A Survey of Graph Layout Problems

JOSEP DÍAZ, JORDI PETIT and MARIA SERNA
Universitat Politècnica de Catalunya

Graph layout problems are a particular class of combinatorial optimization problems whose goal is to find a linear layout of an input graph in such way that a certain objective cost is optimized. This survey considers their motivation, complexity, approximation properties, upper and lower bounds, heuristics and probabilistic analysis on random graphs. The result is a complete view of the current state of the art with respect to layout problems from an algorithmic point of view.

Categories and Subject Descriptors: F.1.3 [**Computation by abstract devices**]: Complexity Measures and Classes—*Reducibility and completeness*; F.2.2 [**Analysis of algorithms and problem complexity**]: Nonnumerical Algorithms and Problems—*Routing and layout*; G.2.2 [**Discrete Mathematics**]: Graph Theory

General Terms: Algorithms, Experimentation, Theory

Additional Key Words and Phrases: Approximation algorithms, complexity, embedding, heuristics, layout, parameterized complexity, random graphs.

CONTENTS

This research was partially supported by the IST Programme of the EU under contract number IST-1999-14186 (ALCOM-FT) and by the Spanish CICYT project TIC2000-1970-CE.
Address: Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, Mòdul C6, Campus Nord, 08034 Barcelona.
E-mail: {diaz,jpetit,mjserna}@lsi.upc.es

1. INTRODUCTION

Graph layout problems are a particular class of combinatorial optimization problems whose goal is to find a linear layout of an input graph in such a way that a certain objective function is optimized. A linear layout is a labelling of the vertices of a graph with distinct integers. A large number of relevant problems in different domains can be formulated as graph layout problems. These include optimization of networks for parallel computer architectures, VLSI circuit design, information retrieval, numerical analysis, computational biology, graph theory, scheduling and archaeology. Most interesting graph layout problems are NP-hard and their decisional versions NP-complete, but, for most of their applications, feasible solutions with an almost optimal cost are sufficient. As a consequence, approximation algorithms and effective heuristics are welcome in practice.

Because of their importance, there exist plenty of results related with layout problems. Here we try to present a complete view of the current state of the art with respect to graph layout problems. Our focus is biased to algorithmic issues. Our purpose is twice-fold: on one hand, to present a global view up to the latest results; on the other to encourage the algorithmic community to work on this very exciting field of research. There exist other surveys that deal with several aspects of different graph layout problems, and this paper generously intersects with them; see for instance references [Chinn et al. 1982; Chung 1988; Monien and Sudborough 1990; Díaz 1992; Mohar and Poljak 1993; Bezrukov 1999; Lai and Williams 1999; Raspaud et al. 2000].

The road map for this survey is as follows: First of all, in Section 2, we formally define the layout problems we are interested in and state some basic results. An historical overview with motivations and applications for the study of layout problems are surveyed in Section 3. Section 4 presents NP-completeness results on layout problems and Section 5 presents some polynomial time solvable graphs classes. In Section 6 we survey fixed parameterized complexity results. Afterwards, in Section 7 we survey approximation algorithms. In Section 8 we consider random graphs as inputs to layout problems and show the performance of some approximation algorithms on them. In Sections 9 and 10 we present, respectively, several techniques to obtain upper and lower bounds and several heuristic methods to obtain good feasible solutions. Finally, in Section 11 we present several extensions to the basic model of graph layout problems. We close the paper with some concluding remarks, mainly open problems.

2. DEFINITIONS AND BASIC OBSERVATIONS

We start with the definition of several graph layout problems and associated concepts. This enables us to treat different layout problems using a unique framework. Finally, we present some basic results on layout problems.

The graph theoretic definitions and notations that we use, for the most part, conform to the standard ones in computer science. Unless otherwise mentioned, graphs are finite, undirected and without loops. Given a graph G, its vertex set is denoted by $V(G)$ and its edge set by $E(G)$. The notation uv stands for the undirected edge $\{u,v\}$. The degree of a vertex u in a graph G is denoted as $\deg(u) = \deg_G(u)$ and the maximal degree of G as $\Delta(G)$. The set of neighbors of a vertex u in G is denoted $\Gamma(u) = \Gamma_G(u) = \{v \in V(G) \ : \ uv \in E(G)\}$.

A *linear layout*, or simply a *layout*, of an undirected graph $G = (V, E)$ with $n = |V|$ vertices is a bijective function $\varphi \colon V \to [n] = \{1, \ldots, n\}$. A layout has also been called a *linear ordering* [Adolphson and Hu 1973], a *linear arrangement* [Shiloach 1979], a *numbering* [Chinn et al. 1982] or a *labeling* [Juvan and Mohar 1992] of the vertices of a graph. We denote by $\Phi(G)$ the set of all layouts of a graph G.

Given a layout φ of a graph $G = (V, E)$ and an integer i, we define the set $L(i, \varphi, G) = \{u \in V \ : \ \varphi(u) \leq i\}$ and the set $R(i, \varphi, G) = \{u \in V \ : \ \varphi(u) > i\}$. The *edge cut* at position i of φ is defined as

$$\theta(i, \varphi, G) = |\{uv \in E \ : \ u \in L(i, \varphi, G) \wedge v \in R(i, \varphi, G)\}|$$

and the *modified edge cut* at position i of φ as

$$\zeta(i, \varphi, G) = |\{uv \in E \ : \ u \in L(i, \varphi, G) \wedge v \in R(i, \varphi, G) \wedge \varphi(u) \neq i\}|.$$

The *vertex cut* or *separation* at position i of φ is defined as

$$\delta(i, \varphi, G) = |\{u \in L(i, \varphi, G) \ : \ \exists v \in R(i, \varphi, G) : uv \in E\}|.$$

Given a layout φ of G and an edge $uv \in E$, the *length* of uv on φ is

$$\lambda(uv, \varphi, G) = |\varphi(u) - \varphi(v)|.$$

To ease future reference, these measures are summarized in Table I.

A common way to represent a layout φ of a graph G is to align its vertices on a horizontal line, mapping each vertex u to position $\varphi(u)$, as shown in Figure 1. This graphical representation gives an easy understanding of the previously defined measures: By drawing a vertical line just after position i and before position $i+1$, the vertices at the left of the line belong to $L(i, \varphi, G)$ and the vertices at the right of the line belong to $R(i, \varphi, G)$. It is easy to compute the cut $\theta(i, \varphi, G)$ by counting the number of edges that cross the vertical line. The modified cut $\zeta(i, \varphi, G)$ counts all the edges in $\theta(i, \varphi, G)$ except those that have vertex $\varphi^{-1}(i)$ as endpoint. It is also easy to compute the separation $\delta(i, \varphi, G)$ by counting the number of vertices at the left of the vertical line that are joined with some vertex at the right of the vertical line. Finally, the length $\lambda(uv, \varphi, G)$ of an edge uv corresponds to the natural distance between its endpoint images.

Given a layout φ of a graph $G = (V, E)$, its *reversed layout* is denoted φ^R and is defined by $\varphi^R(u) = |V| - \varphi(u) + 1$ for all $u \in V$.

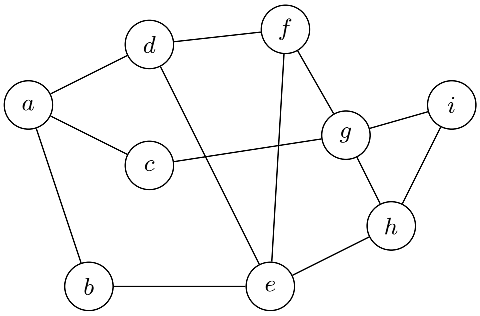
A *layout cost* is a function F that associates to each layout φ of a graph G an integer $F(\varphi, G)$. Let F be a layout cost; the optimization layout problem associated with F consists in determining some layout $\varphi^* \in \Phi(G)$ of an input graph G such that

$$F(\varphi^*, G) = \min_{\varphi \in \Phi(G)} F(\varphi, G).$$

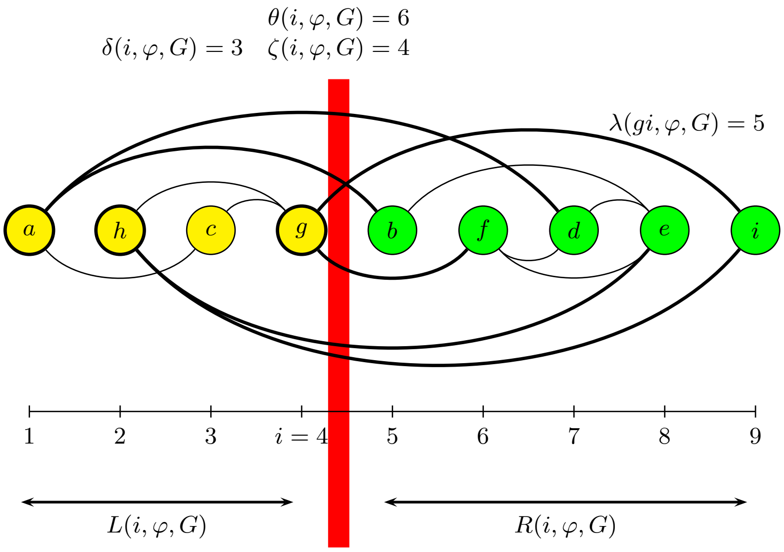
For any F and G, we define $\textsc{min}F(G) = \min_{\varphi \in \Phi(G)} F(\varphi, G)$.

The particular costs we consider in this survey are listed below, together with the layout problems they give raise to:

—*Bandwidth* (Bandwidth): Given a graph $G = (V, E)$, find a layout $\varphi^* \in \Phi(G)$



(a) $G = (V, E)$.



(b) Graphical representation of φ. Dividing the layout at $i = 4$, left vertices are shown in light yellow, right vertices in dark green, cut edges and separator vertices are bold.

Fig. 1. A graph G together with some layout measures and a graphical representation of the layout $\varphi = \{(a, 1), (b, 5), (c, 3), (d, 7), (e, 8), (f, 6), (g, 4), (j, 9), (h, 2)\}$.

such that $\textsc{bw}(\varphi^*, G) = \textsc{minbw}(G)$ where

$$\textsc{bw}(\varphi, G) = \max_{uv \in E} \lambda(uv, \varphi, G).$$

—*Minimum Linear Arrangement* (MinLA): Given a graph $G = (V, E)$, find a layout $\varphi^* \in \Phi(G)$ such that $\textsc{la}(\varphi^*, G) = \textsc{minla}(G)$ where

$$\textsc{la}(\varphi, G) = \sum_{uv \in E} \lambda(uv, \varphi, G).$$

—*Cutwidth* (Cutwidth): Given a graph $G = (V, E)$, find a layout $\varphi^* \in \Phi(G)$ such that $\textsc{cw}(\varphi^*, G) = \textsc{mincw}(G)$ where

$$\textsc{cw}(\varphi, G) = \max_{i \in [|V|]} \theta(i, \varphi, G).$$

—*Modified Cut* (ModCut): Given a graph $G = (V, E)$, find a layout $\varphi^* \in \Phi(G)$ such that $\textsc{mc}(\varphi^*, G) = \textsc{minmc}(G)$ where

$$\textsc{mc}(\varphi, G) = \sum_{i \in [|V|]} \zeta(i, \varphi, G).$$

—*Vertex Separation* (VertSep): Given a graph $G = (V, E)$, find a layout $\varphi^* \in \Phi(G)$ such that $\textsc{vs}(\varphi^*, G) = \textsc{minvs}(G)$ where

$$\textsc{vs}(\varphi, G) = \max_{i \in [|V|]} \delta(i, \varphi, G).$$

—*Sum Cut* (SumCut): Given a graph $G = (V, E)$, find a layout $\varphi^* \in \Phi(G)$ such that $\textsc{sc}(\varphi^*, G) = \textsc{minsc}(G)$ where

$$\textsc{sc}(\varphi, G) = \sum_{i \in [|V|]} \delta(i, \varphi, G).$$

—*Profile* (Profile): Given a graph $G = (V, E)$, find a layout $\varphi^* \in \Phi(G)$ such that $\textsc{pr}(\varphi^*, G) = \textsc{minpr}(G)$ where

$$\textsc{pr}(\varphi, G) = \sum_{u \in V} \left(\varphi(u) - \min_{v \in \Gamma^*(u)} \varphi(v) \right)$$

and $\Gamma^*(u) = \{u\} \cup \{v \in V \ : \ uv \in E\}$.

—*Edge Bisection* (EdgeBis): Given a graph $G = (V, E)$, find a layout $\varphi^* \in \Phi(G)$ such that $\textsc{eb}(\varphi^*, G) = \textsc{mineb}(G)$ where

$$\textsc{eb}(\varphi, G) = \theta(\left\lfloor \tfrac{1}{2}|V| \right\rfloor, \varphi, G).$$

—*Vertex Bisection* (VertBis): Given a graph $G = (V, E)$, find a layout $\varphi^* \in \Phi(G)$ such that $\textsc{vb}(\varphi^*, G) = \textsc{minvb}(G)$ where

$$\textsc{vb}(\varphi, G) = \delta(\left\lfloor \tfrac{1}{2}|V| \right\rfloor, \varphi, G).$$

Strictly speaking, the Edge Bisection and Vertex Bisection problems are not layout problems: Both problems ask for a partition of the set of vertices of the graph in two disjoint subsets of the same size (or differing by one if the number of vertices is odd) rather than for a permutation of the vertices. Nevertheless,

Table I. Layout measures for a layout φ of a graph $G = (V, E)$.

$L(i, \varphi, G)$	$=$	$\{u \in V \ : \ \varphi(u) \leq i\}$.
$R(i, \varphi, G)$	$=$	$\{u \in V \ : \ \varphi(u) > i\}$.
$\theta(i, \varphi, G)$	$=$	$\lvert\{uv \in E \ : \ u \in L(i, \varphi, G) \wedge v \in R(i, \varphi, G)\}\rvert$.
$\zeta(i, \varphi, G)$	$=$	$\lvert\{uv \in E \ : \ u \in L(i, \varphi, G) \wedge v \in R(i, \varphi, G) \wedge \varphi(u) \neq i\}\rvert$.
$\delta(i, \varphi, G)$	$=$	$\lvert\{u \in L(i, \varphi, G) \ : \ \exists v \in R(i, \varphi, G) : uv \in E\}\rvert$.
$\lambda(uv, \varphi, G)$	$=$	$\lvert\varphi(u) - \varphi(v)\rvert, \qquad uv \in E$.

Table II. Layout problems and costs for a graph $G = (V, E)$ with $|V| = n$.

Problem	Name	Cost
Bandwidth	Bandwidth	$\mathrm{BW}(\varphi, G) = \max_{uv \in E} \lambda(uv, \varphi, G)$.
Min. Lin. Arrangement	MinLA	$\mathrm{LA}(\varphi, G) = \begin{cases} \sum_{uv \in E} \lambda(uv, \varphi, G), \\ \sum_{i=1}^{n} \theta(i, \varphi, G). \end{cases}$
Cutwidth	Cutwidth	$\mathrm{CW}(\varphi, G) = \max_{i=1}^{n} \theta(i, \varphi, G)$.
Modified Cut	ModCut	$\mathrm{MC}(\varphi, G) = \sum_{i=1}^{n} \zeta(i, \varphi, G)$.
Vertex Separation	VertSep	$\mathrm{VS}(\varphi, G) = \max_{i=1}^{n} \delta(i, \varphi, G)$.
Sum Cut	SumCut	$\mathrm{SC}(\varphi, G) = \sum_{i=1}^{n} \delta(i, \varphi, G)$.
Profile	Profile	$\mathrm{PR}(\varphi, G) = \begin{cases} \sum_{u \in V} (\varphi(u) - \min_{v \in \Gamma^*(u)} \varphi(v)), \\ \mathrm{SC}(\varphi^R, G). \end{cases}$
Edge Bisection	EdgeBis	$\mathrm{EB}(\varphi, G) = \theta(\lfloor n/2 \rfloor, \varphi, G)$.
Vertex Bisection	VertBis	$\mathrm{VB}(\varphi, G) = \delta(\lfloor n/2 \rfloor, \varphi, G)$.

bisection problems and layout problems are closely related, and bisection problems fit well in our framework for layout problems.

The definitions of the previous problems are summarized in Table II. In Section 3 we will reference the first appearances of these problems.

At this point it is relevant to point out some basic but important facts.

Observation 2.1. For any graph $G = (V, E)$ and any layout φ of G, the total edge length equals the sum of all edge cuts in the layout:

$$\sum_{uv \in E} \lambda(uv, \varphi, G) = \sum_{i \in [|V|]} \theta(i, \varphi, G).$$

This fact was first noticed by Harper [1966]. It follows from the observation that any edge $uv \in E$ with $\varphi(u) < \varphi(v)$ contributes $\varphi(v) - \varphi(u)$ to the left hand side and 1 to each one of the terms $\theta(\varphi(u), \varphi, G), \theta(\varphi(u) + 1, \varphi, G), \ldots, \theta(\varphi(v), \varphi, G)$ in the right hand side.

Observation 2.2. For any graph $G = (V, E)$ and any layout φ of G,

$$\textsc{pr}(\varphi, G) = \textsc{sc}(\varphi^R, G).$$

This identity apparently has been well known for some time; a recent proof can be found in [Golovach and Fomin 1998]. It is due to the fact that each vertex $u \in V$ contributes one unit $\varphi(u) - \min_{v \in \Gamma^*(u)} \varphi(v)$ times to the sum cut in the reversed layout. Notice that, as a consequence, Profile and SumCut are equivalent problems.

Observation 2.3. It is important to stress that the graph layout problems we have formulated explicitly ask for the construction of a layout with an optimal cost, rather than the cost of an optimal layout:

> *"Given a graph G,* find *a layout $\varphi^* \in \Phi(G)$*
> *such that $F(\varphi^*, G) = \textsc{min}F(G)$."*

All the problems can however be restated as decisional problems, where the task is to decide whether or not a graph admits a layout with cost not greater than an integer given as part of the input:

> *"Given a graph G* and *an integer k,* is there *some layout $\varphi \in \Phi(G)$*
> *such that $F(\varphi, G) \leq k$?"*

We will not insist too much if we are considering the optimization or the decisional version of some problem, because they will be clearly differentiated by the context.

The following lemma gives useful relations between some layout costs. The proofs are straightforward.

Lemma 2.4. *Let G be any graph with n vertices and m edges, and let φ be any layout of G. Then,*

$$\begin{aligned}
\textsc{la}(\varphi, G) &\leq n \cdot \textsc{cw}(\varphi, G), & \textsc{minla}(G) &\leq n \cdot \textsc{mincw}(G),\\
\textsc{la}(\varphi, G) &\leq m \cdot \textsc{bw}(\varphi, G), & \textsc{minla}(G) &\leq m \cdot \textsc{minbw}(G),\\
\textsc{mc}(\varphi, G) &\leq \textsc{la}(\varphi, G), & \textsc{minmc}(G) &\leq \textsc{minla}(G),\\
\textsc{sc}(\varphi, G) &\leq n \cdot \textsc{vs}(\varphi, G), & \textsc{minsc}(G) &\leq n \cdot \textsc{minvs}(G),\\
\textsc{vs}(\varphi, G) &\leq \textsc{bw}(\varphi, G), & \textsc{minvs}(G) &\leq \textsc{minbw}(G),\\
\textsc{eb}(\varphi, G) &\leq \textsc{cw}(\varphi, G), & \textsc{mineb}(G) &\leq \textsc{mincw}(G),\\
\textsc{vb}(\varphi, G) &\leq \textsc{vs}(\varphi, G), & \textsc{minvb}(G) &\leq \textsc{minvs}(G),\\
\textsc{cw}(\varphi, G) &\leq \Delta(G) \cdot \textsc{bw}(\varphi, G), & \textsc{mincw}(G) &\leq \Delta(G) \cdot \textsc{minbw}(G).
\end{aligned}$$

The next lemma relates some layout costs of a graph in terms of its connected components. A proof for Bandwidth can be found in [Chvátalová et al. 1975]; for the remaining problems the results can be proven in a similar way.

LEMMA 2.5. *Let G be a graph and $G_1, \dots, G_k$ its connected components. Then,*

$$\mathrm{MINBW}(G) = \max_{i\in[k]} \mathrm{MINBW}(G_i), \qquad \mathrm{MINMC}(G) = \sum_{i\in[k]} \mathrm{MINMC}(G_i),$$

$$\mathrm{MINCW}(G) = \max_{i\in[k]} \mathrm{MINCW}(G_i), \qquad \mathrm{MINLA}(G) = \sum_{i\in[k]} \mathrm{MINLA}(G_i),$$

$$\mathrm{MINVS}(G) = \max_{i\in[k]} \mathrm{MINVS}(G_i), \qquad \mathrm{MINSC}(G) = \sum_{i\in[k]} \mathrm{MINSC}(G_i).$$

A useful consequence of the previous lemma is that, for any layout cost $F \in \{\mathrm{BW}, \mathrm{CW}, \mathrm{VS}, \mathrm{LA}, \mathrm{MC}, \mathrm{SC}\}$, it is possible to obtain an optimal layout for F of a graph just by computing the optimal layouts of its connected components. Observe, however, that the EB and VB bisection costs do not share this property: As a counterexample, consider a graph made of two components of the same size, each one being a clique.

The following lemma relates the layout costs of a graph and a subgraph of it. Given two graphs G and H, it is said that H is an edge induced graph of G if $V(H) = V(G)$ and $E(H) \subseteq E(G)$ and it said that H is a vertex induced graph of G if $V(H) \subseteq V(G)$ and $E(H) = \{uv \in E(G) \ : \ u, v \in V(H)\}$.

LEMMA 2.6. *Let H be an edge or vertex induced subgraph of a graph G. Then, for any layout cost $F \in \{\mathrm{BW}, \mathrm{CW}, \mathrm{VS}, \mathrm{LA}, \mathrm{MC}, \mathrm{SC}\}$, it holds that $\mathrm{MIN}F(H) \le \mathrm{MIN}F(G)$. In the case that H is an edge induced subgraph of G, then $\mathrm{MINEB}(H) \le \mathrm{MINEB}(G)$ and $\mathrm{MINVB}(H) \le \mathrm{MINVB}(G)$.*

3. HISTORICAL PERSPECTIVE AND APPLICATIONS

In its simpler form, a layout is an embedding into the natural line. As we will mention, the terms *layout* and *layout problem* are due to the early application of these problems to the optimal layout of circuits. We present here some background that motivates research on layout problems, as well as some of their applications. We start with a historical overview.

The Minimum Linear Arrangement problem (MINLA) was first stated in [Harper 1964]. Harper's aim was to design error-correcting codes with minimal average absolute errors on certain classes of graphs. Latter, this problem was also considered in [Mitchison and Durbin 1986] as an over-simplified model of some nervous activity in the cortex. MINLA also has applications in single machine job scheduling [Adolphson 1977; Ravi et al. 1991] and in graph drawing [Shahrokhi et al. 2001]. The Minimum Linear Arrangement problem has received some alternative names, as the *Optimal Linear Ordering*, the *Edge Sum*, the *Minimum-1-sum*, the *bandwidth sum* or the *wirelength* problem.

Bandwidth had received much attention during the fifties in order to speed up several computations on sparse matrices. According to Dewdney [1976], the introduction of the Bandwidth problem for graphs (BANDWIDTH) was first stated in [Harary 1967], however the problem was formally defined in [Harper 1966].

The Cutwidth problem (CUTWIDTH) was first used in the seventies as a theoretical model for the number of channels in an optimal layout of a circuit [Adolphson and Hu 1973]; see also the Introduction in [Makedon and Sudborough 1989]. More recent applications of this problem include network reliability [Karger 1999], automatic graph drawing [Mutzel 1995] and information retrieval [Botafogo 1993].

The Vertex Separation problem (VERTSEP) was originally motivated by the general problem of finding good separators for graphs [Lipton and Tarjan 1979], and has applications in algorithms for VLSI design [Leiserson 1980]. As we shall see, this problem is also equivalent to some other well known problems.

The Sum Cut and Profile problems on graphs (SUMCUT and PROFILE) were independently defined in [Díaz et al. 1991] and in [Lin and Yuan 1994b]. The Sum Cut problem was originally proposed as a simplified version of the δ-operator problem [Díaz 1979]. The Profile problem was proposed as a way to reduce the amount of storage of sparse matrices [Tewarson 1973; Lepin 1986; Lin and Yuan 1994a]. Both problems turn to be equivalent to the Interval Graph Completion problem [Ravi et al. 1991], which has applications in archaeology [Kendall 1969] and clone fingerprinting [Karp 1993]. The problem was rediscovered again in [Golovach 1997] under the name *total vertex separation number*.

The Edge Bisection problem (EDGEBIS) has a wide range of applications, notably in the area of parallel computing and VLSI [Bhatt and Leighton 1984; Shing and Hu 1986; Hromkovič and Monien 1992; Leighton 1993; Diekmann et al. 1994]. The Vertex Bisection problem (VERTBIS) is relevant to fault-tolerance and is related with the complexity of sending messages to processors in interconnection networks via vertex-disjoint paths [Klasing 1998].

In the remaining of this section, we present several further applications related to layout problems.

3.1 Layout problems in numerical analysis

In the area of numerical analysis, it is desirable for many engineering applications to reorder the rows and columns of very large sparse symmetric matrices in such a way that their non-zero entries lie as close as possible to the diagonal. Recall that a matrix is sparse when it has very few non-zero entries. Specifically, the *bandwidth* of a symmetric matrix M is the largest integer b for which there is a non-zero entry at $M[j, j+b]$ and the *profile* of M is $\sum_{i\in[n]}(i-p_i)$ where p_j is the index of the first non-zero entry of row j. Reducing the bandwidth and/or the profile of a matrix leads to a reduction of the amount of space needed for some storage schemes and to an improvement of the performance of several common operations such as Choleski factorization of non-singular systems of equations [Saad 1996]. The problem of reducing the bandwidth or the profile of a matrix M consists in finding a permutation matrix P such that $M' = P\cdot M\cdot P^T$ has minimal bandwidth or minimal profile. Recall that a permutation matrix P is an identity matrix with the same size of M whose columns have been permuted.

Observe that if we identify the non-zero entries of a symmetric matrix with the edges of a graph and the permutations of rows and columns with flips of the vertex labels, then the bandwidth of the graph equals the bandwidth of the matrix, and the profile of the graph equals the profile of the matrix.

The problem of reducing the bandwidth or the profile of an sparse symmetric matrix has a long history since it originated in the fifties; see for instance the references given in [Gibbs et al. 1976]. Nowadays, there exist general sparse methods that are more efficient than these "envelope schemes." However, many commercial packages still offer functions to reduce the bandwidth or the profile of sparse matrices as a preprocessing step. Thus, improvements in these methods can be ported to this software without a complete reorganization of their architecture [Barnard et al. 1995]. Efficient algorithms to perform several operations on matrices with small bandwidth can be found, for instance, in [Saad 1996]. Information retrieval to browse hypertext is a recent area where bandwidth and profile reduction techniques are also used [Botafogo 1993; Simon and Teng 1997].

3.2 Layout problems in VLSI

Many layout problems are originally motivated as simplified mathematical models of VLSI layout. Given a set of modules, the VLSI layout problem consists in placing the modules on a board in a non-overlapping manner and wiring together the terminals on the different modules according to a given wiring specification and in such a way that the wires do not interfere among them. There are two stages in VLSI layout: placement and routing. The *placement problem* consists in placing the modules on a board; the *routing problem* consists in wiring together the terminals on different modules that should be connected. A VLSI circuit can be modeled by the means of a graph, where the edges represent the wires and the vertices represent modules. Of course, this graph is an over-simplified model of the circuit, but understanding and solving problems in this simple model can help to obtain better solutions for the real-world model.

A first approach to solve the placement phase was to use the Minimum Linear Arrangement problem in order to minimize the total wire length [Harper 1970; Adolphson and Hu 1973]. Recently an alternative approach to solve this problem has been considered, it consists in finding recursively minimal cuts with minimal capacity among all cuts that separates the graph into two components of equal size. The Edge Bisection problem aims at this approach [Simon and Teng 1997]. Nowadays, current integrated circuit technology has changed substantially and some of the early applications of layouts are obsolete.

The cutwidth of a graph times the order of the graph gives a measure of the area needed to represent the graph in a VLSI layout when vertices are laid out in a row [Lengauer 1982]. In fact, Raspaud et al. [1995] prove a new relation between the cutwidth and the area of the VLSI layout of a graph: the minimal area of a VLSI layout of a graph is not less than the square of its cutwidth. A similar relation between area and edge bisection was well known for graphs with maximum degree 4 [Thompson 1979].

3.3 Layout problems in graph drawing

Perhaps one of the most important goals in graph drawing is to produce aestethic representations of graphs. Reducing the number of crossing edges is a way to improve the readability and comprehension of a graph. A bipartite drawing or 2-layer drawing is a graph representation where the vertices of a bipartite graph are placed in two parallel lines and the edges are drawn with straight lines between them. The bipartite crossing number of a bipartite graph is the minimal number of edge crossings over all bipartite drawings. In [Shahrokhi et al. 2001] it is proved that for a large class of bipartite graphs, reducing the bipartite crossing number is equivalent to reducing the total edge length, that is, to the Minimum Linear Arrangement problem. Moreover, an approximate solution of MinLA can be used to generate an approximate solution to the Bipartite Crossing Number problem. See Shahrokhi et al. [1997] for a bound and applications of crossing numbers and Vrťo [2002] for an extensive bibliography.

3.4 Layout problems as embedding problems

Linear arrangements are a particular case of embedding graphs in d-dimensional grids or other graphs. In its most general form, the embedding of a graph G into a host graph H consists in defining an injective function mapping the vertices of G to the vertices of H and associating a path in H for each edge of G. Three parameters are fundamental to assess the quality of an embedding: the dilation, the congestion and the load. The *dilation* of an embedding is the length of the largest associated path. The *congestion* of an embedding is the maximal number of paths that share an edge of H. The *load* of an embedding is the maximum number of vertices of G that are mapped to a vertex of H. Making use of good embeddings is essential in certain contexts, as in parallel computing where embeddings can be used to simulate an algorithm designed for one type of network on a parallel machine with a different type of network; see [Monien and Sudborough 1990] for a nice survey.

The case in which a graph with n vertices must be embedded into a path graph P_n of n vertices with load 1 is perhaps the simplest nontrivial embedding problem and has intensively been studied in the literature [Kendall 1969; Adolphson and Hu 1973; Adolphson 1977; Harper 1977; Robertson and Seymour 1985; Makedon and Sudborough 1989; Ravi et al. 1991; Botafogo 1993; Karp 1993; Leighton 1993; Saad 1996]. In this particular case, some layout problems and embedding problems are closely related. Specifically, the bandwidth of a graph corresponds to the minimal dilation and the cutwidth to the minimal congestion.

There exists other interesting embeddings on more general graphs than the path. For instance, Raspaud et al. [2000] present a survey on cyclic cutwidth and cyclic bandwidth, that is, when the graph is embedded into a cycle, rather than a path. Few results are known for other cyclic width parameters, Ching Guu solved the cyclic MinLA problem for the d-cube [Guu 1997]. Of practical importance are also the embeddings into grids [Miller 1991; Lin 1994; Bezrukov et al. 2000]. Several variants of embeddings into trees were studied in [Hu 1974; Johnson et al. 1978; Seymour and Thomas 1994; Ding and Oporowski 1995; Wu et al. 1999; Àlvarez et al. 2000].

3.5 Layout problems in parallel and distributed processing

Many parallel computers are made up of a set of processors with their own private memory that exchange messages by the way of a communication network. In order to get good speedups when using such a system, it is important to distribute the total amount of work between the processors as evenly as possible to minimize idle times. It is also important to reduce the amount of communication between the processors, because communicating through the network is much slower than the speed of the processors. Well established mapping and load balancing techniques have been developed to address these situations. In certain cases, these techniques lead to graph partitioning problems; see [Diekmann et al. 1994; Leighton 1993; Hromkovič and Monien 1992].

The Graph Partitioning problem consists in partitioning the vertices of a given graph in k sets of nearly same size in such a way that the number of cutting edges between the k sets is minimal. The Edge Bisection problem is a particular case of Graph Partitioning where $k = 2$. Recursive bisection is a popular technique to obtain partitions when k is a power of 2. See [Simon and Teng 1997] for an analysis of recursive bisection and see the references in [Bezrukov 1999] for more information on graph partitioning.

Edge Bisection can be of use when solving partial differential equations and finite elements methods in parallel systems. Simplifying, in these problems a particular iterative computational task has to be carried out in every vertex of a mesh (or the graph defined by the particular topology of the system) and its computation involves data from this vertex and in its neighbors. A way to distribute the total amount of computation between two processors is to assign to each one half of the vertices in the grid. But as border vertices need to communicate in order to get their operands, it is necessary to reduce the cut of the bisection.

3.6 Some equivalent problems

The Vertex Separation problem is strongly connected with several other important NP-complete problems: Gate Matrix Layout, Pathwidth and Vertex Search Number. The Bandwidth problem is also related to Proper Pathwidth.

The Gate Matrix Layout is a well studied problem with application in CMOS circuit design [Deo et al. 1987]. An instance of the *Gate Matrix Layout problem* consists in a collection of nets $\{N_1, \dots, N_n\}$ and their respective connection to a set of gates $\{G_1, \dots, G_m\}$. Nets are identified with rows and gates are identified with columns. The goal of the problem is to seek a permutation of the columns that minimizes the number of tracks required to lay out the chip, which is equivalent to minimize its area. Figure 2 shows an example of gate matrix layout. We denote by $\textsc{mingml}(G)$ the minimal number of tracks needed by a graph G.

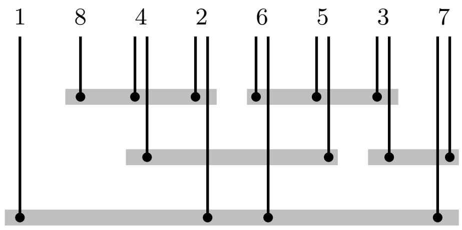
The Pathwidth problem has received a great interest in recent years due to its relation with the graph minors theory [Robertson and Seymour 1985]. A *path-decomposition* of a graph $G = (V, E)$ is a sequence of subsets of vertices $(X_1, \dots, X_r)$ such that

—$\bigcup_{i=1}^{r} X_i = V$;
—both endpoints of any edge $e \in E$ belong to some X_i for $1 \le i \le r$; and
—for all $i \le j \le k$, it is the case that $X_i \cap X_k \subseteq X_j$.

The pathwidth of a path decomposition $(X_1, \dots, X_r)$ is

$$\textsc{minpw}(G, (X_1, \dots, X_r)) = \max_{i \in [r]} |X_i| - 1.$$

The pathwidth of G, denoted $\textsc{minpw}(G)$, is the minimal pathwidth over all possible path decompositions of G. The Pathwidth problem (Pathwidth) consists in determining a path decomposition with minimal pathwidth. Figure 3 shows a



	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8
N_1	1	1	0	0	0	1	1	0
N_2	0	0	0	1	1	0	0	0
N_3	0	0	1	0	0	0	1	0
N_4	0	1	0	1	0	0	0	1
N_5	0	0	1	0	1	1	0	0

Fig. 2. Example of a gate matrix layout with three tracks (figure after Bodlaender).

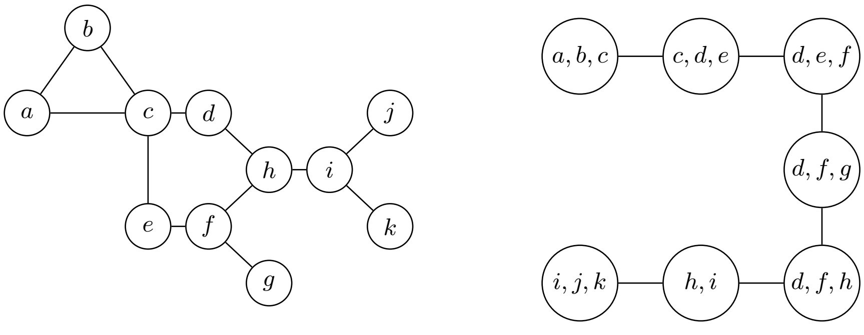


Fig. 3. Example of a graph and one of its path decompositions with pathwidth 2.

graph and one of its path decompositions. See [Bodlaender 1993] and [Downey and Fellows 1999] for more information on the Pathwidth problem.

The Vertex Search Number problem was introduced in [Kirousis and Papadimitriou 1986]. Briefly stated, this problem asks how many searchers are needed to capture an unlimited number of intruders moving around the edges of a given graph. We denote by $\text{MINSN}(G)$ the minimal vertex search number of graph G.

The equivalence between the Gate Matrix Layout, Search Number, Pathwidth and Vertex Separation problems is a consequence of the results in [Kirousis and Papadimitriou 1986; Kinnersley 1992; Fellows and Langston 1994]:

THEOREM 3.1. *For any graph G,*

$$\text{MINVS}(G) = \text{MINPW}(G) = \text{MINSN}(G) - 1 = \text{MINGML}(G) + 1.$$

Let $I_v = \{i \ : \ v \in X_i\}$. A *proper path decomposition* is a path decomposition that also satisfies $I_u \not\subset I_v$ for all $u, v \in V$. The proper pathwidth of G, denoted

MINPPW(G), is the minimal pathwidth over all possible proper path decompositions of G. The Proper Pathwidth problem consists in determining a proper path decomposition with minimal pathwidth. The proper pathwidth of a graph G can also be defined as the minimum cardinality of a maximum clique of a proper interval subgraph of G decreased by one.

The following equivalence between Bandwidth and Proper Pathwidth is due to Kaplan and Shamir [1996]:

THEOREM 3.2. *For any graph G,* MINPPW(G) = MINBW(G).

4. NP-COMPLETENESS RESULTS

It is widely believed that showing that a problem is NP-complete is equivalent to prove its computational intractability [Garey and Johnson 1979]. The following theorem indicates the difficulty of the considered layout problems on arbitrary graphs.

THEOREM 4.1. *The decisional versions of the layout problems* BANDWIDTH, MINLA, CUTWIDTH, MODCUT, VERTSEP, SUMCUT, EDGEBIS *are NP-complete.*

The reduction for BANDWIDTH is given in [Papadimitriou 1976], the proofs for MINLA and EDGEBIS are given in [Garey et al. 1976], for CUTWIDTH the completeness is due to [Gavril 1977], for MODCUT is given in [Monien and Sudborough 1988], for VERTSEP the proof is in [Lengauer 1981], and for SUMCUT it can be found in [Díaz et al. 1991; Lin and Yuan 1994b; Golovach 1997].

Many layout problems remain NP-complete even for certain restricted classes of graphs, Table III draws a synthetic overview on these results. For instance, BANDWIDTH is NP-complete even when restricting its inputs to trees with maximum degree three [Garey et al. 1978]. This result was improved in [Monien 1986] proving that BANDWIDTH remains NP-complete for caterpillars with hairs of length at most three, and for caterpillars with at most one unbounded hair attached to the backbone. The problem is also NP-complete for cyclic caterpillars with hair length 1 as reported in [Muradyan 1999]. Recall that a caterpillar is a particular class of tree made of a set of paths, called the hairs, attached by one of their leaves to the vertices of another path, called the backbone. A cyclic caterpillar has as backbone a cycle instead of a path.

The CUTWIDTH problem is NP-complete even for graphs with maximum degree three [Makedon et al. 1985]. This result was strengthened by showing that CUTWIDTH, VERTSEP and MODCUT are NP-complete even for planar graphs with maximum degree three [Monien and Sudborough 1988]. EDGEBIS is also NP-complete for d-regular graphs [Bui et al. 1987] and for graphs with maximal degree 3 [MacGregor 1978]. It is also known that MINLA is NP-complete when restricted to bipartite graphs [Even and Shiloach 1975]. The NP-completeness of VERTSEP for chordal graphs was proved in [Gustedt 1993], and for bipartite graphs in [Goldberg et al. 1995]. In [Díaz et al. 2001a] it is proved that BANDWIDTH, CUTWIDTH and VERTSEP remain NP-complete even when restricted to grid graphs or unit disk graphs. Finally, it is also known that SUMCUT is NP-complete when restricted to cobipartite graphs [Yuan et al. 1998].

Table III. Review of NP-completeness results for decisional graph layout problems.

Problem	NP-complete	Ref.
Bandwidth	in general	[Papadimitriou 1976]
	for trees with maximum degree 3	[Garey et al. 1978]
	for caterpillars with hair-length ≤ 3	[Monien 1986]
	for caterpillars with ≤ 1 hair per backbone vertex	[Monien 1986]
	for cyclic caterpillars with hair-length 1	[Muradyan 1999]
	for grid graphs and unit disk graphs	[Díaz et al. 2001a]
MinLA	in general	[Garey et al. 1976]
	for bipartite graphs	[Even and Shiloach 1975]
Cutwidth	in general	[Gavril 1977]
	for graphs with maximum degree 3	[Makedon et al. 1985]
	for planar graphs with maximum degree 3	[Monien and Sudborough 1988]
	for grid graphs and unit disk graphs	[Díaz et al. 2001a]
ModCut	for planar graphs with maximum degree 3	[Monien and Sudborough 1988]
VertSep	in general	[Lengauer 1981]
	for planar graphs with maximum degree 3	[Monien and Sudborough 1988]
	for chordal graphs	[Gustedt 1993]
	for bipartite graphs	[Goldberg et al. 1995]
	for grid graphs and unit disk graphs	[Díaz et al. 2001a]
SumCut	in general	[Díaz et al. 1991]
		[Lin and Yuan 1994b]
		[Golovach 1997]
	for cobipartite graphs	[Yuan et al. 1998]
EdgeBis	in general	[Garey et al. 1976]
	for graphs with maximum degree 3	[MacGregor 1978]
	for graphs with maximum degree bounded	[MacGregor 1978]
	for d-regular graphs	[Bui et al. 1987]

The complexity of most of the layouts problems for many families remains open. For instance, it is unknown if MinLA or SumCut are NP-complete for sparse graphs. Also the NP-completeness of MinLA and SumCut remain open for planar and/or for series-parallel graphs. The complexity of VertBis for the general case is unknown.

5. CLASSES OF GRAPHS WITH POLYNOMIAL TIME ALGORITHMS

NP-completeness results do not rule out the existence of efficient algorithms to get optimal solutions on particular classes of graphs. We review now this type of results for layout problems.

In the case of the MinLA problem, the optimal value of minla for hypercubes is known [Harper 1964] and there exists a closed formula for the value of the de Bruijn graph of order four [Harper 1970]. The motivation for the former case was to minimize the total edge length of the wires needed to connect a Viterbi decoder and the motivation for the latter case was to design error-correcting codes with minimal errors. In the case of a d-dimensional hypercube Q_d, $\textsc{minla}(Q_d) = 2^{d-1}(2^d - 1)$.

According to Chung [1988], Goldberg and Klipker were the first to give a $O(n^3)$ algorithm to solve the Minimum Linear Arrangement problem for trees [Goldberg and Klipker 1976]. Adolphson and Hu [1973] provide a $O(n \log n)$ algorithm to compute the MinLA of a rooted tree with n vertices is presented. Shiloach [1979] improves the result by presenting an algorithm to solve the MinLA on trees of n vertices in $O(n^{2.2})$ time. This was further improved in [Chung 1988], who gave a $O(n^{\log 3/\log 2})$ algorithm. The optimal value for the MinLA problem on a complete binary tree with k levels $T_{2,k}$ has an explicit expression discovered by Chung: For all $k \geq 2$,

$$\text{MINLA}(T_{2,k}) = 2^k(\tfrac{1}{3}k + \tfrac{5}{18}) + \tfrac{2}{9}(-1)^k - 2.$$

A recursive expression was also presented by Chung for the case of complete ternary trees.

With respect to parallel algorithms, Díaz et al. [1997] proved that MinLA for trees is in NC, as it can be solved in $O(\log^2 n)$ time using a CREW PRAM with $O(n^{3.6})$ processors; see [Greenlaw et al. 1995] for concepts on parallel complexity.

The MinLA problem on square or rectangular grids has a peculiar history. Let $L_{m,n}$ denote a $m \times n$ rectangular grid graph and let $L_n = L_{n,n}$ denote a $n \times n$ square grid graph. The problem was first solved by Muradyan and Piliposyan [1980] in 1980, in a paper written in Russian, for the general case of rectangular grids. Latter, in 1986, Mitchison and Durbin [1986] presented the solution for square grids only. In 1981, Niepel and Tomasta [1981] incorrectly conjecture that the lexicographic layout is optimal for $\text{MINLA}(L_m)$. In a paper published in 1994, Nakano [1994] references again this conjecture. Still in 2000, Fishburn et al. [2000] presented again a solution for $\text{MINLA}(L_{m,m'})$.

Indeed, the optimal layout for $\text{MINLA}(L_{m,m'})$ on $m \times m'$ rectangular grids has an interesting solution, which is well described in [Bezrukov 1999]. The optimal numbering is schematically shown in Figure 4. The numbering starts with the left lower corner of the grid and then fills the areas $A_1, A_2, \ldots, A_7$, where A_1, A_3 are $a \times a$ squares and A_5, A_7 are $a' \times a'$ squares; see Figure 4(a). The values of a and a' must satisfy

$$a, a' \in \left\{ \left\lceil m - \tfrac{1}{2} - \sqrt{\tfrac{1}{2}m^2 - \tfrac{1}{2}m + \tfrac{1}{4}} \right\rceil, \left\lfloor m + \tfrac{1}{2} - \sqrt{\tfrac{1}{2}m^2 - \tfrac{1}{2}m + \tfrac{1}{4}} \right\rfloor \right\}.$$

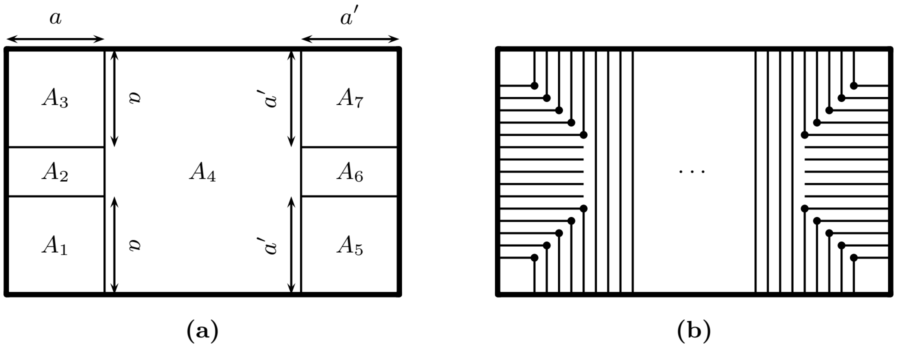
The way to number the areas is shown in Figure 4(b). Each square must be numbered sequentially. The square A_1 must be filled row after column. The square A_2 must be filled by consecutive rows, from bottom to top and from left to right. The square A_3 must be enumerated with the reverse order with respect to A_1. The square A_4 must be numbered by columns from bottom to top and from left to right. Finally, the squares A_5, A_6 and A_7 are filled in the same way. Using this solution, we get

$$\text{MINLA}(L_{m,m'}) = -\tfrac{2}{3}a^3 + 2ma^2 - \left(m^2 + m - \tfrac{2}{3}\right) a + m'(m^2 + m - 1) - m$$

and

$$\text{MINLA}(L_m) = \tfrac{4-\sqrt{2}}{3}m^3 + O(m^2).$$

The following theorem due to Muradyan and Piliposyan [1980; Mitchison and Durbin [1986; Bollobás and Leader [1991; Leighton [1993; Díaz et al. [2000; Fishburn et al. [2000] gathers the optimal values of layout problems for square grids:



definition[theorem]Definition observation[theorem]Observation

Fig. 4. Schematic representation of the optimal layout of $\text{MINLA}(L_{m,m'})$ for rectangular $m \times m'$ grids (figure reproduced from Muradyan:1980:MNV).

THEOREM 5.1. *Let L_m be an square grid of side m. Then,*

$$\text{MINVS}(L_m) = m, \text{MINCW}(L_m) = m + (\text{odd } m),$$
$$\text{MINSC}(L_m) = \tfrac{2}{3}m^3 + \tfrac{1}{2}m^2 - \tfrac{7}{6}m, \text{MINVB}(L_m) = m,$$
$$\text{MINEB}(L_m) = m + (\text{odd } m), \text{MINLA}(L_m) = \tfrac{1}{3}(4 - \sqrt{2})m^3 + O(m^2).$$

Optimal layouts for 5×5 square grids are shown in Figure 5. The lexicographic layout is optimal for VertSep, Bandwidth, EdgeBis and Cutwidth; the Muradyan–Piliposjan layout is optimal for MinLA; the diagonal layout is optimal for VertSep, VertBis, SumCut and Bandwidth.

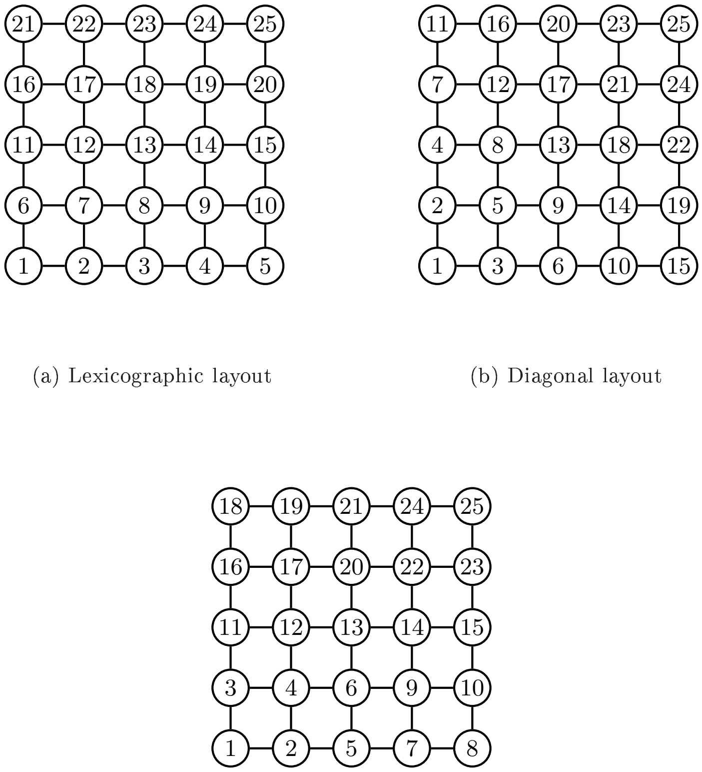
The exact results on VertBis, SumCut and Bandwidth are also known for other multi-dimensional grids and tori; see [Bezrukov 1999]. The VertSep problem for n-dimensional grids was solved in [Bollobás and Leader 1991]. Notice that nothing is known about the optimal solutions of grid graphs with holes, except for the Edge Bisection problem, for which Papadimitriou and Sideri [1996] gave a $O(n^5)$ algorithm. Finding a faster algorithm is interesting due to its practical relevance.

A d-dimensional c-ary clique is a graph with vertices labeled by integers from 0 to $c^d - 1$ and edges connecting vertices whose c-ary representation differ in one and only one digit. Lindsey [1964] solved the MinLA problem on this kind of graphs, another proof can be found in [Nakano 1994]. More exact MinLA results for several other particular classes of graphs have been identified; see [Bezrukov 1999] and references therein.

The Cutwidth problem has a very similar trajectory. Harper [1966] seems to be the one who first solved it for the case of hypercubes. Chung et al. [1982] presented a $O(n \log^{d-2} n)$ time algorithm for the cutwidth of trees with n vertices and with maximum degree d. Yannakakis [1985] improved that result by giving an algorithm to determine the cutwidth of a tree of n vertices in $O(n \log n)$ time. In the case of a k-level t-ary tree $T_{t,k}$, it holds that

$$\text{MINCW}(T_{t,k}) = \left\lceil \tfrac{1}{2}(k-1)(t-1) \right\rceil, \qquad \forall k \geq 3.$$

Exact cutwidths of 2-dimensional grids and 3-dimensional grids have been found



(a) Lexicographic layout

(b) Diagonal layout

(c) Muradyan–Piliposjan layout

Fig. 5. Optimal layouts for 5×5 square grids.

by Rolim et al. [1995], who also present results for cylindrical and toroidal meshes. In particular, for $m, n \geq 2$, they prove that

$$\textsc{mincw}(L_{m,n}) = \begin{cases} 2, & \text{if } m = n = 2, \\ \min\{m+1, n+1\}, & \text{otherwise.} \end{cases}$$

On the other hand, Thilikos et al. [2001] present an algorithm to compute the cutwidth of bounded degree graphs with small treewidth in polynomial time.

With respect to parallel algorithms, Díaz et al. [1997] proved that an optimal layout for the cutwidth of a tree with n vertices and degree Δ can be computed in $O(\Delta \log^2 n)$ time using a CREW PRAM with $O(n^{3.6})$ processors. It is an open problem the parallel complexity of the Cutwidth problem for trees with unbounded degree.

For the SumCut or Profile problems, Lepin [1986] gave the first polynomial exact algorithm for trees. This result was improved by the linear time algorithm presented in [Díaz et al. 1991]. These authors also gave a parallel algorithm for computing the optimal Sum Cut layout of a tree with n vertices in $O(\log n)$ time using a CREW PRAM with $O(n^2 \log n)$ processors. Kuo and Chang [1994] also gave a sequential polynomial time algorithm for the same problem on trees.

In the case of the VertSep problem, Ellis et al. [1979] gave a linear algorithm to compute the optimal vertex separation of a tree, and a $O(n \log n)$ algorithm to find the optimal layout. Recently, a linear time algorithm to find the optimal layout has been presented in [Skodinis 2000]. Further polynomial time algorithms to compute the vertex separation of permutation graphs and cographs were given in [Bodlaender et al. 1995b] and [Bodlaender and Möhring 1993] respectively.

In the case of the Edge Bisection problem, Leighton [1993] showed how to minimize the bisection width of Cartesian products of paths of the same length, provided the length is even. Nakano [1994] solved the problem for odd lengths. Rolim et al. [1995] have determined the optimal edge bisection of ordinary, cylindrical and toroidal 2-dimensional grids and of ordinary and toroidal 3-dimensional grids. The following theorem states their results for $L_{m,n}$:

Theorem 5.2. *Let $L_{m,n}$ be a rectangular grid. For $2 \leq m \leq n$,*

$$\textsc{mineb}(L_{m,n}) = m + (\text{odd } n)$$

and, for $m \geq 2$, $n \geq 3$,

$$\textsc{mineb}(L_{m,n}) = \begin{cases} \min\{2m, n\}, & \text{if } m \text{ and } n \text{ are even,} \\ \min\{2m, n+2\}, & \text{if } m \text{ is odd and } n \text{ is even,} \\ \min\{2m+1, n\}, & \text{if } m \text{ is even and } n \text{ is odd,} \\ \min\{2m+1, n+2\}, & \text{if } m \text{ and } n \text{ are odd.} \end{cases}$$

A related result is considered in [Azizoğlu and Eğecioğlu 2002] for the bisection of d-dimensional generalized cylinders (products of d graphs, each of them being a path or a cycle). The bisection of the hypercube seems to have been solved by many people concurrently (see e.g., [Nakano 1994]). A closed formula for the edge bisection of cube-connected cycles graphs is given in [Manabe et al. 1984]. For the case of the trees, MacGregor [1978] gives an $O(n^3)$ algorithm. A parallel algorithm running in $O(\log^2 n \log\log n)$ time on a CRCW PRAM with $O(n^2)$ processors for the bisection of trees, based in MacGregor's algorithm is presented in [Goldberg and Miller 1988]. A $O(n^2)$ algorithm to compute optimal bisection of partial k-trees is given in [Soumyanath and Deogun 1990]. Recall that a partial k-tree is a graph with bounded treewidth.

In [Muradyan and Piliposyan 1980], the MinLA and Cutwidth problems are solved for complete p-partite graphs. The complete p-partite graph $K(N_1, \ldots, N_p)$ is defined as a graph whose vertex set can be partitioned into p, so that two vertices are adjacent if and only if they belong to different partitions. If the number of vertices in two different partition differ in at most one, a p-partite is called balanced. It should be noted also that $\textsc{minla}(K(N_1, N_2, \ldots, N_p))$ has a nested solution.

As previously mentioned, the Bandwidth problem is NP-complete when restricted to trees. However, in the case of a k-level t-ary tree $T_{t,k}$, it holds that

$$\textsc{minbw}(T_{t,k}) = \left\lceil \frac{t(t^{k-1}-1)}{2(k-1)(t-1)} \right\rceil .$$

The embedding of complete binary trees with optimal bandwidth was presented in [Heckmann et al. 1998]. There also exists a $O(n \log n)$ algorithm to determine the bandwidth of caterpillars with hairs of length at most two [Assman et al. 1981].

Other classes of graphs whose bandwidth can be computed efficiently are interval graphs [Muradyan 1986; Mahesh et al. 1991; Sprague 1994], butterflies [Lai 1997] and chain graphs [Kloks et al. 1998]. Recall that interval graphs are intersection graphs of a set of intervals over the real line, and that chain graphs are bipartite graphs $G = (X, Y, E)$ where there is an ordering $x_1, x_2, \ldots, x_{|X|}$ of X such that $\Gamma(x_1) \subseteq \Gamma(x_2) \subseteq \cdots \subseteq \Gamma(x_{|X|})$. The first paper containing a polynomial time algorithm for Bandwidth for interval graphs was [Muradyan 1986]. The second paper claiming the result was [Kratsch 1987]. The third algorithm was proposed in [Kleitman and Vohra 1990] and takes time $O(n\textsc{bw}(G))$. Flaws in the proof of Kratsch' algorithm were discovered independently in two papers that propose a correct algorithm [Mahesh et al. 1991] and [Sprague 1994]. The algorithm in [Mahesh et al. 1991], takes time $O(n^2)$ and is a corrected version of Kratsch' algorithm. The algorithm in [Sprague 1994] takes time $O(n \log n)$ and extends the result in [Kleitman and Vohra 1990]. SumCut is the other layout problem that has a polynomial time algorithm for interval graphs [Lin and Yuan 1994a].

Table IV summarizes which classes of graphs are known to be optimally solvable in polynomial time for graph layout problems.

Other than grids, several results are known for families of graphs that can be described applying basic composition operations on paths (P_n), cycles (C_n), trees (T_n), complete graphs (K_n) or complete bipartite graphs ($K_{n,m}$). The definitions of the operations are taken from [Lai and Williams 1999].

—The r-th power of a graph G is the graph G^r, with the same vertex set as G and an edge uv if $d(u, v) \leq r$ in G.

—The *sum* of $k \geq 2$ pairwise disjoint graphs $G_1, \ldots, G_k$, denoted as $G_1 + \cdots + G_k$, is the graph G with vertex set $\cup_{1 \leq i \leq k} V(G_i)$ and an edge uv if for some $i \neq j$ $u \in V(G_i)$ and $v \in V(G_j)$ or if for some i $u, v \in V(G_i)$ and $uv \in E(G_i)$.

—The *cartesian product* of two graphs G and H, denoted as $G \times H$, is the graph with vertex set $V(G) \times V(H)$ where (u_1, v_1) is adjacent to (u_2, v_2) if either $u_1u_2 \in E(G)$ and $v_1 = v_2$ or $v_1v_2 \in E(H)$ and $u_1 = u_2$.

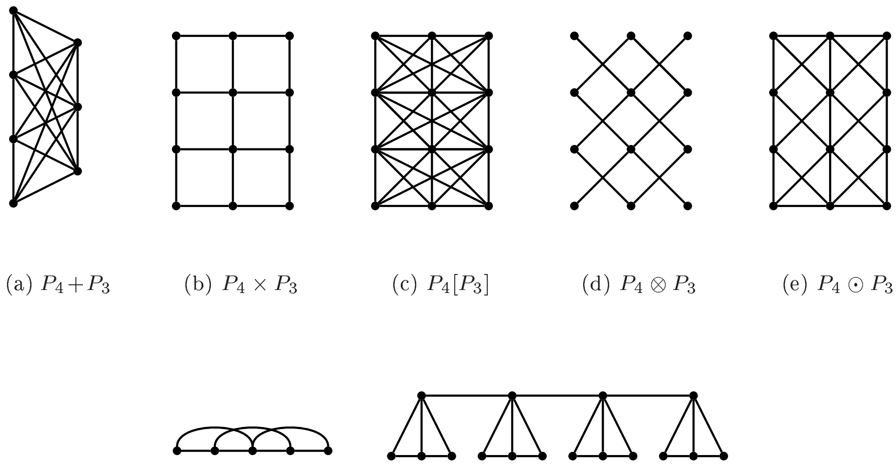
—The *composition* of two graphs G and H, denoted as $G[H]$, is the graph with vertex set $V(G) \times V(H)$ where (u_1, v_1) is adjacent to (u_2, v_2) if either $u_1u_2 \in E(G)$ or $v_1v_2 \in E(H)$ and $u_1 = u_2$.

—The *tensor product* of two graphs G and H, denoted as $G \otimes H$, is the graph with vertex set $V(G) \times V(H)$ where (u_1, v_1) is adjacent to (u_2, v_2) if $u_1u_2 \in E(G)$ and $v_1v_2 \in E(H)$.

—The *strong product* of two graphs G and H, denoted as $G \odot H$, is the graph with vertex set $V(G) \times V(H)$ where (u_1, v_1) is adjacent to (u_2, v_2) if one of the following holds: (a) $u_1u_2 \in E(G)$ and $v_1v_2 \in E(H)$, (b) $u_1 = u_2$ and $v_1v_2 \in E(H)$, or (c) $v_1 = v_2$ and $u_1u_2 \in E(G)$.

Table IV. Review of classes of graphs optimally solvable in polynomial time (n denotes the number of vertices in the graph, m its number of edges and Δ its maximal degree).

Problem	Class of graph	Complexity	Ref.
BANDWIDTH	caterpillars with hair-length ≤ 2	$O(n \log n)$	[Assman et al. 1981]
	hypercubes	$O(n \log n)$	[Harper 1966]
	butterflies	$O(n \log n)$	[Lai 1997]
	interval graphs	$O(n\Delta^2 \log \Delta)$	[Muradyan 1986]
	interval graphs	$O(n \log n)$	[Mahesh et al. 1991]
	interval graphs	$O(n \log n)$	[Sprague 1994]
	chain graphs	$O(n^2 \log n)$	[Kloks et al. 1998]
	complete k-level t-ary tree	$O(n)$	[Heckmann et al. 1998]
	square grids	$O(n)$	[Mai and Luo 1984] [Díaz et al. 2000]
MINLA	trees	$O(n^3)$	[Goldberg and Klipker 1976]
	rooted trees	$O(n \log n)$	[Adolphson and Hu 1973]
	trees	$O(n^{2.2})$	[Shiloach 1979]
	trees	$O(n^{\log 3/\log 2})$	[Chung 1988]
	rectangular grids	$O(n)$	[Muradyan and Piliposyan 1980]
	square grids	$O(n)$	[Mitchison and Durbin 1986]
	2-dimensional cylinder	$O(n)$	[Muradyan 1982]
	hypercubes	$O(n)$	[Harper 1964]
	de Bruijn graph of order 4	$O(n)$	[Harper 1970]
	d-dimensional c-ary cliques	$O(n)$	[Lindsey 1964]
	complete p-partite graphs	$O(n + p\log(p))$	[Muradyan and Piliposyan 1980]
CUTWIDTH	trees	$O(n \log^{\Delta-2} n)$	[Chung et al. 1982]
	trees	$O(n \log n)$	[Yannakakis 1985]
	hypercubes	$O(n)$	[Harper 1964]
	d-dimensional c-ary cliques	$O(n)$	[Nakano 1994]
	max degree $\leq \Delta$ and treewidth $\leq k$	$O(n^{\Delta k^2})$	[Thilikos et al. 2001]
	ordinary 2- and 3-dim. meshes	$O(n^2)$	[Rolim et al. 1995]
	toroidal and cylindrical 2-dim. meshes	$O(n^2)$	[Rolim et al. 1995]
	toroidal 3-dim. meshes	$O(n^2)$	[Rolim et al. 1995]
	complete p-partite graphs	$O(n + p\log(p))$	[Muradyan and Piliposyan 1980]
VERTSEP	trees	$O(n \log n)$	[Ellis et al. 1979]
	trees	$O(n)$	[Skodinis 2000]
	cographs	$O(n)$	[Bodlaender and Möhring 1993]
	permutation graphs	$O(n^2)$	[Bodlaender et al. 1995b]
	n-dimensional grids	$O(n^2)$	[Bollobás and Leader 1991]
SUMCUT	trees	$O(n^{2.3})$	[Lepin 1986]
	trees	$O(n)$	[Díaz et al. 1991]
	trees	$O(n^{1.722})$	[Kuo and Chang 1994]
	square grids	$O(n)$	[Díaz et al. 2000]
	interval graphs		[Lin and Yuan 1994a]
EDGEBIS	trees	$O(n^3)$	[MacGregor 1978]
	hypercubes	$O(n)$	[Nakano 1994]
	d-dimensional c-ary arrays	$O(n)$	[Nakano 1994]
	d-dimensional c-ary cliques	$O(n)$	[Nakano 1994]
	ordinary 2- and 3-dim. meshes	$O(n^2)$	[Rolim et al. 1995]
	toroidal and cylindrical 2-dim. meshes	$O(n^2)$	[Rolim et al. 1995]
	toroidal 3-dim. meshes	$O(n^2)$	[Rolim et al. 1995]
	grid graphs	$O(n^5)$	[Papadimitriou and Sideri 1996]
	treewidth $\leq k$	$O(n^2)$	[Soumyanath and Deogun 1990]
	cube-connected cycles graphs	$O(n)$	[Manabe et al. 1984]



(a) $P_4 + P_3$ (b) $P_4 \times P_3$ (c) $P_4[P_3]$ (d) $P_4 \otimes P_3$ (e) $P_4 \odot P_3$

(f) P_5^2 (g) $P_4 \wedge P_3$

Fig. 6. Examples of graph composition: sum (a), cartesian product (b), composition (c), tensor product (d), strong product (e), power (f) and corona (e).

—The *corona* of two graphs G and H is denoted as $G \wedge H$ and contains a copy of G and a copy of H for each vertex of G. Each vertex of G is connected to every vertex in the corresponding copy of H.

Figure 6 illustrates these definitions and Table V summarizes the known results.

The problem of computing the bandwidth of a *Hamming graph* is an interesting open problem, because it gives a measure of the effects of noise in the multi-channel transmission of data with that numbering [Berger-Wolf and Reingold 2002]. The best known result, due to Harper [2001], shows that the bandwidth of the Hamming graph is asymptotic to $\sqrt{\frac{2}{\pi\, d}} n^d$ as $d \to \infty$.

6. FIXED PARAMETER RESULTS

In defining a parameterization, the issue is not whether a problem is hard, but what makes the problem hard or easy to compute. To study the *structural hardness* of a difficult problem, the approach is to split the input into two parts: a *difficult* part (the non-parameterized) and an *easy* part (the parameterized), where we impose some restrictions. For several problems, it is known that the parameterization of the input does not break the NP-completeness barrier. A classical example is the *coloring problem* parameterized by the number k of colors that can be used. It is well known that the problem is NP-complete, for $k \geq 3$. On the other hand, graph problems like the *maximum independent set* or the *minimum vertex cover* become polynomially solvable, for every k, when we parameterize them by the size k of the maximum independent set or the minimum vertex cover.

Even in the cases where a parameterization of a NP-complete problem leads to a polynomial time algorithm, there are different type of upper bounds of the running

Table V. Review of families of composite graphs optimally solvable in polynomial time or with a closed formula. Graph names and operators are defined in the text

Problem	Class of graph	Ref.
BANDWIDTH	$G_1 + \cdots + G_k$	[Lai et al. 1994]
	$P_n \times \ldots^d \times P_n$	[Chvátalová 1975]
	$C_n \times C_m$	[Lai and Williams 1995]
	$K_n[G]$, $P_n[G]$, $C_n[G]$	[Hendrich and Stiebitz 1992]
	$K_{1,n}[G]$, $K_n[G]$ for $G \in \{P_m, T_m, C_m\}$	
	or $G = C_{n_1} \times \cdots \times C_{n_k}$ with $n_i \leq 5$	
	or $G = P_{n_1} \times \cdots \times P_{n_k}$ with $n_i > 1$	[Liu and Williams 1995]
	$P_n^r[G]$, $C_n^r[G]$	[Chinn et al. 1995]
	$P_n \times P_m[G]$, $P_n \times C_m[G]$ with $2n \neq m$	
	and $C_n \times C_m[G]$ $6 \leq 2n \leq 2s$	[Zhou and Yuan 1998]
	$K_n \odot P_m$	[Hendrich and Stiebitz 1992]
	$P_n \odot P_m$, $C_n \odot P_m$, $P_n \odot C_m$, $C_n \odot C_m$	[Lai and Williams 1995]
	$P_n \otimes P_m$, $C_n \otimes P_m$, $C_n \otimes C_m$	[Lai and Williams 1997]
	$P_k \otimes K_{nm}$	[Williams 1994]
	$P_k \otimes K_n$, $C_k \otimes K_n$	[Williams 1996]
	$K_n \wedge K_m$, $C_n \wedge K_1$, $P_n \wedge K_1$, $C_n \wedge (K_1 \cup \cdots \cup K_1)$	[Chinn et al. 1992]
MINLA	$G_1 + \cdots + G_k$ all G_i are sum deterministic	[Lai and Williams 1994]
	$P_n \times C_m$	[Muradyan 1982]
	$P_n[P_m]$, $P_n[C_m]$	[Liu 1992]
	$K_n[P_m]$, $K_n[C_m]$	[Liu and Williams 1995]
	$P_k \otimes K_{nm}$	[Williams 1994]
	$P_n \wedge P_m$, $K_n \wedge K_1$	[Williams 1993]
CUTWIDTH	C_n^r, $P_n \times P_m$, $P_n \times C_m$, $C_n \times C_m$, $K_n \times P_m$	
	and $K_n \times C_m$, $C_n^s \times C_m^r$, $K_n \times K_m$, $P_n \odot C_m$	[Liu and Yuan 1995]
SUMCUT	$G_1 + G_2$	[Lin and Yuan 1994b]
	$P_n \times K_m$, $C_n \times K_m$, $C_n \times C_m$	[Mai 1996]
	$K_n[G]$, $P_n[G]$, $C_n[G]$	[Lai 1997]
	$P_k \times K_{nm}$	[Lai 2001]
	$P_n \wedge G$, $C_n \wedge G$, $K_n \wedge G$, $K_{nm} \wedge G$	[Lai 1997]
	$G \wedge H$ where H is a 1-caterpillar, K_n or C_n	[Chang and Lai 2001]

time for the best known algorithm. For instance if the problem has k as a parameter, the running time could be $O(n^{f(k)})$ or it could be $O(f(k)n^\alpha)$ or it could be $O(f(k) + n^\alpha)$, where $f(k)$ is a function of k and $\alpha \in \mathbb{N}$ and the big-oh notations hides a constant independent of k. A parameterized problem is said to be *Fixed Parameter Tractable* if there exist an $O(f(k)n^\alpha)$-algorithm that solves the problem. The class FPT is the class of all fixed parameter tractable problems. Downey and Fellows defined a parameterized complexity hierarchy, the W-*hierarchy*. Similarly to the theory developed in Structural Complexity, the W-hierarchy consists of classes of parameterized problems, with different levels of parameterized complexity, between FPT and W [P]: $\mathsf{FPT} \subseteq \mathsf{W}[1] \subseteq \mathsf{W}[2] \subseteq \cdots \subseteq \mathsf{W}[\mathsf{P}]$, along with suitable notions of reducibility and completeness. The interested reader can have a look at [Downey and Fellows 1999] for a nice exposition of parameterized complexity. It is an open problem whether the inclusion in the hierarchy is strict. Moreover, there is evidence that if a problem is complete for some level of the W-hierarchy, then it is not expected to have an $O(f(k)n^\alpha)$-algorithm. For example, the parameterized *vertex cover problem* belongs to FPT [Balasubramanian et al. 1998], while the the parameterized *independent set problem* is known to be W[1]-complete [Downey and Fellows 1995].

Several parameterized complexity results are known for some layout problems. In these results the problems are parameterized by the minimum value of the layout measure. For the case of BANDWIDTH, we will consider the following parameterization for the BANDWIDTH problem:

BANDWIDTH(k): Given a graph G determine whether MINBW$(G) \leq k$.

We will use the same notation and the same parameterization for the remaining layout problems.

The earliest result proved that it is possible to decide the BANDWIDTH(2) problem and the CUTWIDTH(2) problem in linear time Garey et al. [1978]. Regarding the BANDWIDTH problem, Saxe [1980] presented a $O(n^{k+1})$ algorithm to decide BANDWIDTH(k) for any constant k. Gurari and Sudborough [1984] improved this result thanks to a $O(n^k)$ algorithm. This is essentially the best that can be done, as Bodlaender et al. [1994] proved that, for any k, the BANDWIDTH(k) problem is W[k]-hard. Recent work addresses the problem of simplifying the algorithm for the BANDWIDTH(2) problem, see [Makedon et al. 1993] for biconnected graphs and and [Caprara et al. 2002] for general graphs.

In the case of CUTWIDTH, [Gurari and Sudborough 1984] presented a $O(n^k)$ algorithm to decide the CUTWIDTH(k) problem for any input graph with n vertices and any constant k. This result was improved in [Makedon and Sudborough 1989] with a $O(n^{k-1})$ algorithm. Latter, Fellows and Langston [1988] obtained a $O(n^2)$ algorithm. The result for CUTWIDTH(k) has recently been improved in [Thilikos et al. 2000], where a linear time algorithm is presented. Recall that the treewidth notion is similar to the pathwidth notion, but for a tree decomposition rather than a path decomposition.

On the other hand, Fellows and Langston [1988] have proved that VERTSEP and MODCUT are fixed-parameter tractable. In particular, Bodlaender [1996] proved that VERTSEP(k) can be decided in linear time.

Furthermore, in [Bui and Peck 1992] it is shown that the EDGEBIS ($k \log n$) can be solved in polynomial time for planar graphs.

Table VI summarizes these results.

The fixed parameter complexity of the layout problems not included in Table VI remains an open problem. Furthermore, when considering the efficiency of the algorithms for parameterized problems, bear in mind that even if k is a constant, the multiplicative factor hidden in the big-oh notation is often exponential in k. This rises as an open problem the quest of practicable linear time algorithms.

7. APPROXIMATION ALGORITHMS

One of the approaches to deal with intractable problems is to design an approximation algorithm that in polynomial time will give a feasible solution "close" to the optimal one. In this section we precise this idea and present approximability results for layout problems. For a complete text on the theory of approximability the reader can look at [Garey and Johnson 1979; Ausiello et al. 1999; Vazirani 2001].

Table VI. Fixed parameterized complexity results for layout problems (n denotes the size of the graph and k the parameter).

Problem	Complexity	Ref.
BANDWIDTH(2)	$O(n)$	[Garey et al. 1978]
BANDWIDTH(k)	$O(n^{k+1})$	[Saxe 1980]
BANDWIDTH(k)	$O(n^k)$	[Gurari and Sudborough 1984]
BANDWIDTH(k)	W[k]	[Bodlaender et al. 1994]
CUTWIDTH(2)	$O(n)$	[Garey et al. 1978]
CUTWIDTH(k)	$O(n^k)$	[Gurari and Sudborough 1984]
CUTWIDTH(k)	$O(n^{k-1})$	[Makedon and Sudborough 1989]
CUTWIDTH(k)	$O(n^2)$	[Fellows and Langston 1992]
CUTWIDTH(k)	$O(n)$	[Thilikos et al. 2000]
MODCUT(k)	$O(n^2)$	[Fellows and Langston 1992]
VERTSEP(k)	$O(n^2)$	[Fellows and Langston 1988]
VERTSEP(k)	$O(n)$	[Bodlaender 1996]

Recall that, given a minimization problem Π, an $r(n)$-approximation algorithm is an algorithm that, for any input x of size n, finds a solution to Π whose cost is at most $r(n)$ times the cost of an optimal solution to the problem's instance. When a problem Π has some $r(n)$-approximation algorithm, it is said to be $r(n)$-*approximable*. When there exists an algorithm for Π such that, for all $\epsilon < 1$, A_ϵ returns a feasible solution σ such that the ratio between the obtained value and the optimal value is less than $1 + \epsilon$ and it runs in polynomial time with respect to $|x|$, A_ϵ is said to be a *polynomial time approximation scheme*. Moreover, when A_ϵ runs in polynomial time with respect to $|x|$ and $1/\epsilon$, A_ϵ is said to be a *fully polynomial time approximation scheme*. A combinatorial optimization problem belongs to the class APX if it is ϵ-approximable for some constant $\epsilon > 1$, to the class PTAS if it admits an approximation scheme and to the class FPTAS if it admits a fully approximation scheme. It is known that $\mathsf{FPTAS} \subseteq \mathsf{PTAS} \subseteq \mathsf{APX}$, where the inclusions are strict if and only if $\mathsf{P} \neq \mathsf{NP}$. There are also other parallel analogues to sequential approximation classes; see [Díaz et al. 1997] for an introduction to parallel approximation.

For the graph layout problems under consideration, the set of instances I corresponds to the set of all undirected graphs, an instance $x \in I$ corresponds to a particular undirected graph G, the set of feasible solutions $S(G)$ corresponds to $\Phi(G)$, and the objective function is a layout cost $f \in \{\text{LA}, \text{BW}, \text{SC}, \text{VS}, \text{CW}, \text{MC}, \text{EB}, \text{VB}\}$.

Many of the more recent positive results put the emphasis on improving the approximation bounds, rather than improving the time complexity. Two of the techniques used are relaxations of semidefinite programs and spreading metrics. A *spreading metric* on a graph is an assignment of lengths to its edges or its vertices, so that nontrivial subgraphs are spread apart in the associated metric space [Seymour 1995]. The volume of a spreading metric, defined as the sum of the lengths of all edges or vertices, provides a lower bound of solving the problem that guides a divide and conquer strategy. The drawback of the algorithms based on spreading metrics is that they require solving a linear program with an exponential number of constraints, which can make them impractical. In fact, their running time is dominated by the complexity of finding a spreading metric, which can be computed by general linear programming algorithms, as the Ellipsoid method. In general, most of them take $\tilde{O}(m^2n)$ time, where $\tilde{O}$ ignores poly-logarithmic factors; see [Even et al. 1999]. *Flow metrics* are a new approach to approximate some layout problems [Bornstein and Vempala 2002]. Even if flow metrics do not improve previous results, they represent a much simpler framework where the exponential number of constraints is reduced to a polynomial number.

For the Bandwidth problem, some particular kinds of graphs have approximation algorithms. In the case of γ-dense graphs, there exists a polynomial time 3-approximation, appeared in [Karpinski et al. 1997]. Recall that a graph with n vertices is γ-dense if its minimum degree is at least γn. The proof of this result uses the construction of perfect matchings in bipartite graphs as a building block together with a simplest 4-approximation algorithm. There is a 2-approximation algorithm to bandwidth for asteroidal triple free graphs, permutation graphs and trapezoidal graphs [Kloks et al. 1999]. There are also polynomial time $O(\log n)$-approximation algorithms for caterpillars [Haralambides et al. 1991], and for a larger class of trees, denoted as GHB-trees, which are characterized as trees such that for any node v, the depth difference of any two non-empty subtrees rooted at v is bounded by a constant [Haralambides and Makedon 1997]. For the case of general trees and chordal graphs, Gupta [2001] presents a randomized $O(\log^{2.5} n)$-approximation algorithm. For general graphs, Bandwidth has several polylogarithmic approximation algorithms running in polynomial randomized time due to Blum et al. [2000], using semidefinite-relaxation, to Feige [2000], using spreading metrics and volume respecting embeddings and to Dunagan and Vempala [2001] using semidefinite-relaxation and Euclidean embeddings. On the negative side, in [Blache et al. 1998] it is shown that it is NP-complete to find a $\frac{3}{2}$-approximation for general graphs, and it is also NP-complete to find a $\frac{4}{3}$-approximation for trees. As a consequence Bandwidth does not belong to PTAS. In fact, Unger [1998] proves that, for any constant k, it is NP-complete to find any k-approximation even for caterpillars. Therefore, Bandwidth does not belong to APX. The approximability of Bandwidth between a constant and a polylogarithmic factor remains open.

Recall that graphs with n vertices and m edges are dense if $m = \Theta(n^2)$. There exist fully polynomial time approximation schemes for dense graphs, Arora et al. [1996; Arora et al. [1999] for MinLA and Cutwidth and Frieze and Kannan [1996] for EdgeBis. The Regularity Lemma of Szemerédi is the base technique to prove these results.

Several approximation algorithms have been proposed for MinLA, Cutwidth and SumCut problems. The first nontrivial approximation algorithm for MinLA and Cutwidth on general graphs was a direct application of a $O(\log n)$ approximation algorithm for finding balanced partitions in a graph [Leighton and Rao 1999], which produced a $O(\log^2 n)$ approximation algorithm for MinLA and Cutwidth. The original paper appeared in as conference paper in 1989. Using ideas from the previous work, Hansen [1989] gave a $O(\log^2 n)$-approximation algorithm for MinLA. Using the spreading metrics technique Even et al. [2000] gave a $O(\log n \log\log n)$-approximation algorithms for MinLA and SumCut. Up to date, the best polynomial time approximation algorithms for MinLA and SumCut are $O(\log n)$-approximations for general graphs and $O(\log\log n)$-approximations for planar graphs. Both results appear in [Rao and Richa 1998], who also use the spreading metric technique. Their technique works for the general case of graphs with weighted edges.

In the case of the VertSep problem, Bodlaender et al. [1995a] presents a polynomial time $O(\log^2 n)$-approximation algorithm for general graphs, and shows how to use results from Seymour and Thomas [1994] to get a $O(\log n)$-approximation algorithm for planar graphs.

The first approximation algorithm for the EdgeBis problem on general graphs with sublinear approximation ratio was given in [Feige et al. 2002], this result was further improved in [Feige and Krauthgamer 2002], where a $O(\log^2 n)$ approximation ratio is obtained. For graphs with excluding any fixed graph as a minor (e.g. planar graphs), an improved ratio of $O(\log n)$ is obtained. Albeit its importance in many contexts, no better approximability results are currently known for this problem.

Table VII summarizes all these approximability results.

8. RANDOM GRAPHS

Worst-case analysis does not always catch the real difficulty of a problem. A standard way of evaluating from a practical point of view the *real* efficiency of an algorithm or heuristic is to evaluate its performance on random instances.

Recall that a sequence of events $(E_n)_{n\geq 1}$ occurs *with high probability* if $\Pr[E_n] \to 1$ and *with overwhelming probability* if $\Pr[E_n] \geq 1-2^{-\Omega(n)}$ for all n. Also, if $(X_n)_{n\geq 1}$ is a sequence of random variables and X is a random variable, it is said that X_n converges *in probability* to X ($X_n \xrightarrow{\Pr} X$) if $\Pr[|X_n - X| > \epsilon] \to 0$, for all $\epsilon > 0$.

8.1 Binomial random graphs

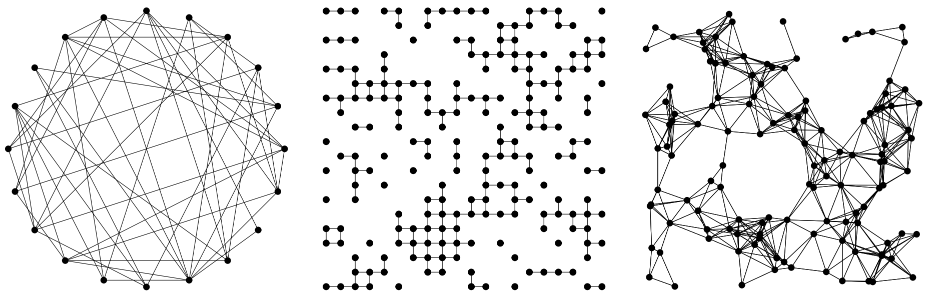
Given a positive integer n and a probability p_n, the class of *binomial random graphs*, $\mathcal{G}_{n,p_n}$, is a probability space over the set of undirected graphs $G = (V, E)$ on the node set $V = [n]$ determined by $\Pr[uv \in E] = p_n$, with these events mutually independent [Janson et al. 2000]. This well studied model, offers a natural starting point to analyze approximability properties of layout problems in a probabilistic setting.

Bandwidth can be approximated within a constant on binomial random graphs; see [Bollobás 1985, page 374] and [Turner 1986]. The same result for EdgeBis was obtained in [Boppana 1987]. These results can be extended to all layout problems considered in this survey using a unique framework involving "mixing graphs", which is a concept close to expander graphs developed in [Crescenzi et al. 2001]. The following result appears in [Díaz et al. 2001c].

Theorem 8.1. *Let $\epsilon \in (0, \frac{1}{6})$, $\gamma \in (0,1)$ and define $C_{\epsilon,\gamma} = 3(1+\ln 3)(\epsilon\gamma)^{-2}$. Consider a sequence $(c_n)_{n\geq 1}$ such that $C_{\epsilon,\gamma} \leq c_n \leq n$ for all $n \geq n_0$ for some natural n_0. Then, with overwhelming probability,* Bandwidth, MinLA, CutWidth, ModCut, SumCut, VertSep, EdgeBis *and* VertBis, *can be approximated within a $O(\epsilon+\gamma)$ factor on binomial random graphs $\mathcal{G}_{n,p_n}$ where $p_n = c_n/n$.*

Table VII. Review of approximability results for layout problems (n is the number of vertices of the input graph, m is its number of edges and ϵ is the parameter of the approximation scheme). Refer to the text for more information on the polynomial running times.

Problem	Approximability	Time	Ref.
BANDWIDTH	3-approximable for δ-dense graphs	$n^{O(1/\delta)}$	[Karpinski et al. 1997]
	2-approximable for AT-free graphs	$O(n^3)$	[Kloks et al. 1999]
	$O(\log n)$-approximable for caterpillars	$O(n^2)$	[Haralambides et al. 1991]
	$O(\log n)$-approximable for GHB-trees	$O(n^2)$	[Haralambides and Makedon 1997]
	random $O(\log^{4.5} n)$-approximable	$O(m(\log n)^4 \log\log n)$	[Feige 2000]
	random $O(\log^3 n\sqrt{\log\log n})$-approximable	$\mathrm{pol}(n)$	[Dunagan and Vempala 2001]
	random $O(\sqrt{n/\mathrm{BW}(G)}\log n)$-approximable	$\mathrm{pol}(n)$	[Blum et al. 2000]
	random $O(\sqrt{\log^{2.5} n})$-approx. for trees and chordal graphs	$\mathrm{pol}(n)$	[Gupta 2001]
	no PTAS	—	[Blache et al. 1998]
	no PTAS for trees	—	[Blache et al. 1998]
	no APX	—	[Unger 1998]
VERTSEP	$O(\log^2 n)$-approximable	$\mathrm{pol}(n)$	[Bodlaender et al. 1995a]
	$O(\log n)$-approximable for planar graph	$\mathrm{pol}(n)$	[Bodlaender et al. 1995a]
MINLA	PTAS for dense graphs	$n^{O(1/\epsilon^2)}$	[Arora et al. 1996]
	$O(\log^2 n)$-approximable	$\mathrm{pol}(n)$	[Hansen 1989; Leighton and Rao 1999]
	$O(\log n \log\log n)$-approximable	$\mathrm{pol}(n)$	[Even et al. 2000]
	$O(\log n)$-approximable	$\mathrm{pol}(n)$	[Rao and Richa 1998]
	$O(\log\log n)$-approximable for planar graphs	$\mathrm{pol}(n)$	[Rao and Richa 1998]
CUTWIDTH	PTAS for dense graphs	$n^{O(1/\epsilon^2)}$	[Arora et al. 1996]
	$O(\log^2 n)$-approximable	$\mathrm{pol}(n)$	[Leighton and Rao 1999]
SUMCUT	$O(\log n \log\log n)$-approximable	$\mathrm{pol}(n)$	[Even et al. 2000]
	$O(\log n)$-approximable	$\mathrm{pol}(n)$	[Rao and Richa 1998]
	$O(\log\log n)$-approximable for planar graphs	$\mathrm{pol}(n)$	[Rao and Richa 1998]
EDGEBIS	PTAS for dense graphs	$\tilde{O}(1/\epsilon^2)n + 2^{\tilde{O}(1/\epsilon^2)}$	[Frieze and Kannan 1996]
	$O(\log^2 n)$-approximable	$\mathrm{pol}(n)$	[Feige and Krauthgamer 2002]
	$O(\log n)$-approximable for planar graph	$\mathrm{pol}(n)$	[Feige and Krauthgamer 2002]



(a) Binomial random graph (b) Random grid graph (c) Random geometric graph

Fig. 7. Different models of random graphs.

Actually, the proof of the above theorem shows that *any* algorithm computing a feasible layout, no matter how good or bad, will perform rather well on random graphs, pointing out that evaluating heuristics on binomial random graphs may be unworthy for layout problems.

Theorem 8.1 has been improved for EdgeBis: Luczak and McDiarmid [2001] show that if $c > \ln 4$, then $\textsc{mineb}(\mathcal{G}_{n,c/n}) = \Theta(n)$ with high probability; while if $c < \ln 4$, then $\textsc{mineb}(\mathcal{G}_{n,c/n}) = 0$ with high probability.

8.2 Random grid graphs

Given an integer m and a probability p, the class of *random grid graphs*, $\mathcal{L}_{m,p}$, is a probability space over the set of grid graphs $G = (V, E)$ on the node set $V \subseteq [m]^2$ determined by $\Pr[u \in V] = p$ with these events mutually independent and $E = \{uv \ : \ u, v \in V \wedge \|u - v\|_1 = 1\}$.

The study of random grid graphs is based on *percolation* theory [Grimmett 1999]. Consider a *site percolation* process, where nodes from the infinite grid ($\mathbb{Z}^2$) are selected with some probability p. Let C_0 be the connected component where the origin belongs. A basic question in percolation theory is whether or not C_0 can be infinite. Let $\vartheta(p)$ denote the probability that $|C_0| = \infty$ and set $p_c = \inf\{p \ : \ \vartheta(p) > 0\}$, the *critical value* of p. In the *subcritical* limiting regimes $p \in (0, p_c)$, all components are almost surely finite.

The following theorem from Díaz et al. [2000] states the behavior of some layout problems on subcritical random grid graphs:

Theorem 8.2. *Let $p \in (0, p_c)$; then there exist constants $0 < c_1 < c_2$ such that*

$$\lim_{m\to\infty} Pr\left[c_1 \le \frac{\textsc{vs}(\mathcal{L}_{m,p})}{\sqrt{\log m}} \le \frac{\textsc{cw}(\mathcal{L}_{m,p})}{\sqrt{\log m}} \le c_2\right] = 1$$

and there exist two constants $\beta_{\textsc{la}}(p) > 0$ and $\beta_{\textsc{sc}}(p) > 0$ such that, as $m \to \infty$,

$$\textsc{la}(\mathcal{L}_{m,p})/m^2 \xrightarrow{\Pr} \beta_{\textsc{la}}(p) \qquad \textit{and} \qquad \textsc{sc}(\mathcal{L}_{m,p})/m^2 \xrightarrow{\Pr} \beta_{\textsc{sc}}(p).$$

Results for the supercritical case $p \in (p_c, 1)$ are derived in Penrose [2000]:

THEOREM 8.3. *Let $p \in (p_c, 1)$. Then, there exist a constant $c > 0$ such that, with overwhelming probability,*

$$\begin{aligned} cm^2 &\le \textsc{minsc}(\mathcal{L}_{m,p}) \le m^2, & cm^2 &\le \textsc{minla}(\mathcal{L}_{m,p}) \le 4m^2,\\ cm &\le \textsc{minbw}(\mathcal{L}_{m,p}) \le m, & cm &\le \textsc{minvs}(\mathcal{L}_{m,p}) \le m,\\ cm &\le \textsc{mincw}(\mathcal{L}_{m,p}) \le 4m. \end{aligned}$$

Moreover, for all p such that $\vartheta(p) > \frac{1}{2}$, with overwhelming probability,

$$cm \le \textsc{mineb}(\mathcal{L}_{m,p}) \le 4m.$$

8.3 Random geometric graphs

Random geometric graphs $\mathcal{G}(n; r_n)$ are graphs whose n nodes are n points uniformly distributed in the unit square and whose edges between any pair of nodes exist when their distance is smaller than some parameter r_n. Random geometric graphs have been proposed as a possible model to take into account the structural characteristics of instances that appear in many practical applications [Johnson et al. 1989; Berry and Goldberg 1999; Lang and Rao 1993].

As random grid graphs, random geometric graphs also exhibit a phase transition Penrose [1995]: When $nr_n^2 \to \lambda$ there exists a critical parameter λ_c such that if $\lambda < \lambda_c$, graphs $\mathcal{G}(n; r_n)$ are likely to have $O(\log n)$ points in each connected component, while if $\lambda > \lambda_c$, there is likely to be a component with $\Theta(n)$ nodes. The next two theorems, proven in [Díaz et al. 2000] and [Penrose 2000] respectively, characterize the behavior of several layout costs in the subcritical and supercrital phases.

THEOREM 8.4. *Suppose $nr_n^2 \to \lambda \in (0, \lambda_c)$. Then, there exist constants $0 < c_1 < c_2$, $0 < c_3 < c_4$, $\tilde\beta_{\text{LA}}(\lambda) > 0$ and $\tilde\beta_{\text{SC}}(\lambda) > 0$ such that*

$$\textsc{mineb}(\mathcal{G}(n; r_n)) \xrightarrow{\Pr} 0, \qquad \textsc{minvb}(\mathcal{G}(n; r_n)) \xrightarrow{\Pr} 0,$$

$$\textsc{minla}(\mathcal{G}(n; r_n))/n \xrightarrow{\Pr} \tilde\beta_{\text{LA}}(\lambda), \qquad \textsc{minsc}(\mathcal{G}(n; r_n))/n \xrightarrow{\Pr} \tilde\beta_{\text{SC}}(\lambda),$$

$$\mathit{Pr}\left[c_1 \le \frac{\textsc{vs}(\mathcal{G}(n; r_n))}{\log n / \log\log n} \le c_2\right] \to 1, \quad \mathit{Pr}\left[c_3 \le \frac{\textsc{cw}(\mathcal{G}(n; r_n))}{(\log n / \log\log n)^2} \le c_4\right] \to 1.$$

THEOREM 8.5. *Suppose $nr_n^2 \to \lambda \in (\lambda_c, \infty)$. Then,*

$$\begin{aligned} \textsc{mincw}(\mathcal{G}(n; r_n)) &= \Theta(n^2 r_n^3), & \textsc{mineb}(\mathcal{G}(n; r_n)) &= \Theta(n^2 r_n^3),\\ \textsc{minsc}(\mathcal{G}(n; r_n)) &= \Theta(n^2 r_n), & \textsc{minla}(\mathcal{G}(n; r_n)) &= \Theta(n^3 r_n^3),\\ \textsc{minvs}(\mathcal{G}(n; r_n)) &= \Theta(n r_n), & \textsc{minbw}(\mathcal{G}(n; r_n)) &= \Theta(n r_n). \end{aligned}$$

In the case that the radius is slightly greater, Díaz et al. [2001a] show that it is possible to obtain nice approximations:

THEOREM 8.6. *Let $(r_i)_{i \ge 1}$ be a sequence of positive numbers with $r_n \to 0$ and $nr_n^2/\log n \to \infty$. Then, with high probability,* BANDWIDTH, MINLA, CUTWIDTH, MODCUT, SUMCUT, VERTSEP, EDGEBIS *and* VERTBIS *can be approximated within a constant on random geometric graphs $\mathcal{G}(n; r_n)$. Moreover, for* BANDWIDTH *and* VERTSEP, *the approximation factor can be arbitrarily close to 1.*

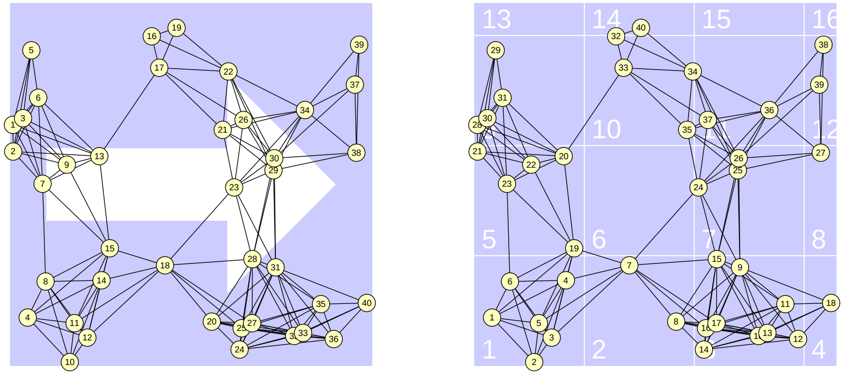


Fig. 8. Illustration of the projection (left) and dissection (right) algorithm. The Projection algorithm creates a layout by ordering the vertices according to their projection onto the x-axis. The Dissection algorithm creates a layout dissecting the unit square in boxes of the appropriate size and enumerating the points, following the order of the boxes in lexicographic order (points in the same box are enumerated arbitrarily).

The proofs of the above results involve the use of isoperimetric inequalities to get lower bounds and the analysis of two simple heuristics using geometrical information to get matching upper bounds; see Figure 8. In the case of the Bandwidth and Vertex Separation problems, the solutions returned by either of the heuristics are asymptotically optimal.

These results on random geometric graphs can be applied to give empirical evidence of the goodness of several well-known heuristics for layout and partitioning problems [Díaz et al. 2001c]. These heuristics include global methods, such as spectral, multilevel and greedy methods, and local methods, such as simulated annealing or Kernighan-Lin.

Geometric graphs have also been considered as a possible model for wireless communication, where two transmitters are in contact whenever their distance is at most r. Therefore it seems natural to investigate the behavior of layout problems with respect to unreliable random geometric graphs, that is, random geometric graphs whose edges or nodes may fail at random independently. In [Díaz et al. 2001b] it is shown that random geometric networks can tolerate a constant edge or node failure probability maintaining the order of magnitude of the MINEB, MINLA and MINCW measures.

9. UPPER AND LOWER BOUNDS

As we already indicated, all the layout problems presented are hard to solve for general graphs. Therefore, it is important to have algorithms to obtain good lower bounds. On the other hand, it is also interesting to have upper bounds for the layout costs restricted to general classes of graphs. Formally, given a layout cost F we say that an algorithm L computes a lower bound of the cost of a graph G for F if $F(\varphi, G) \geq L(G)$ for all $\varphi \in \Phi(G)$. Also, we say that U is an upper bound of the cost of a graph G for F if $F(\varphi, G) \leq U(G)$ for all $\varphi \in \Phi(G)$.

For some layouts costs and some classes of graphs, even if no exact solutions are known, there exist lower bounds that asymptotically match upper bounds. This happens, for instance, for the edge and vertex bisection of the de Bruijn and Kautz graphs [Rolim et al. 1998], the cutwidth of the de Bruijn graph [Raspaud et al. 1995], the edge bisection of transposition graphs [Stacho and Vrťo 1998], the cutwidth of the d-dimensional mesh of r-ary trees [Vrťo 2000], and for the minimum linear arrangement and cutwidth on graphs with bounded maximal degree and genus [Sýkora and Vrťo 1993]. In [Barth et al. 1995] bounds on bandwidth and cutwidth of a given graph are obtained from bounds for its quotient graphs, which is an useful result because some techniques involve clustering of vertices. In [Diks et al. 1993], edge separation problems are investigated for planar graphs, outerplanar graphs and trees of given maximum degree. The results lead to estimations on the edge bisection and the cutwidth of these graphs. Specifically, Diks et al. [1993] prove that the $\text{MINEB}(G)$ and $\text{MINCW}(G)$ are $O(\sqrt{kn})$ for graphs G with n vertices and maximal degree k and that there exist graphs whose edge bisection and the cutwidth is $\Omega(\sqrt{kn})$.

The remaining of this section presents several approaches to get lower bounds for some layout problems under consideration.

9.1 The Path method

The Path method was introduced in [Juvan and Mohar 1992] to compute a lower bound for the MinLA and Bandwidth problems. Let $P_n^k = (V_n, E_n^k)$ denote the k-th power graph of the path graph: $V_n = [n]$ and $E_n^k = \{ij \ : \ 0 < |i-j| \leq k\}$. A direct calculation gives

$$\text{MINLA}(P_n^k) = \tfrac{1}{6}k(k+1)(3n-2k-1).$$

Let $c(n,m)$ be the largest k for which $|E(P_n^k)| \leq m$. Then,

$$c(n,m) = n - \tfrac{1}{2}\sqrt{(2n-1)^2 - 8m} - \tfrac{1}{2}.$$

The use of these expressions to get a lower bound to MINLA and MINBW is given by the following theorem due to Juvan and Mohar [1992]:

Theorem 9.1. *Let G be a graph with n nodes and m edges and let $k = \lfloor c(n,m) \rfloor$. Then,* $\text{MINLA}(G) \geq \text{MINLA}(P_n^k)$ *and* $\text{MINBW}(G) \geq k$.

9.2 Bounds based on spectral properties

Let $G = ([n], E)$ be a graph and let L_G be its Laplacian matrix, defined by

$$L_G[u,v] = \begin{cases} -1 & \text{if } uv \in E, \\ 0 & \text{if } uv \notin E, \\ \deg(u) & \text{if } u = v. \end{cases}$$

By construction, L_G is positive semidefinite. Therefore it has n nonnegative real eigenvalues $0 \leq \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$. The sequence $\lambda_1, \lambda_2, \ldots, \lambda_n$ is known as the *spectrum* of the graph G. It is know that the multiplicity of the value 0 as an eigenvalue of L_G is equal to the number of connected components of G; in particular, if G is connected $0 = \lambda_1 < \lambda_2$ [Mohar and Poljak 1993].

The following theorem gathers results from Juvan and Mohar [1992] and Mohar and Poljak [1993].

THEOREM 9.2. *Let G be a connected graph with n vertices and let λ_2 be the second smallest eigenvalue of the Laplacian matrix of G. Then,*

$$\begin{aligned}
\textsc{minla}(G) &\geq \lambda_2(n^2-1)/6,\\
\textsc{mincw}(G) &\geq \lambda_2 \left\lfloor \tfrac{1}{2}n \right\rfloor \left\lceil \tfrac{1}{2}n \right\rceil /n,\\
\textsc{mineb}(G) &\geq \begin{cases} \lambda_2 n/4 & \text{if } n \text{ is even,}\\ \lambda_2(n^2-1)/4n & \text{if } n \text{ is odd.}\end{cases}
\end{aligned}$$

Newer bounds for the EDGEBIS problem related to the level structure of a graph can be found in [Bezrukov et al. 2000].

The following result, presented in [Helmberg et al. 1995], bounds the bandwidth of a graph using the ratio between the two extremal eigenvalues of its spectrum:

THEOREM 9.3. *Let G be a graph with n vertices and at least one edge. Let λ_2 and λ_n denote the second smallest and the largest eigenvalue of the Laplacian of G, respectively. Let α be the largest integer smaller than $n\lambda_2/\lambda_n$. Then,*

$$\textsc{minbw}(G) \geq \begin{cases} n-1 & \text{if } \alpha \geq n-2,\\ \alpha+1 & \text{if } \alpha \leq n-2 \text{ and } n \text{ is even,}\\ \alpha & \text{otherwise.}\end{cases}$$

The same reference also contains lower bounds for MINLA that involve more complex eigenvalues. Lower bounds for VERTSEP and BANDWIDTH can be obtained using the following theorem Mohar and Poljak [1993]:

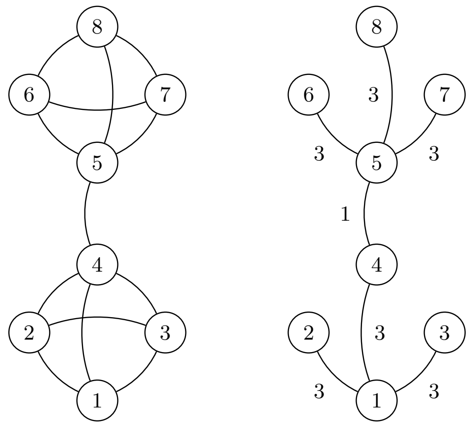
THEOREM 9.4. *Let $G = (V, E)$ be a connected graph with n vertices and maximal degree Δ. Let λ_2 be the second smallest eigenvalue of the Laplacian matrix of G. If $S \subset V$ separates vertices sets A and B then*

$$|C| \geq \frac{4\lambda_2|A||B|}{\Delta n - \lambda_2|A \cup B|}.$$

A slightly weaker version of this result was obtained by Alon and Milman [1985].

9.3 Bounds based on fundamental cuts

Let $G = (V, E)$ be a graph with n vertices and let s and t be two distinguished vertices of G, which we call the source and the sink, respectively. The well-known max-flow min-cut theorem states that the maximal flow value from s to t is equal to the minimal edge cut separating s and t [Ford and Fulkerson 1962]. As there exist efficient algorithms to compute such a minimal cut, and there are $\frac{1}{2}n(n-1)$ possible choices for s and t, it is possible to build a symmetric $n \times n$ matrix f where $f[i, j]$ stores the maximal flow value between two distinct vertices i and j. In [Gomory and Hu 1975] it is shown that matrix f can simply be represented by a weighted spanning tree of G where each edge represents a *fundamental cut* of G and has weight equal to the corresponding minimal cut. The maximum flow $f[i, j]$ between any pair of vertices i and j can be obtained by finding the unique path between i and j in the weighted spanning tree and finding the minimal weight over



f	1	2	3	4	5	6	7	8
1		3	3	3	1	1	1	1
2	3		3	3	1	1	1	1
3	3	3		3	1	1	1	1
4	3	3	3		1	1	1	1
5	1	1	1	1		3	3	3
6	1	1	1	1	3		3	3
7	1	1	1	1	3	3		3
8	1	1	1	1	3	3	3	

Fig. 9. A graph G, its Gomory–Hu tree and its matrix f of max-flows min-cuts. Applying Theorem 9.5, $\textsc{minla}(G) \geq 19$, $\textsc{mincw}(G) \geq 3$ and $\textsc{minbw}(G) \geq 3$. In fact, $\textsc{minla}(G) = 21$, $\textsc{mincw}(G) = 3$ and $\textsc{minbw}(G) = 3$.

all the edges in this path. Figure 9 shows a graph, its corresponding Gomory–Hu tree and its corresponding matrix f. An algorithm to construct the Gomory–Hu tree can be found in the same paper.

By construction, it is easy to see that the maximal fundamental cut for a graph is a lower bound of the Cutwidth problem. Moreover, Adolphson and Hu [1973] proved that the total cut capacity of the $n-1$ fundamental cuts is a lower bound on the cost of the Minimum Linear Arrangement problem. Using Lemma 2.4 we conclude that $\textsc{minbw}(G) \geq \lceil \textsc{minla}(G)/n \rceil$. Therefore, we have the following lower bounds:

THEOREM 9.5. *Let $G = (V, E)$ be a graph and $T = (V, E', w)$ its weighted Gomory–Hu tree. Then,*

$$\begin{aligned}
\textsc{mincw}(G) &\geq \max_{e \in E'} w(e), \\
\textsc{minla}(G) &\geq \sum_{e \in E'} w(e), \\
\textsc{minbw}(G) &\geq \left\lceil \frac{1}{|V|} \sum_{e \in E'} w(e) \right\rceil .
\end{aligned}$$

Adolphson and Hu [1973] also proved that if the Gomory–Hu tree is a path, then that path is the optimal layout for the MinLA problem.

10. HEURISTICS

Resorting to heuristics is an alternative method to obtain solutions for optimization problems. In general, an *heuristic* is a rule of thumb, simplification or guess that reduces or limits the search for solutions in domains that are difficult and poorly understood. In the context of layout problems, an heuristic is a procedure that, given an input graph G, returns a feasible layout $\varphi \in \Phi(G)$. Unlike approximation algorithms, heuristics do not provide a theoretical guarantee on their cost of the returned layout nor on their running time. In spite of that, heuristics are often used in practice, but the assessment of their effectiveness and efficiency is inherently empirical: "It works well with my inputs." In this section, we review several heuristics for some layout problems.

10.1 Heuristics for Bandwidth

Due to their importance in engineering applications, many heuristics have been developed to reduce the bandwidth and/or profile of sparse matrices. Chinn et al. [1982] references a paper citing 49 different heuristics! The most well-known heuristics for bandwidth/profile reduction are the CutHill–McKee algorithm [Cuthill and Mckee 1969], King's algorithm [King 1970], and the Gibbs–Poole–Stockmeyer algorithms [Gibbs et al. 1976]. Most of them belong to a family of heuristics called *level algorithms*. Level algorithms are based on a level structure of the graph, which partitions its vertex set into levels $L_0, \dots, L_s$ such that the endpoints of every edge in the graph are either in the same level L_i or in two consecutive levels L_i and L_{i+1}. The assessment of the goodness of level algorithms for BANDWIDTH has been considered by Turner, who has analyzed their behavior on a particular distribution of random graphs with bandwidth no larger than an integer B [Turner 1986]. Specifically, he considers graphs $\mathcal{G}_{n,p,B}$, resulting from the following experiment: The vertex set is $[n]$, and the edge set is made by connecting, with probability $p \in (0,1)$, any pair of vertices $u, v \in [n]$ such that $0 < |u-v| \leq B$. Turner first proved that for all $\epsilon > 0$, almost all $G \in \mathcal{G}_{n,p,B}$ satisfy $1 \leq B/\text{MINBW}(G) \leq 1+\epsilon$, and then proved that for any level algorithm A, it holds that $A(G)/\text{MINBW}(G) \leq (1+\epsilon)(3-p)$ for almost all $G \in \mathcal{G}_{n,p,k}$, provided that $B = \omega(\log n)$ and p is fixed. The research in [Feige and Krauthgamer 1998] improve on Turner's results, by allowing smaller edges probabilities.

10.2 Heuristics for Edge Bisection

On the other hand, the areas of VLSI and of parallel computing have given rise to many heuristics for the Edge Bisection problem; see [Elsner 1997] for a nice survey. One of the first proposed heuristics to bisect a graph was the Kernighan–Lin heuristic [Kernighan and Lin 1970]. This heuristic, originally developed to minimize the number of connections in electronic circuits, belongs to a more general family of Local Search heuristics [Aarts and Lenstra 1997].

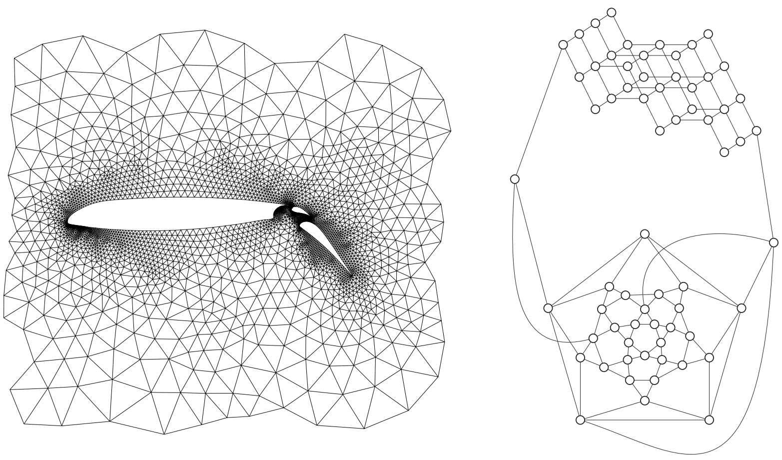
Helpful sets is a more recent local search heuristic developed in [Diekmann et al. 1995]. The basic idea of this heuristic, appears in the proof of a result we have already mentioned to get upper bounds on the bisection of 4-regular graphs [Hromkovič and Monien 1992]. Consider a bisection (V_1, V_2) of a graph $G = (V, E)$. For all $u \in V$, let $\text{ext}(u) = \{uv \in E \ : \ u \in V_i \wedge v \in V_j, \ i \neq j\}$ and let $\text{int}(u) = \{uv \in E \ : \ u, v \in V_i, \ i \in \{1,2\}\}$. Let S be a subset of vertices in V_1; then, for all $u \in S$, let $\text{int}(u, S) = |\{v \in S \ : \ uv \in E\}|$. The *helpfulness* of S is $h(S) = \sum_{u \in S}(\text{ext}(u) - \text{int}(u) + \text{int}(u, S))$. Any S with helpfulness h is said to be h-helpful, as moving S to the other side of the bisection decreases the edge bisection by h edges. On the other hand, given an h-helpful set S in V_1, a subset of vertices S' of $V_2 \cup S$ is a h-balancing set of S if $|S'| = |S|$ and S' is at least $(1-k)$-helpful.

Local search algorithms based on helpful sets find h-helpful sets S and swap their vertices with the ones in some k-balancing set S'. For instance, Helpful sets can be combined with simulated annealing to obtain a nice heuristic for Edge Bisection [Diekmann et al. 1996].

Spectral bisection is another popular heuristic for EDGEBIS based on spectral properties (see Section 9.2). Its basic principle is to compute the Fiedler vector of the Laplacian matrix of an input graph $G = (V, E)$, compute the median M of the components of the Fiedler vector and return a bisection $(A, V \setminus A)$ where A is made of vertices $v \in V$ satisfying $x_v^{(2)} \leq M$. A related algorithm based on eigenvalues is presented and analyzed in [Boppana 1987]. Also, Spielman and Teng [1996] show that spectral bisection methods work well on bounded-degree planar graphs and finite element grids.

There exist several papers that compare the effectiveness of several heuristics for EDGEBIS from an experimental point of view; see [Johnson et al. 1989; Leland and Hendrickson 1994; Diekmann et al. 1995; Battiti and Bertossi 1999]. Unfortunately, these studies do not seem to give an indication on the reason why these heuristics work. Other heuristics for EDGEBIS include multilevel algorithms [Barnard and Simon 1994], tabu search [Glover and Laguna 1997, Section 8.6.2] or genetic algorithms [Kadluczka and Wala 1995].

Simpler local search heuristics for the Edge Bisection problem have been theoretically analyzed on particular random graphs: Jerrum and Sorkin [1998] analyzed the Metropolis algorithm for the Edge Bisection problem on graphs with a *planted bisection*. Their results involve an algorithm that, starting from a random bisection, iteratively takes a pair of vertices in different sides of the current bisection and interchanges them with probability $1/(1 + \exp(\delta/t))$, where δ is the increase of the cut size and t is a parameter called temperature. Their analysis considers planted bisection $\mathcal{G}_{4n,p,r}$ random graphs, generated as follows: A $\mathcal{G}_{4n,p,r}$ graph has $4n$ vertices, half of them black and half of them white; edges between vertices of the same color are included independently with probability p, whereas edges between vertices of different colors are included independently with probability $r < p$. They first prove that the planted bisection, defined by the original coloring, is, with high probability, the unique optimal bisection on $\mathcal{G}_{4n,p,r}$ random graphs. Then, they prove that, with overwhelming probability, for a certain choice of t, the Metropolis algorithm can find the planted bisection of $\mathcal{G}_{4n,p,r}$ random graphs in time $O(n^2)$, provided $p - r = \Omega(n^{-1/6})$. In his thesis Juels improves these results [Juels 1996]. To cope with the simpler Hillclimbing algorithm: interchanges are accepted only if they decrement the size of the bisection. His results state that Hillclimbing can find the planted bisection of $\mathcal{G}_{4n,p,r}$ random graphs in expected time $O(n^2)$ with probability $c > 0$, provided $p - r$ is constant. Repeated executions can boost the probability of success. Juels also reports experimental results that show that Hillclimbing may be much more effective than what his theoretical results indicate on the random graphs he considers. The previous results of Jerrum, Sorkin and Juels have been recently improved in [Carson and Impagliazzo 2001], where it is proved that with a constant non-zero probability, the hill climbing heuristic is also able to find the minimal bisection of random graphs in the $\mathcal{G}_{4n,p,r}$ model with $p - r = \Omega(n^{-1/4} \log^4 n)$.



(a) `airfoil1` (b) `gd96c`

Fig. 10. Two graphs used in the comparison of heuristic for the MinLA problem.

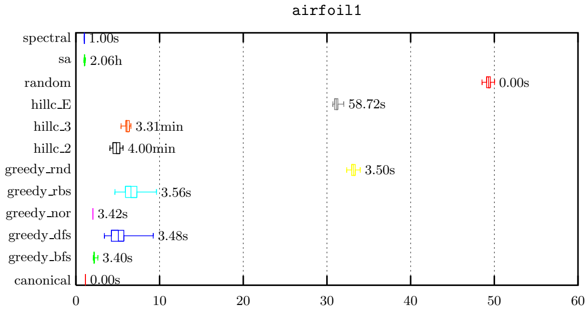
Similar results for a generalized version of $\mathcal{G}_{4n,p,r}$ graphs, but with a successive augmentation heuristic rather than with local search were proved in [Condon and Karp 2001]. Their heuristic is based on a greedy algorithm that repeatedly selects a new pair of vertices and adds one to each side the bisection. They show that their linear time heuristic hits the planted bisection with overwhelming probability, provided $p - r \geq n^{-1/2}$.

10.3 Heuristics for MinLA

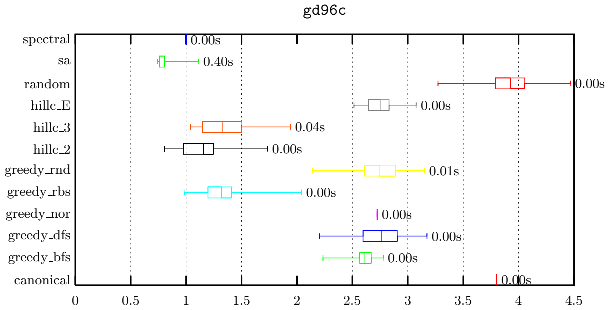
For the MinLA problem, an heuristic that we call *spectral sequencing* was proposed in [Juvan and Mohar 1992]. Spectral sequencing first computes the eigenvector $x^{(2)}$ corresponding to the second smallest eigenvalue λ_2 of the Laplacian matrix $L(G)$ of the input graph G and then ranks each position of $x^{(2)}$. Thus, the heuristic returns an arrangement φ satisfying $\varphi(u) \leq \varphi(v)$ whenever $x_u^{(2)} \leq x_v^{(2)}$.

Another algorithm to get upper bounds on $\textsc{minla}(G)$ for arbitrary G is described in [Muradyan 1985]. This algorithm creates first an initial arrangement of vertices $u_1, u_2, \ldots, u_n$, so that u_1 is a vertex with the minimal degree in G, u_2 is a vertex with the minimal degree in $G \setminus \{u_1\}$, etc. and working out this arrangement, creates a new arrangement for which $\textsc{minla}(G)$ is estimated.

An experimental study of heuristics for the MinLA problem has been undertaken in [Petit 1998; 2001a; 2001b]. The empirical results are based on two random models (binomial random graphs and random geometric graphs), "real life" graphs (including graphs arising from finite element discretizations, VLSI design and graph-drawing) and graphs with known optimal solutions (trees, hypercubes, grids). The first goal was to obtain an evaluation on the behavior of different families of heuristics through experimental results. The second goal was to use, in a systematic way, this experimental knowledge as a fundamental method to guide the design of new heuristics. Furthermore, approximation heuristics and methods



(a) Results for `airfoil1`



(b) Results for `gd96c`

Fig. 11. Comparison of the relative effectiveness and efficiency of several heuristic for the MinLA problem for the graphs in Figure 10. The results are shown using boxplots, which summarize the distribution of the values of a group of samples. The box in a boxplot shows the median value as a line and the first (25th percentile) and third quartile (75th percentile) of the value distribution as the lower and upper parts of the box. The bars shown at the sides of the boxes represent the largest and smallest observed values. The average time elapsed to compute an individual of the sample is displayed at the right of the corresponding boxplot.

to find lower bounds for MinLA are presented, evaluated and compared experimentally when applied to sparse graphs. The methods belong to the families of Successive Augmentation heuristics, Local Search heuristics, and Spectral Sequencing. The conclusion is that the best approximations are obtained using Simulated Annealing, which involves a large amount of computation time. However, solutions found by Spectral Sequencing are also good and can be found in radically less time (see Figure 11).

Motivated by the long running time of Simulated Annealing compared to Spectral Sequencing, Petit [2000] presents and analyzes sequential and parallel versions of a new heuristic to approximate the MinLA problem on graphs embedded and clustered in some geometry. The heuristics consist in obtaining a first global solution using Spectral Sequencing and improving it locally through Simulated Annealing.

The measurements obtained on a commodity cluster of nine work stations show that the new parallel heuristic maintains solution quality, decreases the running time and offers an excellent speedup when run in parallel without sacrificing solution quality.

Figure 11 gives a sample of the obtained measurements for the two graphs depicted in Figure 10.

It must be remarked, however, that the experimental results for MinLA evidence a big gap between the best known upper and lower bounds. Therefore it remains as open problem to devise better techniques to obtain better lower bounds.

10.4 Heuristics for SumCut

Several heuristics have also successfully been applied for the SumCut problem in order to reduce the profile of symmetric matrices. The early heuristics of Cuthill and Mckee [1969; King [1970; Gibbs et al. [1976], have now been improved in order to use simulated annealing, spectral methods, multilevel algorithm and hybrid schemes; see e.g. [Everstine 1979; Lewis 1994; Barnard et al. 1995; Kumfert and Pothen 1997; Hager 2002].

10.5 Implementations

There exist several software libraries that implement many of the above mentioned heuristics. The Party library [Preis and Diekmann 1996] and the Chaco library [Hendrickson and Leland 1997] are packages that include a variety of different methods to partition or bisect graphs. In particular, Chaco enables a fast execution of the spectral sequencing heuristic for MinLA. Also, Metis is a library of programs for partitioning graphs and computing profile reducing orderings of sparse matrices [Karypis 2001]. The `llsh` toolkit offers several heuristics to approximate the MinLA problem [Petit 2001a].

11. VARIATIONS ON THE BASIC MODEL

Several variations on the basic layout model have been considered, there are modifications in the type of layout or in the fact that the graph carries additional information that must be preserved in some particular way in the layout. We survey here some results on stack and queue layouts, as well as associated problems for weighted, directed and colored input graphs.

11.1 Stack and queue layouts

Stack and queue layouts were motivated from different contexts as VLSI design, fault-tolerant processing, parallel processing, sorting networks and parallel matrix computations [Chung et al. 1987; Heath et al. 1992; Heath and Rosenberg 1992; Heath et al. 1993].

A *k-queue layout* of an undirected graph $G = (V, E)$ consists of a linear order φ of the vertex set V and an assignment ψ of each edge in E into exactly one of k-queues. Observe that the ordering assigns a direction to each edge in E. The queue policy operates as follows: the vertices of V are scanned in left-to-right order. When vertex i is encountered, any edge assigned to a queue that have i as right end-point must be at the front of the queue, and it is dequeued. All the edges that have i as left end-point are enqueued in the assigned queues. The *queue-number* of G, $\text{MINQN}(G)$, is the smallest k such that G has a k-queue layout. The *queue* *layout* problem is to find a queue layout that uses minimum number of queues, for a given graph.

In the same manner, a *k-stack layout* of an undirected graph $G = (V, E)$ consists of a linear order of the vertex set V and an assignment of each edge in E to exactly one of the k-stacks. The queue policy operates as before scanning in left-to-right order V. When vertex i is encountered, any edge assigned to a stack that have i as right end-point must be at the top of the stack, and are popped. All the edges that have i as left end-point are pushed in the assigned stacks. Similarly, we can define the *stack-number* of a given graph. The *stack layout* problem is to find, for a given graph, a stack layout that uses minimum number of stacks.

Stack layouts were introduced in [Bernhart and Kainen 1979] under the name *embedding graphs in books*; the stack-number is also referred as the *page-number* or the *page thickness*. In [Chung et al. 1987] and [Bilski 1993] optimal stack layouts are constructed for a variety of graph classes. In [Heath and Rosenberg 1992], the queue layout problem is introduced. In the same paper, the authors prove that recognizing graphs with queue-number 1 is an NP-complete problem. On the other hand recognizing graphs with stack-number 1 can be done in linear time as this class of graphs equals the class of outerplanar graphs [Bernhart and Kainen 1979]. Notice that, when the vertex ordering of the layout is fixed, the minimum number of queues required for the layout can be obtained in polynomial time, while computing the minimum number of stacks is NP-hard [Even and Itai 1971; Garey et al. 1980]. A comparative study of the power of queues and stack layouts was performed in [Heath et al. 1992].

11.2 Directed acyclic graphs

A linear layout φ of a *directed* graph is a layout that provides a topological sorting of the vertices, that is, for any arc (u, v) we have that $\varphi(u) < \varphi(v)$. This restricted type of layout can only be possible for directed acyclic graphs, and several layout problems have been considered also for this class of directed graphs.

The Bandwidth problem remains NP-complete for directed trees with maximum in-degree 1 and maximum out-degree 2 [Garey et al. 1978].

The Cutwidth and MinLA problems for directed acyclic graphs are NP-complete [Even and Shiloach 1975], but they can be solved in polynomial time for rooted trees [Adolphson and Hu 1973]. A polylogarithmic factor approximation algorithm for the MinLA problem on directed acyclic graphs is given in [Ravi et al. 1991]

In some applications of stack and queue layouts, the input graph is a directed acyclic graph: Heath and Pemmaraju [1999] provide several results for different classes of directed acyclic graphs and Heath et al. [1999] analyze both problems from an algorithmic point of view. In this last paper it is shown that recognizing 1-queue directed acyclic graphs can be done in polynomial time, but recognizing 4-queue or 6-stack directed acyclic graphs are NP-complete problems. Extensions of these results to directed acyclic graphs obtained from posets are reported in [Heath and Pemmaraju 1997] and [Heath et al. 1999].

11.3 Colored graphs

There is a close relationship between intervalizing graph problems and graph layout problems on colored graphs. Interval graphs have been studied intensively because of their wide applicability to practical problems [Golumbic 1980]. In the last years quite a lot of effort has been devoted to the study of problems where the goal is to complete a graph into an interval graph. This kind of problems are used to model ambiguity in Physical Mapping or consistency in Temporal Reasoning [Golumbic et al. 1995]. The problem of *physical mapping* consists in determining the relative position of several fragments of DNA, which can be used by biologists to characterize individual genes. As the DNA fragments are obtained scrambled, in order to get a map of the original sequence, it is necessary to re-assemble them using information of their pairwise overlaps [Karp 1993].

A *k-coloring* of a graph $G = (V, E)$ is a mapping $\kappa : V \longrightarrow \{1, \ldots, k\}$ such that no two adjacent vertices have the same color. A *k-colored* graph is a graph together with a k-coloring. A *colored layout* of a given k-colored graph (G, κ) is a layout φ of G such that for any $u, x, v \in V$, if $(u, v) \in E$ and $\varphi(u) < \varphi(x) < \varphi(v)$ then $\kappa(u) \neq \kappa(x)$. A *proper colored layout* of (G, κ) is a colored layout φ of G such that for all $u, v \in V$ and $x \in V$ with degree at least 2, if $(u, v) \in E$ and $\varphi(u) < \varphi(x) < \varphi(v)$ then it exists a vertex y such that $\varphi(v) < \varphi(y)$ and $(x, y) \in E$. We consider the following layout problems for colored graphs:

—*Colored Layout Problem* (CLP): Given a k-colored graph (G, κ), decide whether there is a colored layout φ of (G, κ).

—*Proper Colored Layout Problem* (PCLP): Given a k-colored graph (G, κ), decide whether there is a proper colored layout φ of (G, κ).

Àlvarez et al. [1998] showed that the CLP is NP-complete for caterpillars with hairs of length at most 2 and that the problem can be solved in NC for caterpillars with hairs of length at most 1, independently of the number of colors. In [Àlvarez et al. 2001] it was shown that the problem is still NP-complete when the input is restricted to be four colored caterpillar with unbounded hair length. The PCLP is also hard for caterpillars of hair length 2 see [Àlvarez and Serna 1999]. These two problems are closely related to intervalizing problems:

—*Interval Colored Graph* (ICG): Given a k-colored graph $(G = (V, E), \kappa)$, decide whether there is an edge superset E' such that the graph $G' = (V, E')$ is an interval graph and κ is still a proper coloring of G'.

—*Proper Interval Colored Graph* (PICG): Given a k-colored graph $(G = (V, E), \kappa)$, decide whether there is an edge superset E' such that the graph $G' = (V, E')$ is a proper interval graph and κ is still a proper coloring of G'.

The ICG problem is a special case of the *Interval Sandwich Problem* [Golumbic and Shamir 1993], and has received a lot of attention, as a simplified model for reconstructing the ordering physical DNA mapping problems [Fellows et al. 1993]. Moreover, the problem was shown to be NP-complete in [Fellows et al. 1993] (see also [Golumbic et al. 1994]) and it is polynomial time equivalent to the CLP [Dinneen 1995; Àlvarez et al. 1998], therefore it is also hard for caterpillars. If the problem is given parameterized on the number of colors, hardness results were obtained in [Bodlaender et al. 1994]. Later it was shown that the problem is NP-complete for the case of 4 or more colors, while if we only have 2 and 3 colors the problem is solvable in linear and quadratic time respectively [Bodlaender and de Fluiter 1996]. A different approach to study the problem is to fix the number of colors and look at the complexity of the problem considering the degree of vertices. For fixed number of colors k and degree bounded by a constant d, there is a $O(n^{k-1})$ algorithm for the ICG [Kaplan and Shamir 1999].

In [Kaplan and Shamir 1996] and [Kaplan et al. 1999] it was shown that the parameterized version of the PICG problem, with parameter the number of colors, is $W[1]$-hard, and this implies the NP-completeness of PICG. They also gave a polynomial time algorithm for constant number of colors. For another simpler NP-completeness proof for this problem see [Goldberg et al. 1995]. When the input graph is a colored tree, the problem is hard even for caterpillars of hair length 2. The proof of this last result exploits the relationship with the PICG problem [Àlvarez and Serna 1999].

11.4 Weighted graphs

When we consider edge weighted graphs the cost of a layout measure, involving edges, is easily extended by multiplying the contribution of an edge by its weight. Some of the hardness results for layout problems are stronger, for example the CUTWIDTH is NP-complete for weighted trees [Monien and Sudborough 1988].

The MINLA is solvable in polynomial time for directed weighted trees [Adolphson and Hu 1973]. Interestingly, the approximability results for several graph layout problems of Even et al. [2000; Rao and Richa [1998] also apply to weighted graphs.

The research reported in [Chirravuri et al. 1996] has shown that a formulation of the physical mapping problem is closely related to the MINLA problem on edge weighted graphs.

11.5 Graph labelings

A vertex *labeling* of a graph G is an assignment ϕ of labels to the vertices of G that induces for each edge $uv \in E(G)$ an edge label depending on $\phi(u)$ and $\phi(v)$. A function ϕ is called a *graceful labeling* of a graph with m edges if ϕ is an injection from $V(G)$ to $\{0, 1, \ldots, m\}$, such that, for all $uv \in E(G)$, each edge label $|\phi(u) - \phi(v)|$ is distinct. A function ϕ is called a *harmonious labeling* of a graph with m edges if ϕ is an injection from $V(G)$ to the group of integers modulo m, such that, for all $uv \in E(G)$, each edge label $\phi(u) + \phi(v) \mod m$ is distinct. See [Gallian 1998] for a survey on graph labeling.

12. CONCLUDING REMARKS

In this paper we have presented a current view on the main known results about graph layout problems. The reader has surely observed that plenty of problems remain open: In general, everything that in this paper is not referred as done might turn into an interesting research problem.

For instance, with regard to complexity, little is known about MINLA for sparse graphs. It is not known if it is NP-complete or it is in the class P. Also, as noted in [Bui and Peck 1992] and [Papadimitriou and Sideri 1996] the NP-completeness of EDGEBIS for planar graphs remains open. Also, it would be interesting to know whether problems different from BANDWIDTH do or not belong to APX for general graphs or even for planar graphs.

Another task to carry out would be the study of other probabilistic classes of graphs for which good estimations of their optimal values could be found easily (binomial random graphs and random geometric graphs were an example of such a class). These classes could serve as a source of new graph generators to help in the benchmark and analysis of heuristics as well as to the design of networks with some specified properties. A concrete example of this class of graphs could be non uniform random geometric graphs, as they can be used as a model for mobile computing networks.

Moreover, the design of efficient hybrid heuristics is another interesting area, where plenty of research can be done.

ACKNOWLEDGMENTS

The authors want to thank Sergei Bezrukov and David Muradyan for pointing, providing and translating key references in Russian. We also would like to thank Uriel Feige, Larry Harper, Imrich Vrťo and two anonymous referees for giving us usefull information.

REFERENCES

AARTS, E. AND LENSTRA, J. K., Eds. 1997. *Local search in combinatorial optimization.* Wiley, New York.

ADOLPHSON, D. 1977. Single machine job sequencing with precedence constraints. *SIAM Journal on Computing 6*, 40–54.

ADOLPHSON, D. AND HU, T. C. 1973. Optimal linear ordering. *SIAM Journal on Applied Mathematics 25,* 3, 403–423.

ALON, N. AND MILMAN, V. 1985. λ_1, isoperimetric inequalities for graphs and superconcentrators. *Journal of Combinatorial Theory. Series B 38*, 73–88.

ÀLVAREZ, C., CASES, R., DÍAZ, J., PETIT, J., AND SERNA, M. 2000. Routing trees problems on random graphs. In *Approximation and Randomized Algorithms in Communication Networks*, J. Rolim et al., Ed. ICALP Workshops 2000. Carleton Scientific, 99–111.

ÀLVAREZ, C., DÍAZ, J., AND SERNA, M. 1998. Intervalizing colored graphs is NP-complete for caterpillars with hair length 2. Technical report LSI 98-9-R, Dept. de Llenguatges i Sistemes Informàtics, Univ. Politècnica de Catalunya.

ÀLVAREZ, C., DÍAZ, J., AND SERNA, M. 2001. The hardness of intervalizing four colored caterpillars. *Discrete Mathematics 235*, 19–27.

ÀLVAREZ, C. AND SERNA, M. 1999. The proper interval colored graph problem for caterpillar trees. Technical report LSI 99-12-R, Dept. de Llenguatges i Sistemes Informàtics, Univ. Politècnica de Catalunya.

ARORA, S., FRIEZE, A., AND KAPLAN, H. 1996. A new rounding procedure for the assignment problem with applications to dense graphs arrangements. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*. 21–30.

ARORA, S., KARGER, D., AND KARPINSKI, M. 1999. Polynomial time approximation schemes for dense instances of NP-hard problems. *Journal of Computer and System Sciences 58,* 1, 193–210.

ASSMAN, S. F., PECK, G. W., SYSLO, M. M., AND ZAK, J. 1981. The bandwidth of caterpillars with hair of lengths 1 and 2. *SIAM Journal on Algebraic and Discrete Methods 2*, 387–393.

AUSIELLO, G., CRESCENZI, P., GAMBOSI, G., KANN, V., MARCHETTI-SPACCAMELA, A., AND PROTASI, M. 1999. *Complexity and approximation.* Springer–Verlag, Berlin.

AZIZOĞLU, C. AND EĞECIOĞLU, Ö. 2002. The isoperimetric number and the bisection width of generalized cylinders. *Discrete Mathematics*. To appear.

BALASUBRAMANIAN, R., FELLOWS, M., AND RAMAN, V. 1998. An improved fixed-parameter algorithm for vertex cover. *Information Proccessing Letters 65*, 163–168.

BARNARD, S. T., POTHEN, A., AND SIMON, H. 1995. A spectral algorithm for envelope reduction of sparse matrices. *Numerical Linear Algebra with Applications 2,* 4, 317–334.

BARNARD, S. T. AND SIMON, H. D. 1994. A fast multilevel implementation of recursive spectral bisection for partitioning unstructured problems. *Concurrency: Practice and Experience 6*, 101–107.

BARTH, D., PELLEGRINI, F., RASPAUD, A., AND ROMAN, J. 1995. On bandwidth, cutwidth, and quotient graphs. *RAIRO Informatique Théorique et Applications. 29,* 6, 487–508.

BATTITI, R. AND BERTOSSI, A. 1999. Greedy, prohibition, and reactive heuristics for graph partitioning. *IEEE Transactions on Computers 48,* 4, 361–385.

BERGER-WOLF, T. Y. AND REINGOLD, E. M. 2002. Index assignment for multichannel communication under failur. *IEEE Transactions on Information Theory*. to appear.

BERNHART, F. AND KAINEN, P. C. 1979. The book thickness of a graph. *Journal of Combinatorial Theory. Series B 27,* 3, 320–331.

BERRY, J. W. AND GOLDBERG, M. K. 1999. Path optimization for graph partitioning problems. *Discrete Applied Mathematics 90*, 27–50.

BEZRUKOV, S. L. 1999. Edge isoperimetric problems on graphs (a survey). In *Graph theory and combinatorial biology (Balatonlelle, 1996)*, L. Lovasz, A. Gyarfas, G. O. H. Katona, A. Recski, and L. Szekely, Eds. János Bolyai Math. Soc., 157–197.

BEZRUKOV, S. L., CHAVEZ, J. D., HARPER, L. H., RÖTTGER, M., AND SCHROEDER, U.-P. 2000. The congestion of n-cube layout on a rectangular grid. *Discrete Mathematics 213,* 1-3, 13–19.

BEZRUKOV, S. L., ELSÄSSER, R., MONIEN, B., PREIS, R., AND TILLICH, J.-P. 2000. New spectral lower bounds on the bisection width of graphs. In *Graph-Theoretic Concepts in Computer Science*, U. Brandes and D. Wagner, Eds. Lecture Notes in Computer Science, vol. 1928. Springer–Verlag, 23–34.

BHATT, S. N. AND LEIGHTON, F. T. 1984. A framework for solving VLSI graph layout problems. *Journal of Computer and System Sciences 28*, 300–343.

BILSKI, T. 1993. Embedding graphs in books with prespecified order of vertices. *Poznańskie Towarzystwo Przyjaciół Nauk. Wydział Nauk Technicznych. Prace Komisji Automatyki i Informatyki. Studia z Automatyki 18*, 5–12.

BLACHE, G., KARPINSKI, M., AND WIRTGEN, J. 1998. On approximation intractability of the bandwidth problem. Technical report TR98-014, Electronic Colloquium on Computational Complexity.

BLUM, A., KONJEVOD, G., RAVI, R., AND VEMPALA, S. 2000. Semi-definite relaxations for minimum bandwidth and other vertex-ordering problems. *Theoretical Computer Science 235,* 1, 25–42.

BODLAENDER, H., FELLOWS, M. R., AND HALLET, M. T. 1994. Beyond NP-completeness for problems of bounded width: hardness for the W-hierarchy. In *Proceedings of the 26th Annual ACM Symposium on the Theory of Computing*. 449–458.

BODLAENDER, H. L. 1993. A tourist guide through treewidth. *Acta Cybernetica 11,* 1-2, 1–21.

BODLAENDER, H. L. 1996. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing 25,* 6, 1305–1317.

BODLAENDER, H. L. AND DE FLUITER, B. 1996. On intervalizing k-colored graphs for DNA physical mapping. *Discrete Applied Mathematics 71,* 1-3, 55–77.

BODLAENDER, H. L., GILBERT, J. R., HAFSTEINSSON, H., AND KLOKS, T. 1995a. Approximating treewidth, pathwidth, and minimum elimination tree height. *Journal of Algorithms 18*, 238–255.

BODLAENDER, H. L., KLOKS, T., AND KRATSCH, D. 1995b. Treewidth and pathwidth of permutation graphs. *SIAM Journal on Discrete Mathematics 8,* 4, 606–616.

BODLAENDER, H. L. AND MÖHRING, R. H. 1993. The pathwidth and treewidth of cographs. *SIAM Journal on Discrete Mathematics 6,* 2, 181–188.

BOLLOBÁS, B. 1985. *Random graphs*. Academic Press, London.

BOLLOBÁS, B. AND LEADER, I. 1991. Edge–isoperimetric inequalities in the grid. *Combinatorica 4*, 299–314.

BOPPANA, R. 1987. Eigenvalues and graph bisection: An average case analysis. In *Proceedings of the 28th Annual Symposium on Foundations of Computer Science*. 280–285.

BORNSTEIN, C. AND VEMPALA, S. 2002. Flow metrics. In *LATIN 2002: Theoretical Informatics*, S. Rajsbaum, Ed. Lecture Notes in Computer Science, vol. 2286. Springer–Verlag, 516–527.

BOTAFOGO, R. A. 1993. Cluster analysis for hypertext systems. In *Proceedings of the 16th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, R. Korfhage, E. M. Rasmussen, and P. Willett, Eds. 116–125.

BUI, T., CHAUDHURI, S., LEIGHTON, T., AND SIPSER, M. 1987. Graph bisection algorithms with good average case behavior. *Combinatorica 7*, 171–191.

BUI, T. N. AND PECK, A. 1992. Partitioning planar graphs. *SIAM Journal on Computing 21,* 2, 203–215.

CAPRARA, A., MALUCELLI, F., AND PRETOLANI, D. 2002. On bandwidth-2 graphs. *Discrete Applied Mathematics. 117*, 1–13.

CARSON, T. AND IMPAGLIAZZO, R. 2001. Hill climbing finds random planted bisections. In *Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms.* 903–909.

CHANG, G. J. AND LAI, Y.-L. 2001. On the profile of the corona of two graphs. Manuscript (provided by S. Bezrukov).

CHINN, P., CHVÁTALOVÁ, J., DEWDNEY, A., AND GIBBS, N. 1982. The bandwidth problem for graphs and matrices—A survey. *Journal of Graph Theory 6*, 223–254.

CHINN, P. Z., LIN, Y. X., AND YUAN, J. J. 1992. The bandwidth of the corona of two graphs. In *Proceedings of the Twenty-third Southeastern International Conference on Combinatorics, Graph Theory, and Computing.* Congressus Numerantium, vol. 91. Utilitas Math., 141–152.

CHINN, P. Z., LIN, Y. X., YUAN, J. J., AND WILLIAMS, K. 1995. Bandwidth of the composition of certain graph powers. *Ars Combinatoria 39*, 167–173.

CHIRRAVURI, S., BHANDARKAR, S. M., AND ARNOLD, J. 1996. Parallel computing of physical maps—A comparative study in SIMD and MIMD parallelism. *Journal of Computational Biology 3,* 4, 503–528.

CHUNG, F. R. K. 1988. *Labelings of graphs.* Academic Press, San Diego, 151–168.

CHUNG, F. R. K., LEIGHTON, F. T., AND ROSENBERG, A. L. 1987. Embedding graphs in books: a layout problem with applications to VLSI design. *SIAM Journal on Algebraic and Discrete Methods 8,* 1, 33–58.

CHUNG, M., MAKEDON, F., SUDBOROUGH, I. H., AND TURNER, J. 1982. Polynomial time algorithms for the min cut problem on degree restricted trees. In *Proceedings of the 23th Annual Symposium on Foundations of Computer Science.* 262–271.

CHVÁTALOVÁ, J. 1975. Optimal labelling of a product of two paths. *Discrete Mathematics 11*, 249–253.

CHVÁTALOVÁ, J., DEWDNEY, A., GIBBS, N., AND KORFHAGE, R. 1975. The bandwidth problem for graphs — A collection of recent results. Technical report CS 75020, Dept. of Computer Science, Southern Methodist University.

CONDON, A. AND KARP, R. M. 2001. Algorithms for graph partitioning on the planted partition model. *Random Structures & Algorithms 18,* 2, 116–140.

CRESCENZI, P., SILVESTRI, R., AND TREVISAN, L. 2001. On weighted vs unweighted versions of combinatorial optimization problems. *Information and Computation 167,* 1, 10–26.

CUTHILL, E. H. AND MCKEE, J. 1969. Reducing the bandwidth of sparse symmetric matrices. In *Proceedings of the 24th ACM National Conference.* 157–172.

DEO, N., KRISHNAMOORTHY, M. S., AND LANGSTON, M. A. 1987. Exact and approximate solutions for the gate matrix layout problem. *IEEE Transactions on Computer Aided Design 6,* 1, 79–84.

DEWDNEY, A. K. 1976. The bandwidth of a graph—some recent results. In *Proceedings of the Seventh Southeastern Conference on Combinatorics, Graph Theory, and Computing.* Number 17 in Congressus Numerantium. Utilitas Math., 273–288.

DÍAZ, J. 1979. The δ-operator. In *Fundamentals of Computation Theory*, L. Budach, Ed. Akademie–Verlag, 105–111.

DÍAZ, J. 1992. Graph layout problems. In *Mathematical Foundations of Computer Science 1992*, I. M. Havel and V. Koubek, Eds. Lectures Notes in Computer Sciences, vol. 629. Springer–Verlag, 14–23.

Díaz, J., Gibbons, A., Pantziou, G., Serna, M., Spirakis, P., and Torán, J. 1997. Parallel algorithms for the minimum cut and the minimum length tree layout problems. *Theoretical Computer Science 181,* 2, 267–288.

Díaz, J., Gibbons, A. M., Paterson, M. S., and Torán, J. 1991. The Minsumcut problem. In *Algorithms and Data Structures*, F. Dehen, R. J. Sack, and N. Santoro, Eds. Lecture Notes in Computer Science, vol. 519. Springer–Verlag, 65–79.

Díaz, J., Penrose, M. D., Petit, J., and Serna, M. 2000. Convergence theorems for some layout measures on random lattice and random geometric graphs. *Combinatorics, Probability and Computing 9,* 6, 489–511.

Díaz, J., Penrose, M. D., Petit, J., and Serna, M. 2001a. Approximating layout problems on random geometric graphs. *Journal of Algorithms 39,* 1, 78–116.

Díaz, J., Petit, J., and Serna, M. 2001b. Faulty random geometric networks. *Parallel Processing Letters 10,* 4, 343–357.

Díaz, J., Petit, J., Serna, M., and Trevisan, L. 2001c. Approximating layout problems on random graphs. *Discrete Mathematics 235,* 1–3, 245–253.

Díaz, J., Serna, M., Spirakis, P., and Torán, J. 1997. *Paradigms for fast parallel approximability.* Cambridge University Press, Cambridge.

Diekmann, R., Lüling, R., and Monien, B. 1994. Communication throughput of interconnection networks. In *Mathematical Foundations of Computer Science 1994*, I. Privara, B. Rovan, and P. Ruzicka, Eds. Lecture Notes in Computer Science, vol. 841. Springer–Verlag, 72–86.

Diekmann, R., Lüling, R., Monien, B., and Spräner, C. 1996. Combining helpful sets and parallel simulated annealing for the graph-partitioning problem. *Parallel Algorithms and Applications 8*, 61–84.

Diekmann, R., Monien, B., and Preis, R. 1995. Using helpful sets to improve graph bisections. In *Interconnection Networks and Mapping and Scheduling Parallel Computations*, D. F. Hsu, A. L. Rosenberg, and D. Sotteau, Eds. DIMACS Series in Discrete Mathematics and Theoretical Computer Science. AMS Press, 57–73.

Diks, K., Djidjev, H., Sýroka, O., and Vrťo, I. 1993. Edge separators for planar and outerplanar graphs with applications. *Journal of Algorithms 14*, 258–279.

Ding, G. and Oporowski, B. 1995. Some results on tree decomposition of graphs. *Journal of Graph Theory 20,* 4, 481–499.

Dinneen, M. J. 1995. Bounded cmbinatorial width and forbidden substructures. Ph.D. thesis, Computer Science Dept., University of Victoria.

Downey, R. G. and Fellows, M. R. 1995. Fixed-parameter tractability and completeness II: On completeness for $W[1]$. *Theoretical Computer Science 141,* 1-2, 109–131.

Downey, R. G. and Fellows, M. R. 1999. *Parameterized complexity.* Springer–Verlag, New York.

Dunagan, J. and Vempala, S. 2001. On euclidean embeddings and bandwidth minimization. In *Approximation, Randomization and Combinatorial Optimization: Algorithms and Techniques*, M. Goemans, K. Jansen, J. Rolim, and L. Trevisan, Eds. Lecture Notes in Computer Science, vol. 2129. Springer-Verlag, 229–240.

Ellis, J., Sudborough, I. H., and Turner, J. 1979. The vertex separation and search number of a graph. *Information and Computation 113*, 50–79.

Elsner, U. 1997. Graph partitioning—A survey. Technical report 393, Technische Universität Chemnitz.

Even, G., Naor, J., Rao, S., and Schieber, B. 1999. Fast approximate graphs algorithms. *SIAM Journal on Computing 28,* 6, 2187–2214.

Even, G., Naor, J., Rao, S., and Schieber, B. 2000. Divide-and-conquer approximation algorithms via spreading metrics. *Journal of the ACM 47,* 4, 585–616.

Even, S. and Itai, A. 1971. Queues, stacks, and graphs. In *Theory of machines and computations.* Academic Press, 71–86.

Even, S. and Shiloach, Y. 1975. NP-completeness of several arrangement problems. Technical report TR-43, Dept. of Computer Science, Technion, Haifa.

EVERSTINE, G. C. 1979. A comparison of three resequencing algorithms for the reduction of matrix profile and wavefront. *International Journal of Numerical Methods Engineering 14*, 837–853.

FEIGE, U. 2000. Approximating the bandwidth via volume respecting embeddings. *Journal of Computer and System Sciences 60,* 3, 510–539.

FEIGE, U. AND KRAUTHGAMER, R. 1998. Improved performance guarantees for bandwidth minimization heuristics (draft). Technical report, Computer Science Dept., Weizmann Inst. Tech.

FEIGE, U. AND KRAUTHGAMER, R. 2002. A polylogarithmic approximation of the minimum bisection. *SIAM Journal on Computing*, To appear.

FEIGE, U., KRAUTHGAMER, R., AND NISSIM, K. 2002. Approximating the minimum bisection size. *Discrete Applied Mathematics*. To appear.

FELLOWS, M. R., HALLET, M. T., AND WAREHAM, W. T. 1993. DNA physical mapping: Three ways difficult. In *Algorithms-ESA'93*, T. Lengauer, Ed. Number 726 in Lecture Notes in Computer Science. Springer-Verlag, 157–168.

FELLOWS, M. R. AND LANGSTON, M. A. 1988. Layout permutation problems and well-partially-ordered sets. In *Advanced research in VLSI.* MIT Press, Cambridge, MA, 315–327.

FELLOWS, M. R. AND LANGSTON, M. A. 1992. On well-partial-order theory and its application to combinatorial problems of VLSI design. *SIAM Journal on Discrete Mathematics 5,* 1, 117–126.

FELLOWS, M. R. AND LANGSTON, M. A. 1994. On search, decision, and the efficiency of polynomial-time algorithms. *Journal of Computer and System Sciences 49,* 3, 769–779.

FISHBURN, P., TETALI, P., AND WINKLER, P. 2000. Optimal linear arrangement of a rectangular grid. *Discrete Mathematics 213*, 123–139.

FORD, L. R. AND FULKERSON, D. R. 1962. *Flows in networks.* Princeton University Press.

FRIEZE, A. AND KANNAN, R. 1996. The regularity lemma and approximation schemes for dense problems. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science.* 12–20.

GALLIAN, J. 1998. A dynamic survey of graph labeling. *Electronic Journal of Combininatorics 5,* 1.

GAREY, M. R., GRAHAM, R. L., JOHNSON, D. S., AND KNUTH, D. 1978. Complexity results for bandwidth minimization. *SIAM Journal on Applied Mathematics 34*, 477–495.

GAREY, M. R. AND JOHNSON, D. S. 1979. *Computers and intractability. A guide to the theory of NP-completeness.* Freeman and Company, New York.

GAREY, M. R., JOHNSON, D. S., MILLER, G. L., AND PAPADIMITRIOU, C. H. 1980. The complexity of coloring circular arcs and chords. *SIAM Journal on Algebraic and Discrete Methods 1,* 2, 216–227.

GAREY, M. R., JOHNSON, D. S., AND STOCKMEYER, L. 1976. Some simplified NP-complete graph problems. *Theoretical Computer Science 1*, 237–267.

GAVRIL, F. 1977. Some NP-complete problems on graphs. In *11th Conference on Information Sciences and Systems.* John Hopkins University, Baltimore, 91–95.

GIBBS, N. E., POOLE, JR., W. G., AND STOCKMEYER, P. K. 1976. An algorithm for reducing the bandwidth and profile of a sparse matrix. *SIAM Journal on Numerical Analysis 13,* 2, 236–250.

GLOVER, F. AND LAGUNA, M. 1997. *Tabu Search*, second ed. Kluwer, Boston.

GOLDBERG, M. AND MILLER, Z. 1988. A parallel algorithm for bisection width in trees. *Computers & Mathematics with Applications 15,* 4, 259–266.

GOLDBERG, M. K. AND KLIPKER, I. A. 1976. An algorithm for minimal numeration of tree vertices. *Sakharth. SSR Mecn. Akad. Moambe 81,* 3, 553–556. In Russian.

GOLDBERG, P. W., GOLUMBIC, M. C., KAPLAN, H., AND SHAMIR, R. 1995. Four strikes against physical mapping od DNA. *Journal of Computational Biology 2,* 1, 139–152.

GOLOVACH, P. A. 1997. The total vertex separation number of a graph. *Diskretnaya Matematika 9,* 4, 86–91. In Russian.

GOLOVACH, P. A. AND FOMIN, F. V. 1998. The total vertex separation number and the profile of graphs. *Diskretnaya Matematika 10,* 1, 87–94. In Russian.

GOLUMBIC, M. C. 1980. *Algorithmic graph theory and perfect graphs.* Academic Press, New York.

GOLUMBIC, M. C., KAPLAN, H., AND SHAMIR, R. 1994. On the complexity of DNA physical mapping. *Advances in Applied Mathematics 15*, 203–215.

GOLUMBIC, M. C., KAPLAN, H., AND SHAMIR, R. 1995. Graph sandwich problems. *Journal of Algorithms 19*, 449–473.

GOLUMBIC, M. C. AND SHAMIR, R. 1993. Complexity and algorithms for reasoning about time: A graph theoretical approach. *Journal of the ACM 40*, 1108–1113.

GOMORY, R. E. AND HU, T. C. 1975. *Multi-terminal flows in a network.* Studies in Mathematics, vol. 11. MAA, 172–199.

GREENLAW, R., HOOVER, H. J., AND RUZZO, W. L. 1995. *Limits to parallel computation: P-completeness theory.* Oxford University Press, New York.

GRIMMETT, G. R. 1999. *Percolation*, second ed. Springer–Verlag, Heidelberg.

GUPTA, A. 2001. Improved bandwidth approximation for trees and chordal graphs. *Journal of Algorithms 40,* 1, 24–36.

GURARI, E. AND SUDBOROUGH, I. H. 1984. Improved dynamic programming algorithms for the bandwidth minimization and the mincut linear arrangement problem. *Journal of Algorithms 5*, 531–546.

GUSTEDT, J. 1993. On the pathwidth of chordal graphs. *Discrete Applied Mathematics 45,* 3, 233–248.

GUU, CH.-J. 1997 The Circular Wirelength Problem for Hypercubes. Ph.D. thesis, Dept. of Mathematics, University of California, Riverside.

HAGER, W. W. 2002. Minimizing the profile of a symmetric matrix. *SIAM Journal on Scientific Computing 23,* 5, 1799–1816.

HANSEN, M. D. 1989. Approximation algorithms for geometric embeddings in the plane with applications to parallel processing problems. In *30th Annual Symposium on Foundations of Computer Science.* 604–609.

HARALAMBIDES, J. AND MAKEDON, F. 1997. Approximation algorithms for the bandwidth minimization problem for a large class of trees. *Theory of Computing Systems 30*, 67–90.

HARALAMBIDES, J., MAKEDON, F., AND MONIEN, B. 1991. Bandwidth minimization: an approximation algorithm for caterpillars. *Mathematical Systems Theory* 24, 169–177.

HARARY, F. 1967. Problem 16. In *Graph Theory and Computing*, M. Fiedler, Ed. Czechoslovak Academy Sciences, 161.

HARPER, L. H. 1964. Optimal assignments of numbers to vertices. *Journal of SIAM 12,* 1, 131–135.

HARPER, L. H. 1966. Optimal numberings and isoperimetric problems on graphs. *Journal of Combinatorial Theory 1,* 3, 385–393.

HARPER, L. H. 1970. Chassis layout and isoperimetric problems. Technical report SPS 37–66, vol II, Jet Propulsion Laboratory.

HARPER, L. H. 1977. Stabilization and the edgesum problem. *Ars Combinatoria 4*, 225–270.

HARPER, L. H. 2001. On the bandwidth of a hamming graph. Manuscript.

HEATH, L. S., LEIGHTON, F. T., AND ROSENBERG, A. L. 1992. Comparing queues and stacks as mechanisms for laying out graphs. *SIAM Journal on Discrete Mathematics 5,* 3, 398–412.

HEATH, L. S. AND PEMMARAJU, S. V. 1997. Stack and queue layouts of posets. *SIAM Journal on Discrete Mathematics 10,* 4, 599–625.

HEATH, L. S. AND PEMMARAJU, S. V. 1999. Stack and queue layouts of directed acyclic graphs. II. *SIAM Journal on Computing 28,* 5, 1588–1626.

HEATH, L. S., PEMMARAJU, S. V., AND RIBBENS, C. J. 1993. *Sparse Matrix-Vector Multiplication on a Small Linear Array.* Springer-Verlag.

HEATH, L. S., PEMMARAJU, S. V., AND TRENK, A. N. 1999. Stack and queue layouts of directed acyclic graphs. I. *SIAM Journal on Computing 28,* 4, 1510–1539.

HEATH, L. S. AND ROSENBERG, A. L. 1992. Laying out graphs using queues. *SIAM Journal on Computing 21,* 5, 927–958.

HECKMANN, R., KLASING, R., MONIEN, B., AND UNGER, W. 1998. Optimal embedding of complete binary trees into lines and grids. *Journal of Parallel and Distributed Computing 49,* 1, 40–56.

HELMBERG, C., RENDL, F., MOHAR, B., AND POLJAK, S. 1995. A spectral approach to bandwidth and separator problems in graphs. *Linear and Multilinear Algebra 39,* 1-2, 73–90.

HENDRICH, U. AND STIEBITZ, M. 1992. On the bandwidth of graph products. *Journal of information processing and cybernetics 28*, 113–125.

HENDRICKSON, B. AND LELAND, R. 1997. The Chaco user's guide: version 2.0. Technical report SAND94–2692, Sandia National Laboratories.
ftp://ftp.cs.sandia.gov/pub/papers/bahendr/guide.ps.gz.

HROMKOVIČ, J. AND MONIEN, B. 1992. *The bisection problem for graphs of degree* 4 *(configuring transputer systems).* Teubner, Stuttgart, 215–233.

HU, T. C. 1974. Optimum communication spanning trees. *SIAM Journal on Computing 3,* 188–195.

JANSON, S., ŁUCZAK, T., AND RUCINSKI, A. 2000. *Random graphs.* Wiley, New York.

JERRUM, M. AND SORKIN, G. 1998. The Metropolis algorithm for graph bisection. *Discrete Applied Mathematics 82,* 1-3, 155–175.

JOHNSON, D. S., ARAGON, C. R., MCGEOCH, L. A., AND SCHEVON, C. 1989. Optimization by simulated annealing: an experimental evaluation; part I, graph partitioning. *Operations Research 37,* 6, 865–892.

JOHNSON, D. S., LENSTRA, J. K., AND KAN, A. H. G. R. 1978. The complexity of the network design problem. *Networks 8,* 4, 279–285.

JUELS, A. 1996. Topics in black-box combinatorial optimization. Ph.D. thesis, University of California at Berkeley.

JUVAN, M. AND MOHAR, B. 1992. Optimal linear labelings and eigenvalues of graphs. *Discrete Applied Mathematics 36,* 2, 153–168.

KADLUCZKA, P. AND WALA, K. 1995. Tabu search and genetic algorithms for the generalized graph partitioning problem. *Control and cybernetics 24,* 4, 459–476.

KAPLAN, H. AND SHAMIR, R. 1996. Pathwidth, bandwidth and completion problems to proper interval graphs with small cliques. *SIAM Journal on Computing 25,* 3, 540–561.

KAPLAN, H. AND SHAMIR, R. 1999. Bounded degree interval sandwich problems. *Algorithmica 24,* 2, 96–104.

KAPLAN, H., SHAMIR, R., AND TARJAN, R. E. 1999. Tractability of parameterized completion problems on chordal, strongly chordal, and proper interval graphs. *SIAM Journal on Computing 28,* 5, 1906–1922.

KARGER, D. R. 1999. A randomized fully polynomial time approximation scheme for the all-terminal network reliability problem. *SIAM Journal on Computing 29,* 2, 492–514.

KARP, R. M. 1993. Mapping the genome: some combinatorial problems arising in molecular biology. In *Proceedings of the 25th Annual ACM Symposium on the Theory of Computing.* 278–285.

KARPINSKI, M., WIRTGEN, J., AND ZELIKOVSKY, A. 1997. An approximating algorithm for the bandwidth problem on dense graphs. Technical report TR 97-017, Electronic Colloquium on Computational Complexity.

KARYPIS, G. 2001. Metis's home page. Web page. http://www-users.cs.umn.edu/~karypis/metis.

KENDALL, D. G. 1969. Incidence matrices, interval graphs, and seriation in archeology. *Pacific Journal of Mathematics 28*, 565–570.

KERNIGHAN, B. W. AND LIN, S. 1970. An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal* 49, 291–307.

KING, I. P. 1970. An automatic reordering schema for simultaneous equations derived from network system. *International Journal for Numerical Methods in Engineering 2*, 523–533.

KINNERSLEY, N. G. 1992. The vertex separation number of a graph equals its path-width. *Information Processing Letters 42,* 6, 345–350.

KIROUSIS, L. M. AND PAPADIMITRIOU, C. H. 1986. Searching and pebbling. *Theoretical Computer Science 47*, 205–216.

KLASING, R. 1998. The relationship between the gossip complexity in vertex-disjoint paths mode and the vertex bisection width. *Discrete Applied Mathematics 83*, 229–246.

KLEITMAN, D. J. AND VOHRA, R. V. 1990. Computing the bandwidth of interval graphs. *SIAM Journal on Discrete Mathematics 3,* 3, 373–375.

KLOKS, T., KRATSCH, D., AND MÜLLER, H. 1998. Bandwidth of chain graphs. *Information Processing Letters 68,* 6, 313–315.

KLOKS, T., KRATSCH, D., AND MÜLLER, H. 1999. Approximating the bandwidth for asteroidal triple-free graphs. *Journal of Algorithms 32,* 1, 41–57.

KRATSCH, D. 1987. Finding the minimum bandwidth of an interval graph. *Information and Computation 74,* 2, 140–158.

KUMFERT, G. AND POTHEN, A. 1997. Two improved algorithms for reducing the envelope and wavefront. *BIT 37,* 3, 001–032.

KUO, D. AND CHANG, G. J. 1994. The profile minimization problem in trees. *SIAM Journal on Computing 23*, 71–81.

LAI, Y.-L. 1997. Bandwidth, edgesum and profile of graphs. Ph.D. thesis, Dept. of Computer Science, Western Michigan Univ.

LAI, Y.-L. 2001. On the profile of the tensor product of path with complete bipartite graphs. Manuscript (provided by S. Bezrukov).

LAI, Y.-L., LIU, J., AND WILLIAMS, K. 1994. Bandwidth for the sum of k graphs. *Ars Combinatoria 37*, 149–155.

LAI, Y.-L. AND WILLIAMS, K. 1994. The edgesum of the sum of k sum deterministic graphs. In *Proceedings of the Twenty-fifth Southeastern International Conference on Combinatorics, Graph Theory and Computing.* Congressus Numerantium, vol. 102. Utilitas Math., 231–236.

LAI, Y.-L. AND WILLIAMS, K. 1995. Bandwidth of the strong product of paths and cycles. In *Proceedings of the Twenty-sixth Southeastern International Conference on Combinatorics, Graph Theory and Computing.* Congressus Numerantium, vol. 109. Utilitas Math., 123–128.

LAI, Y.-L. AND WILLIAMS, K. 1997. On bandwidth for the tensor product of paths and cycles. *Discrete Applied Mathematics. 73,* 2, 133–141.

LAI, Y.-L. AND WILLIAMS, K. 1999. A survey of solved problems and applications on bandwidth, edgesum, and profile of graphs. *Journal of Graph Theory 31,* 2, 75–94.

LANG, K. AND RAO, S. 1993. Finding near-optimal cuts: An empirical evaluation. In *Proceedings of the 4th Annual ACM-SIAM Symposium on Discrete Algorithms.* 212–221.

LEIGHTON, F. T. 1993. *Introduction to parallel algorithms and architectures: arrays, trees, hypercubes.* Morgan Kaufmann, San Mateo.

LEIGHTON, T. AND RAO, S. 1999. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM 46,* 6, 787–832.

LEISERSON, C. E. 1980. Area-efficient graph layouts (for VLSI). In *Proceedings of the 21st Annual Symposium on Foundations of Computer Science.* 270–281.

LELAND, R. AND HENDRICKSON, B. 1994. An empirical study of static load balancing algorithms. In *Scalable High-Performance Computing Conference.* IEEE Computer Society Press, 682–685.

LENGAUER, T. 1981. Black-white pebbles and graph separation. *Acta Informatica 16*, 465–475.

LENGAUER, T. 1982. Upper and lower bounds on the complexity of the min-cut linear arrangements problem on trees. *SIAM Journal on Algebraic and Discrete Methods 3*, 99–113.

LEPIN, V. V. 1986. Profile minimization problem of matrices and graphs. *Academy of Sciences of Belarus. Institute of Mathematics. 33,* 269. In Russian.

LEWIS, R. 1994. Simulated annealing for profile and fill reduction of sparse matrices. *International Journal for Numerical Methods in Engineering 37,* 6.

LIN, Y. AND YUAN, J. 1994a. Profile minimization problem for matrices and graphs. *Acta Mathematicae Applicatae Sinica. English Series 10,* 1, 107–112.

LIN, Y. X. 1994. Two-dimensional bandwidth problem. In *Combinatorics, graph theory, algorithms and applications (Beijing, 1993).* World Sci. Publishing, 223–232.

LIN, Y. X. AND YUAN, J. J. 1994b. Profile minimization problem for matrices and graphs. *Acta Mathematicae Applicatae Sinica. English Series. Yingyong Shuxue Xuebao 10,* 1, 107–112.

LINDSEY, J. H. 1964. Assignments of numbers to vertices. *American Mathematical Monthly* 71, 508–516.

LIPTON, R. J. AND TARJAN, R. E. 1979. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics 36*, 177–189.

LIU, H. E. AND YUAN, J. J. 1995. The cutwidth problem for graphs. *Gaoxiao Yingyong Shuxue Xuebao. Applied Mathematics. A Journal of Chinese Universities. Series A 10,* 3, 339–348. In Chinese.

LIU, J. 1992. On bandwidth sum for the composition of paths and cycles. Technical report Rep/92-06, Dept. of Computer Science, Western Michigan Univ.

LIU, J. AND WILLIAMS, K. 1995. On bandwidth and edgesum for the composition of two graphs. *Discrete Mathematics 143,* 1-3, 159–166.

LUCZAK, M. L. AND MCDIARMID, C. 2001. Bisecting sparse random graph. *Random Structures & Algorithms 18*, 31–38.

MACGREGOR, R. M. 1978. On partitioning a graph: a theoretical and empirical study. Ph.D. thesis, University of California, Berkeley.

MAHESH, R., RANGAN, C. P., AND SRINIVASAN, A. 1991. On finding the minimum bandwidth of interval graphs. *Information and Computation 95,* 2, 218–224.

MAI, J. H. 1996. Profiles of some classes of condensable graphs. *Journal of Systems Science and Mathematical Sciences. Xitong Kexue yu Shuxue 16,* 2, 141–148.

MAI, J. H. AND LUO, H. P. 1984. Some theorems on the bandwidth of a graph. *Acta Mathematicae Applicatae Sinica. Yingyong Shuxue Xuebao 7,* 1, 86–95.

MAKEDON, F., PAPADIMITRIOU, C. H., AND SUDBOROUGH, I. H. 1985. Topological bandwidth. *SIAM Journal on Algebraic and Discrete Methods 6,* 3, 418–444.

MAKEDON, F., SHEINWALD, D., AND WOLFSTHAL, Y. 1993. A simple linear-time algorithm for the recognition of bandwidth-2 biconnected garphs. *Information Processing Letters 46*, 103–107.

MAKEDON, F. AND SUDBOROUGH, I. H. 1989. On minimizing width in linear layouts. *Discrete Applied Mathematics 23,* 3, 243–265.

MANABE, Y., HAGIHARA, K., AND TOKURA, N. 1984. The minimum bisection widths of the cube-connected cycles graph and cube graph. *Systems-Computers-Controls. The Transactions of the Institute of Electronics and Communication Engineers of Japan 15,* 6, 9–18.

MILLER, Z. 1991. Multidimensional bandwidth in random graphs. In *Graph theory, combinatorics, and applications. Vol. 2 (Kalamazoo, MI, 1988).* Wiley, New York, 861–870.

MITCHISON, G. AND DURBIN, R. 1986. Optimal numberings of an $n \times n$ array. *SIAM Journal on Algebraic and Discrete Methods 7,* 4, 571–582.

MOHAR, B. AND POLJAK, S. 1993. Eigenvalues in combinatorial optimization. In *Combinatorial and Graph-Theoretical Problems in Linear Algebra*, R. A. Brualdi, S. Friedland, and V. Klee, Eds. IMA Volumes in Mathematics and its Applications, vol. 50. Springer–Verlag, 107–151.

MONIEN, B. 1986. The bandwidth minimization problem for caterpillars with hair length 3 is NP-complete. *SIAM Journal on Algebraic and Discrete Methods 7,* 4, 505–512.

MONIEN, B. AND SUDBOROUGH, I. H. 1988. Min Cut is NP-complete for edge weighted trees. *Theoretical Computer Science 58,* 1-3, 209–229.

MONIEN, B. AND SUDBOROUGH, I. H. 1990. Embedding one interconnection network in another. In *Computational graph theory,* G. Tinhofer, E. Mayr, H. Noltemeier, and M. M. Syslo, Eds. Computing Supplementa, vol. 7. Springer–Verlag, 257–282.

MURADYAN, D. 1999. Computer Science & Information Technologies Conference.

MURADYAN, D. O. 1982. Minimal numberings of a two-dimensional cylinder. *Akademiya Nauk Armyanskoĭ SSR. Doklady 75,* 3, 114–119. In Russian.

MURADYAN, D. O. 1985. Some estimates for the length of an arbitrary graph. *Mat. Voprosy Kibernet. Vychisl. Tekhn.* 14, 79–86, 126. In Russian.

MURADYAN, D. O. 1986. A polynomial algorithm for finding the bandwidth of interval graphs. *Akademiya Nauk Armyanskoĭ SSR. Doklady 82,* 2, 64–66. In Russian.

MURADYAN, D. O. AND PILIPOSYAN, T. E. 1980. Minimal numberings of vertices of a rectangular lattice. *Akad. Nauk. Armjan. SRR 1,* 70, 21–27. In Russian.

MURADYAN, D. O. AND PILIPOSYAN, T. E. 1980. The problem of finding the length and width of the complete p-partite graph. *Uchen. Zapiski Erevan. Gosunivers. 2*, 18–26. In Russian.

MUTZEL, P. 1995. A polyhedral approach to planar augmentation and related problems. In *Algorithms — ESA'95*, P. Spirakis, Ed. Lecture Notes in Computer Science, vol. 979. Springer–Verlag, 497–507.

NAKANO, K. 1994. Linear layouts of generalized hypercubes. In *Graph-Theoretic Concepts in Computer Science*, J. van Leeuwen, Ed. Lecture Notes in Computer Science, vol. 790. Springer–Verlag, 364–375.

NIEPEL, L. AND TOMASTA, P. 1981. Elevation of a graph. *Czechoslovak Mathematical Journal 31(106),* 3, 475–483.

PAPADIMITRIOU, C. 1976. The NP-completeness of the bandwidth minimization problem. *Computing 16*, 263–270.

PAPADIMITRIOU, C. H. AND SIDERI, M. 1996. The bisection width of grid graphs. *Mathematical Systems Theory 29,* 2, 97–110.

PENROSE, M. 1995. Single linkage clustering and continuum percolation. *Journal of Multivariate Analysis 53*, 94–109.

PENROSE, M. 2000. Vertex ordering and partitioning problems for random spatial graphs. *The Annals of Applied Probability 10*, 517–538.

PETIT, J. 1998. Approximation heuristics and benchmarkings for the MINLA problem. In *Alex '98 — Building bridges between theory and applications*, R. Battiti and A. Bertossi, Eds. 112–128.

PETIT, J. 2000. Combining spectral sequencing and parallel simulated annealing for the minLA problem. Technical report LSI-01-13-R, Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya.

PETIT, J. 2001a. Experiments for the MINLA problem. Technical report LSI-R01-7-R, Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya.

PETIT, J. 2001b. Layout problems. Ph.D. thesis, Universitat Politècnica de Catalunya.

PREIS, R. AND DIEKMANN, R. 1996. The Party partitioning library user guide. Technical report TR-RSFB-96-024, Universität Paderborn.

RAO, S. AND RICHA, A. W. 1998. New approximation techniques for some ordering problems. In *Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms.* 211–218.

RASPAUD, A., SÝKORA, O., AND VRŤO, I. 1995. Cutwidth of the de Bruijn graph. *RAIRO Informatique Théorique et Applications 29,* 6, 509–514.

RASPAUD, A., SÝKORA, O., AND VRŤO, I. 2000. Congestion and dilation, similarities and differences—a survey. In *Proceedings 7th International Colloquium on Structural Information and Communication Complexity.* Carleton Scientific, 269–280.

RAVI, R., AGRAWAL, A., AND KLEIN, P. 1991. Ordering problems approximated: single-processor scheduling and interval graph completition. In *Automata, Languages and Programming*, J. Leach, B. Monien, and M. Rodriguez, Eds. Lecture Notes in Computer Science, vol. 510. Springer–Verlag, 751–762.

ROBERTSON, N. AND SEYMOUR, P. D. 1985. Graph minors—a survey. In *Surveys in combinatorics.* Cambridge University Press, 153–171.

ROLIM, J., SÝKORA, O., AND VRŤO, I. 1995. Optimal cutwidths and bisection widths of 2- and 3-dimensional meshes. In *Graph theoretic concepts in computer science.* Lecture Notes in Computer Science, vol. 1017. Springer, 252–264.

ROLIM, J., TVRDÍK, J., TRDLIČKA, J., AND VRŤO, I. 1998. Bisecting de Bruijn and Kautz graphs. *Discrete Applied Mathematics 85*, 87–97.

SAAD, Y. 1996. *Iterative methods for sparse linear systems.* PWS Publishing Company, Boston.

SAXE, J. B. 1980. Dynamic-programming algorithms for recognizing small-bandwidth graphs in polynomial time. *SIAM Journal on Algebraic Discrete Methods 1,* 4, 363–369.

SEYMOUR, P. D. 1995. Packing directed circuits fractionally. *Combinatorica 15,* 2, 281–288.

SEYMOUR, P. D. AND THOMAS, R. 1994. Call routing and the ratcatcher. *Combinatorica 14,* 2, 217–241.

SHAHROKHI, F., SÝKORA, O., SZÉKELY, L. A., AND VRŤO, I. 1997. *Crossing numbers: bounds and applications*. János Bolyai Math. Soc., Budapest, 179–206.

SHAHROKHI, F., SÝKORA, O., SZÉKELY, L. A., AND VRŤO, I. 2001. On bipartite drawings and the linear arrangement problem. *SIAM Journal on Computing 30,* 6, 1773–1789.

SHILOACH, Y. 1979. A minimum linear arrangement algorithm for undirected trees. *SIAM Journal on Computing 8,* 1, 15–32.

SHING, M. T. AND HU, T. C. 1986. Computational complexity of layout problems. In *Layout design and verification*, T. Ohtsuki, Ed. Advances in CAD for VLSI, vol. 4. North–Holland, Amsterdam, 267–294.

SIMON, H. D. AND TENG, S.-H. 1997. How good is recursive bisection? *SIAM Journal on Scientific Computing 18,* 5, 1436–1445.

SKODINIS, K. 2000. Computing optimal linear layouts of trees in linear time. In *Algorithms — ESA 2000*, M. Paterson, Ed. Lecture Notes in Computer Science, vol. 1879. Springer–Verlag, 403–414.

SOUMYANATH, K. AND DEOGUN, J. S. 1990. On the bisection width of partial k-trees. In *Proceedings of the Twentieth Southeastern Conference on Combinatorics, Graph Theory, and Computing*. Congressus Numerantium, vol. 74. Utilitas Math., 25–37.

SPIELMAN, D. A. AND TENG, S.-H. 1996. Spectral partitioning works: Planar graphs and finite element meshes. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*. 96–105.

SPRAGUE, A. P. 1994. An $O(n \log n)$ algorithm for bandwidth of interval graphs. *SIAM Journal on Discrete Mathematics 7,* 2, 213–220.

STACHO, L. AND VRŤO, I. 1998. Bisection width of transposition graphs. *Discrete Applied Mathematics 84,* 1-3, 221–235.

SÝKORA, O. AND VRŤO, I. 1993. Edge separators for graphs of bounded genus with applications. *Theoretical Computer Science 112*, 419–429.

TEWARSON, R. 1973. *Sparse matrices*. Academic Press, New York.

THILIKOS, D. M., SERNA, M. J., AND BODLAENDER, H. L. 2000. Constructive linear time algorithms for small cutwidth and carving-width. In *ISAAC 2000 Algorithms and Computation*, D. T. Lee and S.-H. Teng, Eds. Lecture Notes in Computer Science, vol. 1969. Springer–Verlag, 192–203.

THILIKOS, D. M., SERNA, M. J., AND BODLAENDER, H. L. 2001. A polynomial time algorithm for the cutwidth of bounded degree graphs with small treewidth. In *Algorithms, ESA 2001*, F. Meyer auf der Heide, Ed. Lecture Notes in Computer Science, vol. 2161. Springer–Verlag, 380–390.

THOMPSON, C. D. 1979. Area-time complexity in VLSI design. In *Proceedings of the 11th Annual ACM Symposium on the Theory of Computing*. 81–88.

TURNER, J. S. 1986. On the probable performance of heuristics for bandwidth minimization. *SIAM Journal on Computing 15,* 2, 561–580.

UNGER, W. 1998. The complexity of the approximation of the bandwidth problem. In *37th Annual Symposium on Foundations of Computer Science*. 82–91.

VAZIRANI, V. V. 2001. *Approximation algorithms*. Springer, Berlin.

VRŤO, I. 2000. Cutwidth of the r-dimensional mesh of d-ary trees. *RAIRO Informatique Théorique et Applications 34,* 6, 515–519.

VRŤO, I. 2002. Crossing numbers of graphs: A bibliography. ftp://ftp.ifi.savba.sk/pub/imrich/crobib.ps.gz.

WILLIAMS, K. 1993. On the minimum sum of the corona of two graphs. In *Proceedings of the Twenty-fourth Southeastern International Conference on Combinatorics, Graph Theory, and Computing*. Congressus Numerantium, vol. 94. Utilitas Math., 43–49.

WILLIAMS, K. 1994. On bandwidth and edgesum for the tensor product of paths with complete bipartite graphs. In *Proceedings of the Twenty-fifth Southeastern International Conference on Combinatorics, Graph Theory and Computing)*. Congressus Numerantium, vol. 102. Utilitas Math., 183–190.

WILLIAMS, K. 1996. On bandwidth for the tensor product of paths and cycles with complete graphs. *Bulletin of the Institute of Combinatorics and its Applications 16*, 41–48.

WU, B. Y., LANCIA, G., BAFNA, V., CHAO, K.-M., RAVI, R., AND TANG, C. Y. 1999. A polynomial time approximation scheme for minimum routing cost spanning trees. *SIAM Journal on Computing 29,* 3, 761–778.

YANNAKAKIS, M. 1985. A polynomial algorithm for the min-cut linear arrangement of trees. *Journal of the ACM 32,* 4, 950–988.

YUAN, J., LIN, Y., LIU, Y., AND WANG, S. 1998. NP-completeness of the profile problem and the fill-in problem on cobipartite graphs. *Journal of Mathematical Study 31,* 3, 239–243.

ZHOU, S. AND YUAN, J. 1998. Harper-type lower bounds and the bandwidths of the compositions of graphs. *Discrete Mathematics 181,* 1-3, 255–266.

Received ???; revised ???; accepted ??