1 Projection of a Transition Landmark Constraint

Consider the following set of constraints for a set of cuts $\mathcal{C}$:

$$\sum_{t \in C} Y_t \geq 1 \qquad \text{for all } C \in \mathcal{C} \tag{1}$$

$$\sum_{\substack{t \in T \\ label(t)=o}} Y_t \leq Y_o \qquad \text{for all } o \in \mathcal{O}. \tag{2}$$

$$Y_t \geq 0 \qquad \text{for all } t \in T \tag{3}$$

$$Y_o \geq 0 \qquad \text{for all } o \in \mathcal{O}. \tag{4}$$

We want to show that these constraints, projected to the operator-counting variables Y_o are equivalent to:

$$\sum_{o \in \mathcal{O}_S} Y_o \geq |S| \qquad \text{for all } S \subseteq \mathcal{C} \tag{5}$$

$$Y_o \geq 0 \qquad \text{for all } o \in \mathcal{O} \tag{6}$$

1.1 General Projection

In general, any constraint implied (1)–(4) can be written as a conic combination of those constraints. If we introduce multipliers $\alpha_C, \alpha_o, \alpha_t, \alpha_{\bar{o}} \geq 0$, the general form of this combination is

$$\sum_{C \in \mathcal{C}} \sum_{t \in C} \alpha_C Y_t + \sum_{o \in \mathcal{O}} (\alpha_o Y_o - \sum_{\substack{t \in T \\ label(t)=o}} \alpha_o Y_t) + \sum_{t \in T} \alpha_t Y_t + \sum_{o \in \mathcal{O}} \alpha_{\bar{o}} Y_o \geq \sum_{C \in \mathcal{C}} \alpha_C$$

Choosing $\alpha_{\bar{o}} > 0$ and all other multipliers as 0 yields constraints (6). As soon as any other multiplier is positive, choosing $\alpha_{\bar{o}} > 0$ only weakens the constraint compared to choosing $\alpha_{\bar{o}} = 0$. So, we can rewrite the constraint above as:

$$\sum_{C \in \mathcal{C}} \sum_{t \in C} \alpha_C Y_t + \sum_{o \in \mathcal{O}} (\alpha_o Y_o - \sum_{\substack{t \in T \\ label(t)=o}} \alpha_o Y_t) + \sum_{t \in T} \alpha_t Y_t \geq \sum_{C \in \mathcal{C}} \alpha_C$$

Grouping the sums by operator, yields

$$\sum_{o \in \mathcal{O}} \left(\sum_{C \in \mathcal{C}} \sum_{\substack{t \in C \\ label(t)=o}} \alpha_C Y_t + \alpha_o Y_o - \sum_{\substack{t \in T \\ label(t)=o}} \alpha_o Y_t + \sum_{\substack{t \in T \\ label(t)=o}} \alpha_t Y_t + \right) \geq \sum_{C \in \mathcal{C}} \alpha_C$$

This can be rewritten to

$$\sum_{o \in \mathcal{O}} \left(\alpha_o Y_o + \sum_{\substack{t \in T \\ label(t)=o}} (\sum_{\substack{C \in \mathcal{C} \\ t \in C}} \alpha_C + \alpha_t - \alpha_o) Y_t \right) \geq \sum_{C \in \mathcal{C}} \alpha_C$$

To project out Y_t, we have to choose the multipliers in a way that the coefficient of Y_t is 0, i.e., any choice of the multipliers that has coefficients of all Y_t at 0 corresponds to an implied constraint, and together all such constraints are equivalent to the projection. That is, the projection of (1)–(4)

to variables Y_o is equivalent to the following constraints:

$$\sum_{o \in \mathcal{O}} \alpha_o Y_o \geq \sum_{C \in \mathcal{C}} \alpha_C \tag{7}$$

$$\sum_{\substack{C \in \mathcal{C} \\ t \in C}} \alpha_C + \alpha_t = \alpha_{label(t)} \qquad \text{for all } t \in T \tag{8}$$

$$\alpha_C, \alpha_t, \alpha_o \geq 0 \tag{9}$$

This can be simplified further to

$$\sum_{o \in \mathcal{O}} \alpha_o Y_o \geq \sum_{C \in \mathcal{C}} \alpha_C \tag{10}$$

$$\max_{\substack{t \in T \\ label(t)=o}} \sum_{\substack{C \in \mathcal{C} \\ t \in C}} \alpha_C \leq \alpha_o \qquad \text{for all } o \in \mathcal{O} \tag{11}$$

$$\alpha_C, \alpha_o \geq 0 \tag{12}$$

For any choice of multipliers, constraint (10) is dominated by the constraint for the same choice of α_C but choosing α_o such that constraint (11) is tight. Thus, the constraints are equivalent to

$$\sum_{o \in \mathcal{O}} \Big(\max_{\substack{t \in T \\ label(t)=o}} \sum_{\substack{C \in \mathcal{C} \\ t \in C}} \alpha_C \Big) Y_o \geq \sum_{C \in \mathcal{C}} \alpha_C \tag{13}$$

$$\alpha_C \geq 0 \tag{14}$$

1.2 Special Case of Disjoint Cuts

For disjoint cuts, a transition t can be at most in one $C \in \mathcal{C}$, so the constraints further simplify to

$$\sum_{o \in \mathcal{O}} \Big(\max_{\substack{C \in \mathcal{C} \\ t \in C, label(t)=o}} \alpha_C \Big) Y_o \geq \sum_{C \in \mathcal{C}} \alpha_C \tag{15}$$

$$\alpha_C \geq 0 \tag{16}$$

Consider a set of multipliers α such that there are at least two different non-zero values, where α_{C_1} and α_{C_2} are the two largest values, i.e., $0 < \alpha_{C_1} = \max(\{\alpha_C \mid C \in \mathcal{C}\} \setminus \{\alpha_{C_2}\}) < \alpha_{C_2} = \max\{\alpha_C \mid C \in \mathcal{C}\}$. Now define multipliers

$$\alpha_C^1 = \begin{cases} \alpha_C - \alpha_{C_1} & \text{if } \alpha_C > \alpha_{C_1}, \\ 0 & \text{otherwise} \end{cases} \qquad \text{and} \qquad \alpha_C^2 = \begin{cases} \alpha_{C_1} & \text{if } \alpha_C > \alpha_{C_1}, \\ \alpha_C & \text{otherwise} \end{cases}$$

Note that $\alpha^1 + \alpha^2 = \alpha$ and that

$$\Big(\max_{\substack{C \in \mathcal{C} \\ t \in C, label(t)=o}} \alpha_C^1 \Big) + \Big(\max_{\substack{C \in \mathcal{C} \\ t \in C, label(t)=o}} \alpha_C^2 \Big) = \Big(\max_{\substack{C \in \mathcal{C} \\ t \in C, label(t)=o}} \alpha_C \Big).$$

This means that constraint (15) for α exactly matches the sum of constraint (15) for α^1 and α^2. Applying this argument inductively shows that we do not have to consider multipliers with more than one non-zero value. If all non-zero multipliers are the same, we can divide the whole constraint by that value to get an equivalent constraint that only uses binary values for α_C. The constraints are then equivalent to

$$\sum_{o \in \mathcal{O}} \Big(\max_{\substack{C \in \mathcal{C} \\ t \in C, label(t)=o}} \alpha_C \Big) Y_o \geq \sum_{C \in \mathcal{C}} \alpha_C$$

$$\alpha_C \in \{0, 1\}$$

The possible choices of multipliers then correspond to the subsets $S \subseteq \mathcal{C}$ with $\alpha_C = 1$ iff $C \in S$. The constraint is then equivalent to

$$\sum_{o \in \mathcal{O}} (\max_{\substack{C \in \mathcal{C} \\ t \in C, label(t)=o}} [C \in S]) Y_o \geq \sum_{C \in \mathcal{C}} [C \in S] \qquad \text{for all } S \subseteq \mathcal{C}$$

or

$$\sum_{o \in \mathcal{O}_S} Y_o \geq |S| \qquad \text{for all } S \subseteq \mathcal{C}.$$

1.3 Further Simplification of the Model

Say two subsets $S_1, S_2 \subseteq \mathcal{C}$ are connected if there is a t_1 in some $C_1 \in S_1$ and t_2 in some $C_2 \in S_2$ such that $label(t_1) = label(t_2)$, i.e. intuitively, there is an operator mentioned in both sets. We now show that we do not have to consider subsets $S \subseteq \mathcal{C}$ that can be split into two disconnected sets S_1 and S_2. In that case, $\mathcal{O}_{S_1}$ and $\mathcal{O}_{S_2}$ are disjoint, so the constraint for S matches the sum of the constraints for S_1 and S_2.

2 Total Unimodularity

Total unimodularity is an important property in linear programming.

Definition 1 (Totally Unimodular)**.** *A matrix is totally unimodular if the determinants of all its square submatrices are either* 0*,* -1*, or* 1*.*

The important property for us is that linear programs with integer bounds and totally unimodular coefficient matrices always have integer valued optimal solutions

Theorem 1 ([Conforti et al., 2014])**.** *Let A be an $n \times m$ totally unimodular matrix and $b \in \mathbb{Z}^m$ an integer-valued vector. Then the polyhedron $P = \{x \mid Ax \geq b, x \geq 0\}$ is integral, meaning that all basic solutions of P are integer.*

One way to show that a matrix is totally unimodular is the following result.

Proposition 1 ([Conforti et al., 2014])**.** *A matrix A with entries from $\{-1, 0, 1\}$ with at most two non-zero entries in each column is totally unimodular iff its rows can be colored red or blue, such that the sum of the red rows minus the sum of the blue rows is a vector with entries from $\{-1, 0, 1\}$.*

Consider again the case of disjoint cuts in constraints (1)–(4):

$$\sum_{t \in C} Y_t \geq 1 \qquad \text{for all } C \in \mathcal{C} \tag{1}$$

$$\sum_{\substack{t \in T \\ label(t)=o}} Y_t \leq Y_o \qquad \text{for all } o \in \mathcal{O}. \tag{2}$$

$$Y_t \geq 0 \qquad \text{for all } t \in T \tag{3}$$

$$Y_o \geq 0 \qquad \text{for all } o \in \mathcal{O}. \tag{4}$$

It is easy to see that all coefficients are from $\{-1, 0, 1\}$. If cuts are disjoint, variable Y_t only occurs in at most one constraint of type (1) with coefficient 1 and in the constraint (2) for $o = label(t)$ with coefficient -1. Variable Y_o only occurs with coefficient 1 in one constraint of type (2). Adding all rows (i.e., coloring all lines red) results in a vector with entries $\{-1, 0, 1\}$ which shows total unimodularity with Proposition 1.

If we treat Y_o as constants, the resulting matrix is still totally unimodular for the same reason. With Theorem 1 this shows that any vector of integer values Y_o, that can be extended to a solution of constraints (1)–(4), can also be extended with integer values for all Y_t. Since we know that every solution to our projection can be extended to a (real-valued) solution to (1)–(4), we can conclude that integer solutions to the projection can be extended to integer solutions of (1)–(4). That is, our projection perfectly captures the information contained in constraints (1)–(4), even when restricting all variables to integers.

References

[Conforti et al., 2014] Conforti, M., Cornuéjols, G., and Zambelli, G. (2014). *Integer Programming*. Springer.