The Cross-Domain Heuristic Search Challenge – An International Research Competition

Edmund K. Burke[1], Michel Gendreau[2], Matthew Hyde[1], Graham Kendall[1], Barry McCollum[3], Gabriela Ochoa[1], Andrew J. Parkes[1], and Sanja Petrovic[1]

[1] Automated Scheduling, Optimisation and Planning (ASAP) Group, School of Computer Science, University of Nottingham, Nottingham, UK
[2] CIRRELT, University of Montreal, Canada
[3] School of Electronics, Electrical Engineering and Computer Science, Queen's University, Belfast, UK

Abstract. The first *Cross-domain Heuristic Search Challenge* (CHeSC 2011) seeks to bring together practitioners from operational research, computer science and artificial intelligence who are interested in developing more generally applicable search methodologies. The challenge is to design a search algorithm that works well, not only across different instances of the same problem, but also across different problem domains. This article overviews the main features of this challenge.

1 Introduction

The *Cross-domain Heuristic Search Challenge*[1] differs from other competitions in search and optimisation, as it aims to measure performance over several problem domains rather than just one. We propose a software framework (*HyFlex*) featuring a common software interface for dealing with different combinatorial optimisation problems. HyFlex provides the algorithm components that are problem specific. In this way, we liberate algorithm designers from needing to know the details of the problem domains and also prevent them from incorporating additional problem specific information in their algorithms. Efforts can instead be focused on designing high-level strategies to intelligently combine the provided problem-specific algorithmic components. The competition is organised and run by the Automated Scheduling, Optimisation and Planning (ASAP) group at the University of Nottingham, Nottingham, UK; with contributions from Queen's University, Belfast, UK; Cardiff University, UK; and the Ecole Polytechnique, Montreal, Canada. Members of these groups will not be allowed to enter the competition.

2 The HyFlex Framework

HyFlex (Hyper-heuristics Flexible framework) [1] is a Java object oriented framework for the implementation and comparison of different iterative general-purpose

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

C.A. Coello Coello (Ed.): LION 5, LNCS 6683, pp. 631–634, 2011.

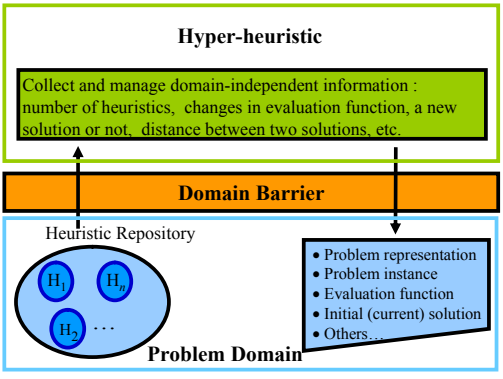

Fig. 1. Hyper-heuristic conceptual framework featuring the domain barrier [2,3]

heuristic search algorithms (also called hyper-heuristics). The framework appeals to modularity and is inspired by the notion of a domain barrier between the low-level heuristics and the hyper-heuristic [2,3] (Figure 1). HyFlex provides a software interface between the hyper-heuristic and the problem domain layers, thus enabling a clearly defined separation, and communication protocol between the domain specific and the domain independent algorithm components.

HyFlex extends the conceptual framework discussed in [2,3] (Figure 1) int that a population of solutions (instead of a single incumbent solution) is maintained in the problem layer. Also, a richer variety of low-level heuristics is provided. Another relevant antecedent to HyFlex is PISA [4], a text-based software interface for multi-objective evolutionary algorithms, which divides the implementation of an evolutionary algorithm into an application-specific part and an algorithm-specific part. HyFlex differs from PISA in that its interface is not text-based but instead given by an abstract Java class. HyFlex is not tied to evolutionary algorithms. It allows the implementation of most single-point and population-based search methods. Moreover, it provides a rich variety of combinatorial optimisation problems including real-world instance data. Each HyFlex problem domain module consists of:

1. A routine to initialise randomised solutions in the population.
2. A set of heuristics to modify solutions classified into four groups:
   mutational : makes a (randomised) modification to the current solution.
   ruin-recreate : destroys part of the solution and rebuilds it using a constructive procedure.
   local search : searches in the *neighbourhood* of the current solution for an improved solution.
   crossover : takes two solutions, combines them and returns a new solution.
3. A varied set of instances that can be easily loaded.
4. A population of one or more solutions that has to be administered.

For testing purposes, four domain modules are provided each containing around 10 low-level heuristics of the types discussed above, and 10 instances of medium to hard difficulty. The domains provided are: permutation flowshop, one dimensional bin packing, Boolean satisfiability (MAX-SAT) and personnel scheduling. Technical reports describing the details of each of these modules, are available at the competition Web site ('Documentation' section: `http://www.asap.cs.nott.ac.uk/chesc2011/documentation.html`).

3 Challenge Description and Scoring System

For the competition, a number instances from each of these four test domains will be considered (including both training and hidden instances). Additionally, at least two hidden domains will also form part of the competition. These additional domains will be revealed only after the competition has been completed. For each instance, a single run will be conducted, and all the competing algorithms will start from the same initial solution generated from the same random seed. The run time will be limited to 10 minutes (measured in CPU time) on a modern PC running Windows XP. This figure was selected empirically after extensive testing on our problem domains' hardest instances. A benchmarking program (for both Windows and Linux) is available from the Web site that will report the time it takes a competitor's computer to execute a set of instructions that in the competition computer takes 10 minutes (600 seconds). It is worth noting that all the competitors will be run on a standard machine therefore creating a "level playing field".

In order to compare the performance of the competing hyper-heuristics and declare the winner, we will use a scoring system inspired by Formula 1. Before 2010, the Formula 1 system had the following structure. The top eight drivers scored 10, 8, 6, 5, 4, 3, 2 and 1 points respectively, in each race. These points are added for all the events, and the winner is the driver accumulating the most points. This is adapted for the cross-domain challenge as follows. Let us assume that m instances (considering all the domains) and n competing algorithms in total are considered. For each experiment (instance) an ordinal value o_k is given representing the rank of the algorithm compared to the others ($1 \leq o_k \leq n$). The top eight ranking algorithms will receive the points as in the Formula 1 system described above, and the remaining algorithms will receive no points. The points will be added across the m instances, and the winner will be the algorithm accumulating the most points. Therefore, if for example, five problem domains are considered with five instances each, the maximum possible score is 250 points. For solving ties we will also follow Formula 1. Full details can be seen on the competition Web site ('Scoring System' section: `http://www.asap.cs.nott.ac.uk/chesc2011/scoring.html`).

4 Final Remarks

Extensive tests (some of them published in [5]), have confirmed that a rich set of state-of-the-art hyper-heuristics can be implemented with HyFlex. Both single

point and population based search algorithms can be designed. The Java jar file implementing the framework can be downloaded from the website, which also provides a tutorial, several examples, and the relevant software and academic documentation. An additional interesting feature is the Leaderboard, a table ranking participants according to their best score on a rehearsal competition conducted every week. This rehearsal competition is based on a set of results submitted by the participants who chose to do so. Note that only the results, and not the algorithms, are required for the Leaderboard submissions.

The prize fund is 3,000 GBP to be split between the first, second and third place competitors. The winners will be announced at OR53 (UK Operational Research Society conference, to be held in Nottingham, UK in September 6 - 8, 2011) and their registration fee will be waived. Our goal is to both promote research into more general search methodologies, and also to gain a deeper understanding of the algorithm design principles and machine learning techniques that work well in practice.

Acknowledgments. We would like to thank Aptia solutions Ltd, EventMap Ltd, Staff Rostering Solutions Ltd, the PATAT steering committee and the UK Operational Research Society for sponsoring the competition and providing the prize money.

References

1. Burke, E.K., Curtois, T., Hyde, M., Kendall, G., Ochoa, G., Petrovic, S., Vazquez-Rodriguez, J.A.: HyFlex: A flexible framework for the design and analysis of hyper-heuristics. In: Multidisciplinary International Scheduling Conference (MISTA 2009), Dublin, Ireland, pp. 790–797 (August 2009)
2. Cowling, P., Kendall, G., Soubeiga, E.: A hyperheuristic approach to scheduling a sales summit. In: Burke, E., Erben, W. (eds.) PATAT 2000. LNCS, vol. 2079, pp. 176–190. Springer, Heidelberg (2001)
3. Burke, E.K., Hart, E., Kendall, G., Newall, J., Ross, P., Schulenburg, S.: Hyper-heuristics: An emerging direction in modern search technology. In: Glover, F., Kochenberger, G. (eds.) Handbook of Metaheuristics, pp. 457–474. Kluwer, Dordrecht (2003)
4. Bleuler, S., Laumanns, M., Thiele, L., Zitzler, E.: PISA – A Platform and Programming Language Independent Interface for Search Algorithms. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) EMO 2003. LNCS, vol. 2632, pp. 494–508. Springer, Heidelberg (2003)
5. Burke, E.K., Curtois, T., Hyde, M., Kendall, G., Ochoa, G., Petrovic, S., Vazquez-Rodriguez, J.A., Gendreau, M.: Iterated local search vs. hyper-heuristics: Towards general-purpose search algorithms. In: IEEE Congress on Evolutionary Computation (CEC 2010), Barcelona, Spain, pp. 3073–3080 (July 2010)