Chandra Chekuri Klaus Jansen
José D.P. Rolim Luca Trevisan (Eds.)

Approximation, Randomization and Combinatorial Optimization

Algorithms and Techniques

8th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2005 and 9th International Workshop on Randomization and Computation, RANDOM 2005
Berkeley, CA, USA, August 2005, Proceedings

LNCS 3624

Springer

Chandra Chekuri Klaus Jansen
José D.P. Rolim Luca Trevisan (Eds.)

Approximation, Randomization and Combinatorial Optimization

Algorithms and Techniques

8th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2005
and 9th International Workshop on Randomization and Computation, RANDOM 2005
Berkeley, CA, USA, August 22-24, 2005
Proceedings

Springer

Volume Editors

Chandra Chekuri
Lucent Bell Labs
600 Mountain Avenue, Murray Hill, NJ 07974, USA
E-mail: chekuri@research.bell-labs.com

Klaus Jansen
University of Kiel, Institute for Computer Science
Olshausenstr. 40, 24098 Kiel, Germany
E-mail: kj@informatik.uni-kiel.de

José D.P. Rolim
Université de Genève, Centre Universitaire d'Informatique
24, Rue Général Dufour, 1211 Genève 4, Suisse
E-mail: jose.rolim@cui.unige.ch

Luca Trevisan
University of California, Computer Science Department
679 Soda Hall, Berkeley, CA 94720-1776, USA
E-mail: luca@cs.berkeley.edu

Library of Congress Control Number: 2005930720

CR Subject Classification (1998): F.2, G.2, G.1

ISSN	0302-9743
ISBN-10	3-540-28239-4 Springer Berlin Heidelberg New York
ISBN-13	978-3-540-28239-6 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springeronline.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Olgun Computergrafik
Printed on acid-free paper SPIN: 11538462 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 3624

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Lecture Notes in Computer Science

For information about Vols. 1–3523

please contact your bookseller or Springer

Vol. 3654: S. Jajodia, D. Wijesekera (Eds.), Data and Applications Security XIX. X, 353 pages. 2005.

Vol. 3639: P. Godefroid (Ed.), Model Checking Software. XI, 289 pages. 2005.

Vol. 3638: A. Butz, B. Fisher, A. Krüger, P. Olivier (Eds.), Smart Graphics. XI, 269 pages. 2005.

Vol. 3634: L. Ong (Ed.), Computer Science Logic. XI, 567 pages. 2005.

Vol. 3633: C. Bauzer Medeiros, M. Egenhofer, E. Bertino (Eds.), Advances in Spatial and Temporal Databases. XIII, 433 pages. 2005.

Vol. 3632: R. Nieuwenhuis (Ed.), Automated Deduction – CADE-20. XIII, 459 pages. 2005. (Subseries LNAI).

Vol. 3627: C. Jacob, M.L. Pilat, P.J. Bentley, J. Timmis (Eds.), Artificial Immune Systems. XII, 500 pages. 2005.

Vol. 3626: B. Ganter, G. Stumme, R. Wille (Eds.), Formal Concept Analysis. X, 349 pages. 2005. (Subseries LNAI).

Vol. 3625: S. Kramer, B. Pfahringer (Eds.), Inductive Logic Programming. XIII, 427 pages. 2005. (Subseries LNAI).

Vol. 3624: C. Chekuri, K. Jansen, J.D.P. Rolim, L. Trevisan (Eds.), Approximation, Randomization and Combinatorial Optimization. XI, 495 pages. 2005.

Vol. 3623: M. Liśkiewicz, R. Reischuk (Eds.), Fundamentals of Computation Theory. XV, 576 pages. 2005.

Vol. 3621: V. Shoup (Ed.), Advances in Cryptology – CRYPTO 2005. XI, 568 pages. 2005.

Vol. 3620: H. Muñoz-Avila, F. Ricci (Eds.), Case-Based Reasoning Research and Development. XV, 654 pages. 2005. (Subseries LNAI).

Vol. 3619: X. Lu, W. Zhao (Eds.), Networking and Mobile Computing. XXIV, 1299 pages. 2005.

Vol. 3615: B. Ludäscher, L. Raschid (Eds.), Data Integration in the Life Sciences. XII, 344 pages. 2005. (Subseries LNBI).

Vol. 3608: F. Dehne, A. López-Ortiz, J.-R. Sack (Eds.), Algorithms and Data Structures. XIV, 446 pages. 2005.

Vol. 3607: J.-D. Zucker, L. Saitta (Eds.), Abstraction, Reformulation and Approximation. XII, 376 pages. 2005. (Subseries LNAI).

Vol. 3606: V. Malyshkin (Ed.), Parallel Computing Technologies. XII, 470 pages. 2005.

Vol. 3603: J. Hurd, T. Melham (Eds.), Theorem Proving in Higher Order Logics. IX, 409 pages. 2005.

Vol. 3602: R. Eigenmann, Z. Li, S.P. Midkiff (Eds.), Languages and Compilers for High Performance Computing. IX, 486 pages. 2005.

Vol. 3599: U. Aßmann, M. Aksit, A. Rensink (Eds.), Model Driven Architecture. X, 235 pages. 2005.

Vol. 3598: H. Murakami, H. Nakashima, H. Tokuda, M. Yasumura, Ubiquitous Computing Systems. XIII, 275 pages. 2005.

Vol. 3597: S. Shimojo, S. Ichii, T.W. Ling, K.-H. Song (Eds.), Web and Communication Technologies and Internet-Related Social Issues - HSI 2005. XIX, 368 pages. 2005.

Vol. 3596: F. Dau, M.-L. Mugnier, G. Stumme (Eds.), Conceptual Structures: Common Semantics for Sharing Knowledge. XI, 467 pages. 2005. (Subseries LNAI).

Vol. 3595: L. Wang (Ed.), Computing and Combinatorics. XVI, 995 pages. 2005.

Vol. 3594: J.C. Setubal, S. Verjovski-Almeida (Eds.), Advances in Bioinformatics and Computational Biology. XIV, 258 pages. 2005. (Subseries LNBI).

Vol. 3592: S. Katsikas, J. Lopez, G. Pernul (Eds.), Trust and Privacy in Digital Business. XII, 332 pages. 2005.

Vol. 3587: P. Perner, A. Imiya (Eds.), Machine Learning and Data Mining in Pattern Recognition. XVII, 695 pages. 2005. (Subseries LNAI).

Vol. 3586: A.P. Black (Ed.), ECOOP 2005 - Object-Oriented Programming. XVII, 631 pages. 2005.

Vol. 3584: X. Li, S. Wang, Z.Y. Dong (Eds.), Advanced Data Mining and Applications. XIX, 835 pages. 2005. (Subseries LNAI).

Vol. 3583: R.W. H. Lau, Q. Li, R. Cheung, W. Liu (Eds.), Advances in Web-Based Learning – ICWL 2005. XIV, 420 pages. 2005.

Vol. 3582: J. Fitzgerald, I.J. Hayes, A. Tarlecki (Eds.), FM 2005: Formal Methods. XIV, 558 pages. 2005.

Vol. 3581: S. Miksch, J. Hunter, E. Keravnou (Eds.), Artificial Intelligence in Medicine. XVII, 547 pages. 2005. (Subseries LNAI).

Vol. 3580: L. Caires, G.F. Italiano, L. Monteiro, C. Palamidessi, M. Yung (Eds.), Automata, Languages and Programming. XXV, 1477 pages. 2005.

Vol. 3579: D. Lowe, M. Gaedke (Eds.), Web Engineering. XXII, 633 pages. 2005.

Vol. 3578: M. Gallagher, J. Hogan, F. Maire (Eds.), Intelligent Data Engineering and Automated Learning - IDEAL 2005. XVI, 599 pages. 2005.

Vol. 3577: R. Falcone, S. Barber, J. Sabater-Mir, M.P. Singh (Eds.), Trusting Agents for Trusting Electronic Societies. VIII, 235 pages. 2005. (Subseries LNAI).

Vol. 3576: K. Etessami, S.K. Rajamani (Eds.), Computer Aided Verification. XV, 564 pages. 2005.

Vol. 3575: S. Wermter, G. Palm, M. Elshaw (Eds.), Biomimetic Neural Learning for Intelligent Robots. IX, 383 pages. 2005. (Subseries LNAI).

Vol. 3574: C. Boyd, J.M. González Nieto (Eds.), Information Security and Privacy. XIII, 586 pages. 2005.

Vol. 3573: S. Etalle (Ed.), Logic Based Program Synthesis and Transformation. VIII, 279 pages. 2005.

Vol. 3572: C. De Felice, A. Restivo (Eds.), Developments in Language Theory. XI, 409 pages. 2005.

Vol. 3571: L. Godo (Ed.), Symbolic and Quantitative Approaches to Reasoning with Uncertainty. XVI, 1028 pages. 2005. (Subseries LNAI).

Vol. 3570: A. S. Patrick, M. Yung (Eds.), Financial Cryptography and Data Security. XII, 376 pages. 2005.

Vol. 3569: F. Bacchus, T. Walsh (Eds.), Theory and Applications of Satisfiability Testing. XII, 492 pages. 2005.

Vol. 3568: W.-K. Leow, M.S. Lew, T.-S. Chua, W.-Y. Ma, L. Chaisorn, E.M. Bakker (Eds.), Image and Video Retrieval. XVII, 672 pages. 2005.

Vol. 3567: M. Jackson, D. Nelson, S. Stirk (Eds.), Database: Enterprise, Skills and Innovation. XII, 185 pages. 2005.

Vol. 3566: J.-P. Banâtre, P. Fradet, J.-L. Giavitto, O. Michel (Eds.), Unconventional Programming Paradigms. XI, 367 pages. 2005.

Vol. 3565: G.E. Christensen, M. Sonka (Eds.), Information Processing in Medical Imaging. XXI, 777 pages. 2005.

Vol. 3564: N. Eisinger, J. Małuszyński (Eds.), Reasoning Web. IX, 319 pages. 2005.

Vol. 3562: J. Mira, J.R. Álvarez (Eds.), Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach, Part II. XXIV, 636 pages. 2005.

Vol. 3561: J. Mira, J.R. Álvarez (Eds.), Mechanisms, Symbols, and Models Underlying Cognition, Part I. XXIV, 532 pages. 2005.

Vol. 3560: V.K. Prasanna, S. Iyengar, P.G. Spirakis, M. Welsh (Eds.), Distributed Computing in Sensor Systems. XV, 423 pages. 2005.

Vol. 3559: P. Auer, R. Meir (Eds.), Learning Theory. XI, 692 pages. 2005. (Subseries LNAI).

Vol. 3558: V. Torra, Y. Narukawa, S. Miyamoto (Eds.), Modeling Decisions for Artificial Intelligence. XII, 470 pages. 2005. (Subseries LNAI).

Vol. 3557: H. Gilbert, H. Handschuh (Eds.), Fast Software Encryption. XI, 443 pages. 2005.

Vol. 3556: H. Baumeister, M. Marchesi, M. Holcombe (Eds.), Extreme Programming and Agile Processes in Software Engineering. XIV, 332 pages. 2005.

Vol. 3555: T. Vardanega, A.J. Wellings (Eds.), Reliable Software Technology – Ada-Europe 2005. XV, 273 pages. 2005.

Vol. 3554: A. Dey, B. Kokinov, D. Leake, R. Turner (Eds.), Modeling and Using Context. XIV, 572 pages. 2005. (Subseries LNAI).

Vol. 3553: T.D. Hämäläinen, A.D. Pimentel, J. Takala, S. Vassiliadis (Eds.), Embedded Computer Systems: Architectures, Modeling, and Simulation. XV, 476 pages. 2005.

Vol. 3552: H. de Meer, N. Bhatti (Eds.), Quality of Service – IWQoS 2005. XVIII, 400 pages. 2005.

Vol. 3551: T. Härder, W. Lehner (Eds.), Data Management in a Connected World. XIX, 371 pages. 2005.

Vol. 3548: K. Julisch, C. Kruegel (Eds.), Intrusion and Malware Detection and Vulnerability Assessment. X, 241 pages. 2005.

Vol. 3547: F. Bomarius, S. Komi-Sirviö (Eds.), Product Focused Software Process Improvement. XIII, 588 pages. 2005.

Vol. 3546: T. Kanade, A. Jain, N.K. Ratha (Eds.), Audio- and Video-Based Biometric Person Authentication. XX, 1134 pages. 2005.

Vol. 3544: T. Higashino (Ed.), Principles of Distributed Systems. XII, 460 pages. 2005.

Vol. 3543: L. Kutvonen, N. Alonistioti (Eds.), Distributed Applications and Interoperable Systems. XI, 235 pages. 2005.

Vol. 3542: H.H. Hoos, D.G. Mitchell (Eds.), Theory and Applications of Satisfiability Testing. XIII, 393 pages. 2005.

Vol. 3541: N.C. Oza, R. Polikar, J. Kittler, F. Roli (Eds.), Multiple Classifier Systems. XII, 430 pages. 2005.

Vol. 3540: H. Kalviainen, J. Parkkinen, A. Kaarna (Eds.), Image Analysis. XXII, 1270 pages. 2005.

Vol. 3539: K. Morik, J.-F. Boulicaut, A. Siebes (Eds.), Local Pattern Detection. XI, 233 pages. 2005. (Subseries LNAI).

Vol. 3538: L. Ardissono, P. Brna, A. Mitrovic (Eds.), User Modeling 2005. XVI, 533 pages. 2005. (Subseries LNAI).

Vol. 3537: A. Apostolico, M. Crochemore, K. Park (Eds.), Combinatorial Pattern Matching. XI, 444 pages. 2005.

Vol. 3536: G. Ciardo, P. Darondeau (Eds.), Applications and Theory of Petri Nets 2005. XI, 470 pages. 2005.

Vol. 3535: M. Steffen, G. Zavattaro (Eds.), Formal Methods for Open Object-Based Distributed Systems. X, 323 pages. 2005.

Vol. 3534: S. Spaccapietra, E. Zimányi (Eds.), Journal on Data Semantics III. XI, 213 pages. 2005.

Vol. 3533: M. Ali, F. Esposito (Eds.), Innovations in Applied Artificial Intelligence. XX, 858 pages. 2005. (Subseries LNAI).

Vol. 3532: A. Gómez-Pérez, J. Euzenat (Eds.), The Semantic Web: Research and Applications. XV, 728 pages. 2005.

Vol. 3531: J. Ioannidis, A. Keromytis, M. Yung (Eds.), Applied Cryptography and Network Security. XI, 530 pages. 2005.

Vol. 3530: A. Prinz, R. Reed, J. Reed (Eds.), SDL 2005: Model Driven. XI, 361 pages. 2005.

Vol. 3528: P.S. Szczepaniak, J. Kacprzyk, A. Niewiadomski (Eds.), Advances in Web Intelligence. XVII, 513 pages. 2005. (Subseries LNAI).

Vol. 3527: R. Morrison, F. Oquendo (Eds.), Software Architecture. XII, 263 pages. 2005.

Vol. 3526: S. B. Cooper, B. Löwe, L. Torenvliet (Eds.), New Computational Paradigms. XVII, 574 pages. 2005.

Vol. 3525: A.E. Abdallah, C.B. Jones, J.W. Sanders (Eds.), Communicating Sequential Processes. XIV, 321 pages. 2005.

Vol. 3524: R. Barták, M. Milano (Eds.), Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. XI, 320 pages. 2005.

Preface

This volume contains the papers presented at the 8th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX 2005) and the 9th International Workshop on Randomization and Computation (RANDOM 2005), which took place concurrently at the University of California in Berkeley, on August 22–24, 2005. APPROX focuses on algorithmic and complexity issues surrounding the development of efficient approximate solutions to computationally hard problems, and APPROX 2005 was the eighth in the series after Aalborg (1998), Berkeley (1999), Saarbrücken (2000), Berkeley (2001), Rome (2002), Princeton (2003), and Cambridge (2004). RANDOM is concerned with applications of randomness to computational and combinatorial problems, and RANDOM 2005 was the ninth workshop in the series following Bologna (1997), Barcelona (1998), Berkeley (1999), Geneva (2000), Berkeley (2001), Harvard (2002), Princeton (2003), and Cambridge (2004).

Topics of interest for APPROX and RANDOM are: design and analysis of approximation algorithms, hardness of approximation, small space and data streaming algorithms, sub-linear time algorithms, embeddings and metric space methods, mathematical programming methods, coloring and partitioning, cuts and connectivity, geometric problems, game theory and applications, network design and routing, packing and covering, scheduling, design and analysis of randomized algorithms, randomized complexity theory, pseudorandomness and derandomization, random combinatorial structures, random walks/Markov chains, expander graphs and randomness extractors, probabilistic proof systems, random projections and embeddings, error-correcting codes, average-case analysis, property testing, computational learning theory, and other applications of approximation and randomness.

The volume contains 20 contributed papers selected by the APPROX Program Committee out of 50 submissions, and 21 contributed papers selected by the RANDOM Program Committee out of 51 submissions.

We would like to thank all of the authors who submitted papers, the members of the program committees

APPROX 2005

Matthew Andrews, Lucent Bell Labs
Avrim Blum, CMU
Moses Charikar, Princeton University
Chandra Chekuri, Lucent Bell Labs (Chair)
Julia Chuzhoy, MIT
Uriel Feige, Microsoft Research and Weizmann Institute
Naveen Garg, IIT Delhi
Howard Karloff, AT&T Labs – Research
Stavros Kolliopoulos, University of Athens
Adam Meyerson, UCLA
Seffi Naor, Technion
Santosh Vempala, MIT

RANDOM 2005

Dorit Aharonov, Hebrew University
Boaz Barak, IAS and Princeton University
Funda Ergun, Simon Fraser University
Johan Håstad, KTH Stockholm
Chi-Jen Lu, Academia Sinica
Milena Mihail, Georgia Institute of Technology
Robert Krauthgamer, IBM Almaden
Dana Randall, Georgia Institute of Technology
Amin Shokrollahi, EPF Lausanne
Angelika Steger, ETH Zurich
Luca Trevisan, UC Berkeley (Chair)

and the external subreferees Scott Aaronson, Dimitris Achlioptas, Mansoor Alicherry, Andris Ambainis, Aaron Archer, Nikhil Bansal, Tugkan Batu, Gerard Ben Arous, Michael Ben-Or, Eli Ben-Sasson, Petra Berenbrink, Randeep Bhatia, Nayantara Bhatnagar, Niv Buchbinder, Shuchi Chawla, Joseph Cheriyan, Roee Engelberg, Lance Fortnow, Tom Friedetzky, Mikael Goldmann, Daniel Gottesman, Sam Greenberg, Anupam Gupta, Venkat Guruswami, Tom Hayes, Monika Henzinger, Danny Hermelin, Nicole Immorlica, Piotr Indyk, Adam Kalai, Julia Kempe, Claire Kenyon, Jordan Kerenidis, Sanjeev Khanna, Amit Kumar, Ravi Kumar, Nissan Lev-Tov, Liane Lewin, Laszlo Lovasz, Elitza Maneva, Michael Mitzenmacher, Cris Moore, Michele Mosca, Kamesh Munagala, Noam Nisan, Ryan O'Donnell, Martin Pal, Vinayaka Pandit, David Peleg, Yuval Rabani, Dror Rawitz, Danny Raz, Adi Rosen, Ronitt Rubinfeld, Cenk Sahinalp, Alex Samorodnitsky, Gabi Scalosub, Leonard Schulman, Roy Schwartz, Pranab Sen, Mehrdad Shahshahani, Amir Shpilka, Anastasios Sidiropoulos, Greg Sorkin, Adam Smith, Ronen Shaltiel, Maxim Sviridenko, Amnon-Ta-Shma, Emre Telatar, Alex Vardy, Eric Vigoda, Da-Wei Wang, Ronald de Wolf, David Woodruff, Hsin-Lung Wu, and Lisa Zhang.

We gratefully acknowledge the support from the Lucent Bell Labs of Murray Hill, the Computer Science Division of the University of California at Berkeley, the Institute of Computer Science of the Christian-Albrechts-Universität zu Kiel and the Department of Computer Science of the University of Geneva. We also thank Ute Iaquinto and Parvaneh Karimi Massouleh for their help.

August 2005

Chandra Chekuri and Luca Trevisan, Program Chairs
Klaus Jansen and José D.P. Rolim, Workshop Chairs

Table of Contents

Contributed Talks of APPROX

The Network as a Storage Device: Dynamic Routing with Bounded Buffers 1
Stanislav Angelov, Sanjeev Khanna, and Keshav Kunal

Rounding Two and Three Dimensional Solutions of the SDP Relaxation of MAX CUT 14
Adi Avidor and Uri Zwick

What Would Edmonds Do? Augmenting Paths and Witnesses for Degree-Bounded MSTs 26
Kamalika Chaudhuri, Satish Rao, Samantha Riesenfeld, and Kunal Talwar

A Rounding Algorithm for Approximating Minimum Manhattan Networks 40
Victor Chepoi, Karim Nouioua, and Yann Vaxès

Packing Element-Disjoint Steiner Trees 52
Joseph Cheriyan and Mohammad R. Salavatipour

Approximating the Bandwidth of Caterpillars 62
Uriel Feige and Kunal Talwar

Where's the Winner? Max-Finding and Sorting with Metric Costs 74
Anupam Gupta and Amit Kumar

What About Wednesday? Approximation Algorithms for Multistage Stochastic Optimization 86
Anupam Gupta, Martin Pál, Ramamoorthi Ravi, and Amitabh Sinha

The Complexity of Making Unique Choices: Approximating 1-in-k SAT ... 99
Venkatesan Guruswami and Luca Trevisan

Approximating the Distortion 111
Alexander Hall and Christos Papadimitriou

Approximating the Best-Fit Tree Under L_p Norms 123
Boulos Harb, Sampath Kannan, and Andrew McGregor

Beating a Random Assignment 134
Gustav Hast

Scheduling on Unrelated Machines
Under Tree-Like Precedence Constraints 146
V.S. Anil Kumar, Madhav V. Marathe, Srinivasan Parthasarathy, and Aravind Srinivasan

Approximation Algorithms for Network Design and Facility Location
with Service Capacities 158
Jens Maßberg and Jens Vygen

Finding Graph Matchings in Data Streams 170
Andrew McGregor

A Primal-Dual Approximation Algorithm for Partial Vertex Cover:
Making Educated Guesses 182
Julián Mestre

Efficient Approximation of Convex Recolorings 192
Shlomo Moran and Sagi Snir

Approximation Algorithms for Requirement Cut on Graphs 209
Viswanath Nagarajan and Ramamoorthi Ravi

Approximation Schemes for Node-Weighted Geometric
Steiner Tree Problems 221
Jan Remy and Angelika Steger

Towards Optimal Integrality Gaps for Hypergraph Vertex Cover
in the Lovász-Schrijver Hierarchy 233
Iannis Tourlakis

Contributed Talks of RANDOM

Bounds for Error Reduction with Few Quantum Queries 245
Sourav Chakraborty, Jaikumar Radhakrishnan, and Nandakumar Raghunathan

Sampling Bounds for Stochastic Optimization 257
Moses Charikar, Chandra Chekuri, and Martin Pál

An Improved Analysis of Mergers 270
Zeev Dvir and Amir Shpilka

Finding a Maximum Independent Set in a Sparse Random Graph 282
Uriel Feige and Eran Ofek

On the Error Parameter of Dispersers 294
Ronen Gradwohl, Guy Kindler, Omer Reingold, and Amnon Ta-Shma

Tolerant Locally Testable Codes 306
Venkatesan Guruswami and Atri Rudra

A Lower Bound on List Size for List Decoding 318
Venkatesan Guruswami and Salil Vadhan

A Lower Bound for Distribution-Free Monotonicity Testing 330
Shirley Halevy and Eyal Kushilevitz

On Learning Random DNF Formulas Under the Uniform Distribution 342
Jeffrey C. Jackson and Rocco A. Servedio

Derandomized Constructions
of k-Wise (Almost) Independent Permutations 354
Eyal Kaplan, Moni Naor, and Omer Reingold

Testing Periodicity 366
Oded Lachish and Ilan Newman

The Parity Problem in the Presence of Noise,
Decoding Random Linear Codes, and the Subset Sum Problem 378
Vadim Lyubashevsky

The Online Clique Avoidance Game on Random Graphs 390
Martin Marciniszyn, Reto Spöhel, and Angelika Steger

A Generating Function Method for the Average-Case Analysis of DPLL .. 402
Rémi Monasson

A Continuous-Discontinuous Second-Order Transition in the Satisfiability
of Random Horn-SAT Formulas 414
Cristopher Moore, Gabriel Istrate, Demetrios Demopoulos,
and Moshe Y. Vardi

Mixing Points on a Circle 426
Dana Randall and Peter Winkler

Derandomized Squaring of Graphs 436
Eyal Rozenman and Salil Vadhan

Tight Bounds for String Reconstruction Using Substring Queries 448
Dekel Tsur

Reconstructive Dispersers and Hitting Set Generators 460
Christopher Umans

The Tensor Product of Two Codes Is Not Necessarily Robustly Testable .. 472
Paul Valiant

Fractional Decompositions of Dense Hypergraphs 482
Raphael Yuster

The Network as a Storage Device: Dynamic Routing with Bounded Buffers

Stanislav Angelov*, Sanjeev Khanna**, and Keshav Kunal***

University of Pennsylvania, Philadelphia, PA 19104, USA
{angelov,sanjeev,kkunal}@cis.upenn.edu

Abstract. We study dynamic routing in store-and-forward packet networks where each network link has bounded buffer capacity for receiving incoming packets and is capable of transmitting a fixed number of packets per unit of time. At any moment in time, packets are injected at various network nodes with each packet specifying its destination node. The goal is to maximize the *throughput*, defined as the number of packets delivered to their destinations.

In this paper, we make some progress in understanding what is achievable on various network topologies. For line networks, Nearest-to-Go (NTG), a natural greedy algorithm, was shown to be $O(n^{2/3})$-competitive by Aiello *et al* [1]. We show that NTG is $\tilde{O}(\sqrt{n})$-competitive, essentially matching an identical lower bound known on the performance of *any* greedy algorithm shown in [1]. We show that if we allow the online routing algorithm to make centralized decisions, there is indeed a randomized polylog(n)-competitive algorithm for line networks as well as rooted tree networks, where each packet is destined for the root of the tree. For grid graphs, we show that NTG has a performance ratio of $\tilde{\Theta}(n^{2/3})$ while no greedy algorithm can achieve a ratio better than $\Omega(\sqrt{n})$. Finally, for an arbitrary network with m edges, we show that NTG is $\tilde{\Theta}(m)$-competitive, improving upon an earlier bound of $O(mn)$ [1].

1 Introduction

The problem of dynamically routing packets is central to traffic management on large scale data networks. Packet data networks typically employ store-and-forward routing where network links (or routers) store incoming packets and schedule them to be forwarded in a suitable manner. Internet is a well-known example of such a network. In this paper, we consider routing algorithms for store-and-forward networks in a dynamic setting where packets continuously arrive in an arbitrary manner over a period of time and each packet specifies its

* Supported in part by NSF Career Award CCR-0093117, NSF Award ITR 0205456 and NIGMS Award 1-P20-GM-6912-1.

** Supported in part by an NSF Career Award CCR-0093117, NSF Award CCF-0429836, and a US-Israel Binational Science Foundation Grant.

*** Supported in part by an NSF Career Award CCR-0093117 and NSF Award CCF-0429836.

C. Chekuri et al. (Eds.): APPROX and RANDOM 2005, LNCS 3624, pp. 1–13, 2005.

destination. The goal is to maximize the *throughput*, defined as the number of packets that are delivered to their destinations. This model, known as the *Competitive Network Throughput* (CNT) model, was introduced by Aiello *et al* [1]. The model allows for both *adaptive* and *non-adaptive* routing of packets, where in the latter case each packet also specifies a path to its destination node. Note that there is no distinction between the two settings on line and tree networks.

Aiello *et al* analyzed *greedy* algorithms, which are characterized as ones that accept and forward packets whenever possible. For the non-adaptive setting, they showed that Nearest-to-Go (NTG), a natural greedy algorithm that favors packets whose remaining distance is the shortest, is $O(mn)$-competitive[1] on any network with n nodes and m edges. On line networks, NTG was shown to be $O(n^{2/3})$-competitive provided each link has buffer capacity at least 2.

Our Results. Designing routing algorithms in the CNT model is challenging since buffer space is limited at any individual node and a good algorithm should use all nodes to buffer packets and not just nodes where packets are injected. It is like viewing the entire network as a storage device where new information is injected at locations chosen by the adversary. Regions in the network with high injection rate need to continuously move packets to intermediate network regions with low activity. However, if many nodes simultaneously send packets to the same region, the resulting congestion would lead to buffer overflows. Worse still, the adversary may suddenly raise the packet injection rate in the region of low activity. In general, an adversary can employ different attack strategies in different parts of the network. How should a routing algorithm coordinate the movement of packets in presence of buffer constraints? This turns out to be a difficult question even on simple networks such as lines and trees.

In this paper, we continue the thread of research started in [1]. Throughout the remainder of this paper, we will assume that all network links have a uniform buffer size B and bandwidth 1. We obtain the following results:

Line Networks: For line networks with $B > 1$, we show that NTG is $\tilde{O}(\sqrt{n})$-competitive, essentially matching an $\Omega(\sqrt{n})$ lower bound of [1]. A natural question is whether non-greedy algorithms can perform better. For $B = 1$, we show that any deterministic algorithm is $\Omega(n)$-competitive, strengthening a result of [1], and give a randomized $\tilde{O}(\sqrt{n})$-competitive algorithm. For $B > 1$, however, no super-constant lower bounds are known leaving a large gap between known lower bounds and what is achievable by greedy algorithms. We show that centralized decisions can improve performance exponentially by designing a randomized $O(\log^3 n)$-competitive algorithm, referred to as *Merge and Deliver*.

Tree Networks: For tree networks of height h, it was shown in [1] that any greedy algorithm has a competitive ratio of $\Omega(n/h)$. Building on the ideas used

[1] The exact bound shown in [1] is $O(mD)$ where D is the maximum length of any given path in the network. We state here the worst-case bound with $D = \Omega(n)$.

Table 1. Summary of results; the bounds obtained in this paper are in boldface.

Algorithm	Line	Tree	Grid	General
Greedy	$\Omega(\sqrt{n})$ [1]	$\Omega(n)$ [1]	$\boldsymbol{\Omega(\sqrt{n})}$	$\boldsymbol{\Omega(m)}$
NTG	$\boldsymbol{\tilde{O}(\sqrt{n})}$	$\boldsymbol{\tilde{O}(n)}$	$\boldsymbol{\tilde{\Theta}(n^{2/3})}$	$\boldsymbol{\tilde{O}(m)}$
Previous bounds	$O(n^{2/3})$ [1]	$O(nh)$ [1]	-	$O(mn)$ [1]
Merge and Deliver (M&D)	$\boldsymbol{O(\log^3 n)}$	$\boldsymbol{O(h \log^2 n)}$	-	-

in the line algorithm, we give a $O(\log^2 n)$-competitive algorithm when all packets are destined for the root. Such tree networks are motivated by the work on buffer overflows of merging streams of packets by Kesselman *et al* [2] and the work on information gathering by Kothapalli and Scheideler [3] and Azar and Zachut [4]. The result extends to a randomized $O(h \log^2 n)$-competitive algorithm when packet destinations are arbitrary.

Grid Networks: For adaptive setting in grid networks, we show that NTG with one-bend routing is $\tilde{\Theta}(n^{2/3})$-competitive. We establish a lower bound of $\Omega(\sqrt{n})$ on the competitive ratio for greedy algorithms with shortest path routing.

General Networks: Finally, for arbitrary network topologies, we show that any greedy algorithm is $\Omega(m)$-competitive and prove that NTG is, in fact, $\tilde{\Theta}(m)$-competitive. These results hold for both adaptive and non-adaptive settings, where NTG routes on a shortest path in the adaptive case.

Related Work. Dynamic store-and-forward routing networks have been studied extensively. An excellent survey of packet drop policies in communication networks can be found in [5]. Much of the earlier work has focused on the issue of stability with packets being injected in either probabilistic or adversarial manner. In stability analysis the goal is to understand how the buffer size at each link needs to grow as a function of the packet injection rate so that packets are never dropped. A *stable* protocol is one where the maximum buffer size does not grow with time. For work in the probabilistic setting, see [6–10]. Adversarial queuing theory introduced by Borodin *et al* [11] has also been used to study the stability of protocols and it has been shown in [11, 12] that certain greedy algorithms are unstable, that is, require unbounded buffer sizes. In particular, NTG is unstable.

The idea of using the entire network to store packets effectively has been used in [13] but in their model, packets can not be buffered while in transit and the performance measure is the time required to deliver all packets. Moreover, packets never get dropped because there is no limit on the number of packets which can be stored at their respective source nodes.

Throughput competitiveness was highlighted as a network performance measure by Awerbuch *et al* in [14]. They used adversarial traffic to analyze store-and-forward routing algorithms in [15], but they compared their throughput to an adversary restricted to a certain class of strategies and with smaller buffers.

Kesselman *et al* [2] study the throughput competitiveness of work-conserving algorithms on line and tree networks when the adversary is a work-conserving algorithm too. Work-conserving algorithms always forward a packet if possible but unlike greedy, they need not accept packets when there is space in the buffer. However, they do not make assumptions about uniform link bandwidths or buffer sizes as in [1] and our model. They also consider the case when packets have different weights. Non-preemptive policies for packets with different values but with unit sized buffers have been analyzed in [16, 17].

In a recent parallel work, Azar and Zachut [4] also obtain centralized algorithms with polylog(n) competitive ratios on lines. They first obtain a $O(\log n)$-competitive deterministic algorithm for the special case when all packets have the same destination (which is termed information gathering) and then show that it can be extended to a randomized algorithm with $O(\log^2 n)$-competitive ratio for the general case. Their result can be extended to get $O(h \log n)$-competitive ratio on trees. Information gathering problem is similar to our notion of balanced instances (see Section 2.3) though their techniques are very different from ours – they construct an online reduction from the fractional buffers packet routing with bounded delay problem to fractional information gathering and the former is solved by an extension of the work of Awerbuch *et al* [14] for the discrete version. The fractional algorithm for information gathering is then transformed to a discrete one.

2 Preliminaries

2.1 Model and Problem Statement

We model the network as a directed graph on n nodes and m links. A node has at most I traffic ports where new packets can be injected, at most one at each port. Each link has an output port at its tail with a capacity $B > 0$ buffer, and an input port at its head that can store 1 packet. We assume uniform buffer size B at each link and bandwidth of each link to be 1.

Time is synchronous and each time step consists of *forwarding* and *switching* sub-steps. During the forwarding phase, each link selects at most one packet from its output buffer according to an *output scheduling policy* and forwards it to its input buffer. During the switching phase, a node clears all packets from its traffic ports and input ports at its incoming link(s). It delivers packets if the node is their destination or assigns them to the output port of the outgoing link on their respective paths. When more than B packets are assigned to a link's output buffer, packets are discarded based on a *contention-resolution policy.* We consider preemptive contention-resolution policies that can replace packets already stored at the buffer with new packets. A routing protocol specifies the output scheduling and contention-resolution policy. We are interested in *online* policies which make decisions with no knowledge of future packet arrivals.

Each injected packet comes with a specified destination. We assume that the destination is different from the source, otherwise the packet is routed optimally by any algorithm and does not interfere with other packets. The *goal* of the routing algorithm is to maximize throughput, that is, the total number of packets delivered by it. We distinguish between two types of algorithms, namely *centralized* and *distributed*. A centralized algorithm makes coordinated decisions at each node taking into account the state of the entire network while a distributed algorithm requires that each node make its decisions based on local information only. Distributed algorithms are of great practical interest for large networks. Centralized algorithms, on the other hand, give us insight into the inherent complexity of the problem due to the online nature of the input.

2.2 Useful Background Results and Definitions

The following lemma which gives an upper bound on the packets that can be absorbed over a time interval, will be a recurrent idea while analyzing algorithms.

Lemma 1. [1] In a network with m links, the number of packets that can be delivered and buffered in a time interval of length T units by *any* algorithm is $O(mT/d + mB)$, where $d > 0$ is a lower bound on the number of links in the shortest path to the destination for each injected packet.

The proof bounds the available bandwidth on all links and compares it against the minimum bandwidth required to deliver a packet injected during the time interval. The $O(mB)$ term accounts for packets buffered at the beginning and the end of the interval, which can be arbitrarily close to their destination.

An important class of distributed algorithms is *greedy algorithms* where each link always accepts an incoming packet if there is available buffer space and always forwards a packet if its buffer is non-empty. Based on how contention is resolved when receiving/forwarding packets, we obtain different algorithms. *Nearest-To-Go* (NTG) is a natural greedy algorithm which always selects a shortest path to route a packet and prefers a packet that has shortest remaining distance to travel, in both choosing packets to accept or forward.

Line and tree networks are of special interest to us, where there is a unique path between every source and destination pair. A *line network* on n nodes is a directed path with nodes labeled $1, 2, \ldots, n$ and a link from node i to $i+1$, for $i \in [1, n)$. A *tree network* is a rooted tree with links directed towards the root. Note that there is a one-to-one correspondence between links and nodes (for all but one node) on lines and trees. For simplicity, whenever we refer to a node's buffer, we mean the output buffer of its unique outgoing link.

A simple useful property of any greedy algorithm is as follows:

Lemma 2. [1] *If at some time t, a greedy algorithm on a line network has buffered $k \leq nB$ packets, then it delivers at least k packets by time $t + (n-1)B$.*

At times it will be useful to geometrically group packets into classes in the following manner. A packet belongs to **class** j if the length of the path on which it is routed by the optimal algorithm is in the range $[2^j, 2^{j+1})$. In the case when paths are unique or specified the class of a packet can be determined exactly.