Building a general MIP

Based on
“Solving Concisely Expressed Combinatorial Auction Problems” (B2)
by Craig Boutilier
Atomic

\[ s_\beta \leq x_{ij} \quad \text{Indicator for the good} \]

\[ v_\beta \leq p \cdot s_\beta \quad \text{Value of the subformula} \]

\textbf{p} is the price at the node
\textbf{i} is the good
\textbf{j} is the agent
\textbf{s} variables represent whether the bid is accepted
\textbf{x} variables indicated whether agent \textbf{j} gets good \textbf{i}
Or

\[ s_\beta \leq \sum_{i \leq d} s_{\beta_i} \]

We get the node if we get any of its children

\[ v_\beta \leq p \cdot s_\beta + \sum_{i \leq d} v_{\beta_i} \]

Value of the node is the sum of the children we got plus some additional bonus (optional feature)

**Key:** \( s_\beta \in \{0, 1\} \)
And

\[ d \cdot s_\beta \leq \sum_{i \leq d} s_{\beta_i} \]  
We get the node if we get all of its children

\[ \nu_{\beta} \leq p \cdot s_\beta + \sum_{i \leq d} \nu_{\beta_i} \]  
Value is calculated the same way as for OR

\[ \nu_{\beta_i} \leq M \cdot s_{\beta_i}, \forall i \leq d \]  
No value for this node if we didn’t get all of its children

Key: \( s_\beta \in \{0, 1\} \)
Xor

\[ s_\beta \leq \sum_{i \leq d} s_{\beta_i} \]
\[ v_\beta \leq p \cdot s_\beta + \sum_{i \leq d} v_{\beta_i} \]
\[ v_{\beta_i} \leq M \cdot t_{\beta_i}, \forall i \leq d \]
\[ \sum_{i \leq d} t_{\beta_i} \leq 1 \]

Same as before
Same as before
Pick at most ONE of the child bids

\( M \) is some big number that will be larger than any possible value for a subformula. Let the MIP solve for the bet choice for \( t \)
Constraints

\[ \sum_{i \in I} \sum_{j} x_{i,j} \leq u_I, \forall I \]

Ensure we don’t allocate more instances than we have for each good (I)

To deal with multiple goods, just let the MIP sort out the best ways of exchanging between buyers and sellers