

# **Graph Algorithms and Applications**

**(Dagstuhl–Seminar 98301)**

**Organizers:**

**Takao Nishizeki (Tohoku University Sendai, Japan)**  
**Roberto Tamassia (Brown University, USA)**  
**Dorothea Wagner (Universität Konstanz, Germany)**

**July 26 – 31, 1998**

Algorithmic graph theory is a classical area of research by now and has been rapidly expanding during the last three decades. In many different contexts of computer science and applications, modelling problems by graphs is a natural and canonical process. Graph–theoretic concepts and algorithms play an important role in many fields of application, e.g. in communication network design, VLSI–design, CAD, traffic optimization or network visualization.

Apart from the design and analysis of algorithms for solving fundamental graph problems, the application of these methods to real world problems is an interesting task for researchers in algorithmic graph theory. Recently, researchers also started developing software systems for graph algorithms to provide effective computational tools to support applications prototyping, algorithm animation or further algorithmic research. Several algorithm libraries, algorithm animation tools or special purpose software packages, e.g. graph editors and graph drawing software, have been developed within the last five to ten years.

This seminar was intended to bring together researchers from different areas in algorithmic graph theory and from application. One aim was to support the collaboration between computer scientists, mathematicians, and applied researchers, both from academia and industry in the field.

Main topics of interest were on the one hand classical problems from graph theory such as connectivity and cuts, paths and flows, coloring problems and theoretical aspects of graph drawing. On the other hand, problems from application where those concepts are of special importance were discussed. Particular emphasis was placed on experimental research and aspects of the

implementation of graph algorithms. One of the central topics was “graph drawing”, which addresses the problem of visualizing structural information. The automatic generation of drawings of graphs has important applications in key computer science technologies such as database design, software engineering, VLSI and network design and visual interfaces. Applications in other sciences concern all fields of visual data mining, e.g. in engineering, chemistry and biology, archeology or sociology and political science. The interaction between theoretical improvements and implemented solutions is an important part of the area of graph drawing.

We had 48 participants from Austria, Germany, France, Italy, Poland, Hungary, Slovenia, Israel, Australia, USA, Canada, Japan and Korea. During the workshop, 38 lectures, some including also software demonstrations, were given. Most of the talks presented very recent research results. The informal character of the workshop made it possible to have intensive discussions. There was an open-problem-session and a lively discussion on new directions in graph drawing. Both, senior researchers as well as young researchers, contributed to this seminar.

The organizers plan to edit a special volume of the *Journal of Graph Algorithms and Applications* with selected papers addressing areas covered by the seminar.

All participants were greatly pleased that several young researchers were able to attend this workshop due to special European funding from the TMR-Program that the Dagstuhl Institution was able to organize. This contributed to the especially inspiring and refreshing atmosphere of the seminar. Schloß Dagstuhl and its staff provided a very convenient and stimulating environment. The organizers wish to thank all those who helped to make the workshop a fruitful research experience.

## Participants

Takao Asano, Chuo University  
Vladimir Batagelj, University of Ljubljana  
Franz J. Brandenburg, Universität Passau  
Ulrik Brandes, Universität Konstanz  
Kyung-Yong Chwa, KAIST  
Gabriele Di Stefano, Università degli Studi di L'Aquila  
Yefim Dinitz, Technion — Haifa  
Peter Eades, University of Newcastle  
Shimon Even, Technion — Haifa  
András Frank, Eötvös Loránd University Budapest  
Hubert de Fraysseix, EHESS  
Uli Fößmeier, Tom Sawyer Software  
Emden Gansner, AT&T Labs-Research  
Ashim Garg, SUNY at Buffalo  
Dagmar Handke, Universität Konstanz  
Michael Himsolt, Universität Passau  
Seokhee Hong, University of Newcastle  
Christoph Hundack, Max-Planck-Institut für Informatik  
Kazuo Iwama, Kyoto University  
Michael Kaufmann, Universität Tübingen  
Giuseppe Liotta, Brown University  
Ernst W. Mayr, Technische Universität München  
Kurt Mehlhorn, Max-Planck-Institut für Informatik  
Petra Mutzel, Max-Planck-Institut für Informatik  
Matthias Müller-Hannemann, Technische Universität Berlin  
Stefan Näher, Universität Halle  
Takao Nishizeki, Tohoku University  
Patrice Ossona de Mendez, EHESS  
András Recski, Technical University of Budapest  
Heike Ripphausen-Lipa, Lufthansa Systems Berlin GmbH  
Martin Skutella, Technische Universität Berlin  
Angelika Steger, Technische Universität München  
Maciej M. Sysło, University of Wrocław  
Roberto Tamassia, Brown University  
Ronit Teplixke, Technion — Haifa  
Ioannis Tollis, University of Texas at Dallas  
Paola Vocca, Università di Roma “Tor Vergata”  
Dorothea Wagner, Universität Konstanz  
Frank Wagner, Freie Universität Berlin

Karsten Weihe, Universität Konstanz  
Sue Whitesides, McGill University  
Peter Widmayer, Eidgenössische Technische Universität Zürich  
Roland Wiese, Universität Tübingen  
Gerhard Wöginger, Technische Universität Graz  
Maria Zapolotsky, Technion — Haifa  
Christos Zaroliagis, Max-Planck-Institut für Informatik

## Program

### Monday, July 27

#### Morning Session

- 9:15– 9:30 Welcome
- 9:30– 10:00 Takao Nishizeki: Disjoint Paths Problem for Partial  $k$ -Trees: —  $\mathcal{NP}$ -Completeness and Tractable Cases
- 10:00– 10:30 Yefim Dinitz: On Single-Source Unsplittable Flows
- 11:00– 11:30 Gabriele Di Stefano: Graphs with Bounded Induced Distance
- 11.30– 12:00 Dagmar Handke: Independent Tree Spanners: Fault-tolerant Spanning Trees with Constant Distance Guarantees

#### Afternoon Session

- 15:00– 15.30 Frank Wagner: Mimicking Networks
- 16:15– 16:45 András Recski: Graph (and Matroid) Algorithms in Statics: Rigidity of Regular Grid-Like Structures
- 16:45– 17:15 Angelika Steger: Approximability of Scheduling with Fixed Jobs
- 17:15– 17:45 Martin Skutella: MaxCut Algorithms for Parallel Machine Scheduling

### Tuesday, July 28

#### Morning Session

- 9:00– 9:30 Heike Ripphausen-Lipa: Airline Crew Scheduling
- 9:30– 10:00 Matthias Müller-Hannemann: Hexahedral Mesh Generation in CAD or Shelling an Unknown Hexahedral Complex
- 10:30– 11:00 Giuseppe Liotta: Upward Planarity Checking: Faces Are More than Polygons
- 11:00– 11:30 Vladimir Batagelj: Analysing and Drawing Genealogies
- 11:30– 12:00 Karsten Weihe: Covering Trains

#### Afternoon Session

- 14:30– 15:00 Kurt Mehlhorn: A Variant of Dijkstra's Algorithm
- 15:00– 15:30 Christos Zaroliagis: An Experimental Study of Dynamic Algorithms for Transitive Closure
- 16:15– 16:45 Kazuo Iwama: Traffic Control for Cities of Mesh Structure
- 16:45– 17:15 Michael Himsolt: Graph Scripting
- 17:15– 17:45 Emden Gansner: Path Routing in Undirected Graphs

## Wednesday, July 29

### Morning Session

9:00 – 9:30	Stefan Näher: Animation of Graph Algorithms
9:30 – 10:00	Franz J. Brandenburg: Symmetries in Graphs
10:00 – 10:30	Peter Eades: Drawing Graphs Symmetrically
11:00 – 11:30	András Frank: Increasing the Connectivity of a Digraph
11:30 – 12:00	Petra Mutzel: New Approximation Algorithms for Planar Augmentation

Afternoon                      Excursion

20:00                              Discussion: New Directions in Graph Drawing  
   Chaired by Ioannis Tollis

## Thursday, July 30

### Morning Session

9:00 – 9:30	Takao Asano: Efficiency of an Algorithm for the Set Cover Problem Based on Semi-Local Improvements
9:30 – 10:00	Ulrik Brandes: Dynamic Orthogonal Graph Layout
10:00 – 10:30	Uli Fößmeier: Compaction and Bend-Saving Strategies in Orthogonal Drawings
11:00 – 11:30	Shimon Even: Area Efficient Layouts of the Batchier Sorting Networks
11:30 – 12:00	Kyung-Yong Chwa: Carrying an Umbrella: An Online Game on a Graph

### Afternoon Session

14:30 – 15:00	Sue Whitesides: Drawing Graphs Nicely Without Defining Nice
15:00 – 15:30	Roland Wiese: Orthogonal Drawing with Constraints
16:15 – 16:45	Ioannis Tollis: The Secrets of Labeling Graph Drawings
16:45 – 17:15	Dorothea Wagner: Graph Drawing in Different Fields of Application
17:15 – 17:45	Michael Kaufmann: Visualization of Parallel Processes
20:00	Open Problem Session Chaired by Roberto Tamassia

## Friday, July 31

### Morning Session

9:00 – 9:30	Hubert de Fraysseix: Distributive Lattices on Planar Graphs: Two Years Before
9:30 – 10:00	Christoph Hundack: Planarity Revisited
10:30 – 11:00	Peter Widmayer: Survivability in Networks
11:00 – 11:30	Maria Zapolotsky: An Optimal Layout of the Butterfly Network
11:30 – 12:00	Maciej Syslo: Hamiltonian Amalgams

# Disjoint Paths Problems for Partial $\gamma$ -Trees – NP-Completeness and Tractable Cases –

Takao Nishizeki  
Graduate School of Information Sciences  
Tohoku University

In this talk we show that the edge-disjoint paths problem is NP-complete for partial  $k$ -trees with bounded  $k$ , say  $k = 3$ , and present some tractable cases.

## On Single-Source Unsplittable Flows

Yefim Dinitz  
Department of Computer Sciences  
Technion, Haifa

Let  $G = (V, E)$  be a capacitated directed network with a source  $s$  and  $k$  sinks  $t_i$  with demands  $d_i$ ,  $1 \leq i \leq k$ . We would like to concurrently route every demand *unsplittably*, i.e., on a *single* path from  $s$  to the corresponding sink, without violating the capacities or violating them minimally. This *unsplittable flow problem* arises from routing in high-speed communication networks, as well as from scheduling and load balancing.

Assuming that there exists a flow from  $s$  to all  $t_i$  satisfying all the demands (a necessary condition), we show how to compute an unsplittable flow such that the total flow through any edge exceeds its capacity by at most the maximum demand. For graphs in which all capacities are at least the maximum demand, we therefore obtain an unsplittable flow with congestion at most 2, improving the best known approximation ratio of 3.23. Moreover, this result is best possible.

Using this result, we show that all demands can be routed unsplittably in 5 rounds, i.e., all demands can be collectively satisfied by the union of 5 unsplittable flows.

This is joint work with N. Garg and M. Goemans.



# Graphs with Bounded Induced Distance

Gabriele Di Stefano  
Dipartimento di Ingegneria Elettrica  
Università di L'Aquila

In this work we introduce *graphs with bounded induced distance* of order  $k$  (BID( $k$ ) for short). In any graph belonging to BID( $k$ ), the length of every induced path between every pair of nodes is at most  $k$  times the distance between the same nodes. In communication networks modeled by these graphs any message can be always delivered through a path whose length is at most  $k$  times the best possible one, even if some nodes fail.

In this work we first provide a characterization of graphs in BID( $k$ ) by means of cycle-chord conditions. After that, we investigate classes with order  $k \leq 2$ . In this context, we note that the class BID(1) is the well known class of distance-hereditary graphs, and show that  $3/2$  is a lower bound for the order  $k$  of graphs that are not distance-hereditary. Then we characterize graphs in BID( $3/2$ ) by means of their minimal forbidden induced subgraphs, and we also show that graphs in BID(2) have a more complex characterization. We prove that the recognition problem for the generic class BID( $k$ ) is Co-NP-complete. Finally, we show that the split composition can be used to generate graphs in BID( $k$ ).

This is joint work with Serafino Cicerone.

## Independent Tree Spanners: Fault-tolerant Spanning Trees with Constant Distance Guarantees

Dagmar Handke  
Fakultät für Mathematik und Informatik  
Universität Konstanz

For any fixed parameter  $t \geq 1$ , a *tree  $t$ -spanner* of a graph  $G$  is a spanning tree  $T$  of  $G$  such that the distance between every pair of vertices in  $T$  is at most  $t$  times their distance in  $G$ . In this talk, we incorporate a concept of fault-tolerance by examining *independent tree  $t$ -spanners*. Given a root vertex  $r$ , this is a pair of tree  $t$ -spanners, such that the two paths from any

vertex to  $r$  are edge (resp., internally vertex) disjoint. It is shown that a pair of independent tree 2-spanners can be found in linear time, whereas the problem for arbitrary  $t \geq 4$  is  $\mathcal{NP}$ -complete.

As a less restrictive concept, we treat *tree  $t$ -root-spanners*, where the distance constraint is relaxed. Here, we show that the problem of finding an *independent* pair of such subgraphs is  $\mathcal{NP}$ -complete for all  $t$ . As a special case, we then consider *direct tree  $t$ -root-spanners*. These are tree  $t$ -root-spanners where paths from any vertex to the root have to be detour-free. In the *edge* independent case, a pair of these can be found in linear time for all  $t$ , whereas the *vertex* independent case remains  $\mathcal{NP}$ -complete.

## Mimicking Networks

Frank Wagner  
Fachbereich Mathematik und Informatik  
Freie Universität Berlin

The idea of this talk is to take a *network* with a set of  $k$  special *terminal* vertices and replace it by a simpler network with  $k$  terminals, that has the same flow properties. We show that in general a  $k$ -terminal network needs a network of size at least  $2^{\frac{k}{2}}$  to mimic it. An upper bound is  $2^{\frac{k}{k}}$ . For small  $k$  (3, 4, 5) we present the optimal mimicking networks (with 3, 5, 6 vertices resp.). For the class of partial  $k$ -trees with  $l$  terminals we can show that  $2^{2^k} \cdot l$  vertices are enough, for outerplanar graphs  $10 \cdot l$  is already enough.

The problem came up as a tool to design new flow and cut algorithms. It seems to be of interest on its own.

This is joint work with Shiva Chaudhuri, K.V. Subramanyam, Klaus Kriegel, Torsten Thiele, and Christos Zaroliagis.

# Graph (and Matroid) Algorithms in Statics: Rigidity of Regular Grid-Like Structures

András Recski

Faculty of Electrical Engineering and Information  
Technical University of Budapest

1. A  $k \times l$  square grid is considered as a bar-and-joint framework in the plane. At least  $k + l - 1$  additional diagonal bars are needed to make it rigid and the minimum systems are just the spanning trees of a bipartite graph where the two vertex classes correspond to the rows and columns, respectively, of the grid and edges indicate diagonals in their intersections (Bolker and Crapo).

This motivated some recent results of H. Gabow, J. Bang-Jensen, T. Jordán and Z. Szigeti how to increase the connectivity of a bipartite graph by inserting new edges but preserving bipartiteness.

2. If the four corners of the grid are pinned down, only  $k + l - 2$  diagonal bars are needed: the corresponding subgraph must be an *asymmetric* 2-component forest (ratio of cardinalities of the vertex classes in the components must be different from  $k/l$ ), see Bolker and Crapo.

Some 10 years ago we extended this construction to arbitrary vertex-weighted graphs: If  $F$  is an arbitrary field and  $w : V(G) \rightarrow F$  is the weight function satisfying  $\sum_{v \in V(G)} w(v) = 0$  then those 2-component forests are asymmetric where  $\sum w(v) \neq 0$  for the individual components. These forests form the bases of some matroids which were shown to be elementary strong maps of the cycle matroid of  $G$ .

3. On the other hand, if we remove one of the original (horizontal or vertical) bars then  $k + l$  diagonal bars will be needed. The good systems were recently characterized (joint result with Zs. Gáspár and N. Radics).

In this talk we use this idea to define two new matroidal families on graphs where the bases are subgraphs with one more (rather than with one less, as in Part 2) edge than in the spanning trees. Representability properties of these matroids and their relations to certain submodular functions (result of A. Frank) are also presented.

## Approximability of Scheduling with Fixed Jobs

Angelika Steger  
Institut für Informatik  
Technische Universität München

In the standard scheduling problem of minimizing the makespan one is given a set of  $n$  independent jobs with processing times  $p_j$  to be scheduled on  $m$  identical machines in such a way that the latest completion time (the makespan) is minimized. In modern industrial software it has become standard to work on a variant of this problem, where some of the jobs are already fixed in the schedule. The remaining jobs are to be assigned to the machines in such a way that they do not overlap with fixed jobs. In this talk we present a polynomial time approximation scheme for the scheduling problem with fixed jobs for the case that the number  $m$  of machines is constant. This result is essentially best possible, as we also show that, unless  $P = NP$ , there exists no polynomial time approximation scheme in the case when the number of machines is part of the input and no fully polynomial time approximation scheme in the constant machine case.

This is joint work with M. Scharbrodt and H. Weisser.

## MaxCut Algorithms for Parallel Machine Scheduling

Martin Skutella  
FB 3 Mathematik  
Technische Universität Berlin

We consider the problem of scheduling a set of jobs on unrelated parallel machines or processors so as to minimize the weighted sum of the completion times of jobs. Whereas the best previously known approximation algorithms for this problem are based on LP relaxations, we give a  $\frac{3}{2}$ -approximation algorithm that relies on a convex quadratic programming relaxation. For two unrelated parallel machines this result can be further improved by using a more sophisticated semidefinite programming relaxation. For the special case of identical parallel machines we establish approximation preserving reductions to MaxCut problems. In particular, we discuss the connections to

the MaxCut results of Goemans and Williamson that are also based on semidefinite relaxations. This is the first time that convex and semidefinite programming techniques are used in the area of scheduling.

## Airline Crew Scheduling

Heike Ripphausen-Lipa  
Lufthansa Systems Berlin GmbH

The airline crew scheduling problem is the problem of finding an optimal, complete, feasible assignment of persons (crew members) to the crews of flights/parings (sequence of flights). Optimality criteria are minimizing costs and maximizing job satisfaction of the crew members. There are three criteria for feasible assignments:

- crew member must have correct qualification
- the duties mapped to crew member must be consistent w.r.t. time and location
- many regulations must be considered.

We describe the typical approach (set partition) with its advantages and disadvantages. Then we will present some heuristics we use for solving the problem in an Esprit Project named “PARROT”. Due to all these heuristics is the usage of certain graphs which we call legality graphs. These graphs indicate legal assignments of duties to crew members. In our heuristics we use basic graph algorithms on these graphs (matching, shortest paths, ...) for finding good solutions to the problem.

# Hexahedral Mesh Generation in CAD or Shelling an Unknown Hexahedral Complex

Matthias Müller-Hannemann  
FB 3 Mathematik  
Technische Universität Berlin

We propose a new method for constructing all-hexahedral finite element meshes. The core of our method is to build up a compatible combinatorial cell complex of hexahedra for a solid body which is topologically a ball and for which a quadrilateral surface mesh of a certain structure is prescribed. The step-wise creation of the hex complex is guided by the cycle structure of the combinatorial dual of the surface mesh. Our method transforms the graph of the surface mesh iteratively by changing the dual cycle structure until we get the surface mesh of a single hexahedron. Starting with a single hexahedron and reversing the order of the graph transformations, each transformation step can be interpreted as adding one or more hexahedra to the so far created hex complex.

Given an arbitrary solid body, we first decompose it into simpler subdomains equivalent to topological balls by adding virtual 2-manifolds. Second, we determine a compatible quadrilateral surface mesh for all created subdomains. Then, in the main part we can use the core routine to build up a hex complex for each subdomain independently. The embedding and smoothing of the combinatorial mesh(es) finishes the mesh generation process.

## Upward Planarity Checking: “Faces Are More Than Polygons”

Giuseppe Liotta  
Dipartimento di Informatica e Sistemistica  
Università di Roma “La Sapienza”

In this paper we look at upward planarity from a new perspective. Namely, we study the problem of checking whether a given drawing is upward planar. Our checker exploits the relationships between topology and geometry of upward planar drawings to verify the upward planarity of a significant family

of drawings. The checker is simple and optimal both in terms of efficiency and in terms of degree.

This is joint work with Giuseppe Di Battista.

## Analysis and Visualization of Genealogies

Vladimir Batagelj  
Department of Theoretical Computer Science  
University of Ljubljana

**Pajek** is a Windows program for analysis and visualization of large graphs and networks. It is freely available for noncommercial use at

<http://vlado.fmf.uni-lj.si/pub/networks/pajek/>

Genealogies are examples of large networks available already in a computerized form on the Internet, usually in **GEDCOM** format which is supported also by **Pajek**. They can be transformed into two different types of graphs: Ore graphs and p-graphs. p-graphs are new representation of genealogies by graphs which is more concise and allows simpler procedures for their analysis. For additional information on p-graphs see

<http://eclectic.ss.uci.edu/~drwhite/pgraph/p-graphs.html>

p-graphs are acyclic directed graphs.

For drawing genealogies represented by p-graphs an algorithm was developed along the standard scheme (Sugiyama): determine the levels of vertices, determine the positions of vertices on the same layers.

To obtain satisfactory drawings we developed a special procedure for determining levels (long jumps between levels are not allowed). Since connected p-graphs are almost trees and determining layers on trees is easy we first remove all pending trees from the graph. On the remaining graph we determine levels according to the connected system of longest paths connecting its first and last vertices. Finally we extend levels to the removed trees.

**Pajek** offers different options for positioning vertices inside layers (line (2D) or plane (3D)); minimization of crossings, **Negopy**-like iterative procedure). 3D drawings can be exported in **VRML** and **MOL** format.

**Pajek** can be used also to analyze (large) genealogies, e.g. searching for the shortest kinship paths between two persons, extracting predecessors and successors graph of selected persons, searching for interesting marriage patterns (blood marriages, relinking marriages), searching for persons having many parents or many children, . . . From sociological point of view biconnected components (in the skeleton) of p-graphs are of special interest.

This is joint work with Andrej Mrvar.

## **Covering Trains by Stations or The Power of Data Reduction**

Karsten Weihe  
Fakultät für Mathematik und Informatik  
Universität Konstanz

Given a set of train courses (regarded as sets of stations), the problem is to find a minimum set of stations such that each train stops at one or more stations in this set. This is a special case of the *hitting-set problem*. The talk demonstrates empirically that sets of trains selected from the European train schedule are invariantly easily tractable from a practical viewpoint. The key step is a simple, yet rigorous application of data-reduction techniques in a preprocessing phase. In all experiments it turned out that the result of this preprocessing is small enough to make a brute-force approach feasible. For many instances the preprocessing even yields an optimum solution 'for free'.



## **Finding Kuratowski Subgraphs and a Parallelization of Dijkstra's Algorithm**

Kurt Mehlhorn  
Max-Planck-Institut für Informatik

The first talk of the talk is a follow-up to the talk given at the Dagstuhl meeting on Graph Algorithms in 96. In 96 I presented a linear time algorithm for finding Kuratowski subgraphs in non-planar graphs. This time I reported about the implementation and demonstrated the algorithm. The algorithms is described in detail in the chapter on Embedded Graphs of the forthcoming LEDA book.

This part of the talk is joint work with Christoph Hundack und Stefan Näher.

In the second talk of the talk I discuss a variant of Dijkstra's algorithm which tries to remove more than one node from the queue in each iteration. Two heuristics are described. For random graphs it can be shown experimentally and theoretically that the heuristics are highly effective.

This part of the talk is joint work with Andreas Crauser, Uli Meyer and Peter Sanders.

## **An Experimental Study of Dynamic Algorithms for Transitive Closure**

Christos Zaroliagis  
King's College — Department of Computer Science  
University of London

We have conducted an extensive experimental study of almost all known dynamic algorithms for the transitive closure problem on digraphs. Our experiments involve incremental algorithms on digraphs, decremental algorithms on DAGs and fully dynamic algorithms on digraphs and DAGs. Besides the original implementation of the theoretical algorithms, we have implemented several variants of them which improved significantly their running times, as well as several simple-minded algorithms (based on static approaches) that were easy to implement and likely to be fast in practice. In addition, we have developed a new algorithm whose decremental part applies to any digraph,

not only to DAGs. All the implementations have been written in C++ using the LEDA library for combinatorial and geometric computing and are part of the LEDA Extension Package on Dynamic Graph Algorithms. Our experiments have been performed by generating several kinds of random inputs, non-random inputs (that are worst-case inputs for the dynamic algorithms), and on a real world graph: the graph modeling the connections among the autonomous systems of the Internet, and more precisely the fragment of the network visible from one of the main European servers. In all but the fully dynamic case on digraphs, the dynamic algorithms performed better than the simple-minded approaches. In the fully dynamic case on digraphs, the simple-minded algorithms were much faster.

This is joint work with D. Frigioni, T. Miller, U. Nanni, G. Pasqualone, and G. Schaefer.

## Traffic Control for Cities of Mesh Structure

Kazuo Iwama  
Department of Information Science  
Kyoto University

Our model of a city is an  $n \times n$  mesh, namely, the city has  $n$  east-west streets and also  $n$  north-south avenues. Each vehicle  $V$  enters the city from the west using some street and goes out of the city to the north using some avenue.  $V$  changes its direction from east to north only once at the crossing of these two roads. There are  $n$  entering vehicles for each street and  $n$  out-going ones for each avenue. Hence we need only  $O(n)$  time if there is no congestion. However, many naive algorithms take  $\Omega(n^2)$  time in the worst case. Our new algorithm runs in  $O(n^{1.5})$ . The basic idea for this  $O(n^{1.5})$  algorithm can be applied to mesh routing; it gives a positive answer to the open question asking whether or not there is an oblivious, constant-buffer-size, routing algorithm which runs essentially faster than  $n^2$  steps.

It is also shown that this  $O(n^{1.5})$  algorithm is furthermore improved to an  $O(n)$  algorithm.

This is joint work with Yahiko Kambayashi and Eiji Miyano.

# Graph Scripting

Michael Himsolt  
Fachbereich Informatik  
Universität Passau

The `Graphscript` programming language is a `Tcl/Tk` based programming language for implementing graph algorithms and graph editors. It supports directed and undirected graphs, attributes such as labels, coordinates or color, and provides a powerful application programmer interface.

A key feature of `Graphscript` is extensibility. New features, even new control structures, can be added as `Tcl` procedures or through `C` or `C++` extensions.

The `Graphlet` graph editor is implemented in `Graphscript`, and lightly integrated with the library of algorithms. Algorithms can use the editor to input graphs, display results or animate data structures.

## Edge Routing in Undirected Graphs

Emden Gansner  
AT&T Labs-Research

Techniques for drawing graphs based on virtual physical models have proven surprisingly successful in producing good layouts of abstract undirected graphs. However, for drawing graphs arising from real-world data, where nodes must be labeled and point vertices are replaced by shapes with non-zero area, problems arise in the form of node-node and node-edge overlap, making the graph drawing hard to understand. We describe the application of two post-processing techniques which can be applied to any initial vertex layout to produce uncluttered layouts. The first technique uses the Voronoi diagram of the vertices to reposition the nodes to avoid node-node overlap while maintaining the general layout. The second technique involves routing edges as smooth curves to avoid node-edge overlap. We describe two methods for constructing edges as Bezier splines.

This is joint work with James Abello, Eleftherios Koutsofios, Stephen C. North and David P. Dobkin

# Visualization of Graph Algorithms, an event-based approach

Stefan Näher  
Fachbereich Mathematik und Informatik  
Universität Halle

We present an event-based approach for the visualization and animation of graph data structures and algorithms. It follows the Observer Design Pattern for separating and decoupling the algorithm implementation from the visualization tool. This is particular important when integrating animation support into an existing software library like the LEDA system.

This is joint work with Andreas Luleich.

## Symmetries in Graphs

Franz J. Brandenburg  
Lehrstuhl für Theoretische Informatik  
Universität Passau

We wish to find symmetries in graphs and then use these symmetries in graph drawing algorithms. Symmetries are expressed in terms of disjoint isomorphic subgraphs. Hence, a given graph is partitioned into  $k \geq 2$  identical copies and a remainder.

For  $k = 2$  and no remainder we obtain PCSP, the pair of common subgraph problem. Formally, PCSP states that a given graph  $G$  partitions into two isomorphic subgraphs with at least some  $K$  edges. PCSP is similar to well-known problems such as GRAPH ISOMORPHISM, LARGEST COMMON SUBGRAPH or GEOMETRIC GRAPH SYMMETRY. But it is new.

We show by reduction from 2-in-4SAT that PCSP is NP-hard. However, the largest common subtrees and induced subgraphs of a tree can be computed efficiently using techniques to compute largest common substrings and largest common subsequences.

The major open problem is PCSP on planar graphs.

## Drawing Graphs Symmetrically

Peter Eades  
Department of Computer Science  
University of Newcastle

Symmetry is one of the most important aesthetic criteria for graph drawing. An automorphism  $p$  of a graph is geometric if it has a symmetric drawing which induces  $p$ . The key to drawing graphs symmetrically is in finding geometric automorphisms. The problem of finding such automorphisms is NP-hard in general. However, for series-parallel digraphs it can be solved in polynomial time. We present an algorithm which finds a maximum size geometric automorphism group of a series parallel digraph. An algorithm for drawing series parallel digraphs symmetrically is also presented.

This is joint work with Seok-Hee Hong.

## Finding Minimum Edge-Coverings of Pairs of Sets

András Frank  
Department of Operations Research  
Eötvös Lorand University Budapest

One goal of this work is to describe a combinatorial polynomial-time algorithm to make a  $(k-1)$ -connected digraph  $k$ -connected by adding a minimum number of new edges, or more generally, a minimum cost of new edges with respect to a node-cost induced cost function. A related algorithm has recently been developed for a generalization of a theorem of E. Györi to find a minimum set of generators of a family of subpaths of a circuit.

These two problems are special cases of a general framework concerning the minimum number of edges covering every member of a crossing family of pairs of sets. We actually exhibit first a constructive proof of a min-max theorem of the author and T. Jordán concerning this abstract model. Then it will be shown how this proof gives rise to a polynomial-time algorithm.

# New Approximation Algorithms for Planar Augmentation

Petra Mutzel  
Max-Planck-Institut für Informatik

We consider new polynomial time approximation algorithms for the planar augmentation problem. The planar augmentation problem is the problem of adding a minimum number of edges to a given planar graph such that the resulting graph is biconnected and still planar. This problem is  $\mathcal{NP}$ -hard. We first discuss a  $\frac{5}{3}$ -approximation algorithm (presented at SODA '98) for the planar augmentation problem. In our extensive computational experiments on a benchmark set of 1100 graphs our algorithm found optimal solutions for around 95% of our instances.

Very recently, we could improve the approximation factor to  $\frac{3}{2}$ . The new algorithm runs in two phases. In the first phase we run a modified version of our  $\frac{5}{3}$ -approximative algorithm, then fix an embedding  $\Pi$  of the resulting graph and delete the newly added edges again. In the second phase, we solve the planar augmentation problem with respect to the fixed embedding. We show, how to solve this problem to optimality in linear time. Furthermore, we give a brief sketch of the analysis of the  $\frac{3}{2}$ -approximation algorithm and show an example in which the bound is tight.

This is joint work with Sergej Fialko.

## Efficiency of an Approximation Algorithm for Set Cover based on Semi-Local Improvement

Takao Asano  
Department of Information and System Engineering  
Chuo University

The best known approximation algorithm for 3-Set Cover is the  $\frac{4}{3}$ -approximation algorithm proposed by Duh and Fürer based on the semi-local improvements. There are, however, several unsolved issues about determining the possibility of semi-local improvements and, if so, about performing them. We show an algorithm for determining whether we can do semi-local

improvements and performing these improvements. We also analyse the time complexity of our modified version of Duh and Fürer's algorithm.

Original Duh and Fürer's algorithm takes  $O(|\mathcal{C}|^2|V|^{4.5})$  time where  $V$  is the set of elements and  $\mathcal{C}$  is a collection of subsets of  $V$  with  $\bigcup_{S \in \mathcal{C}} S = V$ . Our modified version runs in  $O(|\mathcal{C}|^2|V|^2)$  time. Our technique can also be applied to Duh and Fürer's algorithm for general  $k$ -Set Cover with performance ratio  $H_k - \frac{1}{2}$ , and our modified version runs in  $O(|\mathcal{C}| + |\mathcal{C}_3|^2|V|^2)$  time, where  $\mathcal{C}_3$  is the set of the subsets of cardinality 3 in  $\mathcal{C}$ .

This is joint work with Yoshiko Yamashita.

## Dynamic Orthogonal Graph Layout

Ulrik Brandes  
Fakultät für Mathematik und Informatik  
Universität Konstanz

In orthogonal graph drawing, edges are represented by sequences of horizontal and vertical straight line segments. On one hand, the number of bends displayed is an important measure of layout quality. On the other hand, when given a dynamic graph, one has to take into account not only the static criteria of layout quality, but also the effort users spend to regain familiarity with the layout. Therefore, consecutive layouts should compromise between quality and stability. We here present an extension for approaches based on the network flow model introduced by Tamassia in 1987. It allows to specify the relative importance of the number of bends vs. the number of changes between consecutive layouts. We show that optimal layouts in the dynamic model can be computed efficiently by means that are very similar to the static case, namely by solving a minimum cost flow problem in a residual network with additional costs on the arcs.

This is joint work with Dorothea Wagner.

# Compaction and Bend-saving Strategies in Orthogonal Drawings

Uli Fößmeier  
Tom Sawyer Software

Orthogonal drawings of graphs are widely investigated in the literature and many algorithms were presented to compute such drawings. Most of these algorithms lead to unpleasant drawings with many bends and a large area. We present methods how to improve the quality of given orthogonal drawings. Our algorithms try to simulate the thinking of a human spectator in order to achieve good results. We also give instructions how to implement the strategies in a way that a good runtime performance can be achieved.

## Area Efficient Layouts of the Batcher Sorting Networks

Shimon Even  
Department of Computer Science  
Technion, Haifa

The grid-area required by a sorting net for input vectors of length  $N$  is shown to be at least  $(N - 1)^2/2$ . Of all sorting nets which use  $o(N^2)$  comparators, the sorting nets of Batcher have been known to have a layout of  $O(N^2)$ , but the hidden constant factors have not been investigated. A straightforward use of known techniques leads to a layout of grid-area  $20.25N^2$ .

New grid layouts of the Batcher sorting nets are presented. In the case of the bitonic net the grid-area is less than  $4N^2$ , and in the case of the odd-even net the grid-area is less than  $3N^2$ . Slanted lines are used and the layouts have no knock-knees.



## Carrying an Umbrella: An Online Game on a Graph

Kyung-Yong Chwa  
Department of Computer Science  
KAIST

We introduce and study an online problem on a graph, which we call *Carrying an Umbrella*. We give a necessary and sufficient condition under which a competitive algorithm exists.

This is joint work with Jae-Ha Lee and Chong-Dae Park

## Graph Multidrawing: Finding Nice Drawings without Defining Nice

Sue Whitesides  
School of Computer Science  
McGill University

We present a *multidrawing* approach to graph drawing, which calls for systematically producing many drawing of the same graph. This contrasts with typical current graph drawing systems, which produce only one drawing at a time. The multidrawing approach addresses a fundamental problem in graph drawing, namely, how to avoid requiring the user to specify formally and precisely all the characteristics of a single “nice” drawing. We present a proof-of-concept implementation that produces a diverse collection of symmetric-looking drawings for small graphs.

This is joint work with Therese Biedl, Joe Marks and Kathy Ryall.

# Area Efficient Orthogonal Graph Drawing with Constraints

Roland Wiese  
Wilhelm-Schickard-Institut für Informatik  
Universität Tübingen

Making graph algorithms aware of constraints is a very important task. We incorporate several kinds of constraints into the area efficient approach for orthogonal drawings by Biedl/Kaufmann (ESA'97).

First we consider the behaviour of the algorithm in respect to the aspect ratio of the drawn nodes. With the original algorithm the aspect ratio of  $v$  can get as bad as  $2 : deg(v)$ . In the original paper, the authors already gave an idea how to achieve an aspect ratio of 1:2, based on Eulerian paths. We improve this scheme so that it works very satisfying in practice. A flexible approach based on edge flipping heuristics provides even better aspect ratios and keeps the size of drawing area close to the good bounds of the unconstrained algorithm.

Then, we look at edge constraints, where the edges have to be directed in a certain direction. There the row and column assignment in the basic scheme is done such that the constraints for the edge directions are fulfilled. But again we lose the balancing of the edges so that the area bound is even quadrupled. We propose an alternative approach that works especially good for one-dimensional constraints.

This is joint work with Michael Kaufmann.

## A Unified Approach to Labeling Graphical Features

Ioannis G. Tollis  
Computer Science Department  
University of Texas at Dallas

The automatic placement of text or symbol labels corresponding to graphical objects is critical in several application areas such as Cartography, Geographical Information Systems, and Graph Drawing. In this talk we present a

general framework for solving the problem of assigning text or symbol labels to a set of graphical features in two dimensional drawings or maps. Our approach does not favor the labeling of one type of graphical feature (such as a node, edge, or area) over another. Additionally, the labels are allowed to have arbitrary size and orientation. We have applied our framework to drawings of graphs. We have implemented our techniques and have performed extensive experimentation on several styles of drawings of graphs. The resulting label assignments are very practical and indicate the effectiveness of our approach.

This is joint work with Konstantinos G. Kaoulis.

## Different Fields of Application for Graph Drawing

Dorothea Wagner  
Fakultät für Mathematik und Informatik  
Universität Konstanz

We report on two examples for application driven research in Graph drawing. The first problem arises in traffic engineering:

$G = (V, E)$  is the graph induced by train interconnection data as follows. The vertices of  $G$  correspond to the stations, and an edge  $(u, v)$  is contained in  $G$  iff there is a train from  $u$  to  $v$  without an intermediate stop.  $G$  contains two kinds of edges, “direct” edges corresponding to local trains and “transitive” edges corresponding to trains not stopping at every station. We want to draw  $G$  such that these two kinds of edges may be distinguished.

The second problem arises in social network analysis:

Each vertex in a social network has a certain level of centrality. We want to draw a network such that it is easy to realize the centrality of the vertices.

For both problems solution methods are proposed that are based on a random field model.

This is joint work with Ulrik Brandes.

## TreVis — A Tool to Visualize Parallel Processes

Michael Kaufmann  
Wilhelm-Schickard-Institut für Informatik  
Universität Tübingen

Measuring and evaluating the runtime of parallel programs is a difficult task. In this talk we present tools for performance evaluation and visualization in the distributed thread system (DTS), a programming environment for portable parallel applications. We describe a novel layout algorithm which has been tailored to expose the structure of multithreaded applications and show several examples using the graph drawing system **GraVis**.

This is joint work with Björn Steckelbach, Uli Fößmeier, Marcus Ritt, Till Bubeck, and Wolfgang Rosenstiel.

## Distributive Lattices on Planar Graphs

Hubert de Fraysseix  
Centre d'Analyse et de Mathématiques  
EHESS

Numerous problems gathering topological and combinatorial aspects find a natural solution using constrained orientations. So are planar augmentations, representations and planarity testing problems. These orientations have a distributive lattice structure and we show that they are in bijection with other combinatorial objects such as tree decompositions, shelling orders, bipolar orientations, *st*-orders etc. We show how the existence of positive cocircuits in planar triangulations or angle graphs is related to connectivity properties.

This is joint work with P. Ossona de Mendez.

## Planarity Revisited

Christoph Hundack  
Max-Planck-Institut für Informatik

Based on a new decomposition technique for graphs we present a simplified approach towards planarity testing and simultaneous embedding. A DFS-based preprocessing splits up the graph into disjoint level-components and generates a partial embedding of the graph. Step by step this partial embedding is enlarged by some level-components with the planarity test reducing to solving an easy bipartition problem. The decomposition technique can be applied to simplify or accelerate other graph algorithms.

This is joint work with Kurt Mehlhorn, Hermann-Stamm-Wilbrandt.

## Survivability in Networks: Computing all the Diameter Minimizing Swaps in a Tree

Peter Widmayer  
Institut für Theoretische Informatik  
Eidgenössische Technische Hochschule Zürich

With the arrival of fiber optic communication networks, these networks become sparser than wire meshes, and therefore more susceptible to failures of components. We discuss the problem of finding best possible replacement edges for the edges of a (minimum diameter) tree that fail. The tree serves to support a broadcast, and the pre-computation of all best replacement edges (swaps) serves to let the network react quickly on failures. We show how for a given graph  $G = (V, E)$  with  $|V| = n$  and  $|E| = m$  and a given spanning tree of this graph, all best swaps can be found in time  $O(n\sqrt{m})$ , using  $O(m+n)$  preprocessing and  $O(n)$  space.

This is joint work with Enrico Nardelli and Guido Proietti.

# An Optimal Layout of the Butterfly Interconnection Network

Maria Zapolotsky  
Department of Computer Science  
Technion, Haifa

We establish upper and lower bounds on the layout area of the Butterfly network, which differ only in low-order terms. Specifically, the  $N$ -input,  $N$ -output Butterfly network can be laid out in area  $\frac{1}{2}N^2 + o(N^2)$ , while no layout of the network can have area smaller than  $\frac{1}{2}N^2 - o(N^2)$ . These results improve both the known upper bound and the known lower bound on the area of Butterfly network layouts.

This is joint work with Yefim Dinitz and Shimon Even.

## Hamiltonian Amalgams

Maciej M. Sysło  
Institute of Computer Science  
University of Wrocław

An amalgam is obtained from two Halin graphs having skirting cycles of the same length. We are interested in Hamiltonicity of amalgams constructed from two identical Halin graphs without any shift along the skirting cycle. We establish Hamiltonicity of amalgams constructed from cubic Halin graphs. We give a sufficient condition for Hamiltonicity of non-cubic amalgams and characterize infinite classes of non-Hamiltonian amalgams. We also characterize Hamiltonicity of amalgams constructed by shifting the component Halin graphs by one and of general amalgams of higher degree.

This is joint work with Arthur M. Farley, Andrej Proskurowski and Miroslava Skowrovicka.

## Open Problems

Chair: Roberto Tamassia

### Kazuo Iwama: Generating Non-Hamilton Graphs

Unsatisfiable CNF formulas can be generated nondeterministically using the idea of Resolution principle. Also, non-3-colorable graphs are generated by Hajos Calculus. Is there a similar generation system for non-Hamilton graphs?

### Hubert de Fraysseix and Patrice Ossona de Mendez:

#### Problem 1: Complexity of computing indegree bounded orientations for planar graphs

Is there a linear time algorithm to solve the following problem:

*Input:* A planar graph  $G = (V, E)$  and an integer valued function  $f$  on the vertex set of  $G$ .

*Output:* An orientation of  $G$  such that the indegree of any vertex  $x$  is at most  $f(x)$ , if such an orientation exist.

Remark: If  $G$  is not assumed to be planar, the problem may be solved in  $O(m \cdot n)$  time.

**Problem 2** A 3-uniform hypergraph  $H = (V, E)$  is a pair formed by a finite set  $V = \{v_1, \dots, v_n\}$  of vertices and a set  $E = \{e_1, \dots, e_m\}$  of triplets of vertices, the edges of the hypergraph. To a hypergraph is associated an incidence poset  $P(H)$ , which is the vertex to edge inclusion poset. The dimension  $\dim P$  of a poset  $P$  is the minimal number of total orders whose intersection is  $P$ .

A 3-uniform hypergraph  $H$  is “representable” in  $\mathbb{R}^3$  if there exists an injective mapping of the vertices in  $\mathbb{R}^3$ , such that the edges of  $H$  correspond to triangles forming a 2-complex: The triangles  $T_i$  and  $T_j$  (corresponding to edges  $e_i$  and  $e_j$ ) have an intersection which is

- empty if  $e_i$  and  $e_j$  are disjoint,
- $\{p_a\}$  if  $e_i$  and  $e_j$  have exactly one common element  $v_a$  mapped to the point  $p_a$ ,
- the segment  $[p_a, p_b]$  if  $e_i$  and  $e_j$  have exactly two common elements  $v_a$  and  $v_b$  mapped to the points  $p_a$  and  $p_b$ .

Questions:

1. What are the forbidden configurations for representable 3-uniform hypergraphs?  
(Example: the hypergraph on 6 vertices having all the possible triplets for edges is not representable, the same way that  $K_5$  is not representable in the plane.)
2. Is it true that  $\dim(H) \leq 4$  implies that  $H$  is representable?  
(For a graph  $G$ , that is for a 2-uniform hypergraph,  $\dim G \leq 3$  iff  $G$  is planar, as proved by Schnyder.)

**Franz J. Brandenburg and Arunabha Sen: Approximations of GRAPH BIN PACKING**

*Instance:* A graph  $G = (V, E)$ , a size  $s(v)$  for each vertex and a width  $w(e)$  for each edge. Bins of capacity  $S$  for the vertices and  $W$  for the edges.

*Question:* Minimize the number of bins such that the sum of the sizes of the vertices for each bin does not exceed  $S$  and such that the sum of the widths of the edges from the vertices in a bin to vertices in other bins does not exceed  $W$ .

Hence, only edges between two bins are charged; the inner bin edges are for free.

Is there an  $\epsilon$ -approximation for GRAPH BIN PACKING and how small is  $\epsilon$ ?

**Dagmar Handke: The Tree 3-Spanner Problem**

For any fixed parameter  $t \geq 1$ , a *tree  $t$ -spanner* of a graph  $G$  is a spanning tree  $T$  of  $G$  such that the distance between every pair of vertices in  $T$  is at most  $t$  times their distance in  $G$ . We are interested in the following problem:

*Input:* An unweighted graph  $G$  and a positive integer  $t$ .

*Problem:* Does  $G$  contain a tree  $t$ -spanner as a subgraph?

Cai and Corneil (SIAM J. Discrete Math., Vol. 8, 1995) have shown that the Tree-Spanner Problem is solvable in linear time for  $t = 1, 2$ , whereas it is  $\mathcal{NP}$ -complete for all  $t \geq 4$ .

*Question:* What is the time-complexity for  $t = 3$ ?



**Takao Asano**  
Chuo University  
Dept. of Information and  
System Engineering  
Bunkyo-ku  
112-8551 Tokyo  
Japan  
asano@ise.chuo-u.ac.jp  
phone: +81-3-3817-16 86

**Vladimir Batagelj**  
University of Ljubljana  
Dept. of Theoretical Computer Science  
IMPM  
Jadranska 19  
1111 Ljubljana  
SEU  
vladimir.batagelj@uni-lj.si  
phone: +386-61-1766-672

**Franz J. Brandenburg**  
Universität Passau  
Lehrstuhl für Theoretische Informatik  
Innstr. 33  
D-94032 Passau  
Germany  
brandenb@fmi.uni-passau.de  
phone: +49-851-509-30 30

**Ulrik Brandes**  
Universität Konstanz  
Fakultät für Mathematik u. Informatik  
Fach D 188  
D-78457 Konstanz  
Germany  
Ulrik.Brandes@uni-konstanz.de  
phone: +49-7531-88-4263

**Kyung-Yong Chwa**  
KAIST  
Computer Science  
373-1 Kusong-dong, Yusong-gu  
305-701 Taejon  
ROK  
kychwa@jupiter.kaist.ac.kr  
phone: +82-42-869-3513

**Gabriele Di Stefano**  
Università degli Studi di L'Aquila  
Dipartimento di Ingegneria Elettrica  
I-67040 Monteluco di Roio, L'Aquila  
Italy  
gabriele@infolab.ing.univaq.it  
phone: +39-862-434-416

**Yefim Dinitz**  
Technion Haifa  
Dept. of Computer Science  
IL-32000 Haifa  
Israel  
dinitz@cs.technion.ac.il  
phone: +972-4-8294-368

**Peter Eades**  
University of Newcastle  
Dept. of Computer Science  
University Drive  
NSW 2308 Callaghan  
Australia  
eades@cs.newcastle.edu.au  
phone: +61-49-216034

Shimon **Even**  
Technion Haifa  
Dept. of Computer Science  
IL-32000 Haifa  
Israel  
even@cs.technion.ac.il  
phone: +972-4-829-4166

Uli **Fößmeier**  
Tom Sawyer Software  
804 Hearst Avenue  
CA 94710 Berkeley  
USA  
foessmei@tomsawyer.com

Andras **Frank**  
Eötvös Lorand University Budapest  
Dept. of Operations Research  
Muzeum Krt. 6-8  
H-1088 Budapest  
Hungary  
frank@cs.elte.hu  
phone: +361-266-98 33

Hubert **de Fraysseix**  
EHESS  
Centre d'Analyse et de Mathématiques  
54 Boulevard Raspail  
F-75270 Paris  
France  
hf@chess.fr

Emden **Gansner**  
AT&T Labs-Research  
180 Park Avenue  
P.O. Box 971  
NJ 07932-0971 Florham Park  
USA  
erg@research.att.com  
phone: +1-973-360-8646

Ashim **Garg**  
SUNY at Buffalo  
Dept. of Computer Science  
226 Bell Hall  
NY 14260 Buffalo  
USA  
agarg@cs.buffalo.edu  
phone: +1-716-645-3180

Dagmar **Handke**  
Universität Konstanz  
Fakultät für Mathematik u. Informatik  
Fach D 188  
D-78457 Konstanz  
Germany  
Dagmar.Handke@uni-konstanz.de  
phone: +49-7531-88 40 15

Michael **Himsolt**  
Universität Passau  
FB Informatik  
D-94030 Passau  
Germany  
himsolt@fmi.uni-passau.de  
phone: +49-89-509-3034

**Seokhee Hong**

University of Newcastle  
Dept. of Electrical and  
Computer Engineering  
University Drive  
NSW 2308 Callaghan  
Australia  
shhong@cs.newcastle.edu.au  
phone: +61-2-4921-7242

**Giuseppe Liotta**

Università di Roma "La Sapienza"  
Dipartimento di Informatica  
e Sistemistica  
Via Salaria 113  
I-00198 Roma  
Italy  
liotta@dis.uniroma1.it  
phone: +39-6-88 41-923

**Christoph Hundack**

Max-Planck-Institut für Informatik  
Im Stadtwald  
D-66123 Saarbrücken  
Germany  
hundack@mpi-sb.mpg.de

**Ernst W. Mayr**

Technische Universität München  
Institut für Informatik  
Arcisstr. 21  
D-80290 München  
Germany  
mayr@informatik.tu-muenchen.de  
phone: +49-89-289-2 26 80 /-2 26 81

**Kazuo Iwama**

Kyoto University  
Dept. of Information Science  
Sakyo  
606 Kyoto  
Japan  
iwama@kuis.kyoto-u.ac.jp  
phone: +81-75-753-5372

**Kurt Mehlhorn**

Max-Planck-Institut für Informatik  
Im Stadtwald  
D-66123 Saarbrücken  
Germany  
mehlhorn@mpi-sb.mpg.de  
phone: +49-681-9325 100

**Michael Kaufmann**

Universität Tübingen  
Wilhelm-Schickard-Institut für  
Informatik  
Sand 13  
D-72076 Tübingen  
Germany  
mk@informatik.uni-tuebingen.de  
phone: +49-7071-297404

**Petra Mutzel**

Max-Planck-Institut für Informatik  
Im Stadtwald  
D-66123 Saarbrücken  
Germany  
mutzel@mpi-sb.mpg.de  
phone: +49-681-9325-105

**Matthias Müller-Hannemann**  
Technische Universität Berlin  
FB 3 Mathematik  
Skr. MA 6-1  
Str. des 17. Juni 136  
D-10623 Berlin  
Germany  
mhannema@math.tu-berlin.de  
phone: +49-30-314-25181

**Stefan Näher**  
Universität Halle  
FB Mathematik u. Informatik  
Kurt-Mothes-Str. 1  
D-06099 Halle Germany  
naeher@informatik.uni-halle.de  
phone: +49-345-552-47 12

**Takao Nishizeki**  
Tohoku University  
Graduate School of  
Information Sciences  
Sendai 480-77  
Japan  
nishi@ecei.tohoku.ac.jp  
phone: +81-22-263-9301

**Hartmut Noltemeier**  
Universität Würzburg  
Lehrstühle für Informatik  
LST I  
Am Hubland  
D-97074 Würzburg  
Germany  
noltemei@informatik.uni-wuerzburg.de  
phone: +49-931-888 5054 (5055)

**Patrice Osona de Mendez**  
EHES  
Centre d'Analyse et de Mathématiques  
54 Boulevard Raspail  
F-75270 Paris  
France  
pom@ehess.fr

**Andras Recski**  
Technical University of Budapest  
Faculty of Elec. Eng. and Information  
H-1521 Budapest  
Hungary  
recski@math.bme.hu  
phone: +36-1-463-2585

**Heike Ripphausen-Lipa**  
Lufthansa Systems Berlin GmbH  
Fritschestr. 27-28  
D-10585 Berlin  
Germany  
rip@lsb.de  
phone: +49-30-34007-182

**Martin Skutella**  
Technische Universität Berlin  
FB 3 Mathematik  
Skr. MA 6-1  
Str. des 17. Juni 136  
D-10623 Berlin  
Germany  
skutella@math.tu-berlin.de  
phone: +49-30-314-25747

Angelika **Steger**  
Technische Universität München  
Institut für Informatik  
Arcisstr. 21  
D-80290 München  
Germany  
steger@informatik.tu-muenchen.de  
phone: +49-89-289-2 84 73

Maciej M. **Sysło**  
University of Wrocław  
Institute of Computer Science  
Przesmyckiego 20  
PL-51-151 Wrocław  
Poland  
syslo@ii.uni.wroc.pl  
phone: +48-71-324-7382

Roberto **Tamassia**  
Brown University  
Department of Computer Science  
P.O. Box 1910  
RI 02912 Providence  
USA  
rt@cs.brown.edu  
phone: +1-401-863-7639

Ronit **Teplixke**  
Technion – Haifa  
Department of Computer Science  
IL-32000 Haifa  
Israel  
ronitt@cs.technion.ac.il

Ioannis **Tollis**  
University of Texas at Dallas  
Computer Science Department  
EC 31  
P.O. Box 830688  
Richardson TX 75083-0688  
USA  
tollis@utdallas.edu  
phone: +1-972-883 2180

Paola **Vocca**  
Università di Roma “Tor Vergata”  
Dipartimento di Matematica  
Via della Ricerca Scientifica  
I-00133 Roma  
Italy  
vocca@axp.mat.uniroma2.it  
phone: +39-6-7259-4631

Dorothea **Wagner**  
Universität Konstanz  
Fakultät für Mathematik u. Informatik  
Fach D188  
D-78457 Konstanz  
Germany  
Dorothea.Wagner@uni-konstanz.de  
phone: +49-7531-88-2893

Frank **Wagner**  
Freie Universität Berlin  
Fachbereich Mathematik u. Informatik  
Takustr. 9  
D-14195 Berlin  
Germany  
wagner@inf.fu-berlin.de  
phone: +49-30-838-75159/75103

Karsten **Weihe**  
Universität Konstanz  
Fakultät für Mathematik u. Informatik  
Fach D188  
D-78457 Konstanz  
Germany  
karsten.weihe@uni-konstanz.de  
phone: +49-7531-88-4375

Gerhard **Woeginger**  
Technische Universität Graz  
Institut für Mathematik  
Steyrergasse 30  
A-8010 Graz  
Austria  
gwoegi@opt.math.tu-graz.ac.at  
phone: +43-316-873-53 57

Sue **Whitesides**  
McGill University  
School of Computer Science  
3480 University Street  
Montreal PQ H3A 2A7  
Canada  
sue@cs.mcgill.ca  
phone: +1-514-398-7071

Maria **Zapolotsky**  
Technion – Haifa  
Department of Computer Science  
IL-32000 Haifa  
Israel  
zmaria@cs.technion.ac.il

Peter **Widmayer**  
ETH Zürich  
Institut für Theoretische Informatik  
CH-8092 Zürich  
widmayer@inf.ethz.ch  
phone: +41-1-632-74 00

Christos **Zaroliagis**  
University of London  
King's College  
Dept. of Computer Science  
Strand  
WC2R 2LS London  
Great Britain  
zaro@dcs.kcl.ac.uk  
phone: +44-171-873-2807

Roland **Wiese**  
Universität Tübingen  
Wilhelm-Schickard-Institut  
für Informatik  
Sand 13  
D-72076 Tübingen  
Germany  
wiese@informatik.uni-tuebingen.de