Maximizing the reversible DAC lower bound in Max-CSP is NP-complete

Thomas Schiex
Rapport technique INRA-BIA 1998/02

Juillet 1998

Abstract

We show that the decision problem associated with the problem of maximizing the lower-bound used in branch and bound algorithms based on so-called "reversible directed arc consistency counts" (Larrosa *et al.* 1998; 1999) is NP-complete.

Nous montrons que le problème de décision associé au problème de maximisation du minorant utilisé dans l'algorithme de séparation et évaluation basé sur les "compteurs de cohérence d'arc dirigée réversible" (Larrosa *et al.* 1998; 1999) est NP-complet.

Chapter 1

Theorem anf proof

1.1 Preliminaries

We consider the following decision problem, called MAX-REV-DAC: we are given a binary CSP = (X, D, C) where each constraint $c_{ij} \in C$ is weighted with $w_{ij}) \in \mathbb{N}^+$. The problem is to prove the existence of a direction i or j for each constraint such that the lower bound defined by reversible DACs is larger than $\alpha \in \mathbb{N}^+$ (see (Larrosa *et al.* 1998; 1999) for a formal definition of the lower bound).

1.2 Proof

This problem is naturally in NP since if we are given an orientation for each constraint, the problem of computing the lower bound is polynomial.

We will now prove that the problem is NP-complete through a polynomial transformation from MAX-2SAT. Consider an instance of MAX-2SAT, defined by a set Φ of e 2-clauses. The problem is to prove the existence of an interpretation ω that satisfies at least k 2-clauses of Φ.

We note $\mathcal{V}$ the set of all the propositional variables that appear in a 2-clause of Φ. For each propositional variable $v \in \mathcal{V}$, we note $\#v^+$ (resp. $\#v^-$) the number of positive (resp. negative) occurrences of v in the 2-clauses of Φ. Finally, we note $\#v = \max(\#v^+, \#v^-)$.

From Φ, we build a weighted Max-CSP using two types of gadgets: one for representing propositional variables, one for representing 2-clauses.

1. For each propositional variable $v \in \mathcal{V}$, the corresponding gadget includes $2 \times \#v$ CSP variables and $4 \times \#v$ constraints.

   - the first $\#v$ variables are noted $v_0^+, \ldots, v_{\#v-1}^+$. These variables will be called positive instances of the propositional variable v.
   - the last $\#v$ variables are noted $v_0^-, \ldots, v_{\#v-1}^-$. These variables will be called negative instances of the propositional variable v.

   Each variable has three values in its domain noted respectively l (for left), r (for right) and s for satisfied.

   The $4 \times \#v$ constraints are separated in two groups. The first group is composed of $2 \times \#v$ unary constraints each involving a different variable. These constraint have a weight $e + 1$ and simply forbid value s. This generates a DAC equal to $e + 2$ for all the values s[1].

[1]A binary constraint could have been used here, but would introduce one more variable for each constraint

The second group connects all the variables associated to the original propositional variable in a simple cycle that alternatively goes trough an v^+ and v^- variable. These constraints have all the same weight $e+2$.

- the first $\#v$ binary constraints connect v_i^+ with v_i^-. These constraints authorize only two pairs of values: $\{(r, l), (s, s)\}$.
- $\#v$ constraints connect v_i^- with $v_{i+1 \bmod \#v}^+$. These constraints authorize only two pairs of values: $\{(l, r), (s, s)\}$.

It should be noted that unless the two constraints that are incident to a variable are oriented towards the variable, its associated minimum will be equal to 0 since either l or r will have a DAC equal to 0 (and no other constraint will have influence on these DACs).

The gadget principle is that it has only two optimal configurations: either all the constraints are oriented towards positive variable instances or negative variable instances. Any other configuration will generate a loss in the lower bound which is at least equal to $(e+1)$ i.e., strictly larger than the number of 2-clauses.

The configuration where all constraints are directed towards positive (resp. negative) variable instances correspond to the fact that the associated propositional variable is true (resp. false).

2. for each 2-clause, the second type of gadget is a simple binary constraint with weight 1. If v_1 and v_2 are the two variables involved in the two clause, then the constraint will connect an instance of v_1 and an instance of v_2, each with the same sign as in the 2-clause. All these constraints authorizes the two pairs $\{(l, r), (r, l)\}$.

   The idea is that if the 2-clause is satisfied this means that one of its literal is true, which mean that the 2 constraints in the cycle associated to the variable are oriented towards the corresponding variable instance. The DACs being resp. $e+2$, $e+2$ and $e+1$ it is possible to still increase the lower bound by 1 by redirecting the constraint towards this instance.

To illustrate this, consider $\Phi = \{a \vee b, a \vee \neg b, \neg a \vee b, \neg a \vee \neg b\}$. We have $\#a^+ = \#a^- = \#b^+ = \#b^- = 2$ and therefore $\#a = \#b = 2$. The corresponding weighted CSP consistency graph is given in Fig. 1.1. The arrows indicate directions that corresponds to the interpretation $\omega = \{a, b\}$ that satisfies 3 of the 2-clauses of Φ. The corresponding lower bound is $5.(2+2)+3 = 23$.

We will now prove that there is an orientation for each constraint in this problem such that the reversible DAC lower bound is larger than (or equal to) $k + (e+1).\sum_{v\in\mathcal{V}} \#v$ iff there exists an interpretation that satisfies more than k clauses of the original MAX-2SAT problem.

First, consider an interpretation ω that satisfies at least k clauses. For each cycle of variables that represents a propositional variable, we orient the constraints in the cycle towards positive instances (resp. negative) if the value of the variable is true (resp. false) in ω. This means that instance of literals which are true in ω will have DACs of $e+2$, $e+2$ and $e+1$ on the values l, r, s (minimum of $e+1$) respectively while instances of falsified literals will have DACS of respectively 0, 0 and $e+1$ (minimum of 0). This immediately brings a lower bound of $(e+1).\sum_{v\in\mathcal{V}} \#v$.

Then for each satisfied 2-clause, we orient the constraint that represents the 2-clause towards the instance of one of the variable that satisfies the 2-clause. The corresponding variables will have augmented DACs on the values l and r bringing the DAC values to $e+2$ for all three values. The corresponding minimum will therefore also be incremented from $e+1$ to $e+2$. This will bring the lower bound to the expected value of $k + (e+1).\sum_{v\in\mathcal{V}} \#v$. The orientation of the remaining constraints (violated clauses) is done arbitrarily: the clause being violated, the two literals are false and therefore the DACs of l and r are equal to 0, and the the minimum too. These constraints cannot contribute to the lower bound.

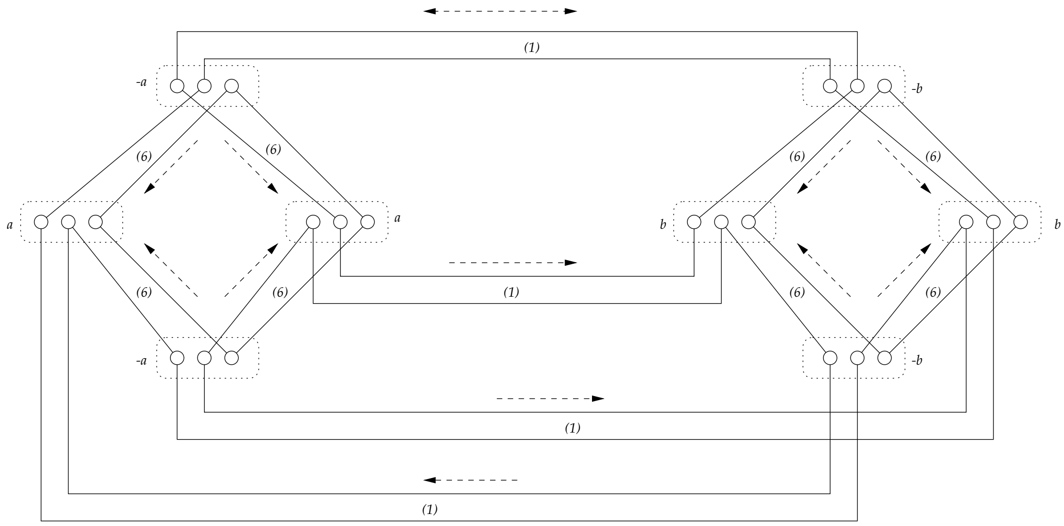

Figure 1.1: The consistency graph for $\Phi = \{a \vee b, a \vee \neg b, \neg a \vee b, \neg a \vee \neg b\}$. Unary constraints are omitted. Weights are indicated between parenthesis.

Conversely, consider that there is an orientation of the constraints that brings a lower bound of $k + (e+1).\sum_{v \in \mathcal{V}} \#v$. Since each cycle cannot contribute more than $(e+1).\#v$ (where v is the propositional variable associated to the cycle) and $k < (e+1)$, it is necessary that each cycle brings exactly $(e+1).\#v$. The two only possible way of obtaining this cost is by orienting all constraints towards either positive or negative instances of the propositional variables. Now consider an interpretation where each propositional variable has the truth value indicated by the constraints above. The only possible way of increasing the existing $\sum_{v \in \mathcal{V}}[(e+1).\#v]$ by k units is by orienting constraints representing 2-clauses to one true literal. There is therefore at least k clauses which involve a true literal and which are therefore satisfied.

References

J. Larrosa, P. Meseguer, T. Schiex, and G. Verfaillie. Reversible DAC and other improvements for solving max-CSP. In *Proc. of the National Conf. on Artificial Intelligence (AAAI'98)*, Madison, WIN, July 1998. American Association for Artificial Intelligence, AAAI Press and the MIT Press.

J. Larrosa, P. Meseguer, and T. Schiex. Maintaining reversible DAC for Max-CSP. *Artificial Intelligence*, 107(1):149–163, January 1999.