A Fixed-Parameter Approach to Two-Layer Planarization ⋆

V. Dujmović⋆⋆, M. Fellows¶, M. Hallett⋆⋆, M. Kitching⋆⋆,
G. Liotta⋆⋆⋆, C. McCartin§, N. Nishimura†, P. Ragde†,
F. Rosamond¶, M. Suderman⋆⋆, S. Whitesides⋆⋆, D. R. Wood‡

Abstract. A bipartite graph is *biplanar* if the vertices can be placed on two parallel lines (*layers*) in the plane such that there are no edge crossings when edges are drawn straight. The 2-Layer Planarization problem asks if k edges can be deleted from a given graph G so that the remaining graph is biplanar. This problem is NP-complete, as is the 1-Layer Planarization problem in which the permutation of the vertices in one layer is fixed. We give the following fixed parameter tractability results: an $O(k \cdot 6^k + |G|)$ algorithm for 2-Layer Planarization and an $O(3^k \cdot |G|)$ algorithm for 1-Layer Planarization, thus achieving *linear* time for fixed k.

1 Introduction

In a *2-layer drawing* of a bipartite graph $G = (A, B; E)$, the vertices in A and B are positioned on two distinct parallel lines in the plane, and the edges are drawn straight. Such drawings have applications in visualization [1, 10], DNA mapping [19], and VLSI layouts [11]; a recent survey [13] gives more details.

A *biplanar* graph is a bipartite graph that admits a 2-layer drawing with no edge crossings. Consider a 2-layer drawing of a bipartite graph produced by first drawing a maximum biplanar subgraph with no crossings and then drawing all the remaining edges. Such a drawing will almost certainly have more crossings than is necessary; however, there is some experimental evidence to suggest that drawings in which all crossings occur in a few edges are more readable than drawings with fewer total crossings [12]. Maximizing the size of a biplanar subgraph is equivalent to minimizing the number of edges not in it. This leads naturally to the definition of the 2-Layer Planarization problem: given a graph G (not

⋆ Research initiated at the International Workshop on Fixed Parameter Tractability in Graph Drawing, Bellairs Research Institute of McGill University, Holetown, Barbados, Feb. 9-16, 2001, organized by S. Whitesides. Research of Canada-based authors supported by NSERC. Research of G. Liotta supported by CNR and MURST. Research of D. Wood supported by the ARC, and completed while visiting McGill University. Contact author: S. Whitesides, School of Computer Science, McGill University, 3480 University St., Montréal, Canada, e-mail: `sue@cs.mcgill.ca`

⋆⋆ McGill University, Canada. ¶ University of Victoria, Canada.
⋆⋆⋆ Università di Perugia, Italy. § Victoria University of Wellington, New Zealand.
† University of Waterloo, Canada. ‡ The University of Sydney, Australia.

necessarily bipartite), and integer k, can G be made biplanar by deleting at most k edges? This problem is the focus of this paper.

Two-layer drawings are of fundamental importance to the "Sugiyama" approach to multilayer drawing [16]. This method first assigns vertices to layers, then makes repeated sweeps up and down the layers to determine an ordering of the vertices in one layer given the ordering for the preceding layer. This involves solving the 1-LAYER PLANARIZATION problem: given a bipartite graph $G = (A, B; E)$, a permutation π of A, and an integer k, can at most k edges be deleted to permit G to be drawn without crossings with π as the ordering of vertices in A? In this paper, we present results on this problem as well.

Instead of deleting edges, one can seek to minimize the number of crossings in a 2-layer drawing (here the input graph must be bipartite). The corresponding problems are called 1- and 2-LAYER CROSSING MINIMIZATION. Both of these well studied problems are NP-complete [7, 6]. The 2-LAYER PLANARIZATION problem is NP-complete [5, 17] even for planar biconnected bipartite graphs with vertices in respective bipartitions having degree two and three [5]. The 1-LAYER PLANARIZATION problem is NP-complete even for graphs with only degree-1 vertices in the fixed layer and vertices of degree at most 2 in the other layer [5], i.e., for collections of 1- and 2-paths. With the order of the vertices in both layers fixed the problem can be solved in polynomial time [5, 14].

Integer linear programming algorithms have been presented for 1- and 2-LAYER CROSSING MINIMIZATION [9, 18]. Jünger and Mutzel [9] survey numerous heuristics proposed for both problems, and experimentally compare their performance with the optimal solutions. They report that the *iterated barycentre* method of Sugiyama *et al.* [16] performs best in practice. However, from a theoretical point of view the *median heuristic* of Eades and Wormald [6] is a better approach for 1-LAYER CROSSING MINIMIZATION. In particular, the median heuristic is a linear time 3-approximation algorithm, whereas the barycentre heuristic is a $\Theta(\sqrt{n})$-approximation algorithm [6]. Furthermore, for graphs with maximum degree three in the free layer, the median heuristic is a 2-approximation algorithm for this problem [1]. Recently, Shahrokhi *et al.* [15] devised a polynomial time algorithm that approximates 2-LAYER CROSSING MINIMIZATION within a factor of $O(\log n)$ for a wide class of n-vertex graphs.

Despite the practical significance of the problems, 1- and 2-LAYER PLANARIZATION have received less attention in the graph drawing literature than their crossing minimization counterparts. Integer linear programming algorithms have been presented [12, 14]. For acyclic graphs G, Shahrokhi *et al.* [15] present an $O(n)$ time dynamic programming algorithm for 2-LAYER PLANARIZATION of weighted acyclic graphs, for which the objective is to minimize the total weight of deleted edges.

1.1 Fixed Parameter Tractability and Our Results

When the maximum number k of allowed edge deletions is small, an algorithm for 1- or 2-LAYER PLANARIZATION whose running time is exponential in k but polynomial in the size of the graph may be useful. The theory of parameterized complexity [2] addresses complexity issues of this nature, in which a problem is specified in terms of one or more parameters. Such a problem with input size n and parameter size k is *fixed parameter tractable*, or in the class *FPT*, if there is an algorithm to solve the problem in $f(k) \cdot n^{\alpha}$ time, for some function f and constant α. A problem in FPT is thus solvable in polynomial time for fixed k.

In a companion paper [3], we proved using bounded pathwidth techniques that the h-layer generalizations of the 2-Layer Crossing Minimization and 2-Layer Planarization problems are in FPT, where h is also considered a parameter of the problem. The 1-layer versions of these problems can also be solved using this approach. Unfortunately, a pathwidth-based approach is not particularly practical, since the running time of the algorithms is $O(2^{32(h+2k)^3} n)$.

In this paper we apply other methods from the theory of fixed parameter tractability to obtain more practical algorithms for the 1- and 2-Layer Planarization problems. In particular, using a "kernelization" approach we obtain an $O(\sqrt{k} \cdot 17^k + |G|)$ time algorithm for 2-Layer Planarization in a graph G, which we improve to $O(k \cdot 6^k + |G|)$ using a "bounded search tree" approach combined with kernelization. Here $|G| = |V| + |E|$ for a graph $G = (V, E)$. For small values of k, the 2-Layer Planarization problem is thus solvable optimally in a reasonable amount of time. We then refine this second algorithm to solve the 1-Layer Planarization problem in $O(3^k \cdot |G|)$ time.

This paper is organized as follows. After definitions and preliminary results in Section 2, we present the "kernelization" approach for 2-Layer Planarization in Section 3. Section 4 describes our "bounded search tree" algorithm for the same problem. In Section 5 we consider the 1-Layer Planarization problem, and present a bounded search tree algorithm for its solution. We conclude in Section 6 with some observations and open problems.

2 Preliminaries

In this paper each graph $G = (V, E)$ is simple and undirected. The subgraph of G induced by a subset E' of edges is denoted by $G[E']$. A vertex with degree one is a *leaf*. If vw is the edge incident to a leaf w, then we say w is a *leaf at* v and vw is a *leaf-edge at* v. The number of non-leaf edges at a vertex v is denoted by $\deg'_G(v)$, or $\deg'(v)$ if the graph is clear from the context. A bipartite graph is *biplanar* if it admits a biplanar drawing. A *claw* is a complete bipartite subgraph $K_{1,3}$. The *2-claw* is the graph consisting of one degree-3 vertex (the *centre*), which is adjacent to three degree-2 vertices, each of which is adjacent to the centre and one leaf. A graph is a *caterpillar* if deleting all leaves produces a (possibly empty) path. This path is the *spine* of the caterpillar. If v is an endpoint of the spine of a caterpillar then there is at least one leaf at v (otherwise v itself would be a leaf.) These graphs are illustrated in Fig. 1.

We define $V_3 = \{v \in V : \deg'(v) \geq 3\}$, and $V'_3 = \{w \in V \setminus V_3 : \deg(w) \geq 1, \exists v \in V_3,\ s.t.\ vw \in E\}$. That is, V_3 is the set of vertices with at least three non-leaf neighbours, and V'_3 is the set of non-leaf neighbours of vertices in V_3 that are not themselves in V_3. These sets will be important in the proofs of correctness of our algorithms. Observe that the centre of a 2-claw is in V_3. In Fig. 1 and subsequent illustrations, vertices in V_3 are black and vertices in V_3' are gray.

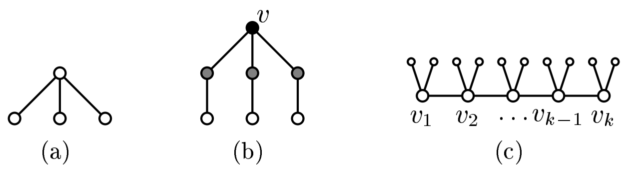


Fig. 1. (a) claw, (b) 2-claw centred at v, (c) caterpillar.

A set of edges S of a (not necessarily bipartite) graph G is called a *biplanarizing set* if $G \setminus S$ is biplanar. The *bipartite planarization number* of a graph G, denoted by $\mathsf{bpr}(G)$, is the size of a minimum biplanarizing set for G. Thus the 2-Layer Planarization problem is: given a graph G and integer k, is $\mathsf{bpr}(G) \leq k$? For a given bipartite graph $G = (A, B; E)$ and permutation π of A, the *1-layer bipartite planarization number* of G, denoted $\mathsf{bpr}(G, \pi)$, is the minimum number of edges in G whose deletion produces a graph that admits a biplanar drawing with π as the ordering of the vertices in A. The 1-Layer Planarization problem asks if $\mathsf{bpr}(G, \pi) \leq k$.

Biplanar graphs are easily characterized, and there is a simple linear-time algorithm to recognize biplanar graphs, as the next lemma makes clear.

Lemma 1 ([4, 8, 17]). *Let G be a graph. The following are equivalent:*
(a) G is biplanar.
(b) G is a forest of caterpillars (see Fig. 2).
(c) G is acyclic and contains no 2-claw.
(d) The graph obtained from G by deleting all leaves is a forest and contains no vertex of degree three or greater.

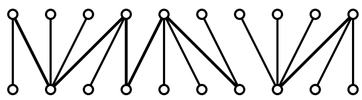


Fig. 2. A biplanar graph is a forest of caterpillars. Spine edges are dark

Since our algorithms seek to find biplanar subgraphs, some terminology concerning caterpillars will prove useful. A path $(v_1, v_2, \ldots, v_k)$ with $\deg_G(v_1) \geq 3$, $\deg_G(v_k) = 1$, and $\deg_G(v_i) = 2$, $1 < i < k$, is called a *pendant* path. A path $(v_1, v_2, \ldots, v_k)$ with $\deg_G(v_1) \geq 3$, $\deg_G(v_k) \geq 3$, and $\deg_G(v_i) = 2$, $1 < i < k$, is called an *internal* path. A *component* caterpillar of a graph is a connected component that is a caterpillar. A *pendant* caterpillar is a caterpillar subgraph C with spine $(v_2, v_3, \ldots, v_k)$ such that there is a leaf edge v_1v_2 of C with $\deg'_G(v_1) \geq 3$, $\deg'_G(v_i) = 2$ for all i, $1 < i < k$, and $\deg'_G(v_k) = 1$. An *internal* caterpillar is a caterpillar subgraph C with spine $(v_2, v_3, \ldots, v_{k-1})$ such that there are leaf-edges v_1v_2 and $v_{k-1}v_k$ of C with $\deg'_G(v_1) \geq 3$, $\deg'_G(v_k) \geq 3$, and $\deg'_G(v_i) = 2$

for all i, $1 < i < k$. A pendant (or internal) path (or caterpillar) is said to be *connected* at v_1 (and v_k), its *connection points*. Note that the leaves of an internal or pendant caterpillar in G are leaves of G.

A graph consisting of a cycle and some number of leaves (possibly zero) is a *wreath*. A *component wreath* of a graph is a connected component that is a wreath. A *pendant wreath connected at* v_1 is a wreath subgraph with cycle $(v_1, v_2, \ldots, v_k)$ such that $\deg'_G(v_1) \geq 3$ and $\deg'_G(v_i) = 2$ for all i, $1 < i \leq k$. These graphs are illustrated in Fig. 3.

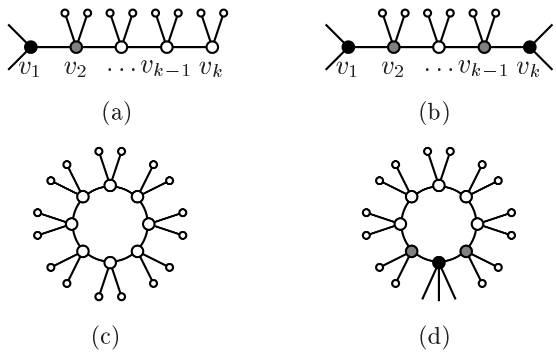


Fig. 3. (a) pendant caterpillar, (b) internal caterpillar, (c) wreath, (d) pendant wreath.

3 Kernelization

A basic method for developing FPT algorithms is to reduce a parameterized problem instance I to an "equivalent" instance I_{kr} (the *kernel*), where the size of I_{kr} is bounded by some function of the parameter. Then the instance I_{kr} is solved using an exhaustive search method, and its solution determines a solution to the original instance I. Downey and Fellows [2, Chapter 3.2] survey this general approach, which is known as *kernelization*.

Here is an overview of our kernelization algorithm for the 2-Layer Planarization problem, which operates in two phases. In the first phase we identify a set of edges $\mathcal{S}_G$ contained in the pendant and component wreaths of G that may as well be in a biplanarizing set. In the second phase, we collapse "long" internal and pendant caterpillars to "short" internal and pendant caterpillars. Intuitively, if a "short" internal or pendant caterpillar can be drawn without crossings then so can a "long" internal or pendant caterpillar. We obtain a graph G_{kr} such that $\mathsf{bpr}(G) = \mathsf{bpr}(G_{\mathsf{kr}}) + |\mathcal{S}_G|$. We then prove that if G_{kr} satisfies a necessary condition for $\mathsf{bpr}(G_{\mathsf{kr}}) \leq k - |\mathcal{S}_G|$ then $|G_{\mathsf{kr}}| \in O(k)$; the condition can be checked in $O(|G_{\mathsf{kr}}|)$ time. Exhaustive search determines if G_{kr} has a biplanarizing set with at most $k - |\mathcal{S}_G|$ edges; if so, this set plus $|\mathcal{S}_G|$ forms a biplanarizing set for G with at most k edges.

We now describe the first phase of the kernelization which constructs $\mathcal{S}_G$:

1. For each component wreath, add to $\mathcal{S}_G$ a cycle-edge from it.

2. For each pendant 3-cycle, add to $\mathcal{S}_G$ the edge not incident with its connection point.
3. For each pendant wreath that is not a 3-cycle, add to $\mathcal{S}_G$ a cycle-edge incident to its connection point.

The proof of the following lemma is relatively long and similar to that of the upcoming Lemma 3. Due to space limitations it is omitted here.

Lemma 2. *For every graph G, $\mathsf{bpr}(G) \le k$ if and only if $\mathsf{bpr}(G \setminus \mathcal{S}_G) \le k - |\mathcal{S}_G|$.* □

We now describe the second phase of the kernelization. It will be convenient to assume that $G = (V, E)$ is a graph with no pendant or component wreaths; i.e., $\mathcal{S}_G = \emptyset$. If G has $\mathcal{S}_G \ne \emptyset$, by Lemma 2, we can instead work with $G' = G \setminus \mathcal{S}_G$.

As illustrated in Fig. 4, apply the following rules to G to obtain a graph $G_{\mathsf{kr}} = (V_{\mathsf{kr}}, E_{\mathsf{kr}})$ called the *kernel* of G.

(a) Replace a set of leaf-edges at $v \in V_3$ by a single leaf-edge at v.
(b) Leave an internal path with at most three edges unchanged in G_{kr} .
(c) Replace an internal path with at least four edges and with endpoints x and y by a claw with leaves at x and y.
(d) Replace an internal caterpillar with at least one leaf at a vertex on the spine and with connection points x and y by a claw with leaves at x and y.
(e) Replace a pendant caterpillar connected at x by a 2-path connected at x.
(f) Delete a component caterpillar.

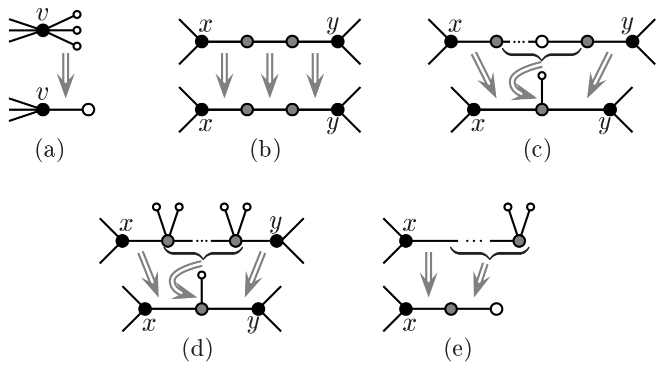


Fig. 4. Second phase of the kernelization.

Clearly G_{kr} can be constructed in $O(|G|)$ time, and every edge of G is mapped to an edge of G_{kr}. Thus, the construction of G_{kr} defines a function $f : E \to E_{\mathsf{kr}}$, indicated by arrows in Fig. 4. Observe that the non-leaf edges incident to vertices in V_3 are preserved under f. Let $E_{2,3}$ be the set of edges with both endpoints in $V_3 \cup V_3'$. Let $E_{2,3}^*$ be the set of edges in $E_{2,3}$ with at least one endpoint in V_3. Clearly f restricted to $E_{2,3}^*$ is 1-1.

Lemma 3. *For every graph G with $\mathcal{S}_G = \emptyset$, $\mathsf{bpr}(G) \le k$ if and only if $\mathsf{bpr}(G_{\mathsf{kr}}) \le k$.*

Proof. First we prove that if $\mathsf{bpr}(G) \leq k$ then $\mathsf{bpr}(G_{\mathsf{kr}}) \leq k$. Let T be a biplanarizing set for G with $|T| = k$. Let $T' = f(T)$. We now prove that T' is a biplanarizing set for G_{kr}. Suppose it is not. By Lemma 1(c), $G_{\mathsf{kr}} \setminus T'$ contains a subgraph C that is either a cycle or a 2-claw. If C is a cycle, then $G \setminus T$ contains a cycle, which contradicts the fact that T is a biplanarizing set for G. If C is a 2-claw, it must be centred at a vertex $v \in V_3$. Let w_1, w_2, w_3 be the neighbours of v in C with corresponding leaves x_1, x_2, x_3. Then for all i, $1 \leq i \leq 3$, $f^{-1}(vw_i) \in E^*_{2,3}$, and f maps at least one edge $ab \in G \setminus T$ to $w_i x_i$ such that ab and $f^{-1}(vw_i)$ are incident to a common vertex. Thus $f^{-1}(C)$ contains a 2-claw in $G \setminus T$, which is a contradiction. Thus, T' is a biplanarizing set for G_{kr}, and since $|T'| \leq |T| = k$, $\mathsf{bpr}(G_{\mathsf{kr}}) \leq k$.

We now prove that if $\mathsf{bpr}(G_{\mathsf{kr}}) \leq k$ then $\mathsf{bpr}(G) \leq k$. Let T be a minimal biplanarizing set for G_{kr} with $|T| \leq k$. T can be transformed into a biplanarizing set contained in $f(E_{2,3})$ by replacing edges not in $f(E_{2,3})$ as follows. Any edge vw in T but not in $f(E_{2,3})$ is a leaf-edge of G_{kr} (say at v). Since T is minimal, $(G_{\mathsf{kr}} \setminus T) \cup \{vw\}$ is not biplanar; that is, $(G_{\mathsf{kr}} \setminus T) \cup \{vw\}$ contains a subgraph C that is either a cycle or 2-claw. Removing a leaf-edge from $G_{\mathsf{kr}} \setminus T$ cannot introduce a cycle; thus, C is a 2-claw and vw is a leaf-edge of C. Let vx be the other edge in C incident to v. Since x is the centre of a 2-claw, $\deg'_G(x) \geq 3$ and $vx \in f(E_{2,3})$. Every other edge vy incident to v must be in T, as otherwise $(C \setminus \{vw\}) \cup \{vy\}$ is a 2-claw in $G_{\mathsf{kr}} \setminus T$. Let $T' = (T \setminus \{vw\}) \cup \{vx\}$, as illustrated in Fig. 5. In $G_{\mathsf{kr}} \setminus T'$, vw is a connected component, and all other connected components are subgraphs of connected components of $G_{\mathsf{kr}} \setminus T$. Thus, T' is a biplanarizing set for G_{kr}. Replace T by T', and repeat this step until $T \subseteq f(E_{2,3})$.

(a) $G_{\mathsf{kr}} \setminus T'$ (b) $G_{\mathsf{kr}} \setminus T''$

Fig. 5. Replacing the edge vw in a biplanarizing set.

Since $T \subseteq f(E_{2,3})$, it is easily seen that $|f^{-1}(T)| \leq |T| \leq k$. A 2-claw or cycle in $G \setminus f^{-1}(T)$ implies a 2-claw or cycle in $G_{\mathsf{kr}} \setminus T$. Thus $f^{-1}(T)$ is a biplanarizing set for G, and $\mathsf{bpr}(G) \leq k$. ☐

The proof of Lemma 3 describes how to obtain a biplanarizing set for G from a biplanarizing set for G_{kr} in $O(|G|)$ time. To enable us to prove that $|G_{\mathsf{kr}}| \in O(k)$ (assuming $\mathsf{bpr}(G_{\mathsf{kr}}) \leq k$ and $\mathcal{S}_G = \emptyset$), we introduce the following potential function, whose definition is suggested by Lemma 1(d). For a graph $G = (V, E)$, define

$$\Phi_G(v) = \max\{\deg'_G(v) - 2, 0\}, \text{ and } \Phi(G) = \sum_{v \in V} \Phi(v) .$$

Intuitively, $\Phi(v)$ approximates the number of edges in the distance-2 neighbourhood of v that must be deleted for G to become biplanar. Graphs G with

$\Phi(G) = 0$ are of particular interest, as they provide another characterization of biplanar graphs.

Lemma 4. *For every graph G,*
(a) G is biplanar if and only if G is acyclic and $\Phi(G) = 0$, and
(b) if $\Phi(G) = 0$ then every component of G containing a cycle is a wreath.

Proof. $\Phi(G) = 0$ if and only if every vertex has non-leaf degree at most two. Thus, part (a) follows immediately from Lemma 1(d), and part (b) follows from the fact that a connected cyclic graph with maximum degree two is a cycle. □

For graphs G with $\Phi(G) = 0$, a minimum biplanarizing set for G consists of one cycle-edge from each component wreath. For graphs with $\Phi(G) > 0$ the observation in the following lemma will be useful.

Let the average non-leaf degree of vertices in V_3 be denoted by d.

Lemma 5. *Let G be a graph with $\Phi(G) > 0$ (that is, $V_3 \neq \emptyset$) then $|V_3| = \frac{\Phi(G)}{d-2}$.*

Proof. By definition, $d|V_3| = \sum_{v \in V_3} \deg'(v) = \sum_{v \in V_3} (\Phi_G(v) + 2) = \Phi(G) + 2|V_3|$. Thus, $(d-2)|V_3| = \Phi(G)$, and the result follows. □

We now prove that Φ provides a lower bound for $\mathsf{bpr}(G)$.

Lemma 6. *For every graph G, $\mathsf{bpr}(G) \geq \frac{1}{2}\Phi(G)$.*

Proof. The result follows from Lemma 4(a) if we prove that deleting one edge vw from G reduces $\Phi(G)$ by at most two. If at least one of v and w (say v) is a leaf, then $\Phi(v) = 0$ and $\Phi(w)$ does not change by deleting vw. If w becomes a leaf by deleting v, then w has one neighbour x for which Φ is reduced by one. If neither v nor w are leaves in G, then there are three cases for what can happen when edge vw is deleted.

Case 1. $\Phi(v)$ and $\Phi(w)$ both decrease: Then before deleting vw, $\deg'(v) \geq 3$ and $\deg'(w) \geq 3$. Thus, v and w do not become leaves by deleting vw, and Φ does not decrease for any other vertices.

Case 2. Exactly one of $\Phi(v)$ and $\Phi(w)$, say $\Phi(v)$, decreases: Then $\deg'(v) \geq 3$ and $\deg'(w) \leq 2$ before deleting vw. Thus, for at most one neighbour x ($\neq v$) of w is $\Phi(x)$ reduced, and $\Phi(x)$ is reduced by at most one. For no neighbour of v, except possibly x, is Φ reduced.

Case 3. Both $\Phi(v)$ and $\Phi(w)$ do not decrease: Thus, $\deg'(v) \leq 2$ and $\deg'(w) \leq 2$ before deleting vw. There is at most one neighbour of each of v and w for which Φ may decrease, and Φ may decrease by at most one for each neighbour (or by two if these neighbours coincide). □

Consider an instance (G, k) of the 2-Layer Planarization problem with $\mathcal{S}_G = \emptyset$. If $\Phi(G_{\mathsf{kr}}) > 2k$ then we can immediately conclude from Lemma 6 that $\mathsf{bpr}(G_{\mathsf{kr}}) > k$ and hence $\mathsf{bpr}(G) > k$. On the other hand, if $\Phi(G_{\mathsf{kr}}) \leq 2k$ then, as we now prove, the size of the kernel is bounded by a function solely of k.

Lemma 7. *For every graph G and integer k, if $\mathcal{S}_G = \emptyset$ and $\Phi(G_{\mathsf{kr}}) \leq 2k$ then the kernel has size $|G_{\mathsf{kr}}| \in O(k)$.*

Proof. By counting the edges in G_{kr} with respect to vertices in V_3 we have

$$|E_{\mathsf{kr}}| \leq \sum_{v \in V_3} (2 \deg'_{G_{\mathsf{kr}}}(v) + 1) = \sum_{v \in V_3} (2\Phi_{G_{\mathsf{kr}}}(v) + 5) = 2\Phi(G_{\mathsf{kr}}) + 5|V_3| .$$

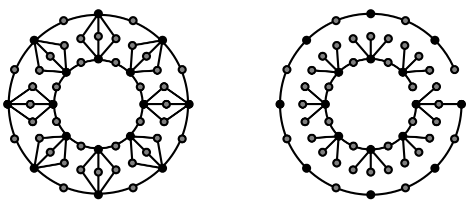
If $V_3 = \emptyset$ then $|E_{\mathsf{kr}}| \leq 2\Phi(G_{\mathsf{kr}})$. Otherwise, by Lemma 5 and since $d \geq 3$, $|V_3| \leq \Phi(G_{\mathsf{kr}})$ and $|E_{\mathsf{kr}}| \leq 7\Phi(G_{\mathsf{kr}})$. Since $\Phi(G_{\mathsf{kr}}) \leq 2k$, $|E_{\mathsf{kr}}| \leq 14k$, and since G_{kr} has no isolated vertices, $|G_{\mathsf{kr}}| \in O(k)$. □

One solution to the 2-Layer Planarization problem is to search through all subsets of size k in E_{kr}. We obtain an algorithm whose running time is $O(k \cdot \binom{14k}{k} + |G|)$. However, as asserted in the following lemma, one need only search through a subset $\mathcal{K}$ of the edges in G_{kr}. Let $\mathcal{K} = f(E_{2,3})$ except in the case of an internal 3-path, in which case $\mathcal{K}$ contains only the middle edge in this 3-path. The set $\mathcal{K}$ is called the *sub-kernel* of G_{kr}.

Lemma 8. *Let (G, k) be an instance of* 2-Layer Planarization *such that $\mathcal{S}_G = \emptyset$ and $0 < \Phi(G_{\mathsf{kr}}) \leq 2k$. If $\mathsf{bpr}(G_{\mathsf{kr}}) \leq k$ then there exists a biplanarizing set for G_{kr} with at most k edges contained in $\mathcal{K}$ and $|\mathcal{K}| \leq 2k(\frac{d}{d-2})$.* □

Due to space limitations, we have omitted the proof of this lemma, which is similar to that of Lemma 3.

Since $d \geq 3$, the size of the sub-kernel $|\mathcal{K}|$ is at most $6k$. There are a number of useful observations (omitted here) that may further reduce the size of the sub-kernel and improve the running time in practice. However, there is a pathological family of graphs for which our bound for the size of the sub-kernel is tight even with such improvements. Consider the graph $G_{p,q}$ $(p, q \in \mathbb{N})$ consisting of an *inner* cycle $(v_1, \ldots, v_{2p})$ and and *outer* cycle $(w_1, \ldots, w_{2p})$ with v_{2i} connected by q 2-paths to w_{2i} for all i, $1 \leq i \leq p$, as illustrated in Fig. 6(a). All vertices in V_3 have non-leaf degree $d = q + 2$. $G_{p,q}$ has $(d + 2)p$ vertices and $2dp$ edges. The sub-kernel of $G_{p,q}$ is the whole graph. The largest biplanar subgraph in $G_{p,q}$ is its spanning caterpillar (see Fig. 6(b)), which has $p(d + 2) - 1$ edges. Thus $\mathsf{bpr}(G_{p,q}) = 2dp - (p(d+2) - 1) = p(d-2) + 1$. The ratio of the number of edges in the sub-kernel of $G_{p,q}$ to $\mathsf{bpr}(G_{p,q})$ is $\frac{2dp}{p(d-2)+1} \to \frac{2d}{d-2}$ as $p \to \infty$. Thus the analysis of the size of the sub-kernel in Lemma 8 is tight for all d.



(a) (b)

Fig. 6. The graph $G_{8,3}$ and a spanning caterpillar of $G_{8,3}$.

We now present our algorithm for the 2-Layer Planarization problem based solely on kernelization.

Algorithm Kernelization (*input*: graph $G = (V, E)$; *parameter*: integer k)

determine $\mathcal{S}_G$ and let $k' = k - |\mathcal{S}_G|$;
determine the kernel $G_{\mathsf{kr}} = (V_{\mathsf{kr}}, E_{\mathsf{kr}})$ of $G \setminus \mathcal{S}_G$;
if $\Phi(G_{\mathsf{kr}}) > 2k'$ **then** return NO.
determine the sub-kernel $\mathcal{K} \subseteq E_{\mathsf{kr}}$ of G_{kr};
return YES iff $\exists T \subseteq \mathcal{K}$ s.t. $|T| \leq k'$, $G_{\mathsf{kr}} \setminus T$ is acyclic, and $\Phi(G_{\mathsf{kr}} \setminus T) = 0$;

Combining the previous lemmas and using Sterling's Formula gives:

Theorem 1. *Given a graph $G = (V, E)$ and integer k, the algorithm* Kernelization (G, k) *determines if* $\mathsf{bpr}(G) \leq k$. *If $\Phi(G) = 0$ then the running time is $O(|G|)$; otherwise it is $O(\sqrt{k} \cdot (\frac{2\mathbf{e}\, d}{d-2})^k + |G|)$ time, where d is the average non-leaf degree of vertices in V_3, and $\mathbf{e}$ is the base of the natural logarithm.* □

Clearly, if Kernelization(G, k) returns YES then a biplanarizing set for G with at most k edges can easily be computed. Since $d \geq 3$, in the worst-case the running time of algorithm Kernelization is $O(\sqrt{k} \cdot (6\mathbf{e})^k + |G|) \in O(\sqrt{k} \cdot 17^k + |G|)$.

4 Bounded Search Trees

A second approach to producing FPT algorithms is called the *method of bounded search trees* [2, Chapter 3.1]. Here one uses exhaustive search in a tree whose size is bounded by a function of the parameter. In this section we present an algorithm for the 2-Layer Planarization problem based on a bounded search tree approach. At each node of the search tree a 2-claw or small cycle C is identified. At least one of the edges in C is in every biplanarizing set. Our algorithm then recursively solves $|C|$ subproblems such that one of the edges in C is deleted from the graph in each subproblem. The following lemma provides a sufficient condition for the existence of such a set C.

Lemma 9. *If there exists a vertex v in a graph G_0 such that $\deg'_{G_0}(v) \geq 3$ then G_0 contains a 2-claw or a 3- or 4-cycle containing v.*

Proof. Let w_1, w_2, w_3 be three distinct non-leaf neighbours of v. If some w_i is adjacent to w_j then there is a 3-cycle containing v. Suppose no w_i is adjacent to a w_j. Let x_i be a neighbour of w_i such that $x_i \neq v$, $1 \leq i \leq 3$. Such an x_i exists since w_i is not a leaf. If $x_i = x_j$ then there is a 4-cycle containing v. Otherwise the x_i's are distinct. Then $\{v, w_1, w_2, w_3, x_1, x_2, x_3\}$ forms a 2-claw. □

We have the following algorithm.

Algorithm Bounded Search Tree (*input*: graph $G_0 = (V_0, E_0)$; *parameter*: integer k)

1 **if** $k = 0$ **then** return YES iff G_0 is acyclic and $\Phi(G_0) = 0$;
2 **else if** $\Phi(G_0) = 0$ **then** return YES iff $k \geq \#$ component wreaths of G_0;
3 **else** ($\exists v \in V_0$ such that $\deg'_{G_0}(v) \geq 3$)
   (a) find a 2-claw, 3-cycle or 4-cycle C in G_0 containing v as described in Lemma 9;
   (b) **for each** edge $xy \in C$ **do**
         if Bounded Search Tree($G_0 \setminus \{xy\}$, $k-1$) **then** return YES;
   (c) return NO;

We could solve 2-Layer Planarization by running Bounded Search Tree (G, k). However, we apply Bounded Search Tree to the kernel of G so that the running time at each node of the search tree is $O(k)$ rather than $O(|G|)$.

Theorem 2. *Given a graph G and integer k, let G_{kr} be the kernel of $G \setminus \mathcal{S}_G$. The algorithm* Bounded Search Tree $(G_{\mathsf{kr}}, k - |\mathcal{S}_G|)$ *determines if* $\mathsf{bpr}(G) \leq k$ *in* $O(k \cdot 6^k + |G|)$ *time.*

Proof. We prove the correctness of the algorithm by induction on k with the following inductive hypothesis: "Bounded Search Tree (G_0, k) returns YES if and only if $\mathsf{bpr}(G_0) \leq k$". The basis of the induction with $k = 0$ (Step 1) follows immediately from Lemma 4(a). Assume the inductive hypothesis holds for $k-1$. If $\Phi(G_0) = 0$ (as in Step 2) then by Lemma 4, every connected component is a caterpillar or a wreath. Caterpillars and wreaths have bipartite planarization numbers of 0 and 1, respectively. Thus $\mathsf{bpr}(G_0)$ is the number of component wreaths of G_0, and hence Step 2 of the algorithm is valid.

Now assume $k > 0$ and $\Phi(G_0) > 0$; that is, there exists a vertex $v \in V$ such that $\deg'_{G_0}(v) \geq 3$. By Lemma 9, G_0 contains a 2-claw or a 3- or 4-cycle C. Every biplanarizing set for G_0 must contain an edge in C. Thus $\mathsf{bpr}(G_0) \leq k$ if and only if there exists an edge $xy \in C$ such that $\mathsf{bpr}(G_0 \setminus \{xy\}) \leq k-1$. By induction, Bounded Search Tree $(G_0 \setminus \{xy\}, k-1)$ determines if $\mathsf{bpr}(G_0 \setminus \{xy\}) \leq k-1$. Therefore the algorithm determines if $\mathsf{bpr}(G_0) \leq k$, and the inductive hypothesis holds. In particular, Bounded Search Tree $(G_{\mathsf{kr}}, k - |\mathcal{S}_G|)$ determines if $\mathsf{bpr}(G_{\mathsf{kr}}) \leq k - |\mathcal{S}_G|$, which holds if and only if $\mathsf{bpr}(G) \leq k$ by Lemmas 2 and 3.

In each recursive call k is reduced by one. Thus the height of the search tree is at most k. At each node of the search tree, there are $|C|$ branches. Since $|C| \leq 6$, the search tree has at most 6^k nodes.

A non-recursive implementation maintains the current copy of G_0 under the operations of the deletion and insertion of edges. Moving from one copy to the next requires $O(|G_0|)$ time. At any given node of the search tree, the algorithm takes $O(|G_0|)$ time. Each G_0 is a subgraph of G_{kr}; thus, by Lemma 7, the time taken at each node of the search tree is $O(k)$. Therefore the running time of the algorithm is $O(k \cdot 6^k + |G|)$. $\square$

The exponential component of the time bound for Kernelization is $(\frac{2\mathrm{e}\,d}{d-2})^k$, and for Bounded Search Tree is 6^k. In the worst case with $d = 3$ the Kernelization bound is approximately 17^k, which is considerably more than 6^k. However, for $d \geq 22$, $\frac{2\mathrm{e}\,d}{d-2} < 6$, and the Kernelization algorithm provides a better upper bound on the running time than the Bounded Search Tree algorithm.

5 1-Layer Planarization

We now consider the 1-Layer Planarization problem: given a bipartite graph $G = (A, B; E)$ and permutation π of A, is $\mathsf{bpr}(G, \pi) \leq k$? It is not clear how one would apply a kernelization strategy to this problem. In this section, we apply a bounded search tree approach to the (unkernelized) graph G. The following result characterizes biplanar graphs with a fixed permutation of one bipartition.

Lemma 10. *There is a biplanar drawing of a bipartite graph $G = (A, B; E)$ with a permutation π of A iff G is acyclic and the following condition holds.*

For every path (x, v, y) of G with $x, y \in A$, if $u \in A$ is between x and y in π, then the only edge incident to u (if any) is uv. $(\star)$

Proof. The necessity of condition $(\star)$ is easily verified by observing that if $(\star)$ does not hold for some path (x, v, y) and edge uw $(w \neq v)$, then regardless of the relative positions of w and v in the permutation of B, uw must cross xv or yv, as illustrated in Fig. 7(a). This observation was also made by Mutzel and Weiskircher [14].

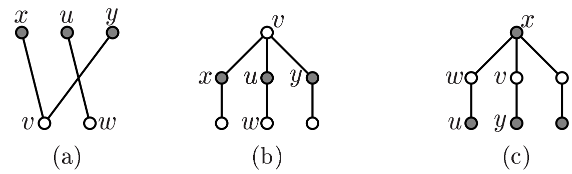


Fig. 7. Forbidden sub-structures for 1-layer biplanarity; vertices in A are gray.

Suppose condition $(\star)$ holds. Suppose for the sake of contradiction that G is not a forest of caterpillars. Since G is acyclic, by Lemma 1, G contains a 2-claw C. We consider two cases; in the first, C contains three vertices in A, as illustrated in Fig. 7(b). Let these vertices be x, u, y in the order they appear in π, and let v be the centre of C. Then condition $(\star)$ does not hold for the path (x, v, y) and the leaf-edge in C incident to u. Now consider the case in which C contains four vertices in A, as illustrated in Fig. 7(c). Let x be the centre of C. Then $x \in A$. Let u and y be two other vertices in $C \cap A$ both to the right or both to the left of x in π. Such a pair of vertices exists by symmetry. Without loss of generality, u is between x and y. Let v be the vertex in C adjacent to x and y, and let w be the vertex in C adjacent to x and u. Then condition $(\star)$ does not hold for the path (x, v, y) and edge uw. Thus G is a forest of caterpillars.

To construct a 2-layer drawing of G, all that remains is to describe the permutation of B. Let $(a_1, \ldots, a_n)$ be the ordering of A defined by π. For each vertex $v \in B$, define $L(v) = \min\{i : a_i v \in E\}$; that is, $L(v)$ is the leftmost neighbour of v in the fixed permutation of A. We say a vertex v with $L(v) = i$ *belongs* to a_i, as does the edge $a_i v$. Order the vertices $v \in B$ by the value of $L(v)$, breaking ties as follows. For each i, $1 \leq i \leq n$, there are at most two non-leaf vertices belonging to a_i, as otherwise there would be a 2-claw centred at a_i. Suppose there are exactly two non-leaf vertices v and w belonging to a_i. Then we can label v and w such that in π, the neighbours of v (not counting a_i) are all to the left of a_i, and the neighbours of w (not counting a_i) are all to the right of a_i, as otherwise $(\star)$ is not satisfied. Let $(v, x_1, \ldots, x_p, w)$ be the order of the vertices belonging to a_i, where $\{x_1, \ldots, x_p\}$ are the leaves belonging to a_i, as illustrated in Fig. 8(a). This defines a 2-layer drawing of G.

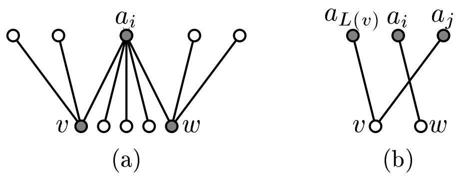


Fig. 8. Construction of the permutation of B.

Suppose there is a crossing between some edges $a_i w$ and $a_j v$ with $a_i, a_j \in A$ $(i < j)$ and $v, w \in B$. Then v is to the left of w in the permutation of B, and thus $L(v) \leq i$. Since the edges belonging to a_i do not cross, $L(v) < i$. This implies that condition $(\star)$ is not satisfied for the path $(a_{L(v)}, v, a_j)$ and the edge $a_i w$, as illustrated in Fig. 8(b). Thus there is no crossing in the drawing of G. $\square$

Lemma 11. *If $G = (A, B; E)$ is a bipartite graph and π is a permutation of A which satisfies condition $(\star)$, then all cycles of G are 4-cycles and every pair of non-edge-disjoint cycles shares exactly two edges. Moreover, the degree of all vertices in B which appear in a cycle is exactly two.* $\square$

It is not difficult to show that this lemma follows from Lemma 10. We omit details due to space limitations.

Let $G = (A, B; E)$ be a bipartite graph with a fixed permutation of A which satisfies condition $(\star)$. A complete bipartite subgraph $H = K_{2,p}$ of G with $|H \cap A| = 2$, $|H \cap B| = p$, and $\deg_G(v) = 2$ for every $v \in H \cap B$, is called a *p-diamond*, see Fig. 9. It follows from Lemma 11 that every cycle of G is in some diamond. We can now directly determine the 1-layer bipartite planarization numbers of graphs and permutations satisfying condition $(\star)$.

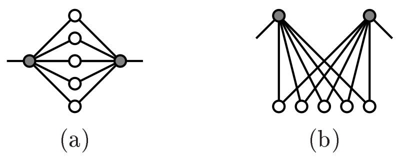


Fig. 9. (a) 5-diamond, (b) 2-layer drawing of a 5-diamond.

Lemma 12. *If $G = (A, B; E)$ is a bipartite graph and π is permutation of A satisfying condition $(\star)$ then* $\mathsf{bpr}(G, \pi) = \displaystyle\sum_{p\text{-diamonds of } G} (p - 1)$.

Proof. For each p-diamond H of G, delete $p - 1$ of the edges incident to one of the vertices in $H \cap A$. The resulting graph is acyclic and satisfies condition $(\star)$, and thus, by Lemma 10, has a biplanar drawing using π. To remove all cycles from G requires the deletion of at least $p - 1$ edges from each p-diamond. □

The following algorithm solves the 1-LAYER PLANARIZATION problem:

Algorithm 1-Layer Bounded Search Tree (*input*: set $F \subseteq E$; *parameter*: integer k)

1 **if** $k > 0$ and $(\star)$ fails for some path (x, v, y) and edge uw of $G[F]$ **then**
  (a) **for each** edge $e \in \{xv, yv, uw\}$ **do**
      if 1-Layer Bounded Search Tree $(F \setminus \{e\}, k - 1)$ **then**
        return YES;
  (b) return NO;
2 return YES iff $k \geq \displaystyle\sum_{p\text{-diamonds of } G[F]} (p - 1)$.

Theorem 3. *Given a bipartite graph $G = (A, B; E)$, fixed permutation π of A, and integer k, algorithm* 1-Layer Bounded Search Tree (E, k) *determines if* $\mathsf{bpr}(G, \pi) \leq k$ *in $O(3^k \cdot |G|)$ time.*

Proof. The correctness of the algorithm follows from Lemmas 10 and 12. We now describe how to check if condition $(\star)$ holds in $O(|A|)$ time. We can assume that the adjacency lists of vertices in B are ordered by π. For each vertex $v \in B$, by simultaneously traversing the adjacency list of v and a list of vertices in A ordered by π, we can identify whenever the condition $(\star)$ does not hold for some 2-path centred at v. Repeating this step for each vertex $v \in B$ or until an appropriate 2-path is found, we can check if $(\star)$ is satisfied. This algorithm runs in $O(|A|)$ time since if the section of the ordered list of A traversed with respect to some vertex $v \in B$ overlaps the corresponding section for some other vertex $w \in B$, then condition $(\star)$ is not satisfied, and the algorithm will immediately terminate. To count the number and size of the diamonds in $G[F]$ takes $O(|G|)$ time. Thus, the algorithm takes $O(|G|)$ time at each node of the search tree. Since each node of the search tree has three children, and the height of the tree is at most k, the algorithm runs in $O(3^k \cdot |G|)$ time. □

6 Conclusion

To the best of our knowledge, this paper and our companion paper [3] give the first applications of fixed-parameter tractability methods to graph drawing problems. Here, we study two problems, 1- and 2-LAYER PLANARIZATION, which are of fundamental interest in layered graph drawing. Our methods yield linear-time algorithms to determine if $\mathsf{bpr}(G) \leq k$ and $\mathsf{bpr}(G, \pi) \leq k$, for fixed k. When k is allowed to vary, the running time of the algorithms is, not surprisingly, exponential in k, but the base of the exponent is small. We believe that further study will demonstrate practical as well as theoretical benefit from this approach, and that these methods will prove more widely applicable.

References

1. G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs.* Prentice-Hall, 1999.
2. R. G. Downey and M. R. Fellows. *Parametrized complexity.* Springer, 1999.
3. V. Dujmović, M. Fellows, M. Hallett, M. Kitching, G. Liotta, C. McCartin, N. Nishimura, P. Ragde, F. Rosemand, M. Suderman, S. Whitesides, and D. R. Wood. On the parameterized complexity of layered graph drawing. In *Proc. 9th European Symp. on Algorithms (ESA '01)*, 2001. To appear.
4. P. Eades, B. D. McKay, and N. C. Wormald. On an edge crossing problem. In *Proc. 9th Australian Computer Science Conference*, pages 327–334. Australian National University, 1986.
5. P. Eades and S. Whitesides. Drawing graphs in two layers. *Theoret. Comput. Sci.*, 131(2):361–374, 1994.
6. P. Eades and N. C. Wormald. Edge crossings in drawings of bipartite graphs. *Algorithmica*, 11(4):379–403, 1994.
7. M. R. Garey and D. S. Johnson. Crossing number is NP-complete. *SIAM J. Algebraic Discrete Methods*, 4(3):312–316, 1983.
8. F. Harary and A. Schwenk. A new crossing number for bipartite graphs. *Utilitas Math.*, 1:203–209, 1972.
9. M. Jünger and P. Mutzel. 2-layer straightline crossing minimization: performance of exact and heuristic algorithms. *J. Graph Algorithms Appl.*, 1(1):1–25, 1997.
10. M. Kaufmann and D. Wagner, editors. *Drawing Graphs: Methods and Models*, volume 2025 of *Lecture Notes in Comput. Sci.* Springer, 2001.
11. T. Lengauer. *Combinatorial Algorithms for Integrated Circuit Layout.* Wiley, 1990.
12. P. Mutzel. An alternative method to crossing minimization on hierarchical graphs. In S. North, editor, *Proc. Graph Drawing (GD'96)*, volume 1190 of *Lecture Notes in Comput. Sci.*, pages 318–333. Springer, 1997. To appear in *SIAM J. Optimization.*
13. P. Mutzel. Optimization in leveled graphs. In P. M. Pardalos and C. A. Floudas, editors, *Encyclopedia of Optimization.* Kluwer, 2001. To appear.
14. P. Mutzel and R. Weiskircher. Two-layer planarization in graph drawing. In K. Y. Chwa and O. H. Ibarra, editors, *Proc. 9th International Symp. on Algorithms and Computation (ISAAC'98)*, volume 1533 of *Lecture Notes in Comput. Sci.*, pages 69–78. Springer, 1998.
15. F. Shahrokhi, O. Sýkora, L. A. Székely, and I. Vrťo. On bipartite drawings and the linear arrangement problem. *SIAM J. Comput.*, 30(6):1773–1789, 2001.
16. K. Sugiyama, S. Tagawa, and M. Toda. Methods for visual understanding of hierarchical system structures. *Trans. Systems Man Cybernet.*, 11(2):109–125, 1981.
17. N. Tomii, Y. Kambayashi, and S. Yajima. On planarization algorithms of 2-level graphs. *Papers of tech. group on elect. comp., IECEJ*, EC77-38:1–12, 1977.
18. V. Valls, R. Marti, and P. Lino. A branch and bound algorithm for minimizing the number of crossing arcs in bipartite graphs. *J. Operat. Res.*, 90:303–319, 1996.
19. M. S. Waterman and J. R. Griggs. Interval graphs and maps of DNA. *Bull. Math. Biol.*, 48(2):189–195, 1986.