Smoothed Analysis of Local Search for the Maximum-Cut Problem*

Michael Etscheid Heiko Röglin

Department of Computer Science
University of Bonn, Germany
{etscheid,roeglin}@cs.uni-bonn.de

Abstract

Even though local search heuristics are the method of choice in practice for many well-studied optimization problems, most of them behave poorly in the worst case. This is in particular the case for the Maximum-Cut Problem, for which local search can take an exponential number of steps to terminate and the problem of computing a local optimum is PLS-complete. To narrow the gap between theory and practice, we study local search for the Maximum-Cut Problem in the framework of smoothed analysis in which inputs are subject to a small amount of random noise. We show that the smoothed number of iterations is quasi-polynomial, i.e., it is bounded from above by a polynomial in $n^{\log n}$ and ϕ where n denotes the number of nodes and ϕ denotes the perturbation parameter. This shows that worst-case instances are fragile and it is a first step in explaining why they are rarely observed in practice.

1 Introduction

The most successful algorithms for many well-studied optimization problems are based on the ad hoc principle of local search: start with an arbitrary feasible solution and perform some kind of local improvements until none is possible anymore. For many important problems like the traveling salesman problem, clustering, and linear programming, local search is the method of choice in practice. Its success, however, lacks a theoretical account. The main reason for the considerable gap between experimental results and our theoretical understanding is that for most problems worst-case analysis predicts that local search is inefficient and can result in very bad solutions. It is not taken into account that worst-case instances are often rather contrived and rarely observed in practice. This indicates that the predominant theoretical measure of worst-case analysis is not well suited to evaluate local search. It suggests to apply more realistic performance measures that help to advance our understanding of this simple, yet powerful algorithmic technique.

We analyze the simple local search algorithm FLIP for the Maximum-Cut Problem. An instance of this problem consists of an undirected graph $G = (V, E)$ with edge weights $w : E \to \mathbb{R}$. We call a partition (V_1, V_2) of the nodes V a cut and define its weight to be the

*This research was supported by ERC Starting Grant 306465 (BeyondWorstCase).

total weight of the edges between V_1 and V_2. The FLIP algorithm starts with an arbitrary cut (V_1, V_2) and iteratively increases the weight of the cut by moving one vertex from V_1 to V_2 or vice versa, as long as such an improvement is possible. It is well-known that any locally optimal cut is a 2-approximation of a maximum cut (see, e.g., [12]). However, it is also known that the problem of finding a locally optimal cut is PLS-complete and that there are instances with cuts from which every sequence of local improvements to a local optimum has exponential length [16].

To narrow the gap between these worst-case results and experimental findings, we analyze the FLIP algorithm in the framework of smoothed analysis, which has been invented by Spielman and Teng [17] to explain the practical success of the simplex method. This model can be considered as a less pessimistic variant of worst-case analysis in which the adversarial input is subject to a small amount of random noise and it is by now a well-established alternative to worst-case analysis. In the model we consider, an adversary specifies an arbitrary graph $G = (V, E)$ with n nodes. Instead of fixing each edge weight deterministically he can only specify for each edge $e \in E$ a probability density function $f_e : [-1, 1] \to [0, \phi]$ according to which the weight $w(e)$ is chosen independently of the other edge weights. The parameter $\phi \geq 1/2$ determines how powerful the adversary is. He can, for example, choose for each edge weight an interval of length $1/\phi$ from which it is chosen uniformly at random. This shows that in the limit for $\phi \to \infty$ the adversary is as powerful as in a classical worst-case analysis, whereas the case $\phi = 1/2$ constitutes an average-case analysis with uniformly chosen edge weights. Note that the restriction to the interval $[-1, 1]$ is merely a scaling issue and no loss of generality.

For a given instance of the Maximum-Cut Problem we define the *number of steps of the FLIP algorithm* on that instance to be the largest number of local improvements the FLIP algorithm can make for any choice of the initial cut and any pivot rule determining the local improvement that is chosen if multiple are possible. Formally, this can be described as the longest path in the transition graph of the FLIP algorithm. We are interested in the *smoothed number of steps of the FLIP algorithm.* This quantity depends on the number n of nodes and the perturbation parameter ϕ and it is defined as the largest expected number of steps the adversary can achieve by his choice of the graph G and the density functions f_e. Our main result is the following theorem.

Theorem 1. *The smoothed number of steps of the FLIP algorithm is bounded from above by a polynomial in $n^{\log n}$ and ϕ.*

This result significantly improves upon the exponential worst-case running time of the FLIP algorithm. While a polynomial instead of a quasi-polynomial dependence on n would be desirable, let us point out that the theorem is very strong in the sense that it holds for all initial cuts and all pivot rules. The theorem shows that worst-case instances, on which FLIP can take an exponential number of steps, are fragile and unlikely to occur in the presence of a small amount of random noise. We see this as a first step in explaining why the worst case is rarely observed in practice because instances that arise in practice are often subject to some noise coming, e.g., from measurement errors, numerical imprecision, rounding errors, etc. The noise can also model influences that cannot be quantified exactly but for which there is no reason to believe that they are adversarial.

1.1 Related Work

Since its invention, smoothed analysis has been successfully applied in a variety of contexts. Two recent surveys [13, 18] summarize some of these results.

Local search methods for numerous optimization problems have been studied both in theory and in practice. In particular, methods based on local search are very successful for the TSP. One commonly used heuristic is k-Opt, which starts with an arbitrary tour and replaces in each local improvement k edges of the current tour by k other edges. Usually the 2-Opt heuristic needs a clearly subquadratic number of improving steps until it reaches a local optimum and the computed solution is within a few percentage points of the global optimum [11]. On the other hand it is known that even on two-dimensional Euclidean instances 2-Opt can take an exponential number of steps to reach a local optimum [9]. Englert et al. [9] analyzed the smoothed number of local improvements of 2-Opt and proved that it is polynomially bounded in the number of nodes and the perturbation parameter.

Another area in which local search methods are predominant is clustering. The k-means method is one of the most widely used clustering algorithms that is very efficient on real-world data (see, e.g., [5]), but exhibits exponential worst-case running time [19]. Arthur and Vassilvitskii initiated the smoothed analysis of the k-means method [3] that culminated in a proof that the smoothed running time of the k-means method is polynomial [2]. Arthur and Vassilvitskii [3] also showed that the smoothed running time of the ICP algorithm for minimizing the difference between two sets of points is polynomial while its worst-case running time is exponential.

Also for the Maximum-Cut Problem local search and in particular the FLIP algorithm have been studied extensively. Not only instances are known on which there exist initial cuts from which any sequence of local improvements to a local optimum has exponential length, but it is also known that the problem of computing a locally optimal cut is PLS-complete [16] even for graphs of maximum degree five [8].

The problem of computing a local optimum with respect to some local search algorithm belongs to PLS if it is possible to compute in polynomial time the objective value of a given solution, an initial feasible solution, and a better solution in the neighborhood of a given locally non-optimal solution [14]. The PLS-completeness of the Maximum-Cut Problem implies that one cannot efficiently compute a locally optimal cut, unless PLS $\subseteq$ P. Also the Maximum-Cut Problem has already been considered in the model of smoothed analysis. Elsässer and Tscheuschner [8] showed that the smoothed number of steps of the FLIP algorithm is polynomially bounded if the graph G has at most logarithmic degree. There is, however, no analysis that can handle graphs of larger degree. The Maximum-Cut Problem is not only interesting for itself, but it is also interesting because it is structurally one of the easiest PLS-complete problems. It is used as a starting point for many PLS-reductions so that the analysis of the FLIP algorithm might also shed some light on other local search algorithms.

In recent years there has been an increased interest in the class PLS due to its connection to algorithmic game theory. For many games the problem of computing pure Nash equilibria belongs to PLS and is often even PLS-complete. This is due to the fact that better- and best-response dynamics followed by agents in a game can be interpreted as variants of local search whose local optima are exactly the pure Nash equilibria. This line of research has been initiated by Fabrikant et al. [10] who showed that for network congestion games the problem of computing a pure Nash equilibrium is PLS-complete. Their proof has been simplified by Ackermann et al. [1] who gave a simple reduction from the Maximum-Cut Problem.

Even the Maximum-Cut Problem itself has been formulated as a *party affiliation game* in which agents (nodes) have to choose one of two sides and the edge weights are a measure for how much two agents like or dislike each other [6]. Our result has direct consequences for these games as well and shows that any sequence of better responses has with high probability at most quasi-polynomial length if the edge weights are subject to random noise. Since every local optimum is a 2-approximation, this implies in particular that after a quasi-polynomial number of better responses the social value is at least half of the optimal value. This is in contrast to the worst-case result of Christodoulou et al. [7] who show that there are instances with exponentially long best response sequences after which the social value is only a $1/n$-fraction of the optimal value.

Christodoulou et al. also show that if agents are allowed to play best responses in a random order then already after one round, in which every agent has been activated at least once, an 8-approximation is achieved. Awerbuch et al. [4] considered α-best response dynamics in which agents only change their strategy if this increases their payoff by a certain factor α. They prove that this α-best response dynamics reaches after a polynomial number of steps a $(2+\varepsilon)$-approximation if every player gets the chance to move at least once every polynomial number of steps. While these positive results work only for certain sequences of best responses, our positive result holds for any quasi-polynomial sequence of better responses.

2 Outline of the Analysis

As mentioned before there are already a few results on smoothed analysis of local search. There is, however, one fundamental difference between our analysis and the previous ones. The previous analyses for the 2-Opt heuristic, the ICP algorithm, the k-means method, and the Maximum-Cut Problem for graphs of logarithmic degree all rely on the observation that on smoothed inputs with high probability for every locally non-optimal solution every available local improvement results in a significant change of the potential function. Together with bounds on the minimal and maximal potential value, this implies that in expectation there cannot be too many local improvements.[1]

For the Maximum-Cut Problem the situation is different because even on smoothed inputs it is very likely that there exists a locally non-optimal cut that allows local improvements that cause an exponentially small change of the potential. It is even likely that there exist longer sequences of consecutive local improvements that all cause only a small change of the potential. The reason for this is the large number of possible cuts. While for any fixed cut it is unlikely that there exists an exponentially small local improvement, it is very likely that for one of the exponentially many cuts there exists such an improvement.

[1]To be more precise, in one case of the analysis of the k-means method, one needs to consider three consecutive local improvements to gain a significant change of the potential function. Also in the analysis of the 2-Opt algorithm two steps are considered in order to improve the degree of the polynomial.

On first glance the situation for the other problems is no different. There is, for example, also an exponential number of different TSP tours. However, for determining the amount by which the length of the tour decreases by a particular 2-Opt step, only the lengths of the four involved edges are important and there is only a polynomial number of choices for these four edges. If, on the other hand, one node changes its side in the Maximum-Cut Problem, to determine the improvement one needs to know the configuration of all nodes in its neighborhood. If the degree of the graph is at most logarithmic, there is only a polynomial number of such configurations. However, in general there is an exponential number.

As it is not sufficient anymore to analyze the potential change of a single step, our analysis is based on considering longer sequences of ℓ consecutive steps for an appropriately chosen ℓ. Let us denote by Δ the smallest improvement made by any sequence of ℓ consecutive local improvements for any initial cut. In order to analyze Δ, one could try to use a union bound over all choices for the initial cut and the sequence of ℓ steps. There are at most $2^n n^\ell$ such choices. Let us assume that an initial cut and a sequence of ℓ consecutive steps are given. Then we get a system of ℓ linear combinations of edge weights that describe the potential increases that are caused by the ℓ steps. Each such linear combination has the form $\sum_{e\in E} \lambda_e w(e)$ for some $\lambda_e \in \{-1, 0, 1\}$, where λ_e is 1 for edges joining the cut, -1 for edges leaving the cut, and 0 for the other edges. For $\varepsilon > 0$, we would like to bound the probability that all these linear combinations simultaneously take values in the interval $(0, \varepsilon]$, that is, they are all improvements by at most ε. A result from [15] implies that this probability can be bounded from above by $(\varepsilon\phi)^r$ where r denotes the rank of the set of linear combinations.

If we could argue that for any initial cut and any sequence of length ℓ the rank is at least $\alpha\ell$ for some constant $\alpha > 0$, then a union bound would yield that $\mathbf{Pr}[\Delta \le \varepsilon] \le 2^n(n\phi^\alpha\varepsilon^\alpha)^\ell$. Choosing $\ell = n$, this would even be sufficient to prove an improved version of Theorem 1 with only polynomial dependence on n. The problem is, however, that we cannot guarantee that for any sequence of length n the rank is $\Omega(n)$. Indeed there are sequences in which only few different nodes move multiple times such that the rank is only poly-logarithmic. Hence with this approach Theorem 1 cannot be proved.

In order to reduce the factor 2^n in the union bound, we make use of the following observation: Consider a node v which moves at least twice and take the linear combination L obtained by adding up the linear combinations belonging to two consecutive moves of node v. As after these two moves node v is in its original partition again, L contains only weights belonging to edges between v and other nodes that have moved an odd number of times between the two moves of node v. Therefore we only need to fix the configuration of the active nodes reducing 2^n to 2^ℓ. While this is no improvement for $\ell \ge n$, it proves valuable when we consider subsequences of smaller length. If both moves of node v yield an improvement in $(0, \varepsilon]$, then L takes a value in $(0, 2\varepsilon]$.

We call a sequence of length ℓ a *k-repeating sequence* if at least $\lceil \ell/k \rceil$ different nodes move at least twice. Given a k-repeating sequence of length ℓ, we argue that the rank of the set of linear combinations constructed in the above way is at least $\lceil \ell/(2k) \rceil$. One can then show that with high probability any k-repeating sequence yields an improvement in the order of $1/(\phi n^{\Theta(k)})$. Then we argue that any sequence of $5n$ consecutive improvements must contain a subsequence that is $\Theta(\log n)$-repeating. Together this implies Theorem 1.

Let us make one remark about the number 2^n of initial cuts that we have to consider. One might be tempted to conjecture that the factor 2^n in the union bound can be avoided if only the cut the FLIP algorithm starts with is considered instead of every possible cut. However, then our analysis would not be possible anymore. We break the sequence of steps of the FLIP algorithm into subsequences of length n each and argue that each such subsequence yields a significant improvement. Hence, not only the initial cut the FLIP algorithm starts with needs to be considered but also the initial cut of each of these subsequences.

3 Analysis

We want to show that each sequence of $5n$ consecutive steps yields a big improvement with high probability. Throughout the analysis, we need a parameter k, which we choose to be $k = \lceil 5 \log_2 n \rceil$.

Definition 2. *We call a sequence of $\ell \in \mathbb{N}$ consecutive steps k-*repeating *if at least $\lceil \ell/k \rceil$ different nodes move at least twice in that sequence.*

As already explained in the outline, for each two consecutive moves of a node we can obtain a linear combination which only contains edges to active nodes. We first show a lower bound for the rank of the set of these linear combinations.

Lemma 3. *Let S be a k-repeating sequence of length ℓ with an arbitrary starting configuration. Consider the union of sets of linear combinations obtained by adding the linear combinations of two consecutive moves of a node which moves multiple times. Then the rank of these linear combinations is at least $\lceil \ell/(2k) \rceil$.*

Proof. Let S be a k-repeating sequence of length ℓ. We construct an auxiliary graph $G' = (V, E')$ in the following way: Let $D \subseteq V$ be the set of nodes which move at least twice in the sequence S. Define $n(v)$ as the number of occurences and $\pi_v(i)$ as the position of the ith occurence of a node $v \in V$ in the sequence S. For a vertex $v \in D$ and $1 \leq i < n(v)$, let $L_v(i)$ be the sum of the linear combinations corresponding to the moves $\pi_v(i)$ and $\pi_v(i+1)$. For any $L_v(i)$, let E_v^i be the set of edges $\{v, w\}$ connecting v with all nodes w which occur in $L_v(i)$, i.e., all nodes w which move an odd number of times between $\pi_v(i)$ and $\pi_v(i+1)$. For every vertex $v \in D$ and every $1 \leq i < n(v)$, the set E_v^i is not empty because $L_i(v)$ cannot be zero in every component as it is the sum of two improving steps. Set $E_v = \bigcup_i E_v^i$ and $E' = \bigcup_{v \in D} E_v$.

Claim 1. *Let $T \subseteq D$. If every node $v \in T$ has a neighbor $u \in V \setminus T$ in the graph G', then there are indices $1 \leq i_v < n(v)$ for all $v \in T$ such that the linear combinations $\{L_v(i_v) : v \in T\}$ are linearly independent.*

Proof. Let $v \in T$, $u \in V \setminus T$ a neighbor of v in the graph G' and i_v such that $\{u, v\} \in E_v^{i_v}$. The edge $\{u, v\}$ cannot be covered by any node in $T \setminus \{v\}$. Hence it does not occur in any linear combination $L_{v'}(i)$, $v' \in T \setminus \{v\}$. As this argument holds for every $v \in T$, the linear combinations selected this way must be linearly independent. □

Claim 2. *There exists a subset $T \subseteq D$ with $|T| \geq |D|/2$ and $1 \leq i_v < n(v), v \in T$, such that the linear combinations $\{L_v(i_v) : v \in T\}$ are linearly independent.*

Proof. We first assume that G' is connected. Choose an arbitrary root $r \in D$ and calculate a BFS tree in G' rooted at r. Define V_0 and V_1 as the sets of nodes whose (unique) path to r in B contains an even or odd number of edges, respectively. Let T be the bigger of the two sets $V_0 \cap D$ and $V_1 \cap D$. Then $|T| \geq |D|/2$. Each node $v \in V_i \setminus \{r\}$, $i \in \{0, 1\}$, has a neighbor in V_{1-i}, namely the neighbor on the path to r in B. The node r has a neighbor in V_1 since $E_r^i \neq \emptyset$ for every $1 \leq i < n(r)$, i.e., V_1 cannot be empty.

If G' is not connected, perform the algorithm on every connected component which contains a node from D. As we select at least half of the nodes belonging to D from every connected component, the inequality $|T| \geq |D|/2$ still holds. Claim 1 can be used to show the existence of the $|T|$ linearly independent linear combinations. □

As S is k-repeating, there are at least $\lceil \ell/k \rceil$ nodes in D. Using Claim 2, we obtain a set T with size $|T| \geq \lceil \ell/k \rceil/2$, i.e., $|T| \geq \lceil \ell/(2k) \rceil$ since $|T|$ is integral. This yields the lemma. □

Lemma 4. *Denote by $\Delta(\ell)$ the smallest improvement made by any k-repeating sequence of length ℓ where every step increases the potential with an arbitrary starting configuration. Then for any $\varepsilon > 0$,*

$$\mathbf{Pr}\left[\Delta(\ell) \leq \varepsilon\right] \leq (2n)^{\ell} (2\phi\varepsilon)^{\lceil \ell/(2k) \rceil}.$$

Proof. We first fix a k-repeating sequence of length ℓ. As there are ℓ steps in this sequence, there are at most n^{ℓ} choices for the sequence. We will use a union bound over all these n^{ℓ} many choices and over all possible starting configurations of the nodes that are active in the sequence. This gives the additional factor of 2^{ℓ} since at most ℓ nodes can move.

For a fixed starting configuration and a fixed sequence, we consider a node v which moves at least twice and linear combinations L_1 and L_2 which correspond to two consecutive moves of node v. As after these two moves node v is in its original partition again, the sum $L = L_1 + L_2$ contains only weights belonging to edges between v and other nodes that have moved an odd number of times between the two moves of node v. In particular, L contains only weights belonging to edges between active nodes, for which we fixed the starting configuration.

Only if $L \in (0, 2\varepsilon]$, both L_1 and L_2 can takes values in $(0, \varepsilon]$. Hence it suffices to bound the probability that $L \in (0, 2\varepsilon]$. Due to Lemma 3, the rank of the set of all linear combinations constructed like L is at least $\lceil \ell/(2k) \rceil$. We can apply Lemma B.3.1 of [15] to obtain a bound of $(2\varepsilon\phi)^{\lceil \ell/(2k) \rceil}$ for the probability that all these linear combinations take values in $(0, 2\varepsilon]$. (Note: A simplified version of this lemma together with a proof can be found in the appendix.) Together with the union bound this proves the claimed bound on $\Delta(\ell)$. □

Lemma 5. *Denote by $\Delta := \min_{1 \leq \ell \leq 5n} \Delta(\ell)$ the minimum improvement by any k-repeating sequence of length at most $5n$ where every step increases the potential, starting with an arbitrary starting configuration. Then Δ is a lower bound for the improvement any sequence of $5n$ steps makes.*

This holds due to the fact that every sequence of $5n$ steps contains a k-repeating subsequence, which we will prove later. Under the assumption that this lemma holds, we can prove Theorem 1 by showing that Δ does not get too small with high probability.

Proof (Theorem 1). As every cut contains fewer than n^2 edges and every edge weight is in the interval $[-1,1]$, the weight of every cut is in $[-n^2, n^2]$. If the minimum improvement any sequence of length $5n$ makes is at least Δ, then the number of steps is bounded by $T \leq 2n^2/\Delta$, i.e., $\mathbf{Pr}[T \geq t] \leq \mathbf{Pr}[\Delta \leq 2n^2/t]$ for every $t > 0$.

To simplify the notation, let $\zeta = 4\phi n^2 (2n)^{2k} = \phi \cdot n^{O(\log n)}$. For $i \geq 2$, let $t_i = \zeta i$. By Lemma 4 we know that for every $i \geq 2$,

$$\begin{aligned}\mathbf{Pr}[T \geq t_i] &\leq \mathbf{Pr}\left[\Delta \leq \frac{2n^2}{t_i}\right] \leq \sum_{\ell=1}^{5n} \mathbf{Pr}\left[\Delta(\ell) \leq \frac{2n^2}{t_i}\right] \leq \sum_{\ell=1}^{5n} (2n)^{\ell} \left(2\phi \frac{2n^2}{\zeta i}\right)^{\lceil \ell/(2k) \rceil} \\ &\leq \sum_{\ell=1}^{5n} \left((2n)^{2k} \cdot \frac{4\phi n^2}{\zeta i}\right)^{\lceil \ell/(2k) \rceil} \leq \sum_{\ell=1}^{5n} \left(\frac{1}{i}\right)^{\lceil \ell/(2k) \rceil} \leq \sum_{\ell=0}^{\infty} 2k \left(\frac{1}{i}\right)^{\ell} - 2k \\ &= 2k\left(\frac{1}{1-1/i} - 1\right) = \frac{2k}{i-1}.\end{aligned}$$

The bound $T \leq 2^n$ is trivial as no configuration of the nodes can occur twice. Together with $t_{i+1} - t_i = \zeta$, we obtain

$$\begin{aligned}\mathbf{E}[T] &= \sum_{t=1}^{2^n} \mathbf{Pr}[T \geq t] \leq 2\zeta + \sum_{i=2}^{2^n} \sum_{t=t_i}^{t_{i+1}-1} \mathbf{Pr}[T \geq t] \leq 2\zeta + \sum_{i=2}^{2^n} \sum_{t=t_i}^{t_{i+1}-1} \mathbf{Pr}[T \geq t_i] \\ &\leq 2\zeta + \sum_{i=2}^{2^n} \zeta \cdot \frac{2k}{i-1} \leq \zeta(2 + 2k(\log(2^n)+1)) = \zeta \cdot O(n \log n) = \phi \cdot n^{O(\log n)}. \qquad \square\end{aligned}$$

What remains to show is Lemma 5, i.e., Δ is a lower bound for the minimum improvement we make in any sequence of length $5n$. Assume that it is not, then there is a sequence of length $5n$ which does not contain any k-repeating subsequence. We show that this is not possible because then more than n nodes would have to move in that sequence.

Definition 6. *We call a sequence $A_1, \ldots, A_q$ of sets a* non-k-repeating block sequence of length ℓ *if the following conditions hold.*

- *(i) For every $1 \leq i < q$, $|A_i| = k$.*
- *(ii) $1 \leq |A_q| \leq k$.*
- *(iii) $\sum_{i=1}^{q} |A_i| = \ell$.*
- *(iv) For every $i \leq j$, the number of elements that are contained in at least two sets from $A_i, \ldots, A_j$ is at most $j - i$.*

We denote by $n_k(\ell)$ the cardinality of $A_1 \cup \ldots \cup A_q$ minimized over all non-k-repeating block sequences of length ℓ.

Proof (Lemma 5). It is easy to see that any sequence S of length ℓ which does not contain a k-repeating subsequence corresponds to a non-k-repeating block sequence of length ℓ with $\lceil \ell/k \rceil$ blocks if we subdivide S into blocks of length k. It then suffices to show that there is no non-k-repeating block sequence of length $5n$ with at most n elements. In other words, $n_k(5n) > n$.

If $n \leq 3$, then there are at least two blocks as $5n > k$, but $k = \lceil 5 \log_2 n \rceil > n$ such that the first condition of Definition 6 cannot be satisfied. Therefore we can assume $n \geq 4$.

Let $q = \lceil 5n/k \rceil$ and let $A_1, \ldots, A_q$ be a non-k-repeating block sequence of length $5n$ with exactly $n_k(5n)$ different elements $x_1, \ldots, x_{n_k(5n)}$ contained in $A_1 \cup \ldots \cup A_q$. Construct an auxiliary graph H as follows: Introduce a vertex $i \in V(H)$ for each set A_i. For an element x_i, let $\rho(i)$ be the number of different sets which contain x_i and let $A_{i_1}, \ldots, A_{i_{\rho(i)}}$ ($i_1 < \ldots < i_{\rho(i)}$) be these sets. Define $P_i = \{\{i_j, i_{j+1}\} \colon 1 \leq j < \rho(i)\}$. The edges in P_i connect neighbored occurences of the element x_i. Define then $E(H) = \dot{\bigcup}_{i=1}^{n_k(5n)} P_i$ as the disjoint union of these edge sets. Note that we allow parallel edges. Define the *length* of an edge $\{v, w\}$ as $|w - v|$. Now we group the edges by their lengths: For $1 \leq i \leq \lceil \log q \rceil$, let $E_i = \{\{v, w\} \in E(H) \colon 2^{i-1} \leq |w - v| \leq 2^i\}$. Furthermore, we define *cuts* $S_j = \{\{v, w\} \in E(H) \colon v \leq j < w\}$ for every $1 \leq j < q$.

For a cut S_j and some E_i, consider an arbitrary edge $\{v, w\} \in S_j \cap E_i$: As this edge has a length of at most 2^i, we know that $j - 2^i < v, w \leq j + 2^i$. Because $A_1, \ldots, A_q$ is a non-k-repeating block sequence, there can only be at most $j + 2^i - (j - 2^i) \leq 2^{i+1}$ elements which occur multiple times in $A_{\max\{j-2^i,1\}}, \ldots, A_{\min\{j+2^i,q\}}$. By construction, every element can generate at most one edge in S_j. Hence $|S_j \cap E_i| \leq 2^{i+1}$.

On the other hand, every edge in E_i has a length of at least 2^{i-1}. Therefore every edge in E_i occurs in at least 2^{i-1} cuts. Thus we can bound the cardinality of E_i by

$$\begin{aligned} |E_i| &\leq \frac{1}{2^{i-1}} \sum_{j=1}^{q-1} |S_j \cap E_i| \leq \frac{1}{2^{i-1}} \sum_{j=1}^{q-1} 2^{i+1} \leq 4(q-1) \\ &= 4 \left(\left\lceil \frac{5n}{\lceil 5 \log n \rceil} \right\rceil - 1 \right) < 4 \frac{n}{\log n}. \end{aligned}$$

As the union of the E_i is a covering of $E(H)$, we can bound the total number of edges by

$$|E(H)| \leq \sum_{i=1}^{\lceil \log q \rceil} |E_i| < 4 \frac{n}{\log n} \lceil \log q \rceil = 4 \frac{n}{\log n} \left\lceil \log \left\lceil \frac{5n}{k} \right\rceil \right\rceil \leq 4n,$$

where the last inequality stems from

$$\left\lceil \log \left\lceil \frac{5n}{k} \right\rceil \right\rceil \leq \log \frac{5n}{k} + 1 \leq \log \frac{5n}{\lceil 5 \log n \rceil} + \log 2 \leq \log \frac{2n}{\log n} \leq \log n$$

for $n \geq 4$. This suffices to show $n_k(5n) > n$ because with every edge, the number of different elements needed decreases by exactly 1. This yields

$$n_k(5n) = \sum_{i=1}^{q} |A_i| - |E(H)| = 5n - |E(H)| > 5n - 4n = n. \qquad \square$$

4 Concluding Remarks

In this paper we showed that the smoothed running time of the FLIP algorithm for the Maximum-Cut Problem is polynomially bounded in $n^{\log n}$ and ϕ. For this purpose we introduced the analysis of $\Theta(n)$ consecutive improvement steps, whereas former analyses only looked at a constant number – normally one – of consecutive improvement steps. Although we did not try to optimize the exponent in the running time, experiments indicate that our proof ideas do not suffice to prove a polynomial bound because for $k = o(\log n)$ there seem to exist very long non-k-repeating block sequences. Instead we hope to trigger future research similar to Arthur and Vassilvitskii's paper about the k-means method [3]. They showed the non-polynomial bound $n^{O(k)}$ which inspired further research leading to a polynomial bound by Arthur et al. [2]. We also hope that local search algorithms for other PLS-hard problems can be analysed in a similar manner, especially for problems arising from algorithmic game theory.

Acknowledgements

The second author would like to thank Bodo Manthey, Alantha Newman, Shang-Hua Teng, and Berthold Vöcking for fruitful discussions about this line of research.

References

[1] Heiner Ackermann, Heiko Röglin, and Berthold Vöcking. On the impact of combinatorial structure on congestion games. *Journal of the ACM*, 55(6), 2008.

[2] David Arthur, Bodo Manthey, and Heiko Röglin. Smoothed analysis of the k-means method. *Journal of the ACM*, 58(5):19, 2011.

[3] David Arthur and Sergei Vassilvitskii. Worst-case and smoothed analysis of the icp algorithm, with an application to the k-means method. *SIAM J. Comput.*, 39(2):766–782, 2009.

[4] Baruch Awerbuch, Yossi Azar, Amir Epstein, Vahab S. Mirrokni, and Alexander Skopalik. Fast convergence to nearly optimal solutions in potential games. In *ACM Conference on Electronic Commerce (EC)*, pages 264–273, 2008.

[5] Pavel Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, USA, 2002.

[6] Anand Bhalgat, Tanmoy Chakraborty, and Sanjeev Khanna. Approximating pure nash equilibrium in cut, party affiliation, and satisfiability games. In *Proceedings of the 11th ACM Conference on Electronic Commerce (EC)*, pages 73–82, 2010.

[7] George Christodoulou, Vahab S. Mirrokni, and Anastasios Sidiropoulos. Convergence and approximation in potential games. *Theoretical Computer Science*, 438:13–27, 2012.

[8] Robert Elsässer and Tobias Tscheuschner. Settling the complexity of local max-cut (almost) completely. In *Proceedings of the 38th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 171–182, 2011.

[9] Matthias Englert, Heiko Röglin, and Berthold Vöcking. Worst case and probabilistic analysis of the 2-Opt algorithm for the TSP. *Algorithmica*, 2013. to appear.

[10] Alex Fabrikant, Christos H. Papadimitriou, and Kunal Talwar. The complexity of pure nash equilibria. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC)*, pages 604–612, 2004.

[11] David S. Johnson and Lyle A. McGeoch. The traveling salesman problem: A case study in local optimization. In E. H. L. Aarts and J. K. Lenstra, editors, *Local Search in Combinatorial Optimization*. John Wiley and Sons, 1997.

[12] Jon M. Kleinberg and Éva Tardos. *Algorithm design*. Addison-Wesley, 2006.

[13] Bodo Manthey and Heiko Röglin. Smoothed analysis: Analysis of algorithms beyond worst case. *it - Information Technology*, 53(6):280–286, 2011.

[14] Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, 48(3):498–532, 1994.

[15] Heiko Röglin. *The Complexity of Nash Equilibria, Local Optima, and Pareto-Optimal Solutions*. PhD thesis, RWTH Aachen University, 2008.

[16] Alejandro A. Schäffer and Mihalis Yannakakis. Simple local search problems that are hard to solve. *SIAM J. Comput.*, 20(1):56–87, 1991.

[17] Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM*, 51(3):385–463, 2004.

[18] Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis: An attempt to explain the behavior of algorithms in practice. *Communications of the ACM*, 52(10):76–84, 2009.

[19] Andrea Vattani. k-means requires exponentially many iterations even in the plane. *Discrete and Computational Geometry*, 45(4):596–616, 2011.

A Proof of a Simplified Version of Lemma B.3.1 of [15]

We give a proof for a simplified version of Lemma B.3.1 of [15] which suffices for our purposes in Lemma 4.

Lemma 7. *Let $X_1, \dots, X_m$ be independent real random variables and, for $1 \le i \le m$ and some $\phi \ge 1$, let $f_i \colon \mathbb{R} \to [0, \phi]$ denote the density of X_i. Furthermore, let $\lambda^1, \dots, \lambda^k \in \mathbb{Z}^m$ be linearly independent row vectors. For $i \in \{1, \dots, m\}$ and fixed $\varepsilon \ge 0$, we denote by $\mathcal{A}_i$ the event that $\lambda^i \cdot X$ takes a value in the interval $[0, \varepsilon]$, where X denotes the vector $X = (X_1, \dots, X_m)^T$. Under these assumptions,*

$$\mathbf{Pr}\left[\bigcap_{i=1}^{k} \mathcal{A}_i\right] \le (\varepsilon\phi)^k.$$

Proof. The main tool for proving the lemma is a change of variables. Instead of using the canonical basis of the m-dimensional vector space $\mathbb{R}^m$, we use the given linear combinations as basis vectors. To be more precise, the basis $\mathcal{B}$ that we use consists of two parts: it contains the vectors $\lambda^1, \dots, \lambda^k$ and it is completed by some vectors from the canonical basis $\{e^1, \dots, e^m\}$, where e^i denotes the i-th canonical row vector, i.e., $e^i_i = 1$ and $e^i_j = 0$ for $j \ne i$. Without loss of generality, we assume that $\mathcal{B} = \{\lambda^1, \dots, \lambda^k, e^{k+1}, \dots, e^m\}$.

Let $A = (\lambda^1, \dots, \lambda^k, e^{k+1}, \dots, e^m)^T$ and let $\Phi \colon \mathbb{R}^m \to \mathbb{R}^m$ be defined by $\Phi(x) = Ax$. Since $\mathcal{B}$ is a basis, the function Φ is a diffeomorphism. We define the vector $Y = (Y_1, \dots, Y_m)$ as $Y = \Phi(X)$. Let $f \colon \mathbb{R}^m \to \mathbb{R}$ denote the joint density of the entries of X, and let $g \colon \mathbb{R}^m \to \mathbb{R}$ denote the joint density of the entries of Y. We can express the joint density g as

$$g(y_1, \dots, y_m) = |\det_\partial \Phi^{-1}| \cdot f(\Phi^{-1}(y_1, \dots, y_m)),$$

where $\det_\partial$ denotes the determinant of the Jacobian matrix of Φ^{-1}.

The matrix A is invertible as $\mathcal{B}$ is a basis of $\mathbb{R}^m$. Hence, for $y \in \mathbb{R}^m$, $\Phi^{-1}(y) = A^{-1}y$ and the Jacobian matrix of Φ^{-1} equals A^{-1}. Thus, $\det_\partial \Phi^{-1} = \det A^{-1} = (\det A)^{-1}$. Since all entries of A are integers, also its determinant must be an integer, and since it is invertible, we know that $\det A \ne 0$. Hence, $|\det A| \ge 1$ and $|\det A^{-1}| \le 1$. For $y \in \mathbb{R}^m$, we decompose $\Phi^{-1}(y) \in \mathbb{R}^m$ into $\Phi^{-1}(y) = (\Phi^{-1}_1(y), \dots, \Phi^{-1}_m(y))$. Due to the independence of the random vectors $X_1, \dots, X_m$, we have $f(x_1, \dots, x_m) = f_1(x_1) \cdot \ldots \cdot f_m(x_m)$. This yields

$$\begin{aligned} g(y) \le f(\Phi^{-1}(y)) &= f_1(\Phi^{-1}_1(y)) \cdot \ldots \cdot f_m(\Phi^{-1}_m(y)) \le \phi^k \cdot f_{k+1}(\Phi^{-1}_{k+1}(y)) \cdot \ldots \cdot f_m(\Phi^{-1}_m(y)) \\ &\le \phi^k \cdot f_{k+1}(y_{k+1}) \cdot \ldots \cdot f_m(y_m) \end{aligned}$$

as $f_1, \dots, f_k$ are bounded from above by ϕ and the i-th row of A is e^i for $k < i \le m$. Hence, the probability we want to estimate can be upper bounded by

$$\begin{aligned} \mathbf{Pr}\left[\bigcap_{i=1}^{k} \mathcal{A}_i\right] &= \int_{(y_1,\dots,y_m) \in [0,\varepsilon]^k \times (-\infty,\infty)^{m-k}} g(y_1, \dots, y_m)\, d(y_1, \dots, y_m) \\ &\le (\varepsilon\phi)^k \cdot \prod_{i=k+1}^{m} \int_{-\infty}^{\infty} f_i(y_i)\, dy_i = (\varepsilon\phi)^k, \end{aligned}$$

where the last equation follows because $f_{k+1}, \dots, f_m$ are density functions. □