

Otfried Cheong
Kyung-Yong Chwa
Kunsoo Park (Eds.)

LNCS 6507

Algorithms and Computation

21st International Symposium, ISAAC 2010
Jeju Island, Korea, December 2010
Proceedings, Part II

2
Part II



Springer

Commenced Publication in 1973

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison

Lancaster University, UK

Takeo Kanade

Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler

University of Surrey, Guildford, UK

Jon M. Kleinberg

Cornell University, Ithaca, NY, USA

Alfred Kobsa

University of California, Irvine, CA, USA

Friedemann Mattern

ETH Zurich, Switzerland

John C. Mitchell

Stanford University, CA, USA

Moni Naor

Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz

University of Bern, Switzerland

C. Pandu Rangan

Indian Institute of Technology, Madras, India

Bernhard Steffen

TU Dortmund University, Germany

Madhu Sudan

Microsoft Research, Cambridge, MA, USA

Demetri Terzopoulos

University of California, Los Angeles, CA, USA

Doug Tygar

University of California, Berkeley, CA, USA

Gerhard Weikum

Max Planck Institute for Informatics, Saarbruecken, Germany

Otfried Cheong
Kyung-Yong Chwa
Kunsoo Park (Eds.)

Algorithms and Computation

21st International Symposium, ISAAC 2010
Jeju Island, Korea, December 15-17, 2010
Proceedings, Part II

Volume Editors

Otfried Cheong
KAIST
Department of Computer Science
Daejeon 305-701, Korea
E-mail: otfried@kaist.edu

Kyung-Yong Chwa
KAIST
Department of Computer Science
Daejeon 305-701, Korea
E-mail: kychwa@jupiter.kaist.ac.kr

Kunsoo Park
Seoul National University
School of Computer Science and Engineering
Seoul 151-742, Korea
E-mail: kpark@snu.ac.kr

Library of Congress Control Number: 2010939852

CR Subject Classification (1998): F.2, I.3.5, E.1, C.2, G.2, F.1

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

ISSN	0302-9743
ISBN-10	3-642-17513-9 Springer Berlin Heidelberg New York
ISBN-13	978-3-642-17513-8 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

springer.com

© Springer-Verlag Berlin Heidelberg 2010
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180

Preface

This volume contains the proceedings of the 21st Annual International Symposium on Algorithms and Computations (ISAAC 2010), held in Jeju, Korea during December 15–17, 2010. Past editions have been held in Tokyo, Taipei, Nagoya, Hong Kong, Beijing, Cairns, Osaka, Singapore, Taejon, Chennai, Taipei, Christchurch, Vancouver, Kyoto, Hong Kong, Hainan, Kolkata, Sendai, Gold Coast, and Hawaii over the years 1990–2009.

ISAAC is an annual international symposium that covers the very wide range of topics in algorithms and computation. The main purpose of the symposium is to provide a forum for researchers working in algorithms and the theory of computation where they can exchange ideas in this active research community.

In response to the call for papers, ISAAC 2010 received 182 papers. Each submission was reviewed by at least three Program Committee members with the assistance of external referees. Since there were many high-quality papers, the Program Committee’s task was extremely difficult. Through an extensive discussion, the Program Committee accepted 77 of the submissions to be presented at the conference. Two special issues, one of *Algorithmica* and one of the *International Journal of Computational Geometry and Applications*, were prepared with selected papers from ISAAC 2010.

The best paper award was given to “From Holant to #CSP and Back: Dichotomy for Holant^c Problems” by Jin-Yi Cai, Sangxia Huang and Pinyan Lu, and the best student paper award to “Satisfiability with Index Dependency” by Hongyu Liang and Jing He. Two eminent invited speakers, David Eppstein from University of California, Irvine, and Matt Franklin from University of California, Davis, also contributed to this volume.

We would like to thank all Program Committee members and external referees for their excellent work, especially given the demanding time constraints; they gave the conference its distinctive character. We thank all who submitted papers for consideration; they all contributed to the high quality of the conference. We also thank the Organizing Committee members for their dedicated contribution that made the conference possible and enjoyable. Finally, we thank our sponsor SIGTCS (Special Interest Group on the Theoretical Computer Science) of KIISE (The Korean Institute of Information Scientists and Engineers) for the assistance and support.

December 2010

Otfried Cheong
Kyung-Yong Chwa
Kunsoo Park

Organization

Program Chairs

Otfried Cheong	KAIST, Korea
Kyung-Yong Chwa	KAIST, Korea
Kunsoo Park	Seoul National University, Korea

Program Committee

Lars Arge	University of Aarhus, Denmark
Takao Asano	Chuo University, Japan
Danny Chen	University of Notre Dame, USA
Rudolf Fleischer	Fudan University, China
Satoshi Fujita	Hiroshima University, Japan
Mordecai Golin	Hong Kong UST, Hong Kong
Seok-Hee Hong	University of Sydney, Australia
Oscar Ibarra	University of California - Santa Barbara, USA
Giuseppe Italiano	University of Rome “Tor Vergata”, Italy
Tao Jiang	UC Riverside, USA
Mihyun Kang	Humboldt University Berlin, Germany
Ming-Yang Kao	Northwestern University, USA
Tak-wah Lam	University of Hong Kong, Hong Kong
Gad Landau	University of Haifa, Israel
Peter Bro Miltersen	Aarhus University, Denmark
David Mount	University of Maryland, USA
Ian Munro	University of Waterloo, Canada
Yoshio Okamoto	Tokyo Institute of Technology, Japan
Frank Ruskey	University of Victoria, Canada
Kunihiko Sadakane	National Institute of Informatics, Japan
Steven Skiena	Stony Brook University, USA
Takeshi Tokuyama	Tohoku University, Japan
Ryuhei Uehara	Japan Advanced Institute of Science and Technology, Japan
Peter Widmayer	ETH Zurich, Switzerland
Chee Yap	Courant, NYU, USA
Hsu-Chun Yen	National Taiwan University, Taiwan
Afra Zomorodian	Dartmouth College, USA

Host Institute

KAIST (Korea Advanced Institute of Science and Technology)

Organizing Committee

Joon-Soo Choi	Kookmin University, Korea
Kyung-Yong Chwa	KAIST, Korea
Seung Bum Jo	KAIST, Korea
Hyunseob Lee	KAIST, Korea
Jung-Heum Park	The Catholic University of Korea, Korea

Referees

Mohammad Abam	Arash Farzan	Jiongxin Jin
Peyman Afshani	Sandor Fekete	Shinhaeng Jo
Hee-Kap Ahn	Andreas Feldmann	Daniel Johannsen
Laila El Aïmani	Holger Flier	Naoyuki Kamiyama
Toru Araki	Mathew Francis	Tom Kamphans
Abdullah Arslan	Robert Fraser	Iyad Kanj
Ilia Averbouch	Sorelle Friedler	Bruce Kapron
Laszlo Babai	Hiroshi Fujiwara	Akinori Kawachi
Christian Bachmaier	Oliver Gableske	Daniel Keren
Jeremy Barbay	Joachim von zur Gathen	Shuji Kijima
Amir Barghi	Petr Golovach	Hyo-Sil Kim
Peter van Beek	Martin Golumbic	Jae-Hoon Kim
Renévan Bevern	Joachim Gudmundsson	Masashi Kiyomi
Binay Bhattacharya	Prosenjit Gupta	Jan Willem Klop
Danny Breslauer	Gregory Gutin	Koji Kobayashi
Gerth Stølting Brodal	Carsten Gutwenger	Darek Kowalski
Joshua Brody	Michel Habib	Kasper Dalgaard Larsen
Jonathan Buss	Kristoffer Arnsfelt	Hyunseob Lee
Ho-Leung Chan	Hansen	Lap-Kei Lee
Kun-Mao Chao	Sariel Har-Peled	Mira Lee
Ho-Lin Chen	Masud Hasan	Taehyung Lee
Ming-Yang Chen	Mohammad Khairul	Avivit Levy
Minkyung Cho	Hasan	Chun-Cheng Lin
Marek Chrobak	Meng He	Maarten Loeffler
David Cohen	Pinar Heggernes	Daniel Lokshтанov
Amin Coja-Oghlan	Danny Hermelin	Alejandro Lopez-Ortiz
Maxime Crochemore	Tomio Hirata	Vadim Lozin
Pooya Davoodi	Christian Hoffmann	Jun Luo
Yann Disser	Ming-Deh Huang	Thomas Mølhave
Reza Dorrigiv	John Iacono	Khalegh Mamakani
David Doty	Keiko Imai	Maurice Margenstern
Scot Drysdale	Toshimasa Ishii	Dimitri Marinakis
Vida Dujmovic	Takehiro Ito	Tomomi Matsui
Robert Elsaesser	Kazuo Iwama	Yusuke Matsumoto
Guy Even	Li Jian	Yuichiro Miyamoto

Matthias Mnich	Bala Ravikumar	Rahul Tripathi
Morteza Monemizadeh	Iris Reinbacher	Shi-Chun Tsai
Petra Mutzel	Daniel Roche	Kostas Tsakalidas
Hiroshi Nagamochi	Juanjo Rué	Takeaki Uno
Shin-Ichi Nakano	Daniel Russel	Yushi Uno
C. Thach Nguyen	Toshiki Saitoh	Leslie Valiant
Patrick Nicholson	Jagan Sankaranarayanan	Antoine Vigneron
Takao Nishizeki	Ignasi Sau	Bow-Yaw Wang
Martin Noellenburg	Saket Saurabh	Haitao Wang
Katsuyuki Okeya	Joe Sawada	Osamu Watanabe
Hirotsuka Ono	Dominik Scheder	Oren Weimann
Yota Otachi	Akiyoshi Shioura	Joel Wein
Konstantinos	Michiel Smid	Renato Werneck
Panagiotou	Aaron Sterling	Aaron Williams
Daniel Panario	Mizuyo Takamatsu	Peter Winkler
Eunhui Park	Kenjiro Takazawa	Alexander Wolff
Jeong-Hyeon Park	Nicholas Tran	Prudence Wong
Jung-Heum Park	Xuehou Tan	Prudence W.H. Wong
Anders Sune Pedersen	Siamak Tazari	Mutsunori Yagiura
Benny Pinkas	My Thai	Katsuhisa Yamanaka
Greg Plaxton	Hing-Fung Ting	Koichi Yamazaki
Sheung-Hung Poon	Alexander Tiskin	Deshi Ye
Sanguthevar	Etsuji Tomita	Xiao Zhou
Rajasekaran	Lorenzo Traldi	Binhai Zhu
Jörg Rambau	Amitabh Trehan	Anna Zych

Table of Contents – Part II

Session 6A. Data Structure and Algorithm II

D ² -Tree: A New Overlay with Deterministic Bounds	1
<i>Gerth Stølting Brodal, Spyros Sioutas, Kostas Tsichlas, and Christos Zaroliagis</i>	
Efficient Indexes for the Positional Pattern Matching Problem and Two Related Problems over Small Alphabets	13
<i>Chih-Chiang Yu, Bing-Feng Wang, and Chung-Chin Kuo</i>	
Dynamic Range Reporting in External Memory	25
<i>Yakov Nekrich</i>	
A Cache-Oblivious Implicit Dictionary with the Working Set Property	37
<i>Gerth Stølting Brodal, Casper Kejlberg-Rasmussen, and Jakob Truelsen</i>	

Session 6B. Graph Algorithm II

The (p, q) -total Labeling Problem for Trees	49
<i>Toru Hasunuma, Toshimasa Ishii, Hirotaka Ono, and Yushi Uno</i>	
Drawing a Tree as a Minimum Spanning Tree Approximation	61
<i>Emilio Di Giacomo, Walter Didimo, Giuseppe Liotta, and Henk Meijer</i>	
k -Cyclic Orientations of Graphs	73
<i>Yasuaki Kobayashi, Yuichiro Miyamoto, and Hisao Tamaki</i>	
Improved Bounds on the Planar Branchwidth with Respect to the Largest Grid Minor Size	85
<i>Qian-Ping Gu and Hisao Tamaki</i>	

Session 7A. Computational Geometry II

Maximum Overlap of Convex Polytopes under Translation	97
<i>Hee-Kap Ahn, Siu-Wing Cheng, and Iris Reinbacher</i>	
Approximate Shortest Homotopic Paths in Weighted Regions	109
<i>Siu-Wing Cheng, Jiongxin Jin, Antoine Vigneron, and Yajun Wang</i>	

Spanning Ratio and Maximum Detour of Rectilinear Paths in the L_1 Plane.....	121
<i>Ansgar Grüne, Tien-Ching Lin, Teng-Kai Yu, Rolf Klein, Elmar Langetepe, D.T. Lee, and Sheung-Hung Poon</i>	

Session 7B. Graph Coloring II

Approximation and Hardness Results for the Maximum Edge q -Coloring Problem	132
<i>Anna Adamaszek and Alexandru Popa</i>	
3-Colouring AT-Free Graphs in Polynomial Time	144
<i>Juraj Stacho</i>	
On Coloring Graphs without Induced Forests	156
<i>Hajo Broersma, Petr A. Golovach, Daniël Paulusma, and Jian Song</i>	

Session 8A. Approximation Algorithm II

On the Approximability of the Maximum Interval Constrained Coloring Problem	168
<i>Stefan Canzar, Khaled Elbassioni, Amr Elmasry, and Rajiv Raman</i>	
Approximability of Constrained LCS	180
<i>Minghui Jiang</i>	
Approximation Algorithms for the Multi-Vehicle Scheduling Problem ...	192
<i>Binay Bhattacharya and Yuzhuang Hu</i>	
On Greedy Algorithms for Decision Trees	206
<i>Ferdinando Cicalese, Tobias Jacobs, Eduardo Laber, and Marco Molinaro</i>	

Session 8B. Online Algorithm

Single and Multiple Device DSA Problem, Complexities and Online Algorithms	218
<i>Weiwei Wu, Wanyong Tian, Minming Li, Chun Jason Xue, and Enhong Chen</i>	
The Onion Diagram: A Voronoi-Like Tessellation of a Planar Line Space and Its Applications (Extended Abstract)	230
<i>Sang Won Bae and Chan-Su Shin</i>	
Improved Online Algorithms for 1-Space Bounded 2-Dimensional Bin Packing.....	242
<i>Yong Zhang, Jingchi Chen, Francis Y.L. Chin, Xin Han, Hing-Fung Ting, and Yung H. Tsin</i>	

On the Continuous CNN Problem	254
<i>John Augustine and Nick Gravin</i>	

Session 9A. Scheduling

Policies for Periodic Packet Routing	266
<i>Britta Peis, Sebastian Stiller, and Andreas Wiese</i>	
Increasing Speed Scheduling and Flow Scheduling	279
<i>Sebastian Stiller and Andreas Wiese</i>	
A Tighter Analysis of Work Stealing	291
<i>Marc Tchiboukdjian, Nicolas Gast, Denis Trystram, Jean-Louis Roch, and Julien Bernard</i>	
Approximating the Traveling Tournament Problem with Maximum Tour Length 2	303
<i>Clemens Thielen and Stephan Westphal</i>	

Session 9B. Data Structure and Algorithm III

Alphabet Partitioning for Compressed Rank/Select and Applications . .	315
<i>Jérémy Barbay, Travis Gagie, Gonzalo Navarro, and Yakov Nekrich</i>	
Entropy-Bounded Representation of Point Grids	327
<i>Arash Farzan, Travis Gagie, and Gonzalo Navarro</i>	
Identifying Approximate Palindromes in Run-Length Encoded Strings	339
<i>Kuan-Yu Chen, Ping-Hui Hsu, and Kun-Mao Chao</i>	

Session 10A. Graph Algorithm III

Minimum Cost Partitions of Trees with Supply and Demand	351
<i>Takehiro Ito, Takuya Hara, Xiao Zhou, and Takao Nishizeki</i>	
Computing the (t, k) -Diagnosability of Component-Composition Graphs and Its Application	363
<i>Sun-Yuan Hsieh and Chun-An Chen</i>	
Why Depth-First Search Efficiently Identifies Two and Three-Connected Graphs	375
<i>Amr Elmasry</i>	
Beyond Good Shapes: Diffusion-Based Graph Partitioning is Relaxed Cut Optimization	387
<i>Henning Meyerhenke</i>	

Induced Subgraph Isomorphism on Interval and Proper Interval Graphs	399
<i>Pinar Heggernes, Daniel Meister, and Yngve Villanger</i>	

Session 10B. Computational Geometry III

Testing Simultaneous Planarity When the Common Graph Is 2-Connected.....	410
<i>Bernhard Haeupler, Krishnam Raju Jampani, and Anna Lubiw</i>	
Computing the Discrete Fréchet Distance with Imprecise Input	422
<i>Hee-Kap Ahn, Christian Knauer, Marc Scherfenberg, Lena Schlipf, and Antoine Vigneron</i>	
Connectivity Graphs of Uncertainty Regions.....	434
<i>Erin Chambers, Alejandro Erickson, Sándor Fekete, Jonathan Lenchner, Jeff Sember, Srinivasan Venkatesh, Ulrike Stege, Svetlana Stolpner, Christophe Weibel, and Sue Whitesides</i>	
$\pi/2$ -Angle Yao Graphs Are Spanners	446
<i>Prosenjit Bose, Mirela Damian, Karim Douïeb, Joseph O'Rourke, Ben Seamone, Michiel Smid, and Stefanie Wührer</i>	
Identifying Shapes Using Self-assembly (Extended Abstract).....	458
<i>Matthew J. Patitz and Scott M. Summers</i>	
Author Index	471

Table of Contents – Part I

Invited Talks

Regular Labelings and Geometric Structures (Abstract).....	1
<i>David Eppstein</i>	
Algorithmic Aspects of Secure Computation and Communication (Abstract)	2
<i>Matt Franklin</i>	

Session 1A. Approximation Algorithm I

Faster Algorithms for Feedback Arc Set Tournament, Kemeny Rank Aggregation and Betweenness Tournament	3
<i>Marek Karpinski and Warren Schudy</i>	
A 3/2-Approximation Algorithm for Generalized Steiner Trees in Complete Graphs with Edge Lengths 1 and 2	15
<i>Piotr Berman, Marek Karpinski, and Alexander Zelikovsky</i>	
Approximate Periodicity	25
<i>Amihoud Amir, Estrella Eisenberg, and Avivit Levy</i>	
Approximating the Average Stretch Factor of Geometric Graphs	37
<i>Siu-Wing Cheng, Christian Knauer, Stefan Langerman, and Michiel Smid</i>	

Session 1B. Complexity I

Satisfiability with Index Dependency	49
<i>Hongyu Liang and Jing He</i>	
Anonymous Fuzzy Identity-Based Encryption for Similarity Search	61
<i>David W. Cheung, Nikos Mamoulis, W.K. Wong, S.M. Yiu, and Ye Zhang</i>	
Improved Randomized Algorithms for 3-SAT	73
<i>Kazuo Iwama, Kazuhisa Seto, Tadashi Takai, and Suguru Tamaki</i>	
Quantum Counterfeit Coin Problems	85
<i>Kazuo Iwama, Harumichi Nishimura, Rudy Raymond, and Junichi Teruyama</i>	

Session 2A. Data Structure and Algorithm I

Priority Range Trees	97
<i>Michael T. Goodrich and Darren Strash</i>	
Should Static Search Trees Ever Be Unbalanced?.....	109
<i>Prosenjit Bose and Karim Douieb</i>	
Levelwise Mesh Sparsification for Shortest Path Queries	121
<i>Yuichiro Miyamoto, Takeaki Uno, and Mikio Kubo</i>	
Unit-Time Predecessor Queries on Massive Data Sets	133
<i>Andrej Brodnik and John Iacono</i>	

Session 2B. Combinatorial Optimization

Popularity at Minimum Cost	145
<i>Telikepalli Kavitha, Meghana Nasre, and Prajakta Nimbhorkar</i>	
Structural and Complexity Aspects of Line Systems of Graphs	157
<i>Jozef Jirásek and Pavel Klavík</i>	
Neighbor Systems, Jump Systems, and Bisubmodular Polyhedra	169
<i>Akiyoshi Shioura</i>	
Generating Trees on Multisets	182
<i>Bingbing Zhuang and Hiroshi Nagamochi</i>	

Session 3A. Graph Algorithm I

Seidel Minor, Permutation Graphs and Combinatorial Properties.....	194
<i>Vincent Limouzy</i>	
Simultaneous Interval Graphs	206
<i>Krishnam Raju Jampani and Anna Lubiw</i>	
Unbalanced Graph Partitioning	218
<i>Angsheng Li and Peng Zhang</i>	
On the Intersection of Tolerance and Cocomparability Graphs	230
<i>George B. Mertzios and Shmuel Zaks</i>	
Flows in One-Crossing-Minor-Free Graphs	241
<i>Erin Chambers and David Eppstein</i>	

Session 3B. Complexity II

From Holant to #CSP and Back: Dichotomy for Holant ^c Problems	253
<i>Jin-Yi Cai, Sangxia Huang, and Pinyan Lu</i>	

Computing Sparse Multiples of Polynomials	266
<i>Mark Giesbrecht, Daniel S. Roche, and Hrushikesh Tilak</i>	
Fractal Parallelism: Solving SAT in Bounded Space and Time	279
<i>Denys Duchier, Jérôme Durand-Lose, and Maxime Senot</i>	
Interpretation of Stream Programs: Characterizing Type 2 Polynomial Time Complexity	291
<i>Hugo Férée, Emmanuel Hainry, Mathieu Hoyrup, and Romain Péchoux</i>	
New Upper Bounds on the Average PTF Density of Boolean Functions	304
<i>Kazuyuki Amano</i>	

Session 4A. Computational Geometry I

An Optimal Algorithm for Computing Angle-Constrained Spanners	316
<i>Paz Carmi and Michiel Smid</i>	
Approximating Minimum Bending Energy Path in a Simple Corridor . . .	328
<i>Jinhui Xu, Lei Xu, and Yulai Xie</i>	

Session 4B. Graph Coloring I

Analysis of an Iterated Local Search Algorithm for Vertex Coloring	340
<i>Dirk Sudholt and Christine Zarges</i>	
Bounded Max-colorings of Graphs	353
<i>Evrripidis Bampis, Alexander Kononov, Giorgio Lucarelli, and Ioannis Milis</i>	

Session 5A. Fixed Parameter Tractability

Parameterized Algorithms for Boxicity	366
<i>Abhijin Adiga, Rajesh Chitnis, and Saket Saurabh</i>	
On Tractable Cases of Target Set Selection	378
<i>André Nichterlein, Rolf Niedermeier, Johannes Uhlmann, and Mathias Weller</i>	
Combining Two Worlds: Parameterised Approximation for Vertex Cover	390
<i>Ljiljana Brankovic and Henning Fernau</i>	
Listing All Maximal Cliques in Sparse Graphs in Near-Optimal Time . . .	403
<i>David Eppstein, Maarten Löffler, and Darren Strash</i>	

Session 5B. Optimization

Lower Bounds for Howard’s Algorithm for Finding Minimum Mean-Cost Cycles	415
<i>Thomas Dueholm Hansen and Uri Zwick</i>	
Solving Two-Stage Stochastic Steiner Tree Problems by Two-Stage Branch-and-Cut	427
<i>Immanuel Bomze, Markus Chimani, Michael Jünger, Ivana Ljubić, Petra Mutzel, and Bernd Zey</i>	
An Optimal Algorithm for Single Maximum Coverage Location on Trees and Related Problems	440
<i>Joachim Spoerhase</i>	
A Faster Algorithm for the Maximum Even Factor Problem	451
<i>Maxim A. Babenko</i>	
Author Index	463

D²-Tree: A New Overlay with Deterministic Bounds

Gerth Stølting Brodal¹, Spyros Sioutas²,
Kostas Tsihclas³, and Christos Zaroliagis⁴

¹ MADALGO (Center for Massive Data Algorithmics, a Center of the Danish National Research Foundation), Aarhus University

`gerth@madalgo.au.dk`

² Ionian University, Department of Informatics

`sioutas@ionio.gr`

³ Aristotle University of Thessaloniki, Department of Informatics

`tsichlas@csd.auth.gr`

⁴ CTI and Dept. of Computer Engineering & Informatics, University of Patras

`zaro@ceid.upatras.gr`

Abstract. We present a new overlay, called the *Deterministic Decentralized tree* (D^2 -tree). The D^2 -tree compares favourably to other overlays for the following reasons: (a) it provides matching and better complexities, which are deterministic for the supported operations; (b) the management of nodes (peers) and elements are completely decoupled from each other; and (c) an efficient deterministic load-balancing mechanism is presented for the uniform distribution of elements into nodes, while at the same time probabilistic optimal bounds are provided for the congestion of operations at the nodes.

1 Introduction

Decentralized systems and in particular Peer-to-Peer (P2P) networks have become very popular of late and are widely used for sharing resources and store very large data sets. Data are stored at the nodes (or peers) and the most crucial operations are data search (identify the node that stores the requested information) and updates (insertions/deletions of data). Searching and updating is typically done by building a logical *overlay network* that facilitates the assignment and indexing of data at the nodes. Sometimes, we distinguish between the overlay structure per se and the indexing scheme used to access the data.

Following the typical modeling, a decentralized communication network is represented by a graph. Its nodes correspond to the network nodes, while its edges correspond to communication links. We assume constant size messages between nodes through links and asynchronous communication. It is assumed that the network provides an upper bound on the time needed for a node to send a message and receive an acknowledgment. The complexity of an operation is measured in terms of the number of messages issued during its execution. Throughout the paper, when we refer to cost we shall mean number of messages (internal computations at nodes are considered insignificant). The *overlay* is

another graph defined over the communication network. The nodes of the overlay correspond to nodes of the original network, while its edges (links) may not correspond to existing communication links, but to communication paths.

With respect to its *structure*, the overlay supports the operations *Join* (of a new node v ; v communicates with an existing node u in order to be inserted into the overlay), and *Departure* (of an existing node u ; u leaves the overlay announcing its intent to other nodes of the overlay). The overlay is used to implement an *indexing scheme* for the stored data. Such a scheme supports the operations *search* for an element, *insert* a new element, *delete* an existing element, and *range query* for elements in a specific range.

In terms of efficiency, an overlay network should address the following issues:

- *Fast queries and updates*: updates and queries must be executed in a minimal number of communication rounds and using a minimal number of messages.
- *Ordered data*: keeping the data in order facilitates the implementation of various enumeration queries when compared to a simple dictionary that can only answer membership queries, including those arising in DNA databases, location-based services, and prefix searches for file names or data titles. Indeed, the ever-wider use of P2P infrastructures has found applications that require support for range queries (e.g., [6]).
- *Size of nodes (peers)*: the size of a node is the routing information (links and related data) maintained by this node and it is not related to the number of data elements stored in it. Keeping the size of a node small allows for more efficient update operations, but in general reduces the efficiency of access operations while aggravating fault tolerance.
- *Fault Tolerance*: the structure should be able to discover and heal failures at nodes or links.
- *Congestion*: it refers to the distribution of the load of search (access) operations per node, aiming at distributing this load equally across all nodes. The congestion is an *expected* quantity defined as the maximum, among all nodes, of the fraction of the expected number of accesses of a node due to a random sequence of operations on the structure.
- *Load Balancing*: it refers to the distribution of data elements on the nodes. The goal of load balancing is to distribute equally the n elements stored in the N nodes of the network (typically $N \ll n$). That is, ideally each node should carry approximately k elements, where $\lfloor n/N \rfloor \leq k \leq \lceil n/N \rceil + 1$.

There has been considerable recent work in devising effective distributed search and update techniques. Existing structured P2P systems can be classified into two broad categories: distributed hash table (DHT)-based systems and tree-based systems. Examples of the former, which constitute the majority, include Chord [11], Pastry [14], Symphony [12], and Tapestry [17]. DHT-based systems support exact match queries well and use (successfully) probabilistic methods to distribute the workload among nodes equally. DHT-based systems work with little synchrony and high *churn* (the collective effect created by independent burstly arrivals and departures of nodes), a fundamental characteristic of the

Internet. Since hashing destroys the ordering on keys, DHT-based systems typically do not possess the functionality to support straightforwardly range queries, or more complex queries based on data ordering (e.g., nearest-neighbor and string prefix queries). The most recent effort towards range queries is reported in [16].

Tree-based systems are based on hierarchical structures. They support range queries more naturally and efficiently as well as a wider range of operations, since they maintain the ordering of data. On the other hand, they lack the simplicity of DHT-based systems, and they do not always guarantee data locality and load balancing in the whole system. Important examples of such systems include Skip Graphs (SG) [47], NoN SG [13], SkipNet (SN), Deterministic SN [9], Bucket SG [3], Family Trees [15], Skip Webs [1], BATON [10], Rainbow Skip Graphs (RSG) [8], and Strong RSG [8].

In this work, we focus on tree-based overlay networks that support directly range and more complex queries. Let N be the number of nodes present in the network and let n denote the size of data ($N \ll n$). Let M be the size of each node, $Q(n, N)$ be the cost of a single query, $U(n, N)$ be the cost of an update, $C(n, N)$ be the congestion per node (measuring the load) incurred by search operations, and let $L(n, N)$ be the cost for load balancing the overlay w.r.t. element updates. With respect to congestion, each node issues one operation, while the destination node of the operation is assumed to be selected uniformly at random among all nodes of the network. Congestion depends on the distribution of elements into nodes as well as on the topology of the overlay. It provides hints as to how well the structure avoids the existence of *hotspots* (i.e., nodes which are accessed multiple times during a sequence of operations – the root of a tree is usually a hotspot in decentralized tree structures).

A comparison of the aforementioned tree-based overlays is given in Table 1. We would like to emphasize that w.r.t. load balancing, there are solutions in the literature either as part of the overlay (e.g., [10]) or as a separate technique (e.g. [37]). These solutions are either heuristics, or provide expected bounds under

Table 1. A comparison between previous methods and the D^2 -tree. By \hat{O} we represent expected bounds, by \bar{O} we represent amortized bounds, and by \overline{O} expected amortized bounds. All other bounds are worst-case. Typically, $N \ll n$.

Methods	N	M	$Q(n, N)$	$U(n, N)$	$C(n, N)$	$L(n, N)$
SG [47]	$\leq n$	$O(\log N)$	$\hat{O}(\log N)$ w.h.p.	$\hat{O}(\log N)$ w.h.p.	$\hat{O}(\frac{\log N}{N})$	$\bar{O}(\log N)$
NoN SG [13]	n	$O(\log^2 n)$	$\hat{O}(\frac{\log n}{\log \log n})$	$\hat{O}(\log^2 n)$	$\hat{O}(\frac{\log^2 n}{n})$	–
Determ. SN [9]	n	$O(\log n)$	$O(\log n)$	$O(\log^2 n)$	$O(\frac{n^{0.32}}{n})$	–
BATON [10]	$\leq n$	$O(\log N)$	$O(\log N)$	$O(\log N)$	–	$\overline{O}(\log n)$
Family Trees [15]	n	$O(1)$	$\hat{O}(\log n)$	$\hat{O}(\log n)$	$\hat{O}(\frac{\log n}{n})$	–
Bucket SG [3]	$\leq n$	$O(\frac{n}{N} + \log N)$	$\hat{O}(\log N)$	$\hat{O}(\log N)$	$\hat{O}(\frac{1}{N} + \frac{\log N}{n})$	No Bounds
Skip Webs [1]	n	$O(\log n)$	$\hat{O}(\frac{\log n}{\log \log n})$	$\hat{O}(\frac{\log n}{\log \log n})$	$\hat{O}(\frac{\log n}{n})$	–
Rainbow SG [8]	n	$O(1)$	$\hat{O}(\log n)$ w.h.p.	$\bar{O}(\log n)$ w.h.p.	$\hat{O}(\frac{\log n}{n})$	–
Strong RSG [8]	n	$O(1)$	$O(\log n)$	$\bar{O}(\log n)$	$\bar{O}(\frac{n^c}{n})$	–
D²-tree	$\leq n$	$O(1)$	$O(\log N)$	$\bar{O}(\log N)$	$\hat{O}(\frac{\log N}{N})$	$\bar{O}(\log N)$

certain assumptions, or amortized bounds but at the expense of increasing the size per node (see [5] for a detailed discussion).

Our Contribution. In this paper we present a new tree-based overlay, called the *Deterministic Decentralized tree* or D^2 -tree. The D^2 -tree (see also Table 1) uses $O(1)$ space per node, achieves a deterministic $O(\log N)$ query bound and a deterministic (amortized) $O(\log N)$ update bound for elements as well as for node joins and departures, achieves *optimal* congestion, and exhibits a deterministic (amortized) $O(\log N)$ bound for load-balancing. Moreover, it supports ordered data queries optimally, and tolerates node failures.

The D^2 -tree is an overlay consisting of two levels. The upper level is a perfect binary tree, while the lower level consists of buckets (sets of nodes), where each bucket is structured as a doubly linked list. Each bucket contains $O(\log N)$ nodes. Since N changes, the size of buckets is dynamically maintained by the overlay.

In the D^2 -tree, we separate the index from the overlay structure using the load-balancing mechanism. The number of elements per node is dynamic w.r.t. node joins and departures and it is controlled by the load-balancing mechanism. Moreover, the number of nodes of the perfect binary tree is not connected by any means to the number of elements stored in the structure. The overlay structure supports the operations of node join and node departure, while at the same time it tackles failures of nodes whenever these are discovered.

Our load-balancing technique distributes almost equally the elements among nodes by making use of weights. Weights are used to define a metric of load-balance, which shows how uneven is the load between nodes. When the load is uneven, then a data migration process is initiated to equally distribute elements.

Our load-balancing technique is quite general and can be applied to any hierarchical decentralized overlay (e.g., BATON, Skip Graphs) with the following specifications: (i) The overlay structure must be a tree with height $O(\log N)$ with each node having $O(1)$ children. (ii) Nodes at level i having the same father have approximately (within constant factors) the same weight, which is $\Omega(i^4)$. (iii) Updates are performed at the leaves. Alternatively, if each node has access to a leaf in $O(1)$ messages then this is enough, since the update is simply forwarded to this leaf.

We discuss the load balancing technique in Section 2, and present the D^2 -tree in Section 3. We conclude in Section 4. Due to space constraints, some details and proofs are deferred to the full version [5].

2 Deterministic Load Balancing

The load-balancing mechanism distributes almost equally the elements among nodes by making use of weights, which are used to define a metric showing how uneven is the load between nodes. When the load is uneven, then a data migration process is initiated to equally distribute elements.

A few definitions are in place. Assume that the overlay structure is a tree \mathcal{T} . Based on \mathcal{T} ancestor-descendant relationships are defined. There is a node that has no ancestor (the *root*) and there are nodes with no descendants (the *leaves*).