The Method of Extremal Structure on the k-MAXIMUM CUT Problem

Elena Prieto

School of Electrical Engineering and Computer Science
The University of Newcastle
Australia

Abstract

Using the *Method of Extremal Structure*, which combines the use of reduction rules as a preprocessing technique and combinatorial extremal arguments, we will prove the fixed-parameter tractability and find a problem kernel for k-MAXIMUM CUT. This kernel has $2k$ edges, the same as that found by Mahajan and Raman in (Mahajan & Raman 1999), but using our methodology we also find a bound of k vertices leading to a running time of $\mathcal{O}(k \cdot 2^{k/2} + n^2)$.

1 Introduction

k-MAXIMUM CUT is a very well studied problem both within and outside the field of parameterized complexity. The basic idea is to find a partition of the vertex set into two subsets which maximizes the set E' of edges crossing from one side of the partition to the other. It was first called BIPARTITE SUBGRAPH and proven NP-complete in 1976 by Garey and Johnson (Garey & Johnson& Stockmeyer 1976). It it is APX-complete (Papadimitriou& Yannakakis 1991), approximable within 1.1383 (Goemans & Williamson 1995) and admits a PTAS if $|E| = \Theta(|V^2|)$ (Arora & Karger & Karpinski 1995). It has applications in numerous fields (Poljak & Tuza 1995). The parameterized version of the problem, where we take as a parameter the number of edges in the set E' is formally defined as follows:

Instance:	A graph $G = (V, E)$ and a positive integer k
Parameter:	k
Question:	Is there a partition of the vertex set into two subsets A and B such that the subset $E' \subseteq E$ of edges with one endpoint in A and the other in B is at least k?

It is to be noted that the parametric dual of this problem, k-EDGES FROM BIPARTITE, in which we are asked to minimize the number of edges we erase to render the graph bipartite was a long standing open problem in the field of *Parameterized Complexity* and has only recently been proved fixed-parameter tractable (Reed& Smith& Vetta 2003).

The best known fixed-parameter algorithm for k-MAXIMUM CUT runs in time $\mathcal{O}(|E| + |V| + k \cdot 4^k)$ (Mahajan & Raman 1999). Fedin and Kulikov (Fedin& Kulikov 2002) give the current best known exact algorithm for the problem running in time $\mathcal{O}(poly(|E|) \cdot 2^{|E|/4})$.

2 Parameterized Complexity and Kernelization: The Method of Coordinatized Kernels

The subject of Parameterized Complexity is concretely motivated by an abundance of natural examples of two different kinds of complexity behavior. To illustrate this lets look at VERTEX COVER and DOMINATING SET. Although both problems are NP-complete, the introduction of the parameter k provides a new approach to their complexity. After many rounds of improvement on algorithms which efficiently solve the problems, the best known algorithm for VERTEX COVER runs in time $O(1.286^k + kn)$ (Chen& Kanj & Jia 2001) but nothing substantially better than the brute force algorithm of trying all k-subsets has been found to solve DOMINATING SET. As a way to formalize the difference between these two problems Downey and Fellows make the following definition:

Definition 1 *(Fixed Parameter Tractability) A parameterized problem $L \subseteq \Sigma^* \times \Sigma^*$ is* fixed-parameter tractable *if there is an algorithm that correctly decides for input $(x, y) \in \Sigma^* \times \Sigma^*$, whether $(x, y) \in L$ in time $f(k)\, n^\alpha$, where n is the size of the input x, |x|=n, k is the parameter, α is a constant (independent of k) and f is an arbitrary function.*

The family of fixed-parameter tractable problems is denoted FPT.

In the example above it is easy to see that VERTEX COVER is in FPT whereas it is known that DOMINATING SET is complete for $W[2]$ (Downey& Fellows 1999) and therefore unlikely to be in FPT. For an explanation of what we mean by $W[2]$ and its structural complexity we refer readers to the comprehensive study on Parameterized Complexity by Downey and Fellows (Downey& Fellows 1999).

2.1 k-MAXIMUM CUT is Fixed-Parameter Tractable

There are many different techniques to prove that a problem is fixed-parameter tractable. One of these ways is provided by Robertson and Seymor's Graph Minor Theorem:

Lemma 1 *The k-MAXIMUM CUT problem is in* FPT.

Proof of Lemma 1. It is enough to show that the YES-instances of the problem are closed under the minor operations. Let (G, k) be a YES-instance of k-MAXIMUM CUT, that is, G is a graph for which there exist two disjoint subsets A and B of V such that $V = A \cup B$ and $H = (A \cup B, E')$ is bipartite and $E' \subseteq A \times B$, $|E'| \geq k$. We prove that every reverse minor operation transforms (G, k) into another YES-instance.

- *Edge additions*: Let $(x, y) \notin E$ be an edge to be added to G with $x, y \in V$. If $x \in A$ and $y \in B$, then $E' + xy$ is a solution for $(G^*, k+1)$, and thus also (G^*, k) is a YES-instance. Otherwise assume w.l.o.g. that $x, y \in A$. Then E' is a solution for G^*.
- *Vertex additions*: Let $v \notin V$ be a vertex to be added to G. If the neighbors $N(v)$ of v are a subset of A or $N(v) \subseteq B$, then without loss of generality we assume $N(v) \subseteq A$. We create a solution for G^* in adding the edges from v to $N(v)$ to E' (and this extending B by v. If $N(v) \not\subset A$ and $N(v) \not\subset B$, then we place v in A, add to E' the edges going from v to vertices in B (ignoring the edges generated within A). Thus (G^*, k) is a YES-instance.
- *Vertex expansions*: Assume without loss of generality that the vertex z we are expanding (by replacing z with $x, y \in V$ and $(x, y) \in E$) is in A of the bipartite subgraph induced by solution E'. Then we create a solution for G^* from E' by replacing z with x in A, updating E' accordingly, and adding y to A. Thus, (G^*, k) is a YES-instance.

By the Graph Minor Theorem and its companion result that order testing in the minor order is in FPT we conclude that k-MAXIMUM CUT is in FPT (for details see (Downey& Fellows 1999)). □

The problem with using the Graph Minor Theorem to prove membership in FPT is that it is non-constructive and produces a fast growing function $f(k)$. In this paper we will use the *Method of Extremal Structure* to find a feasible fixed parameter algorithm for k-MAXIMUM CUT.

2.2 The Method of Extremal Structure

The kernel of a problem is defined as follows:

Definition 2 *A parameterized problem L is kernelizable if there is a parametric transformation of L to itself that satisfies:*

1. *The running time of the transformation of (x, k) into (x', k'), where $|x| = n$, is bounded a polynomial $q(n, k)$ (so that in fact this is a polynomial time transformation of L to itself, considered classically, although with the additional structure of a parametric reduction),*
2. *$k' \leq k$, and*
3. *$|x'| \leq h(k)$, where h is an arbitrary function.*

These polynomial time transformations are generally known as 'reduction rules'. It is important to note the following result by (Downey& Fellows& Stege 1999) whose proof is based on ideas from (Cai& Chen & Downey & Fellows 1997).

Lemma 2 *A parameterized problem L is in FPT if and only if it is kernelizable.*

Therefore, the goal is to try to produce reduction rules to reduce the size of the instance to a function of k. Then, simply performing an 'intelligent' brute force on the remaining instance will suffice to solve the problem in fixed-parameter tractable terms.

The *Method of Extremal Structure* is as systematic way to design FPT algorithms. It first appeared under the name of *Method of Coordinatized Kernels* in (Fellows & McCartin& Rosamond & Stege 2000) and it proceeds as follows:

- First we design a list of reduction rules specific to the targeted problem. We say an instance is 'irreducible' or 'reduced' when these rules are no longer applicable.
- Next we prove a *Kernelization Lemma* to obtain a problem kernel.
- The final step is to give a brute-force algorithm we apply to the kernel and the running time for the whole process.

In the core of the *Method of Extremal Structure,* the aim is to prove a Kernelization Lemma. In the case of solving a graph maximization problem such as k-MAXIMUM CUT, this lemma takes the following form:

Lemma 3 *(General Kernelization Lemma) If $G = (V, E)$ is an irreducible instance of a problem Π and G has more than $f(k)$ vertices, then (G, k) is a YES-instance for the problem.*

To prove such a lemma, we have to set up a list of reduction rules (leading to a specific notion of irreducibility) and a suitable bounding function $f(k)$.

Our techniques for deriving a smaller kernel are based on polynomial time reduction rules and combinatorial extremal arguments. The extremality of the argument will be used to prove a so-called *Boundary Lemma*. Generally, this lemma takes the following form:

Lemma 4 *(General Boundary Lemma) Let G be irreducible and let (G, k) be a YES-instance and $(G, k+1)$ be a NO-instance for Π. Then the number of vertices of G is less than $g(k)$ for some function g.*

We prove this lemma by induction. Since we want the function g mentioned above to be as small as possible, in many cases there will be a need to establish a set of one or more 'inductive priorities' to refine the structural claims even further.

The *combinatorial extremal argument* operates in the following fashion in the reduced instance:

1. Assume a solution (or "witness" structure) of size k satisfying certain inductive priorities. Here the fact that G doesn't have a solution of size $k+1$ will be implicitly understood as the zeroth inductive priority.
2. Set structural claims for the problem which are only possible if all inductive priorities are satisfied.

3. Stop when all claims are satisfied and the total size of graph is less than the function $g(k)$.

Once the Boundary Lemma is proved, the Kernelization Lemma follows by simple contradiction.

This *Method of Extremal Structure* was first devised to solve the k-MAX LEAF (Fellows & McCartin& Rosamond & Stege 2000) and has been used to tackle graph maximization problems such as k-STAR PACKING (Prieto & Sloper 2004) proving to be equally successful. Its usefulness has proven also important with minimization problems and even with non-graph problems like SET SPLITTING (Dehne & Fellows& Rosamond 2003). A complete description and analysis of the *Method of Extremal Structure* can be found in (Prieto 2004).

3 The Method on k-MAXIMUM CUT

3.1 Reduction Rules

First a trivial but important observation:

Observation 1 *If there exists a vertex $v \in V(G)$ of degree k or more, then G is a* YES*-instance for k-*MAXIMUM CUT.

Obviously we would put v on one side of the partition and all its neighbors on the other.

In this reduction rules, $G \setminus w$ denotes the graph obtained from G by deleting w (and of course all incident edges). Similarly, $G \setminus (u, v)$ denotes the graph obtained from G by deleting the edge (u, v).

Reduction Rule 1 *Let $u \in V$ be a vertex of degree 0. (G, k) is a* YES*-instance of k-*MAXIMUM CUT *if and only if $(G \setminus u, k)$ is a* YES*-instance of k-*MAXIMUM CUT.

Proof of Reduction Rule 1. True since u will never contribute with any edge to E'. □

Reduction Rule 2 *Let $u \in V$ be a vertex of degree 1 and let $v = N(u)$. (G, k) is a* YES*-instance of k-*MAXIMUM CUT *if and only if $(G \setminus u, k-1)$ is a* YES*-instance.*

Proof of Reduction Rule 2. We can always make u and v be in different subsets of the partition without affecting the rest of the graph. □

Reduction Rule 3 *Let u and v be two consecutive vertices in V of degree 2 adjacent to the same vertex $w \in V$. (G, k) is a* YES*-instance of k-*MAXIMUM CUT *if and only if $(G \setminus \{u, v\}, k-2)$ is a* YES*-instance.*

Proof of Reduction Rule 3.

"⇒" We may assume without loss of generality $w \in A$. An optimal solution of k-MAXIMUM CUTfor G would always put at least one vertex in u, v in the set B. Two different cases arise:

- Only one of the vertices is in B. Assume without loss of generality the vertex is u. Since v is in A both edges (u, v) and (v, w) are in E' and therefore by erasing u and v we decrease the size of k by 2.
- If both u and v are in B then the edges (u, w) and (v, w) are in E' and therefore by erasing u and v we decrease the size of k by 2 as in the previous case.

"⇐" We have to prove that by adding two consecutive degree 2 vertices u and v to any given vertex w we will always obtain a solution with 2 more edges in E'. We may assume again without loss of generality that w is in A. By simply making u and v be in B we gain two edges in E', namely (u, w) and (v, w). □

Reduction Rule 4 *Let u be a vertex of degree 2 in V and let $N(u) = \{x, y\}$ with $(x, y) \in E$. (G, k) is a* YES*-instance of k-*MAXIMUM CUT *if and only if $(G \setminus \{u, (x, y)\}, k-2)$ is a* YES*-instance.*

Proof of Reduction Rule 4.

"⇒" G is a graph with a bipartition of its vertex set which ensures k edges in E'. Two cases arise in the analysis of the instance:

- Both x and y are in the same side of the partition. Any optimal solution would place u in the opposite side of this partition. By erasing u we are going to remove for certain two edges from E'.
- If both x and y are in the opposite sides of the partition then the edge $(x, y) \in E'$. No matter where we place u it is always going to have one adjacency with a vertex on the other side of the partition which would add another element to the set E'. By erasing both (x, y) and u we are going to decrease the size of E' by two.

"⇐" Let G be a graph with a bipartition of its vertex set which ensures $k-2$ edges in E'. Take any two independent vertices x and y. The same two cases as in the previous implication arise depending on whether x and y are in the same or opposite sides of the partition. Working the prove above backwards would add two edges to E' as required. □

Reduction Rule 5 *Let x, y and z be three consecutive vertices of degree 2 in V. Let $N(x) = \{u, y\}$, $N(y) = \{x, z\}$, $N(z) = \{y, v\}$. (G, k) is a* YES*-instance of k-*MAXIMUM CUT *if and only if $(G \setminus \{y, z\} \cup (x, v), k-2)\}$ is a* YES*-instance.*

Proof of Reduction Rule 5.

"⇒" G is a graph with a bipartition of its vertex set which ensures k edges in E'. Two cases arise in the analysis of the instance:

- Both u and v are in the same side of the partition. Without loss of generality we may assume this side is A since the argument is symmetrical if they are both in B. Any optimal solution would place x and z in B and y in A. The edges $(u, x), (x, y), (y, z)$ and (z, v) are then all in E'. By erasing y and z we are going to remove for certain three edges from E'. By adding (x, v) we add one. The result is an instance with $k-2$ edges in E'.

- If both u and v are in the opposite sides of the partition then, again without loss of generality, we may assume that $u \in A$ and $v \in B$. An optimal solution could place x and z in B and y in A which will position edges $(u, x), (x, y)$ and (y, z) in E'. By erasing y and z we are going to remove for certain two edges from E'. By adding (x, v) we don't add any edges to E' since both x and v are in B. The result is an instance with $k - 2$ edges in E'.

"⇐" Let G be a graph with a bipartition of its vertex set which ensures $k - 2$ edges in E'. Take any two vertices u and v in V. The same two cases as in the previous implication arise depending on whether u and v are in the same or opposite sides of the partition. Working the prove above backwards would add two edges to E' as required. □

3.2 Kernelization/Boundary Lemmas

Lemma 5 ***(Kernelization Lemma (Vertices))*** *If a graph G is reduced and $|V(G)| > k$, then it does contain a k-*MAXIMUM CUT.

Lemma 6 ***(Boundary Lemma (Vertices))*** *If a graph G is reduced and is a* YES*-instance for k-*MAXIMUM CUT, *but a* NO*-instance for $(k+1)$-*MAXIMUM CUT *then $|V(G)| \leq k$.*

Proof of Lemma 6. Assume in contradiction that there exists a counterexample G', such that G' is reduced and contains a k-MAXIMUM CUT, but there is no partition of the vertex set which induces $k+1$ edges and that $|V(G)| > k$.

Let us consider as a witness structure the partition of the vertex set $V = (A, B)$ with k edges $(u, v) \in E$ where $u \in A$ and $v \in B$. These edges constitute E'. The graph has more edges which are not in E'. We will call these edges *ghost edges*. Each element $u \in A$ has two degrees associated to it, namely $deg_{|A}(u)$and $deg_{|B}(u)$. It is clear that $deg_{|A}(u)$+$deg_{|B}(u)$= $deg(u) \leq k$ by observation 1. Similarly, each element $v \in B$ has two degrees associated to it $deg_{|A}(v)$and $deg_{|B}(v)$with the same property.

We will also consider the following inductive priority:

- Among all counterexamples, choose the one where the size of A is minimum.

Based on the reduction rules derived in section 3.1, we can now prove the claimed kernel size bound of $|V(G)| = k$ by resorting to the extremal properties of our assumed counter example. We will now start a series of claims that bound the number of vertices in G.

Structural Claims

Claim 1 *Every vertex in A is adjacent to at least one vertex in B and viceversa.*

Proof of Claim 1. Assume in the contrary that there exist a vertex $v \in A$ which is not connected to any vertex in B. By reduction rule 1 we know that there are no isolated vertices in the graph, so v must be adjacent to a vertex in A. Then by moving v to B we would add one edge to E' and therefore our graph would become a YES-instance for $(k+1)$-MAXIMUM CUT, a contradiction. An analogous argument holds for B. □

Claim 2 *There are no* ghost edges *between A and B.*

Proof of Claim 2. If there are any then G would be a YES-instance for $(k+1)$-MAXIMUM CUT. □

Claim 3 $|A| \leq |B| \leq k$.

Proof of Claim 3. $|A| \leq |B|$ since otherwise we could contradict our inductive priority by simply inverting the roles of A and B. The size of the largest one of the two sets is less than or equal to k since every vertex in the graph is adjacent to at least one vertex on the other side of the partition. □

Claim 4 $|A| \leq \frac{k}{2}$

Proof of Claim 4. If $deg_{|B}(u)$=l then u contributes to E' with l edges. We will argue that each vertex in A contributes with at least two edges to E' and therefore we can have at most $\frac{k}{2}$ vertices. Assume that there exists a vertex $u \in A$ such that $deg_{|B}(u)$=1. By reduction rule 2 there are no vertices of degree 1 in the graph so therefore u must have at least one neighbor u' in A. But if this is the case, we could move u to B. We would lose one edge in E', namely $(u, N_{|B}(u))$ but we gain at least one (u, u'). The result is still a YES-instance for k-MAXIMUM CUT, but the set A has one element less contradicting the inductive priority.[1] □

Claim 5 *For all $u \in A$, $deg_{|B}(u)$> $deg_{|A}(u)$.*

Proof of Claim 5. If $deg_{|B}(u)$< $deg_{|A}(u)$, moving u to B would cause us to lose $deg_{|B}(u)$ edges from E' and gain $deg_{|A}(u)$ thus creating a YES-instance for $(k+1)$-MAXIMUM CUT, a contradiction. If $deg_{|B}(u)$= $deg_{|A}(u)$, moving u to B would create another YES-instance for k-MAXIMUM CUT where the size of A would be smaller, contradicting the minimality of A ensured by the inductive priority. □

Claim 6 *For all $v \in B$, $deg_{|A}(v)$≥ $deg_{|B}(v)$.*

Proof of Claim 6. If $deg_{|A}(v)$< $deg_{|B}(v)$, moving v to A would cause us to lose $deg_{|A}(v)$ edges from E' and gain $deg_{|B}(v)$ thus creating a YES-instance for $(k+1)$-MAXIMUM CUT, a contradiction. □

[1]Note that the kernel size so far has less than $\frac{3}{2}k$, but the number of edges is still quadratic on k.

Claim 7 *For all $u \in A$ and $v \in B$, $\sum_{u \in A} deg_{|B}(u)$= $\sum_{v \in B} deg_{|A}(v)$= k.*

Proof of Claim 7. By the nature of the witness structure. □

The degree of a vertex v is the number of neighbors of v in G. We define degree* of a vertex v to be the degree of v excluding those neighbors lying on the same side of the partition. Hence, the average degree* of the graph is the total number of edges excluding the *ghost edges* divided by the number of vertices in the graph.

Claim 8 *If the average degree* is at least 2, then there are at most k vertices in $V(G)$.*

Proof of Claim 8. By claim 7 $\sum_{u \in A} deg_{|B}(u)$= $\sum_{v \in B} deg_{|A}(v)$= k and therefore the if the average degree is 2 we have

$$\frac{\sum_{u \in A} deg_{|B}(u) + \sum_{v \in B} deg_{|A}(v)}{|V(G)|} = \frac{2k}{|V(G)|} \geq 2$$

$$\Rightarrow |V(G)| \leq k$$

□

Claim 8 states that ***if*** the average degree* is greater than 2 ***then*** the number of vertices is less than k. Now we need to prove that the average degree* is in fact at least 2. To do this we proceed by looking at the elements of A (which we already know have degree* greater than 2 by Claim 4). We demonstrate that for each u in A,

$$\frac{deg^*(u) + deg^*(N_{|B}(u))}{|1 + N_{|B}(u)|} \geq 2$$

We will call this quantity the *average degree** of $\langle u, N_{|B}(u) \rangle$.

Claim 9 *Let $u \in A$ and let $deg_{|B}(u)$= l. Assume $N_{|B}(u) = \{x_1, x_2, \ldots, x_l\}$. There are at most $\lfloor \frac{l}{2} \rfloor$ vertices amongst $N_{|B}(u)$ such that $deg_{|A}(x_i) = 1$.*

Proof of Claim 9. If there were more we could move them all to A and move u to B. By doing this, for each vertex we move to A we are gaining two edges in $|E'|$ and therefore a total of $2 \cdot (\lfloor \frac{l}{2} \rfloor + 1) \geq l + 1$ against the l vertices we lose by moving u to B. We obtain a set E' with $k + 1$ edges, a YES-instance for $(k + 1)$-MAXIMUM CUT, and thus a contradiction. □

Claim 10 *Let $u \in A$ such that $deg_{|B}(u)$$\geq 3$. The average degree* of $\langle u, N_{|B}(u) \rangle$ is at least 2.*

Proof of Claim 10. By claim 9, at most $\lfloor \frac{l}{2} \rfloor$ vertices in $N_{|B}(u)$ have degree 1 in A. Let's say there is l_1 of those vertices. The vertices in $N_{|B}(u)$ with degree ≥ 2 will be l_2. It is clear that by construction $l_1 + l_2 = l$. The average degree* of $\langle u, N_{|B}(u) \rangle$ is at least

$$\frac{1 \cdot l_1 + 2 \cdot l_2 + l}{l + 1} = \frac{l_1 + l_2 + l + l_2}{l + 1}$$
$$= \frac{2l + l_2}{l + 1} = \frac{2l + 2}{l + 1} + \frac{l_2 - 2}{l + 1} = 2 + \frac{l_2 - 2}{l + 1}$$

This quantity is less that 2 only when $\frac{l_2 - 2}{l+1} < 0$ and since $l_2 < \lceil \frac{l}{2} \rceil$ we have that the average degree is less that 2 when

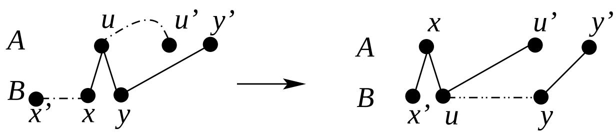
$$\lceil \frac{l}{2} \rceil - 2 < 0 \Rightarrow l < 3$$

□

Claim 11 *Let $u \in A$ such that $deg_{|B}(u)$ = 2. The average degree* of $\langle u, N_{|B}(u) \rangle$ is at least 2.*

Proof of Claim 11. Let's denote $N_{|B}(u) = \{x, y\}$. If $deg_{|B}(u) = 2$ for $u \in A$ then $deg_{|A}(u) \leq 1$ by claim 5. Two cases arise:

1. $deg_{|A}(u)$ = 1.
   Then both x and y must have other neighbors in A besides u and thus the average degree* is at least $\frac{2+2+2}{3}$ and thus at least 2.
   Proof. Assume that at least one of them did not have any neighbors in A other than u. Without loss of generality say $N_{|A}(x) = \{u\}$. By moving u to B and x to A we obtain an instance with one more edge in E'. It is easy to see that in the original witness structure we had 3 edges in E' and when we move u to B and x to A we have 4. This contradicts the fact that G is a NO-instance for $(k+1)$-MAXIMUM CUT.



□

2. $deg_{|A}(u)$ = 0.
   The following statements hold:
   (a) Either x or y or both have neighbors in A besides u.

   Proof. Assume on the contrary that $N_{|A}(x) = N_{|A}(y) = \{u\}$. The by moving x and y to A and u to B we generate a witness structure with 4 edges in E' when before we only had 2. This contradicts the fact that G is a NO-instance for $(k + 1)$-MAXIMUM CUT. □

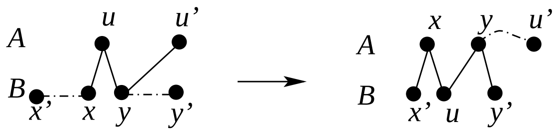
   Unfortunately, if only one of the neighbors of u in B has degree 1 in A, the average degree is not greater than or equal to 2, but further analysis will prove that the claim holds true.□

(b) If x has no neighbors in A other than u then $deg(x) = 2$.

Proof. Assume on the contrary that $deg(x) > 2$. Then since we are assuming that u is its only neighbor in A it must have at least two neighbors in B. This would imply that $deg_{|A}(x) < deg_{|B}(x)$, contradicting claim 6.

(c) If $deg_{|A}(x)$= 1 then $deg_{|A}(y) \geq 3$.

Proof. Assume on the contrary that $deg_{|A}(y)$= 2. By Claim 2a we know that $deg_{|A}(y)$cannot be 1. It is clear that y cannot be of degree 2 in the graph by Reduction Rule 5 because $deg(u) = deg(x) = 2$, so y must have another neighbor y' in B. By Reduction Rule 4, $y' \neq x$. Now, sending x and y to A and u to B would generate one extra edge in E' thus contradicting that G is a NO-instance for $(k+1)$-MAXIMUM CUT.



□

A problem arises now, as two different elements in A could share their high degree* neighbors in B thus making the average degree* less than 2. In the next statement we prove this is not possible.

(d) Let $u_1, u_2 \in A$ such that $deg_{|B}(u1)$= $deg_{|B}(u2)$= 2. Assume for all u_i $i = \{1, 2\}$ there exists $x_i \in B$ such that $x_i \in N(u_i)$ and $N_{|A}(x_i) = \{u_i\}$. By the result we proved in Case 2c there exists $y_i \in N_{|B}(u_i)$ such that $deg_{|A}(y_i) = 3$. It is not possible that $y_1 = y_2 = y$.

Proof. By sending x_1, x_2 and y to A and $u_1, u_2 \in A$ to B we get a YES-instance for $(k+1)$-MAXIMUM CUT, a contradiction. □

If $u \in A$ with $deg_{|A}(u)$= 1 then case 1 ensures that the average degree* will be no worse than $\frac{2+2+2}{3} \geq 2$. Every $u \in A$ with $deg_{|A}(u)$= 0 has two neighbors x and y and these two neighbors cannot have both degree 1 by case 2a. If the two of them have degree* greater than or equal to 2 then the average degree* is favorable, so the worst case would occur when one of them has degree 1. By what we saw in case 2c if one of them has degree one then the other must have degree at least three and by case 2d this vertex of degree greater that 2 cannot be shared by more than one such $u \in A$. Therefore the average degree is no worse than $\frac{1+3+2}{3} \geq 2$ as required.
□

By Claim 8 we know that if the average degree* is greater than or equal to 2 then the graph G has at most k vertices. Claims 9 to 11 ensure that the average degree is in fact greater than or equal to 2. Thus, our claim that if a graph G is reduced and is a YES-instance for k-MAXIMUM CUT, but a NO-instance for $(k+1)$-MAXIMUM CUT then $|V(G)| \leq k$ holds true.

□

Now we are in a position to prove lemma 5.

Proof of Lemma 5. Assume in contradiction to the stated theorem that there exists a graph G of size $|V(G)| > k$ but has no k-MAXIMUM CUT.

Let $k' < k$ be the largest k' for which G is a YES-instance. By the Boundary Lemma 6 we know that $|V(G)| \leq k' < k$. This contradicts the assumption. □

Corollary 1 ***(Kernelization Lemma (Edges))*** *If a graph G is reduced and $|E(G)| > 2k$, then it does contain a k-MAXIMUM CUT.*

Corollary 2 ***(Boundary Lemma (Edges))*** *If a graph G is reduced and is a YES-instance for k-MAXIMUM CUT, but a NO-instance for $(k+1)$-MAXIMUM CUT then $|E(G)| \leq 2k$.*

Proof of Corollary 2. Under the same assumptions, inductive priority and witness structure as those of lemma 6 we can add the following structural claims:

Claim 1 *The number of ghost edges in the subgraph induced by A is $< k/2$.*

Proof of Claim 1. For all $u \in A$, $\sum_{u \in A} deg_{|A}(u) < \sum_{u \in A} deg_{|A}(v) = k$ as we saw in claims 5 and 7 of the lemma. By Euler's theorem the sum of the degrees in a graph is exactly twice the number of edges, thus

$$|E(\langle A \rangle)| = \frac{\sum_{u \in A} deg_{|A}(u)}{2} < \frac{k}{2}$$

□

Claim 2 *The number of ghost edges in the subgraph induced by B is $\leq k/2$.*

Proof of Claim 2. Same argument as in claim 1.
□

Claim 3 *The number of edges in the $E(G)$ is $< 2k$.*

Proof of Claim 3. The total number of edges in the graph is the sum of all *ghost edges* plus those edges between A and B. Thus,

$$|E(G)| = |E(\langle A \rangle)| + |E(\langle B \rangle)| + |E'| < k + \frac{k}{2} + \frac{k}{2}$$

□

□

Proof of Corollary 1. Assume in contradiction to the stated theorem that there exists a graph G of size $|E(G)| > 2k$ but has no k-MAXIMUM CUT.

Let $k' < k$ be the largest k' for which G is a YES-instance. By the Boundary Lemma 2 we know that $|E(G)| \leq 2k' < 2k$. This contradicts the assumption.
□

4 Kernel Algorithm and Running Time

```
Step 1.  Apply Reduction Rules 1 to
         5 until all of them are no
         longer applicable.
Step 2.  If |E(G')| > 2k or |V(G')| > k
         then answer YES and halt.
Step 3.  If |V(G')| < k/2 then try
         all possible partitions
         of the vertex set into two
         subsets A and B and count
         the number of edges in the
         cut between them.  If it
         is more than k answer YES.
Step 4.  Else if |V(G')| ≥ k/2, find a
         maximum cut in the kernel.
         If the cut is greater than
         k answer YES.
Step 5.  If the answer is YES in
         Step 3 or Step 4, answer
         YES and halt.  Else
         answer NO and halt.
```

Lemma 7 *k-MAXIMUM CUT can be solved in time $\mathcal{O}(k \cdot 2^{k/2} + n^2)$.*

Proof of Lemma 7. We will analyze each step in the algorithm separately.

Step 1. In the previous section we established a kernel for this problem with no more than k vertices and $2k$ edges. The reduction rules can all be performed in linear time on the number of vertices as they only require to check the degree of the vertices and this degree is 1 or 2 then check the degree of their neighbors.

Step 2. This step can be performed in linear time on the number of vertices/edges. It is correct by Lemma 5 and Corollary 1.

Step 3. The total number of such partitions is given by the Stirling number of the second kind, $\left\{ {k \atop 2} \right\} = 2^{n-1} - 1$. Checking how many edges exist in the bipartite subgraph induced by each partition of $V(G)$ into $A \cup B$ can always be performed in time at most $2k$ since there are only $2k$ edges in the graph.

Step 4. Applying the exact algorithm proposed by Fedin and Kulikov (Fedin& Kulikov 2002) to the kernel leads to a running time of $\mathcal{O}^*(1.414^k)$ as the running time of this algorithm is $\mathcal{O}^*(2^{m/4})$ and by Step 2 we know that the graph has at most $2k$ edges, where k is the size of the cut.

Again, checking if the cut has size greater than k can be performed in time at most $2k$ since there are only $2k$ edges in the graph.

Step 5. This last part of the algorithm can obviously be performed in constant time.

Thus, the total running time of this algorithm is the minimum of the running times of Steps 3 and 4.

$$\mathtt{min}(k \cdot 2^{k/2} + n, 2^{n-1} - 1) = \mathcal{O}(k \cdot 2^{k/2})$$

as by Step 3 we know that $n = |V(G')| < \frac{k}{2}$ □

5 Conclusions and Further Research

The *Method of Extremal Structure* has been used to tackle a variety of NP-hard optimization problems. Even though most of these problems are graph maximization ones (Prieto 2004), this method of FPT algorithm design, which combines the use of reduction rules as a preprocessing technique and combinatorial extremal arguments, has proven useful with problems as different as MINIMUM MAXIMAL MATCHING (Prieto 2004) and SET SPLITTING (Dehne & Fellows& Rosamond 2003).

In this paper we have analyzed in the context of the *Method of Extremal Structure*the k-MAXIMUM CUT problem. The problem had already an FPT algorithm running in time $\mathcal{O}(|E| + |V| + k \cdot 4^k)$ (Mahajan & Raman 1999). We improve that time by finding a kernel with at most $2k$ edges and k vertices and applying to it the exact algorithm by Fedin and Kulikov (Fedin& Kulikov 2002). Even though the improvement from (Mahajan & Raman 1999) might not seem huge, methodologically we have provided a much easier approach which can be transferred to the following problem:

Instance:	A graph $G = (V, E)$ and a positive integer k
Parameter:	k
Question:	Is there a partition of the vertex set into two subsets A and B such that the subset $E' \subseteq E$ of edges with one endpoint in A and the other in B is at least $\lceil |E|/2 \rceil + k$?

This problem was suggested by Mahajan and Raman in their paper as a more natural parameterization to k-MAXIMUM CUT since every simple graph has a cut of size at least $\lceil |E|/2 \rceil$.

References

S. Arora & D. Karger & M. Karpinski, Polynomial time approximation schemes for dense instances of NP-hard problems, *Proc. 27th Ann. ACM Symp. on Theory of Comp.*, (1995), pages 284–293

L. Cai & J. Chen & R. Downey & M. Fellows, Advice Classes of Parameterized Tractability, *Annals of Pure and Applied Logic 84*, (1997), pages 119–138

J. Chen & I. Kanj& W. Jia, Vertex cover: Further Observations and Further Improvements, *Journal of Algorithms, vol. 41*, (2001), pages 280–301

R. Downey& M. Fellows & U. Stege, Parameterized Complexity: A Framework for Systematically Confronting Computational Intractability , *AMS-DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Volume 49*, (1999), pages 49-99

R. Downey & M. Fellows, Parameterized Complexity, *Springer-Verlag* , (1999)

F. Dehne & M. Fellows & F.Rosamond, An FPT Algorithm for Set Splitting, *Proceedings 29th Workshop on Graph Theoretic Concepts in Computer Science*, (2003)

S. Fedin& A. Kulikov, A $2^{|E|/4}$-time Algorithm for MAX-CUT, *Zapiski nauchnyh seminarov POMI, No.293. English translation is to appear in Journal of Mathematical Sciences*, (2002), pages 129–138

M. Fellows & C. McCartin& F. Rosamond & U. Stege, Coordinatized Kernels and Catalytic Reductions: An Improved FPT Algorithm for Max Leaf Spanning Tree and Other Problems, *Foundations of Software Technology and Theoretical Computer Science*, (2000)

M. Garey & D. Johnson & L. Stockmeyer, Some Simplified NP-complete Graph Problems, *SIAM J. of Computing 1* , (1976), pages 237–267

M. Goemans & D. Williamson, Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming, *Journal ACM 42*, (1995), pages 1115–1145

J. Gramm & E. Hirsch & R. Niedermeier& P. Rossmanith, Worst-case upper bounds for MAX-2-SAT with application to MAX-CUT, *Discrete Applied Mathematics, 130(2)*, (2003), pages 139–155

M. Mahajan & V. Raman, Parameterizing above guaranteed values: MaxSat and MaxCut, *Journal of Algorithms Volume 31, Issue 2*, (1999), pages 335–354

E. Prieto, Systematic Kernelization in FPT Algorithm Design, *PhD Thesis. University of Newcastle, Australia. Manuscript under revision*, (2004)

E. Prieto& C. Sloper, Looking at the Stars, *Proc. International Workshop on Parameterized and Exact Computation*, (2004), pages 474–483

S. Poljak & Z. Tuza, Maximum cuts and large bipartite subgraphs, *Combinatorial optimization. American Mathematical Society*, (1995), pages 181–244

C. Papadimitriou& M. Yannakakis, Optimization, approximation, and complexity classes, *J. Comput. System Sci. 43*, (1991), pages 425–440

N. Robertson& PD. Seymour, Graph Minors XIII. The Disjoint Paths Problem, *Journal of Combinatorial Theory, Series B, Volume 63, Issue 1*, (1995), pages 65–110

B. Reed& K. Smith & A. Vetta, Finding Odd Cycle Transversals, *accepted to Operations Research Letters*, (2003)