Integer Programming Using Branch-and-Bound Algorithm.

Cheng Li, Zhangxiaowen Gong, Xuan Wang, Yiyi Wang

September 29, 2016

1. Definitions

An **integer programming** problem is a mathematical optimization or feasibility program in which some or all of the variables are restricted to be integers. In many settings the term refers to **integer linear programming** (ILP), in which the objective function and the constraints (other than the integer constraints) are linear. Integer programming is a NP-hard problem. It can be used in many application areas, such as production planning, scheduling, telecommunications networks, and cellular networks.[1]

Branch and bound is an algorithm design paradigm for discrete and combinatorial optimization problems, as well as general real valued problems. A branch-and-bound algorithm consists of a systematic enumeration of candidate solutions by means of state space search: the set of candidate solutions is thought of as forming a rooted tree with the full set at the root. The algorithm explores branches of this tree, which represent subsets of the solution set. Before enumerating the candidate solutions of a branch, the branch is checked against upper and lower estimated bounds on the optimal solution, and is discarded if it cannot produce a better solution than the best one found so far by the algorithm.[2]

2. Parallel programming for branch and bound algorithm.

Large-scale system with distributed memory are a natural fit to solving the computationally complex integer programming problem. As branch-and-bound algorithm uses state space search to explore the desired property, we can divide the tree into many subproblems (or subtrees) and parallelize branching computations [1]. Each processors can be responsible for different subproblems. In this

[1] wikipedia: Integer programming

[2] wikipedia: Branch and bound

way we can highly increase the performance of solving an integer programming problem.

3. Goal

We plan to leverage the branch-and-bound algorithm to solve integer programming problems in parallel in Charm++. Specifically we target at Travelling Salesman Problem(TSP) as a start point. TSP can be formulated as an integer linear program[2][3][4]. There are several things that we need to do. First, we would need to implement the Simplex Method for finding the linear programming optimal solution, which is a key part of solving the integer programming problem. Next, we need to define TSP to solve. In the future, we might extend the framework with an interface so that user can define any arbitrary integer programming problem. After that we would implement the branch-and-bound algorithm in Charm++. In the end, we would test our implementation under different scenarios. The LP part would be solved by off-the-shelf sovler like Gurobi[5], GLPK[6] or Cbc[7]. Some of the challenges include the intelligent pruning of the state space, load balancing, and low-overhead communication between chares. We hope that we could solve those problems at the time we implementing our algorithm.

4. References

[1] W. E. P. Bader David A.; Hart, "Parallel algorithm design for branch and bound," in *In greenberg, h. j. tutorials on emerging methodologies and applications in operations research*, Kluwer Academic Press, 2004.

[2] S. Papadimitriou C.H., *Combinatorial optimization: Algorithms and complexity.* Mineola, NY: Dover, pp.308-309.

[3] A. W. (. Tucker, *On directed graphs and integer programs.* IBM Mathematical research Project (Princeton University).

[4] G. B. (. Dantzig, *Linear programming and extensions.* Princeton, NJ: PrincetonUP, pp. 545–7, ISBN 0-691-08000-3, sixth printing, 1974.

[5] "Gurobi." [Online]. Available: http://www.gurobi.com/

[6] "GLPK. (gnu linear programming kit)." [Online]. Available: https://www.gnu.org/software/glpk/

[7] "Cbc." [Online]. Available: https://projects.coin-or.org/Cbc