Cutting Up Is Hard To Do: the Parameterized Complexity of k-Cut and Related Problems

Vladimir Estivill-Castro
Griffith University, Australia

Rodney G. Downey
Victoria University, Wellington, New Zealand

Michael R. Fellows, Elena Prieto-Rodriguez and Frances A. Rosamond
University of Newcastle, Australia

EXTENDED ABSTRACT – 1 June 2002

Abstract

The GRAPH k-CUT problem is that of finding a set of edges of minimum total weight, in an edge-weighted graph, such that their removal from the graph results in a graph hagving at least k connected components. An algorithm with a running time of $O(n^{k^2})$ for this problem has been known since 1988, due to Goldschmidt and Hochbaum. We show that the problem is hard for the parameterized complexity class $W[1]$. We also investigate the complexity of a related problem, CUTTING A FEW VERTICES FROM A GRAPH, that asks for the minimum cost of separating at least k vertices from an edge-weighted connected graph. We show that this problem also is hard for $W[1]$.

1 Introduction

Graph partitioning problems have been much-studied, because of their important applications in VLSI design, parallel supercomputing, image processing and communications networks (see [Gon81, Har75, JAMc89, TZTS92] for a few examples of the extensive literature in this area). The following problem has been prominently studied, and can be solved in polynomial-time for every fixed k.

GRAPH k-CUT
Input: A graph $G = (V, E)$, an edge-weighting function $w : E \to I\!N$, and positive integers k, m.
Parameter: k
Question: Is there a set $C \subseteq E$ with $\sum_{e \in C} w(e) \leq m$, such that the graph $G' = G - C$ obtained from G by removing the edges of the *cut* C has at least k connected components?

A classic theoretical result concerning the k-CUT problem, due to Goldschmidt and Hochbaum [GH88, GH94] (also Karger and Stein [KS96]) is an algorithm with a running time of $O(n^{k^2})$, thus a polynomial-time algorithm for every fixed k.

This is not practical for $k \geq 3$ because of the exponential dependence on k. It is therefore natural to ask whether it might be possible to find an improved algorithm with a running time of the form $f(k)n^c$ where c is a constant independent of k. Improvements of this kind are not uncommon, and can have great practical significance.

The following are two relatively recent examples of this issue, one quite canonical for introductory discussions of parameterized complexity, and the other a bit less so.

Example 1: DOMINATING SET and VERTEX COVER. Both of these well-known problems (see [GJ79] for the definitions) take as input a graph G and a positive integer k. Both can trivially be solved in polynomial-time for every fixed k by trying all k-subsets, that is, in time $O(n^{k+1})$. Much attention has recently been focused on the fact that VERTEX COVER can be solved by an algorithm with a running time of the form $f(k)n^c$, and after many rounds of improvement, the best algorithm now known (due to Chen, Kanj and Jia [CKJ99]) has a running time of $(1.271)^k + kn$. A coarse-grained parallel implementation by Dehne shows that this algorithm is practical for a very large range of k [Deh02]. Both problems are *NP*-complete and cannot be distinguished in that framework. However, in the framework of parameterized complexity, they can be distinguished. DOMINATING SET is known to be $W[2]$-complete [DF95a]. This provides strong evidence that DOMINATING SET is unlikely to admit any algorithm whose complexity is of the form we are seeking. The nature of this evidence will be discussed in the next section. (As an example of an open problem of this sort, DIRECTED FEEDBACK VERTEX SET, which asks whether k vertices can cover all cycles in a directed graph, is still unresolved with respect to this issue.)

Example 2: THE EUCLIDEAN TRAVELING SALESMAN PROBLEM. In 1996, Arora [Ar96] described a polynomial-time approximation scheme (PTAS) for the Euclidean TSP, but one having a running time of approximately $O(n^{3000k})$ to produce a solution within a factor of $(1 + 1/k)$ of optimal. This is, of course, completely impractical, although polynomial-time for every fixed k (the definition of a PTAS). It would be natural to ask whether the exponent of the polynomial need be a function of k. There is good news, in this case, as in 1997 Arora produced an improved algorithm with a running time of our desired form [Ar97]. For many PTAS's (many of which have massive functions of k in the exponent) the analogous question remains unresolved, although in a few cases there are W-hardness results showing that such improvements are unlikely. For example, the PTAS's due to Khanna and Motwani for three planar logic problems [KM96] have been shown to be $W[1]$-hard with respect to this parameter by Cai, Fellows, Juedes and Rosamond [CFJR01] (the proof can be found in the survey [Fe01]). [1]

Our results here are only "lower bounds." We offer two negative results concerning the parameterized complexity of one relatively well-known problem, GRAPH k-CUT, and another related problem that has recently received some attention, CUTTING k VERTICES FROM A GRAPH [FKN00, FKN01]. We show that both of these problems are hard for $W[1]$.

2 The Nature of the Negative Evidence

This section has two related purposes, one of which is tangential to our main objective of bad news about the two target problems, but nevertheless of interest to anyone who wishes to investigate the parameterized complexity of concrete problems. The first purpose is just to quickly review some

[1] Valerie King seems to have been the first to suggest parameterizing with respect to the goodness of approximation [Ki94].

of the basics of parameterized complexity, as it currently stands, and explain why a $W[1]$-hardness demonstration indicates that a problem is unlikely to admit an algorithm having a running time of the form we seek.

The second purpose is to quickly sketch what seems to be a "revolutionary" development in the foundations of the theory. The entire framework for negative results can now be set up in just a few easy pages — at the price of a weaker core hypothesis. This all seems sufficiently interesting to us to merit some inclusion in this "background" section, although the main objective of this paper is just the two negative results on the concrete problems about cutting up graphs.

2.1 A Review of the Fundamental Notions

Parameterized complexity is fundamentally a 2-dimensional complexity theory, although apart from that, it is in many ways, including the "look-and-feel" of intractability arguments, very much akin to *NP*-completeness, and it could perhaps be argued that it is a natural sequel. The basic object of study is a parameterized problem. The parameter can be essentially *anything* (e.g., *anything that might end up in the exponent*, including $k = 1/\epsilon$ for PTAS's, as in Example 2 of §1). The reader can find an even more radical example in §2.2.

Definition. A *parameterized language* is a subset $L \subseteq \Sigma^* \times \Sigma^*$.

Definition. A parameterized language L is *fixed-parameter tractable* (*FPT*) if there is a function f (unrestricted), and an algorithm to determine if $(x, k) \in L$ in time $f(k) + n^c$ where $|x| = n$ and c is a constant independent of k.

One might think that if the above definition were modified by replacing " $f(k) + n^c$ " with " $f(k) \cdot n^c$ " then we might end up with a different class of parameterized languages. But no, "additively *FPT* " and "multiplicatively *FPT* " are equivalent notions.

The usual sort of problem reduction in parameterized complexity is defined:

Definition. A parameterized language L is *many:1 parametrically reducible* to a parameterized language L' if there is an FPT algorithm that transforms (x, k) into (x', k') so that:

1. $(x, k) \in L$ if and only if $(x', k') \in L'$, and
2. $k' = g(k)$ (where g is an unrestricted function; that is, k' is purely a function of k)

One can also formulate parameterized Turing reductions.

Our negative evidence for the two concrete problems about cutting up graphs is based on the following fundamental theorem, which is in some sense the parameterized analog of Cook's Theorem.

Theorem ([DF95b]+[CCDF97]). The following two problems are equivalent with respect to *FPT* reductions:
(1) The k-CLIQUE problem: given a graph G and a positive integer parameter k, does G contain a k-clique?

(2) The k-STEP NONDETERMINISTIC HALTING PROBLEM: given a nondeterministic Turing machine M (with unrestricted nondeterminism and alphabet size) and a positive integer parameter k, is it possible for M to halt in at most k steps, starting with an empty tape?

This equivalence seems quite remarkable, that the k-CLIQUE problem is fixed-parameter tractable if and only if the k-STEP HALTING PROBLEM is fixed-parameter tractable, and the proof is intricate. A more streamlined presentation of this basic result, rephrasing the proof in logic formalism (as contrasted with the circuit combinatorics of [DF98] and [CCDF97]) has been attempted in [Gr01].

The significance of this theorem to the working complexity theorist is very much akin to Cook's Theorem, which shows that 3SAT and the POLYNOMIAL-TIME HALTING PROBLEM FOR NONDETERMINISTIC TURING MACHINES are polynomial-time equivalent. The latter problem is trivially complete for NP for the usual definition of the class. The connection to 3SAT is significant, because we then have a combinatorially simple starting point for demonstrations that problems are unlikely to be in P — because our intuition testifies strongly against expecting the amorphous and opaque P-TIME NDTM HALTING PROBLEM to be in P.

Similar intuitions inform our expectation that there is little one can do with the k-STEP NONDETERMINISTIC TURING MACHINE HALTING PROBLEM other than explore the possible k-step computation paths exhaustively in time $O(|M|^{O(k)})$. The k-STEP NONDETERMINISTIC HALTING PROBLEM is complete for $W[1]$, and the connection supplied by the theorem above provides k-CLIQUE as a combinatorially useful starting point for demonstrations of likely parametric intractability.

In both of our intractability results, we will reduce from k-CLIQUE.

2.2 A New Approach to the Foundations of Intractability

Our goal in this section is to quickly sketch how we can have k-CLIQUE to work with for intractability demonstrations, in a completely self-contained and relatively simple way that does not depend on the difficult fundamental theorem of §2.1. The price for this simplicity is a seemingly weaker intuitive reference point, codified in the following conjecture.

Conjecture. There is no algorithm with running time $2^{o(n)}$ that determines, for a Boolean circuit C of total description size n, whether there is an input vector x such that $C(x) = 1$.

Circuits are also unstructured and opaque, and it is hard to imagine radically improving on the brute force approach of trying all possible inputs. In any case, no one knows how to do this.

Note that n is not the number of inputs to the circuit C, but the total size of a description of C. We will refer to this problem as CIRCUIT SAT. We will need the following simple algebraic facts whose proof is left to the reader.

Lemma 1. The following functions are FPT.
(a) $2^{o(k \log n)}$
(b) $(\log n)^k$

The following *parameterized miniaturization* of CIRCUIT SAT is our starting point. The investigation of such parameterized miniatures of *NP*-complete problems was essentially initiated by Cai and Juedes [CJ01].

MINI-CIRCSAT
Input: Positive integers k and n in unary, and a Boolean circuit C of total description size at most $k \log n$.
Parameter: k
Question: Is there any input vector x such that $C(x) = 1$?

The reader should pause to note that the definition is actually "legal". Note too that trying all possible inputs gives a brute force $O(n^k)$ algorithm. Now we have the crucial foundational lemma, essentially due to Cai and Juedes [CJ01].

Lemma 2. MINI-CIRCSAT is fixed-parameter tractable if and only if the conjecture fails.

Proof. One direction follows from Lemma 1(a). In the other direction, suppose we are given a Boolean circuit C of size N, and suppose that MINI-CIRCSAT is solvable in FPT time $f(k)n^c$. Take $k = f^{-1}(N)$ and $n = 2^{(N/k)}$. Then, of course, $N = k \log n$. For example, if $f(k) = 2^{2^k}$ then $f^{-1}(N) = \log\log N$. In general, $k = f^{-1}(N)$ will be some slowly growing function of N, and therefore $N/k = o(N)$, and also $cN/k = o(N)$ since c is a constant, and furthermore by trivial algebra $cN/k + \log N = o(N)$. Using the FPT algorithm, we thus have a running time of

$$f(f^{-1}(N))(2^{N/k})^c = N2^{cN/k} = 2^{cN/k+\log N} = 2^{o(N)}$$

to analyze the circuit C.

We now have the opportunity for some elegant and interesting combinatorics, much of which "recycles" some familiar combinatorics from the theory of *NP*-completeness. We would like to have a complexity class in which to place MINI-CIRCSAT, so we simply define $MINI[1]$ to be the parameterized problems that are *FPT*-equivalent to MINI-CIRCSAT. It turns out that *many* (but not all) $k \log n$ *miniatures* of familiar *NP*-complete problems are $MINI[1]$-complete [DFPR02]. The following are some examples that give us a way to complete our alternative foundational sketch.

MINI-3SAT
Input: Positive integers k and n in unary, and a 3SAT expression E of size at most $klogn$.
Parameter: k
Question: Is E satisfiable?

One can similarly define MINI-SAT by not insisting on size 3 clauses.

MINI-VERTEX COVER
Input: Positive integers k and n in unary, a graph G of total description size at most $k \log n$, and a positive integer r.
Parameter: k
Question: Does G have a vertex cover of size at most r?

One can similarly define MINI-INDEPENDENT SET. In fact, note that they are essentially the same problem (because here the parameter is *not* the size of the vertex sets).

Lemma 3. MINI-SAT, MINI-3SAT, MINI-Vertex Cover and MINI-Independent Set are all $MINI[1]$-hard.

Proof. The usual reductions (from *NP*-completeness theory) of Circuit Sat to SAT to 3SAT are all (crucially, for our purposes here) *linear size* reductions. Applied to the miniatures, these are then FPT reductions. The usual reduction ("a triangle for every variable, and a triangle for every clause") of 3SAT to Vertex Cover is also linear size.

Our final step in this "easy alternative foundation" for parametric intractability is to reduce MINI-Independent Set to the usual parameterized Independent Set problem, where the parameter is the size of the independent set.

Lemma 4. Independent Set (parameterized by the size of the independent set) is $MINI[1]$-hard.

Proof. Let $G = (V, E)$ be the miniature, for which we wish to determine whether G has an independent set of size r. Here, of course, $|V| \leq k \log n$ and we may regard the vertices of G as organized in k blocks $V_1, ..., V_k$ of size $\log n$. We now employ a simple but useful *counting trick* that can be used when reducing miniatures to "normal" parameterized problems. Our reduction is a Turing reduction, with one branch for each possible way of writing r as a sum of k terms, $r = r_1 + \cdots + r_k$, where each r_i is bounded by $\log n$. By Lemma 1(b) there are FPT-many branches. A branch represents a commitment to choose r_i vertices from block V_i (for each i) to be in the independent set.

We now produce (for a given branch of the Turing reduction) a graph G' that has an independent set of size k if and only if the miniature G has an independent set of size r, distributed as indicated by the commitment made on that branch. The graph G' consists of k cliques, together with some edges between these cliques. The ith clique consists of vertices in 1:1 correspondence with the subsets of V_i of size r_i. An edge connects a vertex x in the ith clique and a vertex y in the jth clique if and only if there is a vertex u in the subset $S_x \subseteq V_i$ represented by x, and a vertex v in the subset $S_y \subseteq V_j$ represented by y, such that $uv \in E$.

Verification that the reduction works correctly is straightforward.

A consequence of Lemma 4 is that $FPT \subseteq MINI[1] \subseteq W[1]$. There are reasons to conjecture that these containments are proper [DFPR02]. We now have completed our excursion through an alternative foundational discussion, having arrived (one way or the other) at the point of having k-Clique (trivially FPT equivalent to k-Independent Set) available to use as a convenient starting point for parametric intractability demonstrations. In the next two sections we apply this tool to the cutting-up problems.

3 The k-Cut Problem is Parametrically Intractable

Theorem 1. Graph k-Cut is hard for $W[1]$.

Proof. We reduce from k-Clique. Let $G = (V, E)$ be the graph on n vertices for which we wish to determine if G has a k-clique. We construct a graph G' as follows:

(1) Start with $n+2$ disjoint cliques of size n^4. We will refer to these as the *top clique* C_{top}, the *bottom clique* C_{bottom} and the *n vertex cliques* C_v, $v \in V$.
(2) For each $v \in V$, connect C_v to C_{bottom} by a matching of size $n^2 + n - \deg(v)$.
(3) For each $v \in V$, connect C_v to C_{top} by a matching of size $\binom{k}{2}$.
(4) For each edge $uv \in E$, make a single edge from C_u to C_v.
Each edge weight is 1; take $m = k(n^2+n) + (k-1)\binom{k}{2}$; take $k' = k+1$.

We claim that G has a k-clique if and only if (G', k', m) (with edge weights all 1) is a yes-instance for the GRAPH k-CUT problem.

One direction is easy — if G has a k-clique on the k vertices $V' \subseteq V$, then we can cut out k vertex cliques as follows (leaving the rest in a further component, noting $k' = k+1$):
(1) To cut the k vertex cliques C_u, $u \in V'$, from C_{bottom} requires almost all of the m cuts we are allowed. After we have accomplished this much, we have only $(k-1)\binom{k}{2} + \sum_{u \in V'} \deg(u)$ further cuts that we can make.
(2) The second term in the sum of our remaining budget of cuts from (1) allows us to cut the single edges joining the cliques corresponding to V' to the cliques corresponding to $V - V'$, and to each other (that is, the single edges between C_u and C_v where both $u, v \in V'$), requiring a number of cuts equal to all of this term, except for a savings of 1 for each adjacency of vertices in V'. Since there are $\binom{k}{2}$ such adjacencies, we have a remaining budget of $k\binom{k}{2}$ after this stage of cutting.
(3) The remaining budget of cuts allows us to cut all of the cliques corresponding to V' free from C_{top}.

In the other direction, we must argue that if (G', k', m) (with all edge weights 1) is a yes-instance to GRAPH k-CUT problem, then G must have a k-clique. This is almost obvious from the discussion above, if we are forced into such a tight channel of cuts. But this is clear, since each of the cliques that we began with in step (1) of the construction of G' has n^4 vertices, so there is no point in cutting any of the edges of these cliques, they are invincible. Equivalently stated, if we are dealing with a yes-instance, then we can assume that no edges of the fundamental cliques of G' are cut in a solution. This means that there must be a way to accomplish the three cutting tasks described above: (1) to separate k vertex cliques from the bottom clique, and (2) to separate them from each other and from the remaining vertex cliques, and (3) also from the top clique. Because of the tight budget of cuts, this can only be accomplished if the vertices represented by the k cliques of G' that are separated by the cutting form a k-clique in G.

4 Cutting k Vertices from a Graph is Intractable

We consider here a related problem that has recently received some attention [FKN00, FKN01]. The problem is defined as follows.

CUTTING k VERTICES FROM A GRAPH
Input: A graph $G = (V, E)$, an edge-weighting function $w : E \to I\!N$, and positive integers k, m.
Parameter: k
Question: Is there a set of k vertices $V' \subseteq V$ such that $\sum_{uv \in E : u \in V', v \in V - V'} w(uv) \leq m$?

Another way to phrase the above question is to ask whether there is in G a $(k, n-k)$-*cut* of cost at most m.

Theorem 2. CUTTING k VERTICES FROM A GRAPH is hard for $W[1]$.

Proof. We reduce from k-CLIQUE, and our proof is in many regards similar to the proof of Theorem 1. Again, let $G = (V, E)$ be an n-vertex graph for which we wish to determine whether G has a k-clique. We produce G' as follows:
(1) We start with a clique of size n^3 (the *base clique*) and n *representative* vertices corresponding $1:1$ with the vertices of G.
(2) Every representative vertex v is connected to $n^2 - \deg(v)$ vertices of the base clique.
(3) If $uv \in E$ then in G' the vertices corresponding to u and v are made adjacent.
Take $m = kn^2 - 2\binom{k}{2}$.

We claim that G has a k-clique if and only if (G', m), with all edge-weights 1, is a yes-instance for CUTTING k VERTICES FROM A GRAPH.

One direction is easy. If $V' \subseteq V$ is a k-clique in G, then the number of edges between V' and $V - V'$ is $\sum_{v \in V'} \deg(v) - 2\binom{k}{2}$. Therefore the budget of m cuts allows these k vertices to be cut from the rest of G.

In the other direction, any k vertices to be cut from G' at a cost of m cannot include any vertex of the base clique, since the cost of separating that vertex from the rest of the base clique already exceeds the total budget of cuts m. It follows that all of the k vertices of a "cutting up" solution set V' must be representatives in G'. The rest follows by contraposition: if they are not all adjacent, then there is not enough in the budget to separate them all from the rest of G'.

5 Conclusions and Open Problems

The main contributions of this paper are two:
(1) We have principally shown that two fairly well-known, concrete problems about cutting up graphs are fixed-parameter intractable.
(2) We have also given a sketch of a new, alternative foundation for intractability proofs that is much simpler than the foundations of $W[1]$. (Many of the key ideas in this were implicit, in some sense, in [CJ01], although many things here are new, including the definition of $MINI[1]$ and of miniatures of *NP* problems, most of the combinatorial reductions, especially Lemma 4, the Turing counting trick, and the inclusion of $MINI[1]$ in $W[1]$.)

Everything in this paper seems to be too easy, but that's necessarily not a bad thing. There is a common impression that $W[1]$-hardness demonstrations are typically heroic, and that the foundations of the theory are cryptic and difficult. Our (easy) results should encourage other investigations of the many natural questions that remain open about whether the parameter can be removed from the exponent for problems that are "polynomial for every fixed k" — but only uselessly, by current knowledge.

It would be interesting to know if, for some unresolved parameterized problems, reductions from parametric miniatures (such as MINI-3SAT, etc.) might provide easier routes for intractability demonstrations. There is some evidence that this may be so. We have recently found a relatively easy reduction from MINI-1-IN-3-3SAT to WEIGHT DISTRIBUTION FOR LINEAR CODES — a *much* simpler proof of parametric intractablility (modulo a weaker core conjecture) than the very complicated $W[1]$-hardness proof for this problem described in [DFVW99]. Could it be that some natural parameterized problems are $MINI[1]$-hard but not $W[1]$-hard?

With respect to graph cutting, there are a couple of notable open problems:
(1) Feige, Krauthgamer and Nissim have recently shown that for $k = O(\log n)$ there is a PTAS for the problem of computing a minimum $(k, n-k)$ cut [FKN01]. However, the parameter that governs the goodness of the approximation, $1/\epsilon$, occurs in the exponent of the polynomial running time. It would be interesting to know if this approximation problem is parametrically intractable.
(2) For planar graphs there is an algorithm for the GRAPH k-CUT problem that runs in time $O(n^{ck})$ [DJPSY92]. Does the problem remain $W[1]$-hard for planar graphs?

References

[Ar96] S. Arora, Polynomial time approximation schemes for Euclidean TSP and other geometric problems. In: *Proceedings of the 37th IEEE Symposium on Foundations of Computer Science*, 1996, pp. 2–12.

[Ar97] S. Arora. Nearly linear time approximation schemes for Euclidean TSP and other geometric problems. *Proc. 38th Annual IEEE Symposium on the Foundations of Computing* (FOCS'97), IEEE Press (1997), 554-563.

[CCDF97] Liming Cai, J. Chen, R. Downey and M. Fellows. The parameterized complexity of short computation and factorization. *Archive for Mathematical Logic* 36 (1997), 321-338.

[CFJR01] Liming Cai, M. Fellows, D. Juedes and F. Rosamond. Efficient polynomial-time approximation schemes for problems on planar structures: upper and lower bounds. Manuscript, 2001.

[CJ01] Liming Cai and D. Juedes. Subexponential parameterized algorithms collapse the W-hierarchy. *Proceedings of ICALP 2001*, Crete, Greece, Springer-Verlag LNCS 2076 (2001).

[CKJ99] J. Chen, I.A. Kanj and W. Jia. Vertex cover: further observations and further improvements. *Proceedings of the 25th International Workshop on Graph-Theoretic Concepts in Computer Science* (WG'99), *Lecture Notes in Computer Science 1665* (1999), 313–324.

[Deh02] F. Dehne. Private communication, 2002.

[DF95a] R. G. Downey and M. R. Fellows, Fixed-parameter tractability and completeness I: basic theory. *SIAM Journal of Computing* 24 (1995), 873–921.

[DF95b] R. Downey and M. Fellows. Fixed-parameter tractability and completeness II: completeness for $W[1]$. *Theoretical Computer Science A* 141 (1995), 109-131.

[DF98] R. Downey and M. Fellows. *Parameterized Complexity* Springer-Verlag (1998).

[DFPR02] R. Downey, M. Fellows, E. Prieto-Rodriguez and F. Rosamond. Fixed-parameter tractability and completeness V: parametric miniatures. Manuscript in preparation, 2002.

[DFVW99] R. Downey, M. Fellows, A. Vardy and G. Whittle. The parameterized complexity of some fundamental problems in coding theory. *SIAM J. Computing* 29 (1999), 545-570.

[DJPSY92] E. Dalhaus, D.S. Johnson, C.H. Papadimitriou, P. Seymour and M. Yannakakis. The complexity of multiway cuts. *Proc. 24th Annual ACM Symp. on the Theory of Computing (STOC)* (1992), 241–251.

[Fe01] M. Fellows. Parameterized complexity: the main ideas, connections to heuristics and research frontiers. *Proc. ISAAC 2001*, Springer-Verlag, *Lecture Notes in Computer Science* 2223 (2001), 291–307.

[FKN00] U. Feige, R. Krauthgamer and K. Nissim. Approximating the minimum bisection size. *Proc. 32nd Annual ACM Symposium on the Theory of Computing (STOC)* (2000), 530–536.

[FKN01] U. Feige, R. Krauthgamer and K. Nissim. On cutting a few vertices from a graph. Manuscript, 2001.

[GH88] O. Goldschmidt and D.S. Hochbaum. Polynomial algorithm for the k-cut problem. *Proc. 29th Annual Symp. on the Foundations of Computer Science (FOCS)* (1988), 444-451.

[GH94] O. Goldschmidt and D.S. Hochbaum. A polynomial algorithm for the k-cut problem for fixed k. *Mathematics of Operations Research* 19 (1994), 24–37.

[GJ79] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.*W.H. Freeman, San Francisco, 1979.

[Gon81] T.F. Gonzalez. On the computational complexity of clustering and related problems. *Proc. 10th IFIP Conference on System Modeling and Optimization* (1981), 174–182.

[Gr01] M. Grohe. The parameterized complexity of database queries. *Proc. 20th ACM Symposium on Principles of Database Systems (PODS)* (2001), 82–92.

[Har75] J.A. Hartigan. *Clustering Algorithms.* Wiley, New York, 1975.

[JAMc89] D.S. Johnson, C.R. Aragon, L.A. McGeoch and C. Schevon. Optimization by simulated annealing: an experimental evaluation; Part 1 graph partitioning. *Operations Research* 37 (1989), 865–892.

[Ki94] V. King. Private communication, 1994.

[KM96] S. Khanna and R. Motwani, "Towards a Syntactic Characterization of PTAS," in: *Proc. STOC 1996*, ACM Press (1996), 329–337.

[KS96] D.R. Karger and C. Stein. A new approach to the minimum cut problem. *Journal of the ACM* 43 (1996), 601–640.

[TZTS92] L. Tao, Y.C. Zhao, K. Thulasiraman and M.N.S. Swamy. Simulated annealing and tabu search algorithms for multiway graph partitioning. *Journal of Circuits, Systems and Computers* 2 (1992), 159–185.