Branch and Bound Tree Size Estimation

Wasu Glankwamdee
COR@L Seminar
September 28, 2006

References

- D. E. Knuth. Estimating the efficiency of backtrack program. *Mathematics of Computation*, 29:121-136, 1975.
- P. Kilby, J. Slaney, S.Thiebaux, and T. Walsh. Estimating search tree size. In *Proceeding of the 21st National Conference on AI*, Menlo Park, CA, 2006. AAAI.
- G. Cornuejols, M. Karamanov, and Y. Li. Early Estimates of the size of branch-and-bound trees. *INFORMS Journal on Computing*, 18:86-96, 2006.

Introduction

- Computational grids are incresingly being used as cost effective parallel computing platforms.
- The development of branch and bound tree size estimation is of fundamental importance.
- How long MIP solvers will take before solving a problem?
- We wish to be able to early predict whether or not it is worthwhile to spend vast computing power attempting to solve certain problems.

Knuth Method

- Estimate the size of a general backtrack tree.
- Explore random paths until reaching a leaf node by selecting nodes randomly according to uniform distrubution.
- Average over number of trails (1000/10000) to get the estimated tree size.
- Offline.

Knuth Equation

General Tree

$$\hat{N} = 1 + d_1 + d_1 d_2 + \ldots + \prod_{i=1}^{d_f}$$

Binary Tree

$$\hat{N} = 2^{d_f+1} - 1$$

- $\hat{N}$: Estimated number of evaluated node
- d_i : Number of children of selected node at depth i
- d_f : Depth of a leaf node

Limitation of Knuth

- Tend to overestimated especially tall and skinny tree.
- Cannot apply directly to branch and bound tree unless the optimal solution is provided as a priori or the node selection is depth-first.

Importance Sampling

The successor of a node is selected according to a weighted distribution instead of uniform.

Knuth Method with Importance Sampling

- Online.
- Kilby et al. apply *Weighted Backtrack Estimator* to estimate SAT and TSP tree. (depth-first)

Estimated Number of Evaluated Nodes

$$\hat{N} = \frac{\sum_{n \in P}(f(n) \cdot N)}{\sum_{n \in P} f(n)}$$

- P : Set of leaf nodes
- $f(n)$: Weighting function
- $N = 2^{d_f+1} - 1$

Weighting Functions

Completely Balanced Tree

$$f(n) = 2^{-d_f}$$

- Probability of selecting either one of the children is equally likely.
- $f(n)$: Probability that a random search tree path reaches a leaf node at depth d_f
- $f(n)$: Probability we would have seen the path HAD we done things ramdomly

General Tree

$$f(n) = f(L(n), U(n)) = \frac{1}{\alpha^L \beta^U}$$

- $L(n)$: Number of lower bound fixing of variables in random path
- $U(n)$: Number of upper bound fixing of variables in random path

Computational Experiment

- Test on 38 of MIPLIB3 instances with default MINTO.
- Optimal solution is not known in advance.
- Kill after 8 hours.
- Estimate after 5 seconds of solution time and the number of evaluated nodes is greater than 20 times the maximum depth.
- Error factor, i.e. “the factor by which the predictor under or over estimates the actual tree size.”, of 5 is acceptable.

Computational Result

Completely Balanced Tree

$$f(n) = 2^{-d_f}$$

- 16 instances are with error factor of 5.
- 3 instances have small error.
- 11 instances are unacceptable.
- 8 instances are not solved within 8 hours and the estimations are all greater than the number of node evaluated at time limit.

Computational Result

General Tree

$$f(n) = \frac{1}{(2.1053)^{L}(1.9048)^{U}}$$

- 16 instances are with error factor of 5.
- 1 instances has small error.
- 13 instances are unacceptable.
- 8 instances are not solved within 8 hours and the estimations are all greater than the number of node evaluated at time limit.

Cornuejols Method

- Online.
- Implement as part of MIP solver.
- Update the estimation as time elapses.
- Consume a negligible amount of time compared to branch-and-bound.

γ-Sequence

- i : Depth of node
- $w_T(i)$: Number of node at depth i (width)
- $\gamma_i = \frac{w_T(i+1)}{w_T(i)}$: Change of width
- Extrapolate the number of nodes at each depth from γ-sequence using regression model.

Key Parameters

Depth, Last Full Level, Waist Level

Definitions

- T : Branch and bound tree
- $d_T = \max\{i : w_T(i) > 0\}$: Depth of the tree
- $l_T = \min\{i : \frac{w_T(i+1)}{w_T(i)} < 2, 0 \leq i \leq d_T\}$: Last full level (complete binary tree)
- $b_T = \arg\max\{w_T(i) : 0 \leq i \leq d_T\}$: Waist of the tree
- $\{w_T(i) : 0 \leq i \leq d_T\}$: Profile of the tree
- $n_T = \sum_{i=0}^{d_T} w_T(i)$: Number of nodes in T

Algorithm

- Run branch-and-bound algorithm to collect the number of nodes at each depth, $w_T(i)$.
- Stop when $t > 5s$ and $n_T \geq 20d_T$.
- Find d_T, l_T, b_T ie. depth, last full level, waist level.
- Estimate the tree size.

Linear Model for Estimating γ-Sequence

- $\gamma_i = \frac{w_T(i+1)}{w_T(i)}$ for $0 \le i \le d_T$.
- $w_T(i) = \prod_{j=0}^{i-1} \gamma_j$: Width of depth i
- $n_T = 1 + \sum_{i=1}^{d_T} \prod_{j=0}^{i-1} \gamma_j$: Number of nodes in T
- Branch and bound tree has 3 stages: expand, waist, decay.
- γ-sequence: 2 (last full level), 1 (waist), 0 (deepest level).
- Assume that γ decreases linearly from 2 to 1 to 0.

The Model

$$\gamma_i = \begin{cases} 2, & 0 \le i \le l_T - 1, \\ 2 - \frac{i - l_T + 1}{b_T - l_T + 1}, & l_T \le i \le b_T - 1, \\ 1 - \frac{i - b_T + 1}{d_T - b_T + 1}, & b_T \le i \le d_T. \end{cases}$$

Computational Experiment

- Test on 28 of MIPLIB3 instances with default CPLEX 8.0.
- Kill after 10 hours.
- 15 instances are with error factor of 5.
- 8 instances are unacceptable.
- 5 instances are not solved within 10 hours. The estimations for 4 of them are greater than the number of node evaluated at time limit. The other is within the error factor.

Conclusion and Future Direction

- The estimation still need improvement/refinement to be able to apply to general MIP instances.
- Improve weighting function.
- How to use information other than the depth of leaf node to estimate e.g bounds.
- What to do when the best UB is updated.
- Incorporate the idea of expand, waist and decay.
- Variance/Confidence Interval.
- Test “harder instances” on parallel machines.