A Data Structure Boosting The Performance of Local Search For CSP Solving

Éric Grégoire, Jean-Marie Lagniez, Bertrand Mazure

A data structure boosting the performance of local search for CSP solving

É. Grégoire[1] and J.-M. Lagniez[2] and B. Mazure[1]

[1] CRIL UMR 8188 CNRS, Université d'Artois, rue Jean Souvraz, F-62307 Lens Cedex
{gregoire,mazure@cril.fr}
[2] Institute for Formal Models and Verification, Johannes Kepler University, AT-4040 Linz, Austria
jmlagniez@gmail.com

Abstract

This paper is concerned with local search techniques (LS) for solving CSPs (Constraint Satisfaction Problems). An efficient data structure is presented that allows the performance of LS to be boosted. Experimentations on benchmarks from the last international CSP competitions illustrate its very positive impact. It has been implemented in WCSP$_\delta$: an efficient open-ended and open-source local search platform for CSP that can accommodate various meta-heuristics.

1 CSPs

Let us recall what a (discrete) CSP (Constraint Satisfaction Problem) is. A CSP is pair $(\mathcal{V},\mathcal{C})$ where

1. $\mathcal{V}$ is a finite set of n variables s.t. each variable $x \in \mathcal{V}$ has an associated finite instantiation domain, denoted $dom(x)$, which contains the set of values allowed for x,
2. $\mathcal{C}$ is a set of constraints s.t. each constraint $c \in \mathcal{C}$ involves a subset of variables of $\mathcal{V}$, called scope and denoted $vars(c)$, and has an associated relation $rel(c)$, which contains the set of tuples allowed for the variables of its scope.

Solving a CSP $\mathcal{P} = (\mathcal{V},\mathcal{C})$ consists in checking whether $\mathcal{P}$ admits at least one solution, i.e. an assignment of values for all variables of $\mathcal{V}$ s.t. all constraints of $\mathcal{C}$ are met. When $\mathcal{P}$ admits at least one solution, $\mathcal{P}$ is said to be satisfiable, otherwise $\mathcal{P}$ is said unsatisfiable. This decision problem is naturally generalized into an optimization one, called MAX-CSP, which consists in searching for one largest possible subset of constraints that can be satisfied at the same time.

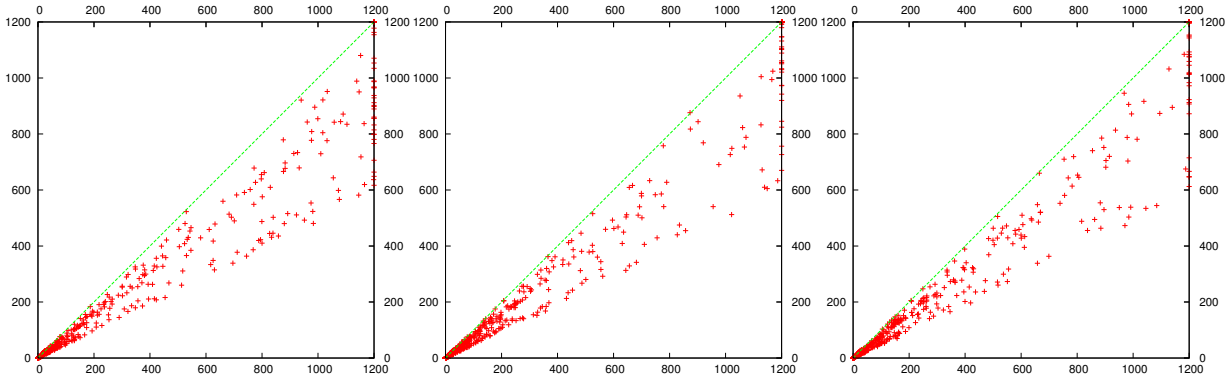
2 Local search for CSP: an enhanced data structure

The use of LS for CSP solving has been the subject of many studies these last decades [4, 1, 6]. It is well-known that smart data structures can play a key role in the performance of the various possible LS algorithms. In this specific context, we considered the data structure first proposed in [1], which is now quite standard in LS for CSPs. The key idea of the structure is to record, for each constraint c, for each possible value val of each variable v occurring in c, the corresponding number of unsatisfied constraints to which the variable v would belong if the next move from the current interpretation consists in adopting the value val for the variable v. Indeed, as most moves will involve adopting the value of a variable that would decrease the number of falsified constraints, such a recorded information allows this selection to be made efficiently. Moreover, updating the data structure only concerns the constraints affected by this last move. Accordingly, this data structure allows many repetitive checks and computations to be avoided during the successive iterative steps of LS.

Although this data structure records information about the impact that the change of value of a variable entails, it does not record whether or not the constraints containing the variables would see their satisfiability status affected by this change, although this is already computed. Accordingly, we enhanced the data structure by recording for each value val for each variable in each constraint c, whether a move to this value val would allow the constraint c to be satisfied

	2-EXT			2-INT			N-EXT			N-INT		
Method	SAT	UNS	TOT	SAT	UNS	TOT	SAT	UNS	TOT	SAT	UNS	TOT
WCSP$_s$(WalkSat)	239	0	239	160	0	160	179	0	179	271	0	271
WCSP$_\delta$(WalkSat)	**248**	0	**248**	**167**	0	**167**	**181**	0	**181**	**283**	0	**283**
WCSP$_s$(novelty)	308	0	308	159	0	159	222	0	222	338	0	338
WCSP$_\delta$(novelty)	**315**	0	**315**	**165**	0	**165**	**223**	0	**223**	**350**	0	**350**
WCSP$_s$(rnovelty)	319	0	319	161	0	161	208	0	208	321	0	321
WCSP$_\delta$(rnovelty)	**325**	0	**325**	**173**	0	**173**	**209**	0	**209**	**325**	0	**325**

Table 1. Comparaison between WCSP$_\delta$ and WCSP$_s$.



(a) WalkSat (b) novelty (c) rnovelty

Fig. 1. Scatter plot comparing the runtimes of WCSP$_s$ (x-axis) and WCSP$_\delta$ (y-axis).

or not. Interestingly, this information can be computed and recorded in constant additional time while initializing and updating the data structure of [1]. Moreover, this simplifies the updating process of the initial data structure since the satisfiability status of a constraint does not have to be computed but is simply consulted in the enhanced data structure.

3 Experimentations

Not surprisingly, this enhancement of the data structures has a very positive impact on the LS performance. It has been experimented extensively on all benchmarks from the CSP'2008 international competition (on Intel Xeon 3.2. GHZ 2G RAM under Linux 2.6) through WCSP$_\delta$, which is our generic, open-ended and open-source LS platform for CSP. As a case study, we considered the seminal WalkSAT [5] LS algorithm and its variants involving the Novelty and RNovelty [3] paradigms. Table 1 shows that adopting the enhanced data structure allowed more instances to be solved within a preset 1200 seconds CPU time limit. Figure 1 also show the very positive impact of the enhanced data structure on the required CPU time to solve instances (x-axes and y-axes give computing time for each instance, with or without the enhancement of the data structure, respectively). The performance of WCSP$_\delta$ has also been demonstrated as a building block of an hybrid CSP solver based on this LS platform [2].

References

1. Galinier P. and Hao J.-K.: A general approach for constraint solving by local search. Proc. of CP 1997, LNCS 1330, pp. 196-208 (1997).
2. Grégoire. É., Lagniez J.-M. and Mazure B.: A CSP solver focusing on FAC variables, Proc. of CP 2011, LNCS 6876, pp. 493-507 (2011).
3. McAllester D.A., Selman B. and Kautz H.A.: Evidence for invariants in local search. Proc. of AAAI'97, pp. 321-326 (1997).
4. Minton S., Johnston M., Philips A. and Laird P.: Solving large-scale constraint satisfaction and scheduling problems using a heuristic repair method. Proc. of AAAI 1990, pp. 17-24 (1990).
5. Selman N., Kautz H.A. and Cohen B.: Noise strategies for improving local search. Proc. of AAAI'94, pp. 337-343 (1994). Proc. of AAAI 1990, pp. 17-24 (1990).
6. Van Hentenryck P. and Michel L. 2005. Constraint-Based Local Search. Cambridge, Mass.: The MIT Press. ISBN 9780262220774.