A GP-Based Hyper-Heuristic Framework for Evolving 3-SAT Heuristics

Mohamed Bader-El-Den
Department of Computer Science
University of Essex
Wivenhoe Park, United Kingdom
mbbade@essex.ac.uk

Ricccardo Poli
Department of Computer Science
University of Essex
Wivenhoe Park, United Kingdom
rpoli@essex.ac.uk

ABSTRACT

We present, GP-HH, a framework for evolving local search 3-SAT heuristics based on GP. Evolved heuristics are compared against well-known SAT solvers with very encouraging results.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*

General Terms

Algorithms, Experimentation.

Keywords

Genetic Programming, Hyper-Heuristic, SAT, Heuristics.

1. INTRODUCTION

Hyper-heuristics are "heuristics to choose heuristics" [1]. There are two main classes of hyper-heuristics. In a first class, the system is provided with a list of pre-existing heuristics for solving a certain problem. The system then tries to discover the best sequence of application for these heuristics in order to solve the problem. A second approach aims at constructing new heuristics. This is the approach we adopt in this paper. The process starts by selecting a suitable set of heuristics that are known to be useful in solving a certain problem. However, instead of directly using these heuristics, the heuristics are first decomposed into their basic elements. A grammar is used to constrain how these elements can be composed. The system then uses a grammar-respecting genetic programming system to evolve new heuristics. We call the system GP-HH.

We apply GP-HH to the satisfiability (SAT) problem. The target of SAT is to determine whether it is possible to set the variables of a given Boolean expression in such a way to make the expression true. The expression is called satisfiable if such an assignment exists. Stochastic local search heuristics have been widely used for solving SAT. These include GSAT [2], HSAT, GWSAT and WalkSat. By analysing these heuristics, we identified their components and designed a simple grammar that can represent any of them within GP-HH, the main parts of the grammar shown in Figure 1.

GECCO'07, July 7–11, 2007, London, England, United Kingdom.
ACM 978-1-59593-697-4/07/0007.

Table 1: Success rate of the evolved heuristic compared to standard ones on uf20, uf50, uf75 SatLib benchmark

	GSAT	GWSAT	HH20	HH50	HH75
uf-20	0.317	0.907	1.0	-	-
uf-50	0.238	0.606	-	0.743	-
uf-70	0.190	0.502	-	-	0.560

```
start  →  FLIP v
v      →  RANDOM l | IFV pro, v, v
          MAX_SCR l [, op] | MIN_SCR l [, op]
          MAX_AGE l [, op]
l      →  ALL | IFL pro, l, l
          ALL_USC | RAND_USC | SCR_Z l [, op]
op     →  TIE_RAND | TIE_AGE | TIE_SCR
pro    →  0.2 | 0.4 | 0.5 | 0.7 | 0.8 | 0.9
```

Figure 1: The main parts of grammar used for evolving heuristics for SAT using GP-HH.

`MAC_SCR l` selects a variable from list `l` that by flipping will maximize the number of satisfied clauses, `MAC_AGE` selects the least recently flipped variable, `ALL` returns all the variables in the formula, while `ALL_USC` returns the variables in all unsatisfied clauses.

The GP-HH was tested on benchmark problems taken form the 3-SAT SatLib. All the problems in our benchmarks were satisfiable uniform random 3-SAT problems, with 20, 50, 75, 100, 125 and 150 variables. The objective of the experiments was to evolve a separate heuristic that best performs on a certain SAT instance. Results of GP-HH on a 20, 50, 75 variable SAT sets are shown Table 1. The table shows the success rate of each heuristics on 1,000 test cases. As one can see the evolved heuristics are able to solve more instances than the shown standard heuristics.

2. REFERENCES

[1] E. Burke, G. Kendall, J. Newall, E. Hart, P. Ross, and S. Schulenburg. Hyper-heuristics: an emerging direction in modern search technology. In F. Glover and G. Kochenberger, editors, *Handbook of Metaheuristics*, pages 457–474. Kluwer Academic Publishers, 2003.

[2] B. Selman, H. J. Levesque, and D. Mitchell. A new method for solving hard satisfiability problems. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 440–446, Menlo Park, California, 1992. AAAI Press.