Probabilistic Algorithms and Complexity Classes

Ryuhei Uehara

The University of Electro-Communications

March 1998

Probabilistic Algorithms and Complexity Classes

Ryuhei Uehara

A Dissertation Submitted to the Department of Computer Science and Information Mathematics of the University of Electro-Communications in Partial Fulfillment of the Requirements for the Degree of Doctor of Science

March 1998

Probabilistic Algorithms and Complexity Classes

Approved by supervisory committee:

Chairperson: Professor Takumi Kasai

Member: Professor Shigeki Iwata

Member: Professor Kiyoshi Hashimoto

Member: Professor Takashi Yokomori

Member: Professor Tetsuro Nishino

Abstract

The main theme in this dissertation is a computation on a *probabilistic Turing machine* (PTM), which is a Turing machine with distinguished states called coin-tossing states. In general, it is assumed that a PTM has a fair coin. This is led by that any fixed biased coin can simulate another fixed biased coin in a constant expected time. We first consider the computation which will always terminates in *bounded* time, as opposed to *expected* time, and show that the coin with probability $\frac{1}{2}$ or $\frac{1}{3}$ is more expressive coin than the others. The complexity classes $\mathsf{PP}, \mathsf{BPP}$ and RP are usually defined via PTMs with a perfect random source. Vazirani and Vazirani proved that RP does not change even if the PTMs only have access to an *arbitrary* semi-random source. Later, Vazirani proved that BPP does not change, either. We show that PP collapses to BPP if the PTMs only have access to an *arbitrary* semi-random source. We also show that RP changes to NP while both BPP and PP change to PSPACE if the PTMs only have access to a *specific* semi-random source. Next problem is to identify a Boolean function f out of a given set of Boolean functions F by asking for the value of f at adaptively chosen inputs. We consider the cases where F consists of functions obtained from one function g on n inputs by replacing k inputs by given constants, and g is one of the functions disjunction, parity and threshold. For many of the problems we get almost optimal bounds. The *maximal path set* (MPS) problem is to find, given an undirected graph $G = (V, E)$, a maximal subset F of E such that the subgraph induced by F is a forest in which each connected component is a path. We design two parallel algorithms for the problem. The first algorithm is faster and more efficient than the previously known algorithm. The second one is faster, runs on a weaker computation model, and is more efficient for input graphs of bounded degree. We then use the results to design an NC approximation algorithm for a variation of the shortest superstring problem. We also consider the vertex counterpart of the MPS problem. The NP-completeness of several problems are shown in the last chapter. The first problem is the generalized popular puzzle known as Hi-Q. The second problem is the vertex cover problem on a cubic, planar, and 3-connected graph of girth greater than 3. Problems on a triangulation of the sphere are the last.

Acknowledgement

I would like to express my sincere thanks to Professor Takumi Kasai, my supervisor, who has granted me the chance to start my research, and has given me innumerable advice and unrelenting encouragements through the whole work.

I am also deeply indebted to the other members of the doctoral committee, Professor Shigeki Iwata, Professor Kiyoshi Hashimoto, Professor Takashi Yokomori, and Associate Processor Tetsuro Nishino. Their helpful suggestions, comments, and encouragement were essential for the preparation of this dissertation. I would like to note in particular Professor Shigeki Iwata, who gave me the chance to achieve my first result when I was an undergraduate student.

Along the way I had substantial help from my colleagues and friends. I would like to note in particular Professor Xin He of State University of New York at Buffalo, Professor Ingo Wegener of University Dortmund, Associate Professor Kensei Tsuchida of Toyo University, and Associate Professor Zhi-Zhong Chen of Tokyo Denki University, who have supported me and have achieved the results contributing for this dissertation. I would like to note in particular Dr. Zhi-Zhong Chen, who has given me innumerable advice and essential discussions on my researches. I wishes to thank Associate Professor Seinosuke Toda of Nihon University, and Associate Professor Osamu Watanabe of Tokyo Institute of Technology, for their helpful advice during the course of this work. I thank Associate Professor Misao Nagayama of Tokyo Woman's Christian University, Assistant Professor Sei'ichi Tani of Tokai University, and Dr. Koichi Yamazaki of the University of Electro-Communications for providing me with much help and advice. I am grateful to staff of Center for Information Science in Tokyo Woman's Christian University for their maintenance of the environment and resources which made this effort possible.

Contents

1 Introduction **1**

2 Preliminaries **7**
2.1 Graphs, digraphs, and trees . . . 7
2.2 Models of computation . . . 8
2.3 Model of an efficient computation . . . 11
2.4 Complexity classes with a semi-random source . . . 12
2.5 Computation trees with a semi-random source . . . 13
2.6 Complexity classes based on one Boolean function . . . 15

3 Efficient Simulations by a Biased Coin **17**
3.1 Sets efficiently simulated . . . 17
3.2 Relations between biased coins and a fair n-sided die . . . 18

4 Complexity Classes Characterized by Semi-Random Sources **21**
4.1 Probability of a worst tree . . . 21
4.2 The class δ-PP . . . 22
4.3 The class $^{\exists}\delta$-RP . . . 23
4.4 The classes $^{\exists}\delta$-BPP and $^{\exists}\delta$-PP . . . 23
4.5 The class ARP . . . 24
4.6 Remarks . . . 25
4.6.1 Characterization of a worst tree . . . 25
4.6.2 Complexity classes with irrationally δ-random sources . . . 26

5 Optimal Attribute-Efficient Learning of Disjunction, Parity, and Threshold Functions **29**
5.1 Basic results . . . 29
5.2 Disjunctions . . . 29
5.3 Parity functions . . . 31
5.4 Threshold functions . . . 35

6 Fast NC and RNC Algorithms for Finding a Maximal Set of Paths **39**
6.1 Parallel algorithms for the MPS problem . . . 39
6.1.1 Description of the RNC algorithm . . . 39
6.1.2 Complexity analysis . . . 40
6.1.3 Extension to digraphs . . . 42
6.1.4 The NC algorithm . . . 44
6.2 An application to shortest superstrings with flipping . . . 45

6.3 Parallel algorithms for the MLF problem . . . 49
6.3.1 Reducing the MLF problem to a simpler one . . . 49
6.3.2 Description of the RNC algorithm . . . 50
6.3.3 Complexity Analysis . . . 52
6.3.4 The NC algorithm . . . 54
6.4 Remarks . . . 54

7 NP-completeness of the problems on a simple structure 57
7.1 NP-completeness of Hi-Q . . . 57
7.2 NP-complete problems on a 3-connected cubic planar graph . . . 61
7.3 NP-complete problems on a triangulation of the sphere . . . 63