Mathematical Formulation

Goal: Minimize number of courses to satisfy HUB requirements.

- n: total number of available courses
- m: number of HUB requirement categories
- $A \in \{0,1\}^{m\times n}$: requirement matrix where $A_{ji} = 1$ if course i satisfies requirement j
- $b \in \mathbb{Z}^m_{\geq 0}$: required number of credits per requirement
- $\mathcal{G}_1, \ldots, \mathcal{G}_k$: groups of course indices with duplicate course codes
- $x_i \in \{0,1\}$: decision variable (1 if course i selected, 0 if not)

Optimize:

$$\text{Minimize} \quad \sum_{i=1}^{n} x_i$$

$$\text{Subject to:}$$

$$\begin{aligned}
\text{(1) Requirement coverage:} \quad & \sum_{i=1}^{n} A_{ji} x_i \geq b_j \quad \forall j = 1, \ldots, m \\
\text{(2) No duplicate course codes:} \quad & \sum_{i \in \mathcal{G}_r} x_i \leq 1 \quad \forall r = 1, \ldots, k \\
\text{(3) Binary decision variables:} \quad & x_i \in \{0,1\} \quad \forall i = 1, \ldots, n
\end{aligned}$$

Solution using PuLP and CBC

Now we'll walk through the procedure PuLP and the CBC (Coin or Branch and Cut) solver uses to approach the ILP (integer linear program) problem formulated above

Step 1: LP Relaxation

The first step is to consider the linear relaxation of the ILP by relaxing the binary integer constraint:

$$x_i \in \{0,1\} \quad \rightarrow \quad x_i \in [0,1]$$

This converts the ILP into a LP (linear program), which can be solved easily using methods such as the simplex algorithm. If the relaxed solution is composed of only integers - then great, we are finished! Otherwise:

Step 2: Branch-and-Bound Framework

If the relaxed solution is fractional, CBC applies the *branch-and-bound* method. This involves the following:

1. Choose a variable x_p such that $x_p \notin \{0,1\}$.
2. Create two subproblems:

$$\text{(i)} \quad x_p = 0 \qquad \text{(ii)} \quad x_p = 1$$

3. Solve each subproblem as a new LP (again relaxing any remaining integer constraints).
4. Continue branching on fractional solutions.

Each node in this search tree corresponds to a subproblem with a set of additional constraints. CBC holds onto the best integer solution found so far. Any subproblem with lower bound exceeding the objective value is pruned from consideration.

Step 3: Termination and Solution Retrieval

The algorithm terminates when:

- All branches are either solved or pruned.
- The optimal integer solution is found.

The final solution is a binary vector $x^* \in \{0,1\}^n$ such that:

$$\sum_{i=1}^{n} x_i^* \text{ is minimized}, \quad \sum_{i=1}^{n} A_{ji} x_i^* \geq b_j, \quad \sum_{i \in \mathcal{G}_r} x_i^* \leq 1.$$