# Approximating Weighted Max-SAT Problems by Compensating for Relaxations

Arthur Choi, Trevor Standley, and Adnan Darwiche

Computer Science Department
University of California, Los Angeles
Los Angeles, CA 90095
{aychoi,tstand,darwiche}@cs.ucla.edu

**Abstract.** We introduce a new approach to approximating weighted Max-SAT problems that is based on simplifying a given instance, and then tightening the approximation. First, we relax its structure until it is tractable for exact algorithms. Second, we compensate for the relaxation by introducing auxiliary weights. More specifically, we relax equivalence constraints from a given Max-SAT problem, which we compensate for by recovering a weaker notion of equivalence. We provide a simple algorithm for finding these approximations, that is based on iterating over relaxed constraints, compensating for them one-by-one. We show that the resulting Max-SAT instances have certain interesting properties, both theoretical and empirical.

## 1 Introduction

Relaxations are often used to tackle optimization problems, where a tractable relaxed problem is used to approximate the solution of an intractable one. Indeed, they are employed by a few recently proposed solvers for the maximum satisfiability (Max-SAT) problem [1, 2], which have shown to be competitive for certain classes of benchmarks in recent Max-SAT evaluations. In these solvers, a given Max-SAT instance is relaxed enough until it is amenable to an exact solver. Upper bounds computed in the resulting relaxation are then used in a branch-and-bound search to find the Max-SAT solution of the original instance.

Whether a relaxation is used in a branch-and-bound search, or used as an approximation in and of itself, a trade-off must be made between the quality of a relaxation and its computational complexity. The perspective that we take in this paper, instead, is to take a given relaxation, infer from its weaknesses, and compensate for them. Since we assume that reasoning about the original problem is difficult, we can exploit instead what the relaxed problem is able to tell us, in order to find a tighter approximation.

In this paper, we propose a class of weighted Max-SAT approximations that are found by performing two steps. First, we *relax* a given weighted Max-SAT instance, which results in a simpler instance whose solution is an upper bound on that of the original. Second, we *compensate* for the relaxation by correcting for deficiencies that were introduced, which results in an approximation with improved semantics and a tighter upper bound. These new approximations, can in turn be employed by other algorithms that rely on high quality relaxations.

More specifically, we relax a given weighted Max-SAT problem by removing from it certain equivalence constraints. To compensate for each equivalence constraint that we relax, we introduce a set of unit clauses, whose weights restore a weaker notion of equivalence, resulting in a tighter approximation. In the case a single equivalence constraint is relaxed, we can identify compensating weights by simply performing inferences in the relaxation. In the case multiple equivalence constraints are relaxed, we propose an algorithm that iterates over equivalence constraints, compensating for them one-by-one. Empirically, we observe that this iterative algorithm tends to provide monotonically decreasing upper bounds on the solution of a given Max-SAT instance.

Proofs are given in the Appendix, or in the full report [3], in the case of Theorem 4. For an introduction to modern approaches for solving and bounding Max-SAT problems, see [4].

## 2 Relaxing Max-SAT Problems

Max-SAT is an optimization variant of the classical Boolean satisfiability problem (SAT). Given a Boolean formula in clausal form, the goal is to find an assignment of variables $X$ to truth values $x$ or $\bar{x}$, that maximizes the number of satisfied clauses. In the *weighted* Max-SAT problem, each clause is associated with a non-negative weight and the goal is to find an assignment that maximizes the aggregate weight of satisfied clauses. The weighted *partial* Max-SAT problem further specifies clauses that are hard constraints, that must be satisfied by any solution.[1] In this paper, we focus on weighted Max-SAT problems, although we will formulate relaxations in terms of hard constraints, as we shall soon see.

Let $f = \{(C_1, w_1), \ldots, (C_m, w_m)\}$ be an instance of weighted Max-SAT over variables $X$, where $w_j$ is the weight of a clause $C_j$. Let $\mathbf{X}$ denote the set of all variables in $f$, and let $\mathbf{x}$ denote an assignment of variables $X$ to truth values $x$ or $\bar{x}$ (we also refer to truth values as signs). An optimal Max-SAT assignment $\mathbf{x}^\star$ is an assignment that maximizes the aggregate weight of satisfied clauses: $\mathbf{x}^\star = \text{argmax}_{\mathbf{x}} \sum_{\mathbf{x} \models C_j} w_j$. We denote the corresponding value of a Max-SAT solution by $F^\star = \max_{\mathbf{x}} \sum_{\mathbf{x} \models C_j} w_j$. Note that a weighted Max-SAT instance $f$ may have multiple assignments $\mathbf{x}^\star$ that are optimal, so we may refer to just the optimal value $F^\star$ when the particular optimal assignment $\mathbf{x}^\star$ is not relevant.

Let $\mathbf{Z} \subseteq \mathbf{X}$ denote a subset of the variables in instance $f$. We are also interested in Max-SAT solutions under partial assignments $\mathbf{z}$. More specifically, let $\mathbf{x} \sim \mathbf{z}$ denote that $\mathbf{x}$ and $\mathbf{z}$ are compatible instantiations, i.e., they set common variables to the same values. We then denote the value of a Max-SAT solution under a partial assignment $\mathbf{z}$ by

$$F(\mathbf{z}) = \max_{\mathbf{x} \sim \mathbf{z}} \sum_{\mathbf{x} \models C_j} w_j.$$

[1] In practice, hard constraints can be represented by clauses with large enough weights.

We will be particularly interested in the value of a Max-SAT solution when a single variable $X$ is set to different signs, namely $F(x)$ when variable $X$ is set positively, and $F(\bar{x})$ when variable $X$ is set negatively. Note, for an optimal assignment $\mathbf{x}^\star$, we have $F^\star = F(\mathbf{x}^\star) = \max\{F(x), F(\bar{x})\}$, for any variable $X$.

Consider now the notion of an *equivalence* constraint:

$$(X \equiv Y, \infty) \quad \overset{def}{=} \quad \{(x \vee \bar{y}, \infty), (\bar{x} \vee y, \infty)\}$$

that is a hard constraint that asserts that $X$ and $Y$ should take the same sign. We consider in this paper the *relaxation* of Max-SAT instances that result from removing equivalence constraints. Clearly, when we remove an equivalence constraint from a Max-SAT instance $f$, more clauses can become satisfied and the resulting optimal value will be an upper bound on the original value $F^\star$.

It is straightforward to augment a Max-SAT instance, weighted or not, to an equivalent one where equivalence constraints can be relaxed. Consider, e.g.,

$$\{(a \vee b, w_1), (\bar{b} \vee c, w_2), (\bar{c} \vee d, w_3)\}.$$

We can replace the variable $C$ appearing as a literal in the third clause with a clone variable $C'$, and add an equivalence constraint $C \equiv C'$, giving us:

$$\{(a \vee b, w_1), (\bar{b} \vee c, w_2), (\bar{c}' \vee d, w_3), (c \vee \bar{c}', \infty), (\bar{c} \vee c', \infty)\}$$

which is equivalent to the original in that an assignment $\mathbf{x}$ of the original formula corresponds to an assignment $\mathbf{x}'$ in the augmented formula, and vice-versa, where the assignment $\mathbf{x}'$ sets the variable and its clone to the same sign. Moreover, the assignment $\mathbf{x}$ satisfies the same clauses in the original instance that assignment $\mathbf{x}'$ satisfies in the augmented instance (minus the equivalence constraint), and vice-versa. In this particular example, when we remove the equivalence constraint $C \equiv C'$, we have a relaxed formula composed of two independent subproblems: $\{(a \vee b, w_1), (\bar{b} \vee c, w_2)\}$ and $\{(\bar{c}' \vee d, w_3)\}$.

A number of structural relaxations can be reduced to the removal of equivalence constraints, including variable splitting [1, 5, 6], variable relabeling [2], and mini-bucket approximations [7, 5]. In particular, these relaxations, which ignore variables shared among different clauses, can be restored by adding equivalence constraints.

## 3 Compensating for Relaxations

Suppose that we have simplified a Max-SAT instance $f$ by relaxing equivalence constraints, resulting in a simpler instance $h$. Our goal now is to identify a Max-SAT instance $g$ that is as tractable as the relaxed instance $h$, but is a tighter approximation of the original instance $f$.

We propose that we construct a new instance $g$ by introducing auxiliary clauses into the relaxation. More specifically, for each equivalence constraint

$X \equiv Y$ relaxed from the original instance $f$, we introduce four unit clauses, with four auxiliary weights:

$$\{(x, w_x), (\bar{x}, w_{\bar{x}}), (y, w_y), (\bar{y}, w_{\bar{y}})\}.$$

Note that adding unit clauses to an instance does not impact significantly the complexity of solving it, in that the addition does not increase the treewidth of an instance. Thus, adding unit clauses does not increase the complexity of Max-SAT algorithms that are exponential only in treewidth [8]. Our goal now is to specify the auxiliary weights $w_x, w_{\bar{x}}, w_y, w_{\bar{y}}$ so that they compensate for the equivalence constraints relaxed, by restoring a weaker notion of equivalence.

### 3.1 Intuitions: A Simplified Case

In this section, we describe our proposal in the simpler case where a *single* equivalence constraint $X \equiv Y$ is removed. We shall make some claims without justification, as they follow as corollaries of more general results in later sections.

Say we remove a single equivalence constraint $X \equiv Y$ from $f$, resulting in a relaxation $h$. In any optimal assignment for instance $f$, variables $X$ and $Y$ are set to the same sign, because of the equivalence constraint $X \equiv Y$. If a Max-SAT assignment $\mathbf{x}^\star$ for the relaxation $h$ happens to set variables $X$ and $Y$ to the same sign, then we know that $\mathbf{x}^\star$ is also a Max-SAT assignment for instance $f$. However, an optimal assignment for the relaxation $h$ may set variables $X$ and $Y$ to different signs, thus the Max-SAT value $H^\star$ of the relaxation $h$ is only an upper bound on the Max-SAT value $F^\star$ of the original instance $f$. The goal then is to set the weights $w_x, w_{\bar{x}}$ on $X$ and $w_y, w_{\bar{y}}$ on $Y$ to correct for this effect.

Consider first, in the relaxation, the values of $H(x)$ and $H(y)$, the Max-SAT values assuming that a variable $X$ is set to a value $x$, and separately, that variable $Y$ is set to a value $y$. If $H(x) \neq H(y)$ and $H(\bar{x}) \neq H(\bar{y})$, then we know that a Max-SAT assignment for the relaxation $h$ sets $X$ and $Y$ to different signs: the Max-SAT value assuming $X$ is set to $x$ is not the same as the Max-SAT value assuming $Y$ is set to $y$ (and similarly for $\bar{x}$ and $\bar{y}$). Thus, we may want to set the weights $w_x, w_{\bar{x}}, w_y, w_{\bar{y}}$ so that $G(x) = G(y)$ and $G(\bar{x}) = G(\bar{y})$ in the compensation $g$, so that if there is a Max-SAT assignment that sets $X$ to $x$, there is at least a Max-SAT assignment that also sets $Y$ to $y$, even if there is no Max-SAT assignment setting both $X$ and $Y$ to the same sign at the same time.

We thus propose the following weaker notion of equivalence to be satisfied in a compensation $g$, to makeup for the loss of an equivalence constraint $X \equiv Y$:

$$\frac{1}{2}\begin{bmatrix} G(x) \\ G(\bar{x}) \end{bmatrix} = \frac{1}{2}\begin{bmatrix} G(y) \\ G(\bar{y}) \end{bmatrix} = \begin{bmatrix} w_x + w_y \\ w_{\bar{x}} + w_{\bar{y}} \end{bmatrix} \tag{1}$$

Before we discuss this particular choice of weights, consider the following properties of the resulting compensation $g$. First, we can identify when a compensation $g$ satisfying Equation 1 yields exact results, just as we can with a relaxation $h$. In particular, if $\mathbf{x}^\star$ is an optimal assignment for $g$ that sets the variables $X$ and $Y$ to the same sign, then: (1) assignment $\mathbf{x}^\star$ is also optimal for instance $f$; and

(2) $\frac{1}{2}G(\mathbf{x}^\star) = F(\mathbf{x}^\star)$. Moreover, a compensation $g$ yields an upper bound that is *tighter* than the one given by the relaxation $h$:

$$F^\star \leq \frac{1}{2}G^\star \leq H^\star.$$

See Corollary 1 in the Appendix, for details.

To justify the weights we chose in Equation 1, consider first the following two properties, which lead to a notion of an ideal compensation. First, say that a compensation $g$ has *valid configurations* if:

$$G(x) = G(y) = G(x,y) \qquad \text{and} \qquad G(\bar{x}) = G(\bar{y}) = G(\bar{x},\bar{y}),$$

i.e., Max-SAT assignments that set $X$ to a sign $x$ also set $Y$ to the same sign $y$, and vice versa; analogously if $X$ is set to $\bar{x}$ or $Y$ is set to $\bar{y}$. Second, say that a compensation $g$ has *scaled values* if the optimal value of a valid configuration is proportional to its value in the original instance $f$, i.e., $G(x,y) = \kappa F(x,y)$ and $G(\bar{x},\bar{y}) = \kappa F(\bar{x},\bar{y})$ for some $\kappa > 1$. We then say that a compensation $g$ is *ideal* if it has valid configurations and scaled values. At least for finding Max-SAT solutions, an ideal compensation $g$ is just as good as actually having the equivalence constraint $X \equiv Y$. The following tells us that for any possible choice of weights, if the compensation is ideal then it must also satisfy Equation 1.

**Proposition 1.** *Let $f$ be a weighted Max-SAT instance and let $g$ be a compensation that results from relaxing a single equivalence constraints $X \equiv Y$ in $f$. If $g$ has valid configurations and scaled values, with $\kappa = 2$, it also satisfies Eq. 1.*

Although a compensation satisfying Equation 1 may not always be ideal, it at least results in a meaningful approximation that is tighter than a relaxation. Note that we could have chosen a different value of $\kappa$, leading to equations slightly different from Equation 1, although the resulting approximation would be effectively the same. Moreover, the choice $\kappa = 2$ leads to simplified semantics, e.g., in the ideal case we can recover the exact values from the weights alone: $w_x + w_y = F(x,y)$ and $w_{\bar{x}} + w_{\bar{y}} = F(\bar{x},\bar{y})$.

### 3.2 An Example

Consider the following weighted Max-SAT instance $f$ with a single equivalence constraint $X \equiv Y$:

$$\begin{array}{ll} f: & (x \vee \bar{z}, 12)\ (y \vee \bar{z}, 6)\ (z, 30)\ (x \vee \bar{y}, \infty) \\ & (\bar{x} \vee \bar{z}, 3)\ \ (\bar{y} \vee \bar{z}, 9) \qquad\quad\ (\bar{x} \vee y, \infty) \end{array}$$

which has a unique optimal Max-SAT assignment $\mathbf{x}^\star = \{X{=}x, Y{=}y, Z{=}z\}$, with Max-SAT value $F(\mathbf{x}^\star) = 12 + 6 + 30 = 48$. When we relax the equivalence constraint $X \equiv Y$, we arrive at a simpler instance $h$:

$$\begin{array}{ll} h: & (x \vee \bar{z}, 12)\ (y \vee \bar{z}, 6)\ (z, 30) \\ & (\bar{x} \vee \bar{z}, 3)\ \ (\bar{y} \vee \bar{z}, 9) \end{array}$$

The relaxation $h$ has a different optimal assignment $\mathbf{x}^\star = \{X{=}x, Y{=}\bar{y}, Z{=}z\}$, where variables $X$ and $Y$ are set to different signs. The optimal value is now $H(\mathbf{x}^\star) = 12 + 9 + 30 = 51$ which is greater than the value 48 for the original instance $f$. Now consider a compensation $g$ with auxiliary unit clauses:

$$\begin{array}{lllll} g: & (x \vee \bar{z}, 12) & (y \vee \bar{z}, 6) & (z, 30) & (x, 27) \; (y, 21) \\ & (\bar{x} \vee \bar{z}, 3) & (\bar{y} \vee \bar{z}, 9) & & (\bar{x}, 20) \; (\bar{y}, 26) \end{array}$$

This compensation $g$ satisfies Equation 1, as:

$$\frac{1}{2}\begin{bmatrix} G(x) \\ G(\bar{x}) \end{bmatrix} = \frac{1}{2}\begin{bmatrix} G(y) \\ G(\bar{y}) \end{bmatrix} = \begin{bmatrix} w_x + w_y \\ w_{\bar{x}} + w_{\bar{y}} \end{bmatrix} = \begin{bmatrix} 48 \\ 46 \end{bmatrix}$$

The compensation $g$ has an optimal assignment $\mathbf{x}^\star = \{X{=}x, Y{=}y, Z{=}z\}$, the same as for the original instance $f$. It also has Max-SAT value $G(\mathbf{x}^\star) = 12 + 6 + 30 + 27 + 21 = 96$, where $\frac{1}{2}G(\mathbf{x}^\star) = F(\mathbf{x}^\star) = 48$.

Note that in this example, the weights happened to be integral, although in general, the weights of a compensation may be real-valued.

### 3.3 Compensations and their Properties

In this section, we define compensations for the general case when multiple equivalence constraints are removed. Moreover, we formalize some of the properties we highlighted in the previous section.

Say then that we relax $k$ equivalence constraints $X \equiv Y$. We seek a compensation $g$ whose weights satisfy the condition:

$$\frac{1}{1+k}\begin{bmatrix} G(x) \\ G(\bar{x}) \end{bmatrix} = \frac{1}{1+k}\begin{bmatrix} G(y) \\ G(\bar{y}) \end{bmatrix} = \begin{bmatrix} w_x + w_y \\ w_{\bar{x}} + w_{\bar{y}} \end{bmatrix} \tag{2}$$

for each equivalence constraint $X \equiv Y$ relaxed. If a compensation $g$ does indeed satisfy this condition, then it is possible to determine, in certain cases, when the optimal solution for a compensation is also optimal for the original instance $f$.

**Theorem 1.** *Let $f$ be a weighted Max-SAT instance and let $g$ be the compensation that results from relaxing $k$ equivalence constraints $X \equiv Y$ in $f$. If the compensation $g$ satisfies Equation 2, and if $\mathbf{x}^\star$ is an optimal assignment for $g$ that assigns the same sign to variables $X$ and $Y$, for each equivalence constraint $X \equiv Y$ relaxed, then:*

- *assignment $\mathbf{x}^\star$ is also optimal for instance $f$; and*
- $\frac{1}{1+k}G(\mathbf{x}^\star) = F(\mathbf{x}^\star)$.

Moreover, the Max-SAT value of a compensation $g$ is an upper bound on the Max-SAT value of the original instance $f$.

**Theorem 2.** *Let $f$ be a weighted Max-SAT instance and let $g$ be the compensation that results from relaxing $k$ equivalence constraints $X \equiv Y$ in $f$. If the compensation $g$ satisfies Equation 2, then:* $F^\star \leq \frac{1}{1+k}G^\star$

We remark now that a relaxation alone has analogous properties. If an assignment $\mathbf{x}^\star$ is optimal for a relaxation $h$, and it is also a valid assignment for instance $f$ (i.e., it does not violate the equivalence constraints $X \equiv Y$), then $\mathbf{x}^\star$ is also optimal for $f$, where $H(\mathbf{x}^\star) = F(\mathbf{x}^\star)$ (since they satisfy the same clauses). Otherwise, the Max-SAT value of a relaxation is an upper bound on the Max-SAT value of the original instance $f$. On the other hand, compensations are tighter approximations than the corresponding relaxation, at least in the case when a single equivalence constraint is relaxed: $F^\star \leq \frac{1}{2}G^\star \leq H^\star$. Although we leave this point open in the case where multiple equivalence constraints are relaxed, we have at least found empirically that compensations are never worse than relaxations. We discuss this point further in the following section.

The following theorem has implications for weighted Max-SAT solvers, such as [1, 2], that rely on relaxations for upper bounds.

**Theorem 3.** *Let $f$ be a weighted Max-SAT instance and let $g$ be the compensation that results from relaxing $k$ equivalence constraints $X \equiv Y$ in $f$. If compensation $g$ satisfies Equation 2, and if $\tilde{\mathbf{z}}$ is a partial assignment that sets the same sign to variables $X$ and $Y$, for any equivalence constraint $X \equiv Y$ relaxed, then: $F(\tilde{\mathbf{z}}) \leq \frac{1}{1+k}G(\tilde{\mathbf{z}})$*

Solvers, such as those in [1, 2], perform a depth-first brand-and-bound search to find an optimal Max-SAT solution. They rely on upper bounds of a Max-SAT solution, under partial assignments, in order to prune the search space. Thus, any method capable of providing upper bounds tighter than those of a relaxation, can potentially have an impact in the performance of a branch-and-bound solver.

### 3.4 Searching for Weights

We now address the question: how do we actually find weights so that a compensation will satisfy Equation 2? Consider the simpler situation where we want weights for one particular equivalence constraint $X \equiv Y$. Ignoring the presence of other equivalence constraints that may have been relaxed, we can think of a compensation $g$ as a compensation where only the single equivalence constraint $X \equiv Y$ being considered has been removed. The corresponding “relaxation” is found by simply removing the weights $w_x, w_{\bar{x}}, w_y, w_{\bar{y}}$ from $g$, for the single equivalence constraint $X \equiv Y$. More specifically, let $H_{x,y} = G(x, y) - [w_x + w_y]$ denote the Max-SAT value of the “relaxation,” assuming that $X$ and $Y$ are set to $x$ and $y$ (and similarly for other configurations of $X$ and $Y$). Given this “relaxation,” we have a closed form solution for the weights, for a compensation $g$ to satisfy Equation 2, at least for the one equivalence constraint $X \equiv Y$ being considered.

**Theorem 4.** *Let $f$ be a weighted Max-SAT instance, let $g$ be the compensation that results from relaxing $k$ equivalence constraints in $f$, and let $X \equiv Y$ be one of $k$ equivalence constraints relaxed. Suppose, w.l.o.g., that $H_{x,y} \geq H_{\bar{x},\bar{y}}$, and let:*

$$G^+ = \frac{1+k}{k} \max\left\{H_{x,y}, \frac{1}{2}[H_{x,\bar{y}} + H_{\bar{x},y}]\right\} \tag{3}$$

$$G^- = \frac{1+k}{k} \max\left\{H_{\bar{x},\bar{y}}, \frac{1}{1+2k}[H_{x,y} + kH_{x,\bar{y}} + kH_{\bar{x},y}], \frac{1}{2}[H_{x,\bar{y}} + H_{\bar{x},y}]\right\} \tag{4}$$

**Algorithm 1** RelaxEq-and-Compensate (REC)

```
input: a weighted Max-SAT instance $f$ with $k$ equivalence constraints $X \equiv Y$
output: a compensation $g$ satisfying Equation 2
main:
$h \leftarrow$ result of relaxing all $X \equiv Y$ in $f$
$g \leftarrow$ result of adding to $h$ weights $w_x, w_{\bar{x}}, w_y, w_{\bar{y}}$ for each $X \equiv Y$
initialize all weights $w_x, w_{\bar{x}}, w_y, w_{\bar{y}}$, say to $\frac{1}{2}H^\star$.
while weights have not converged do
    for each equivalence constraint $X \equiv Y$ removed do
        update weights $w_x, w_{\bar{x}}, w_y, w_{\bar{y}}$ according to Equations 5 & 6
return $g$
```

*If we set the weights for equivalence constraints $X \equiv Y$ to:*

$$\begin{bmatrix} w_x \\ w_{\bar{x}} \end{bmatrix} = \frac{1}{2}\frac{1}{1+k}\begin{bmatrix} G^+ \\ G^- \end{bmatrix} + \frac{1}{4}\begin{bmatrix} H_{\bar{x},y} - H_{x,\bar{y}} \\ H_{x,\bar{y}} - H_{\bar{x},y} \end{bmatrix} \tag{5}$$

$$\begin{bmatrix} w_y \\ w_{\bar{y}} \end{bmatrix} = \frac{1}{2}\frac{1}{1+k}\begin{bmatrix} G^+ \\ G^- \end{bmatrix} + \frac{1}{4}\begin{bmatrix} H_{x,\bar{y}} - H_{\bar{x},y} \\ H_{\bar{x},y} - H_{x,\bar{y}} \end{bmatrix} \tag{6}$$

*then equivalence constraint $X \equiv Y$ will satisfy Equation 2 in compensation $g$.*

When the original instance $f$ has been sufficiently relaxed, and enough equivalence constraints removed, then we will be able to compute the quantities $H_{x,y}$ efficiently (we discuss this point further in the following section). Theorem 4 then suggests an iterative algorithm for finding weights that satisfy Equation 2 for all $k$ equivalence constraints relaxed, which is summarized in Algorithm 1.

This algorithm, which we call RelaxEq-and-Compensate (REC), initializes the weights of a given compensation $g$ to some value, and iterates over equivalence constraints one-by-one. When it arrives at a particular equivalence constraint $X \equiv Y$, REC sets the weights according to Equations 5 & 6, which results in a compensation satisfying Equation 2, at least for that particular equivalence constraint. REC does the same for the next equivalence constraint, which may cause the previous equivalence constraint to no longer satisfy Equation 2. REC continues, however, until the weights of all equivalence constraints do not change with the application of Equations 5 & 6 (to some constant $\epsilon$), at which point all equivalence constraints satisfy Equation 2.

We now make a few observations. First, if we set all weights $w_x, w_{\bar{x}}, w_y, w_{\bar{y}}$ of an initial compensation $g_0$ to $\frac{1}{2}H^\star$, then the initial approximation to the Max-SAT value is $\frac{1}{1+k}G_0^\star = H^\star$.[2] That is, the initial approximation is the same as the upper bound given by the relaxation $h$. We have observed empirically, interestingly enough, that when we start with these initial weights, every iteration of the REC algorithm results in a compensation $g$ where the value of $\frac{1}{1+k}G^\star$ is no larger than that of the previous iteration. Theorem 2 tells us that a compensation $g$

[2] $G_0^\star = \max_{\mathbf{x}} G_0(\mathbf{x}) = \max_{\mathbf{x}}[H(\mathbf{x}) + \sum_{X \equiv Y} w_x + w_y]$
$= \max_{\mathbf{x}}[H(\mathbf{x}) + \sum_{X \equiv Y} \frac{1}{2}H^\star + \frac{1}{2}H^\star] = \max_{\mathbf{x}}[H(\mathbf{x})] + kH^\star = H^\star + kH^\star$

satisfying Equation 2, which REC is searching for, yields a Max-SAT value $\frac{1}{1+k}G^\star$ that is an upper bound on the Max-SAT value $F^\star$ of the original instance $f$. This would imply that algorithm REC tends to provide monotonically decreasing upper bounds on $F^\star$, when starting with an initial compensation equivalent to the relaxation $h$. This would imply that, at least empirically, the value of $\frac{1}{1+k}G^\star$ is convergent in the REC algorithm. We have observed empirically that this is the case, and we discuss these points further in Section 4.

### 3.5 Knowledge Compilation

One point that we have yet to address is how to efficiently compute the values $H_{x,y}$ that are required by the iterative algorithm REC that we have proposed. In principle, any off-the-shelf Max-SAT solver could be used, where we repeatedly solve Max-SAT instances $g$ where the variables $X$ and $Y$ are set to some values. However, when we remove $k$ equivalence constraints, REC requires us to solve $4k$ Max-SAT instances in each iteration.

If, however, we relax enough equivalence constraints so that the treewidth is small enough, we can efficiently compile a given Max-SAT instance in CNF into decomposable negation normal form (DNNF) [9–12] (for details on how to solve weighted Max-SAT problems by compilation to DNNF, see [1, 13]). Once our simplified instance $g$ is in DNNF, many queries can be performed in time linear in the compilation, which includes computing at once all of the values $G(x), G(\bar{x})$ and $G(y), G(\bar{y})$, as well as the Max-SAT value $G^\star$. Computing each value $H_{x,y}$ can also be performed in time linear in the compilation, although lazy evaluation can be used to improve efficiency; see [1] for details.

We note that the required values can also be computed by message-passing algorithms such as belief propagation [14]. However, this would typically involve converting a weighted Max-SAT instance into an equivalent Bayesian network or factor graph, where general-purpose algorithms do not make use of the kinds of techniques that SAT solvers and compilers are able to. In contrast, knowledge compilation can be applied to solving Bayesian network tasks beyond the reach of traditional probabilistic inference algorithms [15].

## 4 Experiments

We evaluate here the REC algorithm on a selection of benchmarks. Our goals are: (1) to illustrate its empirical properties, (2) to demonstrate that compensations can improve, to varying extents, relaxations, and (3) that compensations are able to improve branch-and-bound solvers, such as CLONE, at least in some cases.

The relaxations that we consider are the same as those employed by the CLONE solver [1],[3] which in certain categories led, or was otherwise competitive with, the solvers evaluated in the 3rd Max-SAT evaluation.[4] CLONE relaxes a

[3] Available at `http://reasoning.cs.ucla.edu/clone/`

[4] Results of the evaluation are available at `http://www.maxsat.udl.cat/08/`

given Max-SAT instance by choosing, heuristically, a small set of variables to "split", where splitting a variable $X$ simplifies a Max-SAT instance by replacing each occurrence of a variable $X$ with a unique clone variable $Y$. This relaxation effectively ignores the dependence that different clauses have on each other due to the variable $X$ being split. Note that such a relaxation is restored when we assert equivalence constraints $X \equiv Y$. For our purposes, we can then assume that equivalence constraints were instead relaxed.

Like CLONE, we relax Max-SAT instances until their treewidth is at most 8. Given this relaxation, we then constructed a compensation which was compiled into DNNF by the C2D compiler.[5] For each instance we selected, we ran the REC algorithm for at most 2000 iterations, and iterated over equivalence constraints in some fixed order, which we did not try to optimize. If the change in the weights from one iteration to the next was within $10^{-4}$, we declared convergence, and stopped.

Our first set of experiments were performed on randomly parametrized grid models, which are related to the Ising and spin-glass models studied in statistical physics; see, e.g., [16]. This type of model is also commonly used in fields such as computer vision [17]. In these models, we typically seek a configuration of variables $\mathbf{x}$ minimizing a cost (energy) function of the form:

$$F(\mathbf{x}) = \sum_i \psi_i(x_i) + \sum_{ij} \psi_{ij}(x_i, x_j)$$

where variables $X_i$ are arranged in an $n \times n$ grid, which interact via potentials $\psi_{ij}$ over neighboring variables $X_i$ and $X_j$. In our experiments, we left all $\psi_i(x_i) = 0$, and we generated each $\psi_{ij}(x_i, x_j) = -\log p$ with $p$ drawn uniformly from $(0, 1)$. These types of models are easily reduced to weighted Max-SAT; see, e.g., [18]. Note, that the resulting weights will be floating-point values, which are not yet commonly supported by modern Max-SAT solvers. We thus focus our empirical evaluation with respect to a version of CLONE that was augmented for us to accommodate such weights.

We generated 10 randomly parametrized $16 \times 16$ grid instances and evaluated (1) the dynamics of the REC algorithm, and (2) the quality of the resulting approximation (although we restrict our attention here to 10 instances, for simplicity, the results we present are typical for this class of problems). Consider first Figure 1, where we plotted the quality of an approximation ($y$-axis) versus iterations of the REC algorithm ($x$-axis), for each of the 10 instances evaluated. We define the quality of an approximation as the error of the compensation $\frac{1}{1+k}G^\star - F^\star$, relative to the error of the relaxation $H^\star - F^\star$. That is, we measured the error

$$E = \frac{\frac{1}{1+k}G^\star - F^\star}{H^\star - F^\star}$$

which is zero when the compensation is exact, and one when the compensation is equivalent to the relaxation. Remember that we proposed to initialize the

[5] Available at `http://reasoning.cs.ucla.edu/c2d/`.

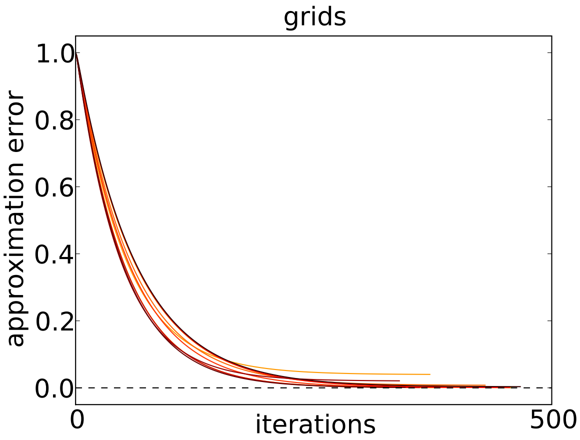

**Fig. 1.** Behavior of the REC algorithm random 16 × 16 grid instances. Note that color is used here to help differentiate plots, and is otherwise not meaningful.

REC algorithm with weights that led to an initial compensation with an optimal value $\frac{1}{1+k}G_0^\star = H^\star$. Thus, we think of the error $E$ as the degree to which the compensation is able to tighten the relaxation.

We make some observations about the instances depicted in Figure 1. First, all of the 10 instances converged before 500 iterations. Next, we see that the REC algorithm yields from iteration-to-iteration errors, and hence values $\frac{1}{1+k}G^\star$, that are monotonically decreasing. If this is the case in general, then this implies that these bounds $\frac{1}{1+k}G^\star$ are convergent in the REC algorithm, since a compensation satisfying Equation 2 is an upper bound on $F^\star$ (by Theorem 2). This implies, at least empirically, that the REC algorithm is indeed tightening a relaxation from iteration-to-iteration. Finally, we find that REC is capable of significantly improving the quality of an approximation, to exact or near-exact levels.

Given such improvement, we may well expect that a solver that relies on relaxations for upper bounds, such as CLONE, may benefit from an improved approximation that provides tighter bounds. In fact, using the relaxation alone, CLONE was unable to solve any of these instances, given a time limit of 10 minutes.[6] We thus augmented CLONE so that it can take advantage of the tighter REC approximation. In particular, we compensate for the relaxation that CLONE would normally use, and have CLONE use its tighter upper-bounds to prune nodes during branch-and-bound search.

This augmented CLONE algorithm now has one additional steps. Before we perform the branch-and-bound search (the CLONE step), we must first compensate for the relaxation (the REC step). The following table records the time, in seconds, to perform each step in the instances we considered:

[6] Experiments were performed on an Intel Xeon E5440 CPU, at 2.83GHz.

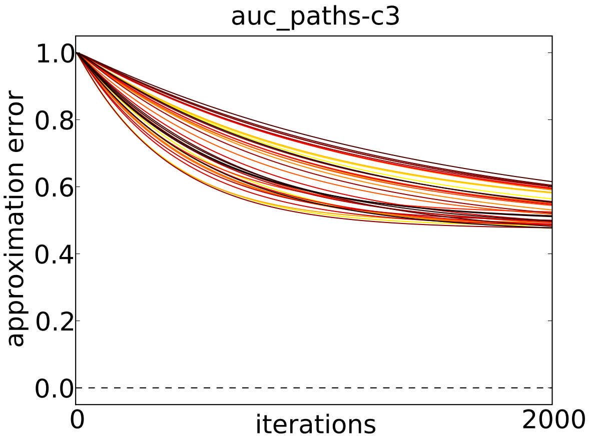

**Fig. 2.** Behavior of the REC algorithm in weighted partial Max-SAT instances (AUC_PATHS benchmarks).

| instance | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| REC | 303 | 302 | 583 | 390 | 308 | 326 | 318 | 249 | 311 | 511 |
| CLONE | 528 | 1 | 45 | 2 | 253 | 121 | 47 | 14 | 12 | 10 |
| total | 831 | 303 | 628 | 392 | 561 | 447 | 365 | 263 | 323 | 521 |

Although CLONE was unable to solve any of these instances within 600 seconds with a relaxation alone, it was able to solve most instances within the same amount of time when enabled with a compensation. We further remark that there is ample room for improving the efficiency of our REC implementation, which is in Java and Jython (Jython is an implementation of the Python language in Java).

Finally, in Figure 2, we plot the performance of the REC algorithm on a subset of the AUC_PATHS benchmark from the weighted partial Max-SAT instances from the 2008 evaluation.[7] We find here that the REC algorithm is able to reduce the approximation error of a relaxation by roughly half in many instances, or otherwise appears to approach this level if allowed further iterations. We also see again that the REC has relatively stable dynamics. As CLONE was already able to efficiently solve all of the instances in the AUC_PATHS benchmark, CLONE did not benefit much from a compensation in this case.

## 5 Conclusion

In this paper, we proposed a new perspective on approximations of Max-SAT problems, that is based on relaxing a given instance, and then compensating for

[7] The instances selected were the 40 instances labeled cat_paths_60_$p$_*.txt.wcnf for $p \in \{100, 110, 120, 130, 140\}$.

the relaxation. When we relax equivalence constraints in a Max-SAT problem, we can perform inference on the simplified problem, identify some of its defects, and then recover a weaker notion of equivalence. We proposed a new algorithm, REC, that iterates over equivalence constraints, compensating for relaxations one-by-one. Our empirical results show that REC can tighten relaxations to the point of recovering exact or near-exact results, in some cases. We have also observed that, in some cases, these compensations can be used by branch-and-bound solvers to find optimal Max-SAT solutions, which they were unable to find with a relaxation alone.

## Acknowledgments

This work has been partially supported by NSF grant #IIS-0713166.

## A Proofs

*Proof (of Proposition 1).* First, observe that:

$$G(x,y) = F(x,y) + [w_x + w_y] = \frac{1}{2}G(x,y) + [w_x + w_y]$$

since $g$ has scaled values. Thus, $\frac{1}{2}G(x,y) = w_x + w_y$; similarly to show $\frac{1}{2}G(\bar{x},\bar{y}) = w_{\bar{x}} + w_{\bar{y}}$. Since $g$ has valid configurations, then $g$ also satisfies Equation 1. □

In the remainder of this section, we call a complete assignment $\mathbf{x}$ *valid* iff $\mathbf{x}$ sets the same sign to variables $X$ and $Y$, for every equivalence constraint $X \equiv Y$ removed from an instance $f$; analogously, for partial assignments $\mathbf{z}$. We will also use $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{z}}$ to denote valid assignments, when appropriate.

**Lemma 1.** *Let $f$ be a weighted Max-SAT instance and let $g$ be the compensation that results from relaxing $k$ equivalence constraints $X \equiv Y$ in $f$. If compensation $g$ satisfies Equation 2, and if $\tilde{\mathbf{x}}$ is a complete assignment that is also valid, then $F(\tilde{\mathbf{x}}) \leq \frac{1}{1+k}G(\tilde{\mathbf{x}})$, with equality if $\tilde{\mathbf{x}}$ is also optimal for $g$.*

*Proof.* When we decompose $G(\tilde{\mathbf{x}})$ into the original weights, i.e., $F(\tilde{\mathbf{x}})$, and the auxiliary weights $w_x + w_y = \frac{1}{1+k}G(x)$ (by Equation 2) we have

$$G(\tilde{\mathbf{x}}) = F(\tilde{\mathbf{x}}) + \sum_{X \equiv Y} [w_x + w_y] = F(\tilde{\mathbf{x}}) + \sum_{X \equiv Y} \frac{1}{1+k} G(x).$$

Note that $x$ is the value assumed by $X$ in assignment $\tilde{\mathbf{x}}$. Since $G(x) \geq G(\tilde{\mathbf{x}})$,

$$G(\tilde{\mathbf{x}}) \geq F(\tilde{\mathbf{x}}) + \sum_{X \equiv Y} \frac{1}{1+k} G(\tilde{\mathbf{x}}) = F(\tilde{\mathbf{x}}) + \frac{k}{1+k} G(\tilde{\mathbf{x}})$$

and thus $\frac{1}{1+k}G(\tilde{\mathbf{x}}) \geq F(\tilde{\mathbf{x}})$. In the case where $G(\tilde{\mathbf{x}}) = G^\star$, we have $G(x) = G(\tilde{\mathbf{x}})$ for all $X \equiv Y$, so we have the equality $F(\tilde{\mathbf{x}}) = \frac{1}{1+k}G(\tilde{\mathbf{x}})$. □

*Proof (of Theorem 1).* Since $\mathbf{x}^\star$ is both optimal and valid, we know that $\frac{1}{1+k}G^\star = \frac{1}{1+k}G(\mathbf{x}^\star) = F(\mathbf{x}^\star)$, by Lemma 1. To show that $\mathbf{x}^\star$ is also optimal for the original instance $f$, note first that $G^\star = \max_{\mathbf{x}} G(\mathbf{x}) = \max_{\tilde{\mathbf{x}}} G(\tilde{\mathbf{x}})$. Then:

$$F(\mathbf{x}^\star) = \frac{1}{1+k}G^\star = \max_{\tilde{\mathbf{x}}} \frac{1}{1+k}G(\tilde{\mathbf{x}}) \geq \max_{\tilde{\mathbf{x}}} F(\tilde{\mathbf{x}}) = F^\star$$

using again Lemma 1. We can thus infer that $F(\mathbf{x}^\star) = F^\star$. □

*Proof (of Theorem 2).* Let $\mathbf{x}^\star$ be an optimal assignment for $f$. Since $\mathbf{x}^\star$ must also be valid, we have by Lemma 1 that

$$F^\star = F(\mathbf{x}^\star) \leq \frac{1}{1+k}G(\mathbf{x}^\star) \leq \frac{1}{1+k}G^\star$$

as desired. □

*Proof (of Theorem 3).* We have that

$$F(\tilde{\mathbf{z}}) = \max_{\tilde{\mathbf{x}}\sim\tilde{\mathbf{z}}} F(\tilde{\mathbf{x}}) \leq \max_{\tilde{\mathbf{x}}\sim\tilde{\mathbf{z}}} \frac{1}{1+k}G(\tilde{\mathbf{x}}) \leq \max_{\mathbf{x}\sim\tilde{\mathbf{z}}} \frac{1}{1+k}G(\mathbf{x}) = \frac{1}{1+k}G(\tilde{\mathbf{z}})$$

where the first inequality follows from Lemma 1. □

Proof of Theorem 4 appears in the Appendix of the full report [3].

**Corollary 1.** *Let $f$ be a weighted Max-SAT instance and let $g$ be the compensation that results from relaxing a single equivalence constraint $X \equiv Y$ in $f$. If compensation $g$ satisfies Equation 1, then*

$$F^\star \leq \frac{1}{2}G^\star \leq H^\star.$$

*Proof.* The first inequality follows from Theorem 2. From the proof of Theorem 4 we know that either $\frac{1}{2}G^\star = F^\star \leq H^\star$ or

$$\frac{1}{2}G^\star = \frac{1}{2}[H(x,\bar{y}) + H(\bar{x},y)] \leq \max\{H(x,\bar{y}), H(\bar{x},y)\} \leq H^\star$$

thus we have the second inequality as well. □

## References

1. Pipatsrisawat, K., Palyan, A., Chavira, M., Choi, A., Darwiche, A.: Solving weighted Max-SAT problems in a reduced search space: A performance analysis. Journal on Satisfiability, Boolean Modeling, and Computation **4** (2008) 191–217
2. Ramírez, M., Geffner, H.: Structural relaxations by variable renaming and their compilation for solving MinCostSAT. In: CP. (2007) 605–619
3. Choi, A., Standley, T., Darwiche, A.: Approximating weighted Max-SAT problems by compensating for relaxations. Technical report, CSD, UCLA (2009)
4. Li, C.M., Manyà, F.: MaxSAT, hard and soft constraints. In Biere, A., Heule, M., van Maaren, H., Walsh, T., eds.: Handbook of Satisfiability. IOS Press (2009) 613–631
5. Choi, A., Chavira, M., Darwiche, A.: Node splitting: A scheme for generating upper bounds in Bayesian networks. In: UAI. (2007) 57–66
6. Siddiqi, S., Huang, J.: Variable and value ordering for MPE search. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence. (2009) To appear.
7. Dechter, R., Rish, I.: Mini-buckets: A general scheme for bounded inference. J. ACM **50**(2) (2003) 107–153
8. Rish, I., Dechter, R.: Resolution versus search: Two strategies for SAT. J. Autom. Reasoning **24**(1/2) (2000) 225–275
9. Darwiche, A.: Decomposable negation normal form. Journal of the ACM **48**(4) (2001) 608–647
10. Darwiche, A.: On the tractability of counting theory models and its application to belief revision and truth maintenance. Journal of Applied Non-Classical Logics **11**(1-2) (2001) 11–34
11. Darwiche, A.: New advances in compiling CNF to decomposable negational normal form. In: Proceedings of European Conference on Artificial Intelligence. (2004) 328–332
12. Darwiche, A., Marquis, P.: A knowledge compilation map. Journal of Artificial Intelligence Research **17** (2002) 229–264
13. Darwiche, A., Marquis, P.: Compiling propositional weighted bases. Artificial Intelligence **157**(1-2) (2004) 81–113
14. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers, Inc., San Mateo, California (1988)
15. Chavira, M., Darwiche, A.: Compiling Bayesian networks with local structure. In: Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI). (2005) 1306–1312
16. Percus, A., Istrate, G., Moore, C.: Where statistical physics meets computation. In Percus, A., Istrate, G., Moore, C., eds.: Computational Complexity and Statistical Physics. Oxford University Press (2006) 3–24
17. Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M.F., Rother, C.: A comparative study of energy minimization methods for Markov random fields. In: ECCV (2). (2006) 16–29
18. Park, J.D.: Using weighted max-sat engines to solve MPE. In: AAAI/IAAI. (2002) 682–687