INTRODUCTION

BOMILP formulation:
\[
\min_{x,y} \ [c_1^T x + d_1^T y, c_2^T x + d_2^T y] \quad \text{s.t.} \quad Ax + By \leq b, \ (x,y) \in \mathbb{R}^m \times \mathbb{Z}^n
\]
Solution set \( \Omega_P \) is a piecewise linear curve formed as the nondominated sub-set of solution sets \( S_1, \ldots, S_k \) to slice problems \( P(y^1), \ldots, P(y^k) \).

\[
P(y^i) := \min_{x,y} \ [c_1^T x + d_1^T y, c_2^T x + d_2^T y] \quad \text{s.t.} \quad Ax + By^i \leq b
\]
where \( y^i \) are fixed integer values.

Current Solution Techniques:
1. LP-based Branch-and-bound (BB) [1, 3]
2. MILP-based Triangle-splitting [2]

FATHOMING RULE FOR BB

Bound sets are subsets of \( \mathbb{R}^2 \). Upper and lower bound sets \( U \) and \( L \) can be formed as:
\[
L \rightarrow \text{the solution set of the LP relaxation associated with a node } s \text{ of the BB tree being considered during iteration } s \text{ of BB.}
\]
Left: an example of applying the mapping \( \vartheta(N) \) for any set \( N \) of data known to be nondominated at iteration \( s \) of BB.

Right: an example of the main fathoming rule. \( L_{\text{sup}} \) is separable from \( U \) so \( \vartheta(N) \) can be fathomed.

STRUCTURE DETAILS

Let \( \pi.l \rightarrow \text{left (right) child of node } \pi \).

**INSERT** \( \pi.\pi \rightarrow \) at node \( \pi \):
- If \( \pi = \emptyset \), put \( \pi \) in the tree, i.e., assign \( \pi = \pi^* \).
- Else: If \( \pi^\pi \) dominates \( \pi \), go to REMOVE NODE.
  - Else: **INSERT** \( \pi^\pi \) above left(\( \pi.l \)) at \( \pi^\pi \).
  - **INSERT** \( \pi^\pi \) below right(\( \pi.r \)) at \( \pi^\pi \).

**REMOVE NODE**:
- If \( \pi \) is a leaf node, delete it.
- Else: Remove \( \pi \) from the tree by replacing it with the left-most node in the subtree rooted at \( \pi.r \) or the right-most node in the subtree rooted at \( \pi.l \).

REBALANCE:
- If balance criterion is not met at \( \pi \), shift nodes of subtree rooted at \( \pi \) until the criterion is met.

CONTRIBUTION

We propose a new data structure (T):
- dynamically updated binary tree
- stores only the nondominated subset of input
- at iteration \( s \) of BB, the stored data can be used as \( N \) to build upper bound set \( U \)
- lower complexity than \( L \), more data than \( P \)
- provides more efficient fathoming

INTUITION

Suppose we insert the data shown below to \( T \) in the order (1)-(5).
- (1) \rightarrow root node
- (2) \rightarrow right child of (1)
- (3) \rightarrow right child of (2)
- Reduce (1), Reduce (4)
- (4) \rightarrow right child of (1)
- Remove (1), replace with (2)
- Reduce (2), Reduce (4)
- (5) \rightarrow right child of (4)

Final stored data, and resulting tree:

PERFORMANCE GUARANTEES

Proposition 1. (Correctness) \( T \) stores only nondominated data, due to the following:
- \( T \) removes stored data which is dominated by newly inserted data
- \( T \) does not store any portion of inserted data which is dominated by currently stored data.

Proposition 2. (Complexity) The worst case complexity of maintaining our tree is \( O(\log t) \), where \( t \) is the number of stored nodes. (Note: Relaxing strict balance requirements allows for a complexity closer to \( O(1) \).)

**EXPERIMENT 1: RANDOM DATA**

**EXPERIMENT 2: BRANCH-AND-BOUND**

Data inserted to \( T \) during BB:

- **Warm-start:** \( M = 100 \)
- **No warm-start**
- **The M solutions from (P) are used to “warm-start” BB**
- **For each instance we select a low (L), medium (M), and high (H) value for M.**

REFERENCES