Graph Algorithms and Applications

(Dagstuhl–Seminar 9620)

Organizers:

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

May 13 - 17, 1996

In many fields of applications *graphs* or *networks* play an important role for understanding a concrete situation and modelling problems. The development of algorithms for communication problems, traffic optimization, scheduling or VLSI-design is for example based on graphs. Algorithmic graph theory is a classical area of research by now and has been rapidly expanding during the last three decades. Especially, the interplay between theory and application gave research in this area again and again new impetus.

While the complexity of today's "real world" problems is increasing the design of "sophisticated" algorithms is an ambitious task. Numerous newly invented problems coming from new fields of applications have attracted the attention of researchers in algorithmic graph theory. But, in recent years also classical problems were studied again. For several fundamental graph problems efficient and simple algorithms were designed in view of the computer implementation of such algorithms.

This seminar was intended to bring together researchers from different areas in algorithmic graph theory. Particular emphasis was placed on applications, experimental research and aspects of the implementation of graph algorithms. The participants had the opportunity to exchange ideas and discuss new trends in algorithmic graph theory. Main topics of interest were *efficient graph algorithms*, *graph drawing*, *algorithm animation with graphs, implementation of graph algorithms* and *applications in VLSI-design, traffic optimization*, and *CAD*.

We had 36 participants from different European countries, Australia, USA, Canada, Japan and Taiwan. During the workshop 28 lectures have been presented and two software demonstrations. There was also an open-problem-session on Tuesday evening and a lively discussion on problems from different fields of application. Schloß Dagstuhl and its staff provided a very convenient and stimulating environment. All participants appreciated the cordial atmosphere. The organizers wish to thank all those who helped make the workshop a fruitful research experience.

Participants

Takao Asano, Chuo University
Tetsuo Asano, Osaka Electro-Communication University
Franz J. Brandenburg, Universität Passau
Ulrik Brandes, Universität Konstanz
Yefim Dinitz, Technion — Haifa
Peter Eades, University of Newcastle
Wen-Lian Hsu, Academica Sinica Taipei
Toshihide Ibaraki, Kyoto University
Michael Jünger, Universität zu Köln
Goos Kant, Utrecht University
Naoki Katoh, Kobe University of Commerce
Michael Kaufmann, Universität Tübingen
Philip Klein, Brown University
Bernhard Korte, Universität Bonn
Han La Poutré, Leiden University
Annegret Liebers, Universität Konstanz
Giuseppe Liotta, Brown University
Rolf Möhring, Technische Universität Berlin
David W. Matula, Southern Methodist University
Kurt Mehlhorn, Max-Planck-Institut für Informatik
Bojan Mohar, University of Ljubljana
Petra Mutzel, Max-Planck-Institut für Informatik
Stefan Näher, Universität Halle
Hiroshi Nagamochi, Kyoto University
Takao Nishizeki, Tohoku University
Stephen North, AT&T Bell Labs
Andras Recski, Technical University of Budapest
Maciej M. Sysło, University of Wroclaw
Roberto Tamassia, Brown University
Ioannis Tollis, University of Texas at Dallas
Dorothea Wagner, Universität Konstanz
Frank Wagner, Freie Universität Berlin
Toshimasa Watanabe, Hiroshima University
Karsten Weihe, Universität Konstanz
Sue Whitesides, McGill University
Christos Zaroliagis, Max-Planck-Institut für Informatik

Program

Monday, May 13, 1996

Morning Session	Chair: Dorothea Wagner
9:00	Welcome
9:15 – 10:00	Takao Nishizeki: Decompositions to degree-constrained subgraphs are simply reducible to edge-colorings
10:30 – 11:00	Hiroshi Nagamochi: Faster edge-splitting algorithms in undirected graphs
11:15 – 11:45	Takao Asano: Approximation algroithms for the maximum satisfiability problem
Afternoon Session	Chair: Peter Eades
14:45 – 15.30	Wen-Lian Hsu: Graph recognition algorithms
16:15 – 16:45	Kurt Mehlhorn: A simple linear time algorithm to find Kuratowski subgraphs of non-planar graphs
16:45 – 17:30	Bojan Mohar: Embedding graphs in an arbitrary surface in linear time
17:30 – 18:00	Sue Whitesides: Geometric representations of graphs

Tuesday, May 14, 1996

Morning Session	Chair: Michael Jünger
9:00 – 9:30	Karsten Weihe: Reconstructing a surface from its polygonal pieces
9:30 – 10:00	Rolf Möhring: Mesh generation using flow methods
10:30 – 10:50	Tetsuo Asano: Space efficient algorithms for image segmentation
10:50 – 11:20	Goos Kant: Vehicle routing problems with time windows
11:30 – 12:00	Yefim Dinitz: Modelling all bottlenecks (edge cuts) of a network, with incremental maintenance
Afternoon Session	Chair: Takao Nishizeki
14:45 – 15.30	Philip Klein: Approximation algorithms for semidefinite programs arising from MAX CUT and COLORING
16:15 – 16:45	Naoki Katoh: Computing subgraphs of minimum weight triangulations based on LMT-skeletons
17:00	Application Problems Chair: Ioannis Tollis
19:30	Open Problems Chair: Roberto Tamassia

Wednesday, May 15, 1996

Morning Session	Chair: Stephen North
9:00 – 9.30	Franz J. Brandenburg: On drawing planar angle graphs
9:30 – 10:00	Peter Eades: Drawing clustered graphs
10:30 – 11:00	Ioannis Tollis: Advances in orthogonal graph drawing
11:00 – 11:45	Michael Kaufmann: On bend-minimum orthogonal graph drawings
Afternoon	Excursion to Trier or hike in the rain

Thursday, May 16, 1996

Morning Session	Chair: Toshimasa Watanabe
9:00 – 9.30	Stefan Näher: Implementation of graph algorithms with LEDA
9:30 – 10:00	Giuseppe Liotta: The proximity drawability problem
10:30 – 11:00	Toshihide Ibaraki: Two arc disjoint paths in Eulerian digraphs
11:00 – 11:45	David W. Matula: Two results on search and edge connectivity
Afternoon Session	Chair: Sue Whitesides
15:00 – 15.30	Michael Jünger: Practical performance of MIN CUT algorithms
16:15 – 17:00	Petra Mutzel: Optimization on hierarchical graphs
17:00 – 17:30	Maciej Sysło: The bandwith problem of clique caterpillars and tolerance graphs
17:30 – 17:45	Dorothea Wagner: PlaNet: A demonstration package for algorithms on PLAnar NETworks
17:45 – 18:00	David W. Matula: Painting graph algorithms
Evening	Software Demonstrations

Friday, May 17, 1996

Morning Session	Chair: Frank Wagner
9:00 – 9.30	Bernhard Korte: L_1-Steiner trees
9:30 – 10:10	Christos Zaroliagis: All-pairs MIN CUT in sparse networks
10:45 – 11:30	Andras Recski: Some linear time solvable subcases of the multilayer routing problem

Decompositions to degree-constrained subgraphs are simply reducible to edge-colorings

Takao Nishizeki
Graduate School of Information Sciences
Tohoku University

The degree-constrained subgraphs decomposition problem, such as an f-coloring, f-factorization and $[g, f]$-factorization, is to decompose a given graph $G = (V, E)$ to edge-disjoint subgraphs degree-constrained by integer-valued functions f and g on V. In this talk we show that the problem can be simply reduced to the edge-coloring problem in polynomial-time. That is, for any positive integer k, we give a polynomial-time transformation of G to a new graph such that G can be decomposed to at most k degree-constrained subgraphs if and only if the new graph can be edge-colored with k colors.

Faster edge-splitting algorithms in undirected graphs

Hiroshi Nagamochi
Graduate School of Engineering
Kyoto University

This paper presents a deterministic $O(n(m + n \log n) \log n) = \tilde{O}(nm)$ time algorithm for splitting off all edges incident to a vertex s of even degree in a multigraph G, where n and m are the numbers of vertices and links (= vertex pairs between which G has an edge) in G, respectively. Based on this, many graph algorithms using edge-splitting can run faster. For example, the edge-connectivity augmentation problem in an undirected multigraph can be solved in $\tilde{O}(nm)$ time, which is an improvement over the previously known randomized $\tilde{O}(n^3)$ bound and deterministic $\tilde{O}(n^2m)$ bound.

Approximation algorithms for the maximum satisfiability problem

Takao Asano
Department of Information and System Engineering
Chuo University

The maximum satisfiability problem (MAX SAT) is : given a set of clauses with weights, find a truth assignment that maximizes the sum of the weights of the satisfied clauses.

In this talk, we present approximation algorithms for MAX SAT, including a 0.76544-approximation algorithm. The previous best approximation algorithm for MAX SAT was proposed by Goemans-Williamson and has a performance guarantee of 0.7584. Our algorithms are based on semidefinite programming and the 0.75-approximation algorithms of Yannakakis and Goemans-Williamson.

Graph recognition algorithms

Wen-Lian Hsu
Institute of Information Science
Academica Sinica Taipei

We shall give a brief survey of our recent recognition algorithms for planar graphs, interval graphs and the consecutive ones property. We shall also talk about algorithms that can tolerate a small percentage of errors in the input data.

A simple linear time algorithm to find Kuratowski subgraphs of non-planar graphs

Kurt Mehlhorn
Max-Planck-Institut für Informatik

We extend the Lempel-Even-Cederbaum planarity test such that it yields Kuratowski subgraphs of non-planar graphs. The extension runs in linear time.

This is joint work with Christoph Hundack and Stefan Näher.

Embedding graphs in an arbitrary surface in linear time

Bojan Mohar
Department of Mathematics
University of Ljubljana

For an arbitrary fixed surface S, a linear time algorithm is presented that for a given graph G either finds an embedding of G in S or identifies a subgraph of G that is homeomorphic to a minimal forbidden subgraph for embeddability in S. A side result of the proof of the algorithm is that minimal forbidden subgraphs for embeddability in S cannot be arbitrarily large. This yields a constructive proof of the result of Robertson and Seymour that for each closed surface there are only finitely many minimal forbidden subgraphs. The results and methods of this work can be used to solve more general embedding extension problems.

Geometric representations of graphs

Sue Whitesides
School of Computer Science
McGill University

Traditionally we represent graphs in a visual way by mapping vertices to points and edges to curves. Graphs may also be represented geometrically, for example as intersection or contact graphs, as proximity graphs, as visibility graphs.

We survey several results on a 3-dimensional visibility representation that maps vertices to polygons floating parallel to the xy-plane and edges to visibility lines parallel to the z-axis.

Reconstructing a surface from its polygonal pieces

Karsten Weihe
Fakultät für Mathematik und Informatik
Universität Konstanz

In the computer aided design of vehicles, engines, and machines of all kinds, the surface of a workpiece is usually modeled by a set of openly disjoint polygons in the three–dimensional space. These polygons do not fit exactly together, but incident polygons are only placed (more or less) close to each other. The input describes solely the geometries of all polygons, and the neighborhoods are not part of the input. These neighborhoods must be computed by CAD packages automatically, and the results are by far not sufficient.

In this talk we present an algorithm that seems to yield better results. In contrast to the other approaches that we know of, we abstract from numerical and geometrical aspects as far as possible and focus on structural aspects.

This is joint work with Thomas Willhalm.

Mesh generation using flow methods

Rolf Möhring
Fachbereich Mathematik
Technische Universität Berlin

Network flow techniques are applied to a problem arising in the computer aided design of cars, planes, ships or components of them: Refine a coarse mesh of spheric polygons that approximates the surface of a workpiece such that the resulting mesh is suitable for a numerical analysis (in particular, we are asked to achieve a specified mesh density and to generate meshes consisting of conforming quadrilaterals only).

This turns out to be a difficult discrete problem (strongly $\mathcal{NP}$-hard). We show how to formulate it as a two-step problem consisting of a choice among refinement templates and a network flow problem to realize a refinement for the given template choice. More specifically, the network flow problem is a bidirected flow problem on an undirected graph G with upper and lower capacities on the edges and some additional node balance conditions. For a given choice of templates, the problem is then reduced to finding a feasible flow in that graph that satisfies all these constraints. We describe this model and a generic refinement algorithm based on it, and discuss the encouraging results of a first implementation.

This is joint work with Matthias Müller-Hannemann and Karsten Weihe.

Space efficient algorithms for image segmentation

Tetsuo Asano
Department of Engineering Informatics
Osaka Electro-Communication University

This paper studies the space complexity of a traditional algorithm for region segmentation which first grows homogeneous regions and then builds its associated region adjancy graph which reflects a structure of segmentation results. A naive method in which the region number is kept at each pixel needs $O(n \log k)$ bits in total, where n and k are numbers of pixels and regions obtained, respectively. Application of on-line graph coloring techniques can save the space complexity of the region growing process to $\Theta(\log \log k)$ bits per pixel with some difficulty to build a region adjacency graph. Finally, we demonstrate the advantage of a geometric approach which requires only some constant (4 or 5) bits per pixel for region growing with reasonable time for establishing region adjacency under some assumption. We also study several algorithmic issues associated with the implementation of a region growing algortihm.

Vehicle routing problems with time windows

Goos Kant
Department of Computer Science
Utrecht University

In the Vehicle Routing Problem with Time Windows (VRPTW) there are 2 basic heuristic algorithms: the sequential insertion algorithm and the Savings Method. First I gave a brief introduction of this field.

Then I explained different data structures for implementing the Savings Method. The matrix, (partial) heap, (recursive) grid and array implementation are explained, and time and memory requirements are given. From experimental evaluation the array implementation turned out to be most efficient in practice. In the last part of the talk I presented an algorithm for computing the shortest duration of a given tour. This algorithm is based on computational geometry and can be used as a subprocedure in insertion and savings methods. It is also explained (and experimentally evaluated) that storing shortest path information at nodes really helps for dynamic testing of savings and insertion costs.

This is joint work with A. van Vliet and C. de Jong.

Modelling all bottlenecks (edge cuts) of a network, with incremental maintenance

Yefim Dinitz
Departement of Computer Science
Technion, Haifa

Assume we worry about the (near) minimum edge cuts of a network. Those can be traffic bottlenecks or most probable communication disconnections due to link failures. In various cases, there can be a polynomial or exponential number of such bottlenecks. The first question: can one be shown all the bottlenecks in a single picture (model), which size is linear in the number of nodes, with simple visual rules to identify bottlenecks? Now assume we add an edge to the network; clearly, exactly the bottlenecks affected change their status. The second question: can the picture discussed, if exists, visually support those dynamics for an edge insertion? for several insertions? We answer these questions positively, more or less, for the cases of the minimum cuts, of the minimum and subminimum cuts, and of the cuts that are minimum among dividing a set of poles (distinguished nodes) of the network. The main graph types used for modeling are cactus tree and, for the third case, also DAG. Open questions: How to draw adequately the models presented? When can we keep, in the picture, certain relations to geographic positions of nodes? Can such a drawing be maintained undergoing discussed dynamics and remain recognizable?

Approximation algorithms for semidefinite programs arising from MAX CUT and COLORING

Philip Klein
Department of Computer Science
Brown University

Linear programming has proved a useful tool in the design of approximation algorithm. Goemans and Williamson showed (1994) that a more general kind of mathematical programming, namely semidefinite programming, is similarly useful; they gave an approximation algorithms for MAX CUT that depends on solving a semidefinite program. Karger, Motwani, and Sudan then showed (1994) a similar approach yields an approximation algorithm for graph coloring. In each of these algorithms, the computational bottleneck is solving the semidefinite program. We show that for constant ϵ a $(1+\epsilon)$-approximate solution to these programs can be obtained in $O(nm \log^3 n)$, where n is the number of nodes and m is the number of edges.

This is joint work with Hsueh-I Lu.

Computing subgraphs of minimum weight triangulations based on LMT-skeletons

Naoki Katoh
Department of Management Science
Kobe University of Commerce

We present improvements in finding the LMT-skeleton, which is a subgraph of all minimum weight triangulation. Our improvements consist of: (1) A criterion is proposed to identify edges in all minimum weight triangulation (2) A faster algorithm is presented for performing one pass of the Dickenson and Montague method. (3) Improvements in the implementation that may lead to substantial space reduction for uniformly distributed point sets.

This is joint work with Manabu Sugai and Sin-Wing Chang.

On drawing planar angle graphs

Franz J. Brandenburg
Lehrstuhl für Theoretische Informatik
Universität Passau

Recently, some advanced algorithms for straight line drawings of planar graphs have been introduced. However, these algorithms produce drawings with many small angles. Their drawings are not aesthetically nice.

Our goal is to resolve these problems by a beautifier. We consider the angles of planar straight line drawings. First, we compute maximal angles by repeatedly solving a linear program. However, the existence of a planar straight line drawing with these angles is not guaranteed. We approach drawings either by a linear program for the edge length or by a new spring embedder, which emphasizes on angles.

Our experiments are very promising. The approach outperforms other advanced algorithms for straight line drawings and unfolds drawings with too many small angles.

This is joint work with H. Steine and A. Stübinger.

Drawing clustered graphs

Peter Eades
Department of Computer Science
University of Newcastle

We show that every c-planar clustered graph has a straight-line c-planar drawing. To prove this result for clustered graphs we prove a similar result about hierarchical graphs.

This is joint work with Q.W. Feng.

Advances in orthogonal graph drawing

Ioannis Tollis
Computer Science Department
University of Texas at Dallas

An orthogonal drawing of a graph is a drawing such that nodes are placed on grid points and edges are drawn as sequences of vertical and horizontal segments. In this talk we present the following results:

(1) An algorithm for constructing orthogonal drawings with at most $2n + 2$ bends and $0.76n^2$ area.

(2) Interactive scenaria for constructing orthogonal drawings such that the drawing remains almost unchanged after an insertion or deletion of an edge or vertex.

The algorithms run in linear time on graphs with n vertices and of maximum degree four.

This is joint work with Achilleas Papakostas.

On bend-minimum orthogonal graph drawings

Michael Kaufmann
Institut für Informatik
Universität Tübingen

In this talk we present old and new techniques to minimize the number of bends in an orthogonal drawing for a given embedded planar graph. First we review Tamassia's algorithm which solves the problem for 4-planar graphs. Then we discuss how to get a suitable model for orthogonal drawings of vertices of higher degree. We adopt a model used in a drawing of a graph from astrophysics and present an extension of Tamassia's network flow method that achieves the optimum while keeping the vertices small. Finally we discuss two approaches for 0-bend drawings (so-called 2D-visibility drawings) where the vertices are represented by rectangles of different sizes. Besides of some motivating pictures we also give two simple theoretical results on this model.

This is joint work with Uli Fößmeier.

Implementation of graph algorithms with LEDA

Stefan Näher
Fachbereich Mathematik und Informatik
Universität Halle

LEDA is a Library of Efficient Datatypes and Algorithms that can be used as a platform for various kinds of combinatorial and geometric computing. In this talk we demonstrate that LEDA is particularly useful for implementing graph algorithms by providing an efficient and comfortable graph data type and many basic graph algorithms and data structures related to graphs. In particular, the programs for computing a minimum spanning tree (Kruskal), shortest paths in non-negative edge weights (Dijkstra) and *st*-numberings for biconnected graphs are presented. In the second part of the talk some details of the implementation of the Kuratowski-Algorithm (Hundack, Mehlhorn, Näher) are discussed.

The proximity drawability problem

Giuseppe Liotta
Department of Computer Science
Brown University

A proximity drawing of a graph G is a straight-line drawing of G such that pairs of adjacent vertices are deemed "close" according to some proximity measure. For example, in a *Gabriel drawing* two vertices are adjacent if and only if the circle having the two vertices as antipodal points is empty (i.e. it does not contain any other third vertex).

Aim of the talk is to survey different types of proximity drawings and describe some of the main combinatorial results and drawing algorithms that can be found in the literature.

Two arc disjoint paths in Eulerian digraphs

Toshihide Ibaraki
Graduate School of Engineering
Kyoto University

Let G be an Eulerian digraph, and $\{x_1, x_2\}$, $\{y_1, y_2\}$ be two pairs of vertices in G. An instance $(G; \{x_1, x_2\}\, \{y_1, y_2\})$ is called feasible if it contains two arc disjoint $x'x''$– and $y'y''$–paths, where $\{x', x''\} = \{x_1, x_2\}$ and $\{y', y''\} = \{y_1, y_2\}$. An $O(m + n \log n)$ time algorithm is presented to decide whether G is feasible, where n and m are the numbers of vertices and arcs in G, respectively. The algorithm is based on a structural characterization of minimal infeasible instances.

This is joint work with Hiroshi Nagamochi and Andras Frank.

Two results on search and edge connectivity

David W. Matula
Depart. of Computer Science & Engineering
Southern Methodist University

We give two results employing iterated search to provide information on the maximum number of edge disjoint $u-v$ paths, $\lambda(u,v)$, for all u,v in an n vertex m edge graph. Here each unit flow augmentation is determined by a breath first search (BFS).

Our first result is that in time $O(nm)$ we can confirm either:

(1) $\lambda(u,v) = \min\{\deg(u), \deg(v)\}$ for all vertex pairs u,v; or

(2) for some x,y there is a non trivial x,y min cut $(A,\bar{A})$,

$$x \in A,\ y \in \bar{A},\text{ with size } |(A,\bar{A})| \leq \min\{\deg(x), \deg(y)\} - 1.$$

In particular our algorithm needs to find at most $[n - max\,\text{degree}\,G]$ BFS's with additional work only $O(m)$. In case (1) we note the Gomory-Hu cut tree must then be a star with center vertex a max degree vertex of the graph G.

Our second result employs maximum adjacency search vertex ordering (MAS) which can be determined in linear time $O(n+m)$. An MAS ordering requires for each i that v_i have maximum adjacency with $\{v_1, v_2, \ldots, v_{i-1}\}$ among the remaining vertices $V - \{v_1, v_2, \ldots, v_{i-1}\}$. Ibaraki and Nagamochi have shown an edge labeling associated with an MAS that partitions the edges onto nested spanning forests $F_1, F_2, \ldots F_{\deg(v\backslash)}$ such that an edge labeled k has k edge-disjoint paths between its endpoints, one each in $F_1, F_2, \ldots F_k$. We describe an associated minimum acyclic arc labeling (MA^2L) procedure determining two partitions of the edges into nested spanning forests $F_1, F_2, \ldots F_{\deg(v\backslash)}$ and $F'_1, F'_2, \ldots, F'_{\deg(v_1)}$. Each edge having two labels, k from the first forest and j from the second, indicates there are $k+j-1$ mutually edge disjoint paths of G between its endpoints, one each in $F_1, F_2, \ldots, F_k$ and $F'_1, F'_2, \ldots, F'_{j-1}$. The MA^2L can be found in time only slightly greater than linear by the inverse Ackermann-function (embedded Union-Find Problem). The MA^2L results generally provide paths giving a tight lower bound on the $\lambda(u,v)$ values providing the path dual to the Gomory-Hu cut tree.

Practical performance of MIN CUT algorithms

Michael Jünger
Institut für Informatik
Universität Köln

The computation of minimum weight cuts in simple graphs with nonnegative edge weights has direct applications in connectivity problems, and is used as a subroutine in algorithms for some difficult combinatorial optimization problems. E.g., branch and cut algorithms for the traveling salesman problem make thousands of calls to mincut subroutines in their separation of valid inequalities. In the recent years, a lot of progress has been made, and some interesting new methods have been proposed. We give a computational comparison of several of them on various data sets. The study includes implementations of algorithms by Gomory & Hu (Gusfield), Padberg & Rinaldi, Karger & Stein, Hao & Orlin, Nagamochi, Ono & Ibaraki, Stoer & Wagner. The outcome of our experiments could not be predicted by the theoretical performance analyses.

This is joint work with Giovanni Rinaldi and Stefan Thienel.

Optimization on hierarchical graphs

Petra Mutzel
Max-Planck-Institut für Informatik

We study optimization problems in hierarchical graphs that have applications in Computational Biology and Automatic Graph Drawing. Methods that score the similarity between various sequences of letters are called *Multiple Alignment Problems*. The *Maximum Weight Trace Problem* (*MWT*) is one of the various models. Given an alignment graph (arising from the given sequences) with edge weights, find a set of edges that can be realized by an alignment (called *Trace*) of maximum weight. We give a characterization in terms of forbidden subgraphs that enables us to give an integer linear programming formulation for the MWT-problem. We report on first computational results on solving the LP-relaxation.

The problems we are considering in Automatic Graph Drawing help to improve the drawings of hierarchical graphs. We are focussing on the *Straightline Crossing Minimization Problem* and on the *L-Planarization Problem* on two layers. Depending on the number of layers, in which the vertices can be permuted freely (zero, one or two) different versions of the problems arise.

All of these problems are tightly connected to each other. E.g., the MWT-problem for two sequences is equivalent to L-planarization on two layers with both layers fixed. The knowledge of the latter problem and of the straight-line crossing minimization problem on two layers with one layer fixed, will give us information about the L-planarization problem on two layers with one layer fixed.

The MWT-problem is joint work with J. Kececioglu (University of Georgia), H.-P. Lenhof, K. Mehlhorn and K. Reinert (MPI Saarbrücken), the crossing minimization problem is joint work with M. Jünger (Universität zu Köln). Thanks to T. Ziegler and R. Weißkircher (MPI Saarbrücken) who provided most of the software for the graph drawing problems.

The bandwidth problem of clique caterpillars and tolerance graphs

Maciej Sysło
Institute of Computer Science
University of Wroclaw

We consider the bandwidth problem restricted to clique caterpillars and tolerance graphs. The bandwidth of a graph G is the minimum of the maximum absolute differece between adjacent labels when vertices are labelled with integers. We show that the bandwidth of cliques with hairs and the bandwidth of clique caterpillars with hairs of length 1 depends only on local density, that is, it is equal to $\max_{H \subseteq G}(|V(H)|-1)/\text{diam}(H)$, where $|V(H)|$ is the number of vertices of H and $\text{diam}(H)$ is the largest length of all shortest paths between any two vertices of H.

We also present algorithms which label optimally these graphs. Then, we show that the bandwidth problem for clique caterpillars with hairs of length at most 2 is $\mathcal{NP}$-complete. Moreover, it is shown that the problem remains $\mathcal{NP}$-complete for tolerance graphs, a generalization of interval graphs (for which there exists a polynomial-time algorithm).

PlaNet: A demonstration package for algorithms on PLAnar NETworks

Dorothea Wagner
Fakultät für Mathematik und Informatik
Universität Konstanz

PlaNet is a package for algorithms on planar networks. It comes with a graphical user interface, which may be used for demonstrating and animating algorithms. Our focus so far has been on disjoint path problems. However, the package is intended to serve as a general framework, where n algorithms for various problems on planar networks may be integrated and visualized. For this aim, the structure of the package is designed so that integration of new algorithms and even new algorithmic problems amounts to applying a short "recipe". The package has been used to develop new variations on well-known disjoint path algorithms, which heuristically optimize additional objections such as the total length of all paths.

A demo version is accessible through the WWW:

`http://www.informatik.uni-konstanz.de/Research/Projects/PlaNet/`

This is joint work with Dagmar Handke, Karsten Weihe, Gabriele Neyer, and Wolfram Schlickenrieder.

Painting graph algorithms

David W. Matula
Depart. of Computer Science & Engineering
Southern Methodist University

Painting Graph Algorithms (PGA) is an interactive program written in Tcl/Tk for use in X Window systems. The program was created to serve as a graph algorithm research tool. It has also proved to be useful for teaching graph algorithms. Currently, PGA is primarily used to refine and extend graph and network search algorithms.

PGA allows the user to draw and edit graph diagrams on a canvas. The user can add and delete vertices and edges as well as move vertices. PGA supports directed, undirected, and weighted graphs. Undirected graphs can be displayed with straight edges or pairs of curved arcs. The adjacency list data structure is dynamically updated as the graph is drawn and edited. The graph data structures can be saved in a file format suitable for input to other graph programs.

The painting process animates graph search algorithms by dynamically coloring vertices and edges as they are reached and visited. Color, and varying brush widths, are applied to edges and vertices to display information visually, with labels added to elements to indicate the current state of computation. PGA can be executed in the wish shell, allowing the user to directly manipulate the data structures as the painting progresses. Painted elements can also be moved on the canvas, for improved visual display. By experimenting with artistic variations of painting the graph, we have realized painted graphs that convey the multiple computational results of search algorithms such as MAS. Painted graphs can be printed or saved as postscript files.

For information on program availability please contact Michael Coming at `coming@seas.smu.edu`.

$L_{\bowtie}$–Steiner trees

Bernhard Korte
Institut für Ökonometrie und OR
Universität Bonn

To construct minimal Steiner trees is an old mathematical problem, which in its simplest form goes back to Fermat. In some sense it is wrongly attributed to the swiss geometer Jacob Steiner. The first modern formulation as a shortest network problem was given by the czech mathematicians Jarnik and Kössler in 1934. Today minimal L_1-Steiner trees are a central issue in VLSI design.

Compared to other combinatorial optimization problems like the TSP the Steiner tree problem was considered much harder. Only very small instances could be solved optimally.

Based on some decomposition result which states that each optimal L_1-Steiner tree consists only of four types of elementary trees (so called firs) and based on powerful reductions, we could develop a branch-and-bound type algorithm which can optimally construct L_1-Steiner trees for up to 100 terminals in reasonable time. This algorithm enables us to do the complete routing of up to a million nets on a chip as optimal L_1-Steiner trees.

All-pairs MIN CUT in sparse networks

Christos Zaroliagis
Max-Planck-Institut für Informatik

Algorithms are presented for the all-pairs min-cut problem in bounded tree-width, planar and sparse networks. The approach used is to preprocess the input n-vertex network so that, afterwards, the value of a min-cut between any two vertices can be efficiently computed. A tradeoff is shown between the preprocessing time and the time taken to compute min-cuts subsequently. In particular, after an $O(n \log n)$ preprocessing of a bounded tree-width network, it is possible to find the value of a min-cut between any two vertices in constant time. This implies that for such networks the all-pairs min-cut problem can be solved in time $O(n^2)$. This algorithm is used in conjunction with a graph decomposition technique of Frederickson to obtain algorithms for sparse and planar networks. The running times depend upon a topological property, γ, of the input network. The parameter γ varies between 1 and $\Theta(n)$; the algorithms perform well when $\gamma = o(n)$. The value of a min-cut can be found in time $O(n + \gamma^2 \log \gamma)$ and all-pairs min-cut can be solved in time $O(n^2 + \gamma^4 \log \gamma)$ for sparse networks. The corresponding running times for planar networks are $O(n + \gamma \log \gamma)$ and $O(n^2 + \gamma^3 \log \gamma)$, respectively. The latter bounds depend on a result of independent interest: outerplanar networks have small “mimicking” networks which are also outerplanar.

This is joint work with S. Arikati and S. Chaudhuri.

Some linear time solvable subcases of the multilayer routing problem

Andras Recski
Faculty of Electrical Engineering and Information
University of Budapest

Most of the interesting special cases of the detailed routing problem are $\mathcal{NP}$-complete, for example to determine whether a given channel routing problem can be solved with given width in the 2-layer Manhattan model.

However, there are some polynomially solvable subcases as well. For example, if all the terminals to be interconnected are on a single boundary of the rectangular routing area (the so called single row routing problem) then the 2-layer Manhattan routing can always be performed, in fact in linear time, by a classical algorithm of Tibor Gallai. Similarly, the solvability of the switchbox routing problem with edge disjoint paths in a single layer can always be decided in linear time by an algorithm of Dorothea Wagner.

In this talk we present linear time solutions for two problems, namely the channel routing problem in the 2-layer unconstrained model and the switchbox routing problem in the multilayer unconstrained model. Unfortunately, none of the algorithms are optimal (the first one does not realize minimum width and the second one does not minimize the number of layers).

This is joint work with Frank Strzyzewski, Endre Boros, and Ferenc Wettl.

Application Problems

Chair: Ioannis Tollis

Goos Kant: Dijkstra's shortest path algorithm using less memory There are several implementations of Dijkstra's single source shortest path algorithm. In practical instances of route maps at the ORTEC company, the number of vertices and edges is very large compared to the maximum edge weight. As an example, a road map of Europe has about 300,000 nodes, and the travel time of a link/edge is at most 60 minutes. If we represent every edge weight by a bucket and a doubly linked list for all vertices, currently labeled with the corresponding weight, we need 2 pointers at every vertex. This together yields a datastructure for the algorithm.

However, 2 pointers is too much memory in practice. To decrease the memory requirements we don't want to shrink the network, and we don't want to apply a considerably larger running time. The solution we propose is based on the following observation: during the algorithm only a few vertices are 'interesting' at any step. At any step we store the 'interesting' vertices together with their 2 pointers in a hash table. Of course the hash table requires memory, but since the number of 'interesting' vertices during the algorithm is very small compared with the total number of vertices, this saves a lot of memory.

Is this approach already known, and are there any good alternatives to apply Dijkstra's algorithm on such networks?

Stephen North: Dynamic graph layout Our group has several years' experience in implementing graph-drawing systems and helping software developers to apply these systems to problems such as software process (work flow) modeling and finite state machine synthesis and analysis. Although users are pleased with automatic graph layout, they say that stable incremental layout is essential. Lack of this prevents them from fully adopting our systems. One of our current projects is to design good heuristic algorithms, interfaces and systems for effective incremental layout. In fact, dynamic graph layout is highly relevant in many settings, as most real-world systems are subject to change. Another strong motivation is to browse large graphs by incrementally adjusting views of small subgraphs.

Stephen North: Efficient algorithms for graphs in external memory One project in our lab involves analyzing massive volumes of telephone network customer data. AT&T has more than 80,000,000 customers and on a recent day set handled a record of more than 239,000,000 telephone calls. We would like to solve graph pattern matching problems on this database (e.g. finding highly connected subgraphs). Besides being large, the graphs are dynamic and only approximate. We are examining the relevance of recent theoretical contributions on external graph algorithms and I/O parallelism.

Ioannis Tollis: Design of survivable networks Find better algorithms for designing cost efficient telecommunications networks, based on the survivable ring architecture model.

Ioannis Tollis: Algorithms for developing lower bounds on the cost of Self-Healing-Ring architectures When solving a difficult optimization problem using heuristics, it is very often the case that we have no idea about the quality of our solution. This is the case when designing Self-Healing-Ring architectures. It is an interesting problem to develop algorithms that for a given network produce a total cost which is a good lower bound for any (optimal) solution.

Goos Kant: Assigning orders to inspectors In several practical environments, the following problem arises: given n orders and k ($k < n$) inspectors distributed in an area, assign the n orders to the k inspectors such that every inspector has roughly the same working time. The working time of an inspector is defined by the travelling salesman tour from his location through all assigned orders back to his location + the visiting time of the assigned orders. Of course, this problem is NP-hard.

Most algorithms in the text books are based on the following technique: Compute for every order the extra cost for assigning this order to every inspector, say cost $c_1, \ldots, c_k$. Let c_α and c_β be the smallest and one but smallest cost, resp. Then this order is assigned to a customer, for which $c_\beta - c_\alpha$ is maximum. After updating the costs for the remaining orders, the procedure is repeated, until all orders are assigned. Variants are: looking at the best p ($p > 2$) inspectors, and special scaling values for popular and impopular areas of the location of the inspector.

However, in almost all cases, it is very difficult to deal with popular (and impopular) areas. In other words: in some areas the number of orders is relatively very large (small) compared with the number of inspectors. This implies that inspectors from impopular areas have to do orders in the direction of the popular regions. Steering this is quite complicated in combination with balancing the working times. Are there any papers or practical algorithms that can deal with these problems?

Open Problems

Chair: Roberto Tamassia

Roberto Tamassia: Robust Steinitz' Theorem Steinitz's Theorem says that a graph G is the skeleton of a convex 3-polytope if and only if it is planar and triconnected. Is there a "robust" version of this theorem? Namely, what is the set of values for constants h, k, and l such that given a planar triconnected graph G with N vertices, we can construct a convex 3-polytope P with skeleton G that verifies the following properties:

- P satisfies the *vertex resolution rule*, i.e., any two vertices of P are at Euclidean distance at least 1;
- the coordinates of the vertices of P are rational numbers with $O(N^h \log N)$ bits;
- P has $O(N^k)$ volume;
- P has $O(N^l)$ *aspect ratio* (ratio of the largest to the smallest of length, width, and height of P).

Ideally, we would like to characterize the set of all minimal triplets (h, k, l) and give efficient constructions for them.

Trivial lower bounds are $h \geq 0$, $k \geq 1$ and $l \geq 0$. However, no construction for these values of the constants is known. An $O(N^{1.2})$-time construction for $h = 1$, $k = 1$ and $l = 1$ is presented in [*]. While it achieves optimal volume, its disadvantages are the high number of bits for the vertex coordinates and the poor aspect ratio.

[*] M. Chrobak, M.T. Goodrich and R. Tamassia, "Convex Drawings of Graphs in Two and Three Dimensions," Proc. ACM Symp. on Computational Geometry, 319-328 (1996).

Peter Eades: The K7-3D-2B Problem Is there an orthogonal grid drawing of K_7 in 3 dimensions with no edge crossings and at most 2 bends per edge?

(An *orthogonal grid* drawing of a graph represents vertices as points on the integer grid and edges as polylines whose segments lie on the integer grid lines.)

Ioannis Tollis: "Better" hierarchical drawings Find algorithms for hierarchical drawings of graphs that take into account the global shape of the graph: For example, draw a graph by removing edges until the remaining (sub)graph is a series-parallel, or an upward planar graph. Draw the subgraph and then insert the edges one at a time.

Petra Mutzel: Separation Problem of Kuratowski Subgraphs Given a nonplanar graph $G = (V, E)$ with weights w_e for all edges $e \in E$, $0 \leq c_e \leq 1$ (most of them are 0, or between 0 and 0.2). Find a subset $K \subseteq E$ that is a subdivision (homeomorph graph) of K_5 or $K_{3,3}$ with minimum weight $\sum_{e \in K} w_e$.

The above question arises as "separation problem" in the maximum planar subgraph problem. Given a solution of a Linear Program, x_e, $0 \leq x_e \leq 1$, find a Kuratowski subgraph (i.e., a subdivision of $K_{3,3}$ or K_5) violating the inequality $\sum_{e \in K} x_e \leq |K| - 1$ or prove that none such subgraph exists. Rewriting the inequality and setting $w_e = 1 - x_e$ gives $\sum_{e \in K} w_e \geq 1$. If the minimum weight Kuratowski subgraph satisfies the inequality, then it is satisfied by all the Kuratowski subgraphs. Otherwise, we found a Kuratowski subgraph that violates the inequality, i.e. $\sum_{e \in K}(1 - x_e) < 1$.

Takao Nishizeki and X. Zhou: Can the vertex-coloring problem be "simply" reduced to the edge-coloring problem? The opposite direction is easy: the edge-coloring problem can be simply reduced to the vertex-coloring problem, because the chromatic index of a graph G is equal to the chromatic number of the line graph of G. Since both problems are NP-complete, either can be reduced to the other plausibly through 3-SAT, due to the theory of NP-completeness. Thus the open problem asks, given a graph G and a positive integer k, whether one can find in polynomial time a new graph G' and a positive integer k' such that G can be vertex-colored with at most k colors if and only if G' can be edge-colored by at most k' colors.

Sue Whitesides: How quickly can we determine if a 4-regular graph G is the union of two Hamilton cycles? Let G be a connected graph on n vertices such that each vertex has degree 4. How quickly can we determine whether the edge set $E(G)$ of G can be partitioned into two sets, each of which forms a Hamiltonian cycle for G? More generally, how quickly can we determine whether the edges of a $2k$-regular graph G can be partitioned into k edge-disjoint Hamilton cycles on G? Are these problems NP-complete?

(These questions arose in discussions with Tom Shermer of Simon Fraser University, Canada.)

Giuseppe Liotta: The planar 2D-visibility representability problem Given a set S of axis-aligned rectangles in the plane, two rectangles of S are horizontally (vertically) visible if they can be connected by an horizontal (vertical) straight-line segment l that does not intersect any other rectangle of S. Segment l is called segment of visibility.

A 2D-visibility representation of a graph G is a drawing of G on the plane such that every vertex is represented by an axis-aligned rectangle and two vertices are adjacent if and only if their corresponding rectangles are either horizontally or vertically visible. An edge of a 2D-visibility representation is either an horizontal or a vertical straight-line segment of visibility. A 2D-visibility representation is planar if no two segments of visibility intersect.

Question: Which planar graphs admit a planar 2D-visibility representation?

Context for the question: graph drawing, visibility representations, zero-bend drawings.

Kurt Mehlhorn: Robustness of network algorithms There are simple examples where network algorithms when implemented with floating point arithmetic produce wrong results. For shortest path algorithms which only add numbers it is fairly easy to show that the computed result is close to the true result. More complicated network algorithms add and subtract and errors may accumulate. Can one still show a relation between computed result and true result?

Ioannis Tollis: Nontrivial properties of planar bipartite graphs The ultimate aim is to partition the vertices of a planar graph into two parts such that each part is bipartite. This is related to the 4-coloring theorem, and would give an independent constructive proof of the theorem. So, find properties of planar bipartite graphs that will allow one to discover algorithms to partition the vertices as mentioned above.

Takao **Asano**
Chuo University
Dept. of Information and System Engineering
Bunkyo-ku
Tokyo 112
Japan
asano@ise.chuo-u.ac.jp
tel: +81-3-3817-1686

Tetsuo **Asano**
Osaka Electro-Communication University
Dept. of Engineering Informatics
Faculty of Information & Computer Science
18-8 Hatsu-cho/Neyagawa
Osaka 572
Japan
asano@amlab.osakac.ac.jp
tel: +81-720-24-1131/ext. 2450

Franz J. **Brandenburg**
Universität Passau
Lehrstuhl für Theoretische Informatik
94030 Passau
Germany
brandenb@fmi.uni-passau.de
tel: +49-851-509-3030

Ulrik **Brandes**
Universität Konstanz
Fakultät für Mathematik und Informatik
Postfach 55 60 D 188
78434 Konstanz
Germany
Ulrik.Brandes@uni-konstanz.de
tel: +49-7531-88-4263

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

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

Wen-Lian **Hsu**
Academica Sinica – Taipei
Institute of Information Science
Nankang
Taipei 115
Taiwan
hsu@iis.sinica.edu.tw

Toshihide **Ibaraki**
Kyoto University
Graduate School of Engineering
Department of Applied Math. and Physics
Kyoto 606
Japan
ibaraki@kuamp.kyoto-u.ac.jp
tel: +81-75-753-5504

Michael **Jünger**
Universität zu Köln
Institut für Informatik
Pohligstr. 1
D–50969 Köln
Germany
mjuenger@informatik.uni-koeln.de
tel: +49-221-470-5313

Goos **Kant**
Utrecht University
Dept. of Computer Science
P.O. Box 80-089
NL–3508TB Utrecht
The Netherlands
goos@cs.ruu.nl

Naoki **Katoh**
Kobe University of Commerce
Department of Management Science
8-2-1 Gakuennishi
Nishi-ku Kobe 651-21
Japan
naoki@kucgw.kobeuc.ac.jp
tel: +81-78-794-6161

Michael **Kaufmann**
Universität Tübingen
Institut für Informatik
Sand 13
D–72076 Tübingen
Germany
mk@informatik.uni-tuebingen.de
tel: +49-7071-29-7404

Philip **Klein**
Brown University
Department of Computer Science
P.O. Box 1910
Providence RI 02912
USA
pnk@cs.brown.edu
tel: +1-401-863-76 44

Bernhard **Korte**
Universität Bonn
Institut für Ökonometrie und OR
Nassestr. 2
D–53113 Bonn
Germany
dm@or.uni-bonn.de
tel: +49-228-738770

Han **La Poutré**
Leiden University
Dept. of Computer Science
Niels Bohrweg 1
NL–2300 RA Leiden
The Netherlands
han@wi.leidenuniv.nl
tel: +31-71-5277112

Annegret **Liebers**
Universität Konstanz
Fakultät für Mathematik und Informatik
Postfach 55 60 D 188
78434 Konstanz
Germany
Annegret.Liebers@uni-konstanz.de
tel: +49-7531-88-4263

Giuseppe **Liotta**
Brown University
Department of Computer Science
115 Waterman Street
Providence RI 02912
USA
gl@cs.brown.edu

David W. **Matula**
Southern Methodist University
Depart. of Computer Science & Engineering
75275 Dallas TX
USA
matula@seas.smu.edu

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

Rolf **Möhring**
TU Berlin
Fachbereich Mathematik, MA 6-1
Straße des 17. Juni 136
10623 Berlin
Germany
moehring@math.tu-berlin.de
tel: +49-30-314-24594

Bojan **Mohar**
University of Ljubljana
Dept. of Mathematics
Jadranska 19
1111 Ljubljana
Slovenia
bojan.mohar@uni-lj.si

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

Stefan **Näher**
Universität Halle
Fachbereich Mathematik und Informatik
Kurt-Mothes-Str. 1
D–06099 Halle
Germany
naeher@informatik.uni-halle.de
tel: +49-345-552-4712

Hiroshi **Nagamochi**
Kyoto University
Graduate School of Engineering
Dept. of Applied Mathematics and Physics
Kyoto 606-01
Japan
naga@kuamp.kyoto-u.ac.jp
tel: +81-75-753-5494

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

Stephen **North**
AT&T Bell Labs
600 Mountain Avenue
Murray Hill NJ 07974
USA
north@research.att.com

Andras **Recski**
Technical University of Budapest
Faculty of Elec. Eng. and Information
Dept. of Mathematics and Computer Science
H–1521 Budapest
Hungary
recski@vma.bme.hu
tel: +36-1-463-2585

Maciej M. **Sysło**
University of Wroclaw
Institut of Computer Science
Przesmyckiego 20
PL–51–151 Wroclaw
Poland
syslo@ii.uni.wroc.pl
tel: +48-71-247-382
fax: +48-71-251271

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

Ioannis **Tollis**
University of Texas at Dallas
Computer Science Department
EC 31
P.O. Box 830688
Richardson TX 75083-0688
USA
tollis@utdallas.edu
tel: +1-214-8832180

Dorothea **Wagner**
Universität Konstanz
Fakultät für Mathematik und Informatik
Postfach 55 60 D188
78434 Konstanz
Germany
Dorothea.Wagner@uni-konstanz.de
tel: +49-7531-88-2893

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

Toshimasa **Watanabe**
Hiroshima University
Faculty of Engineering
Department of Circuits and Systems
Kagamiyama 1-4-1
Higashi-Hiroshima City 739
Japan
watanabe@huis.hiroshima-u.ac.jp
tel: +81-824-24-7662

Karsten **Weihe**
Universität Konstanz
Fakultät für Mathematik und Informatik
Postfach 55 60 D188
78434 Konstanz
Germany
karsten.weihe@uni-konstanz.de
tel: +49-7531-88-4375

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

Christos **Zaroliagis**
Max-Planck-Institut für Informatik
Im Stadtwald
66123 Saarbrücken
Germany
zaro@mpi.sb.mpg.de
tel: +49-681-9325-116