Smoothed Analysis of the Squared Euclidean Maximum-Cut Problem*

Michael Etscheid and Heiko Röglin

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

Abstract. It is well-known that local search heuristics for the Maximum-Cut problem can take an exponential number of steps to find a local optimum, even though they usually stabilize quickly in experiments. To explain this discrepancy we have recently analyzed the simple local search algorithm FLIP in the framework of smoothed analysis, in which inputs are subject to a small amount of random noise. We have shown that in this framework the number of iterations is quasi-polynomial, i.e., it is polynomially bounded in $n^{\log n}$ and ϕ, where n denotes the number of nodes and ϕ is a parameter of the perturbation.
In this paper we consider the special case in which the nodes are points in a d-dimensional space and the edge weights are given by the squared Euclidean distances between these points. We prove that in this case for any constant dimension d the smoothed number of iterations of FLIP is polynomially bounded in n and $1/\sigma$, where σ denotes the standard deviation of the Gaussian noise. Squared Euclidean distances are often used in clustering problems and our result can also be seen as an upper bound on the smoothed number of iterations of local search for min-sum 2-clustering.

1 Introduction

Clustering is nowadays ubiquitous in computer science. Despite intensive research on sophisticated algorithms, simple local search methods are often the most successful and versatile algorithms in practice. These algorithms are based on a simple principle: start with some feasible clustering and perform local improvements until a local optimum is found. Usually local search methods do not work well in the worst case because in most cases there are rather contrived instances on which they perform poorly.

Motivated by this striking discrepancy between theory and practice, we have recently analyzed the simple local search algorithm FLIP for the Maximum-Cut Problem in the framework of smoothed analysis, which 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 [5]. We continue this line of research and consider the special case of the Maximum-Cut Problem in which the nodes

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

are points in a d-dimensional space and the edge weights are given by the squared Euclidean distances between these points. We assume that the input is a finite set $X \subseteq \mathbb{R}^d$ of points that is to be partitioned into two parts X_1 and X_2 such that the weight $\sum_{x \in X_1} \sum_{y \in X_2} \|x-y\|^2$ becomes maximal, where $\|x-y\|$ denotes the Euclidean distance between x and y. The FLIP algorithm starts with an arbitrary cut (X_1, X_2) and iteratively increases the weight of the cut by moving one vertex from X_1 to X_2 or vice versa, as long as such an improvement is possible. Squared Euclidean distances are common in many clustering applications.

In the model we consider, an adversary specifies an arbitrary set $X \subseteq [0,1]^d$ of $n = |X|$ points. Then each point is randomly perturbed by adding a Gaussian vector of standard deviation σ to it. The parameter σ determines how powerful the adversary is. In the limit for $\sigma \to 0$ the adversary is as powerful as in a classical worst-case analysis, whereas for large σ smoothed analysis almost coincides with average-case analysis. Note that the restriction to $[0,1]^d$ is merely a scaling issue and entails 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 standard deviation σ and it is defined as the largest expected number of steps the adversary can achieve by his choice of the point set X.

Theorem 1. *For any constant dimension $d \geq 2$, the smoothed number of steps of the FLIP algorithm for squared Euclidean distances is bounded from above by a polynomial in n and $1/\sigma$. The degree of this polynomial depends linearly on d.*

This result significantly improves upon the exponential worst-case running time of the FLIP algorithm and the quasi-polynomial bound on the smoothed number of steps for general instances. The theorem shows that for squared Euclidean distances worst-case instances are fragile and unlikely to occur in the presence of a small amount of random noise.

We view Theorem 1 as a further step towards understanding the behavior of local search heuristics on semi-random inputs. Its proof is considerably different from our previous analysis for general graphs and also from the smoothed analysis of other local search heuristics in the literature. We believe that the technique used to prove Theorem 1, which we summarize in Section 2, might also be interesting for analyzing local search algorithms for other problems. In that sense, we view Theorem 1 also as a proof of concept of our new technique.

1.1 Related Work

Smoothed analysis has originally been introduced by Spielman and Teng to explain why the simplex method solves linear programs efficiently in practice

despite its exponential worst-case running time [13]. Since then it has gained a lot of attention and it has been used to analyze a wide variety of optimization problems and algorithms (see, e.g., the surveys [8, 14]).

For many optimization problems, local search heuristics are prime examples of algorithms with exponential worst-case running time that work well and efficiently in practice. Consequently, there has been a considerable amount of research on the smoothed analysis of local search. Englert et al. [4] and Manthey and Veenstra [9] have analyzed the smoothed running time of the popular 2-Opt heuristic for the traveling salesman problem. Arthur and Vassilvitskii initiated the smoothed analysis of the k-means method [2] that culminated in a proof that the smoothed running time of the k-means method is polynomial [1].

Both the worst-case and the smoothed running time of the FLIP algorithm for the Maximum-Cut Problem have been studied. It is known that the problem of computing a locally optimal cut is PLS-complete [11] even for graphs of maximum degree five [3]. This means that, unless PLS $\subseteq$ P, there is no efficient algorithm to compute partitions that are locally optimal and it also implies that there are instances on which there exist initial cuts from which any sequence of local improvements to a local optimum has exponential length. Admittedly, these lower bounds do not carry over immediately to squared Euclidean distances but there is also no sub-exponential worst-case upper bound known for this case.

Elsässer and Tscheuschner [3] were the first who analyzed the smoothed running time of the FLIP algorithm and showed that it is polynomially bounded if the graph G has at most logarithmic degree. Later we [5] analyzed the smoothed running time of the FLIP algorithm for general graphs and we proved a quasi-polynomial bound, i.e., a bound that is polynomial in $n^{\log n}$. While it would also be worthwhile to study the quality of locally optimal cuts in the framework of smoothed analysis, this line of research has not been pursued yet. It is well-known that even in the worst case any locally optimal cut is a 2-approximation of a maximum cut (see, e.g., [7]).

Schulman [12] studied the min-sum 2-clustering problem for squared Euclidean distances. In this problem, the input also consists of a finite set of points $X \subseteq \mathbb{R}^d$ and the goal is to find a partition of X into two classes X_1 and X_2 such that the sum of the edge weights inside the two classes (i.e., $\sum_{x,y\in X_1} \|x-y\|^2 + \sum_{x,y\in X_2} \|x-y\|^2$) becomes minimal. This problem is equivalent to the Maximum-Cut Problem with squared Euclidean distances (not in terms of approximation though) and hence the FLIP algorithm can also be seen as a local search algorithm for min-sum 2-clustering. Schulman gives an algorithm that solves the problem optimally in time $O(n^{d+1})$. The bound proven in Theorem 1 does not improve upon the running time of Schulman's algorithm for computing the optimal cut. However, the worst-case running time of the FLIP algorithm might be much worse than that.

Sankar, Spielman and Teng [10] analyzed the condition number of randomly perturbed matrices and proved that it is unlikely that a matrix whose entries are independent Gaussians has a large condition number. We will use this as one crucial ingredient of our analysis of the FLIP algorithm.

2 Outline of our Analysis

The analysis of the FLIP algorithm for squared Euclidean distances differs significantly from our previous analysis for general graphs and also from the smoothed analysis of other local search heuristics in the literature. Theorem 1 as well as all results in the literature are based on finding a lower bound for the improvement made by any local improvement or any sequence of consecutive local improvements of a certain length. Since $X \subseteq [0,1]^d$, the value of any cut is bounded polynomially in n with high probability. Hence, proving that in any local improvement or in any sequence of $\mathrm{poly}(n)$ consecutive local improvements the value of the cut increases by at least $\varepsilon := 1/\mathrm{poly}(n)$ with high probability suffices for proving that the expected number of local improvements is polynomially bounded. We will call an improvement of at least ε *significant* in the following.

We call a configuration (i.e., a partition of X into X_1 and X_2) *bad* if it admits an insignificant local improvement. Any fixed configuration is bad only with probability at most $\mathrm{poly}(n\varepsilon/\sigma)$. With this observation in mind one could try to use a union bound over all possible configurations to bound the probability that there exists a bad configuration. However, since there is an exponential number of configurations, this does not work. In fact, one can even prove that with high probability there do exist bad configurations. We will improve the union bound by not fixing the configuration of all points from X but only some of them, i.e., we will only make a union bound over a small subset of the points. To illustrate this, let us give two examples from the literature.

- An observation that has been exploited by Elsässer and Tscheuschner [3] is that it suffices to fix the flipping vertex and the neighborhood of this vertex in the union bound. For graphs of logarithmic maximum degree, this yields a polynomial bound on the smoothed number of local improvements.
- Another observation that has been used in a much more general form in [5] is the following: Any sequence of constantly many consecutive local improvements that starts and ends with the same vertex flipping yields a significant improvement with high probability. We showed that in order to bound the probability that there exists a sequence of this type in which all improvements are insignificant it suffices to use a union bound over all sequences of this type (there are only polynomially many because they are of constant length). One does, however, not need to specify in the union bound the configuration of the vertices that are not involved in the sequence.

The two examples above have in common that the union bound fixes only the configuration of some of the vertices (which we call *active*). The configuration of the other vertices (which we call *passive*) is not fixed in the union bound. In the examples the active points were chosen such that knowing their configuration suffices to compute the probability that the considered step or sequence of steps is bad. In our analysis we also fix only the configuration of some active vertices. The difference is that the passive vertices are not irrelevant because their configuration has a very essential impact on the improvements made by the flips in the considered sequences.

Let us go into more detail. Remember that we consider complete graphs in which each vertex is a point in $\mathbb{R}^d$ and the weights of the edges are given by the squared Euclidean distance. Our goal is to show that in this setting with high probability there is no sequence in which $9d+16$ different vertices move making only insignificant local improvements. Observe that the length of such a sequence is at most 2^{9d+16} as otherwise one configuration would repeat. We apply a union bound over all such sequences. We call all vertices that flip in the considered sequence active and apply another union bound over all configurations of the active points. With only the information about the sequence and the configuration of the active points, it is not possible to determine linear combinations of the edge weights that describe the improvements made in the sequence because the configuration of the passive points is unknown.

Assume that the passive points P are partitioned into the sets P_1 and P_2. One crucial observation for our analysis is that in the case of squared Euclidean distances it suffices to know $|P_1|$ and the value $c_P := \sum_{x\in P_1} x - \sum_{x\in P_2} x$ in order to determine the improvements made by the active points. This value is unknown if the configuration of the passive points (i.e., the partition (P_1, P_2)) is unknown and we have to assume that $c_P \in \mathbb{R}^d$ is chosen adversarially. We prove that there is a point $c_P^{\text{apx}} \in \mathbb{R}^d$ such that the first flips of the first $d+1$ active points can only all be small improvements if c_P is chosen very close to c_P^{apx} (Phase 1). The point c_P^{apx} can be computed as the solution of a system of linear equations whose coefficients are determined by the considered sequence and the active points alone. In particular, c_P^{apx} does not depend on the passive points.

In fact, if c_P is chosen to be c_P^{apx}, then the improvement of each of the steps is exactly equal to zero. The coefficients in this system of linear equations are normally distributed. Hence we can use the result of Sankar et al. [10] to argue that the condition number is not too large with high probability. From this it follows that c_P has to be chosen close to c_P^{apx} in order to guarantee that each step makes only an insignificant improvement. In order to decrease the probability that the condition number is too large, we repeat Phase 1 nine times, i.e., we consider the first flips of the first $9(d+1)$ active points.

The adversary who determines the position of c_P has no choice but to choose c_P close to c_P^{apx} if he wants to achieve that each of the first 2^{9d+9} steps in the sequence is an insignificant improvement. We then substitute c_P by c_P^{apx} in the formulas describing the improvements of steps. This results in formulas which do not depend on the passive points anymore and by our assumption that c_P is close to c_P^{apx} these formulas are good approximations for the improvements of the last seven active points (Phase 2). We use these formulas to argue that it is unlikely that all of them take values in $(0, \varepsilon)$ without having to use a union bound over the configuration of the passive points. (This approach is remotely inspired by the analysis of the k-means method where approximate centers of clusters are used [1].)

In our analysis we crucially use that the edge weights are given by squared Euclidean distances because for other distance measures the necessary information about the configuration of the passive points is not captured solely by c_P.

3 Preliminaries and Notation

In this section we state some lemmas that we will use later to prove Theorem 1 and we introduce some notation. Throughout the paper, ε denotes the threshold value between an insignificant and a significant step. Due to space limitations, all formal proofs are deferred to the full version. Proof ideas for the most important lemmas are given in this version.

Lemma 2. *Let $D_{\max} := \sigma\sqrt{2n} + 1$ and let X be a set of n Gaussian random vectors in $\mathbb{R}^d$ with mean values in $[0,1]^n$ and standard deviation σ. Then $\mathbf{Pr}\left[X \not\subseteq [-D_{\max}, D_{\max}]^d\right] \leq d/2^n$.*

Up to our proof of the main result in Section 7, we assume without further mention that $X \subseteq [-D_{\max}, D_{\max}]^d$ and $\sigma \leq 1/\sqrt{2n}$, which implies $D_{\max} \leq 2$. Furthermore, we assume $n \geq d$, which is without loss of generality because d is a constant.

Lemma 3. *The weight of any cut is between 0 and $\phi_{\max} := 16dn^2$.*

One crucial ingredient of our analysis is the following result.

Lemma 4 (Sankar, Spielman, Teng [10]). *Let $\bar{A} \in \mathbb{R}^{d\times d}$ with $\left\|\bar{A}\right\|_2 \leq \sqrt{d}$ be arbitrary. Let A be obtained from $\bar{A}$ by adding to each entry an independent Gaussian with mean 0 and standard deviation σ. Then for all $\delta \geq 1$, $\mathbf{Pr}\left[\kappa(A) \geq \delta\right] \leq \frac{14.1d(1+\sqrt{2\ln(\delta)/9d})}{\delta\sigma}$, where $\kappa(A) := \left\|A\right\|_2\left\|A^{-1}\right\|_2$ denotes the condition number of A.*

The following lemma follows from elementary probability theory.

Lemma 5. *Let $k \in \mathbb{N}$ and $\lambda_1, \ldots, \lambda_k \in \mathbb{Z}$ with $\sum_{i=1}^k \lambda_i \neq 0$. Let $u, v_1, \ldots, v_k \in \mathbb{R}^d$ and let z denote a d-dimensional Gaussian random vector with mean $\mu \in \mathbb{R}^d$ and standard deviation σ. Then for every $\tau \in \mathbb{R}$ and $\delta > 0$,*

$$\mathbf{Pr}\left[u \cdot z + \textstyle\sum_{i=1}^k \lambda_i \cdot \left\|z - v_i\right\|^2 \in [\tau, \tau+\delta]\right] \leq \frac{\sqrt{\delta}}{\sigma}.$$

For a point $z \in \mathbb{R}^d$ and a finite set $B \subseteq \mathbb{R}^d$ we write $\Phi(z, B) = \sum_{x\in B}\left\|z-x\right\|^2$. Furthermore we denote by $\mathrm{cm}(B) = \frac{1}{|B|}\sum_{x\in B} x$ the center of mass of B and we use the notation $\Psi(B) = \Phi(\mathrm{cm}(B), B)$.

4 Improvement of a Double Movement

Let us consider the improvement of a single step in which a point $z \in X$ switches sides. If we denote by $X_1^z \subseteq X$ all points on the same side as z before the movement (not including z itself) and by $X_2^z \subseteq X$ all points on the other side then the improvement of the step can be written as

$$\begin{aligned}&\Phi(z, X_1^z) - \Phi(z, X_2^z)\\ &= |X_1^z| \cdot \left\|z - \mathrm{cm}(X_1^z)\right\|^2 + \Psi(X_1^z) - \left(|X_2^z| \cdot \left\|z - \mathrm{cm}(X_2^z)\right\|^2 + \Psi(X_2^z)\right),\end{aligned} \tag{1}$$

where the equation follows from the following lemma.

Lemma 6 ([6]). *For any $z \in \mathbb{R}^d$ and any finite set $X \subseteq \mathbb{R}^d$ it holds $\Phi(z, X) = |X| \cdot \left\| z - \mathrm{cm}(X) \right\|^2 + \Psi(X)$.*

Since the occurrence of $\Psi(X_1^z)$ and $\Psi(X_2^z)$ in (1) is problematic for our analysis, we will eliminate these terms by considering two consecutive steps and adding or subtracting their respective improvements. To be more precise consider two consecutive steps in which the points y and z switch sides (in this order) and let X_1^z and X_2^z be defined as above with the only exception that y is contained in neither of them. If y and z are on different sides before they move, then it is easy to see from (1) that the terms $\Psi(X_1^z)$ and $\Psi(X_2^z)$ cancel out if one adds the improvements of the two steps. If y and z are on the same side before they move, then similarly one can see that the terms $\Psi(X_1^z)$ and $\Psi(X_2^z)$ cancel out if one subtracts the improvements of the two steps. In both cases we denote the resulting term $\xi(z)$ (it is only indexed by z and not by y because we define y to be the unique point that moves before z in the considered sequence of steps). If both steps yield an improvement in $(0, \varepsilon]$ then $\xi(z)$ lies in $[-\varepsilon, 2\varepsilon]$.

The following definition makes the reasoning above more formal. For reasons that will become clear later, we assume that the sets X_1^z and X_2^z are both partitioned into two parts, which we call passive and active.

Definition 7. *For a given sequence of steps and an arbitrary point $z \in X$ that moves during this sequence at least once but not in the first step, let $p(z)$ be the point from X that moves last before the first move of z. For any $\varepsilon > 0$, any such point $z \in X$ and the set $P \subseteq X \setminus \{z, p(z)\}$ of passive points that do not move during the considered sequence, we define the following variables and functions, where $y = p(z)$.*

- *$A^z := (X \setminus P) \setminus \{y, z\}$ is the set of active points.*
- *X_1^z is the set of points that are on the same side as z directly before the first movement of z, excluding y and z. Furthermore, let $X_2^z := (X \setminus X_1^z) \setminus \{y, z\}$ be the set of points that are on the other side, excluding y and z.*
- *We partition X_1^z and X_2^z into active and passive points: $A_1^z := A^z \cap X_1^z$, $A_2^z := A^z \cap X_2^z$, $P_1^z := P \cap X_1^z$, $P_2^z := P \cap X_2^z$.*
- *$\pi(z)$ is 1 if y and z jump in different directions; otherwise it is -1.*
- *$\xi(z)$ is defined as the improvement of the z-movement plus $\pi(z)$ times the improvement of the y-movement.*

In the next lemma we break the term $\xi(z)$ into two parts. One part, called $b(z)$, depends only on the active points and $|P_1^z|$ and $|P_2^z|$ but not on the positions of the passive points. All information about the the passive points is subsumed in the other part. It is important for our analysis that all information needed about the passive points is the value $c_P(z)$ as defined in the following lemma.

Lemma 8. *Let $y := p(z)$,*

$$\begin{aligned} b(z) :=& (|P_1^z| - |P_2^z|) \cdot (\left\|z\right\|^2 - \left\|y\right\|^2) + \Phi(z, A_1^z) - \Phi(z, A_2^z) \\ &- \Phi(y, A_1^z) + \Phi(y, A_2^z) - \begin{cases} 0 & \text{if } \pi(z) = 1, \\ 2\left\|z - y\right\|^2 & \text{if } \pi(z) = -1, \end{cases} \end{aligned}$$

and $c_P(z) := \sum_{x \in P_1^z} x - \sum_{x \in P_2^z} x$. *Then* $\xi(z) = 2c_P(z) \cdot (y - z) + b(z)$.

Lemma 9. *If the movements of* $p(z)$ *and* z *both yield an improvement of at most* $\varepsilon > 0$, *then* $|\xi(z)| \leq 2\varepsilon$.

5 Phase 1

In Phase 1 we consider a sequence of length at most 2^{d+1} in which $d+1$ different points $z^0, \ldots, z^d$ move. Then the points $z^0, \ldots, z^d$ are active and all other points are passive. Assume in this section that an arbitrary such sequence is fixed and that the initial configuration of $z^0, \ldots, z^d$ is also fixed. We will later apply a union bound over all choices for such a sequence and the initial configuration of $z^0, \ldots, z^d$. Let $\mathcal{P}_1 \in \{1, 2\}$ be the side on which z^1 is at the beginning of the sequence, and let $\mathcal{P}_2$ be the other side. We define $P := X \setminus \{z^0, \ldots, z^d\}$ and $c_P := c_P(z^1)$. We assume that the cardinalities $\left|P_1^{z^1}\right|$ and $\left|P_2^{z^1}\right|$ are fixed. We will later also apply a union bound over all choices for these cardinalities.

If any of the movements in the considered sequence yields a significant improvement then we are done. Otherwise we will prove that we have obtained enough information to deduce approximately the position of c_P. In order to see this, observe that by Lemma 8 the first movement of each z^i with $i \geq 1$ determines the following equation:

$$\xi(z^i) = 2c_P(z^i) \cdot (p(z^i) - z^i) + b(z^i).$$

Let σ^i be $+1$ if the first movement of z^i is in the same direction as the first movement of z^1, i.e., from $\mathcal{P}_1$ to $\mathcal{P}_2$, and -1 otherwise. Then $c_P(z^i) = \sigma^i \cdot c_P(z^1)$ holds for every $i \geq 1$. Hence, $\xi(z^i) = 2\sigma^i(p(z^i) - z^i) \cdot c_P + b(z^i)$. This implies that the point c_P satisfies the system $\xi = 2Mc_P + b$ of linear equations where

$$M := \begin{pmatrix} \sigma^1(p(z^1) - z^1) \\ \vdots \\ \sigma^d(p(z^d) - z^d) \end{pmatrix}, \; \xi := \begin{pmatrix} \xi(z^1) \\ \vdots \\ \xi(z^d) \end{pmatrix}, \text{ and } b := \begin{pmatrix} b(z^1) \\ \vdots \\ b(z^d) \end{pmatrix}.$$

If the matrix M is invertible (which it is with probability 1), then $c_P = M^{-1}(\xi - b)/2$. As argued above, we are interested in the case that all movements in Phase 1 yield only a small improvement of at most ε for some $\varepsilon > 0$. In this case each $\xi(z^i)$ satisfies $|\xi(z^i)| \leq 2\varepsilon$ according to Lemma 9. We consider the approximate solution c_P^{apx} of the system of linear equations assuming that each $\xi(z^i)$ is exactly zero: $c_P^{\text{apx}} = -M^{-1}b/2$. If the condition number of M is not too large and each $\xi(z^i)$ is close to zero, then c_P^{apx} is close to c_P. Note that we can calculate c_P^{apx} without uncovering the points in P or knowing their configuration because neither M nor b depends on the positions of the passive points.

Since the sequence of moves is fixed, also the matrix M is fixed. We will first show (using Lemma 4) that it is well-conditioned with high probability.

Lemma 10. *For every* $\delta \geq 1$, $\mathbf{Pr}[\kappa(M) \geq \delta] \leq \frac{72d^3}{\sigma\sqrt{\delta}}$.

In order for the distance between c_P and c_P^{apx} to be small we do not only need that the matrix M is well-conditioned but also that the norm of the right-hand side of the system of linear equations is not too small.

Lemma 11. *Let $\delta \in [0,1]$. If $\|\xi\|_\infty \leq 2\phi_{\max} \cdot \delta$ then $\mathbf{Pr}\left[\|b-\xi\|_2 \leq \delta\right] \leq \frac{12d^{3/4}n\sqrt{\delta}}{\sigma}$.*

As the quotient $\kappa(M)/\|b-\xi\|$ occurs in our analysis under a condition of the form $\|\xi\|_\infty \leq 2\delta$, we define $q_\delta := \begin{cases} \kappa(M)/\|b-\xi\| & \text{if } \|\xi\|_\infty \leq 2\delta, \\ 0 & \text{otherwise.} \end{cases}$ The second case in this definition is necessary because we will treat the event $\|\xi\|_\infty > 2\delta$ separately and do not want it to have any effect on q_δ.

Lemma 12. *Let $\varepsilon > 0$. If $\|\xi\|_\infty \leq 2\varepsilon$ then $\|c_P^{\text{apx}} - c_P\| \leq 4dn\varepsilon \cdot q_\varepsilon$.*

As we will bound the expected value of the smallest improvement of a sequence later on, we will need a bound for $\int_0^\infty \mathbf{Pr}\left[q_{\phi_{\max}/t} \geq t^c\right] \mathrm{d}t$ for some constant $c < 1$. It turns out that the bounds given in Lemma 10 and Lemma 11 are not strong enough to make this integral finite. Therefore, we repeat Phase 1 nine times with $d+1$ different points each time such that the nine repetitions are mutually independent. We consider active points from a repetition also as active in the other repetitions such that they do not account for c_P and c_P^{apx}. Note that we now need $9(d+1)$ active points $Z = \{z_i^0, \ldots, z_i^d \colon i = 1, \ldots, 9\}$ in total. Assume in the following that an arbitrary sequence of length at most $2^{9(d+1)}$ with $9(d+1)$ active points is fixed and that also the initial configuration of the active points is fixed. We will later apply a union bound over all choices. We get nine approximations c_P^{apx} for the same c_P (possibly negated) and nine different q_δ for the quotient $\frac{\kappa(M)}{\|b-\xi\|}$. Let q_δ^* be the minimum of these q_δ and let i^* be the repetition in which this minimum is obtained.

Lemma 13. $\int_0^\infty \mathbf{Pr}\left[q^*_{\phi_{\max}/t} \geq t^{7/15}\right] \mathrm{d}t \leq O\left(\frac{d^{18}n^9}{\sigma^9}\right)$.

6 Phase 2

Assume in the following that an arbitrary sequence of length at most $2^{9(d+1)+7}$ with $9(d+1)+7$ active points is fixed and that also the initial configuration of the active points is fixed. We will later apply a union bound over all choices for the sequence and the initial configuration of the active points. The longest prefix of this sequence in which at most $9(d+1)$ points move forms the nine repetitions of Phase 1, which we have analyzed in the previous section. Phase 2, which we analyze in this section, starts with the first move of point number $9(d+1)+1$. Hence, Phase 2 contains seven active points that do not move in Phase 1. Let $S = \{s^1, \ldots, s^\ell\}$ denote the set of $\ell := 7$ points that move in Phase 2 in this order (i.e., Phase 2 starts with the first movement of s^1).

We will apply the principle of deferred decisions in the following way: Except from the analysis of the error event $\mathcal{F}_1(\delta)$ (Lemma 16) we assume in this section that the positions of all $9d+9$ active points Z of Phase 1 are already uncovered. The points from S are passive in Phase 1 and hence, they belong to the set P.

This implies, in particular, that c_P (but not c_P^{apx}) depends on these points. The set of passive points changes now with every new point that gets active. We define the set of passive points during the first move of s^j as $P^j := P \setminus \{s^1, \dots, s^j\}$ and we assume that in addition to the active points from Phase 1 also the points $s^1, \dots, s^{j-1}$ are uncovered when s^j moves for the first time.

From now on, we mean with c_P^{apx} the point calculated in repetition i^* of Phase 1 and we denote by ξ the corresponding vector from repetition i^*. Define $\hat{\sigma}^j$ like σ^i in Phase 1 but for the new points moving in Phase 2, i.e., $\hat{\sigma}^j$ is 1 if the first movement of s^j is from $\mathcal{P}_1$ to $\mathcal{P}_2$ and -1 otherwise.

According to Lemma 8, we can write $\xi(s^j) = 2\hat{\sigma}^j c_{P^j} \cdot (p(s^j) - s^j) + b(s^j)$. Due to its definition, the point c_{P^j} is just c_P shifted by s^j and the already uncovered points $s^1, \dots, s^{j-1}$. Therefore, we get an approximation $c_{P^j}^{\text{apx}}$ for c_{P^j} by shifting c_P^{apx} in the same (now deterministic up to the randomness of s^j) way as c_P. As $c_{P^j}^{\text{apx}}$ and c_{P^j} are near to each other with high probability under the assumption that all steps in Phase 1 yield insignificant improvements, the "approximate improvement" $\xi^{\text{apx}}(s^j) := 2\hat{\sigma}^j c_{P^j}^{\text{apx}} \cdot (p(s^j) - s^j) + b(s^j)$ is nearly the same as $\xi(s^j)$ (Lemma 14). Thus, $\left|\xi(s^j)\right|$ can only be small if $\left|\xi^{\text{apx}}(s^j)\right|$ is small. But in the definition of $\xi^{\text{apx}}(s^j)$, the only randomness left is the position of the point s^j. Hence, we can derive a bound for the probability of an insignificant improvement by analyzing a term in which only one random point is left. We do this successively for $s^1, \dots, s^\ell$.

Lemma 14. *Let $1 \le j \le \ell$ and $0 \le \varepsilon \le 1$. If $\left\|\xi\right\|_\infty \le 2\varepsilon$ and $\left|\xi(s^j)\right| \le 2\varepsilon$, then $\left|\xi^{\text{apx}}(s^j)\right| \le 74d^{3/2}nq_\varepsilon^* \cdot \varepsilon$. (Note that $\left\|\xi\right\|_\infty \le 2\varepsilon$ does not imply anything for $\left|\xi(s^j)\right|$ as ξ is the vector from Phase 1.)*

As already mentioned, the only randomness left in the definition of $\xi^{\text{apx}}(s^j)$ is the point s^j. Hence, we can rewrite $\left|\xi(s^j)\right|$ in the following way, where C is the set of all points of the form $-c_P^{\text{apx}} + \sum_{v \in Z \cup S} \alpha_v \cdot v$ with $\alpha_v \in \{-1, 0, 1\}$.

Lemma 15. $\left|\xi^{\text{apx}}(s^j)\right| = \nu^j \cdot \left\|s^j\right\|^2 + 2s^j \cdot \hat{\sigma}^j \cdot c^j + \tau^j$, *where $\nu^j \in \mathbb{Z}$ and $\tau^j \in \mathbb{R}$ are known constants, and $c^j \in C$ has known coefficients α_v.*

We want to bound the probability that $\left|\xi^{\text{apx}}(s^j)\right|$ is close to zero. If $\nu^j = 0$, we have to make sure that $\left\|c^j\right\|$ is not too small as otherwise the variance of $2s^j \cdot \hat{\sigma}^j \cdot c^j$ is very small. We cannot guarantee this for every j, but it is unlikely to have three different j with small $\left\|c^j\right\|$.

Lemma 16. *For $\delta \ge 0$, let $\mathcal{F}_1(\delta)$ be the event that there are three distinct points $x_1, x_2, x_3 \in C$ with $\left\|x_i\right\| \le \sqrt{\delta}/2$ for $i = 1, 2, 3$. Then $\mathbf{Pr}\left[\mathcal{F}_1(\delta)\right] \le O(1) \cdot \left(\frac{\sqrt{\delta}}{\sigma}\right)^4$.*

Hence, we know that for at least four different j it is unlikely that $\left|\xi^{\text{apx}}(s^j)\right|$ is small. Now if we define $\Delta := \max_j \left|\xi(s^j)\right|$ and $\Delta^{\text{apx}} := \max_j \left|\xi^{\text{apx}}(s^j)\right|$, we are able to show that it is unlikely that Δ^{apx} and thus Δ is small.

Lemma 17. *For any $\delta \ge 0$, $\mathbf{Pr}\left[\Delta^{\text{apx}} \le \delta\right] \le O\left(\frac{\sqrt{\delta}}{\sigma}\right)^4$.*

Corollary 18. $\int_0^\infty \mathbf{Pr}\left[\Delta^{\text{apx}} \le t^{-8/15}\right] \mathrm{d}t \le O(\sigma^{-4})$.

7 Bounding the Expected Number of Steps

With Lemma 13 and Corollary 18 we have all the ingredients that we need for bounding the expected number of steps of the algorithm. We first outsource a calculation which uses the aforementioned lemmata to yield a bound for the probability of a small improvement by a fixed sequence. Then we are able to show our main result.

Lemma 19. $\int_0^\infty \mathbf{Pr}\left[\Delta \leq \frac{2\phi_{\max}}{t}\right] \mathrm{d}t \leq O\left(\frac{d^{20.5} \cdot n^{12}}{\sigma^9}\right)$.

Proof (Theorem 1). We first stick with our assumption $\sigma \leq 1/\sqrt{2n}$. Let $\mathcal{F}$ be the event that our point set X is not contained in $[-D_{\max}, D_{\max}]^d \subseteq [-2,2]^d$. Let a block be nine repetitions of Phase 1 followed by a repetition of Phase 2. Let us derive a union bound over all possible choices of blocks: There are $n^{O(d)}$ choices for the active points in Phase 1 and Phase 2. Furthermore, we need another factor n for the choice of $|P_1|$. Instead of fixing the whole sequence of steps, it suffices for our purposes to fix the configuration of the active points before every first move of a point, which results in another factor $2^{O(d^2)}$. Together this results in a factor of $2^{O(d^2)} \cdot n^{O(d)}$.

Let T be the number of blocks that are processed during the FLIP algorithm. Then

$$\begin{aligned}\mathbf{E}[T] &= \int_0^{2^n} \mathbf{Pr}[T \geq t]\,\mathrm{d}t \leq \int_0^{2^n} \mathbf{Pr}[\mathcal{F}] + 2^{O(d^2)} \cdot n^{O(d)} \cdot \mathbf{Pr}\left[\Delta \leq \frac{2\phi_{\max}}{t}\right] \mathrm{d}t \\ &\leq 2^n \cdot \frac{d}{2^n} + 2^{O(d^2)} n^{O(d)} \frac{d^{20.5} n^{12}}{\sigma^9} \leq d + \frac{2^{O(d^2)} n^{O(d)}}{\sigma^9} \leq \frac{2^{O(d^2)} n^{O(d)}}{\sigma^9}.\end{aligned}$$

As $O(d)$ different points move in a block, the length of a block is at most $2^{O(d)}$. Hence, the total number of steps is bounded by $2^{O(d)} \cdot 2^{O(d^2)} \cdot n^{O(d)} \cdot \sigma^{-9} = 2^{O(d^2)} \cdot n^{O(d)} \cdot \sigma^{-9}$.

If $\sigma > 1/\sqrt{2n}$, we create an equivalent instance by scaling down the mean values by the factor $1/(\sqrt{2n}\sigma)$ (i.e., the mean values remain in $[0,1]^n$) and setting the standard deviation to $\sigma' = 1/\sqrt{2n} < \sigma$. As these instances are equivalent, we obtain the same expected number of iterations and thus also a bound of $2^{O(d^2)} \cdot n^{O(d)} \cdot (\sqrt{2n})^9 \leq 2^{O(d^2)} \cdot n^{O(d)} \cdot \sigma^{-9}$. □

8 Concluding Remarks

We proved the first polynomial upper bound on the smoothed number of steps of the FLIP algorithm for the Maximum-Cut problem. Our upper bound applies only to squared Euclidean distances because it uses essentially the identity given in Lemma 6, which is special to squared Euclidean distances. It might be possible to extend our analysis to Bregman divergences because these also satisfy Lemma 6. An immediate extension to general graphs does not seem to be possible and it is still a very interesting open question if the result from [5] for general graphs can be improved.

Our result gives only a polynomial smoothed running time if the dimension is constant because the degree of the polynomial grows linearly with d. We think that it is conceivable to improve the smoothed running time to $2^{O(d)}$ times a polynomial in n and $1/\sigma$ whose degree is independent of d by a more careful analysis of the condition number in Phase 1 that does not use the result by Sankar et al. [10] as a black box. Generally, we hope that our work triggers further improvements like, e.g., the first smoothed analysis of the k-means method by Arthur and Vassilvitskii [2], which only gave a polynomial bound for constant k.

A version of the k-means method that works rather well in experiments is Hartigan's method [15]. Telgarsky and Vattani conjecture that the smoothed running time of this algorithm is polynomial [15]. However, so far this conjecture could not be proven and it seems rather challenging. As Hartigan's method has some similarities with the FLIP algorithm for the Maximum-Cut problem for squared Euclidean distances, we believe that our new proof technique might also be helpful for proving Telgarsky and Vattani's conjecture.

References

1. D. Arthur, B. Manthey, and H. Röglin. Smoothed analysis of the k-means method. *JACM*, 58(5):19, 2011.
2. D. Arthur and S. Vassilvitskii. Worst-case and smoothed analysis of the ICP algorithm. *SICOMP*, 39(2):766–782, 2009.
3. R. Elsässer and T. Tscheuschner. Settling the complexity of local max-cut (almost) completely. In *Proc. of 38th ICALP*, pages 171–182, 2011.
4. M. Englert, H. Röglin, and B. Vöcking. Worst case and prob. analysis of the 2-Opt algorithm for the TSP. *Algorithmica*, 68(1):190–264, 2014.
5. M. Etscheid and H. Röglin. Smoothed analysis of local search for the maximum-cut problem. In *Proc. of 25th SODA*, pages 882–889, 2014.
6. T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Wu. A local search appr. algo. for k-means clustering. *Comput. Geom.*, 28:89–112, 2004.
7. J. Kleinberg and É. Tardos. *Algorithm design*. Addison-Wesley, 2006.
8. B. Manthey and H. Röglin. Smoothed analysis: Analysis of algorithms beyond worst case. *it - Information Technology*, 53(6):280–286, 2011.
9. B. Manthey and R. Veenstra. Smoothed analysis of the 2-opt heuristic for the TSP. In *Proc. of 24th ISAAC*, pages 579–589, 2013.
10. A. Sankar, D. Spielman, and S.-H. Teng. Smoothed analysis of the condition numb. and growth factors of matrices. *SIMAX*, 28(2):446–476, 2006.
11. A. Schäffer and M. Yannakakis. Simple local search problems that are hard to solve. *SICOMP*, 20(1):56–87, 1991.
12. L. Schulman. Clustering for edge-cost minimization. In *Proc. of 32nd STOC*, pages 547–555, 2000.
13. D. Spielman and S.-H. Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *JACM*, 51(3):385–463, 2004.
14. D. Spielman and S.-H. Teng. Smoothed analysis: An attempt to explain the behavior of algorithms in practice. *CACM*, 52(10):76–84, 2009.
15. M. Telgarsky and A. Vattani. Hartigan's method: k-means clustering without voronoi. In *Proc. of 13th AISTATS*, pages 820–827, 2010.