Algorithms Beyond the Worst Case

Bodo Manthey

University of Twente, b.manthey@utwente.nl

Abstract

These notes describe some of the material of the course "Algorithms Beyond the Worst Case", which is part of the Mastermath and DIAMANT programs.

Last modified: May 25, 2016.

Contents

1 Some probability theory

Consult any book on probability theory and/or randomized analysis of algorithms for more background. For instance, the book on probability theory by Durrett [8] is even available online, and the book by Mitzenmacher and Upfal on probability and algorithms [15] gives a decent background how algorithms can be analyzed on random inputs.

1.1 Gaussian distributions

The d-dimensional Gaussian distribution with mean $\mu \in \mathbb{R}^d$ and standard deviation σ has density f with

$$f(z) = \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^d \cdot \exp\left(-\frac{\|z-\mu\|^2}{2\sigma^2}\right).$$

We denote this distribution by $N_d(\mu, \sigma^2)$. Gaussian distribution have several nice properties:

- The density is rotation-symmetric around μ.
- Drawing d independent 1-dimensional Gaussians with standard deviation σ in orthogonal directions yields the same distribution.
- Assume that $X \sim N_d(\mu, \sigma^2)$ for some $\mu \in \mathbb{R}^d$. Let $a_1, \ldots, a_d \in \mathbb{R}^d$ be orthogonal vectors. Then $a_1^{\mathrm{T}}X, a_2^{\mathrm{T}}X, \ldots, a_d^{\mathrm{T}}X$ are all independent with $a_i^{\mathrm{T}}X \sim N_1(a_i^{\mathrm{T}}\mu, \|a_i\|^2\sigma^2)$.
- Sums of Gaussian random variables are Gaussian distributed: if $X \sim N_d(\mu, \sigma^2)$ and $Y \sim N_d(\nu, \tau^2)$, then $X + Y \sim N_d(\mu + \nu, \sigma^2 + \tau^2)$.
- The density is bounded from above by $\left(\frac{1}{\sigma\sqrt{2\pi}}\right)^d \leq \left(\frac{1}{2\sigma}\right)^d$

The *chi distribution* is the distribution of the norm of a d-dimensional vector drawn according to a Gaussian distribution of mean 0 and standard deviation 1. Its density is given by

$$x \mapsto \frac{2^{1-\frac{d}{2}} \cdot x^{d-1} \cdot \exp(-x^2/2)}{\Gamma(d/2)},$$

where Γ denotes the gamma function.

1.2 Interval lemmas

Lemma 1.1 (interval lemma). *Let $\phi > 0$, and let X be a real-valued random variable with density $f : \mathbb{R} \to [0, \phi]$. (This means that f is upper-bounded by ϕ.) Let $t \in \mathbb{R}$, and let $\varepsilon > 0$. Then*

$$\mathbb{P}\big(X \in (t, t+\varepsilon]\big) \leq \varepsilon\phi.$$

Proof. $\mathbb{P}\big(X \in (t, t+\varepsilon]\big) = \int_t^{t+\varepsilon} f(x)\,\mathrm{d}x \leq \varepsilon\phi$. □

Corollary 1.2 (interval lemma for Gaussians). *Let X be distributed according to a Gaussian distribution with arbitrary mean and standard deviation $\sigma > 0$. Let $t \in \mathbb{R}$, and let $\varepsilon > 0$. Then*

$$\mathbb{P}\big(X \in (t, t+\varepsilon]\big) \leq \frac{\varepsilon}{2\sigma}.$$

The interval lemma can be generalized to higher-dimensional spaces.

Lemma 1.3. *Let $I \subseteq \mathbb{R}^d$, and let $\mathrm{vol}(I)$ denote the volume of I. Let X be a random variable with density $f : \mathbb{R}^d \to [0, \phi]$ for some $\phi > 0$. Then $\mathbb{P}(X \in I) \leq \phi \cdot \mathrm{vol}(I)$.*

Corollary 1.4. *Let $I \subseteq \mathbb{R}^d$, and let $X \sim N_d(\mu, \sigma^2)$ for some $\mu \in \mathbb{R}^d$ and $\sigma > 0$. Then $\mathbb{P}(X \in I) \leq \frac{\mathrm{vol}(I)}{2^d \sigma^d}$.*

A consequence of Corollary 1.4 is that the probability that Gaussian assumes a value in a hyperball of radius r is upper-bounded by $(r/\sigma)^d$.

Corollary 1.5. *Let $X \sim N_d(\mu, \sigma^2)$, and let $H \subseteq \mathbb{R}^d$ be a hyperplane. Then*

$$\mathbb{P}\big(\mathrm{dist}(X, H) \leq \varepsilon\big) \leq \frac{\varepsilon}{\sigma}.$$

Proof. Let v be orthogonal to H with $\|v\| = 1$. Then $\mathrm{dist}(X, H) \leq \varepsilon$ only if the 1-dimensional Gaussian random variable $v^{\mathrm{T}}X$ falls into an interval of size 2ε. □

1.3 Some useful (in)equalities

Lemma 1.6 (union bound, Boole's inequality). *Let $E_1, \ldots, E_m$ be m events with $\mathbb{P}(E_i) \leq p_i$. Then*

$$\mathbb{P}\big(\textit{at least one } E_i \textit{ occurs}\big) \leq \sum_{i=1}^{m} p_i.$$

Lemma 1.7 (Markov's inequality). *Let X be a random variable that assumes only non-negative numbers, and assume that the expected value $\mathbb{E}\,X$ of X exists. Then*

$$\mathbb{P}(X \geq t) \leq \frac{\mathbb{E}(X)}{t}.$$

Lemma 1.8 (Jensen's inequality). *Let $f : \mathbb{R} \to \mathbb{R}$ be a convex function, and let X be a real-valued random variable. Then (assuming that both expected values exist)*

$$f\big(\mathbb{E}(X)\big) \leq \mathbb{E}\big(f(X)\big).$$

Lemma 1.9 (tail bound for Gaussians). *Let $X \sim N_1(\mu, \sigma^2)$ for some $\mu \in \mathbb{R}$ and $\sigma > 0$. Then*

$$\mathbb{P}(X \geq \mu + \sigma t) = \mathbb{P}(X \leq \mu - \sigma t) \leq \frac{1}{t\sqrt{2\pi}} \cdot \exp\left(-\frac{t^2}{2}\right).$$

The following lemma shows that a d-dimensional Gaussian vector centered at the origin and with standard deviation 1 is unlikely to have norm more than $O(\sqrt{d})$. In particular, we can choose $\varepsilon = 3$, which simplifies the bound below to

$$\mathbb{P}\left(\|X\| \geq 2 \cdot \sqrt{d}\right) \leq 2^{-d}.$$

Lemma 1.10 (tail bound for d-dimensional Gaussians). *Let $X \sim N_d(0, 1)$. Then, for every $\varepsilon > 0$, we have*

$$\mathbb{P}\left(\|X\|^2 \geq (1+\varepsilon) \cdot d\right) \leq \left(\sqrt{1+\varepsilon} \cdot \exp(-\varepsilon/2)\right)^d.$$

Lemma 1.11 (Chernoff bound). *Let $X_1, \ldots, X_n$ be independent random variables that assume only values in $\{0, 1\}$. Let $\mathbb{P}(X_i = 1) = p_i$, let $X = \sum_{i=1}^n X_i$, and let $\mathbb{E}(X) = \sum_{i=1}^n p_i = \mu$. Then*

$$\mathbb{P}\big(X > \mu + a\big) < \exp\left(\frac{-2a^2}{n}\right)$$

for all $a > 0$. By symmetry, we have the same bound for $\mathbb{P}\big(X < \mu - a\big)$.

There is a huge variety of different Chernoff bounds. For most applications, it does not matter much which one you use. We only state one version here for the sake of conciseness.

Lemma 1.12. *Let X be a random variable that assumes only values in $\mathbb{N}$ and has finite expected value. Then*

$$\mathbb{E}(X) = \sum_{t=1}^{\infty} \mathbb{P}(X \geq t).$$

2 Smoothed analysis

2.1 Introduction

Smoothed analysis is a paradigm to analyze the performance of algorithms with a huge discrepancy between worst-case performance and observed performance. The performance is measured by instances that are neither adversarial nor completely random. In this way, smoothed analysis often yields a better prediction of the practical performance than worst-case analysis. On the other hand, the instances considered are not completely random as in

average-case analysis. Completely random instances have often very specific properties, which are exploited in the average-case analysis, but not shared by typical instances coming from applications.

Smoothed analysis has been introduced by Spielman and Teng to explain the practical performance of the simplex method [17]. Since then, it has been applied successfully to a variety of algorithms [12, 13, 18]. We see several examples in this lecture.

2.2 Two models of smoothed analysis

Before actually analyzing algorithms, let us quickly describe the two "standard models" of smoothed analysis. For the algorithms we analyze, we will always briefly describe the probabilistic input model that we use.

Two-step model or classical model. In the classical model, numbers of the given instances are perturbed by Gaussian distributions. This means that an adversary specifies an arbitrary instance. And then the numbers or coefficients (or some of them) are perturbed independently by adding Gaussian-distributed random variables of mean 0 and standard deviation $\sigma > 0$ to them. The smoothed performance is then the maximum expected performance, where the maximum is taken over the adversarial choices, and the expected value is taken over the random perturbation of the adversarial instance. The parameter σ controls how strongly we perturb the instances. The smaller σ, then more powerful the adversary, i.e., the closer is the perturbed instance to the original, adversarial, unperturbed instance.

Instead of Gaussian perturbations, one can also use other probability distributions. However, it turns out that Gaussian distributions have nice properties that helps in the analysis.

One-step model. The one-step model, which has been introduced by Beier and Vöcking [5], allows more general input distributions. Instead of numbers or coefficients, the adversary specifies probability density functions for all numbers that should be perturbed. Then all numbers are drawn independently according to their respective density function.

We have a perturbation parameter ϕ that controls the power of the adversary: the adversary can choose any probability density function that is upper-bounded by ϕ. This means that the larger ϕ, the more powerful the adversary. Roughly speaking, ϕ corresponds to $1/\sigma$ in the two-step model.

Again, we measure the maximum expected performance, where the maximum is taken over all choice of the adversary and the expectation is taken over drawing the numbers according to their density functions.

3 2-opt heuristic for the TSP

The 2-opt heuristic is a very simple local search heuristic for the traveling salesman problem (TSP). Starting with an initial TSP tour, it performs that following operation until it has converged to a local optimum: let T be the current TSP tour, and let $\{y_1, y_2\}$ and $\{y_3, y_4\}$ be two edges of T such that y_1, y_2, y_3, y_4 appear in this order in T. Then we replace these two edges by $\{y_1, y_3\}$ and $\{y_2, y_4\}$ if this decreases the tour length.

In practice, the 2-opt heuristic converges quickly to a close-to-optimal solution. However, there are worst-case examples showing that the convergence can take exponentially long and that the worst-case approximation ratio is poor.

Theorem 3.1 (Englert et al. [9]). *For every $n \in \mathbb{N}$, there exists a set in $\mathbb{R}^2$ consisting of $8n$ point on which 2-opt can make $2^{n+3} - 14$ steps.*

The approximation ratio depends heavily on the initial tour. The following statement is a worst-case statement in two senses: first, it assumes a worst-case point set. Second, it assumes that the initialization is as bad as possible. For the following theorem and Section 3.3, we need the following definitions. We call a TSP tour through a point set 2-optimal if it cannot be shortened by a 2-opt step. For a point set X, we denote by $\mathrm{WLO}(X)$ the length of the longest 2-optimal tour through X. We denote by $\mathrm{TSP}(X)$ the length of the shortest TSP tour.

Theorem 3.2 (Chandra et al. [7]). *Fix any d. Let $X \in [0,1]^d$ be any set of n points. Then*

$$\frac{\mathrm{WLO}(X)}{\mathrm{TSP}(X)} = O(\log n).$$

There exists a constant $c > 0$ such that for infinitely many n, there exist a set $X \subseteq \mathbb{R}^2$ of n points such that

$$\frac{\mathrm{WLO}(X)}{\mathrm{TSP}(X)} \geq c \cdot \frac{\log n}{\log \log n}.$$

If we examine in particular the worst-case instances for the running-time closely, we observe that the location of the points must be chosen very carefully and that a slight random perturbation destroys it.

To do a smoothed analysis in the following, we consider Euclidean instances, where the nodes are from $\mathbb{R}^d$. The dimension d is considered to be a constant. This means in particular that constants depending only on d are sometimes hidden in O or Ω.

More specifically, an adversary specifies a set $X = \{x_1, \ldots, x_n\} \subseteq [0,1]^d$ of n points. Then we obtain $Y = \{y_1, \ldots, y_n\} \subseteq \mathbb{R}^d$ by adding independent Gaussian random variables of standard deviation σ to $x_1, \ldots, x_n$. This

means that $y_i \sim N_d(x_i, \sigma^2)$, where $N_d(x_i, \sigma^2)$ denotes a d-dimensional Gaussian distribution with mean x_i and variation σ^2. Note that Y is not necessarily a subset of the unit hypercube $[0,1]^d$. We denote by $g_i = y_i - x_i$ the independent Gaussian random variables with mean 0 and standard deviation σ that model the perturbation.

3.1 Smoothed analysis of the running-time

Our smoothed analysis of the running-time is based on ideas by Englert et al. [9]. We follow the proof of Manthey and Veenstra [14].

The main idea to analyze the number of iterations until 2-opt converges is to use the current tour length as a potential function. If we can prove

- an upper bound of L_{init} for the length of the initial tour and
- a lower bound of $I_{\min}$ for the minimal improvement obtained by any possible iteration (called 2-opt step) of the 2-opt heuristic,

then we obtain an upper bound of $L_{\text{init}}/I_{\min}$ for the number of iteration that 2-opt needs until convergence.

Note that this approach is still quite pessimistic: First, it is unlikely that we always make the minimal possible improvement. It is more likely that some iterations cause a much larger improvement. Second, we might often face the situation that there are several 2-opt steps possible. In this case, the approach above assumes that we make the worst possible choice.

For simplicity, we make the following assumptions:

- We measure distances between points by the squared Euclidean distance between them. Only at the end we give a result for Euclidean distances (without squaring), but without proof details.
- We assume that $\sigma \leq \frac{1}{2\sqrt{n \ln n}}$. This is not a severe restriction as already such small σ suffice to prove a polynomial bound for the smoothed number of iterations. We note that the analysis can be done also for larger values of σ, but this does not provide too much additional insight and only makes the analysis slightly more technical.
- We assume that $d = 2$. We give a general result for larger values of d at the end. Furthermore, we assume that $n \geq 3$.

Let us first bound L_{init}.

Lemma 3.3. *We have $L_{\text{init}} \leq 18n$ with a probability of at least $1 - \frac{1}{n!}$.*

Proof sketch. If $Y \subseteq [-1,2]^d$, then the longest distance between any two points in Y (measured in squared Euclidean distance) is at most 18. Thus, any tour has a length of at most $18n$ in this case.

If $Y \not\subseteq [-1,2]^2$, then there exists an i such that $\|g_i\|_\infty \geq 1$. Thus, there must exist an $i \in \{1,\dots,n\}$ and a direction $j \in \{1,\dots,j\}$ such that the absolute value of the j-th entry of g_i is at least 1. We use Lemma 1.9 with $\sigma \leq \frac{1}{2\sqrt{n \ln n}}$ and $t = 1/\sigma$. This yields that the probability for a single entry to be of absolute value at least 1 is bounded from above by

$$\frac{1}{\sqrt{2\pi n \ln n}} \cdot \exp\bigl(-2n \ln n\bigr) \leq n^{-2n} \leq (n!)^{-2}.$$

A union bound over the choices of i and j and the poor bound $2n \leq n!$ yields the lemma. □

Let $\Delta_{\min}$ be the smallest improvement by any possible 2-opt step.

In the rest of this section, let $\Delta_{a,b}(c) = \|c-a\|^2 - \|c-b\|^2$. For the analysis of $\Delta_{\min}$, the following lemma is useful.

Lemma 3.4. *Let $a, b \in \mathbb{R}^2$ with $a \neq b$, and let $c \in \mathbb{R}^2$ be drawn according to a Gaussian distribution with standard deviation σ. Let $I \subseteq \mathbb{R}$ be an interval of length ε. Then*

$$\mathbb{P}\bigl(\Delta_{a,b}(c) \in I\bigr) \leq \frac{\varepsilon}{4\sigma \cdot \|a-b\|_2}.$$

Proof. Since Gaussian distributions are rotationally symmetric, we can assume without loss of generality that $a = (0,\dots,0)$ and $b = (\delta, 0, \dots, 0)$ with $\delta = \|a-b\|_2$. Let $c = (c_1, c_2)^{\mathrm{T}}$. Then $\Delta_{a,b}(c) = c_1^2 - (c_1-\delta)^2 = 2c_1\delta + \delta^2$. Thus, $\Delta_{a,b}(c) \in I$ if and only if c_1 falls into an interval of length $\frac{\varepsilon}{2\delta}$. Since c_1 is a 1-dimensional Gaussian random variable with a standard deviation of σ, the lemma follows from Corollary 1.2. □

Lemma 3.5. $\mathbb{P}(\Delta_{\min} \leq \varepsilon) = O\bigl(\frac{n^4\varepsilon}{\sigma^2}\bigr)$.

Proof. Consider any four points $y_1, y_2, y_3, y_4 \in Y$ and the 2-opt step, where the two edges $\{y_1, y_2\}$ and $\{y_3, y_4\}$ are replaced by $\{y_1, y_3\}$ and $\{y_2, y_4\}$. We prove that the probability that this is a 2-opt step that yields an improvement of at most ε is bounded by $O(\varepsilon/\sigma^2)$. Then the lemma follows by a union bound over the choices of the four points y_1, y_2, y_3, y_4.

The improvement caused by the 2-opt step described above is equal to $\Delta_{y_2,y_3}(y_1) - \Delta_{y_2,y_3}(y_4)$. If we fix y_2, y_3, and y_4 arbitrarily, then the improvement is only in the interval $(0,\varepsilon]$ if $\Delta_{y_2,y_3}(y_1)$ falls into an interval of size ε. The probability that this happens is bounded from above by $\frac{\varepsilon}{4\sigma\cdot\|y_2-y_3\|}$. according to Lemma 3.4.

Let f be the probability density function of $\delta = \|y_2 - y_3\|$. Then the probability that the 2-opt step considered yields an improvement of at most ε is bounded from above by

$$\int_{\delta=0}^{\infty} \frac{\varepsilon}{4\sigma\delta} \cdot f(\delta)\,\mathrm{d}\delta.$$

Now we observe that the distribution of $1/\delta$ is stochastically dominated by $1/X$, where X is chi-distributed. From this observation, we obtain that we can replace f by the density function of the chi distribution to get an upper bound for the probability that we have an improvement of at most ε. Solving the resulting integral □

Theorem 3.6. *Let $Y \subseteq \mathbb{R}^2$ be obtained as described above, and let $\sigma \leq 1/(2\sqrt{n \ln n})$. Then the maximum number of iterations that the 2-opt heuristic needs to compute a locally optimal TSP tour with respect to squared Euclidean distances is bounded from above by $O(n^6 \log n/\sigma)$.*

Proof. If 2-opt runs for at least t steps, then we must have $L_{\text{init}} \geq 18n$ or $\Delta_{\min} \leq 18n/t$. The probability that any of these events happens is bounded from above by

$$\frac{1}{n!} + O\left(\frac{n^5}{\sigma t}\right).$$

Since no TSP tour shows up twice in any run of 2-opt, we know that the number of iterations is upper-bounded by $n!$. Let T be the random variable that is the maximum possible number of iterations that 2-opt can need on the (random) point set Y. Then

$$\mathbb{E}(T) = \sum_{t=1}^{n!} \mathbb{P}(T \geq t) \leq \sum_{t=1}^{n!} \frac{1}{n!} + O\left(\frac{n^5}{\sigma t}\right) = O\left(\frac{n^6 \log n}{\sigma}\right).$$

□

3.2 Linked pairs of 2-opt steps

The idea of analyzing linked pairs of 2-opt steps is from Englert et al. [9]. A linked pair of 2-opt steps is a pair of two 2-opt steps that share an edge that is inserted in one and removed in the other 2-opt step. Depending on the number of nodes involved, we distinguish 2-opt steps of types 0, 1a, 1b, and 2. One of the two 2-opt steps will be replacing the edges $\{y_1, y_2\}$ and $\{y_3, y_4\}$ by $\{y_1, y_3\}, \{y_2, y_4\}$. The other 2-opt step is as follows:

Type 0: $\{y_1, y_3\}, \{y_5, y_6\}$ are replaced by $\{y_1, y_5\}, \{y_3, y_6\}$. In total, six different nodes are involved.

Type 1a: $\{y_1, y_3\}, \{y_2, y_5\}$ are replaced by $\{y_1, y_5\}, \{y_2, y_3\}$. In total, five different nodes are involved.

Type 1b: $\{y_1, y_3\}, \{y_2, y_5\}$ are replaced by $\{y_1, y_2\}, \{y_3, y_5\}$. In total, five different nodes are involved.

Type 2: $\{y_1, y_3\}, \{y_2, y_4\}$ are replaced by $\{y_1, y_4\}, \{y_2, y_3\}$. In total, only four different nodes are involved.

Because in type 2 pairs, only four nodes are involved, this type is difficult to analyze. However, in any sufficiently long sequence of 2-opt steps, we find enough disjoint pairs of linked 2-opt steps of types 0, 1a, and 1b according to the following lemma. Roughly speaking, in every sequence of t consecutive 2-opt steps with $t \geq cn^2$ for sufficiently large c, we find at least $c't$ disjoint pairs of linked 2-opt steps for some constant $c' > 0$.

Lemma 3.7. *Every sequence of t consecutive 2-opt steps contains $\geq \frac{t}{6} - \frac{7n(n-1)}{24}$ disjoint pairs of linked 2-opt steps of types 0, 1a, or 1b.*

Let $\Delta_{\min}^{\text{linked}}$ be the smallest improvement by any possible pair of linked 2-opt steps of type 0, 1a, or 1b.

Lemma 3.8. $\mathbb{P}(\Delta_{\min}^{\textit{linked}} \leq \varepsilon) = O\big(\frac{n^6\varepsilon^2}{\sigma^4}\big)$.

Theorem 3.9. *Let $Y \subseteq \mathbb{R}^2$ be obtained as described above, and let $\sigma \leq 1/(2\sqrt{n \ln n})$. Then the maximum number of iterations that the 2-opt heuristic needs to compute a locally optimal TSP tour with respect to squared Euclidean distances is bounded from above by $O(n^4/\sigma^2)$.*

Proof. Let T be the random variable that is the maximum possible number of iterations that 2-opt can need on the (random) point set Y. By Lemma 3.7, there exist constants $c, c' > 0$ such that every sequence of at least t iterations contains at least $c't$ disjoint pairs of linked 2-opt steps.

Then $T \geq t$ only if $t \leq cn^2$ or if $L_{\text{init}} \geq 18n$ or if there is a pair of linked 2-opt steps of type 0, 1a, or 1b that yields an improvement of at most $\frac{18n}{c't}$. Thus, there exist constants $c'', \hat{c} > 0$ such that, by Lemma 3.8, we have

$$\begin{aligned}
\mathbb{E}(T) &\leq cn^2 + \sum_{t \geq cn^2} \mathbb{P}(T \geq t) \\
&\leq cn^2 + \sum_{t \geq cn^2} \min\left\{1, c'' \cdot \frac{n^8}{t^2\sigma^4}\right\} \\
&\leq \hat{c} \cdot \frac{n^4}{\sigma^2} + \sum_{t \geq \hat{c}n^4/\sigma^2} c'' \cdot \frac{n^8}{t^2\sigma^4} = O\left(\frac{n^4}{\sigma^2}\right)
\end{aligned}$$

□

A smoothed analysis of the number of iterations is also possible if we make the (more natural) choice of measuring distances between points by their Euclidean distance. The analysis becomes more technical. We state the following theorem without a proof.

Theorem 3.10. *Let $Y \subseteq \mathbb{R}^2$ be obtained as described above, and let $\sigma \leq 1/(2\sqrt{n \ln n})$. Then the maximum number of iterations that the 2-opt heuristic needs to compute a locally optimal TSP tour with respect to Euclidean distances is bounded from above by $O(n^4/\sigma^4)$.*

3.3 Smoothed analysis of the approximation ratio

In this section, we use the (standard) Euclidean distances to measure the tour length. We call a TSP tour through a point set 2-optimal if it cannot be shortened by a 2-opt step. For a point set Y, we denote by $\mathrm{WLO}(Y)$ the length of the longest 2-optimal tour through Y. We denote by $\mathrm{TSP}(Y)$ the length of the shortest TSP tour. The idea of lower-bounded $\mathrm{TSP}(Y)$ and upper-bounding $\mathrm{WLO}(Y)$ is by Englert et al. [9]. They have used it to analyze the approximation ratio in a more general probabilistic model. We follow the proof of Künnemann and Manthey [11] for Gaussian noise.

The worst-case approximation ratio of $O(\log n)$ for the 2-opt heuristic has been proved by Chandra et al. [7].

Our goal here is to prove a smoothed approximation ratio of $O(1/\sigma)$. This means that $\mathbb{E}(\mathrm{WLO}(Y)/\mathrm{TSP}(Y)) = O(1/\sigma)$. The idea to prove this is as follows:

- Prove that $\mathrm{TSP}(Y) = \Omega(\sigma \cdot n^{1-\frac{1}{d}})$ in expectation and with high probability.
- Prove that $\mathrm{WLO}(Y) = O(n^{1-\frac{1}{d}})$ in expectation and with high probability.
- Because both bounds hold with high probability, we can divide the expected values. If one of the statements does not hold, we use the worst-case upper bound for the expected value, which is $O(\log n)$.

The following lemma is given without a proof. Part of the proof is left as an exercise.

Lemma 3.11. *There exist constants c_d and c'_d, depending only on d, such that*

- $\mathbb{E}(\mathrm{TSP}(Y)) \geq c_d \cdot \sigma n^{1-\frac{1}{d}}$ *and*
- $\mathrm{TSP}(Y) \geq c_d \cdot \sigma n^{1-\frac{1}{d}}$ *with a probability of at least* $1 - \exp(-c'_d n)$.

Chandra et al. [7] proved the following bound on the length of any 2-optimal tour. We only sketch their proof.

Lemma 3.12. *Let $X \subseteq [a,b]^d$ be a set of n points, and let T be any 2-optimal tour through X. Then the length $L(T)$ of T is bounded from above by $O\big((b-a) \cdot n^{1-\frac{1}{d}}\big)$.*

The same upper bound as in Lemma 3.12 holds also for partial 2-optimal tours, where a partial 2-optimal tour is a subset of edges of a 2-optimal tour.

Combining Lemma 3.12 with the fact that not too many points can be too far outside of the unit hypercube, we obtain the following lemma.

Lemma 3.13. *For $d \geq 2$, constants $c_d, c'_d > 0$, and all $\sigma \leq 1$, the following holds: The probability that there exists a partial 2-optimal tour T through Y that has a length of more than $c_d \cdot n^{1-\frac{1}{d}}$ is bounded by $\exp(-c'_d\sqrt{n})$.*

Theorem 3.14. *Fix $d \geq 2$. Let $Y \subseteq \mathbb{R}^d$ be obtained as described above. Then*

$$\mathbb{E}\left(\frac{\mathrm{WLO}(Y)}{\mathrm{TSP}(Y)}\right) = O\left(\frac{1}{\sigma}\right).$$

We note that the upper bound of Theorem 3.14 can be improved to $O\big(\log(1/\sigma)\big)$. This seems to be an almost optimal bound, as there exist instances X of n points such that

$$\mathbb{E}\left(\frac{\mathrm{WLO}(Y)}{\mathrm{TSP}(Y)}\right) = \Omega\left(\frac{\log n}{\log\log n}\right)$$

for $\sigma = O(1/\sqrt{n})$ [11]. The idea to prove this smoothed lower bound for the approximation ratio is to show that the lower bound example of Theorem 3.2 can be made stable for perturbations with $\sigma = O(1/\sqrt{n})$.

4 *k*-means method for clustering

***k*-means clustering.** Given a set $X \subseteq \mathbb{R}^d$ of n points, the goal of k-means clustering is to partition the n points into clusters $C_1, \ldots, C_k$ and to compute cluster centers $c_1, \ldots, c_k \in \mathbb{R}^d$ such that

$$\sum_{i=1}^{k} \sum_{x \in C_i} \|x - c_i\|^2$$

is minimized. Note that the cluster centers are not necessarily points in X and that the number k of clusters is given as part of the input.

If we have to cluster centers, then this specifies (up to tie-breaking) a clustering: every point x is assigned to the cluster whose center is closest to it. The other way round, if we have clusters $C_1, \ldots, C_k$, then the cluster center c_i should be chosen as the center of gravity of C_i. This follows from the following lemma. In the following, let $\mathrm{cm}(C) = \frac{1}{|C|} \cdot \sum_{x \in C} x$ denote the center of mass of a finite set C of points.

Lemma 4.1. *Let $C \subseteq \mathbb{R}^d$ be a finite set of points, and let $z \in \mathbb{R}^d$ be arbitrary. Then*

$$\sum_{x \in C} \|x - z\|^2 = \sum_{x \in C} \|x - \mathrm{cm}(C)\|^2 + |X| \cdot \|\,\mathrm{cm}(C) - z\|^2.$$

***k*-means method.** The k-means method is a simple local improvement heuristic for finding k-means clusterings. It alternates between optimizing the clustering based on the given centers and optimizing the centers based on the given clustering. Its worst-case running-time is exponential in the number k of clusters [19]. We can choose $k = \Theta(n)$, which shows that the worst-case number of iterations can be exponential. This holds even for $d = 2$.

The only known worst-case upper bound for the number of iterations is n^{3kd}. This upper bound is based on the trivial fact that the number of different clusterings is an upper bound for the number of iterations. The non-trivial part of this upper bound is to prove that the number of possible clusterings of n points in d-dimensional space into k clusters is bounded from above by n^{3kd}.

In contrast to the poor worst-case performance, the k-means method is a very popular, if not the most popular, clustering algorithm, and the reason for the its popularity is its speed in practice. In order to explain this speed, we analyze its running-time in the framework of smoothed analysis.

4.1 Smoothed analysis of the *k*-means method

The smoothed analysis that we present uses a combination of ideas by Arthur and Vassilvitskii [2] and Arthur et al. [1]. We only prove a bound polynomial in n^k and $1/\sigma$. It covers the main ideas of analyzing the k-means method. We note that the exponent k can be removed at the expense of a more technical analysis.

Model and idea. The model that we use for the smoothed analysis is the same as for the 2-opt heuristic: an adversary specifies a set $X \subseteq \mathbb{R}^d$ of n points. Then these points are perturbed by independent Gaussian distributions of standard deviation σ. We call the resulting point set Y, and we run the k-means method on this point set Y. We restrict our analysis to the case $\sigma \leq 1$. The upper bound for $\sigma = 1$ holds also for larger values of σ (see exercises). In the following, we also make the (natural) assumption that $k, d \leq n$. In many applications, k and d are even considered to be constant. Using the upper bound of n for k and d sometimes simplifies bounds.

The main idea is similar to the 2-opt heuristic: We use the objective function as potential and show that it has to decrease sufficiently quickly. There are two issues that make this more difficult than for the 2-opt heuristic: First, for the 2-opt heuristic, four points suffice to describe exactly what happens. This allowed us to analyze single 2-opt steps and to apply a union bound. In contrast, such a compact description of iterations does not seem to exist for the k-means method. Second, it can apparently happen that there are iterations in which the objective function decrease only by a negligible amount. (This is only seems likely, we do not have a formal prove

for this.) This makes it necessary to consider longer sequences of iterations, similar to the analysis of linked pairs of 2-opt steps.

Decrease of the objective function. Lemma 4.1 also implies that moving a cluster center c_i by a distance of ε to the center of mass of its point set C_i decreases the objective value by $\varepsilon^2 \cdot |C_i| \geq \varepsilon^2$.

For a hyperplane H and a point z, we denote by $\operatorname{dist}(z, H)$ the distance of z to H. For analyzing the decrease of the objective value caused by reassigning a point, the notion of a bisecting hyperplane is needed: for two points $x, y \in \mathbb{R}^d$ with $x \neq y$, we call a hyperplane H the bisector of x and y if H is orthogonal to $x - y$ and $\operatorname{dist}(x, H) = \operatorname{dist}(y, H)$. This implies

$$H = \left\{ z \in \mathbb{R}^d \mid 2z^{\mathrm{T}}(x - y) = (x + y)^{\mathrm{T}}(x - y) \right\}.$$

Lemma 4.2. *Let c_i and c_j be two cluster centers with bisector H, and let $y \in C_i$. If $\|y - c_j\| < \|y - c_i\|$, then reassigning y to C_j decreases the objective value by*

$$2 \cdot \operatorname{dist}(y, H) \cdot \|c_i - c_j\|.$$

Proof. The decrease is given by

$$\begin{aligned} \|c_i - y\|^2 - \|c_j - y\|^2 &= c_i^{\mathrm{T}} c_i - 2y^{\mathrm{T}} c_i - c_j^{\mathrm{T}} c_j + 2y^{\mathrm{T}} c_j \\ &= (c_j - c_i)^{\mathrm{T}}(2y - c_j - c_i). \end{aligned}$$

Let $v = \frac{c_j - c_i}{\|c_j - c_i\|}$. Then $(2y - c_j - c_i)^{\mathrm{T}} v = 2(y - \frac{c_j + c_i}{2})^{\mathrm{T}} v = 2 \operatorname{dist}(y, H)$, which implies the lemma. □

The rough idea for the smoothed analysis is as follows: if many points are reassigned to a new cluster (called a *dense* iteration), then it is unlikely that all of them are close to the hyperplane separating the cluster from which they come and the cluster that they join. If only few points are reassigned (called a *sparse* iteration), then we hope to be able to prove that at least one cluster center must move significantly. This hope turns out to wrong for a single iteration, but in any short sequence of sparse iterations (called an *epoch*), this is likely to happen.

4.2 Dense iterations

We call an iteration a *dense iteration* if there is at least one cluster that gains or loses in total at least $2kd$ points.

We call the point set Y *ε-separated* if, for all hyperplanes $H \subseteq \mathbb{R}^d$, there are less than $2d$ points $y \in Y$ with $\operatorname{dist}(y, H) \leq \varepsilon$.

Lemma 4.3. *If Y is ε-separated, then the potential decreases by at least $2\varepsilon^2/n$ in every dense iteration.*

Proof. Since the iteration is dense, there must be a cluster C_i that exchanges at least $2kd$ points with other clusters in this iteration. Hence, there must be another cluster C_j with which C_i exchanges at least $2d+1$ points. Since Y is ε-separated, at least one point $y \in Y$ that switches between C_i and C_j is at a distance of at least ε from the hyperplane bisecting $c_i = \mathrm{cm}(C_i)$ and $c_j = \mathrm{cm}(C_j)$.

In order to bound the decrease of the objective value from below by $2\varepsilon^2/n$, we need a lower bound of ε/n for $\|c_i - c_j\|$. There exists a hyperplane H' (the bisector from the previous iteration) that separates C_i from C_j. Among all at least $2d+1$ points that want to switch in the current iteration, at least one point y must be at a distance of at least ε from H' since Y is ε-separated. Assume without loss of generality that $y \in C_i$. Then, since $|C_i| \leq n$, we have $\frac{\varepsilon}{n} \leq \mathrm{dist}(c_i, H') \leq \|c_i - c_j\|$. □

Lemma 4.4. *The probability that Y is not ε-separated is at most $n^{2d} \cdot \left(\frac{2d\varepsilon}{\sigma}\right)^d$.*

Proof. According to Lemma 4.5 below, it suffices to show that the probability that there are two sets P and P' consisting of d points of Y each such that all points of P' are $(2d\varepsilon)$-close to the hyperplane through P is bounded by $n^{2d} \cdot \left(\frac{2d\varepsilon}{\sigma}\right)^d$.

Fix any P and P'. The probability that all points of P' are within distance $2d\varepsilon$ of the hyperplane through P is at most $(2d\varepsilon/\sigma)^d$ by independence of the perturbation of the points. The lemma follows by a union bound over the at most n^{2d} choices for P and P'. □

Lemma 4.5. *Let $P \subseteq \mathbb{R}^d$ be any finite set of at least d points, and let $H \subseteq \mathbb{R}^d$ be an arbitrary hyperplane. Then there exists a hyperplane $H' \subseteq \mathbb{R}^d$ that contains at least d points of P such that*

$$\max_{p \in P}\big(\mathrm{dist}(p, H')\big) \leq 2d \cdot \max_{p \in P}\big(\mathrm{dist}(p, H)\big).$$

Proof. By shifting both P and H, we can assume without loss of generality that $0 \in P$. Let $\ell = \max_{p \in P} \mathrm{dist}(p, H)$. For $p \in P$, let $\pi(p)$ be the projection of p onto H.

Let $V = \{\pi(p) - \pi(0) \mid p \in P\}$. The set V lies in a $(d-1)$-dimensional affine subspace of $\mathbb{R}^d$. According to Lemma 4.6 below, there exists a subset $V_0 \subseteq V$ with $|V_0| \leq d-1$ such that all points in V can be written as linear combinations of vectors in V_0 such that the coefficients involved have absolute values of at most 1.

Let $Q = \{p \in P \mid \pi(p) - \pi(0) \in V_0\}$. If there are several points p with the same $\pi(p) - \pi(0)$, then we select one such point arbitrarily. In this way, we have $|Q| \leq d-1$. Let H' be the hyperplane through 0 and Q.

We have $\|p - \pi(p)\| = \mathrm{dist}(p, H) \leq \ell$ for all $p \in P$ by the definition of π and ℓ. For all $q \in Q \cup \{0\}$, we have $q \in H'$ by the choice of H'. These two observations imply $\mathrm{dist}(\pi(q), H') \leq \|q - \pi(q)\| \leq \ell$.

Finally, we have to show that $\operatorname{dist}(p, H') \leq 2d\ell$ for all $p \in P$. We have

$$
\begin{aligned}
&\operatorname{dist}(p, H') \\
&\leq \|p - \pi(p)\| + \operatorname{dist}(\pi(p), H') && \text{(triangle inequality)} \\
&\leq \ell + \operatorname{dist}(\pi(p), H') && \text{(reasoning above)} \\
&\leq \ell + \operatorname{dist}(\pi(0), H') + \operatorname{dist}(\pi(p) - \pi(0), H') && \text{(triangle inequality)} \\
&\leq 2\ell + \operatorname{dist}(\pi(p) - \pi(0), H') && (\text{since } \operatorname{dist}(\pi(q), H') \leq \ell) \\
&\leq 2\ell + \operatorname{dist}\left(\sum_{q \in Q} c_q\big(\pi(q) - \pi(0)\big), H'\right) && \text{(properties of } Q) \\
&\leq 2\ell + \sum_{q \in Q} \operatorname{dist}\left(c_q\big(\pi(q) - \pi(0)\big), H'\right) && \text{(triangle inequality)} \\
&\leq 2\ell + \sum_{q \in Q} \operatorname{dist}\big(\pi(q) - \pi(0), H'\big) && (0 \in H' \text{ and } |c_q| \leq 1) \\
&\leq 2\ell + \sum_{q \in Q} \operatorname{dist}\big(\pi(q), H'\big) + \operatorname{dist}(\pi(0), H') && (0 \in H', \text{ triangle inequality}) \\
&\leq 2\ell + 2|Q|\ell \leq 2d\ell.
\end{aligned}
$$

□

Lemma 4.6. *Let $V \subseteq \mathbb{R}^d$ be a finite set of vectors. Then there exists a subset $V_0 \subseteq V$ with $|V_0| \leq d$ such that, for all $v \in V$,*

$$v = \sum_{u \in V_0} c_{uv} u$$

with $|c_{uv}| \leq 1$.

Proof. Without loss of generality, we assume that $\operatorname{span}(V) = \mathbb{R}^d$. Otherwise, we consider a lower-dimensional subspace. Choose $u_1, \ldots, u_d \in V$ that maximize $|\det(u_1, \ldots, u_d)|$, i.e., the determined by considering $u_1, \ldots, u_d$ as the columns of a square matrix. We break ties arbitrarily. This implies that $\operatorname{span}(u_1, \ldots, u_d) = \mathbb{R}^d$. Let $V_0 = \{u_1, \ldots, u_d\}$.

Hence, we can write $v \in V$ as $v = \sum_{i=1^d} c_i u_i$ with

$$c_i = \frac{\det(u_1, \ldots, u_{i-1}, v, u_{i+1}, \ldots, u_d)}{\det(u_1, \ldots, u_d)}$$

according to Cramer's rule. By the choice of $u_1, \ldots, u_d$, the absolute value of the denominator is not smaller than the absolute value of the numerator, which proves the lemma. □

By combining Lemmas 4.3 and 4.4, we obtain the following result.

Lemma 4.7. *For $d \geq 3$, the probability that there exists a dense iteration in which the potential decreases by less than ε is bounded from above by*

$$\left(\frac{dn^{2.5}\sqrt{\varepsilon}}{\sigma}\right)^d \leq \left(\frac{n^{3.5}\sqrt{\varepsilon}}{\sigma}\right)^d.$$

4.3 Sparse iterations

We call an iteration an *ℓ-sparse* iteration if every cluster gains and loses in total at most ℓ points.

Let C_i^t be the set of points in the i-th cluster in iteration t of the k-means method. We define an *epoch* to be a sequence of consecutive iterations $t, t+1, \ldots, t+\ell$ in which no cluster center assumes more than two different point sets. This means that $|\{C_i^a \mid t \leq a \leq t+\ell\}| \leq 2$ for all $i \in \{1, 2, \ldots, k\}$.

The length of every epoch is bounded by 3. This means that after the fourth iteration, either the k-means method has terminated, or at least one of the clusters assumes a third point set. To prove a bound that is polynomial in n^k and $1/\sigma$, the following (weaker) observation suffices.

Lemma 4.8. *The length of every epoch is bounded by 2^k.*

Proof. After 2^k iterations, at least one cluster must have assumed a third set of points. Otherwise, a clustering would show up a second time. This is not possible as the objective value strictly decreases in every iteration. □

We call a set $Y \subseteq \mathbb{R}^d$ of data points *(η, ℓ)-coarse* for some numbers $\eta > 0$ and $c \in \mathbb{N}$ if, for all triples $P_1, P_2, P_3 \subseteq Y$ of different subsets with $|P_1 \triangle P_2| \leq \ell$ and $|P_2 \triangle P_3| \leq \ell$, we have $\|\operatorname{cm}(P_i) - \operatorname{cm}(P_{i+1})\| > \eta$ for at least one $i \in \{1, 2\}$. Here, "$\triangle$" denotes the symmetric difference of two sets.

Lemma 4.9. *Assume that Y is (η, ℓ)-coarse, and consider a sequence of 2^k consecutive iterations of the k-means method. If each of these iterations is ℓ-sparse, then the potential decreases by at least η^2.*

Proof. After 2^k iterations, at least one cluster has assumed a third configuration (Lemma 4.8). Since the iterations are ℓ-sparse and the instance is (η, ℓ)-coarse, the cluster center of this center must have moved in at least one iteration by at least η. This decreases the potential by at least η^2 according to Lemma 4.1. □

Lemma 4.10. *The probability that Y is not (η, ℓ)-coarse is at most $(7n)^{2\ell} \cdot (2n\ell\eta/\sigma)^d$.*

Proof. Let $P_1, P_2, P_3 \subseteq Y$ be three sets with $|P_1 \triangle P_2| \leq \ell$ and $|P_2 \triangle P_3| \leq \ell$. Let $A = P_1 \cap P_2 \cap P_3$, and let B_1, B_2, and B_3 be sets such that $P_i = A \cup B_i$ for $i \in \{1, 2, 3\}$ and B_1, B_2, and B_3 are disjoint to A. We have $|B_1 \cup B_2 \cup B_3| \leq 2\ell$ and $B_1 \cap B_2 \cap B_3 = \emptyset$.

We perform a union bound over the choices of for the sets B_1, B_2, and B_3. The number of possible choice for these sets is upper-bounded by $7^{2\ell} \cdot \binom{n}{2\ell} \leq (7n)^{2\ell}$: We select 2ℓ elements of Y. Then we choose for each element in which of the three sets it should belong. None of these elements belongs to all sets, but there can be elements that belong to no set. We need this possibility since we can have $|B_1 \cup B_2 \cup B_3| < 2\ell$.

For $i \in \{1, 2, 3\}$, we have

$$\operatorname{cm}(P_i) = \frac{|A|}{|A| + |B_i|} \cdot \operatorname{cm}(A) + \frac{|B_i|}{|A| + |B_i|} \cdot \operatorname{cm}(B_i).$$

Hence, for $i \in \{1, 2\}$, we can write $\operatorname{cm}(P_i) - \operatorname{cm}(P_{i+1})$ as

$$\begin{aligned} \operatorname{cm}(P_i) - \operatorname{cm}(P_{i+1}) = &\left(\frac{|A|}{|A| + |B_i|} - \frac{|A|}{|A| + |B_{i+1}|} \right) \cdot \operatorname{cm}(A) \\ &+ \frac{|B_i|}{|A| + |B_i|} \cdot \operatorname{cm}(B_i) - \frac{|B_{i+1}|}{|A| + |B_{i+1}|} \cdot \operatorname{cm}(B_{i+1}). \end{aligned} \tag{1}$$

We distinguish two cases. The first case is that $|B_i| = |B_{i+1}|$ for some $i \in \{1, 2\}$. Then (1) simplifies to

$$\begin{aligned} \operatorname{cm}(P_i) - \operatorname{cm}(P_{i+1}) &= \frac{|B_i|}{|A| + |B_i|} \cdot \operatorname{cm}(B_i) - \frac{|B_i|}{|A| + |B_i|} \cdot \operatorname{cm}(B_{i+1}) \\ &= \frac{1}{|A| + |B_i|} \cdot \left(\sum_{y \in B_i \setminus B_{i+1}} y - \sum_{y \in B_{i+1} \setminus B_i} y \right). \end{aligned} \tag{2}$$

Since $B_i \neq B_{i+1}$ and $|B_i| = |B_{i+1}|$, there exists a point $y \in B_i \setminus B_{i+1}$.

We use the principle of deferred decisions. We first fix all points in $(B_i \cup B_{i+1}) \setminus \{y\}$ arbitrarily. Then $\| \operatorname{cm}(P_i) - \operatorname{cm}(P_{i+1}) \| \leq \eta$ is equivalent to the event that y assumes a position in a hyperball of radius $(|A| + |B_i|) \cdot \eta \leq n\eta$. The probability that this happens is bounded from above by $(n\eta/\sigma)^d \leq (2n\eta\ell/\sigma)^d$ by Corollary 1.4.

The second case is that $|B_1| \neq |B_2| \neq |B_3|$. We denote by $\mathcal{B}(c, r) = \{x \in \mathbb{R}^d \mid \|x - c\| \leq r\}$ the hyperball of radius r around c. For $i \in \{1, 2\}$, let

$$r_i = \left(\frac{|A|}{|A| + |B_i|} - \frac{|A|}{|A| + |B_{i+1}|} \right)^{-1}$$

and

$$Z_i = \frac{|B_{i+1}|}{|A| + |B_{i+1}|} \operatorname{cm}(B_{i+1}) - \frac{|B_i|}{|A| + |B_i|} \operatorname{cm}(B_i).$$

We observe (see (2)) that the event $\| \operatorname{cm}(P_i) - \operatorname{cm}(P_{i+1}) \| < \eta$ is equivalent to the event that $\operatorname{cm}(A) \in \mathcal{B}_i = \mathcal{B}(r_i Z_i, |r_i|\eta)$. Consequently, a necessary condition that the event $\| \operatorname{cm}(P_i) - \operatorname{cm}(P_{i+1}) \| < \eta$ occurs for both $i \in \{1, 2\}$ is that the hyperballs $\mathcal{B}_1$ and $\mathcal{B}_2$ intersect.

The two hyperballs intersect if and only if their centers are at a distance of at most $(|r_1| + |r_2|) \cdot \eta$ of each other. Hence,

$$\begin{aligned} & \mathbb{P}\big(\|\operatorname{cm}(P_1) - \operatorname{cm}(P_2)\| \le \eta \text{ and } \|\operatorname{cm}(P_2) - \operatorname{cm}(P_3)\| \le \eta\big) \\ \le\ & \mathbb{P}\left(\|r_1 Z_1 - r_2 Z_2\| \le \big(|r_1| + |r_2|\big)\eta\right) \end{aligned}$$

Some tedious, not very insightful calculations show that the latter probability can be upper-bounded by $(2n\ell\eta/\sigma)^d$. □

The main technical problem in the proof of Lemma 4.10 is that we cannot control the position of $\operatorname{cm}(A)$. The reason is that there are too many possible choices for points in A. Because of this, we cannot simply apply a union bound over all possibilities for A.

The first case in the proof of Lemma 4.10 shows that for the case that the same number of points leaves and enters a cluster, it is already quite likely that the potential decreases significantly. In this case, no epochs are needed. The reason is that the influence of $\operatorname{cm}(A)$ cancels out in $\operatorname{cm}(P_i) - \operatorname{cm}(P_{i+1})$ if $|B_i| = |B_{i+1}|$. In this way, the difficulty that we have to say something about $\operatorname{cm}(A)$ disappears.

If $|B_i| \neq |B_{i+1}|$, then $\operatorname{cm}(A)$ shows up with different coefficients in $\operatorname{cm}(C_i)$ and $\operatorname{cm}(C_{i+1})$. Hence, $\operatorname{cm}(A)$ shows up with a non-zero coefficient in $\operatorname{cm}(P_i) - \operatorname{cm}(P_{i+1})$. This implies that for any position of $\operatorname{cm}(B_i)$ and $\operatorname{cm}(B_{i+1})$, we can choose $\operatorname{cm}(A)$ such that $\operatorname{cm}(P_i)$ and $\operatorname{cm}(P_{i+1})$ are close. However, this is only possible if $\operatorname{cm}(A)$ assumes a position in some hyperball of a certain radius. The center of this hyperball depends only on $\operatorname{cm}(B_i)$ and $\operatorname{cm}(B_{i+1})$. We conclude that we can only have $\|\operatorname{cm}(P_1) - \operatorname{cm}(P_2)\| \le \eta$ and $\|\operatorname{cm}(P_2) - \operatorname{cm}(P_3)\| \le \eta$ simultaneously if these hyperballs intersect. We can upper-bound the probability that these hyperballs intersect by only considering the points in $B_1 \cup B_2 \cup B_3$. Thus, considering three different sets instead of only two different sets allows us to remove the necessity to take the possibilities to choose A into account.

Lemma 4.11. *The probability that there is a sequence of 2^k consecutive sparse iterations that decrease the potential by less than ε is at most*

$$(7n)^{4kd} \cdot \left(\frac{4nkd\sqrt{\varepsilon}}{\sigma}\right)^d \le \left(\frac{c_{\text{sparse}}^k n^{4k+3}\sqrt{\varepsilon}}{\sigma}\right)^d$$

for some sufficiently large constant c_{sparse}.

4.4 Putting things together

Lemma 4.12. *Let $D = 10\sqrt{kd \ln n}$. Then $\mathbb{P}(Y \not\subseteq [-D, D]^d) \le n^{-3kd}$.*

A consequence of the lemma above is that after the first iteration, the potential is bounded by $ndD^2 = c_{\text{init}} nd^2 k \ln n \le c_{\text{init}} n^5$ for some constant c_{init}. (The upper bound of c_{init} is very poor, but simplifies the bounds.)

Theorem 4.13. *The smoothed number of iterations of the k-means method is bounded by* $O(2^k n^{14k+12}/\sigma^2)$.

Proof. We choose $\varepsilon = \sigma^2 \cdot n^{-14k-7}$. By Lemma 4.7, the probability that there is a dense iteration that decreases the potential by at most ε is at most cn^{-3kd} for some constant $c > 0$. By Lemma 4.11, the probability that there is a sequence of 2^k consecutive sparse iterations that decrease the potential by at most ε is also at most $c'n^{-3kd}$ for some constant $c' > 0$. By Lemma 4.12, the probability that the initial potential is more than $O(n^5)$ is also at most n^{-3kd}.

If any of these events happens nevertheless, we bound the number of iterations by its worst-case bound of n^{3kd}. This contributes only $O(1)$ to the expected value. Otherwise, the number of iterations is bounded by $O(2^k n^{14k+12}/\sigma^2)$. □

We observe that the bound obtained in Theorem 4.13 is still quite poor. In particular, it has the number k of clusters in the exponent. We remark that it can be shown that the smoothed number of iterations of k-means is bounded by a polynomial in n and $1/\sigma$ (without k or d in the exponent). The idea for this improved analysis is to refine the partitioning of iterations, not only into sparse and dense iterations [1].

4.5 Approximation ratio of the *k*-means method

The approximation performance of the k-means method is not very good. Of course, it heavily depends on the initialization. In fact, the main reason why the k-means method is so popular is its speed. This allows us to run it many times on the same data set with different initializations. The hope is that for at least one initialization, we get a good clustering.

5 Flip heuristic for Max-Cut

Max-Cut is the following optimization problem: we are given an undirected graph $G = (V, E)$ with edge weights $w = (w_e)_{e \in E}$. The goal is to find a set $C \subseteq V$ of vertices that maximize the total weight $w(C)$ of edges with exactly one endpoint in C:

$$w(C) = \sum_{u \in C, v \notin C, \{u,v\} \in E} w_e.$$

We call C a cut. Max-Cut is NP-hard. A simple local search heuristic for Max-Cut is the so-called flip heuristic: Assume that we have some cut C.

- If there is a vertex $v \in C$ with $w(C \setminus \{v\}) > w(C)$, then replace C by $C \setminus \{v\}$.

- If there is a vertex $v \in V \setminus C$ with $w(C \cup \{v\}) > w(C)$, then replace C by $C \cup \{v\}$.

The flip heuristic seems to converge quickly in experiments, but its worst-case running-time is exponential. A trivial upper bound on the number of iterations that it requires is 2^n with $n = |V|$ since every cut is visited at most once during the execution of the algorithm.

Our goal is again a smoothed analysis of the running-time of the flip heuristic. To do this, we consider the following probabilistic input model:

- An adversary specifies the graph $G = (V, E)$ and one probability density functions $f_e : [0, 1] \to [0, \phi]$ for each edge $e \in E$.
- We draw the weight w_e of edge e independent of the weights of the other edges according to the probability density function f_e.

While being a very simple local search algorithm, the flip heuristic resisted a smoothed analysis for quite a while. The reason is that the improvement caused by flipping a single vertex depends not only on the vertex itself, but on the positions of all of its neighbors. (We see in the exercises an example of a smoothed analysis of the flip heuristic that depends exponentially on the maximum degree of the given graph.)

We want to sketch a proof of the following theorem by Etscheid and Röglin [10].

Theorem 5.1. *The smoothed number of iterations of the flip heuristic is bounded from above by a polynomial in $n^{\log n}$ and ϕ.*

(Running-times of the form $n^{\log^c n}$ for some $c > 0$ or, equivalently, $2^{\log^c n}$ for some $c > 1$ are sometimes called quasi-polynomial.)

5.1 Difficulties and key idea

As said above, the main challenge is that there does not seem to be a compact description of what happens in an iteration, as we had it for the 2-opt heuristic. Therefore, we have to consider sequences of iterations. Consider the smallest improvement caused by any sequence of ℓ consecutive iterations. If we consider all such sequences, we have roughly $2^n n^\ell$ possibilities: there are 2^n possibilities for the initial cut, and then we have to choose ℓ vertices to be flipped in the ℓ iterations.

Consider a fixed sequence of ℓ iterations. From this sequence, we obtain a system of ℓ different linear combinations of edge weights with integer coefficients. The improvement caused by this sequence is at most ε only if all linear combinations assume a value in the interval $(0, \varepsilon]$. Let r denote the rank of this set of linear combinations. It can be shown that the probability that all linear combinations fall into $(0, \varepsilon]$ is bounded from above by $(\phi\varepsilon)^r$

according to the following lemma. (This is not trivial, but also not impossible to prove. We refer to Etscheid and Röglin [10, Lemma A.1]. It is crucial that the coefficients are integers.)

Lemma 5.2. *Let $X_1, \dots, X_m$ be independent, real-valued random variables, where the density of X_i is bounded by ϕ. Let $a^1, \dots, a^k \in \mathbb{Z}^m$ be linearly independent row vectors. Let $X = (X_1, \dots, X_m)^{\mathrm{T}}$, and let $Y_i = a^i X$. Then*

$$\mathbb{P}\big(Y_1, \dots, Y_k \in (0, \varepsilon]\big) \leq (\varepsilon\phi)^k.$$

If there exists a constant $\alpha > 0$ such that the linear combinations of every sequence of ℓ consecutive iterations have a rank of at least $\alpha\ell$, then we would obtain a polynomial bound for the smoothed running-time according to the following lemma.

Lemma 5.3 (useless lemma). *Assume that there exists a constant $\alpha > 0$ such that the linear combinations of every sequence of ℓ consecutive iterations have a rank of at least $\alpha\ell$, then the expected number of iterations of the flip heuristic is bounded from above by $O(n^{3+\frac{1}{\alpha}}\phi)$.*

Proof. We choose $\ell = n$. Fix any sequence of n iterations. Their linear combinations have a rank of at least αn by the assumption of the lemma. Hence, the probability that all of them yield an improvement of at most ε is bounded from above by $(\varepsilon\phi)^{\alpha n}$ by Lemma 5.2. By a union bound over $2^n n^n$ possibilities, we obtain that the probability that any sequence of n iterations yields an improvement of at most ε is at most $p = (2n\varepsilon^\alpha\phi^\alpha)^n$.

We choose $\varepsilon = (4n)^{-1/\alpha}/\phi$. Then $p = 2^{-n}$. Since the weight of the cut is initially at least 0 and in the end at most n^2, the number of iterations is at most n^3/ε with a probability of at least $1-2^{-n}$. Since the worst-case number of iterations is upper-bounded by 2^n, the expected number of iterations is at most $n^3/\varepsilon + 1 = O(n^{3+\frac{1}{\alpha}}\phi)$. □

While the lemma is true, it is useless since its assumption about the rank is false. In fact, if only a few nodes change multiple times, then the rank can be significantly smaller than linear in the length of the sequence.

The following is the key observation to prove Theorem 5.1: Let $v \in V$ be a node that moves at least twice. We consider the two linear combinations of edge weights of two consecutive moves of node v (note that these moves do not have to be consecutive among all moves) and sum them up. Let L be the result. We observe now that L contains only edge weights with non-zero coefficient of edges connecting v to a node u that has moved an odd number of times between the two moves of v. This reduces the factor incurred by the union bound from 2^n for the initial configuration to 2^ℓ, namely the positions of the nodes that are involved in the sequence of iterations that we consider. Note that this is only an improvement if we consider sequences of length much smaller than n.

We call a sequence of ℓ consecutive iterations a k-repeating sequence if at least $\lceil \ell/k \rceil$ distinct nodes move at least twice in this sequence. It turns out that the rank of the set of linear combinations obtained from a k-repeating sequence of ℓ iterations is sufficiently large to prove our bound.

5.2 Sketch of the quasi-polynomial bound

The proof proceeds in three steps:

(i) Prove that the linear combinations obtained from a k-repeating sequence of length ℓ have a sufficiently large rank.

(ii) Prove that any k-repeating sequence of length ℓ is unlikely to yield only a small improvement.

(iii) Prove that it suffices to consider k-repeating sequences of length ℓ to get a lower bound on the smallest improvement caused by any sequence of length ℓ.

The following lemma, whose proof we omit, shows that the rank of the linear combinations obtained from a k-repeating sequence of ℓ iterations is large. It answers Item (i) of the list above.

Lemma 5.4. *Consider an arbitrary k-repeating sequence of ℓ consecutive iterations of the flip heuristic. Consider the set of linear combinations of edge weights obtained by adding the linear combinations of two consecutive moves of the same node. Then the rank of this set of linear combinations is at least $\lceil \frac{\ell}{2k} \rceil$.*

Using Lemma 5.4, we can show that the probability that there exists some k-repeating sequence of length ℓ that yields only a small improvement is small. This yields Lemma 5.5 below, which addresses Item (ii) from the list above. In the following, let $r = \lceil \frac{\ell}{2k} \rceil$. Let $\Delta_{\ell,k}$ denote the smallest improvement caused by any k-repeating sequence of length ℓ.

We call a node $v \in V$ active in a sequence of iterations if v is moved at least once in this sequence.

Lemma 5.5. $\mathbb{P}\big(\Delta_{\ell,k} \leq \varepsilon\big) \leq (2n)^{\ell} \cdot (2\phi\varepsilon)^{r}$.

Proof. There are at most n^{ℓ} choices for the ℓ nodes that move in the consecutive iterations of a k-repeating sequence of length ℓ. Furthermore, there are 2^{ℓ} possible starting configurations of the at most ℓ active nodes in this sequence. Thus, with a union bound over $(2n)^{\ell}$ possibilities, it suffices to consider any fixed k-repeating sequence of length ℓ.

Consider a node $v \in V$ that moves at least twice, and consider linear combinations L_1 and L_2 corresponding to two consecutive moves of v. Then, as argued above, $L = L_1 + L_2$ contains only weights belong to edges

connecting v to active nodes. For such nodes, we have fixed the starting configuration.

If both L_1 and L_2 assume a value in $(0, \varepsilon]$, then L assumes a value in $(0, 2\varepsilon]$. By Lemma 5.4, the rank of all linear combinations constructed like L is at least r. Hence, we can apply Lemma 5.2. □

Finally, Item (iii) remains to be addressed. This is done in the following lemma, the proof of which we omit. For the following lemma, we fix $k = \lceil 5 \log_2 n \rceil$.

Lemma 5.6. *Let $\Delta = \min_{1 \le \ell \le 5n} \Delta_{\ell, \lceil 5 \log_2 n \rceil}$. Then Δ is a lower bound for the improvement caused by any sequence of $5n$ steps.*

Now we can put everything together to prove Theorem 5.1.

Proof of Theorem 5.1. The weight of every cut is in the interval $[0, n^2]$. Let T be the maximum number of iterations that the flip heuristic can need on the given (random) instance. Then $T \ge 5nt$ only if the minimum improvement caused by any sequence of $5n$ iterations is at least Δ with $\Delta = n^2/t$.

By Lemma 5.6, it suffices to consider $\Delta_{\ell,k}$ for $\ell \in \{1, \dots, 5n\}$ to get a lower bound for Δ. By Lemma 5.5, we have

$$\mathbb{P}\big(\Delta \le \varepsilon\big) \le \sum_{\ell=1}^{5n} (2n)^{\ell} \cdot (2\phi\varepsilon)^r \le \sum_{\ell=1}^{5n} \Big((2n)^{2k} \cdot (2\phi\varepsilon)\Big)^r.$$

Let $\zeta = 2\phi n^2 (2n)^{2k}$. Hence,

$$\mathbb{P}\big(T \ge 5nt\big) \le \sum_{\ell=1}^{5n} \Big((2n)^{2k} \cdot (2\phi n^2/t)\Big)^r = \sum_{\ell=1}^{5n} \left(\frac{\zeta}{t}\right)^r.$$

We set $t_i = i \cdot \zeta$. Then, using $r = \lceil \ell/(2k) \rceil$, we obtain

$$\mathbb{P}\big(T \ge 5nt_i\big) \le \sum_{\ell=1}^{5n} \left(\frac{1}{i}\right)^{\left\lceil \frac{\ell}{2k} \right\rceil} \le \sum_{\ell=1}^{\infty} 2k \cdot \left(\frac{1}{i}\right)^{\ell} = \frac{2k}{i-1}.$$

Combining this with the trivial upper bound of 2^n for the number of iterations, we obtain

$$\mathbb{E}\left(\frac{T}{5n\zeta}\right) \le 1 + \sum_{i=2}^{2^n} \mathbb{P}\big(T \ge 5nt_i\big) \le 1 + \sum_{i=2}^{2^n} \frac{2k}{i-1} = O(kn).$$

Hence, the expected number of iterations is bounded from above by

$$5kn^2\zeta\phi = \phi \cdot n^{O(\log n)}.$$

□

6 Solving knapsack in polynomial time

The knapsack problem is the following optimization problem: we are given n items $1, \dots, n$ with non-negative weights $w_1, \dots, w_n$ and profits $p_1, \dots, p_n$ and a capacity t. The goal is to find a subset $I \subseteq \{1, \dots, n\}$ of the items that weighs at most t at maximizes the profit. This means that we want to maximize $\sum_{i \in I} p_i$ such that $\sum_{i \in I} w_i \leq t$. We identify subsets $I \subseteq \{1, \dots, n\}$ with binary vectors $x \in \{0,1\}^n$ in the obvious way: $x_i = 1$ if and only if $i \in I$. In this way, the knapsack problem becomes a very simple integer linear programming problem: maximize $p^{\mathrm{T}}x$ subject to $w^{\mathrm{T}}x \leq t$ and $x \in \{0,1\}^n$.

The knapsack problem is NP-hard. Still, even large practical instances of the knapsack problem can be solved in practice in reasonable time. The results of this section are by Beier et al. [4].

6.1 Pareto curves and the Nemhauser–Ullman algorithm

We can view the knapsack problem as a bi-criteria optimization problem: we simultaneously want to maximize the profit and to minimize the weight.

However, with more than one objective function, we cannot directly compare solutions. There might be a more valuable solution that is heavier than another, less valuable solution. When we try to optimize weight and profit, it is unclear which solution is preferable. To transfer the concepts of "better" and "optimal" to multi-criteria problems, we introduce the concepts of domination and Pareto curves.

Let $x, y \in \{0,1\}^n$ be two binary vectors. We say that x dominates y if $w^{\mathrm{T}}x \leq w^{\mathrm{T}}y$ and $p^{\mathrm{T}}x \geq p^{\mathrm{T}}y$ and at least one of the two inequalities is strict.

A binary vector $x \in \{0,1\}^n$ is called Pareto-optimal if there is no other vector in $\{0,1\}^n$ that dominates x. The Pareto curve (also known as Pareto set or efficient set) is the set of all vectors from $\{0,1\}^n$ that are Pareto-optimal.

In the following, for $x \in \{0,1\}^n$, we denote by $x_{i=1}$ the vector obtained from x by setting the i-th entry of x to 1 and by $x_{i=0}$ the vector obtained from x by setting the i-th entry of x to 0.

The idea of the Nemhauser–Ullman algorithm (NUA) is to compute the Pareto curve iteratively by considering the items $1, \dots, n$ one by one: in the k-th step we look at item k and the set of all Pareto-optimal solutions P_{k-1} of items $1, \dots, k-1$. We compute the Pareto-optimal solutions that we obtain by combining each solution from P_{k-1} with item k.

The NUA is given as Algorithm 1. The following lemma shows that it correctly computes the set P_n of all Pareto-optimal solutions. This implies that it solves the knapsack problem correctly.

Lemma 6.1. *For all $k \in \{0, \dots, n\}$, the set P_n computed by the NUA is the set of Pareto-optimal solutions that use items $1, \dots, k$.*

Algorithm 1 Nemhauser–Ullman algorithm.

input: weights $w_1, \ldots, w_n$, profits $p_1, \ldots, p_n$, capacity t

1: $P_0 \leftarrow \{(0, \ldots, 0)^{\mathrm{T}}\}$
2: **for** $k \leftarrow 1$ **to** n **do**
3: $\quad P'_k \leftarrow P_{k-1} \cup \{x_{k=1} \mid x \in P(k-1)\}$
4: $\quad$ remove all dominated items from P'_k to obtain P_k
5: **end for**

output: $x = \operatorname{argmax}\{p^{\mathrm{T}} y \mid y \in P_n, w^{\mathrm{T}} y \leq t\}$

Proof. Let Q_k be the (true) set of Pareto-optimal solutions of items $1, \ldots, k$. We need the following claim.

Claim 6.2. *If $x \in Q_k$, then $x_{k=0} \in Q_{k-1}$.*

Proof of Claim 6.2. We show that $x_{k=0} \notin Q_{k-1}$ implies $x \notin Q_k$. Assume that there is some y with $y_k = \ldots = y_n = 0$ that dominates $x_{k=0}$. If $x_k = 0$, then $x = x_{k=0}$ is dominated by y. Hence, $x \notin Q_k$. If $x_k = 1$, then x is dominated by $y_{k=1}$. Hence, $x \notin Q_k$. □

The proof is by induction. The lemma clearly holds for $k = 0$. Now assume that $P_{k-1} = Q_{k-1}$.

We first prove $Q_k \subseteq P_k$. Let $x \in Q_k$ be arbitrary. Then $x_{k=0} \in Q_{k-1} = P_{k-1}$ by Claim 6.2 and the induction hypothesis. Thus, $x \in P'_k$. If $x \notin P_k$, then there is some solution $y \in P'_k$ that dominates x. This contradicts $x \in Q_k$.

Finally, since $Q_k \subseteq P_k$ and P_k does not contain any dominated items, we have $P_k = Q_k$. □

If the elements of P_{k-1} are sorted in ascending order according to their weights, then P_k can be computed easily in linear time, and the elements are also sorted in ascending order. This yields the following lemma.

Lemma 6.3. *Let $q_k = |P_k|$ for $k \in \{1, \ldots, n\}$. Then the Nemhauser–Ullman algorithm runs in time*

$$O\left(\sum_{i=1}^{n} q_i\right).$$

This means that the running-time is linear in the size of the Pareto curve. Our goal in the following is to show that $\mathbb{E}(q_k)$ is bounded by a polynomial, which implies that the expected running-time of the NUA is also polynomial. In the worst-case, the size of the Pareto curve and the running time of the NUA are exponential.

6.2 Probabilistic analysis of the Pareto curve

We only perturb the weights, not the profits. The results that we obtain also hold if only the profits are perturbed or of both profits and weights are perturbed independently, but they no longer hold if profits and weights are not independent.

Following the one-step model (see Section 2.2), we use the following probabilistic input model: Let $\phi \geq 1$ be the perturbation parameter. An adversary specifies density functions $f_1, \ldots, f_n$ with $f_i : [0,1] \to [0,\phi]$ and profits $p_1, \ldots, p_n \in [0,1]$. Then we obtain an instance of the knapsack problem by drawing $w_1, \ldots, w_n$ independently according to the densities $f_1, \ldots, f_n$, respectively.

Our goal is to prove the following result.

Theorem 6.4. *Let $p_1, \ldots, p_n \in [0,1]$ be arbitrary profits, let $f_1, \ldots, f_n : [0,1] \to [0,\phi]$ be arbitrary density functions bounded by $\phi \geq 1$, and let $w_1, \ldots, w_n$ be weights drawn according to $f_1, \ldots, f_n$ respectively. Let $P = P_n$ be the set of Pareto-optimal solutions, and let $q = q_n = |P_n|$. Then*

$$\mathbb{E}(q) \leq \phi n^2 + 1.$$

Proof. As $w_1, \ldots, w_n \in [0,1]$, the weight of each solution in P is in the interval $[0,n]$. For any two $x, y \in \{0,1\}^n$ with $x \neq y$, we have $w^{\mathrm{T}}x \neq w^{\mathrm{T}}y$.

We partition $(0,n]$ into k small intervals $M_1^k, \ldots, M_k^k$ of size n/k each, where $M_i^k = \left(\frac{n(i-1)}{k}, \frac{ni}{k}\right]$. If the size of these intervals is sufficiently small, then the weight of any Pareto-optimal solution falls into a unique interval.

Claim 6.5. *We have*

$$\mathbb{E}(q) = 1 + \lim_{k\to\infty} \sum_{i=1}^{k} \mathbb{P}\left(\exists x \in P : w^{\mathrm{T}}x \in M_i^k\right). \tag{3}$$

The 1 comes from the fact that there is no randomness in the solution $(0,\ldots,0)$, which is Pareto-optimal with a probability of 1. Weight 0 is excluded from the other, pairwise disjoint intervals M_i^k and from $(0,n]$.

Proof of Claim 6.5. For any k, we have

$$\begin{aligned}\mathbb{E}(q) &= 1 + \sum_{i=1}^{k} \mathbb{E}\left(\left|\left\{x \in P \mid w^{\mathrm{T}}x \in M_i^k\right\}\right|\right) \\ &\geq 1 + \sum_{i=1}^{k} \mathbb{P}\left(\exists x \in P \mid w^{\mathrm{T}}x \in M_i^k\right),\end{aligned}$$

which proves one direction. For the other direction, we estimate the probability that there exists an $i \in \{1,\ldots,k\}$ such that the weight of at least two

different Pareto-optimal solutions $x, y \in \{0,1\}^n$ fall into the same interval M_i^k. Let $d = x - y \neq 0$. Assume without loss of generality that $d_1 = 1$. Given $w_2, \ldots, w_n$, there is an interval of length at most $\frac{n}{k}$ for w_1 such that x and y fall into the same interval. Then, by letting an adversary choose $w_2, \ldots, w_n$ and applying the principle of deferred decisions and for some appropriately chosen number z, we have

$$\mathbb{P}\left(\exists i : w^{\mathrm{T}}x, w^{\mathrm{T}}y \in M_i^k\right) \leq \mathbb{P}\left(w_1 \in \left(z, z + \frac{n}{k}\right]\right) \leq \frac{\phi n}{k}$$

by Lemma 1.1. By a union bound, the probability that there exist two solutions in $\{0,1\}^n \setminus \{(0,\ldots,0)\}$ that are both Pareto-optimal and fall into the same interval is bounded from above by

$$2^{2n} \cdot \frac{\phi n}{k}.$$

If this happens, we use the trivial upper bound of 2^n for the number of Pareto-optimal solutions. Thus, we have

$$\mathbb{E}(q) \leq 1 + \sum_{i=1}^{k} \mathbb{P}\left(\exists x \in P \mid w^{\mathrm{T}}x \in M_i^k\right) + 2^{2n} \cdot \frac{\phi n}{k} \cdot 2^n.$$

We observe that $\lim_{k\to\infty} 2^{2n} \cdot \frac{\phi n}{k} \cdot 2^n = 0$, which yields the other direction and completes the proof. □

To analyze the right-hand side of (3), we want to bound

$$\mathbb{P}\big(\exists x \in P : w^{\mathrm{T}}x \in (t, t+\varepsilon]\big)$$

for any $t \in [0, n]$ and $\varepsilon > 0$. To do this, we define the notions of winner and loser. A solution $x^\star$ is called a winner if $x^\star$ is the most valuable solution that satisfies the weight bound:

$$x^\star = \operatorname{argmax}\left\{p^{\star\mathrm{T}}x \mid x \in \{0,1\}^\star, w^{\mathrm{T}}x \leq t\right\}.$$

For any $t \geq 0$, such a solution $x^\star$ exists since $w^{\mathrm{T}}0 = 0 \leq t$.

A solution x is a loser if $p^{\mathrm{T}}x > p^{\mathrm{T}}x^\star$ and $w^{\mathrm{T}}x > t$. Losers are all solutions that more valuable than a winner $x^\star$, but are not eligible as a winner since they violate the weight constraint. We denote by $\hat{x}$ the loser of the smallest weight:

$$\hat{x} = \operatorname{argmin}\big\{w^{\mathrm{T}}x \mid x \in \{0,1\}^\star, p^{\mathrm{T}}x > p^{\mathrm{T}}x^\star\big\}.$$

If such an $\hat{x}$ does not exist, we define $\hat{x} = \perp$. We now define

$$\Lambda(t) = \begin{cases} w^{\mathrm{T}}\hat{x} - t & \text{if } \hat{x} \neq \perp \text{ and} \\ \perp & \text{if } \hat{x} = \perp. \end{cases}$$

Claim 6.6. *For every t, the following two statements are equivalent:*

(i) There exists a solution $x \in P$ with $w^{\mathrm{T}}x \in (t, t+\varepsilon]$.

(ii) $\Lambda(t) \leq \varepsilon$.

Claims 6.5 and 6.6 together yield the following:

$$\mathbb{E}(q) = 1 + \lim_{k\to\infty} \sum_{i=0}^{k-1} \mathbb{P}\left(\Lambda\left(\frac{ni}{k}\right) \leq \frac{n}{k}\right).$$

The probability $\mathbb{P}(\Lambda(t) \leq \varepsilon)$ remains to be analyzed. To do this, we introduce auxiliary random variables. For $i \in \{1, \ldots, n\}$ and $j \in \{0, 1\}$, we define

- $S^{x_i=j} = \{x \in \{0,1\}^n \mid x_i = j\}$,
- $x^{\star,i} = \operatorname{argmax}\{p^{\mathrm{T}}x \mid x \in S^{x_i=0} \wedge w^{\mathrm{T}}x \leq t\}$, and
- $\hat{x}^i = \operatorname{argmin}\{w^{\mathrm{T}}x \mid x \in S^{x_i=1} \wedge p^{\mathrm{T}}x > p^{\mathrm{T}}x^{\star,i}\}$.

The vector $x^{\star,i}$ is the winner among all solutions that do not contain the i-th item and $\hat{x}^i$ is the loser among those that do contain the i-th element. If no such $\hat{x}^i$ exists, we set $\hat{x}^i = \perp$.

We define Λ^i by using $x^{\star,i}$ and $\hat{x}^i$ in a similar way as we defined Λ:

$$\Lambda^i(t) = \begin{cases} w^{\mathrm{T}}\hat{x}^i - t & \text{if } \hat{x}^i \neq \perp \text{ and} \\ \perp & \text{otherwise.} \end{cases}$$

The difference of Λ^i compared to Λ is that the winner $x^{\star,i}$ does not contain the i-th element, but the loser $\hat{x}^i$ does. The following claim relates Λ to $\Lambda^1, \ldots, \Lambda^n$. After that, we will bound the probability that Λ is small by the probability that some Λ^i is small.

Claim 6.7. *For all w, p, and t, either $\Lambda(t) = \perp$ or there exists an $i \in \{1, \ldots, n\}$ with $\Lambda(t) = \Lambda^i(t)$.*

Proof. Since $w^{\mathrm{T}}\hat{x} > w^{\mathrm{T}}x^{\star}$, there is an $i \in \{1, \ldots, n\}$ such that $\hat{x}_i = 1$ and $x^{\star}_i = 0$. We have $x^{\star} = x^{\star,i}$ since $x^{\star} \in S^{x_i=0}$. We have $\hat{x} = \hat{x}^i$ by the choice of i. Hence, $\Lambda(t) = \Lambda^i(t)$. □

Claim 6.8. *For all $i \in \{1, \ldots, n\}$, $\varepsilon > 0$, and t, we have*

$$\mathbb{P}\left(\Lambda^i(t) \in (0, \varepsilon]\right) \leq \phi\varepsilon.$$

Proof. We assume that all weights except for w_i are fixed arbitrarily. This fixes the weight of all solutions in $S^{x_i=0}$. Thus, $x^{\star,i}$ is determined without revealing w_i.

As the weight w_i affects all solutions in $S^{x_i=1}$ equally, the loser $\hat{x}^i$ does not depend on w_i: There is no randomness in the condition that candidates for $\hat{x}^i$ must be more profitable than $x^{\star,i}$, and we seek the lightest of those solutions. Thus, $\hat{x}^i$ is also determined without revealing w_i. Hence, $\Lambda^i(t) \in (0, \varepsilon]$ is equivalent to $w^{\mathrm{T}}\hat{x}^i - t \in (0, \varepsilon]$. This is equivalent $w_i \in (z, z + \varepsilon]$ for some $z \in \mathbb{R}$. The lemma now follows from Lemma 1.1. □

Using a union bound, we obtain

$$\begin{aligned}\mathbb{P}\big(\Lambda(t) \leq \varepsilon\big) &\leq \mathbb{P}\big(\exists i \in \{1, \ldots, n\} : \Lambda^i(t) \in (0, \varepsilon]\big) \\ &\leq \sum_{i=1}^{n} \mathbb{P}\big(\Lambda^i(t) \in (0, \varepsilon]\big) \leq \phi n \varepsilon.\end{aligned}$$

Putting everything together finishes the proof:

$$\begin{aligned}\mathbb{E}(q) &= 1 + \lim_{k\to\infty} \sum_{i=0}^{k-1} \mathbb{P}\left(\Lambda\left(\frac{ni}{k}\right) \leq \frac{n}{k}\right) \\ &\leq 1 + \lim_{k\to\infty} \sum_{i=0}^{k-1} \frac{\phi n^2}{k} = 1 + \phi n^2.\end{aligned}$$

□

Combining Lemma 6.3 and Theorem 6.4 yields the following bound for the running-time of the Nemhauser-Ullman algorithm.

Corollary 6.9. *Let $p_1, \ldots, p_n$ be arbitrary profits and $w_i, \ldots, w_n$ be drawn according to densities $f_1, \ldots, f_n : [0, 1] \to [0, \phi]$ for $\phi \geq 1$. Then the expected running-time of the Nemhauser-Ullman algorithm is $O(\phi n^3)$.*

Proof. For all $i \in \{1, \ldots, n\}$, the expected size q_i of the restricted Pareto curve P_i is bounded from above by $1 + i^2\phi$ by Theorem 6.4. By linearity of expectation, this yields $\sum_{i=1}^{n} q_i \leq n^3\phi + n \in O(n^3\phi)$. The result follows from Lemma 6.3. □

We remark that Theorem 6.4 is tight in terms of the dependency on n: if we draw the n weights uniformly and independently, then the expected size of the Pareto curve is $\Omega(n^2)$.

7 Successive shortest path algorithm

7.1 Min-cost flows and residual networks

Minimum-cost flows. The minimum-cost flow problem (MCF) is a standard optimization that includes several other optimization problems (such as maximum flow or matching in bipartite graphs) as special cases. We are given a graph $G = (V, E)$, two special nodes $s, t \in V$ called source and sink, respectively, edge costs $c = (c_e)_{e \in E}$, edge capacities $u = (u_e)_{e \in E}$, and a budget b. For the rest of this section, we have $n = |V|$ and $m = |E|$.

The goal is to ship b units of flow from s to t in the cheapest possible way. MCF can be written as a linear optimization problem as follows:

$$\begin{aligned} &\text{minimize} \quad c(f) = \sum_{e \in E} c_e f_e \\ &\text{such that} \quad \sum_{u:e=(v,u) \in E} f_e - \sum_{u:e=(u,v) \in E} f_e = \begin{cases} -b & \text{if } v = t, \\ b & \text{if } v = s, \text{ and} \\ 0 & \text{otherwise,} \end{cases} \\ &\qquad\quad 0 \leq f_e \leq u_e \text{ for all } e \in E. \end{aligned} \tag{4}$$

The first type of constraints are called flow constraints, the second type are the capacity constraints.

If f satisfies all constraints of (4) for some value b, then we call f a feasible b-flow. We denote by $|f| = b$ the amount of flow that is shipped from s to t. Let

$$b_{\max} = \max\bigl\{b \mid \text{there exists a feasible } b\text{-flow}\bigr\}.$$

We denote by $c(f)$ the costs of the flow f. Let $\mathcal{C} : [0, b_{\max}] \to \mathbb{R}$ be the function that maps some value b to the costs of a feasible b-flow of minimum costs:

$$\mathcal{C}(b) = \min\bigl\{c(f) \mid f \text{ is a feasible } b\text{-flow}\bigr\}. \tag{5}$$

We call a feasible b-flow of costs $\mathcal{C}(b)$ an optimal feasible b-flow. We denote by $f_{\max}$ an optimal feasible $b_{\max}$-flow.

We remark that there are more general settings of MCF, where all nodes $v \in V$ have a budget b_v with $\sum_{v \in V} b_v = 0$. We restrict ourselves to one source and one sink for the sake of simplicity, although the successive shortest path algorithm and its smoothed analysis can be extended to the general case.

Residual networks. For an edge $e = (u, v)$, we denote by $e^{-1} = (v, u)$ the corresponding edge in the reverse direction.

Let f be some flow. The residual network $G_f = (V, E^f_{\text{for}} \cup E^f_{\text{back}})$ is the following graph:

- For every edge $e \in E$ with $f_e < u_e$, we have $e \in E^f_{\text{for}}$ with a capacity of $u^f_e = u_e - f_e$ and costs c_e. These edges are called forward edges.
- For every edge $e \in E$ with $f_e > 0$, we have $e^{-1} \in E^f_{\text{back}}$ with a capacity of $u^f_{e^{-1}} = f_e$ and costs $-c_e$. These edges are called backward edges.

(There is some ambiguity in the definition above in the case that both $(u,v),(v,u) \in E$. This ambiguity does not cause any severe technical problems, but it would require notational overhead to avoid them.) The capacities of the edges in the residual network are also called residual capacities.

Let $E^{-1} = \{e^{-1} \mid e \in E\}$. For a subset $F \subseteq E \cup E^{-1}$ of edges, let $F^{-1} = \{e^{-1} \mid e \in F\}$. Let $c(F) = \sum_{e \in F} c_e = \sum_{e \in F \cap E} c_e - \sum_{e \in F \setminus E} c_{e^{-1}}$ be the costs of F. We have $c(F^{-1}) = -c(F)$.

Crucial for min-cost flows and the SSP algorithm is the notion of augmenting a given flow f along a path or a cycle: Let $F \subseteq E^f_{\text{for}} \cup E^f_{\text{back}}$ such that F is either a direction s-t path in G_f or a directed cycle in G_f. We can augment f along F as follows:

- $\delta = \min\{u^f_e \mid e \in F\}$ is the minimum residual capacity of all edges of F. By construction of G_f, we have $\delta > 0$.
- If $e \in F \cap E^f_{\text{for}}$, then $f'_e = f_e + \delta$.
- If $e \in F \cap E^f_{\text{back}}$, then $f'_{e^{-1}} = f_{e^{-1}} - \delta$.
- For all $e \in E$ with $e, e^{-1} \notin F$, we have $f'_e = f_e$.

(The above case distinction does not cover the case that $e, e^{-1} \in F$. In this case, $f'_e = f_e$ as δ is both added and subtracted.) The new flow f' has the following properties:

- f' is a feasible flow.
- If F is a directed cycle, then $|f'| = |f|$.
- If F is a directed s-t path, then $|f'| = |f| + \delta$.
- $c(f') = c(f) + \delta \cdot c(F)$.

In particular, if F is a directed cycle with $c(F) < 0$ and f is a b-flow, then f' is a cheaper b-flow. Instead of adding or subtracting δ, we can add or subtract any other value $\eta \in [0, \delta]$. We say that we augment f along F by η.

The following theorem characterizes optimal flows. A proof can be found in many books on the topic.

Theorem 7.1. *Let f be a feasible b-flow. Then f is an optimal feasible b-flow if and only if $c(C) \geq 0$ for all directed cycles C in G_f.*

Algorithm 2 Successive shortest path algorithm.

input: directed graph $G = (V, E)$ with $s, t \in V$; edge capacities $u = (u_e)_{e \in E}$, edge costs $c = (c_e)_{e \in E}$, budget b

1: start with the empty flow $f_0 = 0$
2: **for** $j = 1, 2, \ldots$ **do**
3: **if** $G_{f_{j-1}}$ does not contain a (directed) s-t path **then**
4: **output** that there does not exist a flow with value b
5: **end if**
6: find a shortest s-t path P_j in $G_{f_{j-1}}$ with respect to the edge costs
7: augment the flow along path P_j to obtain a new flow f_j
8: **end for**
9: choose j such that $|f_{j-1}| \leq b \leq |f_j|$
10: augment f_{j-1} along P_j by $b - |f_{j-1}|$ to obtain $f^\star$
11: **output** $f^\star$

7.2 Successive shortest path algorithm

There exist many different algorithms with different worst-case running-times for MCF. The problem can be solved in time $O\big(m \log n(m + n \log n)\big)$ using the enhanced capacity scaling algorithm [16].

An algorithm that works very well in practice for solving MCF is the successive shortest path algorithm (SSP). Its worst-case running-time is only pseudo-polynomial, and there exist instances on which SSP needs an exponential number of iterations to compute a minimum-cost flow [20]. However, SSP outperforms some polynomial-time algorithms in practice, and SSP is appealing because of its simplicity. SSP is given as Algorithm 2.

We note the following: In the main loop of the algorithm, the budget b is ignored. We augment until we have a $b_{\max}$-flow. In practice, one would of course stop if $|f_j| \geq b$. In this case, we would not add or subtract the maximum possible value to the edges of P_j, but only the flow that remains to obtain a feasible b-flow (Lines 9 to 11).

Notation and remarks. Let $\mathcal{F}^c = \{f_0, f_1, \ldots\}$ be the set of all flows that SSP computes in its for loop. Note that this set depends on c. This superscript is important as we consider modified edge costs in the probabilistic analysis.

Let $b_j = |f_j|$. The flow f_0 is called the empty flow. If $j_{\max}$ is the last flow computed in the for loop of SSP, then $f_{\max} = f_{j_{\max}}$ is a maximum s-t flow of minimum costs. SSP needs $j_{\max}$ iterations.

Note that we did not specify a tie-breaking rule for the case that there are multiple paths with the same costs. This is not important for the probabilistic analysis, as all paths will have different costs with a probability of 1.

Let f_{j-1} and f_j be two flows encountered by SSP, and let P_j be the path along which SSP augmented f_{j-1} to obtain f_j. We call P_j the next path of f_{j-1} and the previous path of f_j. These terms become important when we reconstruct flows without knowing all edge costs.

Let f be some flow. We say that an edge $e \in E^f_{\text{for}} \cup E^f_{\text{back}}$ is empty in G_f

- $e \in E^f_{\text{for}}$ and $f_e = 0$ or
- $e \in E^f_{\text{back}}$ and $f_{e^{-1}} = u_{e^{-1}}$.

Correctness and properties. The SSP algorithm correctly computes a flow of minimum costs according to the following theorem.

Theorem 7.2. *The flows f_j computed by SSP are optimal feasible b_j-flows, and $f^\star$ is an optimal b-flow.*

The following lemma is a crucial not only for the correctness of SSP, but also for the probabilistic analysis in the next section.

Lemma 7.3. *Let $P_1, P_2, \ldots$ be the paths that SSP constructs. Then $c(P_j) \leq c(P_{j+1})$ for all j.*

7.3 Probabilistic analysis of SSP

The results of this section are by Brunsch et al. [6].

Probabilistic model. We use the one-step model to analyze the SSP: An adversary specifies

- the graph $G = (V, E)$ as well as $s, t \in V$,
- the edge capacities $u = (u_e)_{e \in E}$,
- the budget b (although this does not influence the analysis, as we will see), and
- density functions $g_e : [0, 1] \to [0, \phi]$ for the edge costs for all $e \in E$.

We obtain our MCF instance by drawing c_e independently according to g_e.

Note that we only perturb the objective function. The solution space, namely the flow polytope specified by the graph and the capacities, remains adversarial.

Outline. We identify iteration j of SSP and a flow f_j computed by SSP with the costs $c(P_j)$ of the previous path of f_j. In this way, we obtain a sequence $(c(P_1), c(P_2), \ldots)$ of real numbers. This sequence is monotonically increasing by Lemma 7.3. It is even strictly monotonically increasing since all paths have pairwise different costs with a probability of 1 since the costs come from a continuous probability distribution. All numbers $c(P_j)$ come from the interval $[0, n]$ as all paths are simple and the edge costs are from $[0, 1]$. (The residual networks can contain edges of negative costs, but G_{f_0} contains only edges of non-negative costs and path costs cannot decrease by Lemma 7.3.

In the same way as for the knapsack problem, we partition the interval $(0, n]$ into k small intervals of size n/k. Then we estimate the probability that SSP encounters a path with costs in a certain interval. For sufficiently large k, there is at most one path in every interval $(d, d + \varepsilon]$ for $\varepsilon = n/k$.

Assume that $\ell_j \in (d, d + \varepsilon]$ for some j. If there is at most one path with costs in this interval, then $\ell_{j-1} \leq d$. We show that the path P_j along which we augment to obtain f_j must contain an empty edge e. We show that we can reconstruct f_{j-1} without knowing the costs c_e of e. Assume that we know $e = (u, v)$. This allows us to apply the principle of deferred decisions: First, we reveal the costs of all edges except for e. This allows to identify P_j: the flow f_{j-1} specifies $G_{f_{j-1}}$, then P_j consists of a shortest s-u path in $G_{f_{j-1}}$ followed by $e = (u, v)$ followed by a shortest v-t path in $G_{f_{j-1}}$.

The costs $c(P_j)$ can fall into $(d, d + \varepsilon]$ only if c_e falls into some interval of size ε, which happens with a probability of at most $\varepsilon\phi$.

Since we do not know which edge e is the empty edge of P_j, we take a union bound over the $2m$ possible choices for e. This yields an upper bound of $2m\varepsilon\phi$ for the probability that SSP encounters a path with costs in the interval $(d, d + \varepsilon]$.

Preparation. Before starting the actual analysis, we set up some more notation and collect a few properties of the flow network, the paths encountered by SSP, and the function $\mathcal{C}$.

With a probability of 1, we have the following properties:

- All paths have pairwise different costs.
- SSP uses every path at most once.
- For all flows f, the residual network does not contain a directed cycle of costs 0.

These properties imply the following observations:

1. For every $b \in [0, b_{\max}]$, there is a unique optimal feasible b-flow.
2. $c(P_j) < c(P_{j+1})$ for all j, where P_j and P_{j+1} are paths constructed by SSP (this is a small but crucial strengthening of Lemma 7.3).

The optimality of the flows computed by SSP yields the following facts about the function $\mathcal{C}$ defined in (5)):

1. The function $\mathcal{C}$ is continuous, piecewise linear, and monotonically increasing.
2. $\mathcal{C}$ is convex. This follows from $c(P_{j-1}) \leq c(P_j)$ for all j.
3. If $\mathcal{C}$ is not differentiable at some point b, then $b = b_j$ for some j.
4. Between b_{j-1} and b_j, the function $\mathcal{C}$ is linear with slope $c(P_j)$.
5. Since $c(P_{j-1}) < c(P_j)$ with a probability of 1, the points where $\mathcal{C}$ is not differentiable are exactly $b_1, b_2, \ldots$. We call the points the breakpoints of $\mathcal{C}$.

Similar to the analysis of the size of the Pareto curve in Section 6.2, we analyze the number of iterations that SSP needs by analyzing the probability that there exist paths that SSP uses in certain small intervals. Since all paths encountered by SSP are simple, the costs of all paths fall into the interval $[0, n]$. With a probability of 1, we do not have a path of costs 0. We split $(0, n]$ into k intervals $M_i^k = \left(\frac{n(i-1)}{k}, \frac{ni}{k}\right]$ of size n/k in the same way as we did it in the analysis of the knapsack problem.

Let T be the (random) number of iterations that SSP needs. Then we obtain

$$\mathbb{E}(T) = \lim_{k\to\infty} \sum_{i=1}^{k} \mathbb{P}\left(\exists j : c(P_j) \in M_i^k\right). \tag{6}$$

(The proof that both sides are indeed equal is quite similar to the proof that (3) holds. We bound the probability that there are two paths in any residual network that fall into the same interval of size n/k and use a trivial upper bound in terms of n and m in case that two paths do fall into the same interval.)

For a feasible flow f (encountered by an execution of SSP), let $\ell_-^c(f)$ be the costs of the path along which we augmented to get f, and let $\ell_+^c(f)$ be the costs of the shortest s-t path in the residual network G_f. We set $\ell_-^c(f_0) = 0$, where f_0 is the empty flow, and $\ell_+^c(f_{\max}) = \infty$. The superscript c is important here, as we will use modified cost functions in the analysis.

We have $c(P_{j-1}) = \ell_-^c(f_{j-1}) \leq \ell_+^c(f_{j-1}) = \ell_-^c(f_j) = c(P_j)$. The inequality is strict with a probability of 1.

Analysis. The following observation is crucial for applying the principle of deferred decisions: every path used by SSP contains at least one empty edge. This edge can be a forward edge or a backward edge of the residual network. In the analysis, we fix the costs of all other edges and only use the randomness of the costs of this empty edge.

Lemma 7.4. *With a probability of* 1*, the following holds: For all* j*, the path* P_j *computed by SSP contains at least one empty edge.*

Proof. We prove a slightly stronger statement: in all residual graphs G_{f_j} encountered by SSP, all s-t paths contain an empty edge. All edges in G_{f_0} are empty. Hence, the lemma holds for $j = 1$.

Consider any path P in the residual network G_{f_j} for some arbitrary j. We have $c(P) > c(P_j)$ with a probability of 1. Assume to the contrary that P does not contain any empty edge. This means that G_{f_j} contains all edges of P in both directions. Let $Q = P^{-1} \cup P_j$. If an edge occurs in both P^{-1} and P_j, then it is contained twice in Q. The set Q forms a (not necessarily simple) cycle with $c(Q) < 0$. We can decompose Q into simple cycles. At least one of these simple cycles has negative costs. This contradicts the optimality of f_j by Theorem 7.1. □

Lemma 7.5. *Let* $f \in \mathcal{F}^c$ *with* $f \neq f_{\max}$ *be a flow computed by SSP with* $\ell^c_-(f) < \ell^c_+(f)$*. Let* $d \in [\ell^c_-(f), \ell^c_+(f)]$ *be arbitrary. Let* e *be an empty edge of the next path of* f*. Define edge costs* c' *as follows:*

- $c'_e = 1$ *if* $e \in E^f_{\text{for}}$.
- $c'_e = 0$ *if* $e \in E^f_{\text{back}}$.
- *All other edge costs are the same as in* c.

Then $f \in \mathcal{F}^{c'}$ *and* $\ell^{c'}_-(f) \leq \ell^c_-(f) \leq d < \ell^c_+(f) <\leq \ell^{c'}_+(f)$.

Proof. Let $\mathcal{C}$ and $\mathcal{C}'$ be the cost functions mapping budgets b to the costs of an optimal b-flow under c and c', respectively. We define an auxiliary cost function $\tilde{\mathcal{C}}$ depending on whether $e \in E^f_{\text{for}}$ or $e \in E^f_{\text{back}}$.

If $e \in E^f_{\text{for}}$, then $\tilde{\mathcal{C}} = \mathcal{C}'$. Since we only increase the costs of e, we have $\tilde{\mathcal{C}}(b) \geq \mathcal{C}(b)$ for all b. As $f_e = 0$, we have $\tilde{\mathcal{C}}(|f|) = \mathcal{C}(|f|)$.

If $e \in E^f_{\text{back}}$, then we define $\tilde{\mathcal{C}}$ by $\tilde{\mathcal{C}}(b) = \mathcal{C}' + u_{e^{-1}} c_{e^{-1}}$. The function $\tilde{\mathcal{C}}$ is piecewise linear and has the same breakpoints and the same slopes as $\mathcal{C}'$. Since the flow on edge e^{-1} is between 0 and $u_{e^{-1}}$, reducing the edge costs $c_{e^{-1}}$ to $c_{e^{-1}} = 0$ reduces the costs of flows by an amount between 0 and $u_{e^{-1}} c_{e^{-1}}$. Hence, $\tilde{\mathcal{C}}(b) \geq \mathcal{C}(b)$ for all b. Since $f_{e^{-1}} = u_{e^{-1}}$, we have $\tilde{\mathcal{C}}(|f|) = \mathcal{C}(|f|)$.

By construction, the left-hand derivative of $\tilde{\mathcal{C}}$ is at most the left-hand derivative of $\mathcal{C}$, and the right-hand derivative of $\tilde{\mathcal{C}}$ is at least the right-hand derivative of $\mathcal{C}$. We conclude that $f \in \mathcal{F}^{c'}$.

The inequalities stated in the lemma follow from their correspondence to slopes of $\mathcal{C}$ and $\tilde{\mathcal{C}}$. □

Lemma 7.6. *Let* $f \in \mathcal{F}^c$*, and let* e *be an empty edge of the next path of* f*. Let* $d \in [\ell^c_-(f), \ell^c_+(f)]$*. Then* Reconstruct$(e, d)$ *outputs* f*.*

Algorithm 3 Reconstruction algorithm.

Reconstruct (e, d)
input: e (with $e \in E$ or $e^{-1} \in E$); $d \in [0, n]$
1: **if** $e \in E$ **then**
2: change costs of e to 1
3: **else if** $e^{-1} \in E$ **then**
4: change costs of e to 0
5: **end if**
6: run SSP on this modified instance; stop when the costs of the path P computed by SSP exceeds t
7: **output** the flow before augmenting along P

Proof. Using Lemma 7.5 yields $f \in \mathcal{F}^{c'}$ and $\ell_-^{c'}(f) \leq d < \ell_+^{c'}(f)$. By the monotonicity of the costs of the paths encountered by SSP, this shows that the reconstruction algorithm stops once it encounters f and outputs f. □

Lemma 7.7. *Fix any $d \geq 0$ and $\varepsilon > 0$. The probability that there exists a flow $f \in \mathcal{F}^c \setminus \{f_0\}$ with $\ell_-^c(f) \in (d, d + \varepsilon]$ is bounded from above by $2m\varepsilon\phi$.*

Proof. Among all flows f with previous path costs $\ell_-^c(f) \in (d, d + \varepsilon]$, let $\hat{f}$ be the flow with smallest value $\ell_-^c(\hat{f})$. Let $f^\star$ be the flow preceding $\hat{f}$. The flow $f^\star$ exists because $\hat{f}$ is not the empty flow.

By the choice of $\hat{f}$ and $f^\star$, we have

$$\ell_-^c(f^\star) \leq d < \ell_-^c(\hat{f}).$$

The shortest path in the residual network $G_{f^\star}$ contains at least one empty edge $e = (u, v)$ according to Lemma 7.4. Applying the reconstruction algorithm (Algorithm 3) with input e and d yields $f^\star$ according to Lemma 7.6.

The shortest s-t path in the residual network $G_{f^\star}$ is equal to $\ell_-^c(\hat{f})$. It consists (given that it uses the empty edge e) of a shortest s-u path followed by e follows by a shortest v-t path.

Let z be the costs of the shortest s-u path plus the costs of the shortest v-t path in $G_{f^\star}$. Then $\ell_-^c(\hat{f}) \in (d, d + \varepsilon]$ only if $c(e) \in (d - z, d - z + \varepsilon]$. The probability of this event is bounded from above by $\varepsilon\phi$.

Since we do not know the edge e, we take a union bound over the $2m$ possible choices of $e \in E \cup E^{-1}$. □

Theorem 7.8. *The smoothed number of augmentation steps that SSP performs is bounded from above by $2mn\phi$.*

Proof. Since all paths encountered by SSP are simple, their costs increase monotonically, and the edge costs are from $[0, 1]$, all paths have costs from the interval $[0, n]$.

Let T be the number of augmentation steps that SSP needs. We have

$$\begin{aligned}\mathbb{E}(T) &= \lim_{k\to\infty} \sum_{i=1}^{k} \mathbb{P}\big(\text{SSP encounters a flow of costs in } M_i^k\big) \\ &\leq \lim_{k\to\infty} \sum_{i=1}^{k} \frac{2m\phi n}{k} = 2mn\phi\end{aligned}$$

by Lemma 7.7. □

Corollary 7.9. *The smoothed running-time of the SSP algorithm is at most* $O(mn\phi(m + n\log n))$.

Proof. The most expensive operation in each iteration is computing the shortest paths. This can be done in time $O(m + n\log n)$. (Note that edge costs can be negative in residual graphs, but there are no cycles of negative costs by Theorem 7.1 and 7.2. Because of this, we can maintain node potentials $\pi : V \to \mathbb{R}$ such that the costs $c'_e = c_e - \pi(u) + \pi(v)$ are non-negative for each edge $e = (u, v)$ of the residual network. Modifying the edge costs in this way does not change the shortest path.) □

Theorem 7.10. *For given positive integers* n *and* $m \in \{n, \ldots, n^2\}$ *and* $\phi \leq 2^n$*, there exists a minimum-cost flow network with* $O(n)$ *nodes and* $O(m)$ *edges and random edge costs drawn according to density functions bounded by* ϕ *on which SSP requires* $\Omega(m\phi\min\{n, \phi\})$ *iterations with a probability of* 1.

8 Clustering under approximation stability

8.1 Motivation and definition

In this section, which is based on work by Balcan, Blum, and Gupta [3], we do not do smoothed analysis, but consider a completely different concept called "approximation stability". Approximation stability is an example of a property of instances that allows us to find good solutions. The main idea of approximation stability applied to clustering is as follows: although we try to optimize an objective function when clustering, we are not really interested in this objective function. Instead, we would like to get a clustering that is close to the "true" clustering of that data. Since we do not know this true clustering, we use an objective function as a proxy. The hope is that if clustering is good under the objective function, then it is close to the true clustering.

We consider clustering with k-median objective: We are given n data points $X \subseteq \mathbb{R}^d$ and a $k \in \mathbb{N}$. Our goal is to find a clustering $\mathcal{C}$ consisting of

k clusters $C_1, \dots, C_k$ that partition the data points X together with cluster centers $c_1, \dots, c_k$ such that

$$w(\mathcal{C}) = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - c_i\| \tag{7}$$

is minimized.

We assume that there is an underlying target clustering $\mathcal{C}^{\mathrm{T}}$ with clusters $C_1^{\mathrm{T}}, \dots, C_k^{\mathrm{T}}$, which we do not know. Our goal is to find a clustering $\mathcal{C}$ that is as close as possible to $\mathcal{C}^{\mathrm{T}}$. Thus, we have to define what "close" means. Let $\mathcal{C}$ and $\mathcal{C}'$ be two clusterings with clusters $C_1, \dots, C_k$ and $C'_1, \dots, C'_k$, respectively. Then we define

$$\mathrm{dist}(\mathcal{C}, \mathcal{C}') = \min_{\pi \in S_k} \frac{1}{n} \cdot \sum_{i=1}^{k} \left| C_i \setminus C'_{\pi(i)} \right|, \tag{8}$$

where S_k denotes the set of all permutations of $\{1, \dots, k\}$. We call $\mathcal{C}$ and $\mathcal{C}'$ ε-close if $\mathrm{dist}(\mathcal{C}, \mathcal{C}') < \varepsilon$.

If $\mathcal{C}$ and $\mathcal{C}'$ are ε-close and $|C_i| \geq 2\varepsilon n$ for all $i \in \{1, \dots, k\}$, then we have

$$|C_i - C_{\pi(i)}| \geq |C_i| - (\varepsilon n - 1) \geq \frac{1}{2} \cdot |C_i|$$

for the permutation π that minimizes in (8). In this case, π is unique, and we call π the optimal permutation. We say that $\mathcal{C}$ and $\mathcal{C}'$ agree on $x \in X$ if $x \in C_i \cap C_{\pi(i)}$ for some i. Otherwise, we say that $\mathcal{C}$ and $\mathcal{C}'$ disagree on x.

In the following, we denote by $\mathcal{C}^\star$ a clustering that minimizes (7). We call $\mathcal{C}^\star$ an optimal clustering. It has clusters $C_1^\star, \dots, C_k^\star$ and centers $c_1^\star, \dots, c_k^\star$.

Definition 8.1. We say that X is (c, ε)-approximation stable with respect to $\mathcal{C}^{\mathrm{T}}$ if all clusterings $\mathcal{C}$ satisfy the following condition: If $w(\mathcal{C}) \leq c \cdot w(\mathcal{C}^\star)$, then $\mathrm{dist}(\mathcal{C}, \mathcal{C}^{\mathrm{T}}) < \varepsilon$.

Note that the distance is measured to $\mathcal{C}^{\mathrm{T}}$ and the objective value is compared to $\mathcal{C}^\star$. Note also that if we have a polynomial-time c-approximation algorithm for our clustering problem, then we can use it to compute a clustering that is ε-close to $\mathcal{C}^{\mathrm{T}}$. Thus, the interesting case is if we can compute a clustering $\mathcal{C}$ that is ε-close to $\mathcal{C}^{\mathrm{T}}$ in the case that there is no polynomial-time c-approximation algorithm.

In the following, we often write $c = 1 + \alpha$ and consider implications of $(1 + \alpha, \varepsilon)$-approximation stability in terms of α and ε. We also often drop the reference to $\mathcal{C}^{\mathrm{T}}$.

We remark that ε, α, and k do not need to be constants for our algorithms to run in polynomial time. All three parameters can also depend on the number n of data points.

8.2 Basic properties

The two clusterings $\mathcal{C}^{\mathrm{T}}$ and $\mathcal{C}^\star$ do not have to be identical, but we make the following observations. In the following, let $\varepsilon^\star = \mathrm{dist}(\mathcal{C}^\star, \mathcal{C}^{\mathrm{T}})$.

Observation 8.2. If X is (c, ε)-approximation stable, then

(i) $\varepsilon^\star < \varepsilon$ and

(ii) X is $(c, \varepsilon + \varepsilon^\star)$-approximation stable with respect to $\mathcal{C}^\star$.

Also the following observation is simple but useful.

Observation 8.3. Let $\mathcal{C}$ and $\mathcal{C}'$ be clusterings such that $\mathcal{C}'$ is obtained from $\mathcal{C}$ by assigning ℓ points to a different cluster. Then we have the following properties:

(i) $\mathrm{dist}(\mathcal{C}, \mathcal{C}') \leq \ell/n$.

(ii) If $|C_i| \geq 2\ell$ for all $i \in \{1, \ldots, k\}$, then $\mathrm{dist}(\mathcal{C}, \mathcal{C}') = \ell/n$.

The fact that Part (ii) of Observation 8.3 does not hold without the lower bound on the size of the clusters makes it more difficult to analyze implications of (c, ε)-approximation stability in the presence of small clusters. To keep it simple, we restrict ourselves to the case that all clusters are large, but postpone the exact definition of large.

We define $w(x) = \min_{i \in \{1,\ldots,k\}} \|x - c_i^\star\|$ to be the distance of x to its closest cluster center. This means that $w(\mathcal{C}^\star) = \sum_{x \in X} w(x)$. We define $w_2(x)$ to be the distance of x to its second-closest cluster center. If $w_2(x) = w(x)$, then there are two cluster centers that are closest to x. Let

$$w_{\mathrm{avg}} = \frac{1}{n} \cdot \sum_{x \in X} = \frac{w(\mathcal{C}^\star)}{n}$$

be the average of the closest distances.

The quantity $w_2(x) - w(x)$ is of particular interest: if it is too small, then we can reassign x without incurring too high costs. If $w_2(x) - w(x)$ is small for too many x, then we can reassign many points without increasing the costs by much. The following lemma bounds the number of points for which this quantity is small.

Lemma 8.4. *Assume that X is $(1 + \alpha, \varepsilon)$-approximation stable and that $|C_i^\star| \geq 2\varepsilon n$ for all $i \in \{1, \ldots, k\}$. Then we have the following properties:*

(a) Less than $(\varepsilon - \varepsilon^\star) \cdot n$ points $x \in X$ on which $\mathcal{C}^{\mathrm{T}}$ and $\mathcal{C}^\star$ agree satisfy $w_2(x) - w(x) < \frac{\alpha w_{\mathrm{avg}}}{\varepsilon}$.

(b) At most $\frac{t\varepsilon n}{\alpha}$ points $x \in X$ have $w(x) \geq \frac{\alpha w_{\mathrm{avg}}}{t\varepsilon}$.

Proof. We first prove (a). Assume to the contrary that there are at least $(\varepsilon - \varepsilon^\star) \cdot n$ points x on which $\mathcal{C}^{\mathrm{T}}$ and $\mathcal{C}^\star$ agree such that $w_2(x) - w(x) < \frac{\alpha w_{\mathrm{avg}}}{\varepsilon}$. By taking $\mathcal{C}^\star$ and reassigning these points to the cluster with their second-closest center increases the costs by at most $\frac{\alpha w_{\mathrm{avg}}}{\varepsilon} \cdot (\varepsilon - \varepsilon^\star) \cdot n \leq \alpha w(\mathcal{C}^\star)$, we obtain a clustering $\mathcal{C}'$. We have $\mathrm{dist}(\mathcal{C}^{\mathrm{T}}, \mathcal{C}') = \varepsilon$ by Observation 8.3(ii) and since $\mathcal{C}^{\mathrm{T}}$ and $\mathcal{C}^\star$ agree on all points that we reassign. This new clustering contradicts $(1+\alpha, \varepsilon)$-approximation stability.

Part b follows by Markov's inequality. □

In the following, let $d_{\mathrm{crit}} = \frac{\alpha w_{\mathrm{avg}}}{5\varepsilon}$. A point $x \in X$ is called good if $w(x) < d_{\mathrm{crit}}$ and $w_2(x) - w(x) \geq 5 d_{\mathrm{crit}}$. Else, x is called bad. Let $Y_i \subseteq C_i^\star$ be the set of good points in the i-th cluster of $\mathcal{C}^\star$, and let $B = X \setminus \left(\bigcup_{i=1}^k Y_i\right)$ be the set of bad points. There cannot be too many bad points according to the following lemma.

Lemma 8.5. *Assume that X is $(1+\alpha, \varepsilon)$-approximation stable and that $|C_i^{\mathrm{T}}| \geq 2\varepsilon n$ for all $i \in \{1, \ldots, k\}$. Then $|B| \leq (1 + \frac{5}{\alpha}) \cdot \varepsilon n$.*

Proof. By Lemma 8.4(a), there are less than $(\varepsilon - \varepsilon^\star) \cdot n$ points x on which $\mathcal{C}^{\mathrm{T}}$ and $\mathcal{C}^\star$ agree with $w_2(x) - w(x) < 5 d_{\mathrm{crit}}$. There are at most $\varepsilon^\star n$ points on which $\mathcal{C}^{\mathrm{T}}$ and $\mathcal{C}^\star$ disagree. Using Lemma 8.4(b) with $t = 5$ bounds the number of points x with $w(x) \geq d_{\mathrm{crit}}$ by $\frac{5\varepsilon}{\alpha} \cdot n$. □

The τ-threshold graph $G_\tau = (X, E_\tau)$ has the data points X as nodes, and $\{x, y\} \in E_\tau$ if $\|x - y\| < \tau$. We have the following property.

Lemma 8.6. *Let X be $(1+\alpha, \varepsilon)$-approximation stable, and let $\tau = 2 d_{\mathrm{crit}}$. Then the τ-threshold graph G_τ has the following properties:*

(i) For all $i \in \{1, \ldots, k\}$ and $x, y \in Y_i$, we have $\{x, y\} \in E_\tau$.

(ii) For all $i, j \in \{1, \ldots, k\}$ with $i \neq j$ and all $x \in Y_i$ and $y \in Y_j$, we have $\{x, y\} \notin E_\tau$. Furthermore, x and y do not have any common neighbors in G_τ.

Proof. By the triangle inequality and because x and y are good, we have $\|x - y\| \leq \|x - c_i^\star\| + \|y - c_i^\star\| < 2 d_{\mathrm{crit}}$, which shows Part (i).

To see Part (ii), we observe that the distance of x to any cluster center of $\mathcal{C}^\star$ other than $c_i^\star$ is at least $5 d_{\mathrm{crit}}$. This holds in particular for $c_j^\star$. Furthermore, $\|y - c_j^\star\| < d_{\mathrm{crit}}$ since y is good. Thus, by the triangle inequality, we have

$$\|x - y\| \geq \|x - c_j^\star\| - \|y - c_j^\star\| \geq 5 d_{\mathrm{crit}} - d_{\mathrm{crit}} = 2\tau.$$

Since each edge in G_τ connects points at a distance of less than τ, the two points x and y cannot have any common neighbors. □

Let $N_\tau(Y_i)$ denote all nodes incident to at least one node in Y_i. The graph G_τ has a simple structure:

(P1) Each Y_i is a clique.

(P2) $N_\tau(Y_i) \setminus Y_i \subseteq B$, i.e., the neighbors of Y_i outside of Y_i lie entirely in B.

(P3) $N_\tau(Y_i) \cap N_\tau(Y_j) = \emptyset$ for $i \neq j$ because nodes from Y_i and Y_j do not share any neighbors.

8.3 Algorithm

We now describe an algorithm to find a clustering that is ε-close to $\mathcal{C}^{\mathrm{T}}$. We do this in two steps. First, we assume that the algorithm knows w_{avg}. After that, we describe how to get rid of this assumption.

Lemma 8.7. *There is a polynomial-time algorithm that does the following: Let X be $(1+\alpha, \varepsilon)$-approximation stable, and let $\tau = 2d_{\mathrm{crit}}$. Let $b \in \mathbb{N}$ with $b \geq |B|$ and $|Y_i| \geq b+2$ for all $i \in \{1, \dots, k\}$. Then the algorithm computes a k-clustering $\mathcal{C}$ with each Y_i contained in a distinct cluster.*

Proof. We construct an auxiliary graph $H = (X, E)$ with $\{x, y\} \in E$ if x and y have at least b common neighbors. in G_τ. By Lemma 8.6(1) and since $|Y_i| \geq b+2$ for all i, we have $\{x, y\} \in E$ for all i and $x, y \in Y_i$.

Fix any $i \in \{1, \dots, k\}$ and consider any $x \in Y_i \cup N_\tau(Y_i)$ and $y \notin Y_i \cup N_\tau(Y_i)$. By (P3), all common neighbors of x and y lie in B. Since at least one of x and y lies itself in B, the two nodes x and y have at most $b-1$ common neighbors. This implies that $\{x, y\} \notin E$.

We conclude that each X_i lies in a distinct component of H. Such components can also contains points from B. There can also be components consisting solely of points from B. Since $|Y_i| > |B|$ for all i, we can obtained the clustering claimed by taking the largest k components of H as the k clusters and adding all remaining points arbitrarily to one of the clusters. □

Now we are prepared to find a clustering that is ε-close to $\mathcal{C}^{\mathrm{T}}$, given that all clusters of $\mathcal{C}^{\mathrm{T}}$ are sufficiently large and that we know w_{avg}.

Theorem 8.8. *There is a polynomial-time algorithm that does the following: given a $(1+\alpha, \varepsilon)$-approximation stable X such that $|C_i^{\mathrm{T}}| \geq (3+\frac{10}{\alpha}) \cdot \varepsilon n + 2$ for all $i \in \{1, \dots, k\}$ and w_{avg}, it computes a clustering $\tilde{\mathcal{C}}$ with $\mathrm{dist}(\tilde{\mathcal{C}}, \mathcal{C}^{\mathrm{T}}) < \varepsilon$.*

Proof. We define $b = (1+\frac{5}{\alpha}) \cdot \varepsilon n$. Then $|C_i^{\mathrm{T}}| \geq 2b + \varepsilon n + 2$ for all $i \in \{1, \dots, k\}$ by assumption. Since $\mathrm{dist}(\mathcal{C}^\star, \mathcal{C}^{\mathrm{T}}) = \varepsilon^\star < \varepsilon$, we have $|C_i^\star| \geq 2b+2$ for all $i \in \{1, \dots, k\}$. By Lemma 8.5, we have $|B| \leq b$. We conclude that

$$|Y_i| = |C_i^\star \setminus B| \geq b + 2$$

for all $i \in \{1, \dots, k\}$.

Given w_{avg}, we construct G_τ for $\tau = 2d_{\text{crit}}$ (the value of d_{crit} can be determined using w_{avg}). By Lemma 8.7, we can compute a clustering $\mathcal{C}$ such that each X_i is contained in a distinct cluster of $\mathcal{C}$. This clustering $\mathcal{C}$ differs from $\mathcal{C}^\star$ only in B. Hence, by the triangle inequality,

$$\text{dist}(\mathcal{C}, \mathcal{C}^{\text{T}}) \leq \varepsilon^\star + \frac{|B|}{n} = O\left(\varepsilon + \frac{\varepsilon}{\alpha}\right).$$

We have to modify the clustering slightly to obtain a clustering $\tilde{\mathcal{C}}$ that is ε-close to $\mathcal{C}^{\text{T}}$.

We call a point x "red" if $w_2(x) - w(x) < 5d_{\text{crit}}$. We call x "yellow" if x is not red and $w(x) \geq d_{\text{crit}}$. We call x "green" if x is neither red nor yellow. This means that x is green if x is contained in some Y_i, and B is partitioned into yellow and red nodes.

We know that $\mathcal{C}$ agrees with $\mathcal{C}^\star$ on the green points. Without loss of generality, we assume that $Y_i \subseteq C_i$. We construct a new clustering $\tilde{\mathcal{C}}$ that agrees with $C^\star$also on the yellow points. To do this, let $d_j(x)$ be the median of the distances of x to all points in C_j. Then insert x into $\tilde{C}_i$ if $i = \text{argmin}_{j \in \{1,\dots,k\}} d_j(x)$. Now consider any point x that is either green or yellow. Since x is not red, we have $w_2(x) - w(x) \geq 5d_{\text{crit}}$.

Assume that $x \in C_i^\star$. For every green point $g_1 \in C_i^\star$, we have $\|x - g_1\| \leq w(x) + d_{\text{crit}}$. For every green point $g_2 \notin C_i^\star$, we have $\|x - g_2\| \geq w_2(x) - d_{\text{crit}} \geq w(x) + 4d_{\text{crit}}$. Hence, $\|x - g_1\| < \|x - g_2\|$ for all such green points g_1 and g_2. Since each cluster of $\mathcal{C}$ has a strict majority of green points (even with x removed), $d_j(x)$ is determined by the distance of x to a green point of C_j. This means that $\tilde{\mathcal{C}}$ and $\mathcal{C}^\star$ agree on all green and yellow points.

There are less than $(\varepsilon - \varepsilon^\star)n$ red points on which $\mathcal{C}^{\text{T}}$ and $\mathcal{C}^\star$ agree by Lemma 8.4(a). The two clusterings $\mathcal{C}^{\text{T}}$ and $\tilde{\mathcal{C}}$ might disagree on all of them. Hence,

$$\text{dist}(\tilde{\mathcal{C}}, \mathcal{C}^{\text{T}}) < (\varepsilon - \varepsilon^\star) + \varepsilon^\star = \varepsilon.$$

□

Now we explain how to get rid of the assumption that we know w_{avg}. The price we have to pay is a slightly larger lower bound for the size of all clusters $C_1^{\text{T}}, \dots, C_k^{\text{T}}$.

Theorem 8.9. *There is a polynomial-time algorithm that does the following: given a $(1+\alpha, \varepsilon)$-approximation stable X such that $|C_i^{\text{T}}| \geq (4 + \frac{15}{\alpha}) \cdot \varepsilon n + 2$ for all $i \in \{1, \dots, k\}$ and w_{avg}, it computes a clustering $\tilde{\mathcal{C}}$ with $\text{dist}(\tilde{\mathcal{C}}, \mathcal{C}^{\text{T}}) < \varepsilon$.*

Proof. We run the algorithm described in the proof of Theorem 8.8 with different "guesses" of w_{avg}, starting with $w_{\text{avg}} = 0$ (in this case, G_τ does not contain any edges). In every iteration, we increase w_{avg} to the smallest

value such that the corresponding G_τ contains at least one new edge. Thus, we have at most $O(n^2)$ iterations.

In every iteration, we construct H as in the proof of Theorem 8.8 with the current guess of w_{avg}. We use the current value of w_{avg} as the true value if the following conditions are satisfied:

- The k largest components of H contain all but at most $b = (1 + \frac{5}{\alpha}) \cdot \varepsilon n$ nodes.
- Each of the k largest components has size at least $b + 1$.

We observe that our guess of w_{avg} is either correct or too small, but not too large. This means that the graph H obtained from our guess can only have fewer edges than the "true" graph as in the proof of Theorem 8.8. But since the largest k components miss at most b points, each Y_i is contained in its own component. Hence, we classify all good points correctly. We might misclassify all bad points, but there are at most b such bad points, and we might fail to classify at most b points from Y_i's, which do not lie in the largest k components. Still, each cluster contains at least $b + 2$ correctly clustered green points (in the terminology of the proof of Theorem 8.8) and at most b misclassified points. Hence, by running the final step, we obtain a clustering the correctly clusters all non-red points. □

References

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

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

[3] Maria-Florina Balcan, Avrim Blum, and Anupam Gupta. Clustering under approximation stability. *Journal of the ACM*, 60(2):8, 2013.

[4] René Beier, Heiko Röglin, and Berthold Vöcking. The smoothed number of Pareto optimal solutions in bicriteria integer optimization. In Matteo Fischetti and David P. Williamson, editors, *Proc. of the 12th Int. Conf. on Integer Programming and Combinatorial Optimization (IPCO)*, volume 4513 of *Lecture Notes in Computer Science*, pages 53–67. Springer, 2007.

[5] René Beier and Berthold Vöcking. Random knapsack in expected polynomial time. *Journal of Computer and System Sciences*, 69(3):306–329, 2004.

[6] Tobias Brunsch, Kamiel Cornelissen, Bodo Manthey, Heiko Röglin, and Clemens Rösner. Smoothed analysis of the successive shortest path algorithm. *SIAM Journal on Computing*, 44(6):1798–1819, 2015.

[7] Barun Chandra, Howard Karloff, and Craig Tovey. New results on the old k-opt algorithm for the traveling salesman problem. *SIAM Journal on Computing*, 28(6):1998–2029, 1999.

[8] Richard Durrett. *Probability: Theory and Examples*. Cambridge University Press, 2013.

[9] Matthias Englert, Heiko Röglin, and Berthold Vöcking. Worst case and probabilistic analysis of the 2-Opt algorithm for the TSP. *Algorithmica*, 68(1):190–264, 2014.

[10] Michael Etscheid and Heiko Röglin. Smoothed analysis of local search for the maximum-cut problem. In *Proc. of the 25th Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 882–889. SIAM, 2014.

[11] Marvin Künnemann and Bodo Manthey. Towards understanding the smoothed approximation ratio of the 2-opt heuristic. In Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann, editors, *Proc. of the 42nd Int. Coll. on Automata, Languages and Programming (ICALP)*, volume 9134 of *Lecture Notes in Computer Science*, pages 859–871. Springer, 2015.

[12] Bodo Manthey. Smoothed analysis of local search algorithms. In Frank Dehne, Jörg-Rüdiger Sack, and Ulrike Stege, editors, *Proc. of the 14th Algorithms and Data Structures Symposium (WADS)*, volume 9214 of *Lecture Notes in Computer Science*, pages 518–527. Springer, 2015.

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

[14] Bodo Manthey and Rianne Veenstra. Smoothed analysis of the 2-Opt heuristic for the TSP: Polynomial bounds for Gaussian noise. In Leizhen Cai, Siu-Wing Cheng, and Tak-Wah Lam, editors, *Proc. of the 24th Ann. Int. Symp. on Algorithms and Computation (ISAAC)*, volume 8283 of *Lecture Notes in Computer Science*, pages 579–589. Springer, 2013.

[15] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005.

[16] James B. Orlin. A faster strongly polynomial minimum cost flow algorithm. *Operations Research*, 41(2):338–350, 1993.

[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.

[20] Norman Zadeh. A bad network problem for the simplex method and other minimum cost flow algorithms. *Mathematical Programming*, 5:255–263, 1973.