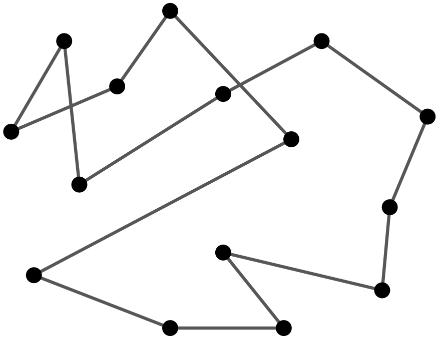
Smoothed Analysis of Local Search

Jesse van Rhijn

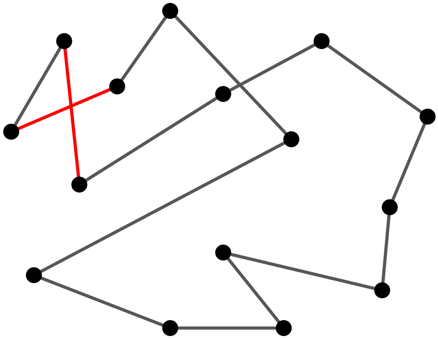
University of Twente

February 8, 2023

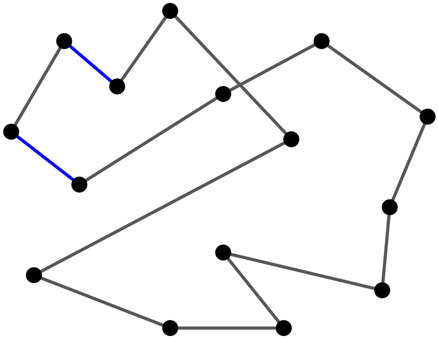
2-Opt and TSP



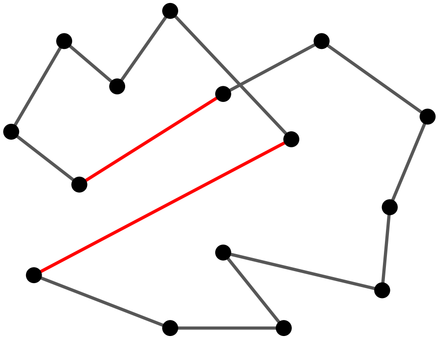
2-Opt and TSP



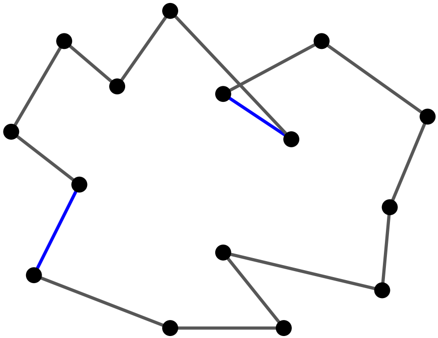
2-Opt and TSP



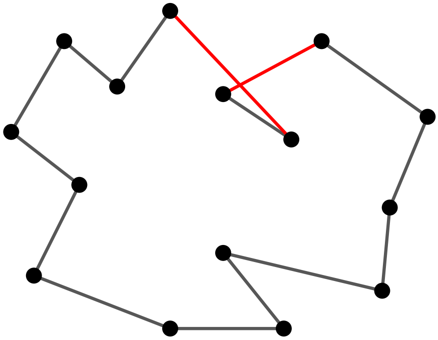
2-Opt and TSP



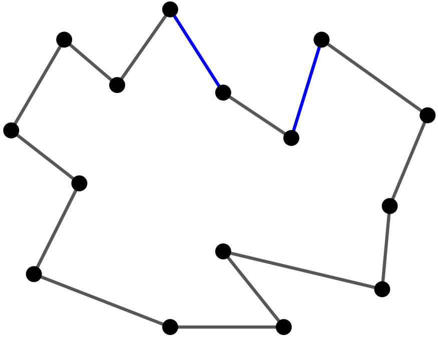
2-Opt and TSP



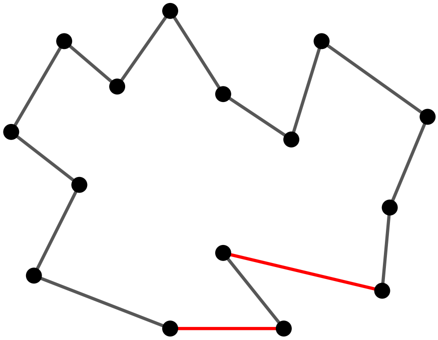
2-Opt and TSP



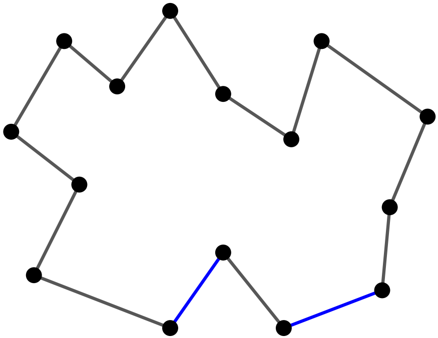
2-Opt and TSP



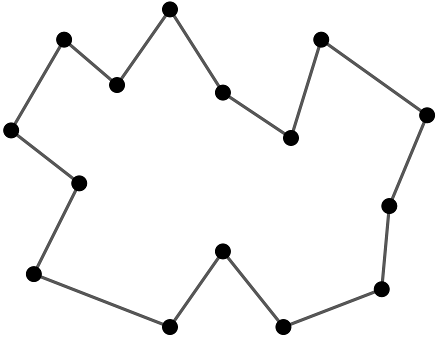
2-Opt and TSP



2-Opt and TSP



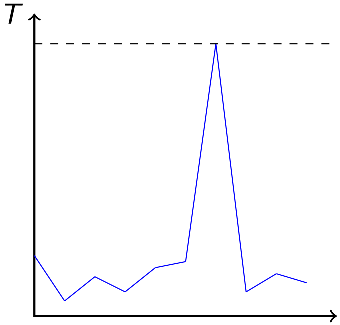
2-Opt and TSP



	#iterations	approximation ratio
theoretical	$2^{\Omega(n)}$ (Englert, Röglin, Vöcking)	$O(\log n)$, $\Omega(\frac{\log n}{\log \log n})$ (Chandra, Karloff, Tovey)
practical	$o(n^2)$ (Johnson, McGeoch)	1.05

Worst Case Analysis

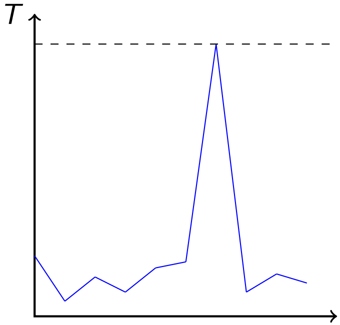
Standard measure of algorithm performance.



Formally: $W(n) = \max_{I \in \mathcal{I}_n} T(I)$.

Worst Case Analysis

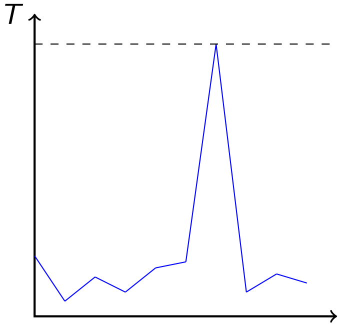
Standard measure of algorithm performance.



Formally: $W(n) = \max_{I \in \mathcal{I}_n} T(I)$.

Worst Case Analysis

Standard measure of algorithm performance.

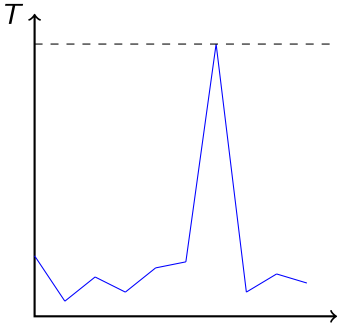


Formally: $W(n) = \max_{I \in \mathcal{I}_n} T(I)$.

Advantage: strong guarantee on performance.

Worst Case Analysis

Standard measure of algorithm performance.

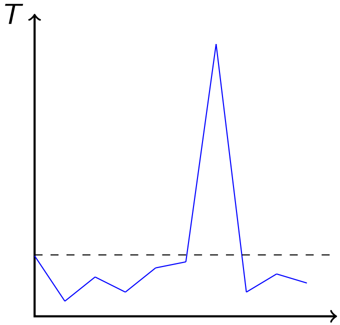


Formally: $W(n) = \max_{I \in \mathcal{I}_n} T(I)$.

Advantage: strong guarantee on performance.
Disadvantage: may be overly pessimistic.

Average Case Analysis

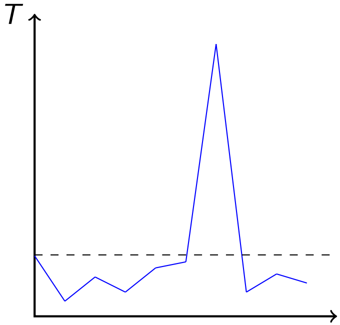
Possible solution to pessimism.



Formally: $A(n) = \mathbb{E}_{I \sim \pi_n}(T(I))$.

Average Case Analysis

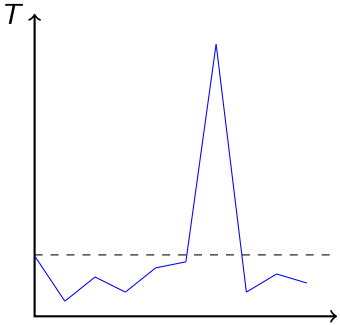
Possible solution to pessimism.



Formally: $A(n) = \mathbb{E}_{I \sim \pi_n}(T(I))$.

Average Case Analysis

Possible solution to pessimism.

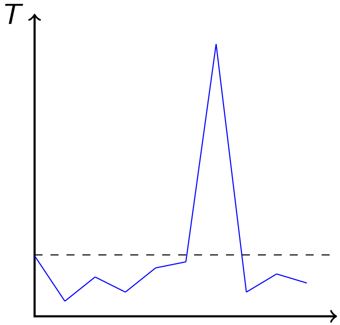


Formally: $A(n) = \mathbb{E}_{I \sim \pi_n}(T(I))$.

Advantage: reduced impact of pathologies.

Average Case Analysis

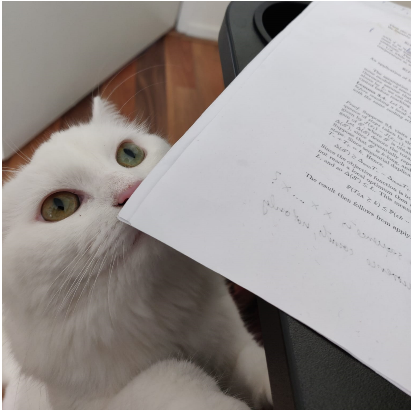
Possible solution to pessimism.

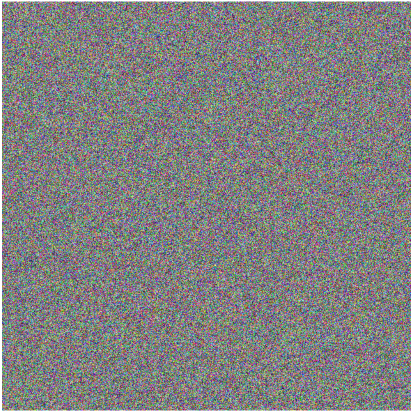


Formally: $A(n) = \mathbb{E}_{I \sim \pi_n}(T(I))$.

Advantage: reduced impact of pathologies.
Disadvantage: may be unrealistic, choice of π_n.

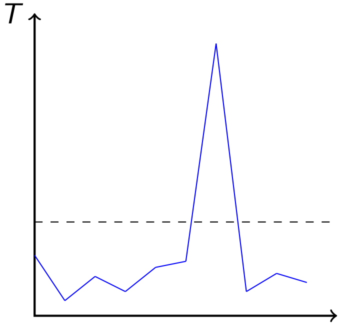
Average Case Shortcomings





Smoothed Analysis

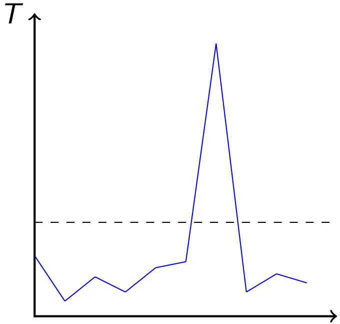
Smoothed analysis: random perturbations of worst case instances.



Formally: $S(n, \sigma) = \max_{I \in \mathcal{I}_n} \mathbb{E}_{g \sim \pi_\sigma}(T(I, g))$.

Smoothed Analysis

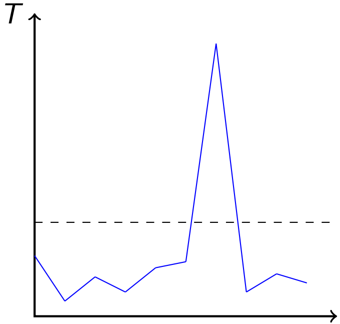
Smoothed analysis: random perturbations of worst case instances.



Formally: $S(n, \sigma) = \max_{I \in \mathcal{I}_n} \mathbb{E}_{g \sim \pi_\sigma}(T(I, g))$.

Smoothed Analysis

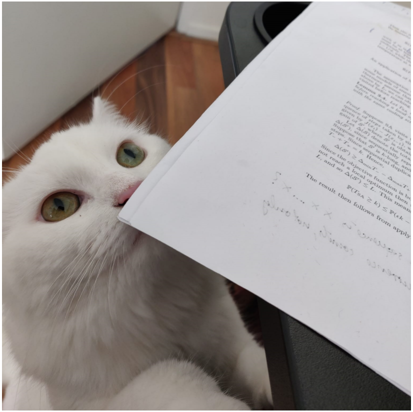
Smoothed analysis: random perturbations of worst case instances.



Formally: $S(n, \sigma) = \max_{I \in \mathcal{I}_n} \mathbb{E}_{g \sim \pi_\sigma}(T(I, g))$.

Combines average case and worst case analysis.

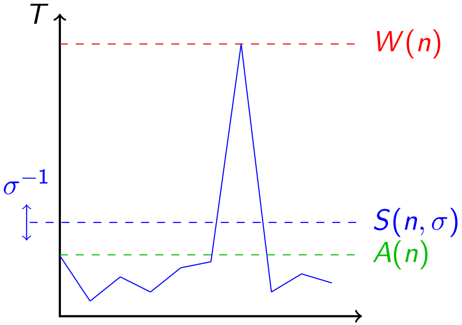
Smoothed Analysis: Original and Perturbed



Moving Between W and A

Smoothed complexity is parametrized by σ.

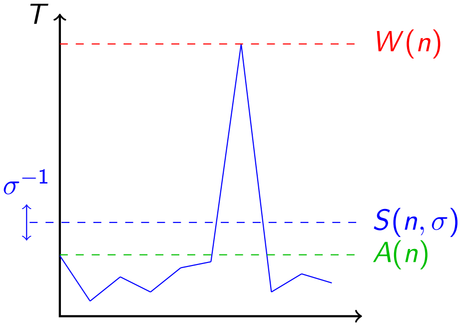
Allows interpolation between worst/average case.



Moving Between W and A

Smoothed complexity is parametrized by σ.

Allows interpolation between worst/average case.



Two-Step and One-Step Models

Most common models of SA:

- Two step model: take an arbitrary input and perturb it (by Gaussians).
- One step model: draw numbers in the input from independent bounded probability densities.

Two step model is parametrized by σ, one step model by $\phi \geq 1$.

Rough correspondence: $\phi \simeq \sigma^{-1}$.

Two-Step and One-Step Models

Most common models of SA:

- Two step model: take an arbitrary input and perturb it (by Gaussians).
- One step model: draw numbers in the input from independent bounded probability densities.

Two step model is parametrized by σ, one step model by $\phi \geq 1$.

Rough correspondence: $\phi \simeq \sigma^{-1}$.

Two-Step and One-Step Models

Most common models of SA:

- Two step model: take an arbitrary input and perturb it (by Gaussians).
- One step model: draw numbers in the input from independent bounded probability densities.

Two step model is parametrized by σ, one step model by $\phi \geq 1$.

Rough correspondence: $\phi \simeq \sigma^{-1}$.

Two-Step and One-Step Models

Most common models of SA:

- Two step model: take an arbitrary input and perturb it (by Gaussians).
- One step model: draw numbers in the input from independent bounded probability densities.

Two step model is parametrized by σ, one step model by $\phi \geq 1$.

Rough correspondence: $\phi \simeq \sigma^{-1}$.

Two-Step and One-Step Models

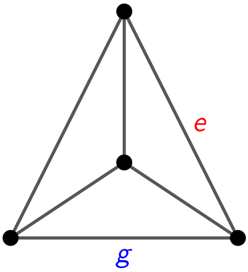
Most common models of SA:

- Two step model: take an arbitrary input and perturb it (by Gaussians).
- One step model: draw numbers in the input from independent bounded probability densities.

Two step model is parametrized by σ, one step model by $\phi \geq 1$.

Rough correspondence: $\phi \simeq \sigma^{-1}$.

Comparing the Models: Weighted Graphs

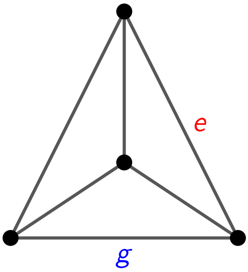


Weights: $w(e) \sim f_e$, $w(g) \sim f_g$, etc.

- $f_e, f_g \leq \phi$.
- $w(e)$, $w(g)$ are independent!

Metric instances?

Comparing the Models: Weighted Graphs

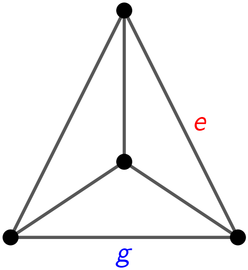


Weights: $w(e) \sim f_e$, $w(g) \sim f_g$, etc.

- $f_e, f_g \leq \phi$.
- $w(e)$, $w(g)$ are independent!

Metric instances?

Comparing the Models: Weighted Graphs

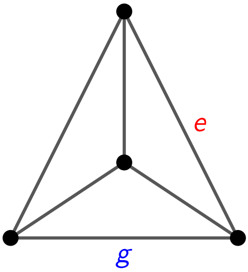


Weights: $w(e) \sim f_e$, $w(g) \sim f_g$, etc.

- $f_e, f_g \leq \phi$.
- $w(e)$, $w(g)$ are independent!

Metric instances?

Comparing the Models: Weighted Graphs

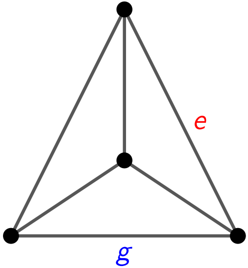


Weights: $w(e) \sim f_e$, $w(g) \sim f_g$, etc.

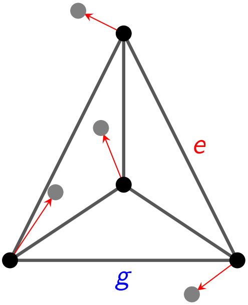
- $f_e, f_g \leq \phi$.
- $w(e)$, $w(g)$ are independent!

Metric instances?

Comparing the Models: Weighted Graphs

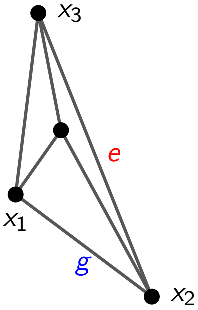


Comparing the Models: Weighted Graphs



Perturbations: usually Gaussian.

Comparing the Models: Weighted Graphs



Perturbations: usually Gaussian.

Weights: $w(e) = d(x_2, x_3)$, $w(g) = d(x_1, x_2)$.

Questions

Intermezzo: questions?

Local Search

Local search: simple combinatorial optimization paradigm.

For a solution x, define a neighborhood $N(x)$ of better solutions.

Choose some $y \in N(x)$ as the new solution.

Continue until neighborhood is empty $\rightarrow$ local optimum.

Ingredients:

- Neighborhood: what solutions are neighbors?
- Pivot rule: how to select next solution?
- Initialization: how to compute starting solution?
- Cost: how to compare solutions?

Local Search

Local search: simple combinatorial optimization paradigm.

For a solution x, define a neighborhood $N(x)$ of better solutions.

Choose some $y \in N(x)$ as the new solution.

Continue until neighborhood is empty $\rightarrow$ local optimum.

Ingredients:

- Neighborhood: what solutions are neighbors?
- Pivot rule: how to select next solution?
- Initialization: how to compute starting solution?
- Cost: how to compare solutions?

Local Search

Local search: simple combinatorial optimization paradigm.

For a solution x, define a neighborhood $N(x)$ of better solutions.

Choose some $y \in N(x)$ as the new solution.

Continue until neighborhood is empty $\rightarrow$ local optimum.

Ingredients:

- Neighborhood: what solutions are neighbors?
- Pivot rule: how to select next solution?
- Initialization: how to compute starting solution?
- Cost: how to compare solutions?

Local Search

Local search: simple combinatorial optimization paradigm.

For a solution x, define a neighborhood $N(x)$ of better solutions.

Choose some $y \in N(x)$ as the new solution.

Continue until neighborhood is empty $\rightarrow$ local optimum.

Ingredients:

- Neighborhood: what solutions are neighbors?
- Pivot rule: how to select next solution?
- Initialization: how to compute starting solution?
- Cost: how to compare solutions?

Local Search

Local search: simple combinatorial optimization paradigm.

For a solution x, define a neighborhood $N(x)$ of better solutions.

Choose some $y \in N(x)$ as the new solution.

Continue until neighborhood is empty $\rightarrow$ local optimum.

Ingredients:

- Neighborhood: what solutions are neighbors?
- Pivot rule: how to select next solution?
- Initialization: how to compute starting solution?
- Cost: how to compare solutions?

Local Search

Local search: simple combinatorial optimization paradigm.

For a solution x, define a neighborhood $N(x)$ of better solutions.

Choose some $y \in N(x)$ as the new solution.

Continue until neighborhood is empty $\rightarrow$ local optimum.

Ingredients:

- Neighborhood: what solutions are neighbors?
- Pivot rule: how to select next solution?
- Initialization: how to compute starting solution?
- Cost: how to compare solutions?

Local Search

Local search: simple combinatorial optimization paradigm.

For a solution x, define a neighborhood $N(x)$ of better solutions.

Choose some $y \in N(x)$ as the new solution.

Continue until neighborhood is empty $\rightarrow$ local optimum.

Ingredients:

- Neighborhood: what solutions are neighbors?
- Pivot rule: how to select next solution?
- Initialization: how to compute starting solution?
- Cost: how to compare solutions?

Local Search

Local search: simple combinatorial optimization paradigm.

For a solution x, define a neighborhood $N(x)$ of better solutions.

Choose some $y \in N(x)$ as the new solution.

Continue until neighborhood is empty $\to$ local optimum.

Ingredients:

- Neighborhood: what solutions are neighbors?
- Pivot rule: how to select next solution?
- Initialization: how to compute starting solution?
- Cost: how to compare solutions?

Local Search

Local search: simple combinatorial optimization paradigm.

For a solution x, define a neighborhood $N(x)$ of better solutions.

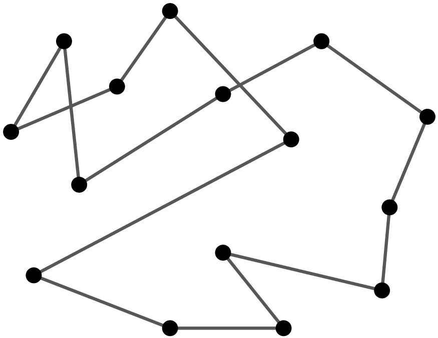
Choose some $y \in N(x)$ as the new solution.

Continue until neighborhood is empty $\rightarrow$ local optimum.

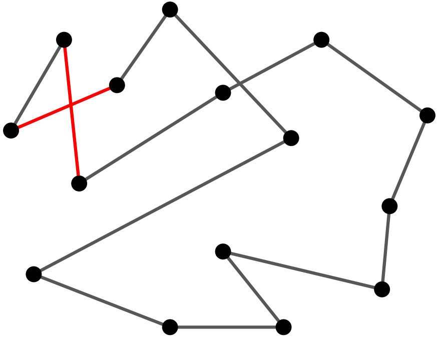
Ingredients:

- Neighborhood: what solutions are neighbors?
- Pivot rule: how to select next solution?
- Initialization: how to compute starting solution?
- Cost: how to compare solutions?

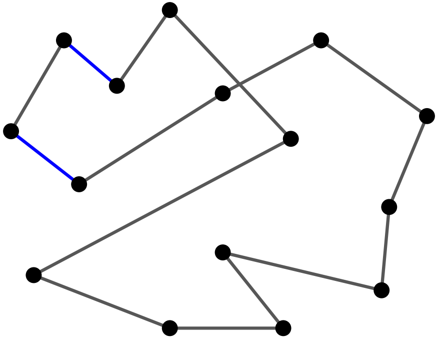
Local Search: 2-Opt



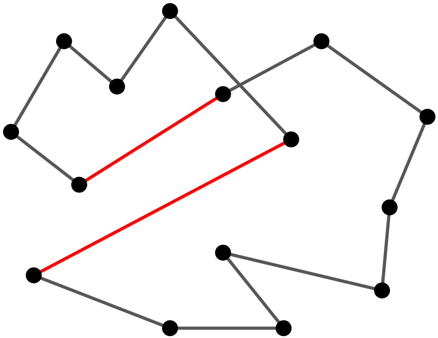
Local Search: 2-Opt



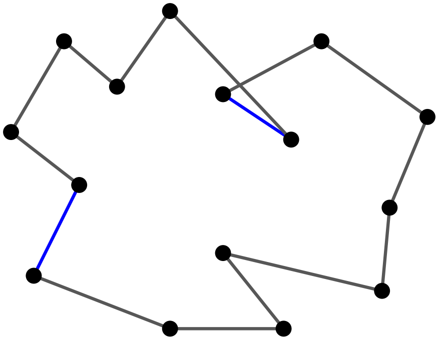
Local Search: 2-Opt



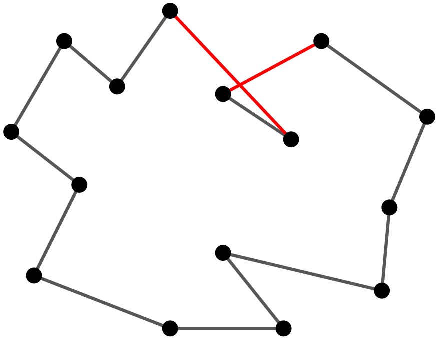
Local Search: 2-Opt



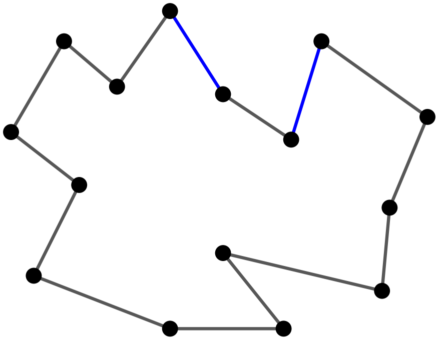
Local Search: 2-Opt



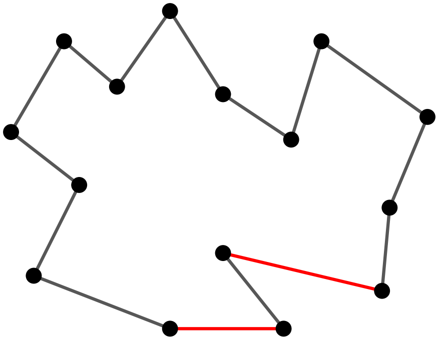
Local Search: 2-Opt



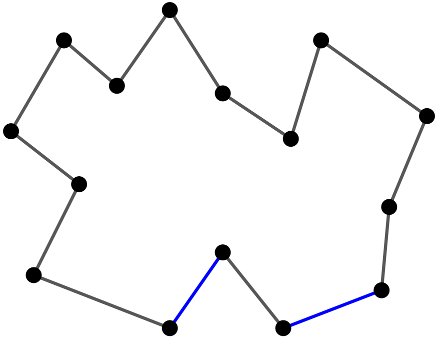
Local Search: 2-Opt



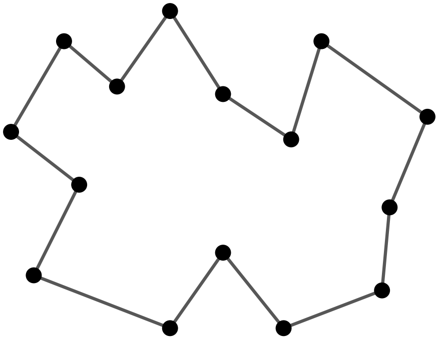
Local Search: 2-Opt



Local Search: 2-Opt



Local Search: 2-Opt



Local Search: 2-Opt

Theorem (Englert, Röglin & Vöcking)

There exist instances of 2-opt where the heuristic may take $2^{\Omega(n)}$ *iterations to converge.*

Local Search: 2-Opt

Theorem (Englert, Röglin & Vöcking)

There exist instances of 2-opt where the heuristic may take $2^{\Omega(n)}$ iterations to converge.

Theorem (Englert, Röglin & Vöcking)

The smoothed complexity of 2-opt on general graphs is $m^{1+o(1)} n\phi$.

Local Search: 2-Opt

Theorem (Englert, Röglin & Vöcking)

There exist instances of 2-opt where the heuristic may take $2^{\Omega(n)}$ iterations to converge.

Theorem (Englert, Röglin & Vöcking)

The smoothed complexity of 2-opt on general graphs is $m^{1+o(1)} n\phi$.

Theorem (Manthey & van Rhijn)

The smoothed complexity of 2-opt on Euclidean graphs is $O(n^{4+\frac{1}{3}}/\sigma^2)$.

Local Search: FLIP for MAX-CUT

Definition (MAX-CUT)

Input: weighted graph $G = (V, E, w)$.
Goal: find a set $S \subset V$ such that

$$\sum_{\substack{e=\{u,v\}\in E \\ u\in V, v\notin V}} w(e)$$

is maximized.

Theorem

MAX-CUT is NP-hard.

Local Search: FLIP for MAX-CUT

Definition (MAX-CUT)

Input: weighted graph $G = (V, E, w)$.
Goal: find a set $S \subset V$ such that

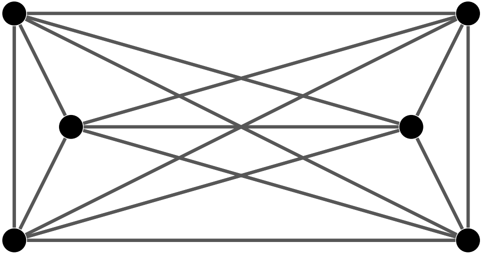
$$\sum_{\substack{e=\{u,v\}\in E \\ u\in V, v\notin V}} w(e)$$

is maximized.

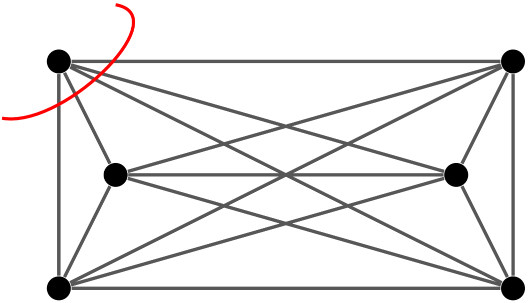
Theorem

MAX-CUT is NP-hard.

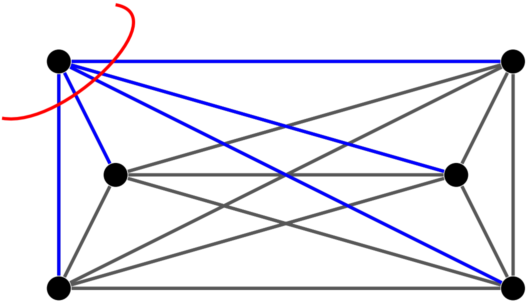
Local Search: Flip for MAX-CUT



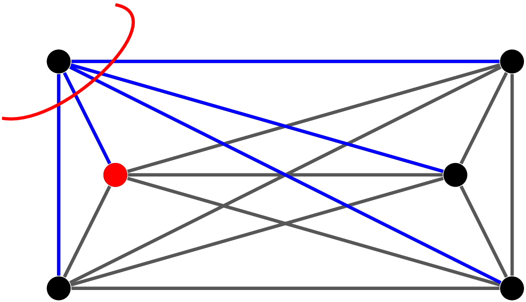
Local Search: Flip for MAX-CUT



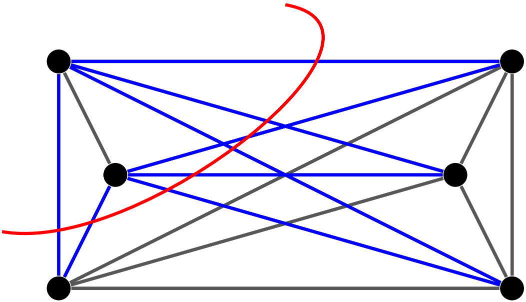
Local Search: Flip for MAX-CUT



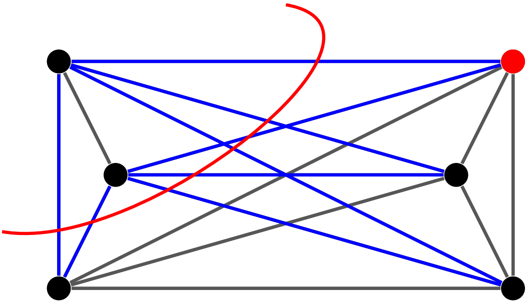
Local Search: Flip for MAX-CUT



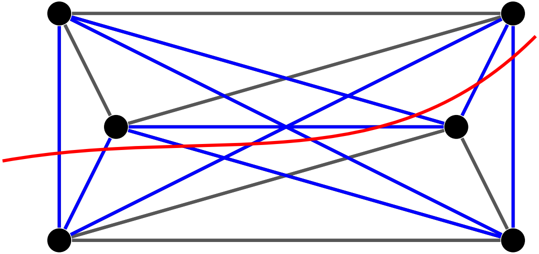
Local Search: Flip for MAX-CUT



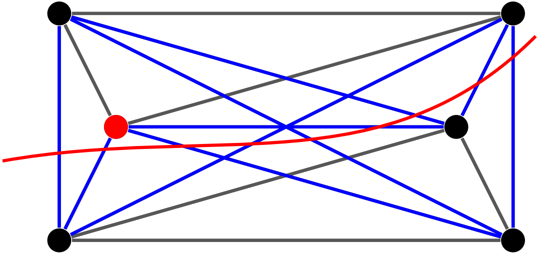
Local Search: Flip for MAX-CUT



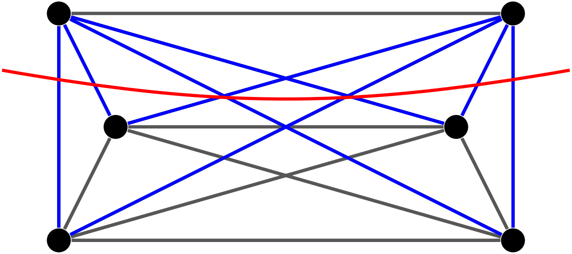
Local Search: Flip for MAX-CUT



Local Search: Flip for MAX-CUT



Local Search: Flip for MAX-CUT



Local Search: Flip for MAX-CUT

Theorem (Schäffer & Yannakakis)

There exist instances of MAX-CUT on which every run of Flip has exponential length.

Local Search: Flip for MAX-CUT

Theorem (Schäffer & Yannakakis)

There exist instances of MAX-CUT on which every run of Flip has exponential length.

Theorem (Bibak, Carlson & Chandrasekaran)

The smoothed complexity of the Flip heuristic is $O(n^{7.84}\phi)$ *for complete graphs.*

Local Search: Flip for MAX-CUT

Theorem (Schäffer & Yannakakis)

There exist instances of MAX-CUT on which every run of Flip has exponential length.

Theorem (Bibak, Carlson & Chandrasekaran)

The smoothed complexity of the Flip heuristic is $O(n^{7.84}\phi)$ for complete graphs.

Ongoing work: general (non-complete) graphs, MAX-k-CUT.

Local Search: Lloyd for k-means

Definition (k-means clustering)

Input: a set of points $X \subseteq \mathbb{R}^d$, an integer $k > 0$.
Goal: find a partition $\{C_i\}_{i=1}^k$ of X such that

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

is minimized, where $c_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$ (center of mass).

Theorem

k-means clustering is NP-hard.

Local Search: Lloyd for k-means

Definition (k-means clustering)

Input: a set of points $X \subseteq \mathbb{R}^d$, an integer $k > 0$.
Goal: find a partition $\{C_i\}_{i=1}^k$ of X such that

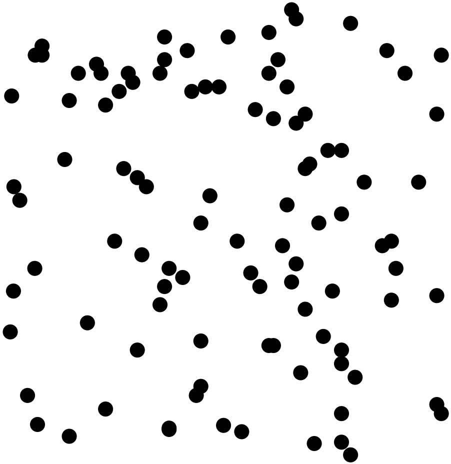
$$\sum_{i=1}^{k} \sum_{x \in C_i} \|x - c_i\|^2$$

is minimized, where $c_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$ (center of mass).

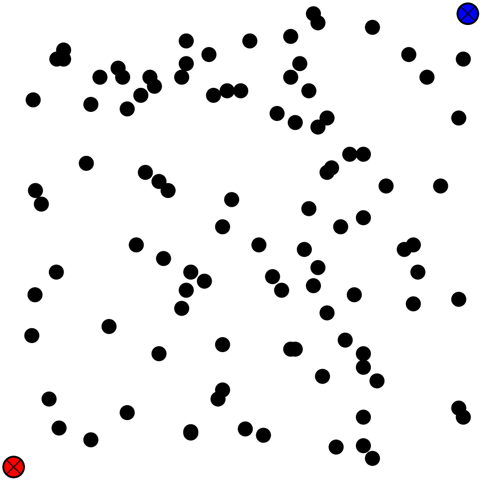
Theorem

k-means clustering is NP-hard.

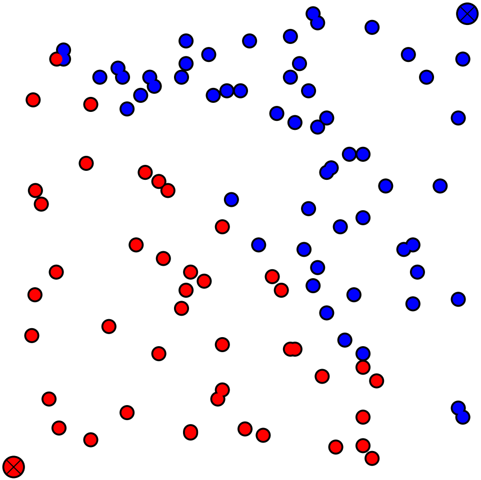
Local Search: k-means Clustering



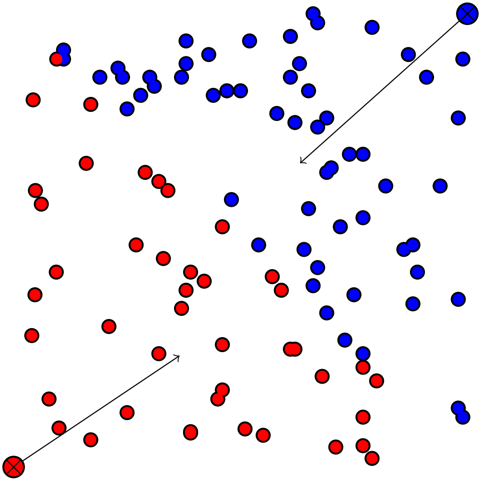
Local Search: *k*-means Clustering



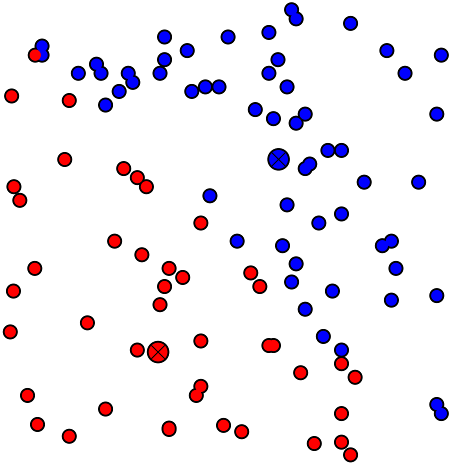
Local Search: k-means Clustering



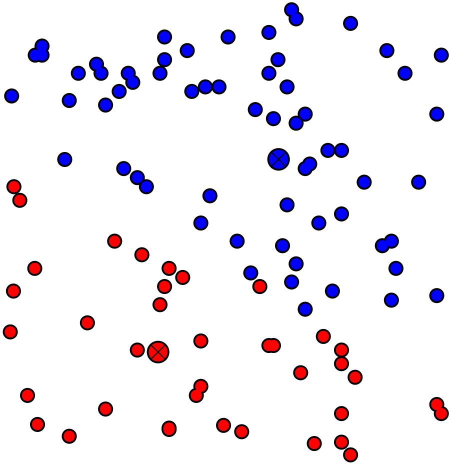
Local Search: k-means Clustering



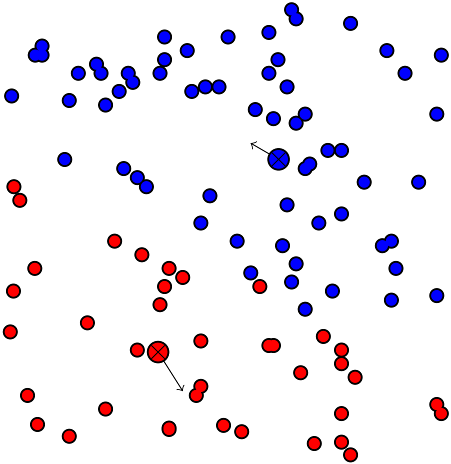
Local Search: k-means Clustering



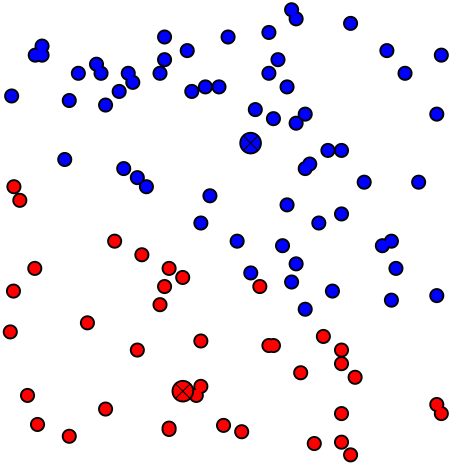
Local Search: k-means Clustering



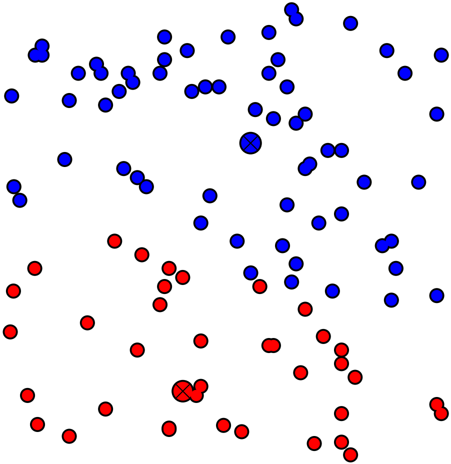
Local Search: k-means Clustering



Local Search: k-means Clustering



Local Search: k-means Clustering



Local Search: k-means Clustering

Theorem (Vattani)

There exist instances of k-means that require $2^{\Omega(n)}$ iterations, even in the plane.

Local Search: k-means Clustering

Theorem (Vattani)

There exist instances of k-means that require $2^{\Omega(n)}$ iterations, even in the plane.

Theorem (Arthur, Manthey & Röglin)

The smoothed complexity of k-means is $\tilde{O}(n^{34} k^{34} / \sigma^6)$.

Local Search: k-means Clustering

Theorem (Vattani)

There exist instances of k-means that require $2^{\Omega(n)}$ iterations, even in the plane.

Theorem (Arthur, Manthey & Röglin)

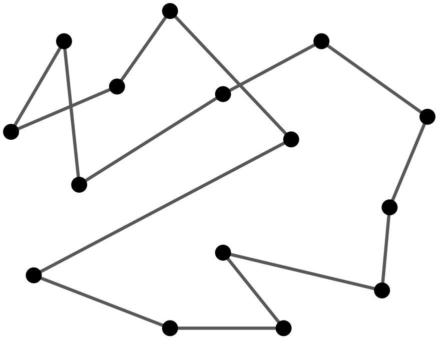
The smoothed complexity of k-means is $\tilde{O}(n^{34}k^{34}/\sigma^6)$.

Open: different norms.

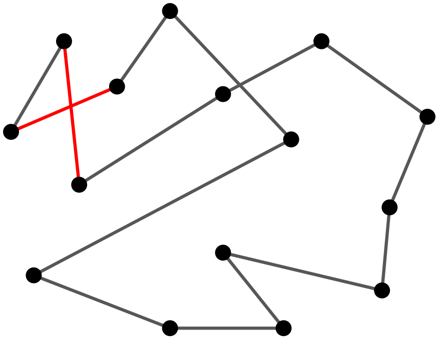
Questions

Intermezzo: questions?

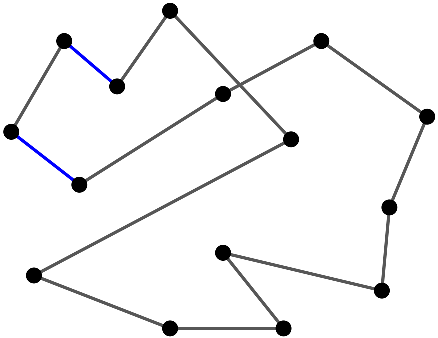
Reminder: 2-Opt



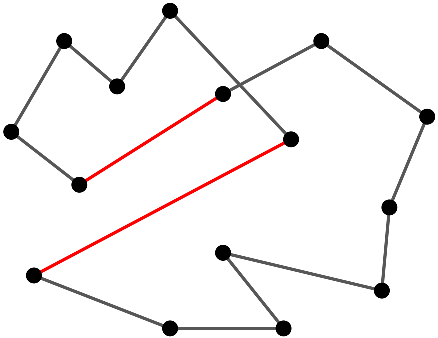
Reminder: 2-Opt



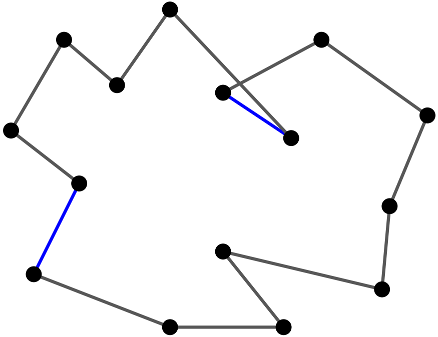
Reminder: 2-Opt



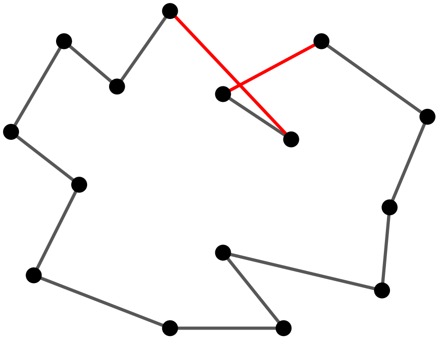
Reminder: 2-Opt



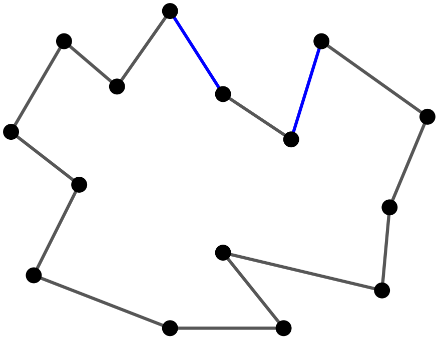
Reminder: 2-Opt



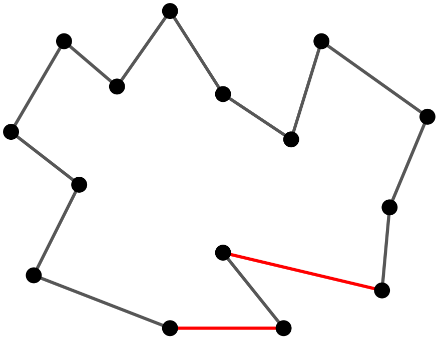
Reminder: 2-Opt



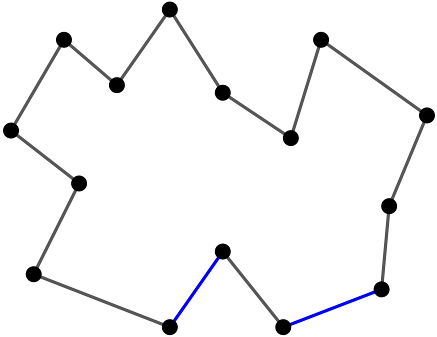
Reminder: 2-Opt



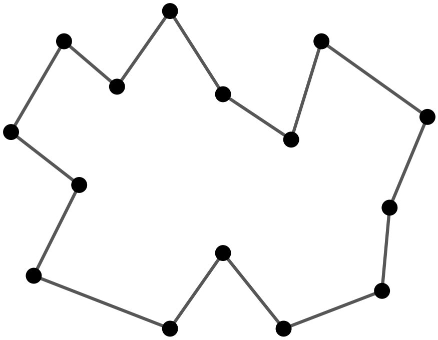
Reminder: 2-Opt



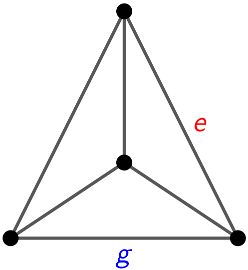
Reminder: 2-Opt



Reminder: 2-Opt



Analyze in the One-Step Model

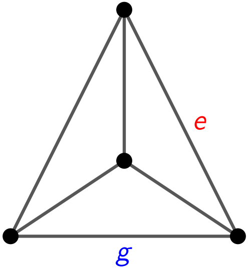


$|V| = n$ and $|E| = m$.

Weights: $w(e) \sim f_e$, $w(g) \sim f_g$, etc., with weights in $[0, 1]$.

Potential argument: if all steps improve tour by at least $\Delta_{\min} > 0$, how long before we terminate?

Analyze in the One-Step Model

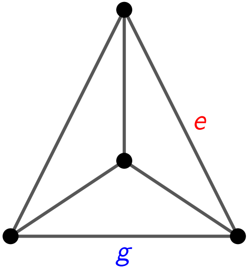


$|V| = n$ and $|E| = m$.

Weights: $w(e) \sim f_e$, $w(g) \sim f_g$, etc., with weights in $[0, 1]$.

Potential argument: if all steps improve tour by at least $\Delta_{\min} > 0$, how long before we terminate?

Analyze in the One-Step Model



$|V| = n$ and $|E| = m$.

Weights: $w(e) \sim f_e$, $w(g) \sim f_g$, etc., with weights in $[0, 1]$.

Potential argument: if all steps improve tour by at least $\Delta_{\min} > 0$, how long before we terminate?

Some Necessary Tools

Lemma (Union Bound/Boole's Inequality)

Let $\{E_i\}_{i=1}^k$ be a collection of events. Then

$$\mathbb{P}\left(\bigcup_{i=1}^k E_i\right) \leq \sum_{i=1}^k \mathbb{P}(E_i).$$

Lemma (Interval Lemma)

Let X be a random variable whose density is bounded from above by ϕ. Let I be an interval of size at most ϵ. Then

$$\mathbb{P}(X \in I) \leq \phi \cdot \epsilon.$$

Proof: $\mathbb{P}(X \in I) = \int_I f_X(x)\mathrm{d}x \leq \phi \cdot \int_I \mathrm{d}x \leq \phi \cdot \epsilon.$ □

Some Necessary Tools

Lemma (Union Bound/Boole's Inequality)

Let $\{E_i\}_{i=1}^k$ *be a collection of events. Then*

$$\mathbb{P}\left(\bigcup_{i=1}^k E_i\right) \leq \sum_{i=1}^k \mathbb{P}(E_i).$$

Lemma (Interval Lemma)

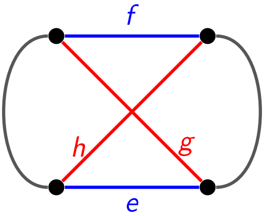
Let X *be a random variable whose density is bounded from above by* ϕ*. Let* I *be an interval of size at most* ϵ*. Then*

$$\mathbb{P}(X \in I) \leq \phi \cdot \epsilon.$$

Proof: $\mathbb{P}(X \in I) = \int_I f_X(x)\mathrm{d}x \leq \phi \cdot \int_I \mathrm{d}x \leq \phi \cdot \epsilon.$ $\square$

Strategy of Analysis

Consider a single iteration:



Improvement:

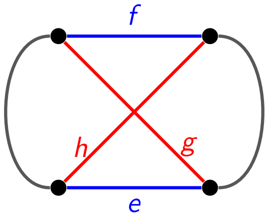
$$\Delta = w_h + w_g - w_e - w_f .$$

Assume w_g, w_e, w_f are fixed, so that $w_g - w_e - w_f = t$. Then

$$\mathbb{P}(\Delta \leq \epsilon) = \mathbb{P}(\Delta \in (0, \epsilon]) = \mathbb{P}(w_h \in (-t, -t + \epsilon]).$$

Strategy of Analysis

Consider a single iteration:



Improvement:

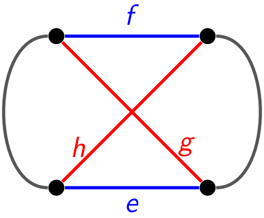
$$\Delta = w_h + w_g - w_e - w_f .$$

Assume w_g, w_e, w_f are fixed, so that $w_g - w_e - w_f = t$. Then

$$\mathbb{P}(\Delta \leq \epsilon) = \mathbb{P}(\Delta \in (0, \epsilon]) = \mathbb{P}(w_h \in (-t, -t + \epsilon]).$$

Strategy of Analysis

Consider a single iteration:



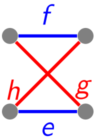
Improvement:

$$\Delta = w_h + w_g - w_e - w_f .$$

<u>Assume</u> w_g, w_e, w_f are fixed, so that $w_g - w_e - w_f = t$. Then

$$\mathbb{P}(\Delta \leq \epsilon) = \mathbb{P}(\Delta \in (0, \epsilon]) = \mathbb{P}(w_h \in (-t, -t + \epsilon]).$$

Bounding the Improvement



Density of w_h:

$$f_h(x) \leq \phi \quad \text{for} \quad x \in [0, 1].$$

We have (Interval Lemma)

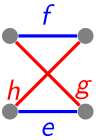
$$\mathbb{P}(\Delta \leq \epsilon) = \mathbb{P}(w_h \in (-t, -t + \epsilon]) \leq \phi \cdot \epsilon.$$

There are m edges, so m^2 choices for g and h. Union bound:

Lemma

If $\Delta_{\min}$ is the improvement of the worst 2-opt step, then $\mathbb{P}(\Delta_{\min} \leq \epsilon) = O(m^2 \cdot \phi \cdot \epsilon)$.

Bounding the Improvement



Density of w_h:

$$f_h(x) \leq \phi \quad \text{for} \quad x \in [0,1].$$

We have (Interval Lemma)

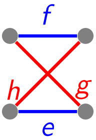
$$\mathbb{P}(\Delta \leq \epsilon) = \mathbb{P}(w_h \in (-t, -t+\epsilon]) \leq \phi \cdot \epsilon.$$

There are m edges, so m^2 choices for g and h. Union bound:

Lemma

If $\Delta_{\min}$ is the improvement of the worst 2-opt step, then $\mathbb{P}(\Delta_{\min} \leq \epsilon) = O(m^2 \cdot \phi \cdot \epsilon)$.

Bounding the Improvement



Density of w_h:

$$f_h(x) \leq \phi \quad \text{for} \quad x \in [0,1].$$

We have (Interval Lemma)

$$\mathbb{P}(\Delta \leq \epsilon) = \mathbb{P}(w_h \in (-t, -t+\epsilon]) \leq \phi \cdot \epsilon.$$

There are m edges, so m^2 choices for g and h. Union bound:

Lemma

If $\Delta_{\min}$ is the improvement of the worst 2-opt step, then $\mathbb{P}(\Delta_{\min} \leq \epsilon) = O(m^2 \cdot \phi \cdot \epsilon)$.

Completing the Analysis

T: number of iterations to terminate. Then smoothed complexity $= \mathbb{E}(T)$.

Tail sum:

$$\mathbb{E}(T) \leq \sum_{t=1}^{n!} \mathbb{P}(T \geq t).$$

Every tour has length $\leq n$, so can take at most $n/\Delta_{\min}$ steps:

$$\mathbb{P}(T \geq t) \leq \mathbb{P}(t \leq n/\Delta_{\min}) = \mathbb{P}(\Delta_{\min} \leq n/t) = O(\phi \cdot m \cdot n/t).$$

Thus,

$$\begin{aligned} \mathbb{E}(T) &\leq \sum_{t=1}^{n!} O(\phi \cdot m \cdot n/t) = O\left(\phi \cdot m \cdot n \cdot \int_1^{n!} \frac{1}{t} \mathrm{d}t\right) \\ &= O(\phi \cdot m \cdot n^2 \cdot \log n). \end{aligned}$$

Completing the Analysis

T: number of iterations to terminate. Then smoothed complexity $= \mathbb{E}(T)$.

Tail sum:

$$\mathbb{E}(T) \leq \sum_{t=1}^{n!} \mathbb{P}(T \geq t).$$

Every tour has length $\leq n$, so can take at most $n/\Delta_{\min}$ steps:

$$\mathbb{P}(T \geq t) \leq \mathbb{P}(t \leq n/\Delta_{\min}) = \mathbb{P}(\Delta_{\min} \leq n/t) = O(\phi \cdot m \cdot n/t).$$

Thus,

$$\begin{aligned}\mathbb{E}(T) &\leq \sum_{t=1}^{n!} O(\phi \cdot m \cdot n/t) = O\left(\phi \cdot m \cdot n \cdot \int_1^{n!} \frac{1}{t} \mathrm{d}t\right) \\ &= O(\phi \cdot m \cdot n^2 \cdot \log n).\end{aligned}$$

Completing the Analysis

T: number of iterations to terminate. Then smoothed complexity $= \mathbb{E}(T)$.

Tail sum:

$$\mathbb{E}(T) \leq \sum_{t=1}^{n!} \mathbb{P}(T \geq t).$$

Every tour has length $\leq n$, so can take at most $n/\Delta_{\min}$ steps:

$$\mathbb{P}(T \geq t) \leq \mathbb{P}(t \leq n/\Delta_{\min}) = \mathbb{P}(\Delta_{\min} \leq n/t) = O(\phi \cdot m \cdot n/t).$$

Thus,

$$\begin{aligned} \mathbb{E}(T) &\leq \sum_{t=1}^{n!} O(\phi \cdot m \cdot n/t) = O\left(\phi \cdot m \cdot n \cdot \int_1^{n!} \frac{1}{t} \mathrm{d}t\right) \\ &= O(\phi \cdot m \cdot n^2 \cdot \log n). \end{aligned}$$

Completing the Analysis

T: number of iterations to terminate. Then smoothed complexity $= \mathbb{E}(T)$.

Tail sum:

$$\mathbb{E}(T) \leq \sum_{t=1}^{n!} \mathbb{P}(T \geq t).$$

Every tour has length $\leq n$, so can take at most $n/\Delta_{\min}$ steps:

$$\mathbb{P}(T \geq t) \leq \mathbb{P}(t \leq n/\Delta_{\min}) = \mathbb{P}(\Delta_{\min} \leq n/t) = O(\phi \cdot m \cdot n/t).$$

Thus,

$$\begin{aligned} \mathbb{E}(T) &\leq \sum_{t=1}^{n!} O(\phi \cdot m \cdot n/t) = O\left(\phi \cdot m \cdot n \cdot \int_1^{n!} \frac{1}{t} \mathrm{d}t\right) \\ &= O(\phi \cdot m \cdot n^2 \cdot \log n). \end{aligned}$$

Result

Theorem

The smoothed complexity of 2-opt on general graphs is $O(\phi \cdot m \cdot n^2 \log n)$.

Result

Theorem

The smoothed complexity of 2-opt on general graphs is $O(\phi \cdot m \cdot n^2 \log n)$.

Considering sequences of iterations:

Theorem (Englert, Röglin & Vöcking)

The smoothed complexity of 2-opt on general graphs is $m^{1+o(1)} n\phi$.

Challenges

- Bounds are often loose: how close to reality can you get?
- Union bound step is often costly: techniques to reduce it?
- Smoothed approximation performance.
- More complicated heuristics: Lin-Kernighan for TSP?

Challenges

- Bounds are often loose: how close to reality can you get?
- Union bound step is often costly: techniques to reduce it?
- Smoothed approximation performance.
- More complicated heuristics: Lin-Kernighan for TSP?

Challenges

- Bounds are often loose: how close to reality can you get?
- Union bound step is often costly: techniques to reduce it?
- Smoothed approximation performance.
- More complicated heuristics: Lin-Kernighan for TSP?

Challenges

- Bounds are often loose: how close to reality can you get?
- Union bound step is often costly: techniques to reduce it?
- Smoothed approximation performance.
- More complicated heuristics: Lin-Kernighan for TSP?

Concluding Remarks

- Powerful method, but can be technically involved.
- May give insight into heuristic performance and design.
- Very active and relatively young field.

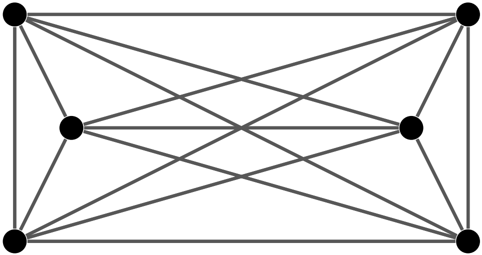
Concluding Remarks

- Powerful method, but can be technically involved.
- May give insight into heuristic performance and design.
- Very active and relatively young field.

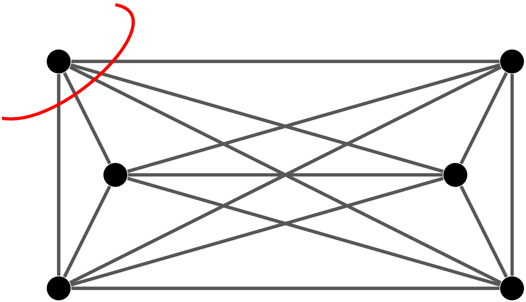
Concluding Remarks

- Powerful method, but can be technically involved.
- May give insight into heuristic performance and design.
- Very active and relatively young field.

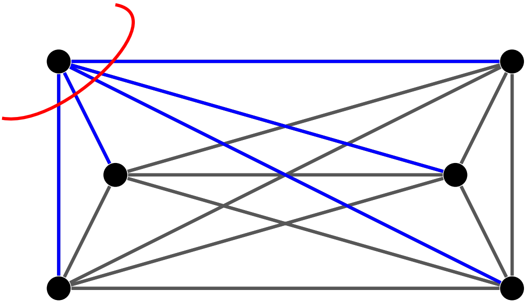
Bonus: Max-Cut/Flip



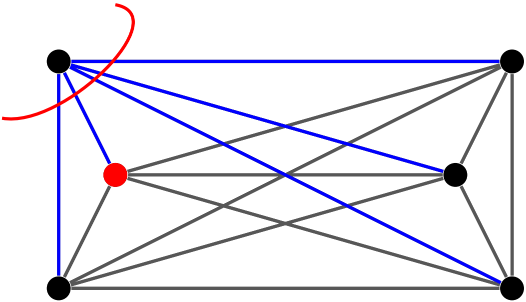
Bonus: Max-Cut/Flip



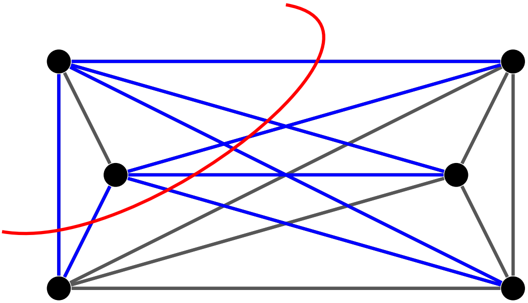
Bonus: Max-Cut/Flip



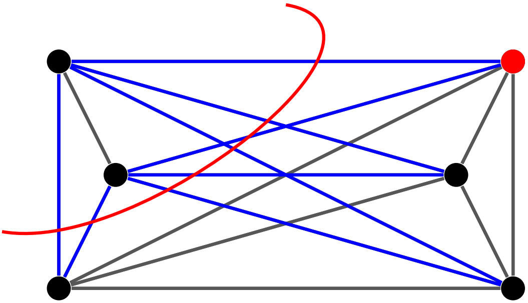
Bonus: Max-Cut/Flip



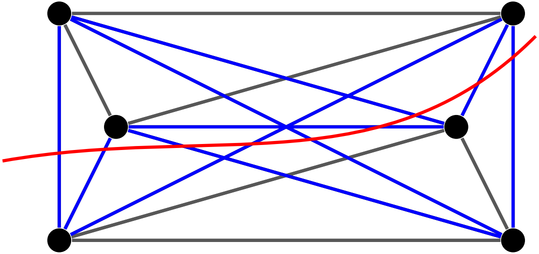
Bonus: Max-Cut/Flip



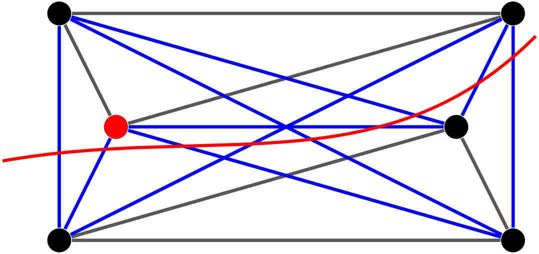
Bonus: Max-Cut/Flip



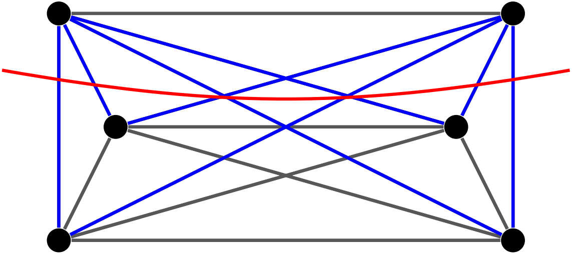
Bonus: Max-Cut/Flip



Bonus: Max-Cut/Flip



Bonus: Max-Cut/Flip



Naive Analysis

Consider a vertex v that flips. Gain:

$$\Delta = \sum_{e=\{u,v\}\in E} \lambda_e w_e, \quad \lambda \in \{1, -1\}$$

Assume all w_e but one $w_{e'}$ are fixed:

$$\Delta = \lambda_{e'} w_{e'} + \sum_{e \neq e'} \lambda_e w_e = w_{e'} + t.$$

Like in 2-opt:

$$\mathbb{P}(\Delta \leq \epsilon) = \mathbb{P}(\Delta \in (0, \epsilon]) = \mathbb{P}(w_{e'} \in (-t, -t+\epsilon]) \leq \phi \cdot \epsilon.$$

Union bound incurs factor $O(n^n) \rightarrow$ useless.

Naive Analysis

Consider a vertex v that flips. Gain:

$$\Delta = \sum_{e=\{u,v\}\in E} \lambda_e w_e, \quad \lambda \in \{1,-1\}$$

Assume all w_e but one $w_{e'}$ are fixed:

$$\Delta = \lambda_{e'} w_{e'} + \sum_{e\neq e'} \lambda_e w_e = w_{e'} + t.$$

Like in 2-opt:

$$\mathbb{P}(\Delta \leq \epsilon) = \mathbb{P}(\Delta \in (0,\epsilon]) = \mathbb{P}(w_{e'} \in (-t,-t+\epsilon]) \leq \phi \cdot \epsilon.$$

Union bound incurs factor $O(n^n) \rightarrow$ useless.

Naive Analysis

Consider a vertex v that flips. Gain:

$$\Delta = \sum_{e=\{u,v\}\in E} \lambda_e w_e, \quad \lambda \in \{1,-1\}$$

Assume all w_e but one $w_{e'}$ are fixed:

$$\Delta = \lambda_{e'} w_{e'} + \sum_{e \neq e'} \lambda_e w_e = w_{e'} + t.$$

Like in 2-opt:

$$\mathbb{P}(\Delta \leq \epsilon) = \mathbb{P}(\Delta \in (0,\epsilon]) = \mathbb{P}(w_{e'} \in (-t, -t+\epsilon]) \leq \phi \cdot \epsilon.$$

Union bound incurs factor $O(n^n) \rightarrow$ useless.

Naive Analysis

Consider a vertex v that flips. Gain:

$$\Delta = \sum_{e=\{u,v\}\in E} \lambda_e w_e, \quad \lambda \in \{1,-1\}$$

Assume all w_e but one $w_{e'}$ are fixed:

$$\Delta = \lambda_{e'} w_{e'} + \sum_{e \neq e'} \lambda_e w_e = w_{e'} + t.$$

Like in 2-opt:

$$\mathbb{P}(\Delta \leq \epsilon) = \mathbb{P}(\Delta \in (0,\epsilon]) = \mathbb{P}(w_{e'} \in (-t, -t+\epsilon]) \leq \phi \cdot \epsilon.$$

Union bound incurs factor $O(n^n) \rightarrow$ useless.

Smarter Analysis

Consider a sequence of ℓ flips:

$$S = (v_1 v_2 \dots v_\ell).$$

Suppose some v flips twice in S.

Gain of two flips of v:

$$\Delta_v = \underbrace{\Delta_1}_{\text{first flip of } v} + \underbrace{\Delta_2}_{\text{second flip of } v} .$$

Edge weight $\{u, v\}$ appears in $\Delta \iff u$ flips *odd* # of times between flips of v.

Smarter Analysis

Consider a sequence of ℓ flips:

$$S = (v_1 v_2 \ldots v_\ell).$$

Suppose some v flips twice in S.

Gain of two flips of v:

$$\Delta_v = \underbrace{\Delta_1}_{\text{first flip of } v} + \underbrace{\Delta_2}_{\text{second flip of } v} .$$

Edge weight $\{u, v\}$ appears in $\Delta \iff u$ flips *odd* $\#$ of times between flips of v.

Smarter Analysis

Consider a sequence of ℓ flips:

$$S = (v_1 v_2 \dots v_\ell).$$

Suppose some v flips twice in S.

Gain of two flips of v:

$$\Delta_v = \underbrace{\Delta_1}_{\text{first flip of } v} + \underbrace{\Delta_2}_{\text{second flip of } v} .$$

Edge weight $\{u, v\}$ appears in $\Delta \iff u$ flips *odd* $\#$ of times between flips of v.

Smarter Analysis

We now have

$$\Delta_v = \sum_{\substack{e=\{u,v\} \\ u \text{ flips off } \# \text{ of times}}} \lambda_e w_e.$$

In particular: only *active* vertices appear in Δ_v.

Definition

A k-repeating subsequence of length ℓ is a sequence of flips in which at least $\lceil \ell/k \rceil$ vertices flip at least twice.

Lemma (Etscheid & Röglin, 2017)

Any sequence of at least $5n$ flips contains a $\lceil 5 \log_2 n \rceil$-repeating subsequence.

Smarter Analysis

We now have

$$\Delta_v = \sum_{\substack{e=\{u,v\} \\ u \text{ flips off } \# \text{ of times}}} \lambda_e w_e.$$

In particular: only *active* vertices appear in Δ_v.

Definition

A k-repeating subsequence of length ℓ is a sequence of flips in which at least $\lceil \ell/k \rceil$ vertices flip at least twice.

Lemma (Etscheid & Röglin, 2017)

Any sequence of at least $5n$ *flips contains a* $\lceil 5 \log_2 n \rceil$*-repeating subsequence.*

Smarter Analysis

We now have

$$\Delta_v = \sum_{\substack{e=\{u,v\} \\ u \text{ flips off } \# \text{ of times}}} \lambda_e w_e.$$

In particular: only *active* vertices appear in Δ_v.

Definition

A k-repeating subsequence of length ℓ is a sequence of flips in which at least $\lceil \ell/k \rceil$ vertices flip at least twice.

Lemma (Etscheid & Röglin, 2017)

Any sequence of at least $5n$ flips contains a $\lceil 5 \log_2 n \rceil$-repeating subsequence.

Smarter Analysis

We now have

$$\Delta_v = \sum_{\substack{e=\{u,v\} \\ u \text{ flips off } \# \text{ of times}}} \lambda_e w_e.$$

In particular: only *active* vertices appear in Δ_v.

Definition

A k-repeating subsequence of length ℓ is a sequence of flips in which at least $\lceil \ell/k \rceil$ vertices flip at least twice.

Lemma (Etscheid & Röglin, 2017)

Any sequence of at least $5n$ flips contains a $\lceil 5 \log_2 n \rceil$-repeating subsequence.

Smarter Analysis

Need two more ingredients:

Lemma (Etscheid & Röglin, 2017)

A k-repeating sequence of length ℓ contains at least $\lceil \ell/(2k) \rceil$ linearly independent pairs of flips.

Lemma (Etscheid & Röglin, 2017)

Let $X_1 \dots X_m$ be independent random variables with densities bounded by ϕ. Let $\lambda_1, \dots, \lambda_k$ be linearly independent integral vectors. Then

$$\mathbb{P}\left(\textit{all } \lambda_i^T X \textit{ fall into } [0, \epsilon]\right) \leq (\epsilon\phi)^k.$$

To apply: $\lambda_i^T X$ correspond to Δ_v for each v that flips twice in a k-repeating sequence.

Smarter Analysis

Need two more ingredients:

Lemma (Etscheid & Röglin, 2017)

A k-repeating sequence of length ℓ contains at least $\lceil \ell/(2k) \rceil$ linearly independent pairs of flips.

Lemma (Etscheid & Röglin, 2017)

Let $X_1 \dots X_m$ be independent random variables with densities bounded by ϕ. Let $\lambda_1, \dots, \lambda_k$ be linearly independent integral vectors. Then

$$\mathbb{P}\left(\text{all } \lambda_i^T X \text{ fall into } [0, \epsilon]\right) \leq (\epsilon\phi)^k.$$

To apply: $\lambda_i^T X$ correspond to Δ_v for each v that flips twice in a k-repeating sequence.

Smarter Analysis

Need two more ingredients:

Lemma (Etscheid & Röglin, 2017)

A k-repeating sequence of length ℓ contains at least $\lceil \ell/(2k) \rceil$ linearly independent pairs of flips.

Lemma (Etscheid & Röglin, 2017)

Let $X_1 \dots X_m$ be independent random variables with densities bounded by ϕ. Let $\lambda_1, \dots, \lambda_k$ be linearly independent integral vectors. Then

$$\mathbb{P}\left(\textit{all } \lambda_i^T X \textit{ fall into } [0, \epsilon]\right) \leq (\epsilon\phi)^k.$$

To apply: $\lambda_i^T X$ correspond to Δ_v for each v that flips twice in a k-repeating sequence.

Smarter Analysis

Need two more ingredients:

Lemma (Etscheid & Röglin, 2017)

A k-repeating sequence of length ℓ contains at least $\lceil \ell/(2k) \rceil$ linearly independent pairs of flips.

Lemma (Etscheid & Röglin, 2017)

Let $X_1 \dots X_m$ be independent random variables with densities bounded by ϕ. Let $\lambda_1, \dots, \lambda_k$ be linearly independent integral vectors. Then

$$\mathbb{P}\left(\textit{all } \lambda_i^T X \textit{ fall into } [0, \epsilon]\right) \leq (\epsilon\phi)^k.$$

To apply: $\lambda_i^T X$ correspond to Δ_v for each v that flips twice in a k-repeating sequence.

Smarter Analysis

Lemma

Let Δ_{min} be the minimum improvement of any k-repeating sequence of length ℓ. Then

$$\mathbb{P}(\Delta \le \epsilon) \le 2^{\ell} \cdot n^{\ell} \cdot (\epsilon\phi)^{\lceil \ell/(2k) \rceil} = (2^{2k} n^{2k} \epsilon\phi)^{\lceil \ell/(2k) \rceil}.$$

Proof.

Fix a k-repeating sequence S. Since there are at least $\lceil \ell/(2k) \rceil$ linearly independent pairs of flips in S, probability that all pairs yield an improvement $\le \epsilon$ is at most $(\epsilon\phi)^{\lceil \ell/(2k) \rceil}$. Union bound over n^{ℓ} different sequences of length ℓ and all 2^{ℓ} starting configurations finishes the proof. $\square$

Smarter Analysis

Lemma

Let Δ_{min} be the minimum improvement of any k-repeating sequence of length ℓ. Then

$$\mathbb{P}(\Delta \leq \epsilon) \leq 2^{\ell} \cdot n^{\ell} \cdot (\epsilon\phi)^{\lceil \ell/(2k) \rceil} = (2^{2k} n^{2k} \epsilon\phi)^{\lceil \ell/(2k) \rceil}.$$

Proof.

Fix a k-repeating sequence S. Since there are at least $\lceil \ell/(2k) \rceil$ linearly independent pairs of flips in S, probability that all pairs yield an improvement $\leq \epsilon$ is at most $(\epsilon\phi)^{\lceil \ell/(2k) \rceil}$. Union bound over n^{ℓ} different sequences of length ℓ and all 2^{ℓ} starting configurations finishes the proof. $\square$

Finalizing the Proof

Theorem

The smoothed complexity of Max-Cut/Flip is $n^{O(\log n)}$.

Proof.

Fix any sequence of steps. Since any sequence of $5n$ steps contains a $\lceil 5\log_2 n\rceil$-repeating subsequence, we split the sequence into blocks of $5n$, and identify such a subsequence in each. The probability that any of these sequences improves the cut by at most ϵ is then at most $(2^{\lceil 5\log_2 n\rceil} n^{5\lceil \log_2 n\rceil}\epsilon\phi)^{n/\log_2 n}$. Write T for the number of iterations until Flip terminates. Then (cf. 2-opt)

$$\mathbb{P}(T \geq t) \leq (2^{\lceil 5\log_2 n\rceil} n^{5\lceil \log_2 n\rceil} n^2\phi/t)^{n/\log_2 n}.$$

The rest of the proof is analogous to 2-opt. $\square$

Finalizing the Proof

Theorem

The smoothed complexity of Max-Cut/Flip is $n^{O(\log n)}$.

Proof.

Fix any sequence of steps. Since any sequence of $5n$ steps contains a $\lceil 5 \log_2 n \rceil$-repeating subsequence, we split the sequence into blocks of $5n$, and identify such a subsequence in each. The probability that any of these sequences improves the cut by at most ϵ is then at most $(2^{\lceil 5 \log_2 n \rceil} n^{5 \lceil \log_2 n \rceil} \epsilon \phi)^{n / \log_2 n}$. Write T for the number of iterations until Flip terminates. Then (cf. 2-opt)

$$\mathbb{P}(T \geq t) \leq (2^{\lceil 5 \log_2 n \rceil} n^{5 \lceil \log_2 n \rceil} n^2 \phi / t)^{n / \log_2 n}.$$

The rest of the proof is analogous to 2-opt. $\square$