Kernels in planar digraphs

Gregory Gutin, Ton Kloks[1⋆], and C. M. Lee[2⋆⋆]

[1] Department of Computer Science
Royal Holloway University of London
Egham, Surrey TW20 OEX, UK
gutin,ton@cs.rhul.ac.uk

[2] Department of Computer Science and Information Engineering
Chung Cheng University
Ming-Shiun, Chaiyi 621, Taiwan
cmlee@cs.ccu.edu.tw

Abstract. A set S of vertices in a digraph $D = (V, A)$ is a kernel if S is independent and every vertex in $V - S$ has an out-neighbour in S. We show that there exists an $O(3^{\delta\sqrt{k}}n)$ [1] algorithm to check if a planar digraph has a kernel with at most k vertices. We found a reduction which allows us to separate the sub-exponential part additively from the digraph size. Our result implies an algorithm that checks for a kernel with at most k vertices in time $O(3^{\delta'\sqrt{k}} + n)$ for any constant $\delta' > \delta$. For plane digraphs we introduce a new parameter which we call the kernel number. For a plane digraph the kernel number is the minimum number of faces to be deleted (i.e., the vertices of the boundary) such that the remaining digraph has a kernel. We show that determining whether the kernel number is at most some constant k is NP-complete for every $k \geq 0$.

Revelation

After decades of silken repose, being told beddy-bye by means of the poisoned apple called NP-completeness, the princess of computer science has finally been kissed awake. At tremendous speed choking barricades are torn down and a colourful palette of new optiques has emerged. The visionary Don Quichote, christened *fixed parameter complexity*, has cast out the apocalyptic intractable messenger which was a mere catafalque for efficient algorithms.

Definition 1. *Consider an algorithm for a parameterised problem (I, k), where I is the problem instance and k the parameter. The algorithm is called* uniformly polynomial *if it runs in time $O(f(k)|I|^c)$, where $|I|$ is the size of I, $f(k)$ an arbitrary function, and c a constant independent of k. A parameterised problem is* fixed-parameter tractable (FPT) *if it admits a uniformly polynomial algorithm.*

⋆ Supported by an EPSRC grant.
⋆⋆ Corresponding author.
[1] Throughout this paper the constants δ and c are the same as the comparative constants mentioned in [1].

Parameterised complexity was introduced some ten years ago [22]. There are by now numerous algorithms known for fixed parameter tractable problems. For PLANAR DOMINATION, part of the stumbling history is reflected in [24, 22, 2]. A stunning breakthrough with a catalytic impact was made with the discovery of sub-exponential solutions (i.e., $O(c^{\sqrt{k}}n)$) for various domination problems on planar graphs [13, 1, 3, 4, 19].
It is shown [17] that for all MAX SNP-hard problems (such as dominating set) finding exact solutions in sub-exponential time is not possible unless $W[1] = FPT$. PLANAR DOMINATING SET (and various other problems) has no EPTAS running in time $O(2^{o(\sqrt{\frac{1}{\epsilon}})}p(n))$ unless $W[1] = FPT$ [18]. These results indicate that the square-root solutions for planar domination are optimal.
It is well-known that if a problem is fixed parameter tractable then it also yields an *additive* FPT solution (see Section 2). These algorithms are invariably obtained by reduction to a *problem kernel*[2] [22]. The swift progress in this area is nicely illustrated by improvements on the solutions for the VERTEX COVER PROBLEM initiated by S. Buss [22, 6, 25, 40, 20]. Other examples are k-LEAF SPANNING TREE [22] and 2-LAYER PLANARIZATION [26]. The first *sub-linear additive* FPT algorithm appeared recently in [19] for PLANAR MAXIMUM CLIQUE TRANSVERSAL. In this paper we show that also PLANAR INDEPENDENT DOMINATION yields a sub-linear kernelization.

We use the standard terminology and notation on digraphs, see [7, 31]. We consider digraphs without loops or parallel arcs. Thus, a digraph $D = (V, A)$ is a pair consisting of vertex set V and arc set A, a set of ordered pairs of vertices. We denote an arc a as (u, v) where u is the *tail* and v is the *head* of a; v is an *out-neighbour* of u and u is an *in-neighbour* of v; u and v are *adjacent*. The set of out-neighbours of a vertex x is denoted by $N^+(x)$ and the set of in-neighbours by $N^-(x)$. The set of *neighbours* is defined as $N(x) = N^+(x) \cup N^-(x)$. The *in-degree* (*out-degree*) of x is $d^-(x) = |N^-(x)|$ ($d^+(x) = |N^+(x)|$); the *degree* of x is $d(x) = d^+(x) + d^-(x)$.
The *underlying graph* $G = (W, E)$ of a digraph $D = (V, A)$ is an undirected graph (with no parallel edges) that has the same vertex set as D, i.e., $W = V$ and two vertices are adjacent in G if and only if they are adjacent in D. A *super-orientation* of an undirected graph G is a digraph D such that G is a underlying graph of D.
Kernels in digraphs were introduced in different ways in [10, 37, 39]. It seems that Neumann [39] was the first to introduce kernels when describing winning positions in 2-person games. For important applications of kernels in game theory see [27, 28, 38]. Applications of kernels are widespread and appear in diverse fields as logic, computational complexity, graph theory, combinatorics, coding theory and other areas.

[2] In this paper we use two standard similarly sounding terms: a problem kernel for special reductions and a kernel in a digraph. The terms have different meaning and the reader will be able to see which one is used from the context.

Definition 2. *Let $D = (V, A)$ be a digraph. A set S of vertices is a* kernel *if S is independent and every vertex in $V - S$ has an out-neighbour in S.*

Chvátal proved (see [30]) that the problem of deciding whether a digraph has a kernel (called the *kernel problem* below) is NP-complete. Fraenkel [27] showed that the kernel problem remains NP-complete even for planar digraphs D with degree constraints $d^+(x) \leq 2$, $d^-(x) \leq 2$ and $d(x) \leq 3$ for all vertices x. It is easy to see that every acyclic digraph has a kernel. This sufficient condition for a digraph to have a kernel has been generalised by several authors. For short accounts on the topic see [7, 11]. In passing we would mention a result by Boros and Gurvich [14], generalised by Cai and Corneil [15], that every super-orientation of a perfect graph is kernel solvable (a graph G is *kernel solvable* if for every super-orientation D of G for which every clique has a kernel, D itself has a kernel). Berge and Duchet [10, 11] conjectured that the opposite is also true, i.e., every kernel solvable graph is perfect. From results in [29] and [34] it follows that every distance hereditary digraph with vertex-clique covering number at most 3 has a kernel.
Finding kernels in special classes of digraphs seems to be a big open field. It has been shown [8] that the kernel problem is polynomial time solvable for *locally semicomplete digraphs*, digraphs in which the out-neighbours (in-neighbours) of every vertex are adjacent.
In this paper we prove that there are polynomial algorithms to determine the existence of small kernels (of size $O(log^2 n)$) and large kernels (of size $\Omega(n - log^2 n)$) in planar digraphs.

We introduce a new concept, the kernel number of a plane digraph. For a plane digraph the kernel number is the minimum number of faces to be deleted (i.e., the vertices of the boundary) such that the remaining digraph has a kernel. We show that deciding if the kernel number is at most k is NP-complete for plane digraphs even when restricted to plane digraphs with in– and out-degree at most 2 and degree at most 3.

1 Ways to locate kernels in planar digraphs

Definition 3. *A* tree decomposition *$(T, \mathcal{S})$ for an undirected graph $G = (V, E)$ is a pair where T is a tree and $\mathcal{S}$ is a set of subsets of vertices, called* bags. *$\mathcal{S}$ is in 1-1 correspondence with the vertices of the tree T such that the following conditions are satisfied:*

1. *Every vertex is contained in at least one bag,*
2. *Both endpoints of every edge are contained in at least one bag,*
3. *For every vertex x of the graph, if x appears in bags S_i and S_j then it appears in every bag corresponding to the vertices which lie on the path in the tree T between the points i and j.*

The *width* of a tree decomposition $(T, \mathcal{S})$ is the maximum cardinality of a bag S_i minus one, over all nodes in the tree T. The *treewidth* of the graph is the minimum width over all possible tree decompositions of the graph.

Computing treewidth for graphs in general is NP-complete, however the problem is FPT. Moreover there exists a linear time algorithm to check if a graph has bounded treewidth [12, 36].

The following result [13, 1, 3], is of eminent importance to us.

Theorem 1. *If a planar graph has a dominating set of size at most k then the treewidth is bounded by $\delta\sqrt{k}$ for some constant δ. There is an $O(c^{\sqrt{k}}n)$ algorithm that finds a dominating set of cardinality at most k if it exists. Here c is some constant.*

Remark 1. In [1, 3] upper bounds $c \leq 3^{6\sqrt{34}}$ and $\delta \leq 6\sqrt{34}$ are communicated.

We sketch an algorithm to find a kernel of size k in a planar digraph in time at most $O(c^{\sqrt{k}}n)$ if it exists. If such a kernel does not exist, our algorithm will accordingly report this. Our algorithm is based on the following simple observation.

Lemma 1. *If a planar digraph D has a kernel of size at most k, then its underlying graph G has a dominating set of cardinality at most k. Hence the treewidth of the underlying graph is at most $\delta\sqrt{k}$.*

In the rest of this section we assume that we are given a tree decomposition of width at most $O(\sqrt{k})$ for the underlying graph G of a digraph $D = (V, A)$ for which we want to find a kernel of size at most k. (If such a tree decomposition does not exist, then clearly, the digraph cannot have a kernel of this size.)

Remark 2. It is worth mentioning that only the treewidth of the underlying graph is important to us in this paper. *Directed treewidth* is defined in [35] (see also [41, 43]). It is easy to see that if a digraph has bounded directed treewidth, and *a tree-decomposition of bounded width is given* then the kernel problem can be solved also using this directed tree-decomposition. However, the theory of directed treewidth is still immature.

Our algorithm is a simple dynamic programming procedure which computes candidates for a kernel for each node in the tree, starting with the leaves. The following notions are useful for describing the algorithm. We denote by Y_i the union of the bags S_j of the subtree of T with root node i.

Definition 4. *An* almost-kernel *for node i is an independent set K in $D[Y_i]$ such that all vertices of Y_i, except possibly some vertices in S_i, are either contained in K or have an out-neighbour in K. A* characteristic *for node i is a pair (f, ϵ), where f is a 3-colouring of the vertices S_i and ϵ is some integer $1 \leq \epsilon \leq k$. A vertex is coloured* red *if it is an element of the kernel. It is coloured black if it is not in the kernel but has at least one out-neighbour in the kernel, and it is coloured grey if it is not in the kernel and if it has no out-neighbour in the kernel. For a 3-colouring f, ϵ is the* minimum cardinality *of an almost kernel K for node i such that the grey vertices of f are exactly those vertices without an out-neighbour in K.*

The main part of the algorithm computes the table of all characteristics for every node i in T from the tables of its children. If we reach the root r of the tree we can read from the table at this root whether there exists a kernel with size ϵ if there exists a characteristic (f, ϵ) in the table at the root without grey vertices. Notice that the number of nodes in T is $O(n)$ and the work done for each bag is $O(3^{\delta\sqrt{k}})$ (the number of 3-colourings of S_i). We obtain the following:

Theorem 2. *If a digraph $D = (V, A)$ has a kernel of size at most k, then a tree decomposition of width $\delta\sqrt{k}$ can be found in linear time. Hence, there is an algorithm running in $O(3^{\delta\sqrt{k}}n)$ time to decide whether D has a kernel of size at most k.*

Remark 3. It would be interesting (since we are dealing with planar digraphs [33, 42]) to see whether a kernel algorithm can be described just as easily in terms of a branch-decomposition.

In Section 2 we need an extension of this result. It is readily checked that the following theorem holds as well:

Theorem 3. *Let $D = (V, A)$ be a digraph and let R and B be disjoint subsets of V. An (R, B)-kernel is a kernel K with $R \subseteq K$ and $B \cap K = \emptyset$. There is an algorithm running in time $O(3^{\delta\sqrt{k}}n)$ which decides whether a planar digraph $D = (V, A)$ with disjoint sets $R, B \subseteq V$ has an (R, B)-kernel of cardinality at most k.*

For a connected digraph $D = (V, A)$ and kernel K in D, $V - K$ is a dominating set in the underlying graph of D. This observation implies the following:

Corollary 1. *If a connected digraph D has a kernel of order at least $n - k$, then there exists an $O(3^{\delta\sqrt{k}}n)$ algorithm to compute a kernel.*

Remark 4. The kernel spotted by this algorithm need not have cardinality $n - k$. This depends on its implementation.

2 Simplifier to a problem kernel

We start by a short description of a general algorithm to obtain an additive FPT algorithm from a multiplicative one (see [16, 23]).

Lemma 2. *Assume we have an algorithm which runs in time $O(f(k)n)$ for a parameterised problem (on digraphs) with parameter k for some function $f(k)$. Then, alternatively, we can obtain an algorithm which runs in time $O(n^2 + h(k))$ for some function $h(k)$.*

Proof. First compute a table T of solutions for all digraphs with at most $f(k)$ vertices. (This table is referred to as the "advice" in [16]). Now consider a digraph with n vertices. If $n \leq f(k)$ the solution can be looked up in T. On the other hand, if $n > f(k)$, our multiplicative algorithm runs in $O(f(k)n) = O(n^2)$ time. □

Remark 5. Since our digraphs have no parallel arcs or loops, we have an upper bound of $2^{c^{2\sqrt{k}}}$ for the number of digraphs with $c^{\sqrt{k}}$ vertices. Hence, direct application of the proof of Lemma 2 gives an (impractical) algorithm with running time at least $\Omega\left(n^2 + 2^{c^{2\sqrt{k}}}\right)$. Notice also that we can obtain an algorithm that runs in time $O(n^\alpha + h'(k))$ for *any* $\alpha > 1$ at the cost of a blow-up of the function h.

In the rest of this section we describe an algorithm which is only singly exponential in $\sqrt{k}$ and linear in n. Let D be a planar digraph. Assuming that D has a kernel of cardinality at most k we show first that we can reduce D to a digraph D' of size polynomial in k, and a set $\mathcal{S}$ of subsets of vertices of D'.
We colour some of the vertices red in the process. We colour the vertices of D' red if they *must* be contained in any kernel of cardinality at most k of D' for a reason described below. All red vertices remain vertices of D'. Some vertices are removed from D', and other vertices remain uncoloured. During the process of constructing D' we keep the following condition invariant:

> D has a kernel K of size at most k if and only if K is a kernel in D' containing all red vertices and such that for every subset $S \in \mathcal{S}$ at least one element is in K.

Initially $D' = D$, all vertices are uncoloured and $\mathcal{S} = \emptyset$. Clearly the invariant is valid in this initial stage. We prove that the final size of D' is $O(k^3)$.

Lemma 3. *If two vertices x and y of D have at least $3k+2$ common neighbours, then at least one of x and y must be in a kernel of cardinality at most k. Hence none of the common neighbours can be in a kernel of cardinality at most k.*

Proof. Let K be a kernel of cardinality at most k. Since D is planar, if neither x nor y is in K, then the common neighbourhood C together with x and y determine at least $3k + 1$ finite faces. Each $z \in C$ must be in K or must have an outgoing arc into K. Since a vertex of K can be an out-neighbour of boundary vertices of at most two faces, we see that at least $k + 1$ vertices are needed to give every $z \in C$ which is not in K an out-neighbour in the kernel. This means K has more than k vertices, a contradiction. □

Lemma 4. *Assume $k \geq 1$. The number of pairs of vertices $\{x, y\}$ which have at least $3k + 2$ common neighbours is $O(n)$.*

Proof. Every plane embedding of $K_{2,3}$ has a vertex which is not on the outerface (an outerplanar graph can not have a $K_{2,3}$ subgraph).
Given a vertex c, there is at most one pair $\{x, y\}$ such that c is the middle vertex of a plane $K_{2,3}$ with x and y of degree 3. □

Our algorithm consists of three stages. We describe each stage in detail. The **first** stage in our construction of D' is to remove the set C of common neighbours of every pair of vertices x and y with $|C| \geq 3k + 2$ from the current D'. For

every removed vertex z we make a subset S_z of its out-neighbours that remain in D' and we remove z from all existing subsets thus far. (A vertex z occurs in a subset S_u only if z is an out-neighbour of u and if u is no vertex of D'.) Notice that the invariant remains valid after completion of the first stage.
Henceafter assume that the vertices of every pair in D have at most $3k+1$ neighbours in common. The next lemma shows that vertices of large degree must belong to the kernel.

Lemma 5. *If $d(x) \geq 3k^2 + 2k + 1$ then x must be in any kernel of cardinality at most k.*

Proof. Consider a kernel K with at most k vertices. Assume $d(x) \geq 3k^2 + 2k + 1$ and assume $x \notin K$. Every neighbour of x must have at least one outgoing arc in K or must be in K itself.
By assumption every element of K has at most $3k+1$ neighbours in common with x. Since x has at most k neighbours in K, it has at least $3k^2 + k + 1$ neighbours not in K. Each of these $3k^2 + k + 1$ neighbours must have an outgoing arc into K. But every vertex in K has at most $3k+1$ common neighbours with x. Hence we would have $3k^2 + k + 1 \leq k(3k+1)$ which is a contradiction. □

This lemma leads to the validity of the **second** stage in our construction of D': In this step we colour every vertex x red if its degree is at least $3k^2 + 2k + 1$. The vertices of $N^-(x)$ are removed from the digraph D' (they have a neighbour in the kernel), and the vertices of $N(x) - N^-(x)$ are removed and their out-neighbourhoods are made subsets. Notice that the invariant remains valid.
This second step completes the construction of D'.

Lemma 6. *If D' has a kernel with at most k vertices then the total number n_1 of vertices in D' is at most $k(3k^2 + 2k + 1)$.*

Proof. Let K be a kernel in D with at most k vertices. Let $x \in K$. If x is red, then it has no neighbours in D'. If x is not red then its degree is at most $3k^2 + 2k$ by the definition of a red vertex. Finally, every vertex in D' must be a neighbour of a vertex in K. It follows that the number of vertices in D' is at most $k(3k^2 + 2k + 1)$. □

A careful analysis, which we have to omit due to space limitations, shows that the collection $\mathcal{S}$ of subsets can be constructed in linear time. Notice that a subset is constructed as a list of out-neighbours of a vertex only once, and a vertex is removed from subsets in $\mathcal{S}$ only once. Furthermore, a vertex p is removed from a subset S_q only if p is an out-neighbour of q. It follows that there exists a simple data structure representing the subsets and elements in it, such that the total time needed for creating the subsets and removing vertices from the subsets is linear.
In the **third** step of our algorithm we delete multiple copies of the same subset. Using the fact that we are dealing with a planar graph it follows that we can construct a *set* $\mathcal{S}$ of subsets (without repeated elements) in linear time. Using a radix sort without initialisation we can obtain the following [21]:

Lemma 7. *Call two subsets of $\mathcal{S}$ equivalent if they contain the same elements. The equivalence classes can be determined in $O(n)$ time.*

Notice that also in this final stage the invariant is valid. We next show that the number of subsets is at most a polynomial in k.

Lemma 8. *There exist at most $4n_1^3$ different subsets in $\mathcal{S}$.*

Proof. Mark every subset S of $\mathcal{S}$ by a set T of cardinality 3 such that $T \subseteq S$ if $|S| \geq 3$ and by S itself, otherwise. Observe that, since the underlying graph of D does not contain $K_{3,3}$, not more than two subsets with mark of cardinality 3 have the same mark. Therefore, there are not more than $3n_1^3$ marks and thus not more than $4n_1^3$ subsets in $\mathcal{S}$. □

Combining the result of Lemma 8 with that of Lemma 6 we obtain:

Theorem 4. *There exists an algorithm that runs in time $O(3^{\delta\sqrt{k}}k^{\tau} + n)$, where $\tau \leq 9$, to decide whether a planar digraph $D = (V, A)$ has a kernel of size at most k.*

Proof. Construct a planar digraph F from D' and $\mathcal{S}$ as follows. The vertex set of F equals the vertex set of D' extended with one new vertex for each subset. Each such a vertex is coloured *black*. Its out-neighbourhood equals the subset, and its in-neighbourhood is $\emptyset$.
Let B be this set of black vertices and let R be the set of red vertices in D'. Clearly, we have reduced the problem of existence of kernel of cardinality at most k in D to the problem of finding a (R, B)-kernel of size of most k in F. By Theorem 3 there exist an algorithm for the last problem that runs in time $O(3^{\delta\sqrt{k}}\kappa^3)$ where $\kappa = 3k^3 + 2k^2 + k$. Hence the upper bound on τ. □

Remark 6. Clearly, we can just as well write $O(3^{\delta'\sqrt{k}} + n)$ for this time bound for any constant $\delta' > \delta$.

Notice that our proof holds as well for independent dominating sets in planar graphs. Hence we obtain the following:

Theorem 5. *There exists an algorithm that runs in time $O(3^{\delta\sqrt{k}}k^{\tau} + n)$ to determine if a planar graph has an independent dominating set of cardinality at most k.*

Conjecture 1. We conjecture that a similar result holds for DOMINATING SET.

Conjecture 2. We conjecture that the upper bound for τ in Theorem 4 can be improved.

3 Kernel numbers

The question whether a planar digraph has a kernel is a parameter-free question. We consider the cardinality as the prior choice for parameterisation. In this section we investigate a second intrinsic option.

Definition 5. *Let D be a plane digraph. The* kernel number *of D is the minimum number of faces to be deleted such that the remaining digraph has a kernel.*

Remark 7. Notice the analogy with the k-FACE COVER PROBLEM. This latter problem is FPT [1]. Indeed, if the vertices of a plane digraph D can be covered with at most k faces then clearly, the kernel number of D is at most k. We shall see that the FPT result for k-FACE COVER is unlikely to hold as well for the kernel number.

Correctness of the following lemma is readily checked.

Lemma 9. *The kernel number of a plane digraph is equal to the minimum number of new vertices we have to put inside faces with arcs to and from the border vertices of these faces such that the resulting plane digraph has a kernel.*

Proof. Let there be k faces whose removal results in a digraph with a kernel. Add a new vertex inside each of these k faces and make it incident with arcs to and from all vertices on the border of the face. Then this new digraph has a kernel.
Assume that we can add a new vertex inside k faces and obtain a kernel K. We may assume that k is minimal, i.e., all the new vertices are in the kernel. Hence the border vertices of these faces are not in the kernel. This implies that if we delete these k faces we get a graph with a kernel. □

Remark 8. Computing the kernel number can be done in linear time for digraphs whose underlying graphs are of bounded treewidth since the problem can be formulated in second monadic order logic (for recent maturation see [32]).

Theorem 6. *Let k be a fixed non-negative integer. Unless P=NP, there is no polynomial time algorithm to find out if the kernel number of a plane digraph equals k.*

Proof. Let D be a plane digraph. Consider an auxiliary digraph D' constructed from D as follows. Take k concentric directed triangles around D and k concentric directed triangles beside it. Observe that if we remove the infinite face of D', then the kernel number of D' will decrease by 1. Now it is easy to see that the kernel number of D' equals k if and only if D has a kernel. □

From [27] and the construction given in the proof of Theorem 6 we have the following:

Corollary 2. *The result of Theorem 6 remains valid for plane digraphs with in– and out-degree at most 2 and degree at most 3.*

Remark 9. Theorem 6 implies that it is unlikely that we can find any *efficient* PTAS for the kernel number problem (unless FPT=W[1]) [22]. So it is unlikely that, for example, Baker's method [5] can be made functional.

4 Comments

In this paper we have shown that the PLANAR KERNEL problem is FPT. We have shown that there exists a *additive sublinear* FPT algorithm to solve this problem. As an immediate consequence of our proof method we obtained that also PLANAR INDEPENDENT DOMINATION allows an additive sublinear FPT solution. Because of the wide applicability it would be of great interest to obtain a similar result for PLANAR DOMINATING SET. In view of the recent developments we feel quite sure that the finding of such an algorithm is a matter of time. The current state of research indicates that such a result would be best possible.

Thank you kindly

Hans Bodlaender, Maw-Shang Chang, and Annelies Jacobs. These people are credit to this paper.

References

1. Alber, J., H. Bodlaender, H. Fernau, and R. Niedermeier, Faster algorithms for domination of planar graphs, *Proc. of the* 17^{th} *SWAT'00*, Springer–Verlag LNCS 1851 (2000), pp. 97–110.
2. Alber, J., H. Fan, M. Fellows, H. Fernau, R. Niedermeier, F. Rosamond, U. Stege, Refined search tree techniques for the planar dominating set problem, Proceedings 26^{th} MFCS 2001.
3. Alber, Jochen, Henning Fernau, and Rolf Niedermeier, Parameterized complexity: Exponential speed-up for planar graph problems, Proceedings 28^{th} ICALP, 2001.
4. Alber, J., H. Fernau, R. Niedermeier, Graph separators: A parameterized view, Proceedings 7^{th} COCOON 2001.
5. Baker, B. S., Approximation algorithms for NP-complete problems on planar graphs, *J. ACM* **41**, (1994), pp. 153–180.
6. Balasubramanian R., M. R. Fellows, and V. Raman, An improved fixed parameter algorithm for vertex cover, *Information Processing Letters* **65**, (1998), pp. 163–168.
7. Bang-Jensen, Jorgen and Gregory Gutin, *Digraphs: Theory, Algorithms and Applications*, Springer, London, 2000.
8. Bang-Jensen, J., Y. Guo, G. Gutin, and L. Volkmann, A classification of locally semicomplete digraphs, *Discrete mathematics* **167/168**, (1997), pp. 101–114.
9. Bazgan, C., Schémas d'approximation et complexité paramt́rée. Thesis, INRIA, Orsay, France, 1995.
10. Berge, C., and P. Duchet, Strongly perfect graphs, *Ann. Discrete Math.* **21**, (1984), pp. 57–61.
11. Berge, C., and P. Duchet, Recent problems and results about kernels in directed graphs, *Discrete Math.* **86**, (1990), pp. 27–31.
12. Bodlaender, H., A linear time algorithm for finding tree decompositions of small treewidth, *SIAM J. Comput.* **25**, (1996), pp. 1305–1317.
13. Bodlaender, Hans, L., Ton Kloks, Jochen Alber, Henning Fernau, and Rolf Niedermeier, Faster algorithms for planar dominating set and related problems, 40. Workshop Komplexitätstheorie, Datenstrukturen und Effiziente Algorithmen, Technische Universität Ilmenau, 2000.

14. Boros, E., A. V. Gurvich, perfect graphs are kernel solvable, *Discrete math.* **159**, (1996), pp. 35–55.
15. Cai, Leizhen, and Derek G. Corneil, A generalisation of perfect graphs–i-perfect graphs, *J. Graph Theory* **23**, (1996), pp. 87–103.
16. Cai, Liming, Jianer Chen, Rod Downey, and Mike Fellows, Advice classes of parameterised tractability, *Annals of Pure and Applied Logic* **84**, (1997), pp. 119–138.
17. Cai, Liming and D. W. Juedes, Subexponential parameterized algorithms collapse the W-hierarchy, Proceedings 28^{th} ICALP, 2001.
18. Cai, Liming, Michael Fellows, David Juedes, and Frances Rosamond, Efficient polynomial-time approximation schemes for problems on planar structures: upper and lowerbounds. Manuscript 2001.
19. Chang M.-S, T. Kloks, and C. M. Lee, Maximum clique transversals, Proceedings 27^{th} WG 2001.
20. Chen, J., I. A. Kanj, and W. Jia, Vertex cover: Further observations and further improvements, *Proceedings 25^{th} WG* Springer-Verlag LNCS 1665, (1999), pp. 313–324.
21. T.H. Cormen, C.E. Leiserson, and R.L. Rivest, Introduction to algorithms, MIT press, Cambridge MA, 1990.
22. Downey, Rod G., and Mike R. Fellows, em Parameterised complexity, Monographs in Computer Science, Springer–Verlag, 1999.
23. Downey, Rod G., Michael R. Fellows, and Ulrike Stege, Parameterised complexity: A framework for systematically confronting computational intractability, *Contemporary trends in discrete mathematics*, (R. Graham, J. Kratochvíl, J. Nešetřil, and F. Roberts, eds.), Proceedings of the DIMACS–DIMATIA Workshop, Prague, 1997, AMS–DIMACS *Series in Discrete Mathematics and Theoretical Computer Science* **49**, (1999), pp. 49–99.
24. Downey, R. G. and M. R. Fellows, Parameterized computational feasibility, *Proceedings of feasible mathematics II* (eds. P. Clote and J. B. Remmel), Birkhauser (1995), pp. 219–244.
25. Downey, R. G., M. R. Fellows, U. Stege, Parameterized complexity: A framework for systematically confronting computational intractability, *Contemporary trends in discrete mathematics: From DIMACS to DIMATIA to the future* Volume 49 of *AMS-DIMACS proceedings Series*, pp. 49–99. AMS, 1999.
26. Dujmović, Vida, Michael Fellows, Michael Hallett, Matthew Kitching, Giuseppe Liotta, Catherine McCartin, Naomi Nishimura, Prabhakar Ragde, Fran Rosamond, Matthew Suderman, Sue Whitesides, and David R. Wood, A fixed parameter approach to two-layer planarization. Manuscript 2001.
27. Fraenkel, Aviezri S. Planar kernel and Grundy with $d \leq 3$, $d^+ \leq 2$, $d^- \leq 2$ are NP-complete, *Discrete Appl. Math.* **3**, (1981), pp. 257–262.
28. Fraenkel, Aviezri, S., Combinatorial game theory foundations applied to digraph kernels, *Elec. J. Combin.* **4**, (1997), approx. 17 pages.
29. Galeana-Sánchez, H. and Xueliang Li, Kernels in a special class of digraphs, *Discrete Math.* **178**, (1998), pp. 73–80.
30. Garey, M. R., and D. S. Johnson, *Computers and intractability*. W. H. Freeman and Co., San Francisco, 1979.
31. Gondran, M., and M. Minoux, *Graphs and algorithms*, (Wiley, Chistester, UK).
32. Grohe, Martin, Generalized model-checking problems for first order logic, Proceedings 18^{th} STACS 2001.
33. Hicks, Ilya V., Planar branch decompositions. Manuscript 2001.

34. Howorka, E., A characterisation of distance-hereditary graph, *Quart. J. Math. Oxford, ser. 2* **28**, (1977), pp. 417–420.
35. Johnson, Thor, Neil Robertson, P. D. Seymour, and Robin Thomas, Directed treewidth. Manuscript 2001.
36. Kloks, T., *Treewidth–Computations and approximations*, Springer–Verlag, LNCS, 842, 1994.
37. Kwaśnik, M, The generalisation of Richardson theorem, *Discusiones Math. IV* (1981), pp. 11-14.
38. van Leeuwen, Jan, Having a grundy numbering is NP-complete, Report 207, Computer Science Department, Pennsylvania State University, University Park, PA. 1976.
39. von Neumann, J. von, and O. Morgenstern, *Theory of Games and Economic Behaviour.* Princeton University Press, Princeton, 1944.
40. Niedermeier, R. and P. Rossmanith, Upper bounds for vertex cover further improved, *Proceedings 16th STACS*, Springer-Verlag LNCS 1563, (1999), pp. 561–570.
41. Reed, Bruce A., Treewidth and tangles: a new connectivity measure and some applications, *Surveys in Combinatorics*, London Mathematical Society Lecture Series 241, R. A. Bailey ed., Cambridge University Press 1997.
42. Seymour, P. D., and R. Thomas, Call routing and the ratcatcher, *Combinatorica* **14**, (1994), 217–241.
43. Thomas, Robin, Tree-decompositions of digraphs, Lecture notes. School of Mathematics, Georgia Institute of Technology, Atlanta, Georgia 30332, USA.