Graph Drawing 2004

City College, New York

Contents

1 Session 0

1.1 Paul Seymour
The Structure of Claw-Free Graphs

A graph is "claw-free" if no vertex has three pairwise nonadjacent neighbors. Line graphs are claw-free, and there has been a great deal of work on extending various theorems about line graphs to claw-free graphs. Claw-free graphs are more general than line graphs, but how much more general are they?
A graph is "claw-free" if no vertex has three pairwise nonadjacent neighbors. Line Certainly there are claw-free graphs that are not line graphs; for instance, the icosahedron is claw-free, and so is the Schlafli graph, and so is every graph with stability number at most two. Another type of claw-free graph is made as follows: arrange vertices in a circular order, choose some intervals from this circular order, and make two vertices adjacent if and only if one of these intervals contains them both. In general none of these examples are line graphs.
A graph is "claw-free" if no vertex has three pairwise nonadjacent neighbors. Line In joint work with Maria Chudnovsky, we found an explicit construction for all claw-free graphs. We showed that there are a few "basic" types, such as those described above, and every claw-free graph can be built starting from graphs of these types by piecing them together using simple composition operations. We explain this work.

1.2 Erik Demaine
Fast Algorithms for Hard Graph Problems: Bidimen sionality, Minors, and Local Treewidth

The newly developing theory of bidimensional graph problems provides general techniques for designing efficient fixed-parameter algorithms and approximation algorithms for NP-hard graph problems in broad classes of phs. This theory applies to graph problems that are *bidimensional* in the sense that (1) the solution value for the $k \times k$ grid graph (and similar graphs) grows with k, typically as $\Omega(k^2)$, and (2) the solution value goes down when contracting edges and optionally when deleting edges. Examples of such problems include feedback vertex set, vertex cover, minimum maximal matching, face cover, a series of vertex-removal parameters, dominating set, edge dominating set, r-dominating set, connected dominating set, connected edge dominating set, connected r-dominating set, and unweighted TSP tour (a walk in the graph visiting all vertices). Bidimensional problems have many structural properties; for example, any graph embeddable in a surface of bounded genus has treewidth bounded above by the square root of the problem's solution value. These properties lead to efficient—often subexponential—fixed-parameter algorithms, as well as polynomial-time approximation schemes, for many minor-closed graph classes. One type of minor-closed graph class of particular relevance has *bounded local treewidth*, in the sense that the treewidth of a graph is bounded above in terms of the diameter; indeed, we show that such a bound is always at most linear.

The bidimensionality theory unifies and improves several previous results. The theory is based on algorithmic and combinatorial extensions to parts of the Robertson-Seymour Graph Minor Theory, in particular initiating a parallel theory of graph contractions. The foundation of this work is the topological theory of drawings of graphs on surfaces.
This is joint work with Fedor Fomin, MohammadTaghi Hajiaghayi, and Dimitrios Thilikos.

2 City College Session (Session 1)

2.1 N. Bonichon, S. Felsner, M. Mosbah *Convex Drawings of 3-connected Plane Graphs.*

We use Shnyder woods of 3-connected planar graphs to produce convex straight line drawings on a grid of side-length $(n - 2 - \Delta)$. Here $\Delta \geq 0$ depends on the Shnyder wood used for the drawing the range is $0 \leq \Delta \leq \frac{n}{2} - 2$.

2.2 P. Healy, K. Lynch *Building Blocks of Upward Planar Diagraphs.*

We investigate the upward planarity of digraphs. In particular we show that a digraph is upward planar if and only if its biconnected components have upward planar drawings with certain easily detectable properties.

2.3 G. Aloupis, P. Bose, P. Morin *Reconfiguring Triangulations with Edge Flips and Point Moves*

We examine reconfigurations between triangulations and near-triangulations of point sets, and given new bounds on the number of *point moves* and *edge flips* sufficient for any reconfiguration. we show that with $O(n \log n)$ edge flips and point moves, we can transform any geometric near-triangulation on n points to any other geometric near-triangulation on n possibly different points. This improves the previously known bound of $O(n^2)$ edge flips and point moves. We then show that with a slightly more general point move, we can further reduce the complexity to $O(n)$ point moves and edge flips.

2.4 P. Bose, F. Hurtado, E. Rivera-Campo, D. Wood *Partitions of Complete Geometric Graphs into Plane Trees*

Consider the open problem: does every complex geometric graph K_2n have a partition of its edge set into n plane spanning trees? We approach this problem from three directions. First, we study the case of convex geometric graphs. It is well known that the complete convex graph K_2n has a partition into n plane spanning trees. We characterize all such partitions. Second, we give a sufficient conditions, which generalizes the convex case, for a complete geometric graph to have a partition into plane spanning trees. Finally, we consider a relation of the problem in which the trees of the partition are not necessarily spanning. We prove that every complete geometric graph K_n can be partitioned into at most $n - \sqrt{n/12}$ plane trees.

2.5 H. Zhang, X. He

New Theoretical Bounds of Visibility Representation of Plane Graphs

In a *visibility representation* (VR for short) of a plane graph G, each vertex of G is represented by a horizontal line segment such that the line segments representing any two adjacent vertices of G are joined by a vertical line segment. Rosenstiehl and Tarjan [6]. Tamassia and Tollis [7] independently gave linear time VR algorithms for 2-connected plane graph. Afterwards, on of the main concerns for VR is the size of the representation. In this paper, we prove that any plane graph G has a VR with height bounded by $\lfloor \frac{5n}{6} \rfloor$. This improves the previously known bound $\lceil \frac{15n}{16} \rceil$. We also construct a plane graph G with n vertices where any VR of G require a size of $(\lfloor \frac{2n}{3} \rfloor) \times (\lfloor \frac{4n}{3} \rfloor - 3)$. Our result provides an answer to Kant's open question about whether there exists a plane graph G such that all of its VR require width greater that cn, where $c > 1$.

3 Session 2

3.1 R. Anderson, F. Chung, L. Lu
Drawing power law graphs

We present methods for drawing graphs that arise in various information networks. It has been noted that many realistic graphs have power law degree distribution and exhibit the small world phenomenon. Our algorithm is influenced by recent developments on modeling and analysis of such power law graphs.

3.2 D. Eppstein
Algorithms for drawing media

We describe algorithms for drawing media, systems of states, tokens and actions that have state transition graphs in the form of partial cubes. Our algorithms are based on two principles: embedding the state transition graph in a low-dimensional integer lattice and projecting the lattice onto the plane, or drawing the medium as a planar graph with centrally symmetric faces.

3.3 C. Gotsman, Y. Koren
Distributed Graph Layout for Sensor Networks

Sensor network applications frequently require that the sensors know their physical locations in some global coordinate system. This is usually achieved by equipping each sensor with a location measurement device, such as GPS. However, low-end systems or indoor systems, which cannot use GPS, must locate themselves based only on crude information available locally, such as inter-sensor distances. We show how a collection of sensors, capable only of measuring distances to close neighbors, can compute their locations in a purely distributed manner, i.e. where each sensor communicates only with its neighbors. This can be viewed as a distributed graph drawing algorithm. We experimentally show that our algorithm consistently produces good results under a variety of simulated real-world conditions.

3.4 M. Raitner
Visual Navigation of Compound Graphs

This paper describes a local update scheme for the algorithm of Sugiyama and Misue (IEEE Trans. Systems, Man and Cybernetics, 21(4):876892, 1991) for drawing views of compound graphs. A view is an abstract representation of a compound graph; it is generated by contracting subgraphs into meta nodes. Starting with an initial view, the underlying compound graph is explored by repeatedly expanding or contracting meta nodes in the current view. The novelty is a totally local update scheme of the algorithm of Sugiyama and Misue. It is more efficient than a complete relayout, because the expensive steps of the algorithm, e. g., level assignment or crossing minimization, are restricted to the modified part of the compound graph. The locality of the updates also preserves the users mental map: nodes not affected by the expand or contract operation keep their levels and their relative order; expanded edges take the same course as the corresponding contracted edge.

3.5 E. Gansner, Y. Koren, S. North
Graph Drawing by Stress Majorization

One of the most popular graph drawing methods is based on achieving graph-theoretic target distances. This method was used by Kamada and Kawai [15], who formulated it as an energy optimization problem. Their energy is known in the multidimensional scaling (MDS) community as the stress function. In this work, we show how to draw graphs by stress majorization, adapting a technique known in the MDS community for more than two decades. It appears that majorization has advantages over the technique of Kamada and Kawai in running time and stability. We also present a few extensions to the basic energy model which can improve layout quality and computation speed in practice. Majorization-based optimization is essential to these extensions.

4 Session 3

4.1 S. Hong, P. Eades
A Liner Time Algorithm for Constructing Maximally Symmetric Straight line Drawings of Planar Graphs

This paper presents a linear time algorithm for constructing maximally symmetric straight-line drawings of *biconnected* and *one-connected* planar graphs. Previously known algorithms run in quadratic time. We present an algorithm to find a plane embedding with the maximum number of symmetries, and an algorithm for constructing a drawing that achieves that maximum. Both algorithms run in linear time.

4.2 D. Ebner, G. klau, R. Weiskircher
Label Number Maximization in the Slider Model

We consider the NP-hard label number maximization problem lnm: Given a set of rectangular labels, each of which belongs to a point feature in the plane, the task is to find a labeling for a largest subset of the labels. A *labeling* is a placement such that none of the labels overlap and each is placed so that its boundary touches the corresponding point feature. The purpose of this paper is twofold: We present a new force-based simulated annealing algorithm to heuristically solve the problem and we provide the results of a very thorough experimental comparison of the best known labeling methods on widely used benchmark sets. The design of our new method has been guided by the goal to produce labelings that are similar to the results of an experienced human performing the same task. So we are not only looking for a labeling the number of labels placed is high but also where the distribution of the placed labels is good. Our experimental results show that the new algorithm outperforms the other methods in terms of quality while still being reasonably fast and confirm that the simulated annealing method is well-suited for map labeling problems.

4.3 M. Eiglsperger, m. Siebenhaller, M. Kaufmann
An Efficient Implementation of Sugiyama's Algorithm for Layered Graph Drawing

Sugiyama's algorithmic framework for layered graph drawing is commonly used in practical software. The extensive use of dummy vertices to break long edges between non-adjacent layers often leads to unsatisfactorial performance. The worst-case running-time is $O(|V||E| \log |E|)$ requiring $O(|V||E|)$ memory, which makes the approach unusable for the visualization of large graphs. By a conceptually simple new technique we are able to keep the number of dummy vertices and edges linear in the size of the graph and hence reduce the worst-case time complexity of Sugiyama's approach by an order of magnitude to $O((|V| + |E|) \log |E|)$ requiring $O(|V| + |E|)$ space.

4.4 M. Forster
A Fast and Simple Heuristic for Constrained Two-Level Crossing Reduction

The one-sided two-level crossing reduction problem is an important problem in hierarchical graph drawing. Because of its NPhardness there are many heuristics, such as the well-known barycenter and median heuristics. We consider the constrained one-sided two-level crossing reduction problem, where the relative position of certain vertex pairs on the second level is fixed. Based on the barycenter heuristic, we present a new algorithm that runs in quadratic time and generates fewer crossings than existing simple extensions. It is significantly faster than an advanced algorithm by Schreiber [12] and Finnocchi [1, 2, 6], while it compares well in terms of crossing number. It is also easy to implement.

4.5 J. Boyer
Additional PC-tree Planarity Conditions

Recent research efforts have produced new algorithms for solving planarity-related problems. One such method performs vertex addition using the PC-tree data structure, which is similar to but sim- pler than the well-known PQ-tree. For each vertex, the PC-tree is first checked to see if the new vertex can be added without violating certain planarity conditions; if the conditions hold, the PC-tree is adjusted to add the new vertex and processing continues. The full set of planarity conditions are required for a PC-tree planarity tester to report only pla- nar graphs as planar. This paper provides further analysis and new pla- narity conditions needed to produce a correct planarity algorithm with a PC-tree.

5 University of North Texas Session (Session 4)

5.1 V. Dujmovic, D. Wood
Layouts of Graph Subdivisions

A k-stack layout (respectively, k-queue layout) of a graph consists of a total order of the vertices, and a partition of the edges into k sets of non-crossing (non-nested) edges with respect to the vertex ordering. A k-track layout of a graph consists of a vertex k-colouring, and a total order of each vertex colour class, such that between each pair of colour classes no two edges cross. The stack-number (respectively, queue-number, track-number) of a graph G, denoted by sn(G) (qn(G), tn(G)), is the minimum k such that G has a k-stack (k-queue, k-track) layout. This paper studies stack, queue, and track layouts of graph subdivisions. It is known that every graph has a 3-stack subdivision. The best known upper bound on the number of division vertices per edge in a 3-stack subdivision of an n-vertex graph G is improved from O(log n) to O(logminsn(G), qn(G)). This result reduces the question of whether queue-number is bounded by stack-number to whether 3-stack graphs have bounded queue number. It is proved that every graph has a 2- queue subdivision, a 4-track subdivision, and a mixed 1-stack 1-queue subdivision. All these values are optimal for every non-planar graph. In addition, we characterize those graphs with k-stack, k-queue, and k-track subdivisions, for all values of k. The number of division vertices per edge in the case of 2-queue and 4-track subdivisions, namely O(log qn(G)), is optimal to within a constant factor, for every graph G. Applications to 3D polyline grid drawings are presented. For example, it is proved that every graph G has a 3D polyline grid drawing with the vertices on a rectangular prism, and with O(log qn(G)) bends per edge.

5.2 L. Torok, I. Vrto
Layout Volumes of the Hypercube

We study 3-dimensional layouts of the hypercube in a 1-active layer and general model. The problem can be understood as a graph drawing problem in 3D space and was addressed at Graph Drawing 2003 [5]. For both models we prove general lower bounds which relate volumes of layouts to a graph parameter cutwidth. Then we propose tight bounds on volumes of layouts of N-vertex hypercubes. Especially, we have $VOL_{1-AL}(Q_{\log N}) = \frac{2}{3}N^{\frac{3}{2}} \log N + O(N^{\frac{3}{2}})$, for even $\log N$ and $VOL(Q_{\log N}) = \frac{2\sqrt{6}}{9}N^{\frac{3}{2}} + O(N^{4/3} \log N)$, for $\log N$ divisible by 3. The 1-active layer layout can be easily extended to a 2-active layer (bottom and top) layout which improves a result from [5].

5.3 V. Dujmovic', M. Suderman, D. Wood
Really Straight Graph Drawings

We study straight-line drawings of graphs with few segments and few slopes. Optimal results are obtained for all trees. Tight bounds are obtained for outerplanar graphs, 2-trees, and planar 3-trees.We prove that every 3-connected plane graph on n vertices has a plane drawing with at most 5n/2 segments and at most 2n slopes, and that every cubic 3-connected plane graph has a plane drawing with three slopes (and three bends on the outerface). Drawings of non-planar graphs with few slopes are also considered. For example, it is proved that graphs of bounded degree and bounded treewidth have drawings with O(log n) slopes.

5.4 D. Eppstein, M. Goodrich, J. Meng *Confluent Layered Drawings*

We combine the idea of confluent drawings with "Sugiyama" style drawings, in order to reduce the edge crossings in the resultant drawings. Furthermore, it is easier to understand the structures of graphs from the mixed style drawings. The basic idea is to cover a layered graph by complete bipartite subgraphs (bicliques), then replace bicliques with tree-like structures. The biclique cover problem is reduced to a special edge coloring problem and solved by heuristic coloring algorithms. Our method can be extended to obtain multi-depth confluent layered drawings.

5.5 P. Hui, M. Schaefer, D. Stefankovic *Train Tracks and Confluent Drawings*

Imagine a railroad system with train tracks and switches connecting different stations. Can we run a train from one station to another without changing directions? If you were in Chicago, say, you could take the blue line to get from Jackson to Forest Park or from Jackson to Cermak. However, you could not get from Cermak to Forest Park without changing direction, since the two stations are on different branches of the blue line.[1] We can ask what graph is represented by a particular railway system. Dickerson, Eppstein, Goodrich, and Meng [2] introduced the concept of *confluent graphs* to capture the connection properties of train tracks. Confluent graphs are a very natural generalization of planar graphs, and—as the example of railroad maps shows—are an important tool in graph visualization. In this paper we continue the study of confluent graphs introducing strongly confluent graphs and tree-confluent graphs. We show that strongly confluent graphs can be recognized in NP (the complexity of recognizing confluent graphs remains open). We also give a natural elimination ordering characterization of tree-confluent graphs which shows that they form a subclass of the chordal bipartite graphs, and can be recognized in polynomial time.

6 Session 5

6.1 A. Dean, E. Gethner, J. Hutchinson *Unit Bar-visibility Layouts of Triangulated Polygons: Extended Abstract*

A *triangulated polygon* is a 2-connected maximal outerplanar graph. A *unit bar-visibility graph* (UBVG for short) is a graph whose vertices can be represented by disjoint, horizontal, unit-length bars in the plane so that two vertices are adjacent if and only if there is a nondegenerate, unobstructed, vertical band of visibility between the corresponding bars. We give combinatorial and geometric characterizations of the triangulated polygons that are UBVGs. To each triangulated polygon G we assign a character string with the property that G is a UBVG if and only if the string satisfies a certain regular expression. Given a string that satisfies this condition, we describe a linear-time algorithm that uses it to produce a UBV layout of G.

6.2 H. de Fraysseix, P. Ossona de Mendez *Contact and Intersection Representations*

Abstract. A necessary and sufficient condition is given for a connected Bipartite graph to be the incidence graph of a family of segments and points. We deduce that any 4-connected 3-colorable plane graph is the contact graph of a family of segments and that any 4-colored plane graph without an induced C4 using 4 colors is the intersection graph of a family of straight line segments.

6.3 E. Di Giacomo, W. Didimo, G. Liotta, H. Meijer *Computing Radial Drawings on the Minimum Number of Circles*

A radial drawing is a representation of a graph in which the vertices are constrained to be on concentric circles of finite radius. In this paper we study the problem of computing radial drawings of planar graphs by using the minimum number of concentric circles. We assume that the edges are drawn as straight-line segments and that co-circular vertices can be adjacent. It is proved that the problem can be solved in polynomial time. The solution is based on a characterization of those graphs that admit a crossing-free straight-line radial drawing on k circles. For the graphs in this family, a linear time algorithm that computes a radial drawing on k circles is also presented.

6.4 M. Kitching, S. Whitesides *The Three Dimensional Logic Engine*

We consider the following 3D graph embedding question: given a graph G, is it possible to assign its vertices to points in three dimensions such that G is isomorphic to the mutual nearest neighbor graph of the set P of points to which the vertices are assigned? We show that this problem is NP-hard. We do this by extending the "logic engine" method to three dimensions by using building blocks inspired by the structure of diamond and by constructions of Buchminster Fuller.

7 Session 6

7.1 A. Marcus, G. Tardos
Intersection reverse sequences and geometric applications

Pinchasi and Radoičić [8] used the following observation to bound the number of edges of a topological graph without a self-intersecting cycle of length 4: if we make a list of the neighbors for every vertex in such a graph and order these lists cyclicly according to the connecting edge, then the common elements in any two lists have reversed cyclic order. Building on their work we give an estimate on the size of the lists having this property. As a consequence we get that a topological graph on n vertices not containing a self-intersecting C_4 has $O(n^{3/2} \log n)$ edges. Our result also implies that n pseudo-circles in the plane can be cut into $O(n^{3/2} \log n)$ pseudo-segments, which in turn implies bounds on point-curve incidences and on the complexity of a level of an arrangement of curves.

7.2 J. Kyncl, J. Pach, G. Toth
Long alternating paths in bicolored point sets

Given n red and n blue points in convex positions in the plane, we show that there exists a noncrossing alternating path of length $n + c\sqrt{\frac{n}{\log n}}$. We disprove a conjecture of Erdös by constructing an example without any such path of length greater than $4n/3 + c'\sqrt{n}$.

7.3 S. Norine
Drawing Pfaffian Graphs

We prove that a graph is Pfaffian if and only if it can be drawn in the plane (possibly with crossings) so that every perfect matching intersects itself an even number of times.

7.4 M. Newton, O. Sykora, M. Uzovic, I. Vrto
New Exact Results and Bounds for Bipartite Crossing Numbers of Meshes

The bipartite crossing number of a bipartite graph is the minimum number of crossings of edges when the partitions are placed on two parallel lines and edges are drawn as straight line segments between the lines. We prove exact results, asymptotics and new upper bounds for the bipartite crossing numbers of 2-dimensional mesh graphs. We especially show that $\mathrm{bcr}(P_6 \times P_n) = 35n - 47$, for $n \geq 7$.

7.5 J. Balogh, G. Salazar

Improved bounds for the number of $(\leq k)$***–sets, convex quadrilaterals, and the rectilinear crossing number of*** K_n

We use circular sequences to give an improved lower bound on the minimum number of $(\leq k)$–sets in a set of points in general position. We then use this to show that if S is a set of n points in general position, then the number (S) of convex quadrilaterals determined by the points in S is at least $0.37533\binom{n}{4} + O(n^3)$. This in turn implies that the rectilinear crossing number $\bar{cr}(K_n)$ of the complete graph K_n is at least $0.37533\binom{n}{4} + O(n^3)$. These improved bounds refine results recently obtained by Ábrego and Fernández-Merchant, and by Lovász, Vesztergombi, Wagner and Welzl.

8 Session 7

8.1 P. F. Cortese, G. Di Battista, M. Patrignani, M. Pizzonia *Clustering Cycles into Cycles of Clusters*

In this paper we study the clustered graphs whose underlying graph is a cycle. This is simple family of clustered graphs that are "highly non connected". We start by studying 3-cluster cycles, that are clustered graphs such that the underlying graph is a simple cycle and there are three clusters all at the same level. We show that in this case testing the c-planarity can be done efficiently and give an efficient drawing algorithm. Also, we characterize 3-cluster cycles in terms of formal grammars. Finally, we generalize the results on 3-cluster cycles considering clustered graphs that at each level of the inclusion tree have a cycle structure. Even in this case we show efficient c-planarity testing and drawing algorithms.

8.2 M. Baur, U. Brandes, M. Gaertler, D. Wagner *Drawing the AS Graph in 2.5 Dimensions*

We propose a method for drawing AS graph data using 2.5D graph visualization. In order to bring out the pure graph structure of the AS graph we consider its core hierarchy. The k-cores are represented by 2D layouts whose interdependence for increasing k is displayed by the third dimension. For the core with maximum value a spectral layout is chosen thus emphasizing on the most important part of the AS graph. The lower cores are added iteratively by force-based methods. In contrast to alternative approaches to visualize AS graph data, our method illustrates the entire AS graph structure. Moreover, it is generic with regard to the hierarchy displayed by the third dimension.

8.3 S. Basu, R. Dhandapani, R. Pollack *On the realizable weaving patterns of polynomial curves in R^3*

We prove that the number of distinct weaving patters produced by n semi-algebraic curves in $\Re^3$ defined coordinate-wide by polynomials of degrees bounded by some constant d, is bounded by $2^{O(n \log n)}$, where the implied constant in the exponent depends on d. This generalizes a similar bound obtained by Pach, Pollack and Welzl [3] for the case when $d = 1$.

8.4 E. Di Giacomo, W. Didimo, G. Liotta, M. Suderman *Hamiltonian-with-handles Graphs and the k-spine Drawability Problem*

An embedded planar graph G is k-spine drawable, $k \geq 0$, if there exists an embedding-preserving planar drawing of G in which each vertex of G lies on one of k horizontal lines, and each edge of G is drawn as a polyline consisting of at most two line segments. In this paper we:

- Introduce the notion of hamiltonian-with-handles graphs and show that a planar embedded graph is 2-spine drawable if an only if it is hamiltonian-with-handles.
- Give examples of planar graphs that are/are not 2-spine drawable and present linear-time drawing techniques.
- Prove that deciding whether or not an embedded planar graph is 2-spine drawable is NP-Complete.
- Extend the study to k-spine drawings for $k > 2$, provide examples of non-drawable planar graphs, and show that the k-drawability problem remains NP-Complete for each fixed $k > 2$.

8.5 R. Ellis, X. Jia, J. Martin, C. Y *Random Geometric Graph Diameter in the Unit Disk with ℓ_p Metric*

Let n be a positive integer, and $\lambda > 0$ a real number. Let V_n be a set of n points randomly located within the unit disk, which are mutually independent. For $1 \leq p \leq \inf$, define $G_p(\lambda, n)$ to be the graph with the vertex set V_n, in which two vertices are adjacent if and only if their l_p-distance is at most λ. We call this graph a *unit disk random graph*. Let $\lambda = c\sqrt{\ln n/n}$ and let X be the number of isolated points in $G_p(\lambda, n)$. Let a_p be the (constant) ratio of the area of the l_p-ball to the l_2-ball of the same radius. Then, almost always, $X = 0$ when $c > a_p^{-1/2}$, and X $n^{1-a_p c^2}$. Penrose proved that with probability approaching 1, the graph $G_p(\lambda, n)$ is connected when it has a minimum degree 1. We extend Penrose's method to prove that if $G_p(\lambda, n)$ is connected, then there exists a constant K, independent of p, such that the diameter of $G_p(\lambda, n)$ is bounded above by K/λ. We show in addition that when c exceeds a certain constant depending on p, the diameter of $G_p(\lambda, n)$ is bounded above by $(2 \cdot 2^{1/2+1/p} + o(1))/\lambda$. More generally, there is a function $c_p(\delta)$ such that the diameter is at most $2^{1/2+1/p}(1 + \delta + o(1))/\lambda$ when $c > c_p(\delta)$.

9 Session 8

9.1 C. Papamanthou, I. Tollis, M. Do *3D Visualization of Semantic Metadata Models and Ontologies*

We propose an algorithm for the 3D visualization of general ontology models used in many applications, such as semantic web, entity - relationship diagrams and other database models. The visualization places entities in the 3D space. Previous techniques produce drawings that are 2-dimensional, which are often complicated and hard to comprehend. Our technique uses the third dimension almost exclusively for the display of the isa relationships (links) while the property relationships (links) are placed on some layer (plane), thus emphasizing the semantic difference between isa links and property links, which should be as vertical or as horizontal as possible respectively. Special reference is made on a certain model, the CIDOC Conceptual Reference Model.

9.2 M. Bekos, M. Kaufmann, A. Symvonis, A. Wolff *Boundary Labeling: Models and Efficient Algorithms for Rectangular Maps*

In this paper, we present boundary *labeling*, a new approach for labeling point sets with large labels. We first place disjoint labels around an axis-parallel rectangle that contains the points. Then we connect each label to its point such that no two connections, so-called leaders, intersect. Such an approach is common e.g. in technical drawings and medical atlases, but has so far not been studied much in the literature. The new problem is interesting in that it is a mixture of a label-placement and a graph-drawing problem. We consider placing labels at one, two or all four sides of the rectangle.We investigate different types of leaders like rectilinear and straight-line. We present simple and efficient algorithms that minimize the total length of the leaders or, in the case of rectilinear leaders, the total number of bends.

9.3 C. Goerg, P. Mirke, M. Pohl, S. Diehl *Dynamic Graph Drawing of Sequences of Orthogonal and Hierarchical Graphs*

In this paper we introduce two novel algorithms for drawing sequences of orthogonal and hierarchical graphs while preserving the mental map. Both algorithms can be parameterized to trade layout quality for dynamic stability. In particular, we had to develop new metrics which work upon the intermediate results of layout phases. We discuss some properties of the resulting animations by means of examples.

9.4 U. Brandes, C. Pich
GraphML Transformation

The efforts put into XML-related technologies have exciting consequences for XML-based graph data formats such as GraphML. We here give a systematic overview of the possibilities offered by XSLT style sheets for processing graph data, and illustrate that many basic tasks required for tools used in graph drawing can be implemented by means of style sheets, which are convenient to use, portable, and easy to customize.

9.5 S. Hachul, M. Junger
Drawing Large Graphs with a Potential-Field-Based Multilevel Algorithm

Force-directed graph drawing algorithms are widely used for drawing general graphs. However, these methods do not guarantee a sub-quadratic running time in general. We present a new force-directed method that is based on a combination of an efficient multilevel scheme and a strategy for approximating the repulsive forces in the system by rapidly evaluating potential fields. Given a graph $G = (V, E)$, the asymptotic worst case running time of this method is $O(|V| \log |V| + |E|)$ with linear memory requirements. In practice, the algorithm generates nice drawings of graphs containing 100000 nodes in less than 5 minutes. Furthermore, it clearly visualizes even the structures of those graphs that turned out to be challenging for some other methods.

10 Session 9

10.1 C. Erten, S. Kobourov
Simultaneous Embedding of Planar Graphs with Few Bends

We present an $O(n)$ time algorithm for simultaneous embedding of pairs of planar graphs on the $O(n^2) \times O(n^2)$ grid, with at most three bends per edge, where n is the number of vertices. For the case when the input graphs are both trees, only one bend per edge is required. We also describe an $O(n)$ time algorithm for simultaneous embedding with fixed-edges for tree-path pairs on the $O(n) \times O(n^2)$ grid with at most one bend per tree-edge and no bends along path edges.

10.2 A. Por, D. Wood
No-three-in-line-in-3D

The *no-three-in-line* problem, introduced by Dudeney in 1917, asks for the maximum number of points in the $n \times n$ grid with no three points collinear. In 1951, Erdös proved that the answer is $\Theta(n)$. We consider the analogous three-dimensional problem, and prove that the maximum number of points in the $n \times n \times n$ grid with no three collinear is $\Theta(n^2)$. This result is generalized by the notion of a *3D drawing* of a graph. Here each vertex is represented by a distinct gridpoint in Z^3, such that the line-segment representing each edge does not intersect any vertex, except for its own endpoints. Note that edges may cross. A 3D drawing of a complete graph K_n is nothing more than a set of n gridpoints with no three collinear. A slight generalization of our first result is that the minimum volume for a 3D drawing of K_n is $\Theta(n^{3/2})$. This compares favorably to $\Theta(n^3)$ when edges are not allowed to cross. Generalizing the construction for K_n, we prove that every k-colourable graph on n vertices has a 3D drawing with $O(n\sqrt{k})$ volume. For the k-partite Turán graph, we prove a lower bound of $\Omega((kn)^{3/4})$.

10.3 S. Aziza, T. Biedl
Hexagonal Grid Drawings: Algorithms and Lower Bounds

In this paper we study drawings of graphs of maximum degree six on the hexagonal (triangular) grid, with the main focus of keeping the number of bends small. We give algorithms that achieve $3.5n + 3.5$ bends for all simple graphs. We also prove optimal lower bounds on the number of bends for drawing K_7 and some other small graphs, which we then use to prove asymptotic lower bounds for graph classes of varying connectivity.

10.4 M. Patrignani

A Note on the Self-Similarity of Some Orthogonal Drawings

Large graphs are difficult to browse and to visually explore. This note adds up evidence that some graph drawing techniques, which produce readable layouts when applied to medium-size graphs, yield self-similar patterns when launched on huge graphs. To prove this, we consider the problem of assessing the self-similarity of graph drawings, and measure the box-counting dimension of the output of three algorithms, each using a different approach for producing orthogonal grid drawings with a reduced number of bends.

11 Session 10

11.1 B. Finkel, R. Tamassia
Curvilinear Graph Drawing Using the Force-Directed Method

We present a method for modifying a force-directed graph drawing algorithm into an algorithm for drawing graphs with curved lines. Our method is based on embedding control points as dummy vertices so that edges can be drawn as splines. Our experiments show that our method yields aesthetically pleasing curvilinear drawing with improved angular resolution. For example, applying our method to the GEM algorithm on the test suite of the "Rome graphs" resulted in an average improvement of 46% in angular resolution.

11.2 D. Forrester, S. Kobourov, A. Navabi, K. Wampler, G. Yee
System for Generalized Force-Directed Layouts

The graphael system implements several traditional force-directed layout methods, as well as several novel layout methods for non-Euclidean geometries, including hyperbolic and spherical. The system can handle large graphs, using multi-scale variations of the force-directed methods. Moreover, graphael can layout and visualize graphs that evolve though time, using static views, animation, and morphing. The implementation includes a powerful interface that allows the user to put together existing algorithms and visualization techniques, and to easily add new ones. The system is written in Java and is available as a downloadable program or as an applet at http graphael.cs.arizona.edu.

11.3 S. Kobourov, C. Pitta
An Interactive Multi-User System for Simultaneous Graph Drawing

In this paper we consider the problem of simultaneous draw- ing of two graphs. The goal is to produce aesthetically pleasing drawings for the two graphs by means of a heuristic algorithm and with human assistance. Our implementation uses the DiamondTouch table, a multi- user, touch-sensitive input device, to take advantage of direct physical interaction of several users working collaboratively. The system can be downloaded at http dt.cs.arizona.edu where it is also available as an applet.

11.4 S. Hong, T. Murtagh
Visualization of Large and Complex Networks Using PolyPlane

This paper discusses a new method for visualization of large and complex networks in three dimensions. In particular, we focus on visualizing the *core tree structure* of the large and complex network. The algorithm uses the concept of subplanes, where a set of subtrees is laid out. The subplanes are defined using regular polytopes for easy navigation. The algorithm can be implemented to run in linear time. We implemented the algorithm and the experimental results show that is produces nice layouts of large trees with up to ten thousands nodes. We further discuss how to extend this method for more general case.

11.5 S. Kobourov, R. Yusufov
Visualizing Large Graphs with Compound-Fisheye Views and Treemaps

Compound-fisheye views are introduced as a method for the display and interaction with large graphs. The method relies on a hier- archical clustering of the graph, and a generalization of the traditional fisheye view, together with a treemap representation of the cluster tree.

11.6 P. Holleis, F. Brandenburg
QUOGGLES: Query On Graphs - a Graphical Largely Extensible System

We describe the query and data processing language quoggles which is particularly designed for the application on graphs. It uses a pipeline-like technique known from command line processing, and composes its queries as directed acyclic graphs. The main focus is on the extensibility and the ease of use. The language permits queries that select a distinguished subgraph, e.g., the set of all green nodes with degree at least d or the set of edges whose endnodes have a neighbor which has exactly one neighbor. It is sql complete, however, it cannot describe paths of arbitrary length; otherwise NP-hard problems like Hamilton path could directly be expressed. quoggles also enables the user to concatenate queries with algorithms, e.g. with graph drawing algorithms, which are then applied to the selected subgraph.

11.7 U. Dogrusoz, E. Giral, A. Cetintas, A. Civril, E. Demir
A Compound Graph Layout Algorithm for Biological

We present a new, elegant algorithm for compound graph layout. The algorithm is based on the traditional force-directed layout scheme with extensions to handle nesting and other application-specific constraints. The algorithm has been successfully implemented as part of a pathway integration and analysis toolkit named Patika for drawing complicated biological pathways with compartmental constraints and arbitrary nesting relations to represent molecular complexes and various types of pathway abstractions. Experimental results show that the execution time and quality of the produced drawings with respect to commonly accepted layout criteria and pathway conventions are quite satisfactory.

11.8 S. Hong, D. Merrick, H. Nascimento *The Metro Map Layout Problem*

We initiate a new problem of *automatic metro map layout. In general, a metro map consists of a set of lines which have intersections or overlaps. We define a set of aesthetic criteria for good metro map layouts and present a method to produce such layouts automatically. Our method uses a variation of the spring algorithm with a suitable preprocessing step. The experimental results with real world data sets show that our method produces good metro map layouts quickly.*

Index