Hybridization of Genetic Algorithms and Constraint Propagation for the BACP

Tony Lambert[1,2], Carlos Castro[3], Eric Monfroy[1,3],
María Cristina Riff[3], and Frédéric Saubion[2]

[1] LINA, Université de Nantes, France
Firstname.Name@lina.univ-nantes.fr
[2] LERIA, Université d'Angers, France
Firstname.Name@univ-angers.fr
[3] Universidad Santa María, Valparaíso, Chile
Firstname.Name@inf.utfsm.cl

1 Introduction

Constraint Satisfaction Problems (CSP) provide a modelling framework for many computer aided decision making problems. Many of these problems are associated to an optimization criterion. Solving a CSP consists in finding an assignment of values to the variables that satisfies the constraints and optimizes a given objective function (in case of an optimization problem). In this paper, we extend our framework for genetic algorithms (GA) as suggested by the reviewers of our previous ICLP paper [5]. Our purpose is not to solve efficiently the Balanced Academic Curriculum Problem (BACP) [2] but to combine a genetic algorithm with constraint programming techniques and to propose a general modelling framework to precisely design such hybrid resolution process and highlight their characteristics and properties.

2 Computational Frameworks: CP vs GA

Constraint propagation, one of the most famous techniques for solving CSP consists in iteratively reducing domains of variables by removing values that do not satisfy the constraints. These reductions must be interleaved with a splitting mechanism in order to obtain a complete solver. Constraint optimization problems, although similar to constraint solving, is comparatively harder because it only accepts solutions that minimize or maximize a given objective function while satisfying the constraints. The key principle of Genetic Algorithms approach states that, species evolve through adaptations to a changing environment and that the gained knowledge is embedded in the structure of the population and its members, encoded in their chromosomes. Applying a genetic algorithm consists in iteratively generating better and better individuals w.r.t. an evaluation function.

In the context of GA, for the resolution of a given CSP (X, D, C), the search space can be usually defined with the set of tuples $D = D_1 \times \cdots \times D_n$. We

M. Gabbrielli and G. Gupta (Eds.): ICLP 2005, LNCS 3668, pp. 421–423, 2005.

consider populations g of size i, $g \subseteq D$ such as $|g| = i$. An element $s \in g$ is an individual and represents a potential solution to the problem.

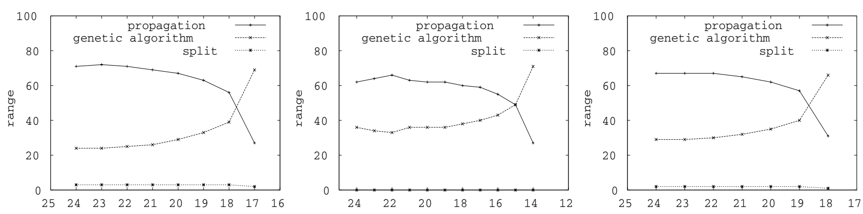
In order to handle the different data structures associated to each side of the resolution (GA & CP), we add a particular genetic factor to the search space to each CSP, on which GA will work and where optimization will be done. A CSP with genetic factor (ogCSP) for optimization is defined by a sequence (D, C, p) where $p = (g_1, \dots, g_n)$ corresponds to sequence of generations.

Based on the framework of chaotic iterations [1], resolution will be achieved according to the generic algorithm through a partial ordering where the set of functions is instantiated by reduction, split and GA functions.

3 Experimentation

The purpose of this section is to highlight the benefit of our framework for hybridization. We consider the bacp8, bacp10 and bacp12 problems issued from the CSPlib [3]. We control the rates of each family of functions *reduction*, *split* and *ga* by giving as strategy a tuple $(\%_{red}, \%_{sp}, \%_{ga})$ of application rates. These values correspond indeed to a probability of application of a function of each family but, in practice, we measure in Fig 1 the real rate of participation during the objective function evolution.

The most interesting in such an hybridization is the completeness of the association GA-CP, and the roles played by GA and CP in the search process : GA optimizes the solutions in a search space progressively being locally consistant (and thus smaller and smaller) using constraint propagation and split.



8-period problem 10-period problem 12-period problem

Fig. 1. Evolution of CP vs GA during the optimization process

Concerning strategies using *GA* **and** *CP* **alone**, in this implementation, CP is unable to find a feasible solution in 10 minutes cpu time. GA is able to find alone the optimal but is 10 times slower w.r.t. the hybrid.

4 Perspectives and Conclusion

In this paper, we have used a more suitable general framework to model hybrid optimization solving algorithms. The results over the BACP show the benefits of our framework and of hybridization. They also allow us to identify the interaction between the different resolution mechanisms. Such studies could be used to tune general purpose hybrid solvers in the future.

References

1. K. R. Apt. From chaotic iteration to constraint propagation. In *Proceedings of ICALP'97*, pages 36–55. Springer-Verlag, 1997.
2. C. Castro and S. Manzano. Variable and value ordering when solving balanced academic curriculum problems. In *Proceedings of 6th Workshop of the ERCIM WG on Constraints. CoRR cs.PL/0110007*, 2001.
3. I. Gent, T. Walsh, and B. Selman. http://www.csplib.org, funded by the UK Network of Constraints.
4. J. H. Holland. *Adaptation in Natural and Artificial Systems*. 1975.
5. E. Monfroy, F. Saubion, and T. Lambert. On hybridization of local search and constraint propagation. *Proc. of ICLP'04*, *LNCS 3132*, pp 299–313. Springer, 2004.