Visualization of Branch and Bound Algorithms

Mentees: Ishan Taneja, Anne McLaughlin, Mentor: David Morrison

WHAT IS IT AND WHY DOES IT MATTER?

• Optimization is integral to our daily lives. For example:
  • What is the most efficient route to deliver a set of packages?
  • If a Google search has a billion related webpages, which 10 should go on the front page?
• Branch and bound algorithms are widely used in mathematics and computer science to solve similar problems.

MOTIVATION

• Currently, visualization programs for branch and bound algorithms are few in number and do not allow for interactivity
• Therefore, writing and debugging code for this algorithm can be confusing, especially when outputted graphs and trees can have hundreds of thousands of nodes

GOALS

• Create a helpful and user-friendly branch and bound visualization program
• Given inputted data, produce an interactive image of a graph
• Allow users to click, drag, and obtain specific information from nodes
• Allow users to watch the algorithm grow in real time, and step through it frame-by-frame

THEORY

Standard Integer Programming problems cannot be solved through solving linear programming methods.

For example, in an IP problem with 20 variables, exponential solutions exist, so enumerating through all possibilities is not practical.

Branch and bound uses “divide and conquer” to find the set of feasible integer solutions and uses bounds to avoid exploring certain parts of the feasible set of solutions.

During the algorithm, certain subproblems may be computed to obtain an upper bound, U, which represents the best feasible solution so far.

1. Partition F into two or more subproblems whose union is F. (This step, the branching, defines a tree whose nodes are subsets of F)
2. Select subproblem F
3. If the subproblem is infeasible, delete it. Otherwise compute b(F)
4. If b(F) ≥ U, delete subproblem. (Bounding step)
5. If b(F) < U, obtain an optimal solution to the problem or break the corresponding subproblem into further subproblems that are added to the list of active subproblems.

OUR PROJECT

• We are in the process of developing a GUI that displays nodes and their edges that are parsed from a JSON file.
• Using the GTK library for interactive functions, and Cairo for graphics
• Our current implementation can display a window with a small number of nodes (< 50).
• Still need to integrate JSON input, add interactive features to the image, and create the user interface

REFERENCES

[1] P-Graph Wiki, “Algorithm ABB”

Acknowledgments

Thanks to our mentor, David Morrison and the PURE program!