Parallel Branch-and-Cut Integer Program Solver

Weixi Zhuo (wz2603)
(ParBnC)
Columbia University

(Dated: 28 November 2022)

I. OVERVIEW

This project aims to implement a parallel Haskell program that solves general integer linear programs (ILP) using the branch-and-cut algorithm[1]. We shall implement both sequential and parallel versions and compare their run-time performances against a benchmark ILP solver, GNU Linear Programming Kit (GLPK)[2].

II. BACKGROUND

It is known that general integer linear programming problems (ILP) are NP-hard. To obtain heuristic-based integral solutions of an ILP, branch-and-bound search algorithm[3] was developed. Essentially, it solves a series of linear-relaxed subproblems on different variables while keeping track of the optimal values as the subproblems branch out their descendants by fixing more and more variables as integers until all subproblems become infeasible (fathomed) or optimal solution is found. The general branch-and-bound algorithm is shown in figure 1a along with a search tree example in figure 1b. This can already benefit from parallelism as different branches can be processed separately while having the same memory on optimal bounds and termination conditions[4]. To make branch-and-bound more efficient, branch-and-cut algorithm[1] introduces a better method to fathom subproblems by including Gomory's cut[5] constraints. This is also the standard way to solve mixed-integer programs (MIP) in most solvers, which is how GLPK solves ILP in particular.

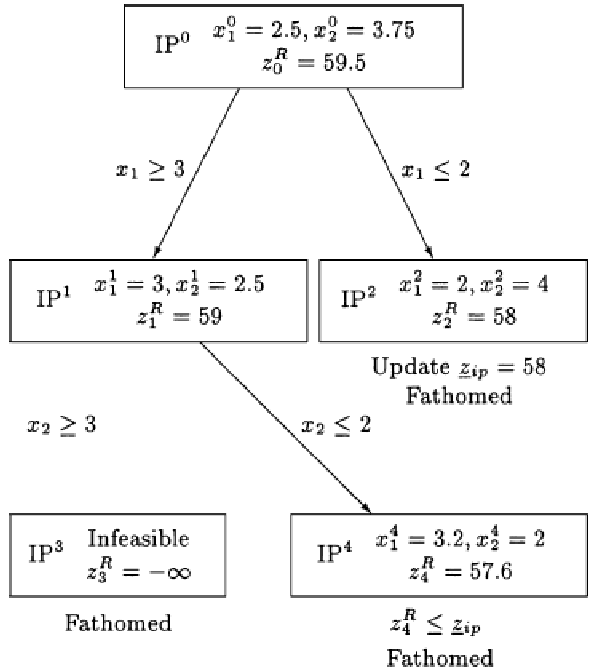
III. OBJECTIVES

We have the following list of objectives:

1. Implement sequential branch-and-bound algorithm by utilizing Haskell's Numeric module for linear subproblems.
2. Implement Gomory cutting plane for subproblem creation and keep it as a switch that can be included in the branch-and-bound program to create a branch-and-cut program.
3. Implement parallel branch-and-cut algorithm by applying parallelism at the first layer of subproblems while updating the same optimal bounds and incumbent solutions for early termination.
4. Compare performances among sequential branch-and-bound, sequential branch-and-cut, parallel branch-and-bound, parallel branch-and-cut, and GLPK solver called from Python CVXPY interface.

1 (*Initialization*): Set $L = \{\mathrm{IP}^0\}$, $\bar{z}_0 = +\infty$, and $\underline{z}_{ip} = \infty$.
2 (*Termination*): If $L = \emptyset$, then the solution x^* which yielded the incumbent objective value $\underline{z}_{ip}$ is optimal. If no such x^* exists (i.e., $\underline{z}_{ip} = -\infty$), then (IP) is infeasible.
3 (*Problem selection and relaxation*): Select and delete a problem IP^i from L. Solve a relaxation of IP^i. Let z_i^R denote the optimal objective value of the relaxation, and let x^{iR} be an optimal solution if one exists. (Thus, $z_i^R = c^\top x^{iR}$, or $z_i^R = -\infty$.)
4 (*Fathoming and Pruning*):
i) If $z_i^R \leq \underline{z}_{ip}$ go to Step 2.
ii) If $z_i^R > \underline{z}_{ip}$ and x^{iR} is integral feasible, update $\underline{z}_{ip} = z_i^R$. Delete from L all problems with $\bar{z}_i \leq \underline{z}_{ip}$. Go to Step 2.
5 (*Partitioning*): Let $\{S^{ij}\}_{j=1}^{j=k}$ be a partition of the constraint set S^i of the problem IP^i. Add problems $\{\mathrm{IP}^{ij}\}_{j=1}^{j=k}$ to L, where IP^{ij} is IP^i with feasible region restricted to S^{ij} and $\bar{z}_{ij} = z_i^R$ for $j = 1, \ldots, k$.
Go to Step 2.

(a) Branch and bound algorithm[3]



(b) Branch and bound example on two integer variables[3]

REFERENCES

[1] J. E. Mitchell, "Integer programming: branch and cut algorithmsinteger programming: Branch and cut algorithms," in *Encyclopedia of Optimization*, edited by C. A. Floudas and P. M. Pardalos (Springer US, Boston, MA, 2009) pp. 1643–1650.

[2] E. Oki, "Glpk (gnu linear programming kit)," (2012).

[3] E. K. Lee and J. E. Mitchell, "Integer programming: branch and bound methodsinteger programming: Branch and bound methods," in *Encyclopedia of Optimization*, edited by C. A. Floudas and P. M. Pardalos (Springer US, Boston, MA, 2009) pp. 1634–1643.

[4] L. Barreto and M. Bauer, "Parallel branch and bound algorithm - a comparison between serial, openmp and mpi implementations," Journal of Physics: Conference Series **256**, 012018 (2010).

[5] J. E. Mitchell, "Integer programming: cutting plane algorithmsinteger programming: Cutting plane algorithms," in *Encyclopedia of Optimization*, edited by C. A. Floudas and P. M. Pardalos (Springer US, Boston, MA, 2009) pp. 1650–1657.