On the polytope of the at-least-m-different predicate

Niko Kaso Serge Kruk*

May 6, 2019

Abstract

Constraint Programming (CP) is method used to model and solve complex combinatorial problems. An alternative to Integer Programming for solving large scale industrial problems it is, under some circumstances, more efficient than IP but its strength lies mainly in the use of predicates to model problems. This paper presents the at-least-m-different predicate and provides a class of facet-defining inequalities of the convex hull of integer solutions. This predicate bounds the number of values that variables in a set may receive. The paper also presents a polynomial time separation algorithm to be used in the context of a branch-and-bound optimization approach.

1 Introduction

Research on Constraint Programming(CP) has a long tradition and growing interest since it is widely used to model and solve many combinatorial problems. Its applications vary in several different fields of interest such as, artificial intelligence, networks, planning, vehicle routing and scheduling. CP is an alternative to Integer Programming (IP) for modeling optimization problems. Although CP and IP originate from two different areas of science, considering their synergy has proven quite useful in practice and theory. While IP originates from Operations Research (OR) in Mathematics, which helps in making better decisions by applying advanced analytical methods, CP was first introduced and mostly developed by the Artificial Intelligence (AI) community.

Although CP approaches, where predicates are used, have a simpler representation of any given optimization problem than IP and allow nearly

*[kaso,kruk]@oakland.edu, Department of Mathematics and Statistics, Oakland University, Rochester, MI.

CONGRESSUS NUMERANTIUM 234 (2019), pp.47-60.

anybody (regardless the background) to understand the formulation of the problem, the models often lead to intractability. This drawback can make it hard to handle real-world problems by using CP only.

On the other hand, IP approaches are quite useful because of their potential to search and identify locally optimal and optimal solutions for a given space where solutions of a certain problem are bounded. However, IP tactics can only be applied successfully by experienced modelers and the models produced are often impenetrable.

In addition IP techniques rely on relaxation techniques and polyhedral analysis, whereas in CP, inference techniques are used to forbid the search of the irrelevant sections of the solution search space. Therefore it has proven fruitful to consider an interaction of both approaches. Considering them simultaneously may yield increased expressiveness and yet provides satisfactory and quick solutions. Over the last two decades, several publications such as, [12], [2] and [11], have exhibited the development of the Integration of AI and OR techniques.

In this paper, pursuing the approach of earlier work [7] and [6], the objective is to provide a useful class of facet-defining inequalities for another predicate, at-least-m-different. Similar work on polytope representations has been done on [7] and [13], where the polytope of the at-least and all-different predicates is fully characterized. Since both predicates have a wide range of applications they have been more studied and therefore extended to the interaction of multiple predicates. Unlike [7] and [13], and as in [10] and [1], this paper does not provide a complete characterization of the polytope of integer solutions of the at-least-m-different predicate but a large class of valid inequalities, as is done in [5], [4], [9] and [8], where some special cases of multiple all-different predicate were considered.

In Section 2, it is provided a full characterization of the polytope for a small dimensional instance of the predicate. In Section 3, two sets of inequalities are given. For the first set of inequalities it is proved that those are facet-defining inequalities for the convex hull of integer solutions of at-least-m-different predicate, while for the second set is only proved validity. In Section 4, some computational results experimentally justify the value of the inequalities. And finally, in Section 5, a separation algorithm is presented.

A motivational example of the at-least-m-different predicate is the following: Let's assume the following optimization sub-problem is embedded in a larger problem:

$$
\begin{aligned}
&\text{maximize } c^T x \\
&\text{subject to } Ax \leq b \\
&\qquad\qquad x \geq 0 \\
&\qquad\qquad \text{all-different}\{x_1, \ldots, x_{10}\}
\end{aligned}
$$

And let us assume that this larger problem is deemed infeasible. It may prove useful to consider a relaxation of the sub-problem as follows:

$$
\begin{aligned}
&\text{maximize } c^T x \\
&\text{subject to } Ax \leq b \\
&\qquad\qquad x \geq 0 \\
&\qquad\qquad \text{at-least-9-different}\{x_1, \ldots, x_{10}\}
\end{aligned}
$$

If this relaxation still proves to be infeasible we can continue to decrease the number of variables needed to be different, until we get a feasible solution.

For future reference, we provide below the formal definition of the at-least-m-different predicate.

Definition 1.

$$
\begin{aligned}
\text{at-least-m-different}\{x_1, x_2, \ldots, x_n\} \Leftrightarrow \quad & \\
& x_i \in \{0, \ldots, l\}, \quad i \in \{1, 2, \ldots, n\}, \\
& J \subseteq \{1, \ldots, n\}, \\
& |J| = m, \\
& x_k \neq x_l \quad \forall k, l \in J.
\end{aligned}
$$

Informally, at least m of the n variables, with integral domain $\{0, \ldots, l\}$, are different.

We aim to characterize as much as possible the convex hull of integer solutions of the at-least-m-different constraint to help an IP solver to cut the search space.

2 Polytope of a small dimensional example.

Before we proceed with the most general formulation possible, we illustrate the facets of the convex hull of a small-dimensional example, a single predicate at-least-4-diff$\{x_1, x_2, x_3, x_4, x_5, x_6\}$, where $x_i \in [0, 5]$. We obtained these facets experimentally, using the well-known algorithm of Fukuda [3].

To simplify notation, given a set S, we use P_k^S to mean the set of subsets of S of size k. We also use $[n]$ to mean $\{1, 2, \ldots, n\}$.

The complete set of facets is

$$\begin{aligned}
1 &\le x_{i_1} + x_{i_2} + x_{i_3} + x_{i_4} \le 19, && \{i_1, i_2, i_3, i_4\} \in P_4^{[6]} \\
3 &\le x_{i_1} + x_{i_2} + x_{i_3} + x_{i_4} + x_5 \le 22, && \{i_1, i_2, i_3, i_4, i_5\} \in P_5^{[6]} \\
6 &\le x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \le 24, && \\
-14 &\le x_{i_1} + x_{i_2} - x_{i_3} - x_{i_4} - x_{i_5} \le 9, && \{i_1, i_2, i_3, i_4, i_5\} \in P_5^{[6]} \\
-17 &\le 2x_1 + 2x_2 - x_3 - x_4 - x_5 - x_6 \le 17, && \\
-12 &\le 2x_1 + 2x_2 + 2x_3 - x_4 - x_5 - x_6 \le 27, && \\
x_i &\ge 0, && \\
x_i &\le 5. &&
\end{aligned}$$

3 A class of facet-defining inequalities

In this section we aim to partially define the polytope of the predicate at-least-m-different$\{x_1, x_2, \ldots, x_n\}$ by displaying two classes of inequalities. The first set of inequalities can be expressed with the aid of

$$B := \frac{(m - n + k - 1)(m - n + k)}{2}$$

as:

$$B \le \sum_{i \in \sigma} x_i \le kl - B, \forall \sigma \in P_k^{[n]}, n - m + 2 \le k \le n. \tag{1}$$

We now proceed to the results of this article by first considering the validity of the above inequalities.

Theorem 1. *The set of inequalities* (1) *is valid for all integer solutions to the predicate* at-least-m-different$\{x_1, x_2, \ldots, x_n\}$.

Proof. Since we are dealing with linear inequalities and every coefficient in front of each variable is 1, it is enough to prove validity for an inequality having as its variables the first k-variables for a specific $k \in \{n-m+2, \ldots, n\}$, i.e., $\frac{(m-n+k-1)(m-n+k)}{2} \le x_1 + x_2 + \ldots + x_k \le kl - \frac{(m-n+k-1)(m-n+k)}{2}$. Every inequality with k-variables is just another k-permutation of the n variables $\{x_1, x_2, \ldots, x_n\}$ which take value in $[0, l]$. In other words, if (1) is a valid inequality for the predicate, this implies the validity of any other inequality obtained as a permutation of size k of the n variables. Because of symmetry, let's consider only the second inequality of the two-chain inequalities. Since $k \ge n - m + 2$, then the $\min\{m, n-k\} = n - k$ $\forall m, n, k$. The last equality helps in considering the maximum the left-hand side can achieve, because we can assume that all the variables which do not appear in the inequality take all different values starting from 0 to $n-k-1$. For the k variables of the left-hand side we need exactly $m-n+k-1$ variables to have different values from the domain with the highest being $l-1$ and the others each one less than the preceding, since we can consider the variables sorted without loss of generality. Thus, so far we have got $m-1$ variables taking all different values less than l. Now, in order for the left-hand side to take the maximum possible value the $n-m+1$ other variables need to take value l. A typical instance of a k-dimensional point that ensures the maximum of the left-hand side of the inequality is:

$$(\overbrace{l, l, \ldots, l}^{n-m+1}, l-1, l-2, \ldots, l-(k-(n-m+1)))$$

After adding up all the coordinates of this point we obtain the right-hand side of the inequality. Obviously, if any other variable takes a larger or lower value then the condition that at least m of them must be different fails to be satisfied. So, the solution above attains the maximum value of the linear expression $\sum_{i=1}^{k} x_i$. □

Theorem 2. *The inequalities* (1) *are facet-defining for the convex hull of integer solutions of the predicate* at-least-m-different$\{x_1, x_2, \ldots, x_n\}$.

Proof. In order to prove that the valid inequalities given in (1) are facets we need to find n affinely independent points that satisfy the inequality with equality. Without loss of generality let's consider the inequality $\sum_{i=1}^{k} x_i \leq kl - \frac{(m-n+k-1)(m-n+k)}{2}$. As described at the proof of validity for this inequality, a k-dimensional point that satisfies the predicate and achieves the maximum of the left-hand side is

$$(\overbrace{l, l, \ldots, l}^{n-m+1}, l-1, l-2, \ldots, l-(k-(n-m+1)))$$

Any other n-dimensional point which on the first k coordinates is not a permutation of the above point is either a non-valid point for the predicate or it yields a strictly smaller left-hand side. So, for the above type of inequality the set of affinely independent points on the hyperplane would start off with an extension of the given point and the rest will be derived as follows:

$$(\overbrace{l, l, \ldots, l}^{n-m+1}, l-1, l-2, \ldots, l-(k-(n-m+1)), n-k-1, \ldots, 0)$$

The above point has exactly m different values, since no two coordinates share the same value due to inequality $n-k-1 < l-(k-(n-m+1))$.

Then, we obtain $n-m+1$ other points by shifting the value $l-1$ to the left by one unit until it reaches the first coordinate;

$$(\overbrace{l,l,\ldots,l}^{n-m},l-1,l,l-2,\ldots,l-(k-(n-m+1)),n-k-1,\ldots,0)$$
$$(\overbrace{l,l,\ldots,l}^{n-m-1},l-1,l,l,l-2,\ldots,l-(k-(n-m+1)),n-k-1,\ldots,0)$$
$$\vdots$$
$$(l-1,\overbrace{l,l,\ldots,l}^{n-m+1},l-2,\ldots,l-(k-(n-m+1)),n-k-1,\ldots,0)$$

For the next set of points, from the second point we shift the $n-m+2$ coordinate which corresponds to the value l by one unit to the right until it reaches the k-th coordinate. Thus, we obtain $k-n+m-2$ other points as follows:

$$(\overbrace{l,l,\ldots,l}^{n-m},l-1,l-2,l,\ldots,l-(k-(n-m+1)),n-k-1,\ldots,0)$$
$$(\overbrace{l,l,\ldots,l}^{n-m},l-1,l-2,l-3,l,\ldots,l-(k-(n-m+1)),n-k-1,\ldots,0)$$
$$\vdots$$
$$(\overbrace{l,l,\ldots,l}^{n-m},l-1,l-2,\ldots,l-(k-(n-m+1)),l,n-k-1,\ldots,0)$$

Finally, the last $n-k$ points are obtained from the last point above by shifting the last zero to the left by one unit until it reaches the $k+1$ coordinate. So, the last set of points is:

$$(\overbrace{l,l,\ldots,l}^{n-m},l-1,l-2,\ldots,l-(k-(n-m+1)),l,n-k-1,\ldots,0,1)$$
$$(\overbrace{l,l,\ldots,l}^{n-m},l-1,l-2,\ldots,l-(k-(n-m+1)),l,n-k-1,\ldots,0,2,1)$$
$$\vdots$$
$$(\overbrace{l,l,\ldots,l}^{n-m},l-1,l-2,\ldots,l-(k-(n-m+1)),l,0,n-k-1,\ldots,1)$$

So, overall we have n points which satisfy the inequality with equality. Now it remains to prove that these points are affinely independent. Considering the matrix having these points as its rows and then applying the row reduction to the augmented matrix we obtain the identity matrix, which implies linear independence, therefore affine independence.

□

Below, we provide another set of inequalities for which we prove to be valid for all integer solutions of the predicate at-least-m-different. For every $i \in \{1, \ldots, m-2\}$, we obtain a family of $S_0 s$ such that $S_0 \in \binom{[n]}{n-m+i+2}$. Then for every S_0 we can find a family of sets obtained by any combination of two elements in S_0, S_1. Thus, $S_1 \in \binom{S_0}{2}$. Finally, for each S_1, we consider its compliment in S_0, $S_2 = S_0 \setminus S_1$.

$$\frac{(l-i-1)(l-i)}{2} - \frac{l(l+1)}{2} - (n-m-1)l \leq \sum_{j \in S_1} i x_j - \sum_{j \in S_2} x_j \tag{2}$$

$$\sum_{j \in S_1} i x_j - \sum_{j \in S_2} x_j \leq 2li - \frac{i(i+1)}{2} \tag{3}$$

Theorem 3. *The inequalities* (2),(3) *are valid for all integer solutions of the predicate* at-least-m-different$\{x_1, x_2, \ldots, x_n\}$.

Proof. We will prove (3). The proof of (2) is essentially the same. Due to symmetry and for simplicity, we only consider the inequality

$$i x_1 + i x_2 - x_3 - \ldots - x_{n-m+i+2} \leq 2li - \frac{i(i+1)}{2}$$

By the definition of the predicate we need at least m of the n variables to be different and the rest can take any other value within the given domain $[0, \ldots, l]$. As can be seen, the only case when all of the variables appear in the inequality is for $i = m-2$. In all other cases, we have $m-i-2$ variables not appearing in the inequality, therefore we can assume that only $m-(m-i-2) = i+2$ variables in the inequality need to be different. The latter holds since the partial inequality is valid for all the points of the extended inequality containing all variables. Since $m < n$, then $i+2 < n-m+i+2$. In order to achieve the maximum of the linear function, by the last inequality it is implied that the two variables associated by positive coefficients must take the maximum value of the domain and $i+1$ out of $n-m+i$ variables associated by negative coefficients must take all different values starting from the lowest value in the domain. A summary of above can be seen at the following solution:

$$(l, l, 0, 1, 2, \ldots, i, \overbrace{0, 0, \ldots, 0}^{n-m-1})$$

Based on the above reasoning, any other point that satisfies the predicate will only decrease the value of the linear function. By substituting each of the above solutions to the function we obtain exactly the right hand side claimed at (3). □

4 Computational results

In this section we experimentally exhibit indications that the facet-defining inequalities provided contribute in reducing the computational cost of solving a combinatorial problem which includes at least one at-least-m-different predicate. We ran two type of experiments. In both cases, we compute the volume of the polytope constructed by the known class of facets and consider the ratio to the volume of the hypercube, l^n, within which each polytope stands.

In the first experiment, we consider the number of variables needed to be different, equal to the total number of variables, i.e. $m = n \in \{2, \ldots, 8\}$ and the domain is fixed to be $[0, \ldots, 10]$. This case, when $m = n$, corresponds to the well studied all-different predicate which is a special case of the predicate at-least-m-different. The results obtained by ratio of the volume of the polytope of integer solutions to the volume of the hypercube, in this case 10^n, are a good indication of the usefulness of the class of facets, already known. As n increases, the number of integer points that do not satisfy the predicate and are within the cube, reduces significantly.

m	n	l	% cube
2	2	10	99
3	3	10	98
4	4	10	96
5	5	10	93
6	6	10	90
7	7	10	85
8	8	10	78

Table 1: Ratio of volume of the polytope to the hypercube, when $n = m \in \{2, \ldots, 8\}$, $l = 10$.

The second experiment is based on a more general case when m ranges from 4 to $n - 1$(or n), where n and l are fixed. The results of this case show that when $m < n$, the polytope obtained from the facets provided doesn't cut off the hypercube as many integer points as one would hope. Therefore, this urges the need of obtaining a complete characterization of the convex hull of integer solutions of the at-least-m-different predicate, in order for the reduction of the computational cost to be significant.

m	n	l	% cube
4	8	10	99
5	8	10	99
6	8	10	99
7	8	10	96
8	8	10	78

Table 2: Ratio of volume of the polytope to the hypercube, when $n = 8$, $l = 10$ and m increases.

5 Separation algorithm

Usually, it is not straightforward to obtain directly the optimal solution of an Integer Programming problem, therefore a Linear Programming relaxation is in most cases considered. However, considering a relaxation of the problem doesn't always give a feasible solution for the original problem. In this case, knowing the convex hull of integer solutions satisfying the predicate helps considerably in reducing the space of infeasible points obtained from an LP relaxation. The latter can be done by adding facet-defining inequalities of the convex hull. Certainly, in the case of a complete characterization, adding all the facets to the solver would produce an integer solution of the predicate, but this cannot always be efficient as the number of facets might be exponentially large. So, a polynomial time algorithm is needed to find and add to the solver, at each step, only facet-defining inequalities violated by the current Linear Programming solution. Such an algorithm is called Separation Algorithm.

For our set of inequalities, the general idea is based on Lemma 1. We claim that a point x violates an inequality if and only if the normalized point $\bar{x}$ where the coordinates of x are sorted, violates another inequality obtained from a permutation of the variables in the original inequality. Obviously, this is derived from some kind of algebraic, and therefore geometric, symmetry of the polytope. Due to Lemma 1, the violated inequality can be found in polynomial time, which reduces significantly the cost of obtaining the integer points satisfying at-least-m-different predicate.

The caller needs to know the inequality violated by the original point, not the normalized point. We will assume, as in the proof of the Lemma 1, that the sorting routine returns, not only the sorted point $\bar{x}$, but a permutation p such that $\bar{x} = x[p]$. The algorithm returns a set S which is either empty if no violated inequality is found or else determines the indices of the variables in the sorted point. Below, along with its proof is given the lemma where the separation algorithm is based on.

If enumeration is not the case for solving the at-least-m-different constraint, then Algorithm 1 is needed while using Linear Programming. Initially, it is solved a Linear Program having a subset of facet-defining inequalities and then each time a cut is violated, it is added to the solver and then resolved as an LP. This process repeats until an integer solution is obtained. The importance of this algorithm is not weakened if in addition to the at-least-m-different constraint is included a linear objective function.

We now proceed by stating the lemma and its proof.

Lemma 1. A point x violates some inequality

$$L \leq x_{i_1} + x_{i_2} + \ldots + x_{i_k} \tag{4}$$

or,

$$x_{i_1} + x_{i_2} + \ldots + x_{i_k} \leq U, \tag{5}$$

for some subset $\{i_1, i_2, \ldots, i_k\} \subseteq I := \{1, 2, \ldots, n\}$ where $k \in \{n - m + 2, \ldots, n\}$ and L, U, are respectively, the lower and upper bound of the violated facet, if and only if the point $\bar{x}$ with the same component as x, but sorted in increasing order, violates the inequality

$$L \leq \bar{x}_1 + \bar{x}_2 + \ldots + \bar{x}_k, \tag{6}$$

$$\bar{x}_{n-k+1} + \bar{x}_{n-k+2} + \ldots + \bar{x}_n \leq U, \tag{7}$$

respectively.

Proof. We start by first proving the necessary condition of the theorem for (4) and then continue to show the sufficiency. If a point x violates (4), for some subset $\{i_1, i_2, \ldots, i_k\} \subseteq I := \{1, 2, \ldots, n\}$ of size k, where $k \in \{n - m + 2, \ldots, n\}$, then

$$x_{i_1} + x_{i_2} + \ldots + x_{i_k} < L.$$

Now, since $\bar{x}$ is a point with the same components as x but ordered in increasing order, it implies the following,

$$\bar{x}_1 + \bar{x}_2 + \ldots + \bar{x}_k \leq x_{i_1} + x_{i_2} + \ldots + x_{i_k}$$

Therefore,

$$\bar{x}_1 + \bar{x}_2 + \ldots + \bar{x}_k < L,$$

which is obviously a violation of (6) by $\bar{x}$.

For the sufficient condition, we assume that the sorted point $\bar{x} = x[p]$ for some permutation p violates some inequality (6). Thus, we have that

$$\bar{x}_1 + \bar{x}_2 + \ldots + \bar{x}_k < L,$$

for some $k \in \{n - m + 2, \ldots, n\}$. By taking the set $S = \{p[1], p[2], \ldots, p[k]\}$, we see that,

$$\bar{x}_1 + \bar{x}_2 + \ldots + \bar{x}_k = \sum_{i \in S} x_i,$$

which implies the violation of (4) by x. The proof for (5) follows by similarity. □

From this we obtain separation Algorithm 1.

Algorithm 1 Separation of a point x from the polytope.

sepa(x,m,l)
$\bar{x}, p = \text{sort}(x)$ $\{\bar{x} = x[p]\}$
$n = \text{size}(x)$
for $k = n - m + 2, \ldots, n$ **do**
  $L = \frac{(m-n+k-1)(m-n+k)}{2}$
  $U = kl - \frac{(m-n+k-1)(m-n+k)}{2}$
  $V_L = \sum_{i=1}^{k} \bar{x}_i$
  $V_U = \sum_{i=1}^{k} \bar{x}_{n-i}$
  if $V_L < L$ **then**
    return $\{p[s] \mid s \in \{1, \ldots, k\}\}$
  end if
  if $V_U \geq U$ **then**
    return $\{p[s] \mid s \in \{1, \ldots, k\}\}$
  end if
end for
return $\emptyset$

Theorem 4. *The run time of the separation algorithm is bounded by a polynomial in n, the size of the input.*

Proof. The cost of sorting the elements is $n \log n$. In the worst case when $m = n$, and no facet is violated the cost of sorting the elements will be followed by a loop of order $n - 1$. Hence, the complexity of the separation algorithm will be $O(n^2)$. □

6 Conclusion

We have provided a new class of facet-defining inequalities for the predicate at-least-m-different. Associated to the polytope we demonstrated a separation algorithm, and suggested experimentally how it reduces the cost of pruning the solution space defined by the predicate. Future work will consist of (i) providing a complete characterization of the studied predicate, (ii) computational experiments on practical problems which include the predicate and (iii) convex hull representations of other useful constraint-satisfaction constraints.

References

[1] G. Appa, D. Magos, and I. Mourtos. On the system of two all_different predicates. *Inform. Process. Lett.*, 94(3):99–105, 2005.

[2] Michele Lombardi Domenico Salvagnin, editor. *Integration of AI and OR techniques in constraint programming for combinatorial optimization problems*, volume 10335 of *Lecture Notes in Computer Science*, Berlin, 2017. Springer.

[3] Komei Fukuda and Vera Rosta. Combinatorial face enumeration in convex polytopes. *Comput. Geom.*, 4(4):191–198, 1994.

[4] Thomas Hayman, Serge Kruk, and László Lipták. Facets of multiple alldifferent predicates of size 2 arranged in a cycle. *Congr. Numer.*, 215:77–90, 2013.

[5] Thomas Hayman, Serge Kruk, and László Lipták. Facets of the `alldifferent-except-zero` predicate. *Congr. Numer.*, 221:111–119, 2014.

[6] Serge Kruk and Niko Kaso. Polytope of two at-least predicates. *Congr. Numer.*, 2018.

[7] Serge Kruk and Niko Kaso. Study of the polytope of the *at_least* predicate. *International Journal of Machine Learning and Cybernetics*, Accepted 2018.

[8] Serge Kruk and Susan Toma. On the system of the multiple *all_different* predicates. In *Proceedings of the Fortieth Southeastern International Conference on Combinatorics, Graph Theory and Computing*, volume 197, pages 47–64, 2009.

[9] Serge Kruk and Susan Toma. On the facets of the multiple *alldifferent* constraint. *Congr. Numer.*, 204:5–32, 2010.

[10] D. Magos, I. Mourtos, and G. Appa. A polyhedral approach to the *alldifferent* system. *Math. Program.*, 132(1-2, Ser. A):209–260, 2012.

[11] Claude-Guy Quimper, editor. *Integration of AI and OR techniques in constraint programming for combinatorial optimization problems*, volume 9676 of *Lecture Notes in Computer Science*, Berlin, 2016. Springer.

[12] Willem-Jan van Hoeve. Introduction to the CPAIOR 2018 fast track issue. *Constraints*, 23(3):294–295, 2018. Held in Delft, June 26–29, 2018.

[13] H. P. Williams and Hong Yan. Representations of the all_different predicate of constraint satisfaction in integer programming. *INFORMS J. Comput.*, 13(2):96–103, 2001.