APPROXIMATION ALGORITHMS FOR COVERING AND PACKING PROBLEMS ON PATHS

by

ARINDAM PAL

Department of Computer Science and Engineering

Submitted

in fulfillment of the requirements of the degree of

Doctor of Philosophy

to the

Indian Institute of Technology Delhi

November 2012

To my family for their love, support and patience

Certificate

This is to certify that the thesis titled **Approximation Algorithms for Covering and Packing Problems on Paths** being submitted by Arindam Pal for the award of the degree of Doctor of Philosophy in Computer Science and Engineering is a record of original bonafide research work carried out by him under our guidance and supervision at the Department of Computer Science and Engineering, Indian Institute of Technology Delhi. The results contained in this thesis have not been submitted in part or full to any other university or institute for the award of any degree or diploma.

Naveen Garg
Professor
Department of
Computer Science and Engineering
Indian Institute of Technology Delhi

Amit Kumar
Professor
Department of
Computer Science and Engineering
Indian Institute of Technology Delhi

Acknowledgments

I am grateful to my doctoral research advisors Professor Amit Kumar and Professor Naveen Garg for being a constant source of inspiration. They gave me the freedom to work on problems that I like. They patiently listened to my ideas, even when some of them were not so great. They also gave valuable suggestions to improve the ideas and gave many ideas of their own. I am indebted to them for their help and advice during my graduate studies. This thesis would not have been possible without their cooperation.

I learned a great deal from my professors – Sandeep Sen, Amitabha Bagchi and Ragesh Jaiswal. The courses taught by them (along with my advisors) built the foundation of my research on different topics of mathematics and theoretical computer science. In addition, I had the good fortune of working with them on some research problems, which also shaped my thoughts on theoretical research.

I would also like to thank my coauthors Sambuddha, Venkat, Yogish, Prashant and Saurav. Without their help, many of the results in my thesis would not have seen the light of the day. Thank you very much for all those discussions and ideas that we had over the last few years.

I spent a great time at IIT Delhi with my friends Muralidhara, Rudra, Ayesha, Syamantak, Anamitra, Shibashis, Pravesh, Brojeswar, Manoj, Anuj, Chinmay, Swati and Mona. Sorry if I forgot anyone's name. You gave me company during both good and bad times. I will fondly remember and cherish those moments in the years to come.

I thank my good friends Dinesh, Vinay and Gopalda for giving me company. Special thanks to Dinesh for those intensive discussions at IBM Research and at my home. Thanks to Vinay for the good times we had in JNU. Thanks also to Gopalda for being such a good friend and mentor. I will always remember the good moments that I spent with all of you.

My special thanks to Roger Federer for playing such great game of tennis and providing quality entertainment over the last 10 years. You showed me by your own example, that hard work, determination, dedication and discipline can help a man to reach great heights. I love you Roger!

Last but not the least, I would like to thank my family members – my uncle Dr Debi Prasad Pal (Jatha), late father Bani Prasad (Baba), mother Bani (Ma), wife Sushmita, sister Anindita (Didi), brother-in-law Kanai (Dada), and nephews Arkajyoti and Debajyoti (Bhagna). Without their moral support, encouragement and cooperation, this thesis would not have been possible. I affectionately dedicate this thesis to them.

Abstract

Routing and scheduling problems are fundamental problems in combinatorial optimization, and also have many applications. Most variations of these problems are NP-Hard, so we need to use heuristics to solve these problems on large instances, which are fast and yet come close to the optimal value. In this thesis, we study the design and analysis of approximation algorithms for such problems. We focus on two important class of problems. The first is the Unsplittable Flow Problem and some of its variants and the second is the Resource Allocation for Job Scheduling Problem and some of its variants. The first is a *packing* problem, whereas the second is a *covering* problem.

In the Unsplittable Flow Problem, we are given a path or a tree, each edge of which has a capacity. We are also given a set of requests, each of which has a start vertex, an end vertex, a demand and a profit. The objective is to select a subset of requests so as to maximize the total profit, subject to the condition that on every edge the total demand of the selected requests is at most it's capacity. We also study variants of this problem such as Unsplittable Flow Problem with Rounds and Unsplittable Flow Problem with Bag Constraints. We give constant factor approximation algorithms for all of these problem on paths and trees under the *no-bottleneck assumption*. We also give a constant factor competitive algorithm for the Online Interval Coloring problem.

In the Resource Allocation for Job Scheduling Problem, the timeline is divided into a set of discrete timeslots.. We are given a set of jobs, each of which has a start time, an end time and a demand requirement. We are also given a set of resources, each of which has a start time, an end time, a capacity and a cost. A feasible solution is a set of resources satisfying the constraint that at any timeslot, the sum of the capacities offered by the resources is at least the demand required by the jobs active at that timeslot, *i.e.*, the selected resources must cover the jobs. The objective is to select a subset of resources of minimum cost, which will cover all the jobs.

This is called the *resource allocation problem* (ResAll). We consider the partial covering version (PartialResAll) and the prize-collecting version (PrizeCollectingResAll) of this problem. We give an $O(\log(n+m))$-approximation algorithm for the PartialResAll problem, where n is the number of jobs and m is the number of resources respectively. We also give a 4-approximation algorithm for the PrizeCollectingResAll problem.

Contents

1 Introduction **21**

1.1 Preliminaries . . . 22

1.1.1 Approximation algorithms and approximation factors . . . 22

1.1.2 Online algorithms and competitive ratios . . . 22

1.2 Routing problems in communication networks . . . 23

1.2.1 Notations . . . 23

1.2.2 Problem definition and motivation . . . 23

1.2.3 Related work . . . 26

1.2.4 Our contributions . . . 30

1.3 Resource allocation for scheduling jobs . . . 32

1.3.1 Problem definition . . . 33

1.3.2 Related work . . . 34

1.3.3 Our contributions . . . 36

1.4 Organization of the thesis . . . 37

2 The Round-UFP Problem **38**

2.1 Preliminaries . . . 38

2.2 Approximation Algorithms for Round-UFP on Paths . . . 39

2.2.1 A 3-approximation algorithm for uniform capacities and arbitrary demands . 39

2.2.2 A 24-approximation algorithm for arbitrary capacities and arbitrary demands 41

2.2.3 How bad can the congestion bound be? . . . 44

2.3 Approximation Algorithms for Round-UFP on Trees 45

3 The Max-UFP and the Bag-UFP Problems 47

3.1 Linear Programming formulation for Max-UFP 48

3.1.1 Integrality gap of the UFP-LP without NBA 48

3.1.2 Integrality gap of the UFP-LP with NBA 49

3.2 Approximation Algorithm for Max-UFP 50

3.2.1 Running time 52

3.3 Approximation Algorithm for Bag-UFP 52

3.3.1 Running time 54

3.4 Approximation Algorithm for Max-UFP on Trees 54

3.4.1 Running time 55

4 Online Algorithms for the Interval Coloring Problem 56

4.1 Preliminaries 56

4.2 Our algorithm 57

4.2.1 Small demands 58

4.2.2 Algorithm for Small Demands 60

4.3 Large demands 61

4.3.1 Running time 64

5 Scheduling Resources for a Partial Set of Jobs 65

5.1 Introduction 65

5.2 Problem Definition 66

5.3 Outline of the Main Algorithm 67

5.4 Overview of Our Algorithm 69

5.5 LSPC Problem: Proof of Theorem 5.4 75

5.5.1 DP Ordering 81

5.5.2 Computing the table A 81

5.5.3 Correctness of the Recurrence Relation (Figure 5.5) 81
5.6 Single Mountain Range: Proof of Theorem 5.2 83
5.6.1 First Step 83
5.6.2 Second Step 84
5.7 Overall Algorithm 88
5.8 The PrizeCollectingResAll problem 88

6 Conclusion and Open Problems **91**