combination is based on a rule frequency matrix. Two hyper-heuristic variants were proposed and tested over a wide range of instances from the literature. The solutions obtained were, in many cases, of better quality than those found by previous state-of-the art approaches. The second paper, "A new Dispatching Rule Based Genetic Algorithm for the Multi-objective Job Shop Problem" by Vázquez-Rodríguez and Petrovic, also deals with a production scheduling problem (job shop) and uses priority rules as low level heuristics, but considers a multi-objective formulation of the problem. The search space is composed of sequences of dispatching rules that are called one at a time and used to sequence a number of operations onto machines. This number of operations is found to be a notoriously sensitive parameter. The proposed hyper-heuristic simultaneously searches for the best sequence of rules, and the number of operations to be handled by each rule. The method is inspired by well known evolutionary algorithms for multi-objective optimization. Better results were obtained on all the studied instances, when compared with a previous hyper-heuristic based on dispatching rules, and a conventional genetic algorithm using a permutation representation. The third paper, "DVRP: A Hard Dynamic Combinatorial Optimisation Problem Tackled by an Evolutionary Hyper-heuristic", by Garrido and Riff brings together interesting aspects to hyper-heuristic research by tackling a dynamic problem (vehicle routing), and considering several types of low-level heuristics: constructive, perturbative, and noise heuristics. The proposed framework evolves a sequence of combinations of these three types of heuristics, which are then applied in order to construct and improve partial solutions. The approach is evaluated using a benchmark comprising a large number of instances with different topologies and degrees of dynamism, and it is compared with some well-known methods proposed in the literature. The hyper-heuristic was found to adapt to the dynamic nature of the problem and produced high quality results.

The next three papers are concerned with perturbative low level heuristics or neighborhood structures. These heuristics produce a new candidate solution from a given complete candidate solution by modifying one or more of the corresponding solution components. The first two articles in this group make use of software agents in a distributed framework. In “A Cooperative Hyper-heuristic Search Framework”, Ouelhadj and Petrovic investigate an approach in which a list of elite solutions is built and maintained by a coordinator hyper-heuristic agent, based on the information exchange between the low level heuristic agents. The results on the flow shop scheduling problem show that the cooperative method based on asynchronous communication between the heuristic agents and the coordinator hyper-heuristic agent outperforms the synchronized communication version. Moreover, agent-based cooperation improves the performance of sequential hyper-heuristics. Meignan, Koukam and Créput, describe another self-adaptive agent based approach, “Coalition-Based Metaheuristic: A Self-adaptive Metaheuristic Using Reinforcement Learning and Mimetism”. In this framework, agents are grouped into coalitions to explore the search space of solutions concurrently via asynchronous communication. The hyper-heuristic level within the proposed framework is associated with two interacting roles based on a reinforcement learning strategy. The authors investigate the performance of their approach over a set of benchmark instances of the vehicle routing problem. The results show that the use of reinforcement learning and mimetism improves the quality of solutions, and improved performance over some state-of-the-art methods was achieved. The third paper in this group, “Autonomous Operator Management for Evolutionary Algorithms”, by Maturana, Lardeux and Saubion, explores an encompassing autonomous evolutionary algorithm that decides online which operators will be included in the algorithm and handles their parameters. Moreover, the operators are automatically generated by composing basic sub-operators (producing over 300 crossover operators). So, this approach is also related to automatic heuristic generation. The authors use the canonical problem of satisfaction in propositional logic (SAT) to test their approach, obtaining results that are competitive with an evolutionary algorithm tailored to the problem.

Last but not least, the paper by Løkketangen and Olsson, “Generating Meta-heuristic Optimization Code using ADATE” represents a good example of automatically designing (generating) algorithms. ADATE is a methodology which generates code in a subset of the functional programming language ML. The authors show how this technique can be used to automatically generate metaheuristic code. They apply the methodology to the Boolean optimization problem (BOOP), which can be regarded as a SAT problem with an additional objective, as there is a profit associated with each variable having a true or false value. Code automatically generated by the ADATE system compares with state-of-the-art handcrafted metaheuristic optimization code. In particular, the programs generated the move selection part of a Tabu Search metaheuristic. The results show that the ADATE system is able to generate highly competitive code that produces more optimal solutions to hard BOOP instances within given iteration limits than the previously published Tabu Search implementation. The automatically generated code also gives new insights into the general design of meta-heuristic mechanisms, and contains novel search mechanisms.

This special issue illustrates that a promising direction for developing improved search techniques is to integrate learning components that can adaptively guide the search. Many techniques have independently arisen in recent years that exploit either some form of learning, or search on a configuration space, to improve problem-solving and decision making. Much is to be gained from a greater awareness of the achievements in various cross-disciplinary approaches. Hyper-heuristic research has the potential of bringing together promising ideas in the fields of modern search heuristics and machine learning, with knowledge (in the form of problem-specific heuristics) accumulated over the years in the field of operational research. We hope that this special issue will encourage researchers to continue developing these ideas.