Parallelization of Markov Chain Monte Carlo Methods

Zeyang Ye[1] and Yuefan Deng[2] (Advisor)

[1]IACS Jr. Researcher Award Winner; Department of Applied Mathematics & Statistics, Stony Brook University, NY 11794-3600, United States

[2]IACS Affiliate Faculty; Department of Applied Mathematics & Statistics, Stony Brook University, NY 11794-3600, United States

Abstract

We analyze a massive search space for finding optimal scalable parallelization strategies, including parallel temperature schedule, mixing strategy, and mixing frequency, for rapid convergence of the parallel Markov Chain Monte Carlo methods. We expect to find out the optimal time and ways for multiple Markov chains to communicate. Also, we adjust the sequential parameter, temperature, to fit for the parallel method. It is impossible to design a general theory applicable to arbitrary objective functions at this stage of our research. We examine the performance of our strategies by testing the optimization of the mobile route recommendation problem. We find that with the careful selection of the parallel strategies, nearly 100% speedup can be achieved. We believe there exists a scheme for these strategies leading to the optimal parallelization.

Introduction

Parallel MCMC Method

Markov Chain Monte Carlo (MCMC) method is a randomized heuristic approach to locate a global minimum or a near global minimum state given an objective function. Because of its, high computing time, it is effective in solving optimization problems of limited sizes. To solve large and realistic problems, parallel MCMC methods are designed by letting multiple Markov Chains run sequential MCMC method independently and communicating in certain amount of time. By adjusting parallel schemes including mixing frequency and mixing strategies, we hope to speed up MCMC method significantly.

Advantages

Computational Cost:
- Takes less computing time than genetic algorithm in some problems.
- Approximate the solution for NP-hard problem in polynomial time

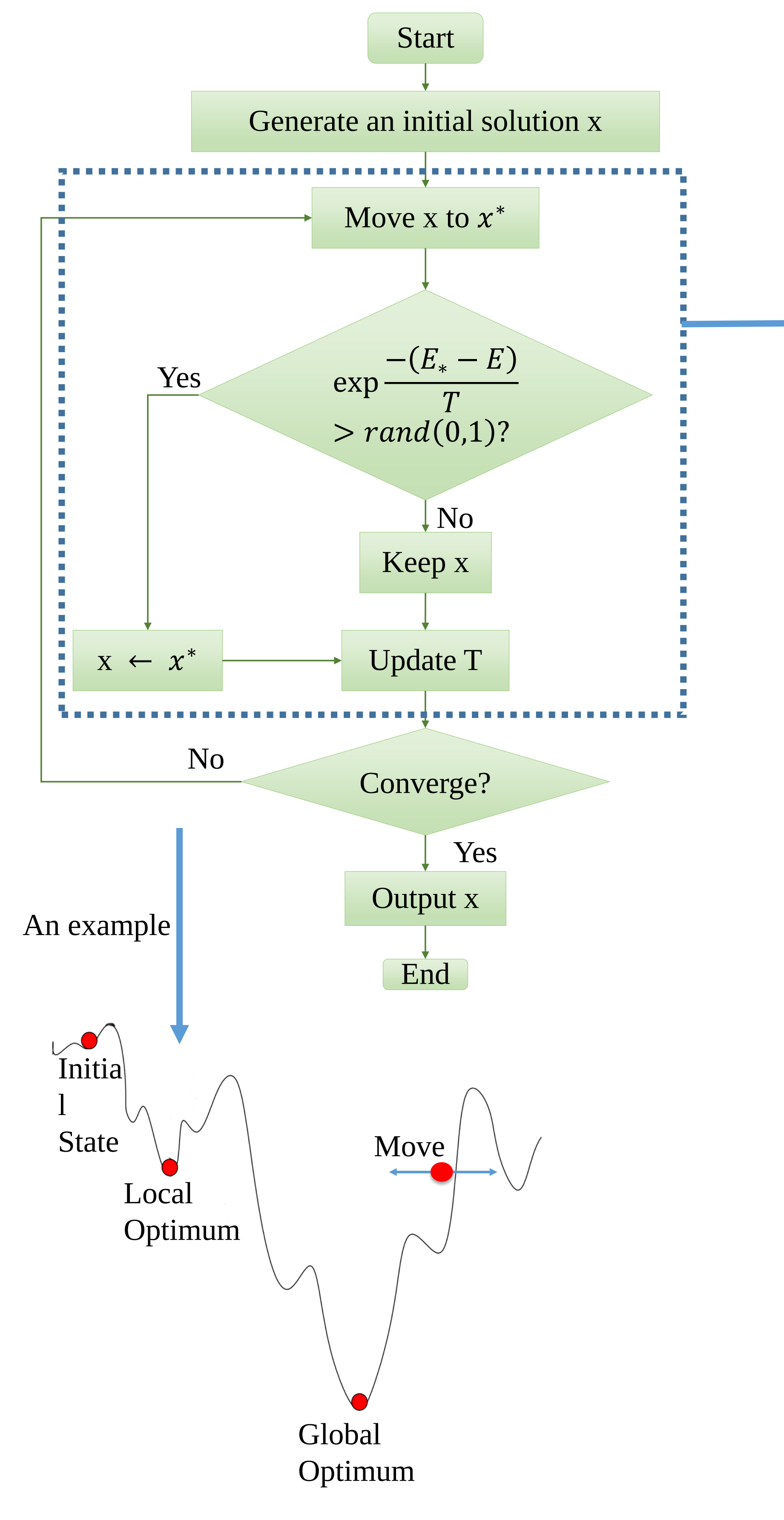
Solution Quality:
- Obtains a better solution than existing methods, e.g. descent method, random search, etc.
- Panacea for the problems with no tailored approximation algorithms

Applications

- Mathematical problems: Graph partitioning problems, Traveling salespersons problem, etc.
- Engineering problems: VLSI design, Football pool problem, etc.
- Biology problems: molecular structure, etc.
- Finance problems: portfolio optimization in high-frequency trading, etc.

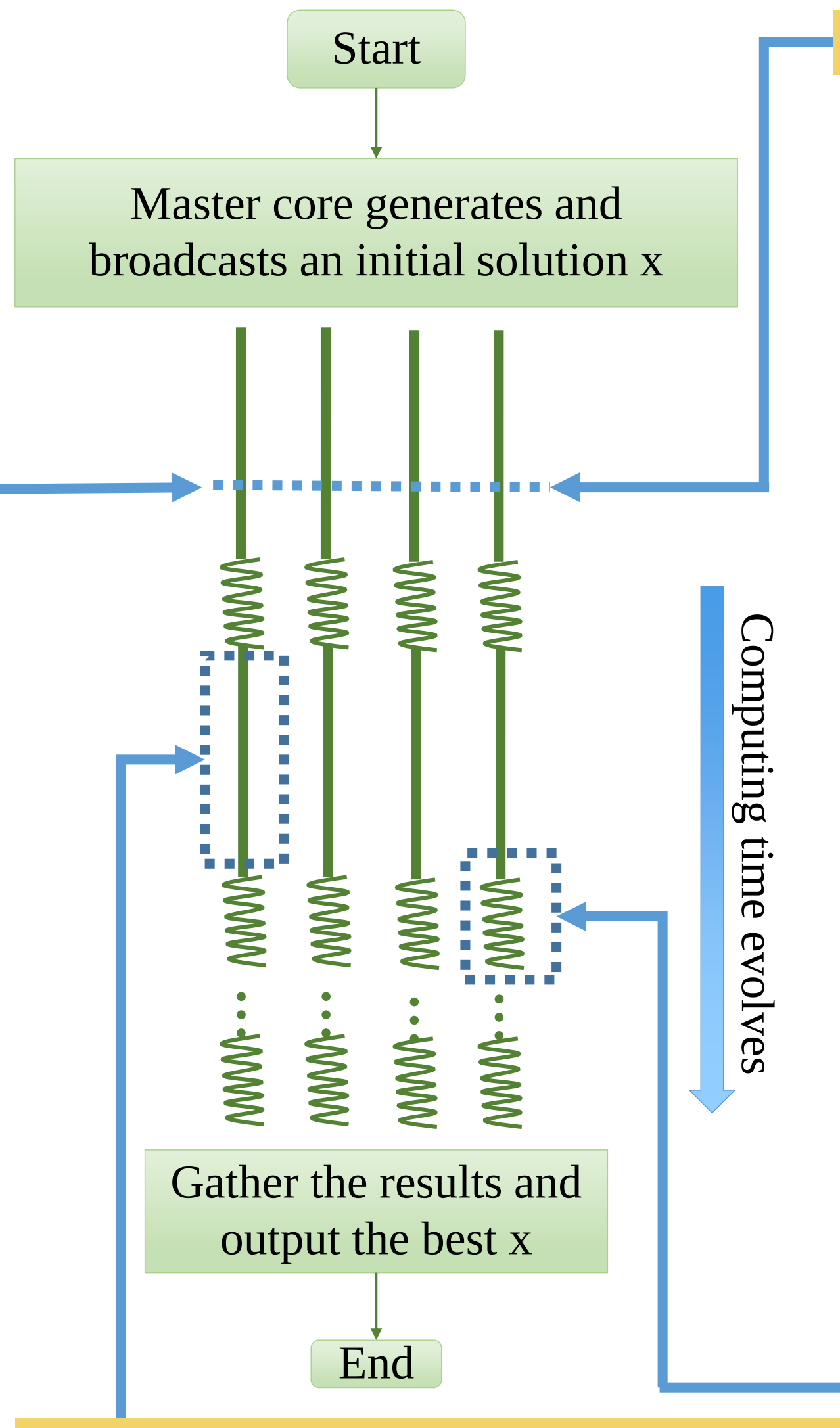
Methodology

Sequential MCMC Methods

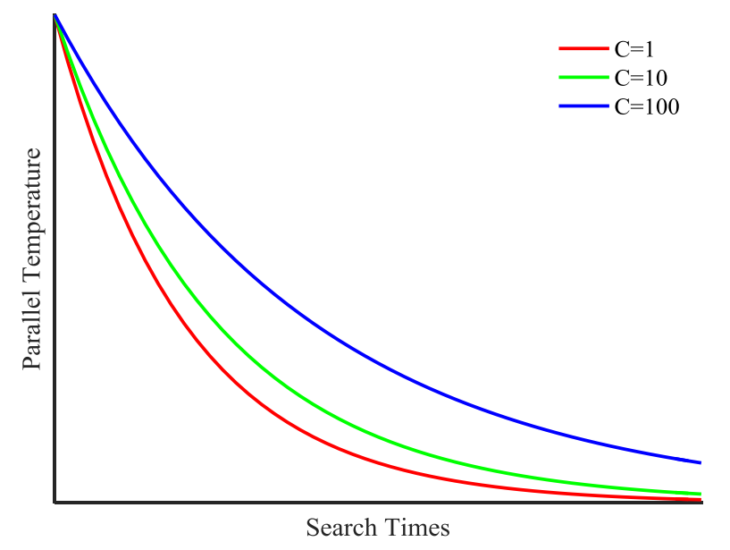


A point moves randomly to left and right within a small amount. When the next step is a lower state, it accepts the attempted move. Otherwise, it accepts the trail with a probability.

Parallel MCMC Methods


Start
Master core generates and broadcasts an initial solution x
Computing time evolves
Gather the results and output the best x
End

Parallel Temperature



Parallel temperature schedule

$$T_p = f(t, C, Mix, OF)$$

t: Computing time

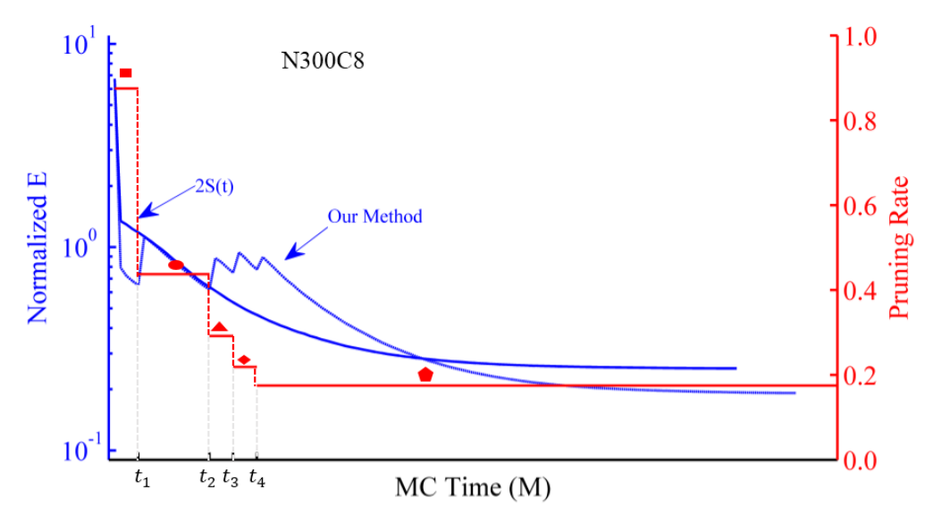
C: Number of cores

Mix: Mixing frequency and strategy

OF: Objective function

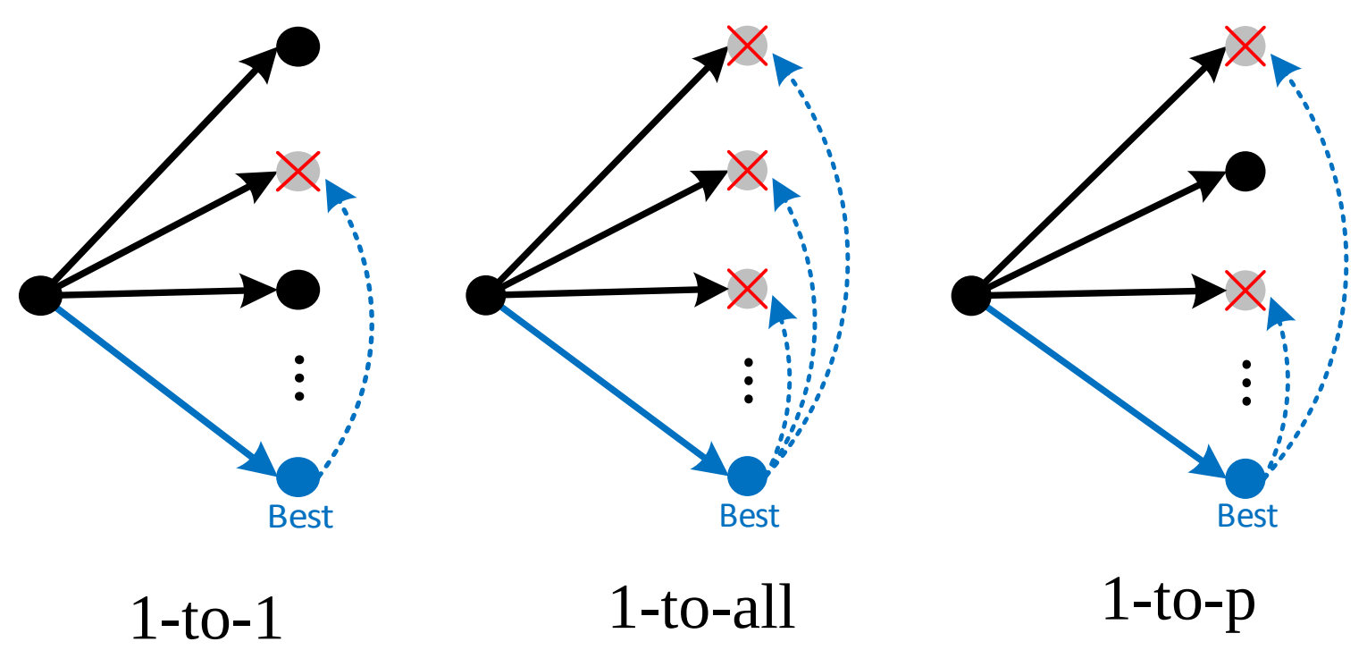
For multiple Markov Chains, C, in one step, the search space is explored for C times, which can be considered as one Markov Chain exploring the search space in C steps. As a result, intuitively, the parallel temperature, T_p has to be dropped C times faster than the sequential one. However, if the 1-to-all mixing strategy is used and the Markov Chains communicate every 1 steps, the whole system tends to have a lower probability to accept a worse solution. Therefore, T_p is a function of mixing strategy and frequency and needs to be dropped slower than we expected. With this respect, T_p controls the upper limited of the parallel speedup of this method.

Mixing Frequency



Mixing frequency determines the time when multiple Markov Chains mix. High frequency makes multiple Markov Chains converge to a local minimum quickly while low frequency leads to the global minimum or near global minimum solution slowly. Suitable selection of the mixing frequency to mimic the sequential algorithm needs to be done.

Mixing Strategy



Mixing strategy offers a way for multiple Markov Chains to mix. When each Markov Chain runs for certain amount of steps, its current energy, E, is different. 1-to-1 indicates that the Markov Chain with the lowest E, replaces the status of the one with the highest E. 1-to-all indicates that the best one replaces the status of all the others. 1-to-p indicates that the best one replaces the status of the others with a probability.

Results and Analysis

Objective Function

The mobile route recommendation (MRR) problem:

Given a set of potential pick-up points $C = \{c_0, c_1, \ldots, c_{|C|}\}$, and the associated probabilities of successfully picking-ups: $P = \{P(c_0), P(c_1), \ldots, P(c_{|C|})\}$. We want to locate the optimal driving route that minimizes the PTD before the next successful pick-up: $\min_{\overrightarrow{R^L} \in \vec{R}} E(Pos, \overrightarrow{R^L}, P_{\overrightarrow{R^L}})$, where $\overrightarrow{R^L}$ is the recommended route with the route length L; $\vec{R}$ is the set of all possible routes; $P_{\overrightarrow{R^L}}$ is the pick-up probabilities for all points in route $\overrightarrow{R^L}$. The distance vector and the probability vector are defined as: $D(\vec{d}) = \langle D_{c_0,c_1}, (D_{c_0,c_1} + D_{c_1,c_2}), \ldots, \sum_{i=1}^{L} D_{c_{i-1},c_i}, D_{max} \rangle$, $P(\vec{d}) = \langle P(c_1), \overline{P(c_1)} \cdot P(c_2), \ldots, \prod_{i=1}^{L-1} \overline{P(c_i)} \cdot P(c_L), \prod_{i=1}^{L} \overline{P(c_i)} \rangle$, where D_{c_{i-1},c_i} is the distance between c_{i-1} and c_i, D_{max} is the penalty for taxi drivers not picking up a passenger, and $\overline{P(c_i)} = 1 - P(C_i)$. The PTD of a route $\vec{r}$ can be computed based on the following function: $E(\vec{r}) = D(\vec{r}) \cdot P(\vec{r})$.

Platform

All of our experiments in the paper are implemented in Sunway BlueLight supercomputer consisting of 700 nodes. Each node contains two Intel Xeon X5675 six core processors and 36 GB DDR3 RDIMM RAM.

Overall Performance of the Parallel MCMC Method

The benchmark method is the best published existing method in the MRR problem. To solve this problem using MCMC method, we first build up the sequential one by adjusting the temperature schedule. Then we parallelize this method by adjusting the parallel temperature schedule, mixing frequency, and mixing strategy. We find that the MCMC method is faster than the benchmark method in magnitude. Also, it can still find near optimal result in high dimensions where the benchmark method cannot reach. Furthermore, by parallelizing this method, the parallel MCMC method can solve this problem within 10 seconds, which makes it applicable in solving the real time problems.

MCMC Method

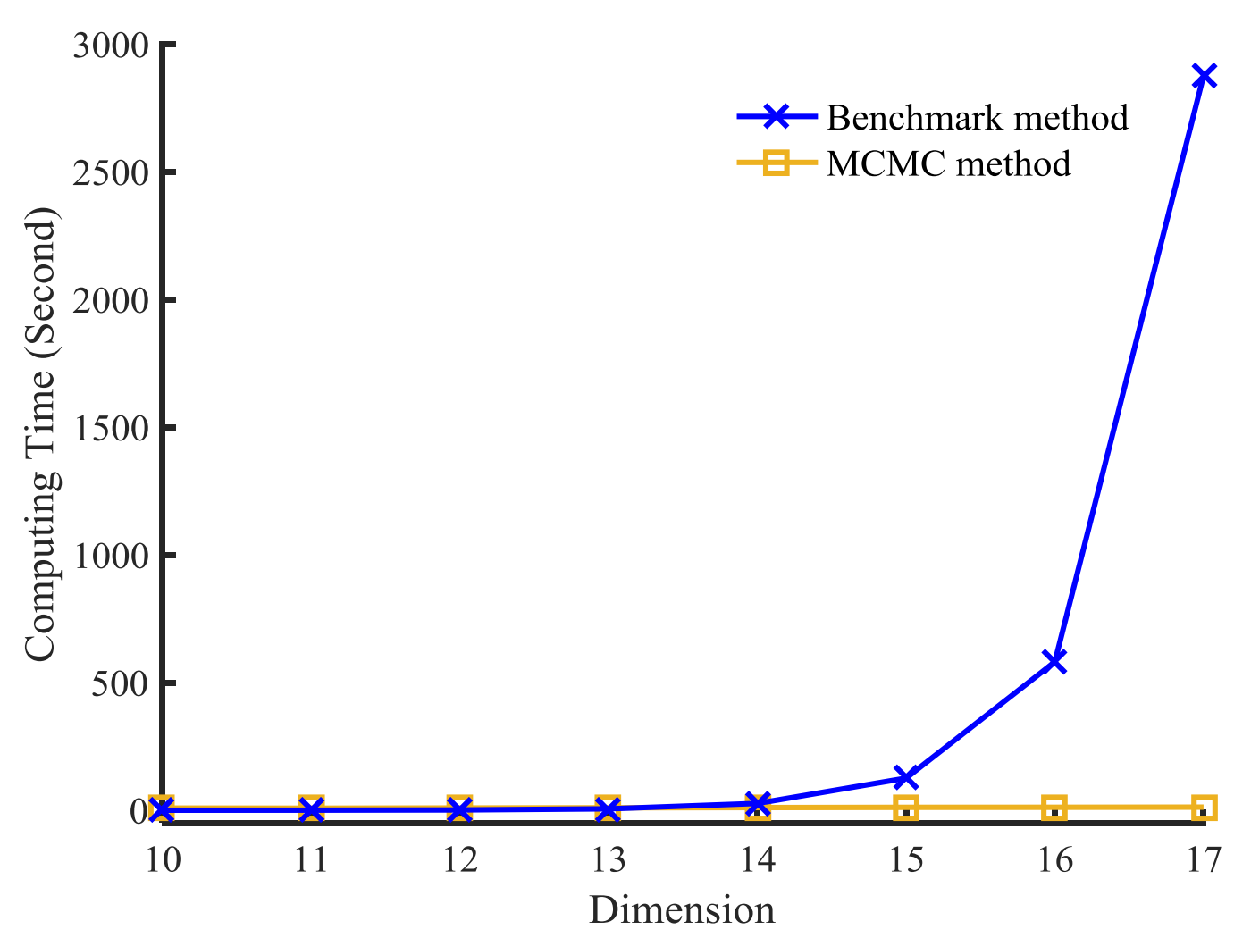


Figure 1(a) Comparisons of Computational Efficiency with Different Dimensions

Higher Dimension

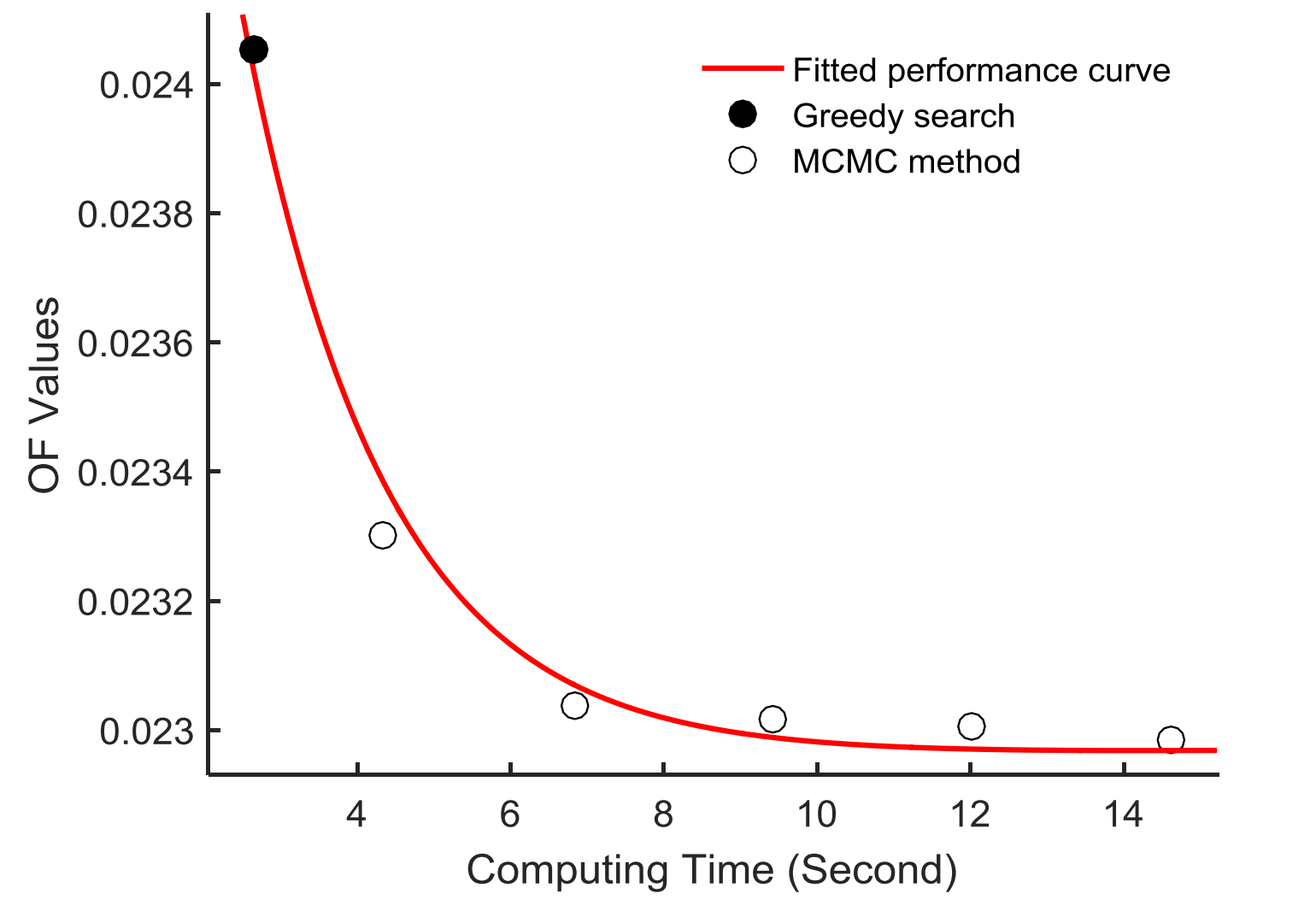


Figure 1(b) Performance curve of the proposed approaches with dimension 5000

Parallelization

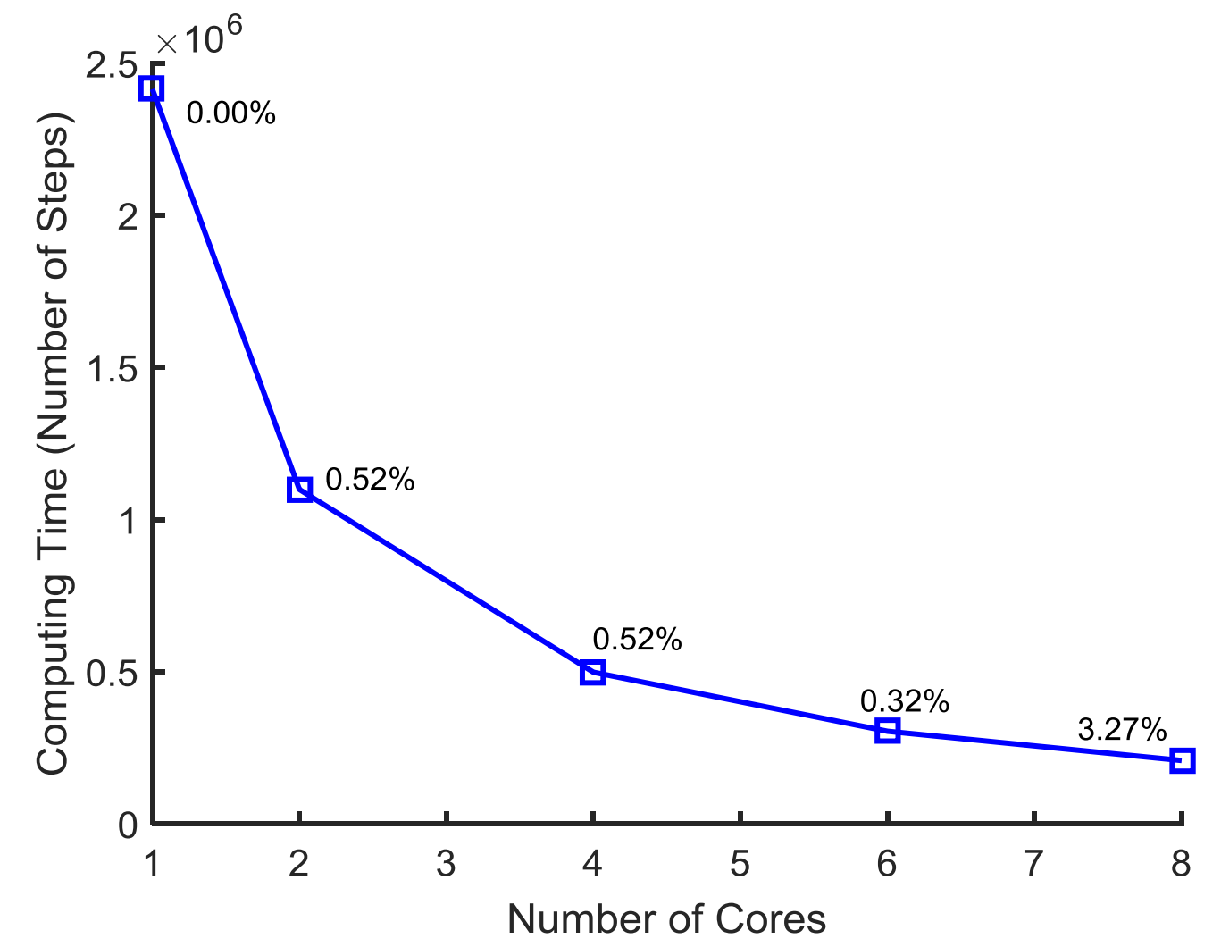


Figure 1(c) Decrease of the computing time with increase of the number of cores. The percentage in the figures are the relative different of E compared to the sequential one.

Achieved Speedup

Given a well built-up MCMC method, we can simply parallelize it by using a basic parallel scheme: 1-to-all mixing strategy, mixing frequency being 1, and parallel temperature the same as sequential temperature schedule. Then, one of the mixing strategy, 1-to-1, 1-to-all, or 1-to-p, is chosen and the relation between the mixing frequency and the computing time is determined. After that, the parallel temperature schedule T_p needs to be decided. MCMC method and its parallel version converge when the temperature is smaller than a problem related number. T_p is expected to decrease to that number quickly without sacrificing the solution quality.

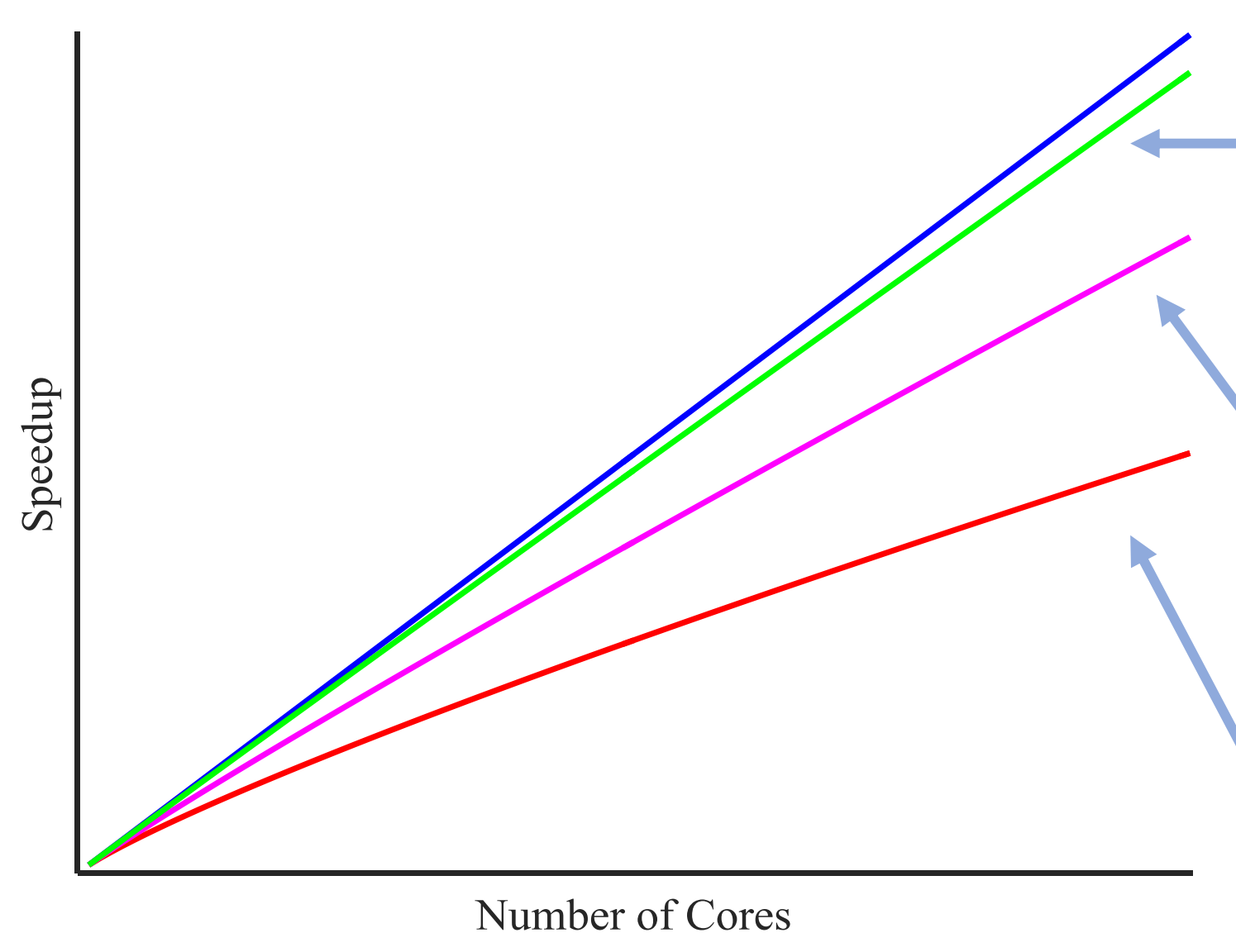


Figure 2 The speedup curve for parallel MCMC method in MRR problem. The blue line indicates 100% speedup.

Obtain the green curve by adjusting the parallel temperature:
1. The parallel temperature schedule T_p decreases faster than the sequential temperature schedule.
2. T_p decreases slower than the temperature schedule that is C times faster the sequential one, indicating the limit of the parallel speedup.

Obtain the pink curve by adjusting the mixing frequency and mixing strategy accordingly.
1. 1-to-all mixing strategy is used.
2. After the adjustment, mixing frequency is a decreasing function of the computing time.

Obtain the red curve by building up the model:
1. Design a suitable cooling schedule, move generation, and stopping condition for the MCMC method.
2. Parallelize the sequential version by using a basic mixing strategy and mixing frequency function.

Conclusions

1. Basic parallel scheme for parallel MCMC method obtains moderate speedup.
2. Having mixing strategies, mixing frequencies, and parallel temperature adjusted, parallel MCMC method achieves nearly 100% speedup.
3. In an optimal parallel scheme, mixing frequency decreases while the computing time increases.
4. 1-to-all mixing strategy can speed up parallel MCMC method as long as it has a suitable parallel temperature schedule.

References

[1] Darema, F., Kirkpatrick, S., and Norton, V. A. 1987. *Parallel techniques for chip placement by simulated annealing on shared memory systems*. Proceedings of 1987 IEEE International Conference on Computer Design.

[2] Suman, B. and Kumar, P. 2006. *A survey of simulated annealing as a tool for single and multiobjective optimization*. Journal of the operational research society. 57, 1143-1160.

A Presentation for the Annual Advisory Board Meeting

Zeyang Ye, zeyang.ye@stonybrook.edu