

# CONTEMPORARY MATHEMATICS

588

## Graph Partitioning and Graph Clustering

10th DIMACS Implementation Challenge Workshop  
February 13–14, 2012  
Georgia Institute of Technology  
Atlanta, GA

David A. Bader  
Henning Meyerhenke  
Peter Sanders  
Dorothea Wagner  
Editors



American Mathematical Society  
Center for Discrete Mathematics  
and Theoretical Computer Science



# Graph Partitioning and Graph Clustering



# CONTEMPORARY MATHEMATICS

---

588

## Graph Partitioning and Graph Clustering

10th DIMACS Implementation Challenge Workshop  
February 13–14, 2012  
Georgia Institute of Technology  
Atlanta, GA

David A. Bader  
Henning Meyerhenke  
Peter Sanders  
Dorothea Wagner  
Editors



American Mathematical Society  
Center for Discrete Mathematics  
and Theoretical Computer Science



---

**American Mathematical Society**  
Providence, Rhode Island

## EDITORIAL COMMITTEE

Dennis DeTurck, Managing Editor

Michael Loss      Kailash Misra      Martin J. Strauss

2010 *Mathematics Subject Classification*. Primary 05C85, 68W05, 05C82, 68W10, 68R05, 68R10, 05C05, 05C65.

---

### Library of Congress Cataloging-in-Publication Data

Cataloging-in-Publication Data has been applied for by the AMS. See  
<http://www.loc.gov/publish/cip/>.

Contemporary Mathematics ISSN: 0271-4132 (print); ISSN: 1098-3627 (online)

---

**Copying and reprinting.** Material in this book may be reproduced by any means for educational and scientific purposes without fee or permission with the exception of reproduction by services that collect fees for delivery of documents and provided that the customary acknowledgment of the source is given. This consent does not extend to other kinds of copying for general distribution, for advertising or promotional purposes, or for resale. Requests for permission for commercial use of material should be addressed to the Acquisitions Department, American Mathematical Society, 201 Charles Street, Providence, Rhode Island 02904-2294, USA. Requests can also be made by e-mail to [reprint-permission@ams.org](mailto:reprint-permission@ams.org).

Excluded from these provisions is material in articles for which the author holds copyright. In such cases, requests for permission to use or reprint should be addressed directly to the author(s). (Copyright ownership is indicated in the notice in the lower right-hand corner of the first page of each article.)

© 2013 by the American Mathematical Society. All rights reserved.

The American Mathematical Society retains all rights  
except those granted to the United States Government.

Copyright of individual articles may revert to the public domain 28 years  
after publication. Contact the AMS for copyright status of individual articles.

Printed in the United States of America.

∞ The paper used in this book is acid-free and falls within the guidelines  
established to ensure permanence and durability.

Visit the AMS home page at <http://www.ams.org/>

10 9 8 7 6 5 4 3 2 1      13 12 11 10 09 08

## Contents

Preface	
DAVID A. BADER, HENNING MEYERHENKE, PETER SANDERS, and DOROTHEA WAGNER	vii
High Quality Graph Partitioning	
PETER SANDERS and CHRISTIAN SCHULZ	1
Abusing a Hypergraph Partitioner for Unweighted Graph Partitioning	
B. O. FAGGINGER AUER and R. H. BISSELING	19
Parallel Partitioning with Zoltan: Is Hypergraph Partitioning Worth It?	
SIVASANKARAN RAJAMANICKAM and ERIK G. BOMAN	37
UMPa: A Multi-objective, multi-level partitioner for communication minimization	
ÜMIT V. ÇATALYÜREK, MEHMET DEVECI, KAMER KAYA, and BORA UÇAR	53
Shape Optimizing Load Balancing for MPI-Parallel Adaptive Numerical Simulations	
HENNING MEYERHENKE	67
Graph Partitioning for Scalable Distributed Graph Computations	
AYDIN BULUÇ and KAMESH MADDURI	83
Using Graph Partitioning for Efficient Network Modularity Optimization	
HRISTO DJIDJEV and MELIH ONUS	103
Modularity Maximization in Networks by Variable Neighborhood Search	
DANIEL ALOISE, GILLES CAPOROSSI, PIERRE HANSEN, LEO LIBERTI, SYLVAIN PERRON, and MANUEL RUIZ	113
Network Clustering via Clique Relaxations: A Community Based Approach	
ANURAG VERMA and SERGIY BUTENKO	129
Identifying Base Clusters and Their Application to Maximizing Modularity	
SRIRAM SRINIVASAN, TANMOY CHAKRABORTY, and SANJUKTA BHOWMICK	141
Complete Hierarchical Cut-Clustering: A Case Study on Expansion and Modularity	
MICHAEL HAMANN, TANJA HARTMANN, and DOROTHEA WAGNER	157

A Partitioning-Based Divisive Clustering Technique for Maximizing the Modularity	
ÜMIT V. ÇATALYÜREK, KAMER KAYA, JOHANNES LANGGUTH, and BORA UÇAR	171
An Ensemble Learning Strategy for Graph Clustering	
MICHAEL OVELGÖNNE and ANDREAS GEYER-SCHULZ	187
Parallel Community Detection for Massive Graphs	
E. JASON RIEDY, HENNING MEYERHENKE, DAVID EDIGER, and DAVID A. BADER	207
Graph Coarsening and Clustering on the GPU	
B. O. FAGGINGER AUER and R. H. BISSELING	223

## Preface

This collection is related to the Workshop of the 10th DIMACS Implementation Challenge, which took place in Atlanta, Georgia (USA) on February 13-14, 2012. The purpose of DIMACS Implementation Challenges<sup>1</sup> is to assess the practical performance of algorithms in a respective problem domain. These challenges are scientific competitions in areas of interest where worst case and probabilistic analysis yield unrealistic results. Where analysis fails, experimentation can provide insights into realistic algorithm performance and thereby help to bridge the gap between theory and practice. For this purpose common benchmark instances, mostly from real applications, are established. By evaluating different implementations on these instances, the challenges create a reproducible picture of the state of the art in the area under consideration. This helps to foster an effective technology transfer within the research areas of algorithms, data structures, and implementation techniques as well as a transfer back to the original applications.

The topics of the previous nine challenges are as follows (in chronological order): *Network Flows and Matching* (1990-91), *Maximum Clique, Graph Coloring and Satisfiability* (1992-93), *Parallel Algorithms for Combinatorial Problems* (1993-94), *Fragment Assembly and Genome Rearrangements* (1994-95), *Priority Queues, Dictionaries, and Multi-Dimensional Point Sets* (1995-96), *Near Neighbor Searches* (1998-99), *Semidefinite and Related Optimization Problems* (1999-2000), *The Traveling Salesman Problem* (2000-01), and *Shortest Path Problems* (2005-06).

### 1. Introducing the 10th Challenge – Graph Partitioning and Graph Clustering

The 10th challenge considered the two related problems of partitioning and clustering graphs. Both are ubiquitous subtasks in many application areas. Generally speaking, techniques for graph partitioning and graph clustering aim at the identification of vertex subsets with many internal and few external edges. To name only a few, problems addressed by graph partitioning and graph clustering algorithms are:

- What are the communities within an (online) social network?
- How do I speed up a numerical simulation by mapping it efficiently onto a parallel computer?
- How must components be organized on a computer chip such that they can communicate efficiently with each other?
- What are the segments of a digital image?
- Which functions are certain genes (most likely) responsible for?

---

<sup>1</sup><http://dimacs.rutgers.edu/Challenges/>



For a more detailed treatment of applications and solution techniques, the interested reader is referred to the surveys of Fortunato<sup>2</sup>, Schaeffer<sup>3</sup>, and Schloegel et al.<sup>4</sup> on the different topics.

Within the algorithms community, techniques for solving the problems above have been developed at least since the early 1970s—while some of the applications are newer. Improving known and developing new solution techniques are aspects of ongoing research.

The primary goal of this challenge was to create a reproducible picture of the state of the art in the area of graph partitioning and graph clustering algorithms. To this end, a standard set of benchmark instances was identified. Then participants were invited to submit solutions to different challenge problems. This way different algorithms and implementations were tested against the benchmark instances. Thereby future researchers are enabled to identify techniques that are most effective for a respective partitioning or clustering problem—by using our benchmark set and by comparing their results to the challenge results.

## 2. Key Results

The main results of the 10th DIMACS Implementation Challenge include:

- Extension of a file format used by several graph partitioning and graph clustering libraries for graphs, their geometry, and partitions. Formats are described on the challenge website.<sup>5</sup>
- Collection and online archival<sup>5</sup> of a common testbed of input instances and generators (including their description) from different categories for evaluating graph partitioning and graph clustering algorithms. For the actual challenge, a core subset of the testbed was chosen.
- Definition of a new combination of measures to assess the quality of a clustering.
- Definition of a measure to assess the work an implementation performs in a parallel setting. This measure is used to normalize sequential and parallel implementations to a common base line.
- Experimental evaluation of state-of-the-art implementations of graph partitioning and graph clustering codes on the core input families.
- A nondiscriminatory way to assign scores to solvers that takes both running time and solution quality into account.
- Discussion of directions for further research in the areas of graph partitioning and graph clustering.
- The paper *Benchmarks for Network Analysis*, which was invited as a contribution to the *Encyclopedia of Social Network Analysis and Mining*.

The primary location of information regarding the 10th DIMACS Implementation Challenge is the website <http://www.cc.gatech.edu/dimacs10/>.

---

<sup>2</sup>Santo Fortunato, Community detection in graphs, *Physics Reports* 486 (2010), no. 3–5, 75–174.

<sup>3</sup>Satu E. Schaeffer, Graph clustering, *Computer Science Review* 1 (2007), no. 1, 27–64.

<sup>4</sup>K. Schloegel, G. Karypis, and V. Kumar, Graph partitioning for high-performance scientific simulations, *Sourcebook of parallel computing* (Jack Dongarra, Ian Foster, Geoffrey Fox, William Gropp, Ken Kennedy, Linda Torczon, and Andy White, eds.) Morgan Kaufmann Publishers, 2003, pp. 491–541.

<sup>5</sup><http://www.cc.gatech.edu/dimacs10/downloads.shtml>

### 3. Challenge Description

**3.1. Data Sets.** The collection of benchmark inputs of the 10th DIMACS Implementation Challenge includes both synthetic and real-world data. All graphs are undirected. Formerly directed instances were symmetrized by making every directed edge undirected. While this procedure necessarily loses information in a number of real-world applications, it appeared to be necessary since most existing software libraries can handle undirected graphs only. Directed graphs (or unsymmetric matrices) are left for further work.

Synthetic graphs in the collection include random graphs (Erdős-Rényi, R-MAT, random geometric graphs using the unit disk model), Delaunay triangulations, and graphs that mimic meshes from dynamic numerical simulations. Real-world inputs consist of co-author and citation networks, road networks, numerical simulation meshes, web graphs, social networks, computational task graphs, and graphs from adapting voting districts (redistricting).

For the actual challenge two subsets were chosen, one for graph partitioning and one for graph clustering. The first one (for graph partitioning) contained 18 graphs, which had to be partitioned into 5 different numbers of parts each, yielding 90 problem instances. The second one (for graph clustering) contained 30 graphs. Due to the choice of objective functions for graph clustering, no restriction on the number of parts or their size was necessary in this category.

**3.2. Categories.** One of the main goals of the challenge was to compare different techniques and algorithmic approaches. Therefore participants were invited to join different challenge competitions aimed at assessing the performance and solution quality of different implementations. Let  $G = (V, E, \omega)$  be an undirected graph with edge weight function  $\omega$ .

**3.2.1. Graph Partitioning.** Here the task was to compute a partition  $\Pi$  of the vertex set  $V$  into  $k$  parts of size at most  $(1 + \epsilon) \lceil \frac{|V|}{k} \rceil$ . The two objective functions used to assess the partitioning quality were edge cut (EC, total number of edges with endpoints in different parts) and maximum communication volume (CV). CV sums for each part  $p$  and each vertex  $v$  therein the number of parts adjacent to  $v$  but different from  $p$ . The final result is the maximum over each part.

For each instance result (EC and CV results were counted as one instance *each*), the solvers with the first six ranks received a descending number of points (10, 6, 4, 3, 2, 1), a scoring system borrowed from former Formula 1 rules.

Three groups submitted solutions to the graph partitioning competition. Only one of the submitted solvers is a graph partitioner by nature, the other two are actually hypergraph partitioners. Both hypergraph partitioners use multilevel recursive bisection. While their quality, in particular for the communication volume, is generally not bad, the vast majority of best ranked solutions (139 out of 170) are held by the graph partitioner KAPA.

**3.2.2. Graph Clustering.** The clustering challenge was divided into two separate competitions with different optimization criteria. For the first competition the objective modularity had to be optimized. Modularity has been a very popular measure in the last years, in particular in the field of community detection. It follows the intra-cluster-density vs. inter-cluster-sparsity paradigm. However, some

criticism has emerged recently.<sup>6</sup> Also, solvers performing implicit optimization based on the intra-cluster-density vs. inter-cluster-sparsity paradigm were supposed to have a fair chance, too. That is why we developed a second competition with a mix of four other clustering objectives. The rationale was that the combination of these measures would lead to meaningful clusters and avoid pathological cases of single measures. The exact definition of the objective functions can be found at the challenge website.<sup>7</sup>

The modularity competition saw the largest number of entries, with 15 solvers from eight groups. Two solvers led the field, CGGCLRG and VNS. Of the two, CGGCLRG scored the most points and obtained the highest number of best ranked solutions. The four solvers entering the mix clustering competition were submitted by two groups (two each). Three solvers headed the top of the ranking, with a slight advantage for the two COMMUNITY-EL implementations.

**3.2.3. Pareto Challenges.** For all quality competitions there was one corresponding Pareto challenge. The rationale of the Pareto challenges was to take the work into account an algorithm requires to compute a solution. Hence, the two dimensions considered here were *quality* and *work*. Work was normalized with respect to the machine performance, measured by a graph-based benchmark. To this end, we used the shortest path benchmark produced for the 9th DIMACS Implementation Challenge. Participants were asked to run this sequential benchmark on their machine. Both the performance obtained in the shortest path benchmark and the number of processing cores (raised to the power of 0.9) used for the 10th DIMACS Implementation Challenge were taken into account for normalizing the amount of work invested for obtaining the solution.

For each challenge instance result, each submitted solver received a Pareto dominance count, which expresses by how many other algorithms it was Pareto-dominated in terms of work *and* running time; then algorithms were ranked by this number (lower count = better) and received points according to the Formula 1 scoring scheme described above.

Several groups submitted solutions from more than one solver to the respective Pareto challenges, making use of the fact that here a lower solution quality might be compensated by a better running time and vice versa. Still, the Pareto challenges were won in all cases by the same groups that also won the respective quality competitions. We attribute this double success (i) to the superior quality which could not be dominated in many cases and (ii) to the Formula 1 scoring scheme, which might have given an advantage to groups who submitted solutions from several solvers. More information on the challenge results are available online.<sup>8</sup>

**3.3. URL to Resources.** The main website of the 10th DIMACS Implementation Challenge can be found at its permanent location <http://www.cc.gatech.edu/dimacs10/>. The following subdirectories contain:

- [archive/data/](#): Testbed instances archived for long-term access.
- [talks/](#): Slides of the talks presented at the workshop.
- [papers/](#): Papers on which the workshop talks are based.

---

<sup>6</sup>Andrea Lancichinetti and Santo Fortunato, Limits of modularity maximization in community detection, Phys. Rev. E 84 (2011), 066122.

<sup>7</sup><http://www.cc.gatech.edu/dimacs10/data/dimacs10-rules.pdf>

<sup>8</sup><http://www.cc.gatech.edu/dimacs10/talks/orga-talk-dimacs-results.pdf>

- **results/**: Partitions submitted as part of the challenge as well as code for their evaluation and the resulting data

All respective files can be found and downloaded by following links from the homepage. Researchers are particularly encouraged to download and use the graphs we compiled and archived.

#### 4. Contributions to this Collection

In this section we give a short overview of the papers that were selected for this collection. All of them were presented at the Workshop of the 10th DIMACS Implementation Challenge and contributed to the success of the event. Not all solvers described in these papers actually entered the challenge. Also, not all solvers that entered the challenge are part of this collection.

**4.1. Graph Partitioning.** The winner in terms of graph partitioning quality was KAPA, by Sanders and Schulz, described in their paper *High Quality Graph Partitioning*. KAPA combines the solutions of several related solvers developed by the same authors. It is a set of algorithms which use a combination of strategies. Among these strategies are network flows, evolutionary algorithms, edge ratings for approximate maximum weighted matchings in the multilevel process, repetitive improvement cycles, and problem-specific local search techniques based on the Fiduccia-Mattheyses (FM) heuristic.

*Abusing a Hypergraph Partitioner for Unweighted Graph Partitioning*, by Fagginger Auer and Bisseling, describes Mondriaan, a package for matrix and hypergraph partitioning, and its (ab)use for graph partitioning. While Mondriaan usually computes worse edge cuts than state-of-the-art graph partitioners, the solutions are generally acceptable.

In *Parallel Partitioning with Zoltan: Is Hypergraph Partitioning Worth It?*, Rajamanickam and Boman describe a partitioner which is very powerful in that it is designed for scalable parallelism on large asymmetric hypergraphs.

Çatalyürek, Deveci, Kaya, and Uçar present in *UMPa: A Multi-objective, multi-level partitioner* a system doing recursive multi-objective hypergraph bipartitioning that takes the bottleneck communication volume as primary objective function into account but also looks for solutions with small total communication.

The related task of repartitioning dynamic graphs is addressed by Meyerhenke in *Shape Optimizing Load Balancing for MPI-Parallel Adaptive Numerical Simulations*. Diffusive methods are employed to determine both *how many* elements have to migrate between processors as well as *which* elements are chosen for migration. The properties of the diffusive processes usually lead to nicely shaped partitions.

In *Graph Partitioning for Scalable Distributed Graph Computations*, by Buluc and Madduri, the authors develop a method for partitioning large-scale sparse graphs with skewed degree distribution. The approach aims to partition the graph into balanced parts with low edge cuts, a challenge for these types of graphs, so that they can be used on distributed-memory systems where communication is often a major bottleneck in running time. The authors derive upper bounds on the communication costs incurred for a two-dimensional partitioning during breadth-first search. The performance results using the large-scale DIMACS challenge graphs shows that reducing work and communication imbalance among partitions is more important than minimizing the total edge cut.

**4.2. Graph Clustering.** *Using Graph Partitioning for Efficient Network Modularity Optimization*, by Djidjev and Onus, describes how to formulate modularity maximization in graph clustering as a minimum cut problem in a complete weighted graph. In general, the according graph contains also negative weights. However, the resulting minimum cut problem can be attacked by applying modifications of existing powerful codes for graph partitioning.

The solver VNS, by Aloise, Caporossi, Hansen, Liberti, and Perron, performs *Modularity Maximization in Networks by Variable Neighborhood Search*, a meta-heuristic and variant of local search. A local search or improving heuristic consists of defining a neighborhood of a solution, choosing an initial solution  $x$ , and then moving to the best neighbor  $x'$  of  $x$  if the objective function value is improved. If no such neighbor exists, the heuristic stops, otherwise it is iterated. VNS improves this simple technique to escape from local optima. To this end, it applies the idea of neighborhood change. By increasing the neighborhood distance iteratively, even "mountain tops" surrounding local optima can be escaped.

The algorithm family  $k$ -COMMUNITY, developed by Verma and Butenko in *Network Clustering via Clique Relaxations: A Community Based Approach*, are based on the relaxation concept of a *generalized community*. Instead of requiring a community to be a perfect clique, a generalized  $k$ -community is defined as a connected subgraph such that the incident vertices of every edge have at least  $k$  common neighbors within the subgraph. The algorithm family computes clusters by finding  $k$ -communities for large (variable)  $k$  and placing them in different clusters.

*Identifying Base Clusters for Maximizing Modularity*, by Srinivasan, Chakraborty, and Bhowmick, introduces the concept of identifying base clusters as a preprocessing step for agglomerative modularity maximization methods. Base clusters are groups of vertices that are always assigned to the same community, independent of the modularity maximization algorithm employed or the order in which the vertices are processed. In a computational study on two agglomerative modularity maximization methods, the CNM method introduced by Clauset et al. and the Louvain method by Blondel et al., the effect of using base clusters as a preprocessing is shown.

*Complete Hierarchical Cut-Clustering: A Case Study on Expansion and Modularity*, by Hamann, Hartmann, and Wagner, studies the behavior of the cut-clustering algorithm of Flake et al., a clustering approach which is based on minimum  $s$ - $t$ -cuts. The algorithm uses a parameter that provides a quality guarantee on the clusterings in terms of expansion. This is particularly interesting since expansion is a measure which is already NP-hard to compute. While Flake et al. examine their algorithm with respect to the semantic meaning of the clusters, Hamann et al. systematically analyze the quality of the clusterings beyond the guaranteed bounds with respect to the approved measures expansion and modularity.

In *A Partitioning-based divisive clustering technique for maximizing the modularity*, by Çatalyürek, Kaya, Langguth and Uçar, the authors present a new, divisive algorithm for computing high modularity clusterings. The approach is based upon recursive bipartitions using graph partitioning subroutines, and steps for refining the obtained clusters. The study includes an experimental evaluation. On a variety of problem instances from the literature, this new method performs well, and in a number of cases, finds the best known modularity scores on these test graphs.

*An Ensemble Learning Strategy for Graph Clustering*, by Ovelgönne and Geyer-Schulz, describes the heuristic CGGC\_LRG, whose main idea is to combine several weak classifiers into a strong classifier. From the maximal overlap of clusterings computed by weak classifiers, the algorithm searches for a solution with high quality. This way difficult choices are deferred after easy decisions have been fixed, which leads to a high quality due to a better control of the search space traversal. It turns out that the quality of the initial clusterings is of minor importance for the quality of the final result given enough iterations.

While graph partitioning is rooted in the parallel computing community, the picture appears to be different for graph clustering as only two clustering papers employ significant parallelism. The agglomerative algorithm in *Parallel Community Detection for Massive Graphs*, by Riedy, Meyerhenke, Ediger, and Bader, starts out with each vertex as its own cluster. In each following iteration, beneficial cluster merges improving the objective function value are identified and performed in parallel by means of weighted matchings. The implementation is capable of clustering graphs with a few billion edges in less than 10 minutes on a standard Intel-based server.

The second paper that uses considerable parallelism to accelerate the solution process is *Graph Coarsening and Clustering on the GPU*, by Fagginger Auer and Bisseling. This paper also uses an agglomerative approach with matchings. It alleviates the problem of small matchings due to star subgraphs by merging siblings, i. e., neighbors of neighbors that do not share an edge. High performance is achieved by careful algorithm design, optimizing the interplay of the CPU and the employed graphics hardware.

## 5. Directions for Further Research

In the field of graph partitioning, important directions for further research mentioned at the workshop are the widespread handling of directed graphs (or unsymmetric matrices in case of matrix partitioning) and an improved consideration of the objective function maximum communication volume. One possible approach—also presented at the workshop—is to use hypergraphs instead of graphs. But this seems to come at the price of worse performance and/or worse edge cut quality. For the related problem of repartitioning with migration minimization, highly scalable tools with a good solution quality are sought.

An active graph clustering research area is the development of objective functions whose optimization leads to realistic and meaningful clusterings. While modularity has been very popular over recent years, current studies show that its deficiencies can be severe and hard to avoid. The analysis of massive graphs for clustering purposes is still in its infancy. Only two submissions for the graph clustering challenge made use of significant parallelism. And only one of them was able to process the largest graph in the challenge core benchmark, a web graph with 3.3 billion edges. Considering the size of today’s online social networks and WWW (to name a few), there is a need to scale the analysis algorithms to larger input sizes.



## Selected Published Titles in This Series

- 588 **David A. Bader, Henning Meyerhenke, Peter Sanders, and Dorothea Wagner, Editors**, Graph Partitioning and Graph Clustering, 2013
- 587 **Wai Kiu Chan, Lenny Fukshansky, Rainer Schulze-Pillot, and Jeffrey D. Vaaler, Editors**, Diophantine Methods, Lattices, and Arithmetic Theory of Quadratic Forms, 2013
- 584 **Clara L. Aldana, Maxim Braverman, Bruno Iochum, and Carolina Neira Jiménez, Editors**, Analysis, Geometry and Quantum Field Theory, 2012
- 583 **Sam Evens, Michael Gekhtman, Brian C. Hall, Xiaobo Liu, and Claudia Polini, Editors**, Mathematical Aspects of Quantization, 2012
- 582 **Benjamin Fine, Delaram Kahrobaei, and Gerhard Rosenberger, Editors**, Computational and Combinatorial Group Theory and Cryptography, 2012
- 581 **Andrea R. Nahmod, Christopher D. Sogge, Xiaoyi Zhang, and Shijun Zheng, Editors**, Recent Advances in Harmonic Analysis and Partial Differential Equations, 2012
- 580 **Chris Athorne, Diane Maclagan, and Ian Strachan, Editors**, Tropical Geometry and Integrable Systems, 2012
- 579 **Michel Lavrauw, Gary L. Mullen, Svetla Nikova, Daniel Panario, and Leo Storme, Editors**, Theory and Applications of Finite Fields, 2012
- 578 **G. López Lagomasino**, Recent Advances in Orthogonal Polynomials, Special Functions, and Their Applications, 2012
- 577 **Habib Ammari, Yves Capdeboscq, and Hyeonbae Kang, Editors**, Multi-Scale and High-Contrast PDE, 2012
- 576 **Lutz Strümgmann, Manfred Droste, László Fuchs, and Katrin Tent, Editors**, Groups and Model Theory, 2012
- 575 **Yunping Jiang and Sudeb Mitra, Editors**, Quasiconformal Mappings, Riemann Surfaces, and Teichmüller Spaces, 2012
- 574 **Yves Aubry, Christophe Ritzenthaler, and Alexey Zykin, Editors**, Arithmetic, Geometry, Cryptography and Coding Theory, 2012
- 573 **Francis Bonahon, Robert L. Devaney, Frederick P. Gardiner, and Dragomir Šarić, Editors**, Conformal Dynamics and Hyperbolic Geometry, 2012
- 572 **Mika Seppälä and Emil Volcheck, Editors**, Computational Algebraic and Analytic Geometry, 2012
- 571 **José Ignacio Burgos Gil, Rob de Jeu, James D. Lewis, Juan Carlos Naranjo, Wayne Raskind, and Xavier Xarles, Editors**, Regulators, 2012
- 570 **Joaquín Pérez and José A. Gálvez, Editors**, Geometric Analysis, 2012
- 569 **Victor Goryunov, Kevin Houston, and Roberta Wik-Atique, Editors**, Real and Complex Singularities, 2012
- 568 **Simeon Reich and Alexander J. Zaslavski, Editors**, Optimization Theory and Related Topics, 2012
- 567 **Lewis Bowen, Rostislav Grigorchuk, and Yaroslav Vorobets, Editors**, Dynamical Systems and Group Actions, 2012
- 566 **Antonio Campillo, Gabriel Cardona, Alejandro Melle-Hernández, Wim Veys, and Wilson A. Zúñiga-Galindo, Editors**, Zeta Functions in Algebra and Geometry, 2012
- 565 **Susumu Ariki, Hiraku Nakajima, Yoshihisa Saito, Ken-ichi Shinoda, Toshiaki Shoji, and Toshiyuki Tanisaki, Editors**, Algebraic Groups and Quantum Groups, 2012
- 564 **Valery Alexeev, Angela Gibney, Elham Izadi, János Kollár, and Eduard Looijenga, Editors**, Compact Moduli Spaces and Vector Bundles, 2012

For a complete list of titles in this series, visit the  
AMS Bookstore at [www.ams.org/bookstore/conmseries/](http://www.ams.org/bookstore/conmseries/).



Graph partitioning and graph clustering are ubiquitous subtasks in many applications where graphs play an important role. Generally speaking, both techniques aim at the identification of vertex subsets with many internal and few external edges. To name only a few, problems addressed by graph partitioning and graph clustering algorithms are:

- What are the communities within an (online) social network?
- How do I speed up a numerical simulation by mapping it efficiently onto a parallel computer?
  - How must components be organized on a computer chip such that they can communicate efficiently with each other?
  - What are the segments of a digital image?
  - Which functions are certain genes (most likely) responsible for?

The 10th DIMACS Implementation Challenge Workshop was devoted to determining realistic performance of algorithms where worst case analysis is overly pessimistic and probabilistic models are too unrealistic. Articles in the volume describe and analyze various experimental data with the goal of getting insight into realistic algorithm performance in situations where analysis fails.

American Mathematical Society  
[www.ams.org](http://www.ams.org)

Center for Discrete Mathematics and Theoretical Computer Science  
[dimacs.rutgers.edu](http://dimacs.rutgers.edu)

ISBN 978-0-8218-9038-7



9 780821 890387

CONM/588