Istituto di Analisi dei Sistemi ed Informatica
"Antonio Ruberti"
CONSIGLIO NAZIONALE DELLE RICERCHE

C. Buchheim, G. Rinaldi

TERSE INTEGER LINEAR PROGRAMS FOR BOOLEAN OPTIMIZATION

R. xxx Maggio 2008

Christoph Buchheim – Institut für Informatik, Universität zu Köln, Pohligstr. 1, 50969 Köln, Germany (buchheim@informatik.uni-koeln.de).

Giovanni Rinaldi – Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti" del CNR, viale Manzoni 30, 00185 Roma, Italy (rinaldi@iasi.cnr.it).

This work was partially supported by the Marie Curie RTN 504438 (ADONET) funded by the European Commission. The first author was supported by Deutsche Forschungsgemeinschaft (DFG) under grant BU 2313/1–1.

ISSN: 1128–3378

Collana dei Rapporti dell'Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti", CNR

viale Manzoni 30, 00185 ROMA, Italy

tel. ++39-06-77161
fax ++39-06-7716461
email: iasi@iasi.rm.cnr.it
URL: http://www.iasi.rm.cnr.it

Abstract

We present a new polyhedral approach to nonlinear boolean optimization problems. Compared to other methods, our approach produces much smaller integer programming models, making it more efficient from a practical point of view. We mainly obtain this by two different ideas: first, we do not require the objective function to be in any normal form. The transformation into a normal form usually leads to the introduction of many additional variables or constraints. Second, we reduce the problem to the degree-two case in a very efficient way, using a slightly extended formulation. The resulting model turns out to be closely related to the maximum cut problem; we show that the corresponding polytope is a face of a suitable cut polytope in most cases. In particular, our separation problem reduces to the one for the maximum cut problem.

In practice, our approach turns out to be very competitive. First experimental results, which have been obtained for some particularly hard instances of the Max-SAT Evaluation 2007, show that our very general implementation can outperform even special-purpose SAT solvers.

Key words: logic optimization, pseudo-boolean optimization, maximum satisfiability

AMS subject classifications: 90C57, 65K05, 03B70

1. Introduction

Nonlinear zero-one optimization problems are often solved by transforming the objective function into an appropriate normal form, e.g., into conjunctive normal form (CNF) or into a polynomial. The problem can then be addressed by a general solver for maximum satisfiability, polynomial zero-one optimization or some other standard problem. However, the transformation often increases the problem size significantly, since new variables or constraints have to be added. Depending on the normal form, an exponential blow-up might be unavoidable, e.g., if negations in a polynomial have to be resolved. But even if the increase is tractable from a theoretical point of view, in practice it might lead to a problem instance that is too large to be solved.

In this paper, we present a novel approach that avoids the transformation into any normal form, by directly modeling arbitrarily constructed boolean functions into an integer linear program. The strength of our approach lies in the fact that, nevertheless, a tight polyhedral description for the resulting model can be obtained. This description is based on a reduction of the general problem to the special case of unconstrained quadratic zero-one optimization, which is known to be equivalent to the maximum cut problem [5].

We first discuss the quadratic case, i.e., the case where all objective function terms contain at most one binary operator. We show that in this case the polytope corresponding to our formulation is isomorphic to a cut polytope, no matter which operators are considered. This is a generalization of [5], where all operators are multiplications.

The situation is more complicated in the general case where we allow arbitrary boolean functions defined recursively by binary operators. Again, our aim is to avoid introducing too many new variables or constraints, as done by other approaches such as lift-and-project. In [3], we developed a new approach for polynomial zero-one optimization problems that uses an efficient reduction to the quadratic case. The reduction can be applied after a slight extension of the variable space; the resulting polytope then turns out to be a face of a polytope corresponding to a quadratic instance of basically the same type. This allows to derive a polyhedral description for a general instance from the polyhedral description of an appropriate quadratic problem.

In the following, we generalize these results. In place of multiplications, we allow arbitrary binary operators. For problem instances not containing any exclusive disjunctions or equivalences, we show that the general polytope is still a face of an appropriate cut polytope, defined on roughly four times as many variables as in the original model. If exclusive disjunctions or equivalences are present, they can be replaced by at most three other operators each.

2. The Problem

We consider an unconstrained boolean optimization problem in the following form: a set of boolean variables $x_i \in \{0,1\}$ for $i \in I$ is given, where I is a finite index set. We set $n = |I|$. Moreover, we have a pseudo-boolean objective function

$$\min \sum_{k \in K} c_k f_k \ , \tag{1}$$

where each f_k is a boolean function $f\colon \{0,1\}^I \to \{0,1\}$ over the variables x_i and the coefficients c_k are arbitrary real numbers. For our purposes, the set of boolean functions is defined recursively as follows: first, each variable x_i for $i \in I$ corresponds to a boolean function f_i, defined by

$$f_i\colon \{0,1\}^I \to \{0,1\}, \quad (x_s)_{s \in I} \mapsto x_i \ .$$

notation	equivalent to	description
$a \wedge b$	$a \cdot b$, $\min(a, b)$	conjunction
$a \vee b$	$\max(a, b)$	disjunction
$a \Rightarrow b$	$a \leq b$, $(\neg a) \vee b$	implication
$a \Leftarrow b$	$a \geq b$, $a \vee (\neg b)$	
$a \bar{\wedge} b$	$(\neg a) \vee (\neg b)$	
$a \bar{\vee} b$	$(\neg a) \wedge (\neg b)$	
$a \not\Rightarrow b$	$a \wedge (\neg b)$	
$a \not\Leftarrow b$	$(\neg a) \wedge b$	
$a \oplus b$	$a \neq b$	exclusive disjunction
$a \Leftrightarrow b$	$a = b$	equivalence
0		constant zero
1		constant one
a		identity in first argument
b		identity in second argument
$\neg a$	$1 - a$	negation of first argument
$\neg b$	$1 - b$	negation of second argument

Table 1: All 16 binary operators.

Second, if g and h are boolean functions and $\circ$ is any binary operator $\{0,1\}^2 \to \{0,1\}$, then $g \circ h$ is a boolean function as well. In the following, our aim is to address problem (1) by an approach that is based on modeling each such boolean function independently.

Example 2.1. All CNF and DNF clauses are boolean functions. In particular, the maximum satisfiability problem is a special case of (1). □

Example 2.2. Binary monomials are boolean functions, since multiplication of binary variables can be considered a binary operator on boolean variables. In particular, a special case of (1) is binary polynomial optimization. □

Example 2.3. If f is any boolean function as defined above, then checking satisfiability of f amounts to checking whether $\min(\neg f) = 0$. Checking whether f is a tautology amounts to checking whether $\min f = 1$. In particular, we can check equivalency of f and any other logical formula g by minimizing the boolean function $f \Leftrightarrow g$. □

For ease of exposition, we assume throughout that all operators appearing in the functions are proper binary operators, i.e., we do not consider unary or constant operators. This situation can always be obtained as follows: constant operators can be resolved easily. The only non-trivial unary operator is negation. In the objective function, negations can be resolved by replacing $\neg a$ by $1-a$. Elsewhere, negations can be merged into binary operators, e.g., we can consider $a \vee (\neg b)$ a binary operator in a and b. In summary, out of the 16 existing binary operators given in Table 1, we only have to consider the first ten.

In theory, problem (1) can model the minimization of an arbitrary function $f\colon \{0,1\}^n \to \mathbb{R}$, as

$$f = \sum_{t: I \to \{0,1\}} f(t) \bigwedge_{t(i)=1} x_i \wedge \bigwedge_{t(i)=0} \neg x_i \;, \tag{2}$$

where $f(t)$ is the value of f when the variables are set according to t. Unfortunately, this representation of f is useless, as computing all $f(t)$ yields the minimum immediately. However, if a much more compact representation of f than (2) is known, the approach presented in the following becomes very efficient. The efficiency of our approach mainly depends on the total number of operators in all functions f_k, $k \in K$. Clearly, the number of functions $\{0,1\}^n \to \{0,1\}$ that can be modeled with a fixed number of arbitrary binary operators, as explained above, is much bigger than, e.g., the number of CNF formulae using the same number of operators.

In this context, we point out that some effort in compactifying the representation of f can be worthwhile. Even if the original instance is given in some normal form such as CNF, it might well pay off to give up the normal form if this leads to a smaller number of operators. In general, the more the large flexibility of our approach is exploited, the more it can be expected to outperform methods designed for instances with specific structure.

3. New Formulation and Reduction to Maximum Cut

In the following, we develop our new model for logic optimization problems. This approach has two main advantages over other methods: the number of variables is kept small, even if the objective function does not conform to any normal form. The second advantage is that the corresponding polytope turns out to be a face of an appropriate cut polytope, if the model is slightly extended and no exclusive disjunctions or equivalences appear in the objective function. This allows to address the general problem by a cutting plane approach that is entirely based on separation algorithms for the maximum cut problem.

3.1. The Model

In order to develop a linear model for problem (1), we have to introduce further binary variables. First, we need to linearize the objective function by adding a variable $x_k \in \{0,1\}$ representing f_k, for every $k \in K$. The objective then translates to min $c^\top x_K$, and it remains to model the connection between the *basic* variables x_I and the *objective* variables x_K. More precisely, we have to make sure that

$$x_k = f_k(x_I) \text{ for every feasible solution } x \in \{0,1\}^{I \cup K} .$$

In order to bridge the gap between basic and objective variables, we additionally introduce *connection* variables $x_j \in \{0,1\}$, $j \in J$. Every such variable corresponds to an intermediate function in the recursive definition of the boolean functions f_k.

More formally, we determine the set of connection variables recursively as follows: we start with $J = \emptyset$. Let $j \in J \cup K$ with $f_j = g \circ h$, for appropriate boolean functions g and h. If $g = f_s$ for some $s \in I \cup J \cup K$, we define $l(j) = s$, otherwise we add a new index $l(j)$ to J, introduce a new variable $x_{l(j)}$ representing g, and define $f_{l(j)} = g$. Analogously, if $h = f_s$ for some $s \in I \cup J \cup K$, we define $r(j) = s$, otherwise we add a new index $r(j)$ to J, introduce a variable $x_{r(s)}$ representing h, and define $f_{r(j)} = h$. We continue like this until every f_j with $j \in J \cup K$ is of the form $f_{l(j)} \circ f_{r(j)}$ for $l(j), r(j) \in I \cup J \cup K$.

In summary, we have constructed a set of variables $\{x_s \mid s \in I \cup J \cup K\}$ and a corresponding set of boolean functions $F = \{f_s \mid s \in I \cup J \cup K\}$. Every non-basic function in F is the result of applying some binary operator to an appropriate pair of other functions in F. Notice that the total number of connection and objective variables $|J \cup K|$ is at most the total number m of operators in the objective function, so the total number of variables in our model is at most $n+m$.

In practice, it is often possible to save a lot of these variables by intelligent decomposition of the objective function.

Every feasible solution in our model corresponds to a truth assignment to the basic variables x_i, i.e., to a function $t: I \to \{0,1\}$. The corresponding characteristic vector $\chi_t \in \{0,1\}^{I\cup J\cup K}$ is defined in the obvious way—every component $(\chi_t)_s$ takes the value of f_s under t, denoted by $t(f_s)$ in the following. Now we define

$$P = \operatorname{conv}\{\chi_t \mid t: I \to \{0,1\}\} \subset \mathbb{R}^{I\cup J\cup K} .$$

We can thus restate problem (1) as $\min c^\top x_K$ s.t. $x \in P$.

In order to solve this problem, it is necessary to find tight linear relaxations of the polytope P. The standard techniques for linearizing polynomial terms in binary programs could be adopted to our model, however, the resulting relaxations for P are weak in general. Instead, we aim at generalizing the results we obtained for binary polynomial optimization, presented in [3], to the more general situation considered here. In the remainder of this section, we will show that P is a face of an appropriate cut polytope of small dimension if the objective function does not contain exclusive disjunctions or equivalences.

The proof is done in two steps: first the result is shown in the quadratic case, i.e., when all objective terms f_k contain at most one operator. In fact, P is isomorphic to a cut polytope in this case. This is a generalization of a result by De Simone showing that binary quadric polytopes are isomorphic to cut polytopes [5]. We obtain this result without restricting the set of allowed operators.

Second, we show that in the case of objective functions of arbitrary degree, the polytope P is a face of a polytope P^* defined by a quadratic instance of our problem, if the objective function does not contain exclusive disjunctions or equivalences. In other words, the case of arbitrary degree can be reduced to the quadratic case then.

3.2. Quadratic Case

In the quadratic case, the polytope P is always isomorphic to an appropriate cut polytope defined on the same number of variables.

Lemma 3.1. *Let f_k contain at most one operator for all $k \in K$. Then the polytope P is isomorphic to a cut polytope. The corresponding graph has $n+m$ edges.*

Proof: In this case, we have $J = \emptyset$ and $l(s), r(s) \in I$ for all $s \in K$. We can thus define a graph $G = (V, E)$ by

$$\begin{aligned} V &= \{r\} \cup \{v_s \mid s \in I\} \\ E &= \{(r, v_s) \mid s \in I\} \cup \{(v_{l(s)}, v_{r(s)}) \mid s \in K\} . \end{aligned}$$

Let $f_s = f_{l(s)} \circ_s f_{r(s)}$ for all $s \in K$. Define a linear map $\psi': \mathbb{R}^E \to \mathbb{R}^{I\cup K}$ as follows:

$$\begin{aligned} e_{(r,v_s)} \quad \mapsto \quad & e_s \\ & + \sum_{\substack{k \in K \\ s = l(k)}} 1/2(-0 \circ_k 0 - 0 \circ_k 1 + 1 \circ_k 0 + 1 \circ_k 1) \cdot e_k \\ & + \sum_{\substack{k \in K \\ s = r(k)}} 1/2(-0 \circ_k 0 + 0 \circ_k 1 - 1 \circ_k 0 + 1 \circ_k 1) \cdot e_k \end{aligned}$$

$$e_{(v_{l(s)},v_{r(s)})} \quad \mapsto \quad 1/2(-0 \circ_s 0 + 0 \circ_s 1 + 1 \circ_s 0 - 1 \circ_s 1) \cdot e_s \,.$$

As ψ' is bijective, the map $\psi\colon \mathbb{R}^E \to \mathbb{R}^{I \cup K}$ given by $x \mapsto \psi'(x) + \sum_{k \in K}(0 \circ_k 0)e_k$ is an affine isomorphism. Hence it suffices to show that ψ induces a bijection between the vertices of the cut polytope $\mathcal{C}(G)$ of G and the vertices of P.

So consider the characteristic vector $\chi_S \in \mathcal{C}(G)$ of any cut $S \subseteq V$, where we may assume that $r \notin S$. Define a truth assignment $t\colon I \to \{0,1\}$ by setting $t(s) = 1$ if and only if $v_s \in S$. We claim that $\psi(\chi_S) = \chi_t$. Indeed,

$$\chi_S = \sum_{\substack{s \in I \\ v_s \in S}} e_{(r,v_s)} + \sum_{\substack{s \in K \\ v_{l(s)} \in S \oplus v_{r(s)} \in S}} e_{(v_{l(s)},v_{r(s)})} \,,$$

thus

$$\psi(\chi_S) = \sum_{\substack{s \in I \\ t(s) = 1}} \psi(e_{(r,v_s)}) + \sum_{\substack{s \in K \\ t(l(s)) \oplus t(r(s)) = 1}} \psi(e_{(v_{l(s)},v_{r(s)})}) \,,$$

so that for $s \in I$ we have $\psi(\chi_S)_s = t(s) = (\chi_t)_s$ and, for $s \in K$,

$$\begin{aligned} \psi(\chi_S)_s &= 1/2(-0 \circ_s 0 + 0 \circ_s 1 + 1 \circ_s 0 - 1 \circ_s 1) \cdot t(l(s)) \oplus t(r(s)) \\ &\quad +1/2(-0 \circ_s 0 - 0 \circ_s 1 + 1 \circ_s 0 + 1 \circ_s 1) \cdot t(l(s)) \\ &\quad +1/2(-0 \circ_s 0 + 0 \circ_s 1 - 1 \circ_s 0 + 1 \circ_s 1) \cdot t(r(s)) + (0 \circ_s 0) \\ &= t(l(s)) \circ_s t(r(s)) = (\chi_t)_s \,. \end{aligned} \tag{3}$$

Conversely, for given $t\colon I \to \{0,1\}$ we define a cut of G by $S = \{v_s \in V \mid t(s) = 1\}$. This construction is obviously inverse to the one above, so the proof is complete. □

The main ingredient in the proof of Lemma 3.1 is the reformulation of an arbitrary binary operator as an affine combination of exclusive disjunction and basic variables, using (3). For the ten operators to be considered, the corresponding formulae are listed in Table 2.

operator	reformulation
$a \wedge b$	$1/2(a + b - a \oplus b)$
$a \vee b$	$1/2(a + b + a \oplus b)$
$a \Rightarrow b$	$1/2(-a + b - a \oplus b) + 1$
$a \Leftarrow b$	$1/2(a - b - a \oplus b) + 1$
$a \bar{\wedge} b$	$1/2(-a - b + a \oplus b) + 1$
$a \bar{\vee} b$	$1/2(-a - b - a \oplus b) + 1$
$a \not\Rightarrow b$	$1/2(a - b + a \oplus b)$
$a \not\Leftarrow b$	$1/2(-a + b + a \oplus b)$
$a \oplus b$	$a \oplus b$
$a \Leftrightarrow b$	$-a \oplus b + 1$

Table 2: Reformulation of binary operators in terms of exclusive disjunctions.

3.3. General Case

In this section, we do not require a quadratic objective function any more. Moreover, we do not assume any normal form, all operators may be mixed arbitrarily. Nevertheless, we can show the following result.

Theorem 3.2. *Assume that no operator in the objective function is an exclusive disjunction or an equivalence. Then the polytope P is isomorphic to a face of a cut polytope. The corresponding graph has at most $n + 4m$ edges.*

Proof: By the previous lemma, it suffices to show that P is a face of some polytope P^* that corresponds to a quadratic instance of our problem with at most $n+m$ basic variables and at most $3m$ operators in total. In order to construct this quadratic instance, define the set of basic variables to be $\{x_s^0 \mid s \in I \cup J\}$ and set $I^* = I \cup J$. Moreover, define a new set of quadratic objective terms over these variables as

$$\{x_s^1 = x_{l(s)}^0 \circ_s x_{r(s)}^0 \mid s \in J \cup K\} \cup \{x_s^2 = x_{l(s)}^0 \wedge x_s^0,\ x_s^3 = x_{r(s)}^0 \wedge x_s^0 \mid s \in J\}\ .$$

Let K^* be an index set for these $3|J| + |K|$ objective terms. Denote the corresponding polytope in $\mathbb{R}^{I^* \cup K^*}$ by P^*. We will show that P is a face of P^*.

For the following, define $c_s = 1 \circ_s 1 + 0 \circ_s 0 - 1 \circ_s 0 - 0 \circ_s 1$, and observe that $c_s \neq 0$ for all (strictly) binary operators. We first claim that

$$P \cong \operatorname{conv}\left(P^* \cap X \cap \{0,1\}^{I^* \cup K^*}\right), \tag{4}$$

where X is the linear subspace of $\mathbb{R}^{I^* \cup K^*}$ given by the equations

$$\begin{aligned}
x_s^1 &= x_s^0 & (5)\\
x_s^2 &= (c_s^{-1}(1 \circ_s 1 - 1 \circ_s 0))x_s^0 & (6)\\
&\quad +(c_s^{-1}(1 \circ_s 1 - 1 \circ_s 0)(0 \circ_s 0 - 1 \circ_s 0) + (1 \circ_s 0))x_{l(s)}^0 &\\
&\quad +(c_s^{-1}(1 \circ_s 1 - 1 \circ_s 0)(0 \circ_s 0 - 0 \circ_s 1))x_{r(s)}^0 &\\
&\quad -c_s^{-1}(1 \circ_s 1 - 1 \circ_s 0)(0 \circ_s 0) &\\
x_s^3 &= (c_s^{-1}(1 \circ_s 1 - 0 \circ_s 1))x_s^0 & (7)\\
&\quad +(c_s^{-1}(1 \circ_s 1 - 0 \circ_s 1)(0 \circ_s 0 - 0 \circ_s 1) + (0 \circ_s 1))x_{r(s)}^0 &\\
&\quad +(c_s^{-1}(1 \circ_s 1 - 0 \circ_s 1)(0 \circ_s 0 - 1 \circ_s 0))x_{l(s)}^0 &\\
&\quad -c_s^{-1}(1 \circ_s 1 - 0 \circ_s 1)(0 \circ_s 0) &
\end{aligned}$$

for all $s \in J$. The isomorphism is induced by the linear map $\varphi\colon \mathbb{R}^{I^* \cup K^*} \cap X \to \mathbb{R}^{I \cup J \cup K}$ that is uniquely defined by $\varphi(e_s^0) = e_s$ for $s \in I^*$ and $\varphi(e_s^1) = e_s$ for $s \in K$.

Indeed, consider a vertex χ_t of P corresponding to an assignment $t\colon I \to \{0,1\}$. Extend it to $t^*\colon I^* \to \{0,1\}$ in the natural way, setting $t^*(s) = t(f_s)$ for all $s \in J$. Now by construction we have $\chi_t = \varphi(\chi_{t^*}^*)$, where $\chi_{t^*}^*$ denotes the characteristic vector of t^* in P^*. To see this, note that

$$x_{l(s)} \circ_s x_{r(s)} = c_s(x_{l(s)} \wedge x_{r(s)}) + (1 \circ_s 0 - 0 \circ_s 0)x_{l(s)} + (0 \circ_s 1 - 0 \circ_s 0)x_{r(s)} + 0 \circ_s 0$$

and that

$$\begin{aligned}
(x_{l(s)} \wedge x_s) &= (1 \circ_s 1 - 1 \circ_s 0)(x_{l(s)} \wedge x_{r(s)}) + (1 \circ_s 0)x_{l(s)}\\
(x_{r(s)} \wedge x_s) &= (1 \circ_s 1 - 0 \circ_s 1)(x_{l(s)} \wedge x_{r(s)}) + (0 \circ_s 1)x_{r(s)}\ .
\end{aligned}$$

Conversely, any point in $P^* \cap \{0,1\}^{I^* \cup K^*}$ is a vertex $\chi^*_{t^*}$ of P^* corresponding to a truth assignment $t^*\colon I^* \to \{0,1\}$. Let $t = t^*|_I$. By (5), all vectors in X satisfy

$$x^0_s = x^1_s = x^0_{l(s)} \circ_s x^0_{r(s)} \text{ for all } s \in J \ ,$$

so that we can inductively show that $x^0_s = t(f_s)$ for $s \in I^*$ supposed that the same holds for $s \in I$. In other words, t^* is the extension of t described above, hence we have $\chi_t = \varphi(\chi^*_{t^*})$ again.

Having proved (4), it remains to show that X induces a face of P^*, since this implies that $P^* \cap X$ is integer so that (4) yields an isomorphism $P \cong P^* \cap X$. This is true for all operators except for exclusive disjunctions and equivalences, i.e., for $\circ_s \in \{\wedge, \vee, \Rightarrow, \Leftarrow, \bar{\wedge}, \bar{\vee}, \not\Rightarrow, \not\Leftarrow\}$.

For each of these operators, we claim that the equations (6) and (7) hold as inequalities for P^*, the direction depending on the operator. Indeed, the right hand side of (6) reads

$$\begin{array}{ll} x^0_s & \text{for } \circ_s \in \{\wedge, \not\Rightarrow\} \\ x^0_{l(s)} & \text{for } \circ_s \in \{\vee, \Leftarrow\} \\ x^0_s + x^0_{l(s)} - 1 & \text{for } \circ_s \in \{\Rightarrow, \bar{\wedge}\} \\ 0 & \text{for } \circ_s \in \{\bar{\vee}, \not\Leftarrow\} \ . \end{array}$$

In the first two cases, this right hand side is greater or equal to $x^0_{l(s)} \wedge x^0_s = x^2_s$ for every integer point in P^*. In the other two cases, this right hand side is less or equal to $x^0_{l(s)} \wedge x^0_s = x^2_s$ for every integer point in P^*. Thus (6) induces a face of P^*. For (7), the same result follows from symmetry.

So let F be the face of P^* induced by the two equations (6) and (7). It remains to show that (5) induces a face of F. From (6) we derive

$$\begin{array}{llll} x^0_s = 0 & \text{or} & x^0_{l(s)} = 1 & \text{if } \circ_s \in \{\wedge, \not\Rightarrow\} \\ x^0_s = 1 & \text{or} & x^0_{l(s)} = 0 & \text{if } \circ_s \in \{\vee, \Leftarrow\} \\ x^0_s = 1 & \text{or} & x^0_{l(s)} = 1 & \text{if } \circ_s \in \{\Rightarrow, \bar{\wedge}\} \\ x^0_s = 0 & \text{or} & x^0_{l(s)} = 0 & \text{if } \circ_s \in \{\bar{\vee}, \not\Leftarrow\} \end{array}$$

and (7) yields

$$\begin{array}{llll} x^0_s = 0 & \text{or} & x^0_{r(s)} = 1 & \text{if } \circ_s \in \{\wedge, \not\Leftarrow\} \\ x^0_s = 1 & \text{or} & x^0_{r(s)} = 0 & \text{if } \circ_s \in \{\vee, \Rightarrow\} \\ x^0_s = 1 & \text{or} & x^0_{r(s)} = 1 & \text{if } \circ_s \in \{\Leftarrow, \bar{\wedge}\} \\ x^0_s = 0 & \text{or} & x^0_{r(s)} = 0 & \text{if } \circ_s \in \{\bar{\vee}, \not\Rightarrow\} \end{array}$$

and hence

$$\begin{array}{ll} x^0_s \le (x^0_{l(s)} \circ_s x^0_{r(s)}) = x^1_s & \text{if } \circ_s \in \{\wedge, \bar{\vee}, \not\Rightarrow, \not\Leftarrow\} \\ x^0_s \ge (x^0_{l(s)} \circ_s x^0_{r(s)}) = x^1_s & \text{if } \circ_s \in \{\vee, \Rightarrow, \Leftarrow, \bar{\wedge}\} \ . \end{array}$$

This completes the proof. □

Example 3.1. To illustrate the construction of Theorem 3.2, consider the case of a single CNF-clause $x_1 \vee x_2 \vee x_3$ containing three non-negated variables, which is the smallest non-trivial example. Then the polytope P corresponding to the problem

$$\begin{array}{rl} \min & x_1 \vee x_2 \vee x_3 \\ \text{s.t.} & x_1, x_2, x_3 \in \{0,1\} \end{array}$$

is defined over five binary variables, corresponding to the boolean functions

$$x_1,\ x_2,\ x_3,\ x_1 \vee x_2,\ (x_1 \vee x_2) \vee x_3\ ,$$

where the first three are basic variables and can thus be chosen freely, while the other two variables are determined by the basic variables. So P is spanned by the 2^3 vectors

$$\begin{matrix} (0\,0\,0\ \ 0\,0) & (0\,1\,0\ \ 1\,1) & (1\,0\,0\ \ 1\,1) & (1\,1\,0\ \ 1\,1) \\ (0\,0\,1\ \ 0\,1) & (0\,1\,1\ \ 1\,1) & (1\,0\,1\ \ 1\,1) & (1\,1\,1\ \ 1\,1) \end{matrix}$$

Now the constructed polytope P^* is defined on four basic variables

$$x_1^0,\ x_2^0,\ x_3^0,\ (x_1 \vee x_2)^0$$

and four quadratic terms

$$x_1^0 \vee x_2^0,\ (x_1 \vee x_2)^0 \vee x_3^0,\ x_1^0 \wedge (x_1 \vee x_2)^0,\ x_2^0 \wedge (x_1 \vee x_2)^0\ .$$

The 2^4 vertices of P^* are

$$\begin{matrix} (0\,0\,0\,0\ \ 0\,0\,0\,0) & (0\,1\,0\,0\ \ 1\,0\,0\,0) & (1\,0\,0\,0\ \ 1\,0\,0\,0) & (1\,1\,0\,0\ \ 1\,0\,0\,0) \\ (0\,0\,0\,1\ \ 0\,1\,0\,0) & (0\,1\,0\,1\ \ 1\,1\,0\,1) & (1\,0\,0\,1\ \ 1\,1\,1\,0) & (1\,1\,0\,1\ \ 1\,1\,1\,1) \\ (0\,0\,1\,0\ \ 0\,1\,0\,0) & (0\,1\,1\,0\ \ 1\,1\,0\,0) & (1\,0\,1\,0\ \ 1\,1\,0\,0) & (1\,1\,1\,0\ \ 1\,1\,0\,0) \\ (0\,0\,1\,1\ \ 0\,1\,0\,0) & (0\,1\,1\,1\ \ 1\,1\,0\,1) & (1\,0\,1\,1\ \ 1\,1\,1\,0) & (1\,1\,1\,1\ \ 1\,1\,1\,1) \end{matrix}$$

Equation (5) reads $(x_1 \vee x_2)^0 = x_1^0 \vee x_2^0$. This equation excludes half the vertices of P^*, namely those where the fourth and fifth entry do not agree, i.e., those not corresponding to solutions of the original problem. Equations (6) and (7) read $x_1^0 \wedge (x_1 \vee x_2)^0 = x_1^0$ and $x_2^0 \wedge (x_1 \vee x_2)^0 = x_2^0$, they are needed to ensure that P is isomorphic to a face of P^*. We end up with a quadratic problem formulation

$$\begin{array}{rlcl} \min & (x_1 \vee x_2)^0 \vee x_3^0 & & \\ \text{s.t.} & x_1^0 \vee x_2^0 & = & (x_1 \vee x_2)^0 \\ & x_1 \wedge (x_1 \vee x_2)^0 & = & x_1^0 \\ & x_2 \wedge (x_1 \vee x_2)^0 & = & x_2^0 \\ & x_1^0, x_2^0, x_3^0, (x_1 \vee x_2)^0 \in \{0,1\}\ . & & \end{array}$$

□

Example 3.2. If one of the operators is an exclusive disjunction or an equivalence, it is not true in general that $\varphi(P)$ is a face of the polytope P^* constructed in Theorem 3.2. To see this, consider the objective function $x_1 \circ (x_2 \oplus x_3)$, for any operator $\circ$. Then P^* is defined over the basic variables

$$x_1^0,\ x_2^0,\ x_3^0,\ (x_2 \oplus x_3)^0$$

and the quadratic terms

$$x_2^0 \oplus x_3^0,\ x_1^0 \circ (x_2 \oplus x_3)^0,\ x_2^0 \wedge (x_2 \oplus x_3)^0,\ x_3^0 \wedge (x_2 \oplus x_3)^0\ .$$

By (4), the polytope $\varphi(P)$ is spanned by a subset of the vertices of P^*. In our example, one can verify by direct computation that the barycenters of $\varphi(P)$ and P^* agree, i.e., that $\varphi(P)$ cuts through the center of P^*. The same construction works with $\Leftrightarrow$ in place of $\oplus$. □

Corollary 3.3. *The polytope P is isomorphic to a projection of a face of a cut polytope. The corresponding graph has at most $n + 12m$ edges.*

Proof: After replacing all exclusive disjunctions and equivalences using the identities

$$\begin{aligned} a \oplus b &= (a \not\Rightarrow b) \vee (a \not\Leftarrow b) \\ a \Leftrightarrow b &= (a \Rightarrow b) \wedge (a \Leftarrow b) \ , \end{aligned}$$

we get a new instance of our problem. Let P' denote the polytope defined by this instance. Then P' is isomorphic to a face of a cut polytope on at most $n + 12m$ edges by Theorem 3.2. On the other hand, it is clear by definition that P is an orthogonal projection of P'. □

3.4. Constraints

So far we have discussed unconstrained logic optimization problems, where all boolean functions in the problem formulation appear in the objective function. However, it is clear that the same approach works if we have constraints of the form $f_k = 0$ or $f_k = 1$, where f_k is any boolean function. In this case, we model f_k exactly as we model the objective terms. If we consider the corresponding polytope P and intersect it with the hyperplane $x_k = 0$ or $x_k = 1$, then we obviously get a face of P.

If we consider linear constraints instead of logical ones, the situation is more complicated. In general, we cannot rescue our polyhedral results in this case. However, in some special cases, the situation is again favorable. To give an example, consider the constraint

$$\sum_{s \in L} f_s \leq 1 \tag{8}$$

for an arbitrary subset $L \subseteq I \cup J \cup K$. This constraint states that at most one of the functions f_s with $s \in L$ may evaluate to one. In order to model (8) without harming our polytope P, we introduce a zero-weight objective term $f_L = \bigvee_{s \in L} f_s$. Moreover, we have to add up to $|L| - 1$ connection variables. Then we can rephrase (8) as

$$\sum_{s \in L} f_s = f_L \ . \tag{9}$$

The latter formulation is preferable since $\sum_{s \in L} f_s \geq f_L$ is a valid constraint for P, so that (9) induces a face of P. In the same way, we can deal with an equation

$$\sum_{s \in L} f_s = 1 \ ,$$

stating that exactly one of the functions f_s with $s \in L$ evaluates to one. Additionally to (9) we have to set $f_L = 1$ here, which again induces a face.

4. Experiments

First results obtained with a straightforward implementation of our approach show that the increased modeling power leads to much faster running times in practice. In this section, in order to demonstrate this by some examples, we shortly discuss two classes of instances taken from the Max-SAT Evaluation 2007 [1]. We chose those classes where the percentage of instances solved by the best participating algorithm was particularly small, namely the `logic-synthesis` and the `SPOT5` instances. Both classes belong to the partial Max-SAT category, i.e., some of the clauses have to be satisfied by every solution, while others have positive integer weights in the objective function.

In our implementation, we first compactify each instance by simple reformulations, yielding an equivalent objective function that may not be a CNF any more, but that still fits into our much more general framework. In our experience, very straightforward techniques can already decrease the number of operators in the objective function by 30–70 %. One compaction method that often turns out to be very effective is the equivalent replacement

$$f_i \vee f_j \quad \text{for all } i,j \in I \text{ with } i \neq j \quad \Longleftrightarrow \quad \sum_{i \in I} (\neg f_i) \leq 1 \ ,$$

where each f_i can be an arbitrary boolean function. Notice that SAT solvers can only handle the former group of constraints, while in our approach we can also deal with the latter constraint, as explained in Section 3.4. This reduces the number of operators from $\binom{|I|}{2}$ to $|I|-1$. To detect such sets of constraints, we apply a simple algorithm for finding maximal cliques in the conflict graph defined on all boolean functions in the instance.

The reduced instance is then handed over to the CPLEX 11.0 MIP solver [4], which tries to optimize it using a typical branch-and-cut algorithm. Our only (but crucial) extension of the standard solver concerns the separation phase, where we make use of our results presented in Section 3.3 above: we first create the graph corresponding to the cut polytope constructed in the proof of Theorem 3.2. After transforming any given fractional solution to the variables space of the cut polytope, we apply a separation algorithm for the maximum cut problem on this graph. In our current implementation, we only separate cycle inequalities [2]. We do this heuristically, as it turns out that otherwise the time for separation, though polynomial, is too long because of the large graphs considered. Any resulting cutting plane can easily be transformed back to the original variable space. All other components of the branch-and-cut algorithm can be applied to the original set of variables. This is true, in particular, for the solution of LP relaxations and for branching. The results reported in the following were obtained on an Intel Xeon 5130 processor with 2 GHz running Linux, i.e., on a machine that is roughly comparable to the one used for the Max-SAT Evaluation 2007. We set the same cpu time limit of 30 minutes per instance.

Instances in the class `logic-synthesis` are unweighted. The running times obtained with our approach described above turn out to be more than competitive: In the given 30 cpu minutes, we could solve to optimality 16 out of the 17 instances in this class. On contrary, half of the 10 participants of the Max-SAT Evaluation could not solve a single of these instances, while the others could solve between 1 and 4. More detailed results are displayed in Table 3. We state the number of solved instance and the average values of the running time in cpu seconds, the running time without preprocessing (i.e., the running time for the branch & cut-algorithm), the number of subproblems in the enumeration tree, the number of nodes and edges in the auxiliary max-cut graph, and the number of cycle inequalities generated. We noticed in our experiments that generating a relatively small number of cycle inequalities can already lead to a significant reduction of the number of subproblems in the enumeration tree. Adding more cycle inequalities can reduce the number of subproblems even further, but at the expense of a longer separation time, which only pays off for larger instances. Notice that preprocessing uses a large portion of running time for the `logic-synthesis` problems, which however pays off as it shrinks these instances by up to 90 %. However, we would like to point out again that all preprocessing techniques we apply are very straightforward.

instances	solved	total	b & c	subs	nodes	edges	cuts
`logic-synthesis`	16/17	198.30	62.92	160.4	4786.6	10678.2	278.0
`SPOT5/DIR`	18/21	137.73	129.88	2781.9	1110.7	7774.6	375.4
`SPOT5/LOG`	14/21	177.24	121.14	2469.4	1634.8	7952.5	669.7

Table 3: Experimental results for `logic-synthesis` and `SPOT5` instances.

The second example class consists of the weighted `SPOT5` instances. Our results are again very positive: as shown in Table 3, we could solve 18 instances in the subclass `DIR` and 14 in `LOG`. In the Max-SAT Evaluation, the best participant could solve only 6 instances in each class. So it seems that our method is consistently superior to other approaches when applied to very hard SAT instances, in spite of the fact that it is designed for much more general applications.

5. Conclusion

We presented a novel integer programming approach to general nonlinear boolean optimization problems. Unlike other approaches, it avoids adding many artificial variables to the model, at the same time allowing to derive tight linear relaxations of the corresponding polytope. In the special case of binary polynomial optimization, which has been investigated in [3], our approach proved to be very successful in practical experiments. First computational results for hard SAT instances reported in this paper seem to confirm the good performance of our approach also for other nonlinear optimization problems.

References

[1] J. Argelich, C. M. Li, F. Manyà, and J. Planes, “Max-SAT Evaluation 2007.” See `http://www.maxsat07.udl.es`.

[2] F. Barahona and A. R. Mahjoub, “On the cut polytope,” *Mathematical Programming*, vol. 36, pp. 157–173, 1986.

[3] C. Buchheim and G. Rinaldi, “Efficient reduction of polynomial zero-one optimization to the quadratic case,” *SIAM Journal on Optimization*, vol. 18, no. 4, pp. 1398–1413, 2007.

[4] CPLEX 11.0, `www.ilog.com/products/cplex`.

[5] C. De Simone, “The cut polytope and the Boolean quadric polytope,” *Discrete Mathematics*, vol. 79, no. 1, pp. 71–75, 1990.