Integer Linear Programming for Container Optimization

December 2024

Model Formulation

Sets and Parameters:

- N: Number of boxes.
- Indices: $i \in \{1, 2, \cdots, N\}$ for boxes
- Container dimensions: W (width), L (length), H (height).
- Each box i has dimensions (w_i, l_i, h_i).
- Each box can have multiple orientations $o \in O_i$. In this problem setting, we are considering axis-aligned boxes so $|O_i| = 6$ for the problem to be finite, because if we allowed a box to rotate freely at arbitrary angles (pitch, yaw, and roll), the set of possible orientations would be infinite. This is not the case for our project, however; we take yaw into account.

For each orientation o of box i, define (w_i^o, l_i^o, h_i^o) as the dimensions of box i in orientation o.

Decision Variables:

1. **Placement and Orientation**:
   - $s_i \in \{0, 1\}$: 1 if box i is places, 0 otherwise
   - $\delta_i^o \in \{0, 1\}$: 1 if box i is placed with orientation o, 0 otherwise.

   We must also ensure a box is either not placed or placed exactly once in one orientation $\sum_{s \in O_i} \delta_i^o = s_i$

2. **Position Variables**:
   - $x_i, y_i, z_i \geq 0$: Continuous variables representing the lower-left-front corner coordinates of box i within the container.

   Given the chosen orientation, the effective dimensions of box i are: $W_i = \sum_{o \in O_i} w_i^o \delta_i^o, L_i = \sum_{o \in O_i} l_i^o \delta_i^o, H_i = \sum_{o \in O_i} h_i^o \delta_i^o$.

3. **Non-Overlap Binary Variables**: Foe each pair (i, j) with i ¡ j, define binary variables to enforce non-overlapping conditions:

   - $A_{ij}, A_{ji} \in \{0,1\}$: Encodes which box is to the left of the other.
   - $B_{ij}, B_{ji} \in \{0,1\}$: Encodes which box is behind the other.
   - $C_{ij}, C_{ji} \in \{0,1\}$: Encodes which box is below the other.

   These binary variables ensure that if both boxes are placed, at least one of these directional seperations are applied, preventing overlap.

Objective Function: An industrial application of such project would ideally want to place as many boxes for possible. Of course this would not always be the case. For some container, the ILP might find very good solutions for small boxes and bad solutions for bigger boxes or vice versa, but for simplicity, let's not consider this variance and assume that the objective is to place as many boxes as possible:

$$\max \sum_{i=1}^{N} s_i$$

Constraints

1. **Orientation Selection**: $\sum_{o \in O_i} \delta_i^o = s_i \forall i$.

2. **Container Fitting**: Any placed box must fit entirely within the container: $x_i + W_i \leq W, y_i + L_i \leq L, z_i + H_j \leq H$.

3. **Non-Overlapping Constraints**: For any two boxes i and j: To say box i is strictly to the left of box j: $x_i + W_i \leq x_j + M(1 - A_{ij})$ and vice versa for other directions and orientations. Here M is a large constant that relaxes the constraint, it enforces the non-overlapping inequality.

4. Finally, if both boxes are placed ($s_i = s_j = 1$) at least one directionaly seperation must hold: $A_{ij} + A_{ji} + B_{ij} + B_{ji} + C_{ij} + C_{ji} \geq s_i + s_j - 1$. If both are placed, this enforces that at least on of the six binary variables is 1, ensuring seperation along at least one dimension.

This model with inputs (dimensions of the container and each box and the set of allowed orientations for each box) picks a box from a known pick-up location, moves it to (x_i, y_i, z_i) in the chosen orientation, stacks it inside the container with no overlap.