A High-performance Approach for Solution Space Traversal in Combinatorial Optimization

Wendy K. Tam Cho [1, 2, 3] and Yan Y. Liu [1, 4, 5]

[1] National Center for Supercomputing Applications
[2] Department of Political Science
[3] Department of Statistics
[4] CyberGIS Center
[5] Department of Geography and Geographic Information Science

{wendycho, yanliu}@illinois.edu

Introduction

Large-scale combinatorial optimization problems are increasingly common with the increase in data availability and the advances in effective optimization algorithms. Optimization techniques that may have been insightful for smaller scale data sets may no longer be feasible for larger data sets. These problems, while substantively distinct, are notably related in that their solution spaces are heroically large, resulting in an extreme-scale optimization problem. More importantly, their solution spaces are so vast, they are not rugged in the traditional sense. The usual peaks and valleys present themselves as a series of vast plateaus rather than a rapid succession of precipices.

We demonstrate how a high performance computing environment allows us to extract insights into these important problems that would otherwise not be possible. The goal is not to find one optimum but to characterize the underlying distribution of all solutions beyond a threshold of goodness.

Application – Causal Inference

Let *C* be a set of control group units, and *T* be a set of treatment units. The goal is to determine a subset of controls, S_C, and a subset of treatments, S_T , so that the ratio of the sizes of the subsets of controls and treatments is fixed, $|S_C|/|S_T| = \beta \in Z^+$, where Z^+ is the set of positive integers. Given any subset of *C* and *T*, define $h : 2^C \times 2^T \rightarrow [0,1]$ to be a measure of balance between any subset $S_C \subseteq C$ and any subset $S_T \subseteq T$. The goal is to determine subsets S_C and S_T and $\beta \in Z^+$, such that *h* is minimized (Cho *et al.* 2013).

The Subset Balancing problem is *NP*-Complete (through a polynomial time transformation function from the Partition problem.

- Solution space is large. For example, choosing subset of 10 from 100 units has 1.73×10^{13} possible solutions
- Need to identify all independent solutions over goodness threshold
- Optimization must employ diversification operator (to ensure adequate traversal of enormous solution space) as well as intensification operator (to climb difficult peaks and identify all solutions)

Optimization Algorithm

Genetic Algorithm (GA) is a generic heuristic method for finding near-optimal or optimal solutions to difficult search and optimization problems.

Principles

•Evolutionary process
- "Survival of the fittest"
- Iterative algorithm

•Solution population: a diverse set of initial solutions

•GA operators
- Selection, crossover, mutation, replacement

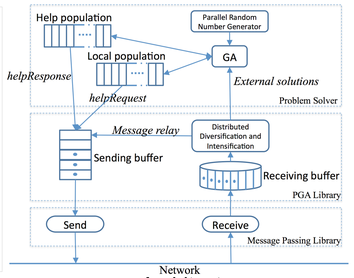
•Stopping criteria
- Solution quality
- Time or the number of iterations

Distributed Diversification and Intensification

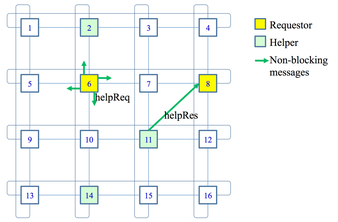
- Parallelization of GA using the island model
- Diversified search at the beginning
  - Guaranteed by the use of parallel random number generator (SPRNG)
  - Peaks of vast plateaus are identified by running massive islands on supercomputer
- Intensified search on promising peaks
  - Islands with promising peaks request helpers to help climb the identified peaks
  - Requests are sent using non-blocking calls to direct neighbors and propagated to all islands
  - An island turns into a helper following a probabilistic model
    - The number of helpers for a single requestor (*m)* << the number of islands (*n)*
    - Currently, a helping candidate flips a coin with probability *m/n*
    - Alternative: taking snapshots of existing helpers and decide *m*
    - The total number of helpers at any time < α * *n*, $\alpha \in [0,1]$
    - A helper sends back a better solution if found; resumes its local search

Parallelization

- MPI non-blocking calls
- Application-level send/ receive buffers
- Propagated broadcasts
- Parallel random number generator
- Messaging system extensible for other heuristic strategies (e.g., Tabu search)



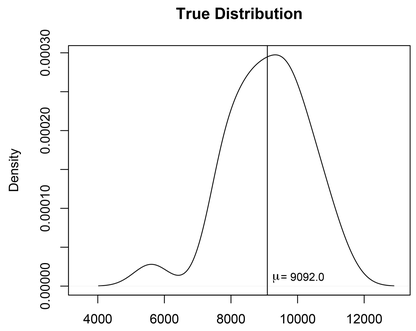
Architecture

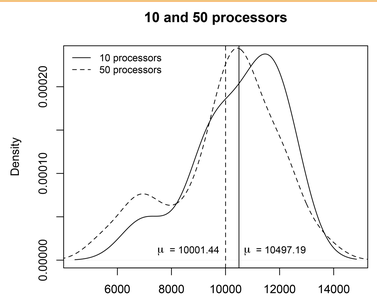


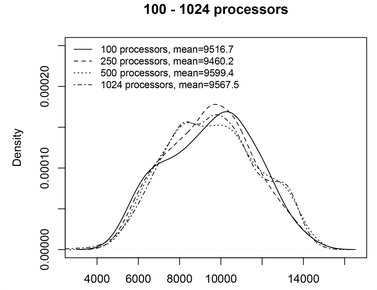
Messaging View

Simulation of Solution Space Traversal

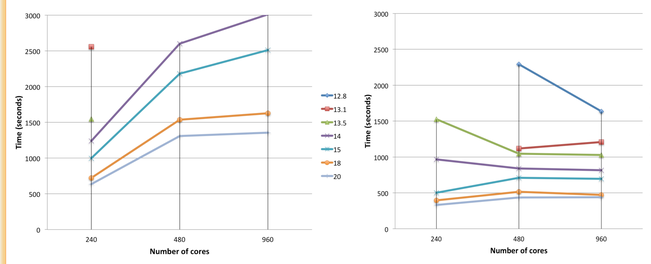
The simulation focuses on finding independent peaks across the solution space. Each of these peaks is an indistinguishably good and valid solution to our problem. The true underlying distribution has a mean of 9092.0. The recovered distribution mean approaches the true mean as the number of processors increases.







Scalability



(a) Synchronous (b) Asynchronous

Time taken to reach multiple solution quality thresholds. With the global synchronization, the message passing time ranged from 42% on 240 cores to 72% on 960 cores. The message passing cost declined considerably with asynchronous communication (0.007% on 240 cores and 0.01% on 960 cores).

Conclusion

- Massive computing power is necessary to enable the traversal of solution spaces that exhibit wide-ranging plateaus, which are common as optimization problem size becomes large
- Simulation on large causal inference problems confirmed our hypothesis: using a larger number of processors could recover true distribution of peaks more accurately, leaving fewer unexplored plateaus
- Non-blocking-based asynchronous message passing significantly improves the numerical efficiency of search algorithms by eliminating the increasing synchronization cost on larger number of processors

Acknowledgements

This research is part of the Blue Waters sustained-petascale computing project, which is supported by the National Science Foundation (award number ACI 1238993) and the state of Illinois. Blue Waters is a joint effort of the University of Illinois at Urbana-Champaign and its National Center for Supercomputing Applications.

References

- W. K. T. Cho, J. J. Sauppe, A. G. Nikolaev, S. H. Jacobson, and E. C. Sewell. An optimization for making causal inferences. *Statistica Neerlandica*, 67(2):211-226, May 2013.
- Y. Y. Liu and S. Wang. A scalable parallel genetic algorithm for the generalized assignment problem. *Parallel Computing*, 46:98-119, July 2015. http://dx.doi.org/10.1016/j.parco.2014.04.008.