Strengthening Lagrangian Bounds for the MAX-SAT Problem
(Extended Abstract)

Paolo Nobili[1], Antonio Sassano[2]

[1]Istituto di Analisi dei Sistemi ed Informatica, CNR-Roma, ITALY.
Email: *nobili@iasi.rm.cnr.it*
[2]Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", ITALY.
Email: *sassano@dis.uniroma1.it*

Abstract

We propose an algorithm for the strengthening of the Generalized Set Covering formulation of the MAX-SAT Problem. Unlike the standard approach, our procedure does not rely on the solution of the linear relaxation to generate the cuts used to strengthen the formulation. On the contrary, it computes a Lagrangian bound and uses the Lagrangian multipliers to guide the generation of cutting planes. Some of the cuts generated by the separation procedure have the additional feature that, once added to the original formulation, cause a strict increase of the lower bound.

Key words: Integer Linear Programming, MAX-SAT Problem, Generalized Set Covering Problem.

1 Introduction

Given a propositional logic formula $\mathcal{F}$ in conjunctive normal form, the well known MAX-SAT problem asks for the maximum number of clauses in $\mathcal{F}$ that can be simultaneously satisfied by a truth assignment to the logical variables. Equivalently, one could consider the minimum number μ of clauses that must be removed from $\mathcal{F}$ in order to make it satisfiable. Of course, $\mathcal{F}$ is satisfiable if and only if $\mu = 0$, hence the decision problem SAT can be solved as an instance of MAX-SAT. Indeed, the MAX-SAT Problem is much more difficult than SAT and is known to be NP-Hard for classes of formulas for which the corresponding SAT Problem can be solved in linear time (Horn formulas, binary formulas). Several ILP formulations have been proposed for MAX-SAT Problems. One of the most successful formulation regards MAX-SAT as a special case of the Generalized

Set Covering Problem. Such a formulation has the advantage of requiring a small number of variables and constraints, but its linear relaxation tends to be rather weak and needs to be strenghthened by the addition of valid cuts. In this paper we propose to solve MAX-SAT problems as Generalized Set Covering problems using the strenghthened formulation provided by the *logical completion* of the constraint matrix. Since such a formulation may have a number of constraints which is exponential in the number of original variables/constraints, we propose to use a branch-and-cut algorithm which, at each node of the enumeration tree, solves a strengthened formulation obtained by adding constraints corresponding to rows of the logical completion.

In the standard branch-and-cut algorithm, constraints are added at each node as long as they cut off the fractional solution of the linear relaxation of the current subproblem. Unfortunately, for medium-large size (generalized) set covering problems the solution of the linear relaxation can be quite time consuming. To overcome this difficulty, several authors have proposed to solve a Lagrangian formulation of the set covering problem by means of the Subgradient algorithm. Such an approach has been very successful (both fast and accurate).

However, while a dual solution to the linear relaxation having a value not worse than the Lagrangian bound can be easily computed from the Lagrange multipliers, there is no easy way to compute a primal solution with the same value. As a consequence, if the lower bound is computed by using a Lagrangian formulation, there is no primal solution to guide the generation of cutting planes which can strengthen the current formulation.

The main contribution of this paper is the description of a bounding procedure which uses a Lagrangian formulation to compute the lower bound and exploits the structure of the dual solution to guide the generation of a set of cutting planes. Some of these cutting planes have the nice property that, once added to the original formulation, cause a strict increase of the value of the Lagrangian lower bound.

2 Formulation of MAX-SAT Problems

Let $\mathcal{P} = \{v_1, \ldots, v_h\}$ be a set of propositional variables which can take values in $\{$*true, false*$\}$. A *truth assignment* is a function $v : \mathcal{P} \rightarrow \{$*true, false*$\}$. A *literal* $\tilde{v}_i$ is either a variable $v_i \in \mathcal{P}$ or its *negation* $\bar{v}_i$, where $\bar{v}_i$ is *true* (*false*) if and only if v_i is *false* (*true*). A *clause* C_i is a disjunction of literals

$$C_i = (\tilde{v}_{i_1} \vee \tilde{v}_{i_2} \vee \ldots \vee \tilde{v}_{i_k})$$

regarded, in the following, as the set of its literals. C_i is satisfied by a truth assignment if at least one of its literals is *true* under such assignment. A *propositional formula* $\mathcal{F}$ in disjunctive normal form is a set of clauses. A propositional formula $\mathcal{F}$ is satisfiable if there exists a truth assignment which simultaneously satisfies all the clauses in $\mathcal{F}$. The MAX-SAT problem asks for the maximum number of clauses that can be satisfied by a truth assignment or, equivalently, the minimum number of clauses that must be removed in order to make $\mathcal{F}$ satisfiable. The clause-variable incidence matrix $\hat{A}$ of a formula $\mathcal{F} = \{C_1, \ldots, C_m\}$ is defined by

$$\hat{A}_{ij} = \begin{cases} -1 & \text{if } \bar{v}_j \in C_i \\ +1 & \text{if } v_j \in C_i \\ 0 & \text{otherwise.} \end{cases}$$

Denote by $\nu(\hat{A})$ the vector that gives the number of -1's in each row of $\hat{A}$ and by $\mathbf{1}$ the vector of dimension m whith entries $+1$. It is easy to see that the MAX-SAT Problem can be formulated as

$$\begin{aligned} & \min \sum_{i=1}^{m} y_i && (1) \\ & \hat{A}z + y \geq \mathbf{1} - \nu(\hat{A}) \\ & z \in \{0,1\}^h, y \in \{0,1\}^m. \end{aligned}$$

Now, let x denote the vector $\begin{pmatrix} z \\ y \end{pmatrix}$, let c denote the vector of dimension $n = h + m$ whose first h components are zero and the last m components are 1, and finally, let $A = [\hat{A} \quad I]$, where I is the identity matrix of proper dimension. Then (1) can be rewritten as the following Generalized Set Covering Problem

$$\begin{aligned} & \min c^T x && (2) \\ & Ax \geq \mathbf{1} - \nu(A) \\ & x \in \{0,1\}^n. \end{aligned}$$

In what follows we will say that a matrix A of the above form is *satisfiable* if the minimum of (GSC) is zero, that is, no clause need to be removed from the corresponding propositional formula. Conversely, if the minimum of (GSC) is strictly greater than zero, the matrix will be said *non-satisfiable.* Finally, we will call *minimally non-satisfiable* a matrix A such that it is not satisfiable but any row submatrix of A is satisfiable.

3 Lagrangian Relaxation

Given a generalized Set Covering Problem formulation of the form (2), we consider the following Lagrangian relaxation:

$$\max L(\lambda)$$

where $\lambda \in I\!R^m_+$ and

$$L(\lambda) = \min\{(c^T - \lambda^T A)x + \lambda^T(\mathbf{1} - \nu(A))\}$$

with the minimum taken over $x_i \in \{0,1\}$, $i = 1, \ldots, n$. For any λ, the value of $L(\lambda)$ can be easily computed by inspection and constitutes a valid lower bound for the value of (2). In fact, the maximum of $L(\lambda)$ with $\lambda \in I\!R^m$ coincides with the optimum value of the linear programming relaxation of (2). After the seminal paper of Balas and Ho [2] it has been widely recognized that, in solving (generalized) set covering problems, it is advisable to consider Lagrangian rather than linear programming relaxations of the ILP formulation. This widespread belief is motivated by the ease of computing, by means of the Subgradient algorithm, a tight approssimation of the value of the linear programming optimum in a fraction of the time required by the Simplex method. Several authors have improved upon Balas and Ho's results ([3, 4, 1]) by introducing effective primal and dual heuristics and "ad hoc" modifications of the Subgradient algorithm.

4 A Stronger Formulation

In this section we discuss a way of strengthening the formulation of the generalized set covering problem presented in Section 2. Let C_1 and C_2 be two clauses in $\mathcal{F}$ and v_j be a variable in $\mathcal{P}$ such that $v_j \in C_1$, $\bar{v}_j \in C_2$ and no other variable satisfies the same property. The *resolvent* of C_1 and C_2 is the clause $C_3 = C_1 \cup C_2 \backslash \{v_j, \bar{v}_j\}$. The clauses C_1 and C_2 are said to be the *antecedents* of C_3. Analogously, let e and f be two rows of A such that $e_j = -f_j$ and $e_k f_k \geq 0$ for any $k \neq j$. The row $g = sgn(e+f)$ is said to be the resolvent of e and f, where $sgn(w)$ is the vector defined by

$$sgn(w)_i = \begin{cases} +1 & \text{if } w_i > 0 \\ -1 & \text{if } w_i < 0 \\ 0 & \text{otherwise.} \end{cases}$$

Rows e and f are said to be the *antecedents* of g. The column of index j is said to be the *hinge* of rows e and f. Let $A^\diamond$ be the matrix obtained from A by recursively adding all the resolvents of pairs of its rows. $A^\diamond$ is called the *logical completion* of A. It is easy to see that (2) can also be formulated by replacing A with $A^\diamond$:

$$(GSC^\diamond) \qquad \begin{array}{l} \min c^T x \\ A^\diamond x \geq \mathbf{1} - \nu(A^\diamond) \\ x \in \{0,1\}^n. \end{array}$$

Clearly, the formulation $(GSC^\diamond)$ is stronger than (GSC). In the following we call Q_{LP} the polytope defined by the linear relaxation of (GSC) and $Q^\diamond$ the polytope defined by the linear relaxation of $(GSC^\diamond)$.

In general, $A^\diamond$ has a huge number of rows. As customary in the polyhedral approach, the linear programming relaxation of $(GSC^\diamond)$ can be solved by a dynamic version of the simplex method which adds constraints only when they are violated by the current optimal solution. Hence, at each step of the simplex method, we will be considering only partial formulations of the form

$$
\begin{array}{ll}
(GSC^t) & \min c^T x \\
 & A^t x \geq \mathbf{1} - \nu(A^t) \\
 & x \in \{0,1\}^n .
\end{array}
$$

where A^t is a row submatrix of $A^\diamond$ containing A.

In the next section, we will describe a mechanism for generating "violated" constraints based on the information gathered by solving the Lagrangian dual instead of the linear programming relaxation.

We close this section with a brief discussion of the relationship among $(GSC^\diamond)$ and another strengthening of (GSC) proposed by Gallo et al. [5] in the special case of the *Max Horn Sat Problem.*

Gallo and his coauthors describe their relaxation in the framework of directed hypergraphs which is particularly suited for Horn formulas. Moreover, in the same paper, they prove that their relaxation is stronger than the one defined by the polytope Q_{LP} and allows them to compute an effective lower bound for the Max Horn Sat Problem. We describe here a slight extension of their formulation to the case of general formulas.

Let B be a *minimally non-satisfiable* submatrix of A and let S_B denote the set of indices of the rows of B. It is easy to check that the inequality

$$\sum_{j \in S_B} x_{h+j} \geq 1$$

is satisfied by all the integral solutions of (GSC). Moreover, the following polytope :

$$
\begin{array}{lll}
Q = \{ & x \in \mathbb{R}^n : & \\
 & Ax \geq \mathbf{1} - \nu(A) & \\
 & \sum_{j \in S_B} x_{h+j} \geq 1 & \text{for each minimally non-satisfiable submatrix } B \text{ of } A \\
 & 0 \leq x_j \leq 1 \qquad j = 1, \ldots, n\} &
\end{array}
$$

is contained in Q_{LP} and hence defines a stronger formulation of the MAX-SAT problem.

In [5] it is proved that the polytope P_c defined as:

$$
\begin{array}{lll}
P_c = \{ & x \in \mathbb{R}^m : & \\
 & \sum_{j \in S_B} x_j \geq 1 & \text{for each minimally non-satisfiable submatrix } B \text{ of } A \\
 & 0 \leq x_j \leq 1 \qquad j = 1, \ldots, m\} &
\end{array}
$$

is contained in the projection of Q_{LP} onto the subspace of the variables $\{x_{h+1}, \ldots, x_n\}$. As a consequence, P_c defines a formulation which is stronger than that defined by Q_{LP}.

Observe now that the projection of Q onto the subspace of the variables $\{x_{h+1}, \ldots, x_n\}$ is contained in P_c. Moreover, we have the following theorem.

THEOREM 4.1 *The polytope $Q^\diamond$ is strictly contained in Q.*

$\square$

It follows that $Q^\diamond$ defines a formulation which is stronger than the one defined by P_c.

5 Improving the Lagrangian Lower Bound

The purpose of this section is to describe an effective algorithm to generate inequalities which correspond to rows of $A^\diamond$ and whose addition to a partial formulation (GSC^t) improves the quality of the Lagrangian bound.

To this purpose, assume that we have solved the Lagrangian relaxation of (GSC^t) and let λ^* be the vector of optimal Lagrange multipliers. If we construct a new formulation (GSC^{t+1}) by adding to A^t a new row a_k from $A^\diamond$, we can still regard λ^* as a vector of Lagrange multipliers for the new formulation by letting $\lambda_k^* = 0$. However, in general λ^* is not optimal any more. It is possible that an improved lower bound can be obtained by letting λ_k^* increase, and modifying consequently the other multipliers. Of course, such a situation would be desirable but, unfortunately, it is not easy to characterize its occurence in general. However, the following theorems describe important special cases.

THEOREM 5.1 *Let $a_{i_1}, a_{i_2}, a_{i_3}, a_{i_4}$ be four rows of A^t whose Lagrange multipliers are strictly positive and assume that there exists two columns j and k such that the associated sub-matrix of A^t has the following form:*

$$\begin{array}{c} \\ i_1 \\ i_2 \\ i_3 \\ i_4 \end{array} \begin{array}{c} \begin{array}{cc} \;\; j & \;\;\; k \end{array} \\ \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ -1 & 1 \\ -1 & -1 \end{bmatrix} \end{array} . \tag{3}$$

Moreover, let

$$\delta = \min\{\lambda_{i_1}, \lambda_{i_2}, \lambda_{i_3}, \lambda_{i_4}\}$$

and let $a_{i_5} = sgn(a_{i_1} + a_{i_2} + a_{i_3} + a_{i_4})$. Then the matrix A^{t+1} obtained from A^t by adding row a_{i_5} is a sub-matrix of $A^\diamond$ and the vector

$$\lambda'_h = \begin{cases} \lambda_h - \delta & \text{if } h = i_1, \ldots, i_4 \\ \delta & \text{if } h = i_5 \\ \lambda_h & \text{otherwise} \end{cases}$$

of Lagrange multipliers for the new matrix has the property that $L(\lambda') \geq L(\lambda) + \delta$.

$\square$

THEOREM 5.2 *Let $a_{i_1}, a_{i_2}, a_{i_3}, a_{i_4}$ be four rows of A^t whose Lagrange multipliers are strictly positive and assume that there exists three columns j, k and l such that the associated sub-matrix of A^t has the following form:*

$$\begin{array}{c c} & \begin{array}{ccc} j & k & l \end{array} \\ \begin{array}{c} i_1 \\ i_2 \\ i_3 \\ i_4 \end{array} & \left[\begin{array}{rrr} 1 & 1 & 0 \\ 1 & -1 & 0 \\ -1 & 0 & 1 \\ -1 & 0 & -1 \end{array} \right] \end{array}. \tag{4}$$

Moreover, let

$$\delta = \min\{\lambda_{i_1}, \lambda_{i_2}, \lambda_{i_3}, \lambda_{i_4}\}$$

and let $a_{i_5} = sgn(a_{i_1} + a_{i_2} + a_{i_3} + a_{i_4})$. Then the matrix A^{t+1} obtained from A^t by adding row a_{i_5} is a sub-matrix of $A^\diamond$ and the vector

$$\lambda'_h = \begin{cases} \lambda_h - \delta & \textit{if } h = i_1, \ldots, i_4 \\ \delta & \textit{if } h = i_5 \\ \lambda_h & \textit{otherwise} \end{cases}$$

of Lagrange multipliers for the new matrix has the property that $L(\lambda') \geq L(\lambda) + \delta$.

$\square$

THEOREM 5.3 *Let a_h and a_k be two rows of A^t which are the antecedents of the resolvent $a_l = sgn(a_h + a_k)$ and whose Lagrange multipliers are strictly positive. Let P_h, N_h (P_k, N_k) be the sets of column indices where row a_h (a_k) has, respectively, positive and negative coefficients, excluded the hinge column. Let $\bar{c} = c - \lambda^T A^t$ be the vector of reduced costs. Moreover, let $\delta = \min\{\lambda_h, \lambda_k\}$ and let*

$$\gamma = \min\{\min_{j \in P_h \Delta P_k} \{\bar{c}_j + \delta - \min\{0, \bar{c}_j\}\}, \min_{j \in P_h \cup P_k} \{\bar{c}_j + 2\delta - \min\{0, \bar{c}_j\}\}.$$

Then the matrix A^{t+1} obtained from A^t by adding row a_l is a sub-matrix of $A^\diamond$ and the vector

$$\lambda'_i = \begin{cases} \lambda_i - \delta & \textit{if } i = h, k \\ \gamma & \textit{if } i = l \\ \lambda_i & \textit{otherwise} \end{cases}$$

of Lagrange multipliers for the new matrix has the property that $L(\lambda') \geq L(\lambda)$.

$\square$

The above observations suggest the following procedure for updating the current partial formulation, possibly improving the lower bound.

Observe that, at each execution of the inner statement, one row is added and at least one row is removed from the set of rows having strictly positive Lagrange multipliers. Hence, the number of such rows does not increase. Moreover, the new row has strictly more non-zero entries than the removed one. It follows that a crude bound for the number of executions of such a step is $n \times q$, where q is the number of rows in A^t.

Procedure *Improve* (A^t, λ);

begin

```
    repeat
    begin
        if there exists some configuration as in Theorems 5.1–5.3
          – update A^t and λ as appropriate;
        else
          – return;
        end if
    end
end
```

Figure 1: Separation Algorithm

6 An Example

Let A^0 be the $0, \pm 1$ matrix defined as follows:

$$A^0 = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5}$$

Consider the Generalized Set Covering Problem (2) associated with A^0. The value of the best Lagrangian relaxation is $L(\lambda^0) = 1/2$, associated with the vector of Lagrange multipliers $\lambda^0 = (1, 1/2, 1/2, 1/2)^T$. The vector of reduced costs is $\vec{c}^0 = (0, 0, 0, 0, 1/2, 1/2, 1/2)^T$.

Step 1.

Theorems 5.1–5.2 do not apply. Conversely, Theorem 5.3 can be applied to rows 1 and 2, producing the new row $(0, -1, 0, 1, 1, 0, 0)$. Hence, we can add such a row to A^0, producing the new matrix

$$A^1 = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}. \tag{6}$$

According to Theorem 5.3, we compute the values $\delta = 1/2$ and $\gamma = 1/2$. The new vector of multipliers (where the last entry is associated with the new row) will be

$\lambda^1 = (1/2, 0, 1/2, 1/2, 1/2)^T$, with $L(\lambda^1) = 1/2$. The new reduced costs will be $\bar{c}^1 = (0, 0, 0, 0, 0, 1/2, 1/2)^T$.

Step 2.

Theorems 5.1–5.2 do not apply. Conversely, Theorem 5.3 can be applied to rows 1 and 3, producing the new row $(0, 0, -1, 1, 0, 1, 0)$. Hence, we can add such a row to A^1, producing the new matrix

$$A^2 = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 1 & 0 \end{bmatrix}. \tag{7}$$

According to Theorem 5.3, we compute the values $\delta = 1/2$ and $\gamma = 1/2$. The new vector of multipliers (where the last entry is associated with the new row) will be $\lambda^2 = (0, 0, 0, 1/2, 1/2, 1/2)^T$, with $L(\lambda^2) = 1/2$. The new reduced costs will be $\bar{c}^2 = (0, 0, 0, 0, 0, 1/2, 1/2, 1/2)^T$.

Step 3.

Theorems 5.1–5.2 do not apply. Conversely, Theorem 5.3 can be applied to rows 4 and 5, producing the new row $(0, 0, 1, 1, 1, 0, 1)$. Hence, we can add such a row to A^2, producing the new matrix

$$A^3 = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}. \tag{8}$$

According to Theorem 5.3, we compute the values $\delta = 1/2$ and $\gamma = 1/2$. The new vector of multipliers (where the last entry is associated with the new row) will be $\lambda^3 = (0, 0, 0, 0, 0, 1/2, 1/2)^T$, with $L(\lambda^3) = 1/2$. The new reduced costs will be $\bar{c}^3 = (0, 0, 0, 0, 1/2, 1/2, 1/2)^T$.

Step 4.

Theorems 5.1–5.2 do not apply. Conversely, Theorem 5.3 can be applied to rows 6 and 7, producing the new row $(0, 0, 0, 1, 1, 1, 1)$. Hence, we can add such a row to A^3, producing the new matrix

$$A^4 = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}. \tag{9}$$

According to Theorem 5.3, we compute the values $\delta = 1/2$ and $\gamma = 1$. The new vector of multipliers (where the last entry is associated with the new row) will be $\lambda^4 = (0,0,0,0,0,0,0,1)^T$, with $L(\lambda^4) = 1$. The new reduced costs will be $\bar{c}^4 = (0,0,0,0,0,0,0)^T$.

Since no further applications of Theorems 5.1–5.3 is possible, the algorithm stops with the partial formulation A^4 (which is a proper sub-matrix of $A^\diamond$) and a lower bound of value 1. □

References

[1] E. BALAS and M. CARRERA, A dynamic subradient-based branch-and-bound procedure for set covering, Management Science Research Report No. 568, Carnegie Mellon University, (1992). To appear in *Operations Research.*

[2] E. BALAS and A. HO, Set covering algorithms using cutting planes, heuristics, and subgradient optimization: a computational study, *Mathematical Programming Study*, **12**, 37–60 (1980).

[3] J.E. BEASLEY, A Lagrangean heuristic for set-covering problems, *Naval Research Logistics*, **37**, 151–164 (1990).

[4] M.L. FISHER and P. KEDIA, Optimal solution of set covering / partitioning problems using dual heuristics, *Management Science*, **36**, 674–688 (1990).

[5] G. GALLO, C. GENTILE, D. PRETOLANI and G. RAGO, Max Horn Sat and the Minimum Cut Problem in Directed Hypergraphs, Dipartimento di Informatica, Universita' degli Studi di Pisa, Technical Report TR-15/95, November 1995.