DESIGN AND ANALYSIS OF SOME COMBINATORIAL AND COMPUTATIONAL GEOMETRY PROBLEMS FOR PARALLEL EXECUTION

By

SANJEEV SAXENA

A thesis submitted in partial fulfilment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Department of Computer Science & Engineering,

INDIAN INSTITUTE OF TECHNOLOGY, DELHI

New Delhi-110 016.

JANUARY 1989

CERTIFICATE

This is to certify that the thesis entitled "DESIGN AND ANALYSIS OF SOME COMBINATORIAL AND COMPUTATIONAL GEOMETRY PROBLEMS FOR PARALLEL EXECUTION" submitted by Mr.Sanjeev Saxena to the department of Computer Science & Engineering, Indian Institute of Technology, Delhi, for the award of the degree of Doctor of Philosophy is a record of the bonafide research work carried out under our supervision.

The thesis or any part thereof has not been submitted to any other university/institution for award of any degree or diploma.

Prof.P.C.P. Bhatt	Dr.V.C.Prasad
Computer Sc. & Engg.,	Electrical Engg.
I.I.T., Delhi,	I.I.T.,Delhi
New Delhi-110016.	New Delhi-110016.

ACKNOWLEDGEMENTS

This work could not have been accomplished without the assistance of my supervisors Prof.P.C.P.Bhatt and Dr.V.C.Prasad. They taught me almost everything that is required to do effective research.

I would like to thank Prof.S.N.Maheshwari, Dr.Shashi Kumar, Dr.R.K.Ghosh and Dr.S.Biswas for useful discussions, Prof.K.K.Biswas, Dr.Phalguni Gupta and Dr.Anshul Kumar for advice and assistance, and Dr Samar Singh, Dr.A.Mukhopadhyaya, Dr.Rajeev Sangal and N.C.Kalra for cooperation.

I am grateful to my friends Dr.J.Cheriyan,Dr.Pratul Dublish, Ravi Mittal, B.M.Subraya, S.K.Bose, Dr.V.Arvind, Dr.S.Kapoor, Manmohan Gupta, Dr.M.Zubair, N.Vikas and Rajiv Jain for companionship provided during the course of this work.

I wish to thank Mr.Jaswant Singh Rawat and Mr.Batra for assistance and cooperation.

I also wish to thank external examiners for their valuable comments and suggestions. This I believe has resulted in a better manuscript.

Finally, I wish to thank I.I.T. Delhi and government of India for providing financial support (fellowship), without which this work could never have been undertaken.

S. Saxena
18.12.89.
Sanjeev Saxena

ABSTRACT

This thesis is concerned with the study of parallel algorithms for some Combinatorial and Computational Geometry problems. An attempt has been made to solve problems in parallel as fast as possible with reasonable resource bounds.

Parallel computation typically entails an environment in which computations may be carried out at more than one processor with the stipulation that the processors might have to communicate to each other. Communication between processing units may either be through a shared memory or via an interconnection network. In this thesis Concurrent Read Concurrent Write (CRCW) Shared Memory Model and Orthogonal Tree Network (OTN) are used as models for parallel computers. The parallel computer models used are described in <u>Chapter 1</u>. The CRCW model is very useful for studying inherent parallelism present in a problem. It is independent of technological fan-in limitations. One method of mapping known algorithms (on CRCW models) on more practical models is by sorting on addresses. It is known that n items can be sorted in $O(lg^2 n)$ time with O(n) processors on Hypercubes. This would imply that algorithms which take t(n) time with O(n) processors on CRCW model can be simulated on Hypercube in $O(t(n)lg^2 n)$ time. Algorithms which take O(n) space and t(n) time with O(n) processors on CRCW model with dynamic priority rule for resolving

concurrent writes can also be simulated on n X n OTN in O(t(n)*lg n) time.

Algorithms on OTN are fast resulting in good AT^2 complexity values. Hypercube uses O(n) processors with each node having lg n - 1 neighbors. OTN on the other hand, uses $O(n^2)$ processors but each node has at most three neighbors and uses about the same area (as that used by Hypercube).

In this thesis, the following representative problems are studied on deterministic parallel computer models from time complexity point of view.

A) Addition

B) Sorting

C) Planar Routing Problems

D) List Ranking

E) Delaunay Triangulation

F) Euclidean Steiner Tree

For the first two problems O(lg n) time parallel solutions are well known. The issues involved for sorting and adding in o(lg n) time are studied in <u>Chapter 2.</u> It is shown that

i) Prefix sums of n numbers of b bits each, can be found in O(t(n,b)) time with n/t(n,b) processors; here t(n,b)=min[lg n/lg(lg n),lg b]. Thus the algorithm achieves optimal (linear) speed-up and matches the lower bound for adding b-bit numbers on CRCW model. In particular,

$O(n*lglg\ n/lg\ n)$ processors are sufficient to add $O(n^{c/lglg\ n})$ bit numbers in $O(lg\ n/lglg\ n)$ time.

ii) n numbers of b bits each can be added in parallel on CRCW model, in which word size is m, $lg\ n \le m \le n$, in time $O(t(n,m))$ with $(nb/m)/t(n,m)$ processors. Thus n, n-bit numbers can be added in $O(lg\ n/lglg\ n)$ time on $O(lg\ n)$ bit computers.

Same processor and time bounds also hold for parallel prefix computations.

iii) n numbers can be sorted in $O(lg\ n/lglg\ m)$ time using $O(n*m)$ processors (if $lg\ m/lglg\ m \ge lglg\ n$). Thus, with $n^{1+(1/lglg\ n)}$ processors it is possible to sort in $O(lg\ n/lglg\ n)$ time.

iv) If the numbers are in the range $[0..n^{O(1)}]$ then they can be sorted using bucket sort, in time $O(lg\ n)$ with $O(n\ lglg\ n/lg\ n)$ processors on CRCW model with arbitrary write rule for resolving write conflicts.

Sequential algorithms for some routing problems are improved and modified for parallel execution in Chapter 3. Presently no parallel algorithm is available for these problems. The following are some of the results obtained.

Planarity test for non-flippable Modules can be done in parallel in $O(lg\ n)$ time on CRCW model. Planarity test for flippable modules can be carried out in $O(lg^2 n)$ time on CREW model. Feasibility test for single row routing problem without inter-street crossings can be carried out in O(lg n) time on CRCW model. These algorithms when implemented sequentially require linear time. These algorithms require O(n) processors.

Each iteration of the heuristic suggested by Tsukiyama and Kuh for double row routing problem can be implemented in parallel in O(lg n*lglg n) time with $O(n^4/\text{lglg } n)$ processors, or alternatively, O(r + n lg n) time with $O(n^2)$ processors. The sequential implementation reported here takes less time than the one obtained by them.

List ranking problem has an O(lg n) time parallel solution on shared memory model and $O(\lg^2 n)$ time solutions on bounded degree network. In Chapter 4 it is shown that

i) If list ranking problem can be solved in O(lg n) time on bounded fan-in model, or in o(lg n) time on CRCW model, then so can all problems which can be solved in logarithmic space i.e., problems in class L. In other words, if the list ranking problem can be solved in O(f(n)) time then $L \subseteq CRCW(f(n))$.

ii) List traversal problem is complete for L under NC^1 reducibility.

For Delaunay triangulation problem $O(\lg^2 n)$ time parallel solution on shared memory model and $O(\lg^4 n)$ time solutions on bounded degree network are known in two dimensions. In Chapter 5 faster solutions in two dimensions are obtained and parallel algorithms for the problem in higher dimensions are sketched. It is shown that

i) The following upper (time,processor) trade-offs are possible for two dimensional Delaunay triangulation :

$(n, \lg n)$, $(\lg n, n^2)$, $(\lg\lg n, n^3)$, $(1, n^{3+e})$, where e is any arbitrary constant. First two bounds are for CREW and the next two for CRCW model.

ii) In d dimensions the following upper bound is obtained $(\lg\lg n, n^{d+1})$, $(1, n^{d+1+e})$.

iii) An $O(\lg^2 n)$ time algorithm for two dimensional Delaunay triangulation on n X n OTN resulting in AT^2 complexity of $O(n^2 \lg^6 n)$ has also been obtained. This is faster than all known algorithms for this problem on bounded degree networks. If m is the number of tetrahedra in three dimensional Delaunay triangulation then $O(m^{0.5} \lg n)$ time algorithm on m X n OTN for Delaunay Triangulation in three dimensions is obtained.

Euclidean Steiner Tree problem is NP-hard and it is highly unlikely that an efficient sequential or parallel solution exists. Therefore an attempt is made to come up with a reasonably efficient parallel heuristic in Chapter 6.

A new iterative improvement heuristic for Planar Euclidean Minimum Steiner tree problem is proposed. Based on Student's t-test, with 99 % confidence, it can be said that the proposed heuristic gives better Steiner Trees after only three iterations.

More over each iteration of the new heuristic can be efficiently implemented in parallel in $O(\lg^2 n)$ time with $O(n)$ processors on CREW model, or alternatively in $O(\lg n)$ time with $O(n^2)$ processors on CRCW PRAM. On n X n OTN each iteration takes $O(\lg^2 n)$ time.

CONTENTS