Branch-and-Bound Schema for EV Charging Station Placement

1 Problem Formulation

The EV charging station placement problem can be formulated as an Integer Linear Program:

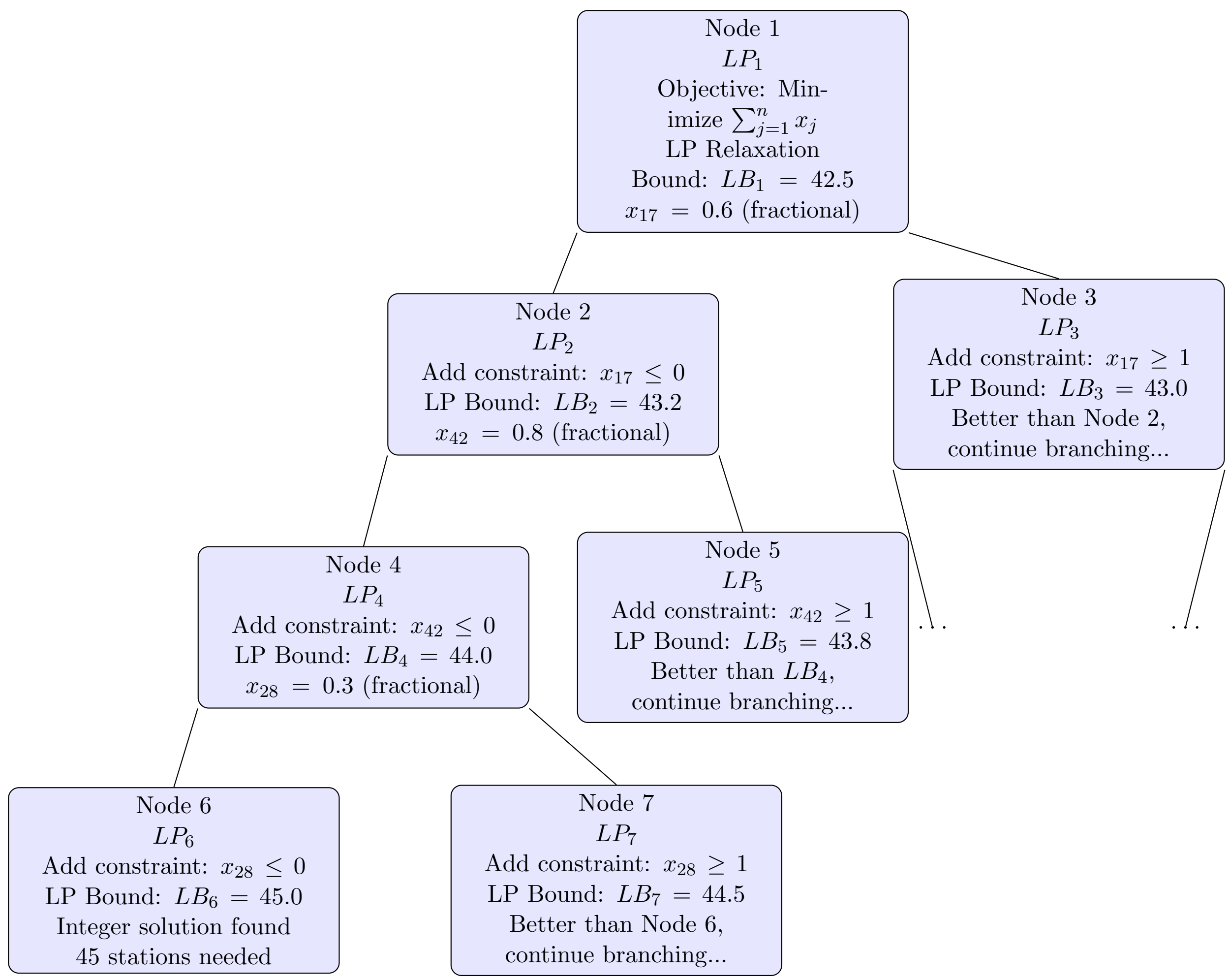
$$
\begin{aligned}
\text{Minimize} \quad & \sum_{j=1}^{n} x_j \\
\text{Subject to:} \quad & \sum_{j=1}^{n} y_{ij} = 1 \quad \forall i \in \{1, \ldots, n\} \\
& y_{ij} \leq x_j \quad \forall i, j \in \{1, \ldots, n\} \\
& y_{ij} = 0 \quad \forall i, j \text{ where } distance(i, j) > \text{BATTERY\_RANGE} \\
& \sum_{i=1}^{n} y_{ij} \leq \text{MAX\_VEHICLES\_PER\_STATION} \cdot x_j \quad \forall j \in \{1, \ldots, n\} \\
& x_j, y_{ij} \in \{0, 1\} \quad \forall i, j \in \{1, \ldots, n\}
\end{aligned}
$$

Where:

- x_j is a binary variable indicating if a charging station is placed at location j
- y_{ij} is a binary variable indicating if vehicle i is assigned to station j
- n is the number of vehicles (and potential station locations)
- BATTERY_RANGE is the maximum distance a vehicle can travel to reach a charging station

- MAX_VEHICLES_PER_STATION is the maximum number of vehicles that can be served by a single station

2 Branch-and-Bound Tree



3 Explanation of the Branch-and-Bound Process

The branch-and-bound algorithm for our EV charging station placement problem works as follows:

1. We start by solving the LP relaxation of the original problem (Node 1), where binary variables x_j and y_{ij} are allowed to take fractional values between 0 and 1.

2. The solution provides a lower bound ($LB_1 = 42.5$) on the optimal number of charging stations needed. However, this solution is not feasible for our original problem because some variables have fractional values (e.g., $x_{17} = 0.6$).

3. We select the fractional variable x_{17} for branching and create two child nodes:

   - Node 2: Add the constraint $x_{17} \leq 0$ (do not place a station at location 17)
   - Node 3: Add the constraint $x_{17} \geq 1$ (place a station at location 17)

4. We solve the LP relaxation for each child node to get new lower bounds.

5. We continue the branching process by selecting the node with the smallest lower bound, which is more likely to contain the optimal solution.

6. At Node 4, we add the constraint $x_{42} \leq 0$ and get a new LP bound of 44.0.

7. The branching process continues until we find an integer solution (like at Node 6) or prove that no better solution exists.

The optimal solution is the feasible integer solution with the minimum objective value found during the search.

4 Key Observations

- The branch-and-bound method systematically explores the solution space by dividing it into smaller subproblems.
- Each node in the tree represents a subproblem with additional constraints.
- Lower bounds from LP relaxations help us prune parts of the tree that cannot contain optimal solutions.
- For our large-scale EV charging station problem with thousands of vehicles, the branch-and-bound tree can become extremely large, highlighting the computational challenge.
- The solver parameters in our code (like time limit, MIP gap tolerance) help manage this complexity by accepting near-optimal solutions.