HORN CARDINALITY RULES

John N. Hooker, Hong Yan

EDRC 70-07-92

Horn Cardinality Rules *

JOHN N. HOOKER
HONG YAN
Graduate School of Industrial Administration
Carnegie Mellon University, Pittsburgh, PA 15213 USA

Revised, June, 1992

Abstract

We address the problem of finding a "tight" representation of Horn cardinality rules in a mixed integer programming model by describing a convex hull of it. A cardinality Horn rule asserts that if at least k of the propositions $A_1, \ldots, A_m$ are true, then B is true. We also show that Horn cardinality rules have properties analogous to ordinary Horn rules.

1 Introduction

As rule-based systems and other types of logic modeling grow in popularity, logical rules and propositions can play an increasingly important role in mathematical programming models. Such simple logical constraints as "if A is produced, then either B or C must be produced" have long been a part of mathematical programming. But much more complex logic models are now being formulated, and they can also be embedded in mathematical programming models.

Propositional Horn formulas have some very attractive properties that account for their popularity in rule-based systems. A Horn inference problem are solvable by the linear programming relaxation. (see, for example, [2], [7]) That is, one can determine the satisfiability of a set of Horn rules simply by checking whether the corresponding LP relaxation is feasible. A Horn clause

*The first author is partially supported by AFOSR grant 91-0287 and ONR grant N00014-92-J-1028. Both authors are partially supported by NSF grant 1-55093.

is a clause with at most one positive proposition. The rules one typically finds in expert systems,

If the propositions $A_1, \ldots, A_m$ are all true, then B is true.

Or

If the propositions $A_1, \ldots, A_m$ are all true.

are Horn rules.

A natural extended class of Horn rules is the Horn *cardinality* rules. A Horn cardinality rule has the form,

If at least (or at most) k of $A_1, \ldots, A_m$ are true, then B is true.

And the *cardinality clauses*

at least (or at most) k of $A_1, \ldots, A_m$ are true.

are special cases. We wish to investigate how the properties of the ordinary Horn rules can be generalized to the Horn cardinality rules.

Furthermore, when logical constraints become a significant component of mathematical programming models, the quality of their representation becomes an important issue. Williams argues in [14] that cardinality conditions are appropriate as a basic form in which to express logical constraints. Therefore, while we consider the properties of the Horn cardinality rules as a whole system, we also wish to investigate the representation of a Horn cardinality rule in an MIP model.

The usual method for representing logical conditions are as the following. They are first rewritten as a conjunction of logical "clauses," which is to say in conjunctive normal form (CNF). A *clause* is a disjunction of atomic propositions or their negations, such as,

$$x_1 \vee \neg x_2 \vee x_3,$$

where $\neg$ means "not," and $\vee$ means "or." Each clause is then written as an inequality in 0-1 variables, which for this example is,

$$x_1 + (1 - x_2) + x_3 \geq 1,$$

where x_j is interpreted as true when $x_j = 1$ and false when $x_j = 0$ This is a very "loose" representation of the logical conditions, in the sense that its linear relaxation (which replaces $x_j \in \{0, 1\}$ with $0 \leq x_j \leq 1$) describes a polytope that has many fractional extreme points. This complicates the solution of the model, since most solution techniques make essential use of the linear relaxation.

The difficulty with CNF representation is not that the individual clauses are poorly represented. In fact, each clause receives the tightest possible representation, namely a *convex hull representation*. This is a set of 0-1 inequalities whose linear relaxation describes the convex hull of the 0-1 points satisfying them. Rather, the difficulty is that there are typically a large number of clauses in CNF. This results in a loose representation of the formula as a whole, even though each individual clause is tightly represented. (R. Jeroslow discusses this principle in [9].) This problem is particularly acute for a cardinality rule, since the number of clauses in its CNF equivalent grows exponentially with the rule's length if no new variables are added (and is apparently quite large even if new variables are added).

Thus when a logical constraint comes in the form of Horn cardinality rule, it is far better to give each a convex hull representation directly than to convert it to CNF first. We will show how to do this. That is, we will state a simple algorithm that generates for any Horn cardinality rule a set of 0-1 inequalities that provide a convex hull representation of it. These inequalities describe the facets of the convex hull.

Several authors have contributed to the inequality representation of logical formulas in CNF. Tseitin [12] showed how to convert any formula of propositional logic to CNF in linear time by adding new variables. Dantzig [3], Blair *et al.* [2], and Williams [13] discussed the use of 0-1 inequalities to represent logical constraints in CNF, and Karp [10] used them to show that integer programming is NP-hard. Hadjiconstantinou and Mitra [5] described an algorithm for the automatic conversion of logical formulas, including cardinality clauses, into inequality form. But they do not consider the tightness of the representation. Hooker [8] described a generalized resolution procedure that generates all the undominated implications of a set of cardinality clauses. In research that proceeded concurrently with ours, Araque and Chandru found a convex hull representation of cardinality clauses.

This paper is organized as the following. Section 2 states our algorithm for generating the convex hull description of a Horn cardinality rule and proves its correctness. In Section 3 we show that they have a natural generalization in cardinality logic with precisely the same properties. We also propose our future related works.

2 CNF OF HORN CARDINALITY RULES

A Horn cardinality rule is written,

$$(A_1, \ldots, A_m)_k \Rightarrow B, \tag{1}$$

and is read, "if at least k of $A_1, \ldots, A_n$ are true, then B is true." We assume $m \geq k \geq 0$ and the phrase "at least" can be replaced with "at most" by writing,

$$(\neg A_1, \ldots, \neg A_m)_{m-k} \Rightarrow \neg B.$$

Ordinary Horn rules have $k = m$. When $m = 1$ we will omit the parentheses, so that $(A)_1 \Rightarrow B$ is written $A \Rightarrow B$.

When $m = 0$, (1) becomes a cardinality clause asserting that B is true. When $B = 0$, it asserts that fewer than k of $A_1, \ldots, A_m$ are true.

We first show that Horn cardinality rules have an exponential CNF expansion.

Theorem 1 *No CNF formula equivalent to a Horn cardinality rule*

$$(x_1, \ldots, x_m)_k \Rightarrow y. \tag{2}$$

whose variables are in $\{x_1, \ldots, x_m, y\}$ *has fewer than* $\binom{m}{k}$ *clauses.*

Proof. Let F be any CNF equivalent. Let $(x^*, y^*) = (x_1^*, \ldots, x_m^*, y^*) \in \{0,1\}^{(m+1)}$ be a *minimal violator* of (2) if it violates (2) but would satisfy it if any x_j^* equal to 1 were switched to 0. There are $\binom{m}{k}$ minimal violators, since each has exactly k x_j^*'s equal to 1, and $y^* = 0$. Every minimal violator (x^*, y^*) must violate some clause C_{x^*} in F. We show that every C_{x^*} is necessarily distinct, from which the theorem follows.

Since (x^*, y^*) violates C_{x^*}, C_{x^*} cannot contain the positive literal x_j when $x_j^* = 1$. But since (x^*, y^*) is a *minimal* violator, C_{x^*} must contain x_j for each $x_j^* = 0$. It follows that the positive literals in C_{x^*} are precisely those for which $x_j^* = 0$, which implies that every C_{x*} is distinct. □

When additional variables are used, there is a polynomial CNF expansion, but it is still long. We will show this by example using the pigeon hole principle, which Haken used to prove that the resolution method of theorem proving has exponential complexity [6]. To convert the formula,

$$(x_1, x_2, x_3, x_4)_3 \Rightarrow y \tag{3}$$

to CNF, we create new variables z_{ij} for $i = 1, \ldots, 4$ and $j = 1, 2$. The CNF equivalent is the conjunction of the following clauses.

$$
\begin{array}{ccccccccccccc}
\neg x_1 & & & & \vee y & \vee z_{11} & & & & \vee z_{12} & & & \\
 & \neg x_2 & & & \vee y & & \vee z_{21} & & & & \vee z_{22} & & \\
 & & \neg x_3 & & \vee y & & & \vee z_{31} & & & & \vee z_{32} & \\
 & & & \neg x_4 & \vee y & & & & \vee z_{41} & & & & \vee z_{42} \\
 & & & & & \vee \neg z_{11} & \vee \neg z_{21} & & & & & & \\
 & & & & & \vee \neg z_{11} & & \vee \neg z_{31} & & & & & \\
 & & & & & \vee \neg z_{11} & & & \vee \neg z_{41} & & & & \\
 & & & & & & \vee \neg z_{21} & \vee \neg z_{31} & & & & & \\
 & & & & & & \vee \neg z_{21} & & \vee \neg z_{41} & & & & \\
 & & & & & & & \vee \neg z_{31} & \vee \neg z_{41} & & & & \\
 & & & & & & & & & \vee \neg z_{12} & \vee \neg z_{22} & & \\
 & & & & & & & & & \vee \neg z_{12} & & \vee \neg z_{32} & \\
 & & & & & & & & & \vee \neg z_{12} & & & \vee \neg z_{42} \\
 & & & & & & & & & & \vee \neg z_{22} & \vee \neg z_{32} & \\
 & & & & & & & & & & \vee \neg z_{22} & & \vee \neg z_{42} \\
 & & & & & & & & & & & \vee \neg z_{32} & \vee \neg z_{42}
\end{array}
\qquad (4)
$$

We interpret the formula (3) as saying that if we have 3 or more pigeons (i.e., 3 or more x_j's are true), then we cannot put them in 2 holes with at most one per hole (y is true). In (4), $z_{ij} = 1$ is interpreted as saying that pigeon j is put in hole i. The first four clauses of (4) say that every pigeon is put into a hole, or else y is true. The remaining clauses say that no 2 pigeons are put into the same hole. Thus (4) forces y to be true precisely when we try to put 3 or more pigeons in 2 holes, just as (3) does.

In general, (2) is read as saying that if we have k or more pigeons, we cannot put them in $k - 1$ holes. Thus the above method of producing a CNF equivalent generates $m(k-1)$ additional variables and $m + (1/2)(k-1)m(m-1)$ clauses. We are aware of no shorter conversion.

3 Convex Hull Representation of Horn Cardinality Rules

In this section, we describe the facets of the convex hull of the feasible points of a Horn cardinality rule. Consider Horn cardinality rules having form (2) Denote by S the set of feasible points satisfying (2), and by $conv(S)$ the convex hull of S. We begin by giving the equivalent 0-1 linear inequality representation of a Horn cardinality rule.

Lemma 1 *The 0-1 linear inequality is an equivalent representation of (2):*

$$-ex + (1 + m - k)y \geq 1 - k. \tag{5}$$

Proof: If (2) is true, then either $y = 1$ or $ex < k$ (or both). It is easily checked that (5) is satisfied in either cases. Furthermore, if (2) is false, then $y = 0$ and $ex \geq k$, which violates (5). □

The following lemma starts our description of the convex hull of (2).

Lemma 2 *When $k \geq 2$ or $m = k = 1$, (5) defines a facet of $conv(S)$.*

Proof: Lemma (1) showed that (5) is valid for (2). The case $m = k = 1$ is trivial. When $k \geq 2$, we need to find $m + 1$ affinely independent points in S such that

$$-ex + (1 + m - k)ey = (1 - k)$$

We construct the first m points satisfy $ex = k - 1$ and $y = 0$, and another point satisfying $ex = m$ and $y = 1$ as the following.

Let A be a $k \times k$ matrix of all ones except for zeros on the diagonal. Let O be a $k \times (m - k)$ zero matrix. B is a $(m - k) \times k$ 0-1 matrix with $k - 2$ ones in each row. And I is a $(m - k) \times (m - k)$ identity matrix.

$$D = \begin{pmatrix} & e & & 1 \\ A & & O & 0 \\ B & & I & 0 \end{pmatrix}.$$

Then the rows of D are the $m + 1$ points required. To show that they are affinely independent, it suffices to show that $det(D) \neq 0$. It is clear that

$$det(D) = det(A)$$

To compute $det(A)$, add all other rows of A to the first row and then subtract all other rows from the new first row divided by $(k - 1)$. Then it is clear that $det(A) = (-1)^{k-1}(k - 1)$. The lemma follows. □

To describe the entire set of facets, we first show that the inequality (5) is the only facet-defining inequality,

$$ax + \beta y \geq c, \tag{6}$$

in which each $a_j \neq 0$ and $\beta \neq 0$. We then obtain the remaining facets recursively by showing that they are the facets of simpler cardinality rules. To do the latter we exploit the facts a) that the remaining facets are facets of the convex hull's projections onto lower dimensional spaces, and b) that each of these projections is itself the convex hull description of a simpler rule.

Lemma 3 *Assume (6) defines a facet of conv(S). If $a_i \neq 0$, for all i, then (6) is a nonzero scalar multiple of (5).*

Proof: We first show $a_i < 0$ for all i. Since (6) defines a facet of $conv(S)$, it must contain a set T of $m+1$ affinely independent points in S. For any $i \in \{1, \ldots, m\}$ we know that some $(\overline{x}, \overline{y}) \in T$, has $\overline{x}_i = 1$. Otherwise the facet is defined by $x_i \geq 0$. Since $(\overline{x}, \overline{y})$ satisfies (2), it must either falsify its antecedent or satisfy its consequent. In either case a point $(x', \overline{y})$ identical to $(\overline{x}, \overline{y})$ except that $x'_i = 0$ also satisfies (2), so that $ax' + \beta\overline{y} \geq c$. Subtracting $a\overline{x} + \beta\overline{y} = c$ from this, we get $a_i < 0$. Without loss of generality, assume $a_1 \leq a_2 \leq \cdots \leq a_m < 0$.

We then show that for any $(x, y) \in T$, $y = 1$ implies $ex = m$. Consider a point $(\overline{x}, 1) \in T$. If $e\overline{x} < m$, (that is, $\overline{x}_t = 0$ for some t), then a point $(x', \overline{y})$ identical to $(\overline{x}, 1)$ except $x'_t = 1$ satisfies (2). Thus $ax' + \beta \geq c$, which implies $a_t > 0$, a contradiction. It is clear that points $(x, y) \in T$ with $ex = m$ satisfy $y = 1$, since T contains only feasible points.

We can also show that for any $(x, y) \in T$, $y = 0$ implies $ex = k - 1$. Consider a point $(\overline{x}, 0) \in T$. We must have $e\overline{x} \geq k - 1$. And if $e\overline{x} < k - 1$, we have as above that $a_t > 0$ for some t. Thus $e\overline{x} = k - 1$.

It follows that the points in T can be partitioned into subsets T_1 and T_2, with $ex = k - 1$ and $y = 0$ for all $(x, y) \in T_1$, and $ex = m$ and $y = l$ for all $(x, y) \in T_2$. Thus there are m points in T_1 and 1 point in T_2.

Now we show that $a_i = \alpha\ \forall i$. First consider the points in T_1. Set $a_m = \alpha$. We will build recursively a set $Q \subseteq \{1, \ldots, m\}$ of indices such that $a_i = \alpha$ for all $i \in Q$. Initially $Q = \{m\}$, and we will augment Q until $Q = \{1, \ldots, m\}$.

At each step of the recursion we have $Q = \{q, \ldots, m\}$. Let T_1^q be the set of points in T_1 satisfying $x_i = 1$ for some $i \in \{q, \ldots, m\}$. Since T_1 contains m independent points with $y = 0$, T_1^q must be nonempty. Let $(\overline{x}, 0) \in T_1^q$ be a point satisfying $\overline{x}_t = 0$ for some $t < q$. We suppose for the moment that such a point exists. Then the point $(x', 0)$ identical to $(\overline{x}, 0)$ except $x'_t = 1$ and $x'_i = 0$ for some $i \in Q$ with $\overline{x}_i = 1$. Then $(x', 0)$ satisfies $ax' \geq c$, since $ex = k - 1$. This implies $a_t \geq a_i$. Since by assumption $a_t \leq a_{t+1} \leq \ldots \leq a_i$, we have $a_t = a_{t+1} = \ldots = a_i = \alpha$, and we update $Q = \{t, t+1, \ldots, m\}$.

It remains to show, then, that there is a point $(\overline{x}, 0) \in T_1^q$ with $x_t = 0$ for some $t < q$ if $q \neq 1$. Suppose to the contrary, We consider two cases:

Case 1. $T_1 = T_1^q$. Then all the points in T_1 have $x_p = 1, \forall p < q$, since otherwise $(\overline{x}, 0)$ exists. But this implies for $p < q$ that $x_p = 1$ for all points in T, which is impossible since otherwise $x_p \leq 1$ would define the facet.

Case 2. $T_1 \neq T_1^q$. Consider any point $(\overline{x}, 0) \in T_1 \setminus T_1^q$. Since it satisfies $e\overline{x} = k-1$, we must have $q > k-1$. On the other hand, consider any point $(x, 0) \in T_1^q$. By hypothesis $x_p = 1$ for all $p < q$, we must therefore have $ex \geq k$, which is impossible $(x, 0) \in T_1$.

We conclude that Q can be updated until $q = 1$. Thus $a_1 = \ldots = a_m = \alpha$.

Therefore, (6) becomes

$$\alpha ex + \beta y \geq c \tag{7}$$

Substituting into (7) a point $(x, y) \in T_1$, we get

$$\alpha(k-1) = c. \tag{8}$$

Doing the same for $(x, y) \in T_2$, we get,

$$\alpha m + \beta = c \tag{9}$$

Thus

$$\beta = -\alpha(m - k + 1)$$

Substituting this and (8) into (7), we have

$$\alpha ex - \alpha(m - k + 1)y \geq \alpha(k-1). \tag{10}$$

Since $\alpha < 0$, the lemma follows. □

The following lemma says that any facet without y is a x variable boundary.

Lemma 4 *Let conv(S) have a facet*

$$ax \geq c. \tag{11}$$

Then there exists an index j such that (11) is equivalent to $x_j \geq 0$ or $x_j \leq 1$.

Proof: Since (11) is feasible, all points with $y = 1$ satisfy it. And since it does not contain y, it therefore has all points feasible. A facet containing all 2^{m+1} points has to be a variable boundary. That is, $x_i \geq 0$ or $x_i \leq 1$ for some i. □

Lemma 5 *If $k < m$, any facet of conv(S) other than (5) is a facet of the projection of conv(S) onto $(x_{i_1}, \ldots, x_{i_{m-1}}, y)$ for some $\{i_1, \ldots, i_{m-1}\} \subseteq \{1, \ldots, m\}$.*

Proof: Consider a facet F of $conv(S)$ other than (5). Thus at least one variable is missing from the inequality describing F. By lemma (4), this missing variable can not be y. Without loss of generality, assume it is x_1. We wish to show that F is a facet for the projection P of $conv(S)$ onto $(x_2, \ldots, x_m, y)$. F is clearly valid for P. So it suffices to show that there are m affinely independent points of P on F.

Consider $m+1$ independent points of S on F. Let A be the $(m+1) \times (m+1)$ 0-1 matrix with each point as a row. Hence $det(A) \neq 0$. On the other hand, $det(A)$ can be expended in the cofactors of its first column:

$$det(A) = a_{11}A_{11} + \cdots + a_{m+1,1}A_{m+1,1}$$

Clearly, each of the cofactors A_{j1} corresponds to a set of m vectors in P. Since $det(A) \neq 0$, we must have that at least one of these cofactors is non-zero. This implies that at least one set of m points in P are independent. Furthermore, they are all on F since F does not contain x_1. Thus, F is a facet of P. □

Lemma 6 *For $k < m$, the projection of S onto $(x_{i_1}, \ldots, x_{i_{m-1}}, y)$ is precisely the set of 0-1 points satisfying $(x_{i_1}, \ldots, x_{i_{m-1}})_k \Rightarrow y$.*

Proof: Without loss of generality, we again assume

$$(x_{i_1}, \ldots, x_{i_{m-1}}) = (x_2, \ldots, x_m).$$

Now denote by $P_1(S)$ the projection of S onto $(x_2, \ldots, x_m)$. Also let

$$S_1 = \{(x, y) | x \in \{0,1\}^{m-1}, y \in \{0,1\}, (x_2, \ldots, x_m)_k \Rightarrow y\}.$$

We wish to show $S_1 = P_1(S)$.

Take a point $(x_2, \ldots, x_m, y) \in S_1$. It is clear that for either $y = 1$ or $y = 0$ we have $(0, x_2, \ldots, x_m, y) \in S$. Thus $(x_2, \ldots, x_m, y) \in P_1(S)$. That is, $S_1 \subseteq P_1(S)$. Now, take $(x_2, \ldots, x_m, y) \in P_1(S)$. Then there exists an x_1 such that $(x_1, x_2, \ldots, x_m, y) \in S$. If $y = 1$, clearly $(x_2, \ldots, x_m, 1) \in S_1$. If $y = 0$, we know that at most $k-1$ of $(x_1, x_2, \ldots, x_m)$ can be 1. Therefore, at most $k-1$ of $(x_2, \ldots, x_m)$ can be 1. That is, $(x_2, \ldots, x_m, y) \in S_1$. Thus $P_1(S) \subseteq S_1$. □

From the results of lemmas 2-6, we have showed the main result of this section. That is, the full description of the convex hull of the general cardinality rule can be stated as follows.

Theorem 2 *If $m > k \geq 2$, the facets of $conv(S)$ are*

$$-(x_1 + \cdots + x_m) + (m - k + 1)y \geq 1 - k,$$

plus the facets of

$$(x_{i_1}, \ldots, x_{i_{m-1}})_k \Rightarrow y$$

for all sets $\{i_1, \ldots, i_{m-1}\} \subseteq \{1, \ldots, m\}$. □

Theorem 3 *If $k = m$, the facets of $conv(S)$ are*

$$-x_1 - \cdots - x_m + y \geq 1 - m, \tag{12}$$

plus $x_i \geq 0$, $x_i \leq 1$ for all i and $y \geq 0$ $y \leq 1$ provided that $m \geq 2$.

Proof: When $k = m$, the rule is an ordinary clause. Lemma 2 implies that (12) defines a facet. It is easily shown that all the bounds also define facet when $m \geq 2$. The case $m = 1$ is obvious. □.

Theorem 4 *If $m > k = 1$, the facets of the $conv(S)$ are the facets of*

$$x_i \Rightarrow y, \tag{13}$$

for $i = 1, \ldots, m$.

Proof: We only need to show that

$$-ex + my \geq 0 \tag{14}$$

does not define a facet of $conv(S)$.

If it does, it should contain $m + 1$ affinly independent points. Consider all the feasible points on $-ex + y = 0$. It is clear that there are only two points $(ex, y) = (m, 1)$ or $(0, 0)$. Thus (14) can not be a facet. Theorem 2 implies the result. □

We will see in next section, that a set of Horn cardinality rule keeps all the nice properties of the ordinary Horn rule. Particularly, the inference problem on a set of Horn cardinality rules can be solved by linear programming. And a linear time algorithm is given for solving the cardinality Horn satisfiability problem.

4 Properties of Horn Cardinality Rules

We have described the convex hull of the Horn cardinality rule as a special case of the general cardinality rule in the last section. On the other hand, it is nature to see that the Horn cardinality rule is an extension of the ordinary Horn rule. Horn inference problem are solvable by the linear programming relaxation. (see, for example, [2], [7]) That is, one can determine the satisfiability of a set of Horn rules simply by checking whether the corresponding LP relaxation is feasible. In this section, we will show that Horn cardinality rules also has this property.

Consider a data base in Horn cardinality rules,

$$CF = (CH_1, \cdots, CH_m) \tag{15}$$

with propositions $(x_1, \cdots, x_n)$. In rule CH_j, denote m_j the number of antecedents, k_j the minimum number of antecedents required to imply the consequence, and S_j the set of antecedent propositions in CH_j. We also use x_{j+} to denote the consequence proposition in CH_j.

From the last section, we know that a Horn cardinality rule has an equivalent 0-1 linear inequality representation. Thus, the data base (15) has an equivalent 0-1 linear inequality system

$$\sum_{i \in S_j} x_j + (1 + m_j - k_j) x_{j+} \geq 1 - k_j, j = 1, \cdots, m, \tag{16}$$

where x_i's are 0-1 variables. We show that CH is true if and only if the linear relaxation of (16) is feasible with the variable boundaries.

This property is closely connected with the unit resolution, or the chaining procedure. However, the normal unit resolution does not work for the cardinality problem. Here, we describe another chaining procedure for Horn cardinality data base CF. We call the procedure *positive chaining* since it is restricted to fixing variables to TRUE.

Positive Chaining Procedure. Pick any positive unit rule l, x_i, of CF; if there is none, exit the procedure. If l is the negation of another unit rule of CF, CF is inconsistent; exit the procedure.

Otherwise fix $x_i = true$, and remove x_i from CH_j if x_i is a antecedent in CH_j and set $k_j = k_j - 1$. If $k_j = 0$, remove all the remaining antecedent variables from CH_j, and x_{j+} is left as a positive unit rule.

Denote (16) simply by $Ax \geq a$. Then we have the following important property analogous to the system of ordinary Horn clauses.

Theorem 5 *A system $Ax \geq a$ of Horn cardinality rules is satisfiable if and only if the positive chaining detects no inconsistency, and if and only if the linear system $Ax \geq a$, $0 \leq x \leq 1$ is feasible.*

Proof. It is obvious that if the rules are true, then the linear system is feasible.

If positive chaining detects no inconsistency and the procedure is terminated. Then a unit clause must have negative form. And all other clause will have at least two literal, one of which is a negated variable. Thus the original system can be satisfied by setting remaining variables to 0, and all the variables fixed in positive chaining are fixed to 1. That is, if positive chaining detects no inconsistency, then the rules must be consistent.

Finally, we show that if the positive chaining detects inconsistency, the linear system is infeasible. In fact, when the positive chaining fixes a variable x_i to true, the corresponding linear constraint $x_i \geq 1$ fixes x_i to 1 along with the bond $x_i \leq 1$. Thus if the positive chaining had found inconsistency, that means there is a pair of constraints in the linear system: $x_i \geq 1$ and $-x_i \geq 0$. This implies that the LP would not have been feasible. □

The linear time algorithm for testing the ordinary Horn satisfiability problem was first proposed by Dowling and Gallier [4]. Minoux gave a simpler graph representation of Horn clauses, and a new version of linear time ($O(K)$, where K is number of literals) algorithm [11]. By modifying the Minoux's algorithm, we give a similar linear time algorithm for testing the Horn cardinality problem. Note that the Horn cardinality inference differences from the ordinary Horn inference only in that a Horn cardinality rule has an extra parameter k which indicates the minimum number of antecedents required to imply the consequence.

Given a set of Horn cardinality rules CF, a directed graph $G(CF)$ associated with it is constructed as follows:

- it contains $n+2$ nodes, one node i for each variable x_i $(i = 1, 2, \ldots, n)$ plus one node t (standing for "true") and one node f (standing for "false"),
- if the j-th rule CH_j contains a single variable x_i in positive form, then there is an arc $u = (t, x_i)$ with label $l(t, x_i) = j$,

- if the j-th rule CH_j is of the form $(x_{i_1}x_{i_2}\ldots x_{i_m})_k$, then $G(CF)$ contains q arcs (x_{i_j}, f) with labels equal to j, $l(x_{i_j}, f) = j$, $r = 1, 2, \ldots, q$,
- if the j-th rule CH_j is of the form $(x_{i_1}x_{i_2}\ldots x_{i_m})_k \Rightarrow x_p$, then $G(CF)$ contains q arcs (x_{i_j}, x_p) with label equal to j and node, $r = 1, \ldots, q$. Associated with node p, assign a valve, $val(p) = k$.

Now, we present the algorithm as follows.

Algorithm CH

step 0. Initialization

Set up $G(CF)$ associated with the given Horn cardinality rules.
For each rule CH_j ($j = 1, 2, \ldots, m$) compute k_j, m_j and set
$S = \{j | m_j = 0\}$,
set $val(i) = 1$, if x_i in CH_j and $CH_j \in S$, $val(i) = 0$ else.

step 1

If $S = \emptyset$, terminate, set all remaining free variables to 0 and all fixed variables to 1. SAT.
If $S \neq \emptyset$, pick $j \in S$ and $S \leftarrow S \setminus \{j\}$.

step 2

If CH_j has no consequence, terminate. Inconsistency is detected.
Else, let x_i be the consequence variable in rule CH_j.
For every arc (x_i, y) with label $r = l(x_i, y)$ in $G(CF)$, do
    $k_r = k_r - 1$
    If $k_r = 0$ and $y = f$, TERMINATE. UNSAT.
    If $k_j = 0$ and $y = x_r$, with $val(r) = 0$ do
        set $m_r = 0$,
        $S = S \cup \{r\}$,
        $val(r) = 1$.
go to *step 1*.

This algorithm is implemented basically as a labelling process. And each arc in $G(CF)$ is examined at most once. Therefore, it is clear that the complexity is $O(K)$, where K is number of arcs, or number of occurrences of literals in the Horn cardinality rules.

5 FUTURE WORKS

A wider and useful class of logic rules are cardinality rules, having the form

$$(A_1, \ldots, A_m)_k \Rightarrow (B_1, \ldots, B_n)_l \tag{17}$$

which says that "if at least k of $A_1, \ldots, A_m$ are true, then at least l of $B_1, \ldots, B_n$ are true." We have observed that logical constraints take this form in a large variety of applications. Also this is a nature generalization of ordinary clauses. We wish to investigate the representations of this rule when we consider to use such logical constraints in mathematical programming models. We also need to know how to convert a logical rule in to a conjunction of cardinality rules.

References

[1] C. Blair. Two rules for deducing valid inequalities for 0-1 problem. *SIAM Journal of Applied Mathematics*, 31, (1976) 614-617.

[2] C. Blair, R.G. Jeroslow and J.K. Lowe. Some results and experiments in programming techniques for propositional logic. *Computer and Operations Research.*

[3] G.B. Dantzig. Linear Programming and Extensions. Princeton University Press, 1963.

[4] W.F. Dowling and J.H. Gallier. Linear time algorithms for testing the satisfiability of Horn formulae. *Journal of Logic Programming* 3 (1984) 267-284.

[5] E. Hadjiconstantinou and G. Mitra, Transformation of propositional calculus statements into integer and mixed integer programs: An approach towards automatic reformulation, manuscript. Brunel University, Uxbridge, Middlesex, U.K., 1990.

[6] A. Haken. The intractability of resolution. *Theoretical Computer Science*, 39 (1985) 297-308.

[7] J.N. Hooker. A quantitative approach to logical inference. *Decision Support System*, 4 (1988) 45-69.

[8] J.N. Hooker. Generalized resolution for 0-1 linear inequalities. ???

[9] R.G. Jeroslow. Representability in mixed integer programming: Characterization results. *Discrete Applied Mathematics*, 17:223–243, 1987.

[10] R.M. Karp. Reducibility among combinatorial problems. *Complexity of computer computations*, R.E. Miller and J.W. Thatcher, eds. Plenum Press (1972) 85-103.

[11] M. Minoux. LTUR: A simplified linear-time unit resolution algorithm for Horn formulae and computer implementation. *Information Processing Letters* 29 (1988) 1-12.

[12] G.S. Tseitin. On the complexity of derivations in the propositional calculus. *Structures in Constructive Mathematics and Mathematical Logic, Part II* A.O. Slisenko, ed., (translated from Russian, 1968) 115-125.

[13] H.P. Williams. *Model Building in Mathematical Programming.* Wiley, Chichester, 1988.

[14] H.P. Williams. Constructing integer programming models by the predicate calculus. *Annals of Operations Research*, 21 (1989) 227-246.