Studies on Sequential and Parallel Randomized Algorithms

C. Pandurangan	S. Rajasekaran	J.H. Reif	S. Sen
Indian Inst. of Tech.	Univ. of Penn.	Duke Univ.	Indian Inst. of Tech.
Madras, India	Philly, USA	Durham, USA	Delhi, India

1 Project Summary

Several important computational problems have been solved elegantly thus far using the powerful technique of randomization. The best known randomized algorithms for a variety of problems have better resource bounds than their deterministic counterparts and are often simpler to implement. This project addresses the issue of exploiting this technique to design efficient parallel algorithms for diverse problems such as: 1) Sorting and Selection 2) Computational Geometry and 3) Graph problems. The PIs have extensive experience in the design of randomized and parallel algorithms and have made several fundamental contributions to the area.

The significance of sorting in computer applications can not be overemphasized. A large number of frequently needed application programs like compilers, operating systems, etc., exploit sorting extensively. Moreover, sorting is also of immense theoretical interest, as revealed by the volume of literature on sorting. Rajasekaran and Reif presented the first logarithmic time optimal parallel algorithm for integer sorting and no such deterministic algorithm has been discovered so far.

We also plan to look at some fundamental graph problems and design optimal or near optimal parallel algorithms. Problems of our interest include depth first search, matching, etc.

Reif and Sen introduced randomization techniques to parallel algorithms in computational geometry and derived optimal speed-up for many fundamental problems like 3D convex hulls, planar point-location etc. We plan to tackle new problems and develop additional techniques.

2 Randomized Algorithms

Classical approaches to introducing randomness in algorithms typically assume a distribution on possible inputs and, compute the expected performance of various (deterministic) algorithms. Quick sort is a good example. If one assumes that each input permutation is equally likely to occur, Quick sort runs in an expected $O(n \log n)$ time to sort n numbers. The credibility of such an approach critically depends on the assumption made on the inputs. There may be applications where the input distribution is quite different from the one used in the probabilistic analysis.

As an attractive alternative, Rabin and Solovay & Strassen proposed introducing randomness in the algorithm itself. A randomized algorithm is one where in certain decisions are made based on the outcomes of coin flips. No matter what the input is, a large fraction of all possible outcomes for the coin flips will ensure 'good performance' of the algorithm. Thus the two approaches differ in the probability space used for analysis. In the former one considers the space of all possible inputs and in the later one employs the space of all possible outcomes for coin flips.

Since the introduction of randomized algorithms in 1976, a wide variety of computational problems have been solved (both sequentially and in parallel) using this technique. In this project we study randomized algorithms for diverse computational problems.

3 Overview of Research

We next provide a brief sketch of each one of our problems areas:

3.1 Sorting and Selection

Both due to their practical value and theoretical interest, comparison problems have fascinated many a researcher. The practical importance of these problems is illustrated by the estimate that 40% of any computer's CPU time is spent on executing a sorting subroutine.

Even though the logarithmic depth sorting circuit due to AKS, the algorithm is rendered impractical due to the large constants in the depth bounds and the irregular nature of the expander graphs. Interestingly, the first logarithmic depth sorting circuit was randomized and works on a more feasible architecture like the CCC. The algorithm was proposed by Reif and Valiant and is known as Flashsort. Also the best known algorithm for integer sorting is randomized (due to Rajasekaran and Reif). Selection and sorting problems have optimal practical parallel algorithms on certain models, but not on some other models. One of the goals of this research is to obtain optimal practical selection and sorting algorithms on the latter models.

The model of our primary interest will be the mesh including the reconfigurable mesh. The reconfigurable mesh is an interesting model from many perspectives. For instance, sorting can be performed in $O(1)$ time using n^3 processors. In contrast one needs $\Omega(\frac{\log n}{\log \log n})$ time on the CRCW PRAM, given only a polynomial number of processors. Mesh is also considered to be the most feasible parallel architecture.

3.2 Graph Algorithms

Graph Theory finds applications in every walk of life, more so in the field of computer science; yet only a handful of problems have been shown to admit polynomial time solutions. In the context of parallel graph algorithms, even some basic problems such as depth first search and matching have not been proven to be in $\mathcal{NC}$, the class that is known to admit efficient speed-ups. Randomization has played a crucial role in arriving at efficient algorithms for several graph problems. Symmetric complementation games and the random mating lemma of Reif, the depth first serach algorithm of Aggarwal and Anderson, and the maximal independent set algorithm of Luby are among the most significant results in this direction.

Our investigations will be directed towards i) Improving further the complexity results of the randomized algorithms (both sequential and parallel) for fundamental graph problems, ii) Identifying new techniques and paradigms for designing randomized algorithms for special classes of graphs such as planar graphs and perfect graphs, and iii) The study of efficient derandomization techniques for problems such as matching and flow problems.

3.3 Computational Geometry

Following the seminal work of Reif and Sen in introducing some versatile random-sampling techniques for parallel algorithms in computational geometry, there has been some interesting work related to obtaining optimal speed-up algorithms. These include constructing arrangements and sublogarithmic time algorithms for constructing convex-hulls. Some important problems are not known to have optimal logarithmic time parallel algorithms. Examples include Voronoi diagram of line segments, higher dimensional Voronoi diagram, etc. We propose to examine these problems.