Constraint-based Local Search For Discrete Optimization

Philippe CODOGNET[*] Daniel DIAZ[†]

Abstract

In the last years, the application of local search techniques for constraint solving started to raise some interest in the CSP community, We propose a generic, domain-independent local search method called Adaptive Search for solving Constraint Satisfaction Problems (CSP), this method has been fully implemented, with both an efficient C-based system and a Java-based framework available as freeware. We have designed a new meta-heuristics that takes advantage of the structure of the problem in terms of constraints and variables and can guide the search more precisely than a global cost function to optimize (such as for instance the number of violated constraints). We also use an adaptive memory in the spirit of Tabu Search in order to prevent stagnation in local minima and loops. This method is generic, can apply to a large class of constraints (e.g. linear and non-linear arithmetic constraints, symbolic constraints, etc) and naturally copes with over-constrained problems. Preliminary results on some classical CSP problems (Nqueens, magic squares, all-interval series, number partitioning, etc) show very encouraging performances. The input of the method is a problem in CSP format, that is, a set of variables with their (finite) domains of possible values and a set of constraints over these variables. We also need, for each constraint, an error function that will give an indication on how much the constraint is violated. For instance the "error" function associated to an arithmetic constraint X - Y < C will be max(0, |X-Y|-C). Adaptive search relies on iterative repair, based on variables and constraint errors information, seeking to reduce the error on the worse variable so far. The basic idea is to compute the error function of each constraint, then combine for each variable the errors of all constraints in which it appears, therefore projecting constraint errors on involved variables. Finally, the variable with the maximal error will be chosen as a "culprit" and thus its value will be modified. In this second step we use the well-known min-conflict heuristics and select the value in the variable domain that has the best temptative value, that is, the value for which the total error in the next configuration is minimal. In order to prevent being trapped in local minima, the Adaptive Search method also includes an adaptive memory à la Tabu Search (variables can be marked Tabu and “frozen” for a few iterations), but also integrates possible restart-based transitions to escape stagnation around local minima. Restarts are partial and are guided by the number of variables being marked Tabu.
We are currently investigating multi-point extensions (à la genetic algorithms) and more flexibility and stochasticity in the choice of the variable to modify.

Keywords: Metaheuristics, constraints, CSP, optimization

---

[*] University of Paris 6, LIP6, case 169, 8 rue du Capitaine Scott, 75015 Paris, France
Philippe.Codognet@lip6.fr

[†] University of Paris 1, CRI, bureau C1407, 90 rue de Tolbiac, 75 634 Paris Cedex 13, France
Daniel.Diaz@univ-paris1.fr