Logical Constraint IP Formulation Techniques

Illinois Institute of Technology
Department of Applied Mathematics

Adam Rumpf
arumpf@hawk.iit.edu

March 26, 2017

The following tables show methods for expressing a logical constraint in the form of an integer program (IP) constraint. Integer variables are often used to model logical decisions, in which case constraints on these decisions must be converted into (ideally linear) mathematical expressions. Most of the techniques listed here are adapted from Bertsimas and Tsitsiklis 1997 [1].

In all examples below, x, y, and z will always represent variables, all other letters will always represent constants, and bold typeface will be used to represent vectors.

Depdendency

Words x can only be true if y is true.

Symbols $\neg y \rightarrow \neg x$

Formulation

$$\begin{aligned} x &\leq y \\ x, y &\in \{0,1\} \end{aligned}$$

Explanation If $y = 0$, then the inequality forces $x = 0$. If $y = 1$, then no restrictions are placed on x.

Number of Choices

Words At most/at least/exactly n out of x_j can be true.

Formulation

$$\begin{aligned} & \sum_j x_n \leq n && \text{(at most } n\text{)} \\ \text{or} \quad & \sum_j x_n \geq n && \text{(at least } n\text{)} \\ \text{or} \quad & \sum_j x_n = n && \text{(exactly } n\text{)} \\ & x_j \in \{0,1\} && \forall j \end{aligned}$$

Disjunctive Constraints

Words At least one out of $\mathbf{a}^T\mathbf{x} \geq b$ and $\mathbf{c}^T\mathbf{x} \geq d$ must be true, where $\mathbf{a}, \mathbf{c}, \mathbf{x} \geq \mathbf{0}$.

Symbols $(\mathbf{a}^T\mathbf{x} \geq b) \vee (\mathbf{c}^T\mathbf{x} \geq d)$

Formulation

$$\begin{aligned} \mathbf{a}^T\mathbf{x} &\geq yb \\ \mathbf{c}^T\mathbf{x} &\geq (1-y)d \\ y &\in \{0,1\} \end{aligned}$$

Explanation We have introduced a binary decision variable y to choose whether to force the first or the second inequality to be true. If $y = 0$, then the constraints reduce to

$$\begin{aligned}\mathbf{a}^T\mathbf{x} &\geq 0\\ \mathbf{c}^T\mathbf{x} &\geq d\end{aligned}$$

Because $\mathbf{a}, \mathbf{x} \geq \mathbf{0}$ the first constraint is always trivially true, and is thus meaningless, so we are actually just enforcing $\mathbf{c}^T\mathbf{x} \geq d$. On the other hand, if $y = 1$, the constraints reduce to

$$\begin{aligned}\mathbf{a}^T\mathbf{x} &\geq b\\ \mathbf{c}^T\mathbf{x} &\geq 0\end{aligned}$$

By the same logic, the second constraint is meaningless, so we are actually just enforcing $\mathbf{a}^T\mathbf{x} \geq b$. In either case at least one must be true, and since y is free, it can be either of them.

Extensions This can be generalized to force at least k out of a total of m constraints of the form $\mathbf{a}_i^T\mathbf{x} \geq b_i$ for $i = 1, \ldots, m$ to be true by introducing variables $y_1, \ldots, y_m$ and forcing at least k of them to be 1, as in

$$\begin{aligned}\mathbf{a}_i^T\mathbf{x} &\geq y_i b_i & i &= 1, \ldots, m\\ \sum_{i=1}^{m} y_i &\geq k\\ y_i &\in \{0, 1\} & i &= 1, \ldots, m\end{aligned}$$

Set Membership

Words x must take a value in the set $\{a_1, \ldots, a_m\}$.

Symbols $x \in \{a_1, \ldots, a_m\}$

Formulation

$$\begin{aligned}x &= \sum_{i=1}^{m} y_i a_i\\ \sum_{i=1}^{m} y_i &= 1\\ y_i &\in \{0, 1\} & i &= 1, \ldots, m\end{aligned}$$

Explanation A binary indicator variable y_i is introduced for each set element a_i to indicate whether x should be equal to that set element. Setting the sum of all y_i to be 1 forces exactly one of them to be 1 and the rest 0, in which case the sum that defines x is simply $1a_i$ for the chosen index i.

Implication

Words If $\mathbf{a}^T\mathbf{x} \geq b$, then $\mathbf{c}^T\mathbf{x} \geq d$.

Symbols $(\mathbf{a}^T\mathbf{x} \geq b) \rightarrow (\mathbf{c}^T\mathbf{x} \geq d)$

Formulation

$$\begin{aligned}
\mathbf{a}^T\mathbf{x} - My &< b \\
\mathbf{c}^T\mathbf{x} + M(1-y) &\geq d \\
y &\in \{0, 1\} \\
M &\gg 0
\end{aligned}$$

Explanation This formulation makes use of the "big-M" technique, where we introduce a constant M large enough to dwarf any other quantity involved in this problem. As with the disjunctive constraint we have a binary indicator y. If $y = 0$, then the first constraint becomes $\mathbf{a}^T\mathbf{x} < b$ while the second becomes $\mathbf{c}^T\mathbf{x} + M \geq d$. If M is sufficiently large then the second constraint will be trivially true for any $\mathbf{x}$, so this reduces to simply $\mathbf{a}^T\mathbf{x} < b$, which is equivalent to $\neg(\mathbf{a}^T\mathbf{x} \geq b)$.

On the other hand, if $y = 1$ then the first constraint is $\mathbf{a}^T\mathbf{x} - M < b$ is trivially true for any $\mathbf{x}$, which leaves us with only the second constraint of $\mathbf{c}^T\mathbf{x} \geq d$. Combining these results for free y, we have $\neg(\mathbf{a}^T\mathbf{x} \geq b) \vee (\mathbf{c}^T\mathbf{x} \geq d)$, which is logically equivalent to $(\mathbf{a}^T\mathbf{x} \geq b) \rightarrow (\mathbf{c}^T\mathbf{x} \geq d)$.

Selectable Inequality Direction

Words Given $p \in \{0, 1\}$, $p = 1$ if and only if $x < y$.

Symbols $(p = 1) \leftrightarrow (x < y)$

Formulation

$$\begin{aligned}
x - y &< M(1-p) \\
y - x &\leq Mp \\
M &\gg 0
\end{aligned}$$

Explanation This is a straightforward application of the big-M technique and the binary indicator p to select whether to enforce $x < y$ or $x \geq y$.

Inequality

Words x must not equal y.

Symbols $x \neq y$

Formulation

$$\begin{aligned}
x - y &\geq \varepsilon - Mz \\
x - y &\leq -\varepsilon + M(1-z) \\
z &\in \{0, 1\} \\
0 < \varepsilon &\ll 1 \\
M &\gg 0
\end{aligned}$$

Explanation Here, M should be a very large constant and ε should be a very small positive constant. The constraint actually being enforced here is actually that $|x - y| > \varepsilon$, which for a sufficiently small value of ε may be close enough to $x \neq y$.

References

[1] D. Bertsimas and J.N. Tsitsiklis. *Introduction to Linear Optimization*, Athena Scientific, Belmont, MA, 1997.