Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

The Traveling Salesman Problem Under Squared Euclidean Distances

Mark de Berg   Fred van Nijnatten   Gerhard Woeginger

TU Eindhoven

René Sitters

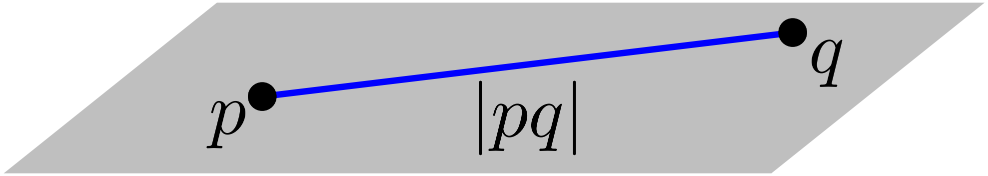
Vrije Universiteit Amsterdam

Alexander Wolff

Universität Würzburg

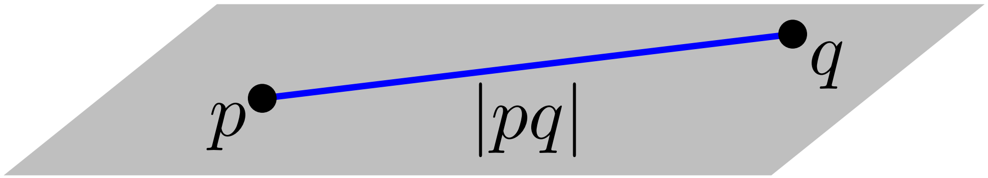
What's the Problem?

Notation. For points $p = (p_1, \ldots, p_d), q = (q_1, \ldots, q_d) \in \mathbb{R}^d$, denote by $|pq| = \sqrt{\sum_{i=1}^{d}(p_i - q_i)^2}$ their *Euclidean distance*.



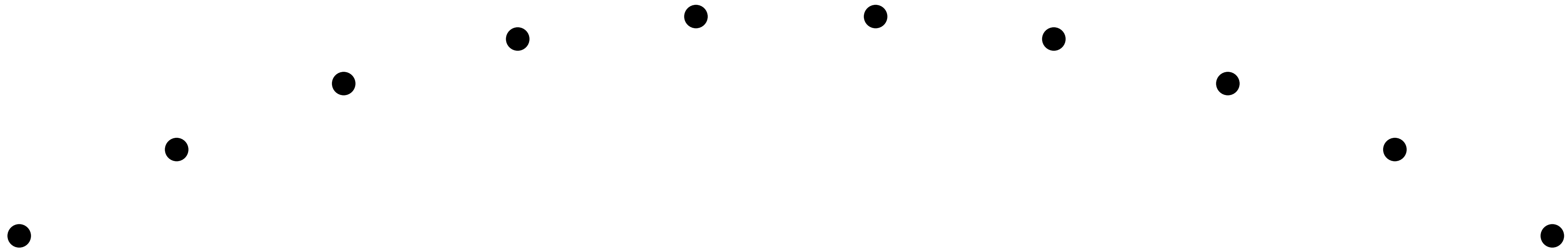
What's the Problem?

Notation. For points $p = (p_1, \ldots, p_d), q = (q_1, \ldots, q_d) \in \mathbb{R}^d$, denote by $|pq| = \sqrt{\sum_{i=1}^{d} (p_i - q_i)^2}$ their *Euclidean distance*.



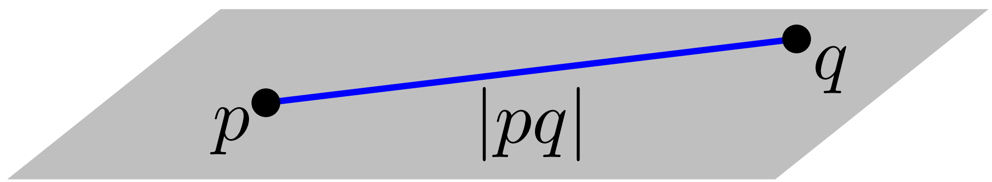
Problem. *Euclidean TSP*

Given a finite set $S \subset \mathbb{R}^d$, find a tour π through all points in S such that π has minimum length among all tours through S w.r.t. $| \cdot |$.



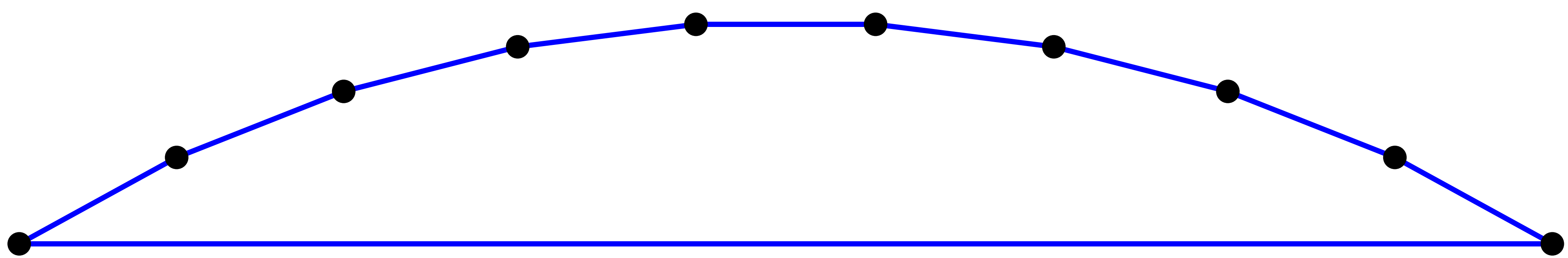
What's the Problem?

Notation. For points $p = (p_1, \ldots, p_d), q = (q_1, \ldots, q_d) \in \mathbb{R}^d$, denote by $|pq| = \sqrt{\sum_{i=1}^{d}(p_i - q_i)^2}$ their *Euclidean distance*.



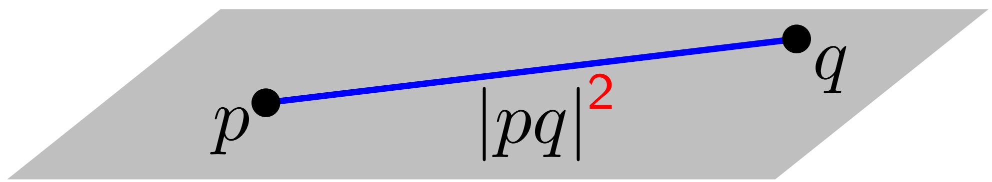
Problem. *Euclidean TSP*

Given a finite set $S \subset \mathbb{R}^d$, find a tour π through all points in S such that π has minimum length among all tours through S w.r.t. $|\cdot|$.



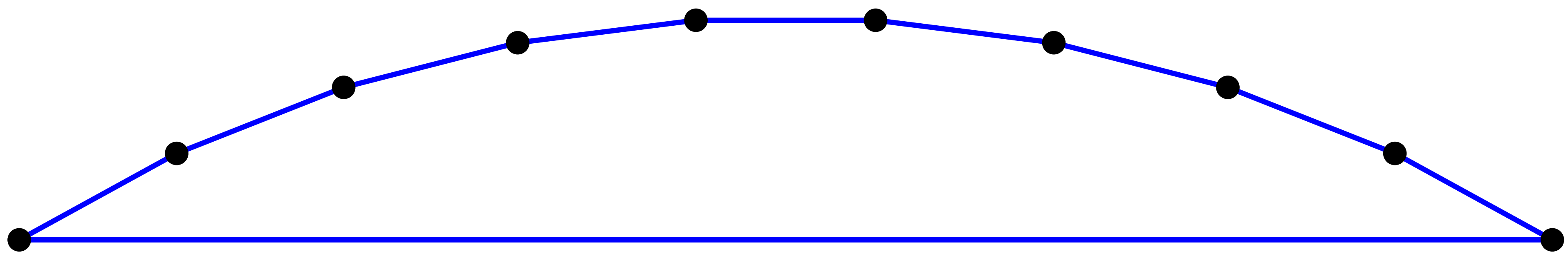
What's the Problem?

Notation. For points $p = (p_1, \ldots, p_d), q = (q_1, \ldots, q_d) \in \mathbb{R}^d$, denote by $|pq|^2 = \sqrt{\sum_{i=1}^{d}(p_i - q_i)^2}^2$ their *squared Euclidean distance*.



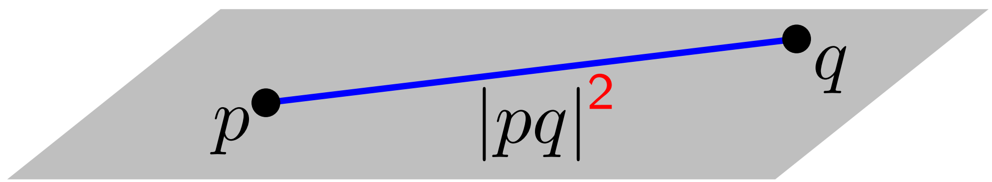
Problem. *Euclidean TSP*

Given a finite set $S \subset \mathbb{R}^d$, find a tour π through all points in S such that π has minimum length among all tours through S w.r.t. $|\cdot|$.



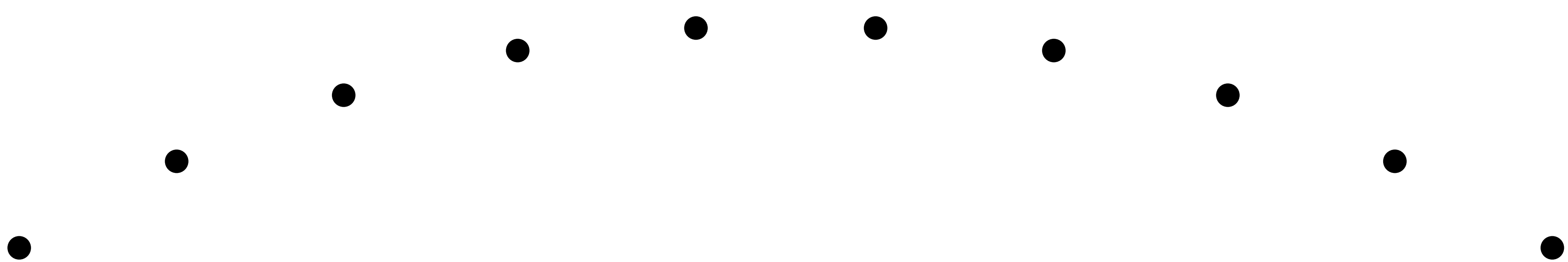
What's the Problem?

Notation. For points $p = (p_1, \ldots, p_d), q = (q_1, \ldots, q_d) \in \mathbb{R}^d$, denote by $|pq|^2 = \sqrt{\sum_{i=1}^{d} (p_i - q_i)^2}^{\,2}$ their *squared Euclidean distance*.



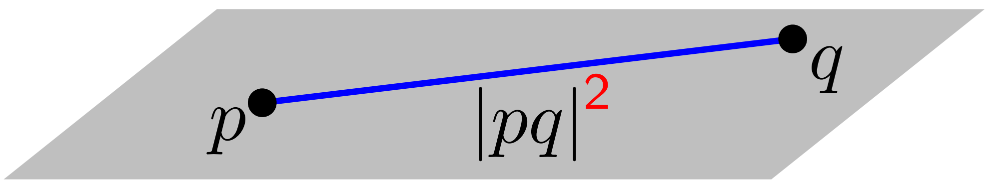
Problem. ~~*Euclidean*~~ *TSP*$(d, 2)$

Given a finite set $S \subset \mathbb{R}^d$, find a tour π through all points in S such that π has minimum length among all tours through S w.r.t. $|\cdot|^2$.



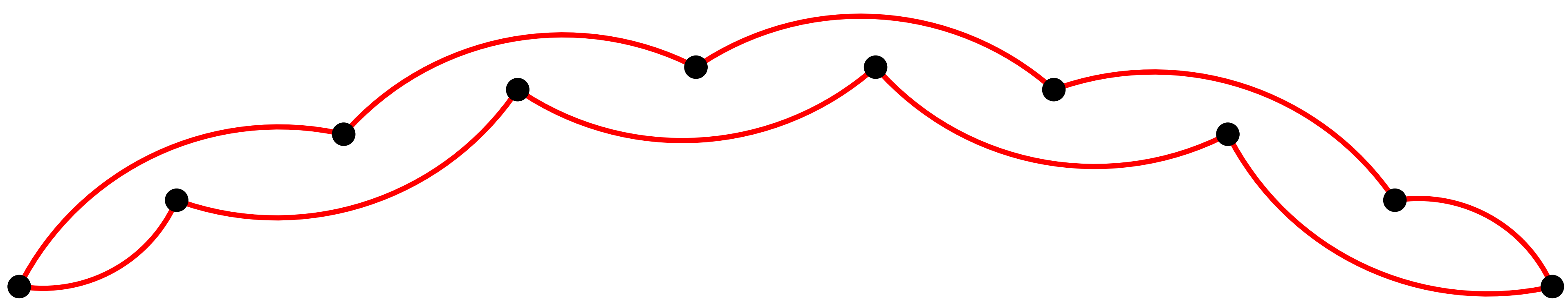
What's the Problem?

Notation. For points $p = (p_1, \ldots, p_d), q = (q_1, \ldots, q_d) \in \mathbb{R}^d$, denote by $|pq|^2 = \sqrt{\sum_{i=1}^{d}(p_i - q_i)^2}^{\,2}$ their *squared Euclidean distance.*



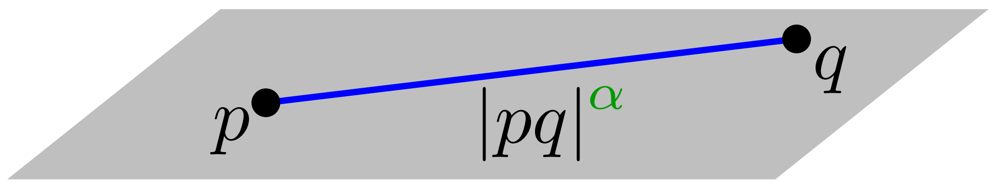
Problem. *~~Euclidean~~ TSP*$(d, 2)$

Given a finite set $S \subset \mathbb{R}^d$, find a tour π through all points in S such that π has minimum length among all tours through S w.r.t. $|\cdot|^2$.



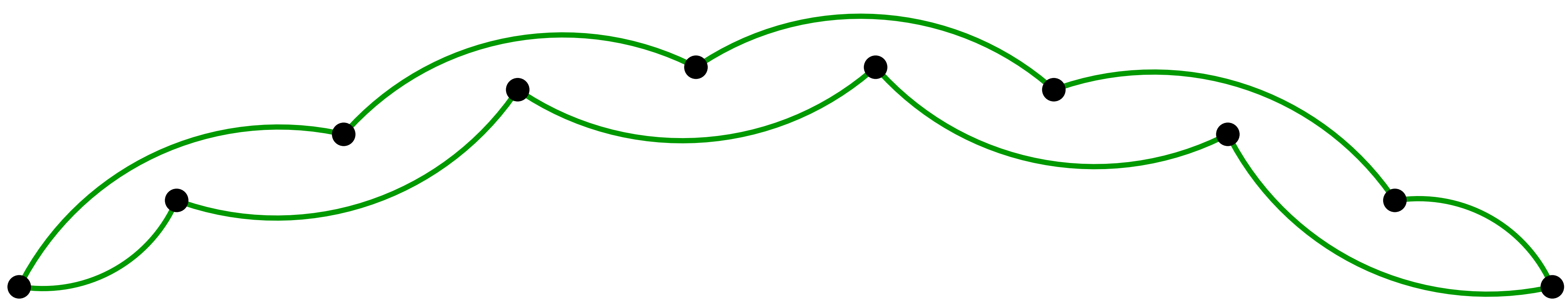
What's the Problem?

Notation. For points $p = (p_1, \ldots, p_d), q = (q_1, \ldots, q_d) \in \mathbb{R}^d$, denote by $|pq|^{\alpha} = \sqrt{\sum_{i=1}^{d}(p_i - q_i)^2}^{\,\alpha}$ their *power-α Euclidean distance.*



Problem. ~~*Euclidean*~~ *TSP*(d, α)

Given a finite set $S \subset \mathbb{R}^d$, find a tour π through all points in S such that π has minimum length among all tours through S w.r.t. $|\cdot|^{\alpha}$.

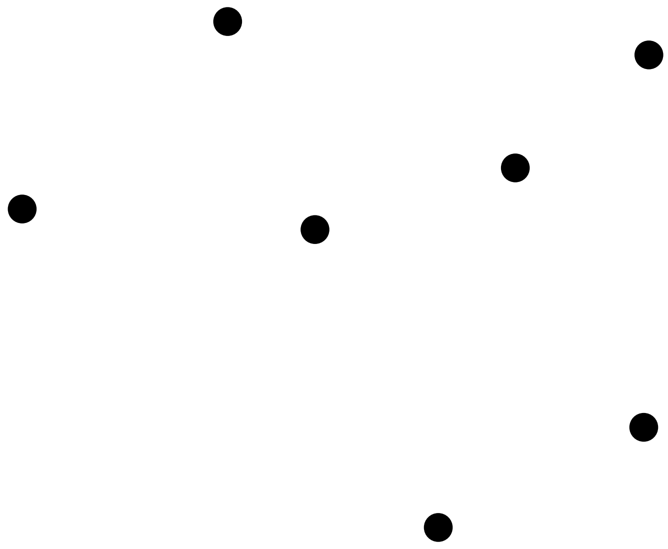


The Metric/Euclidean Case ($\alpha = 1$)

Theorem. [folklore]
The MST yields a 2-approximation for metric TSP.

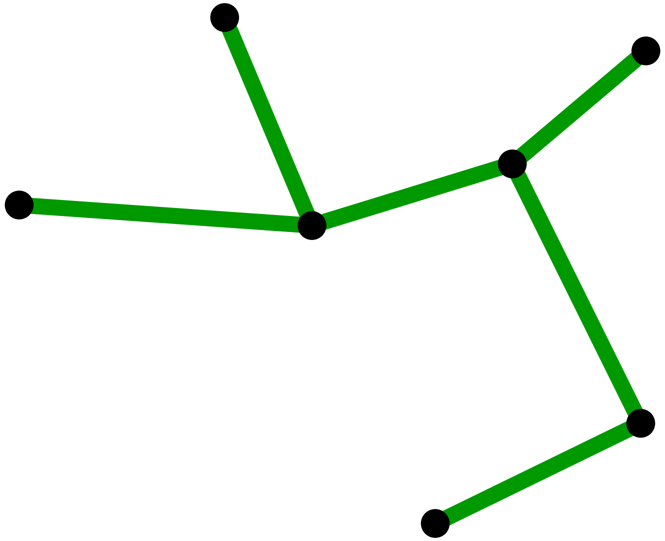
The Metric/Euclidean Case ($\alpha = 1$)

Theorem. [folklore]
The MST yields a 2-approximation for metric TSP.



The Metric/Euclidean Case ($\alpha = 1$)

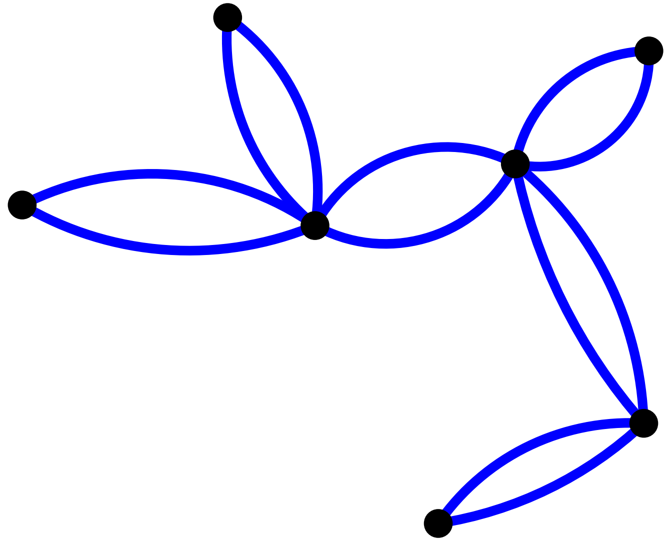
Theorem. [folklore]
The MST yields a 2-approximation for metric TSP.



The Metric/Euclidean Case ($\alpha = 1$)

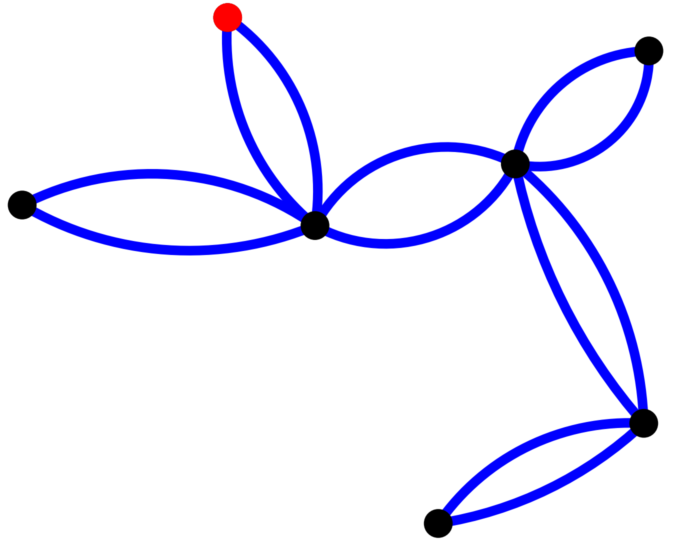
Theorem. [folklore]

The MST yields a 2-approximation for metric TSP.



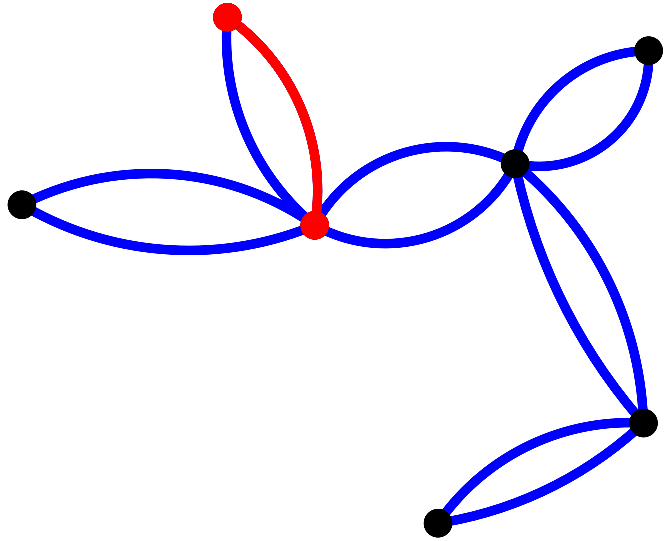
The Metric/Euclidean Case ($\alpha = 1$)

Theorem. [folklore]
The MST yields a 2-approximation for metric TSP.



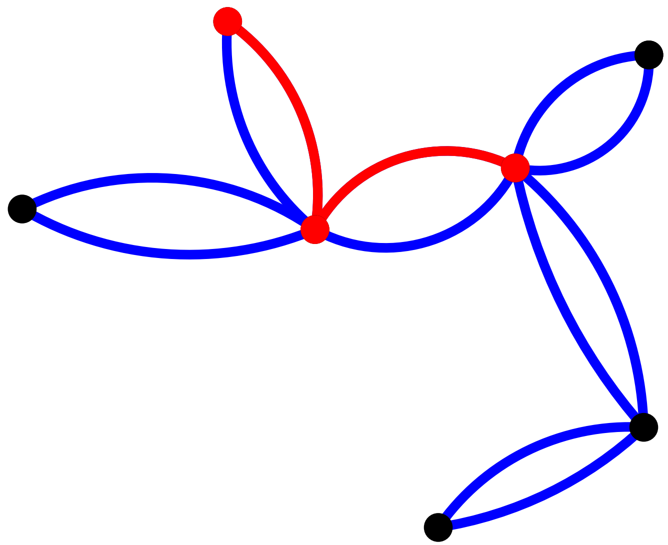
The Metric/Euclidean Case ($\alpha = 1$)

Theorem. [folklore]
The MST yields a 2-approximation for metric TSP.



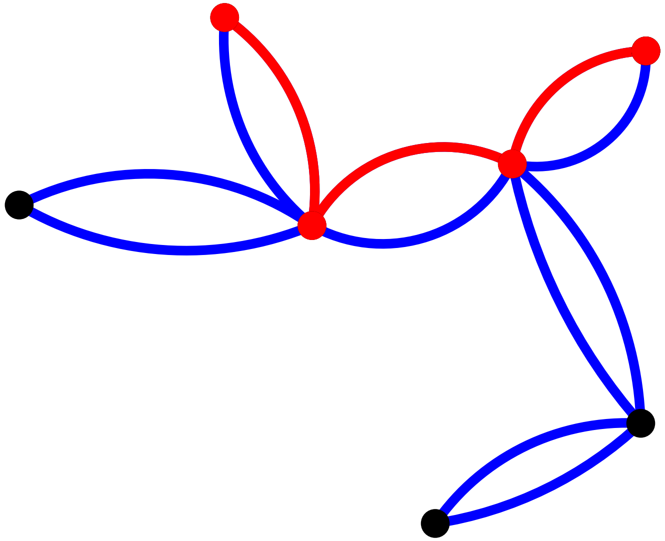
The Metric/Euclidean Case ($\alpha = 1$)

Theorem. [folklore]
The MST yields a 2-approximation for metric TSP.



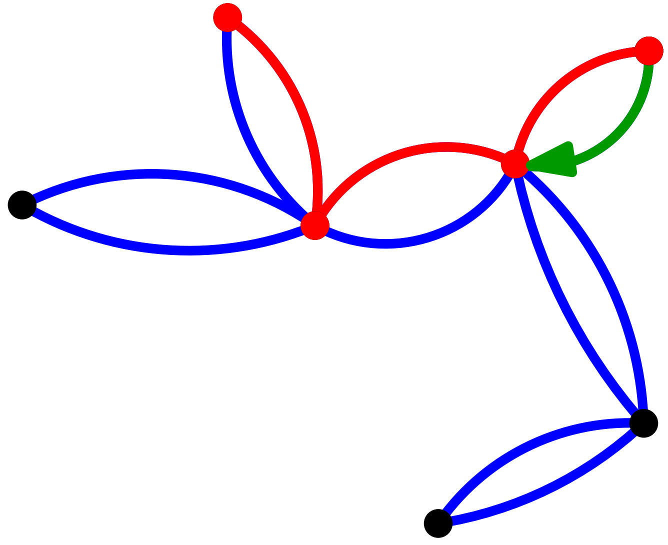
The Metric/Euclidean Case ($\alpha = 1$)

Theorem. [folklore]
The MST yields a 2-approximation for metric TSP.



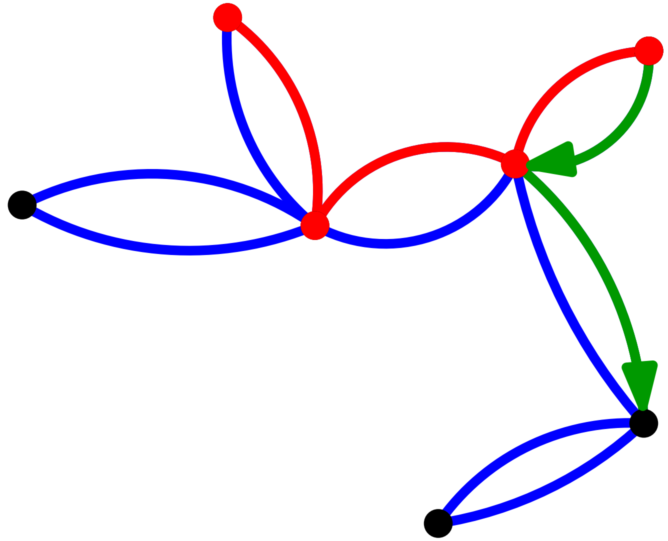
The Metric/Euclidean Case ($\alpha = 1$)

Theorem. [folklore]
The MST yields a 2-approximation for metric TSP.



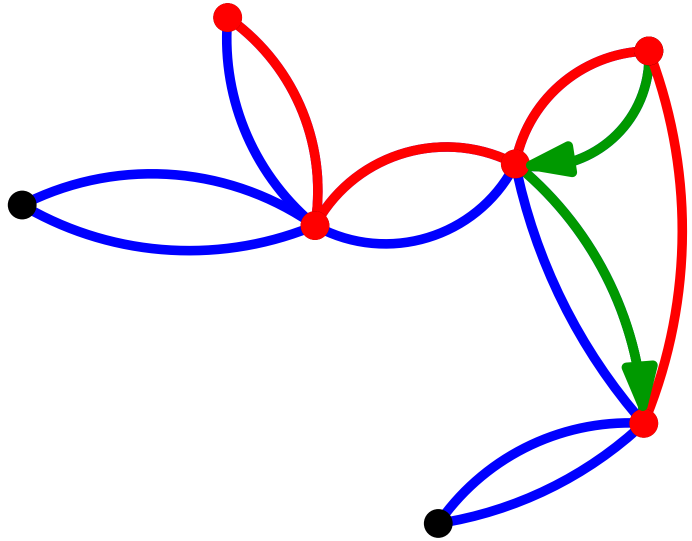
The Metric/Euclidean Case ($\alpha = 1$)

Theorem. [folklore]
The MST yields a 2-approximation for metric TSP.



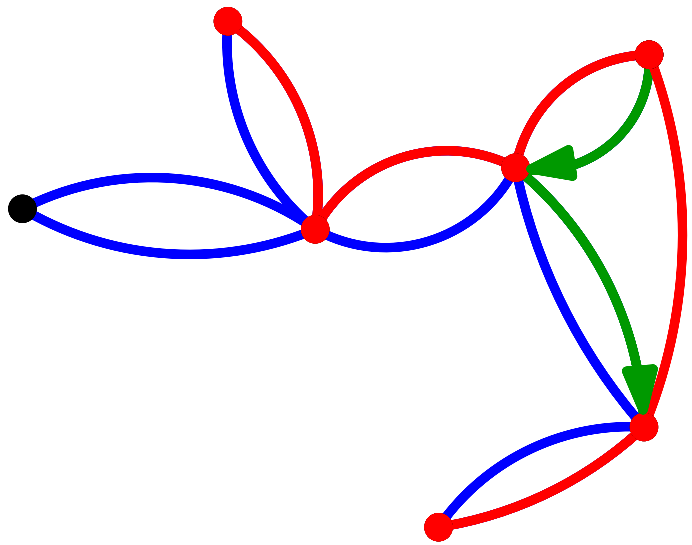
The Metric/Euclidean Case ($\alpha = 1$)

Theorem. [folklore]
The MST yields a 2-approximation for metric TSP.



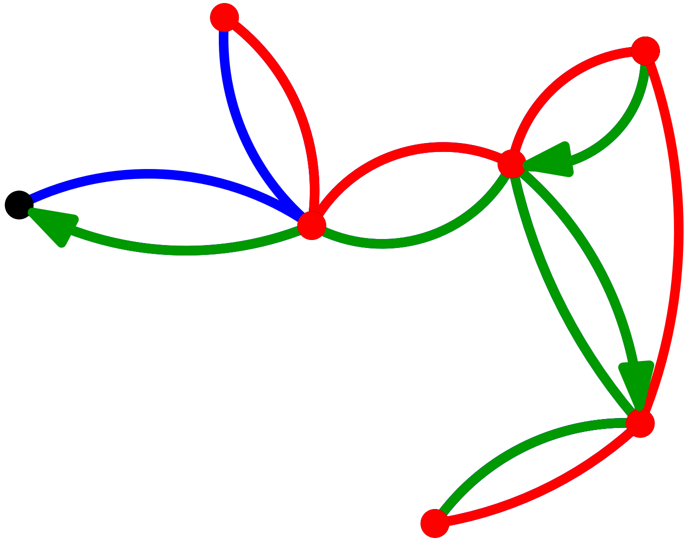
The Metric/Euclidean Case ($\alpha = 1$)

Theorem. [folklore]
The MST yields a 2-approximation for metric TSP.



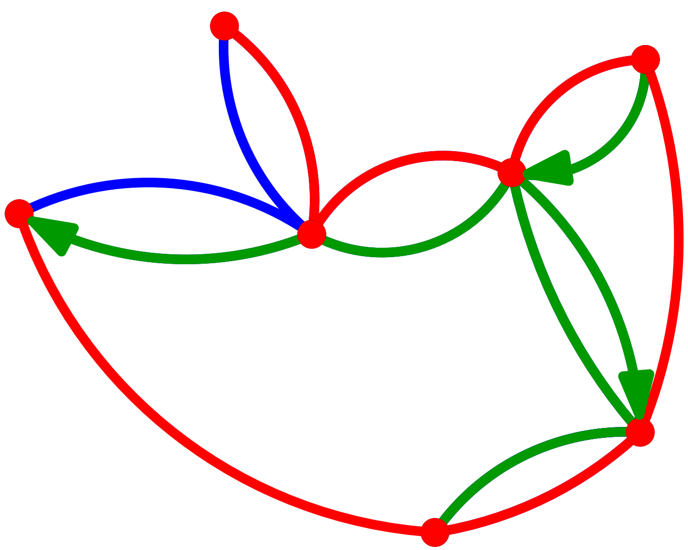
The Metric/Euclidean Case ($\alpha = 1$)

Theorem. [folklore]
The MST yields a 2-approximation for metric TSP.



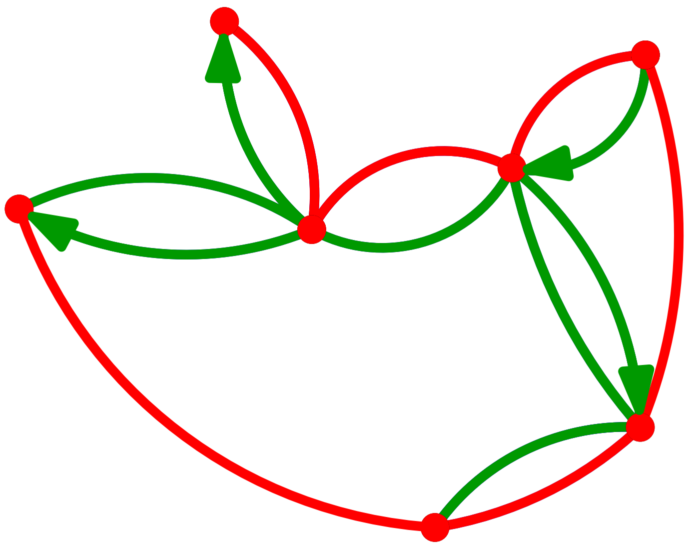
The Metric/Euclidean Case ($\alpha = 1$)

Theorem. [folklore]
The MST yields a 2-approximation for metric TSP.



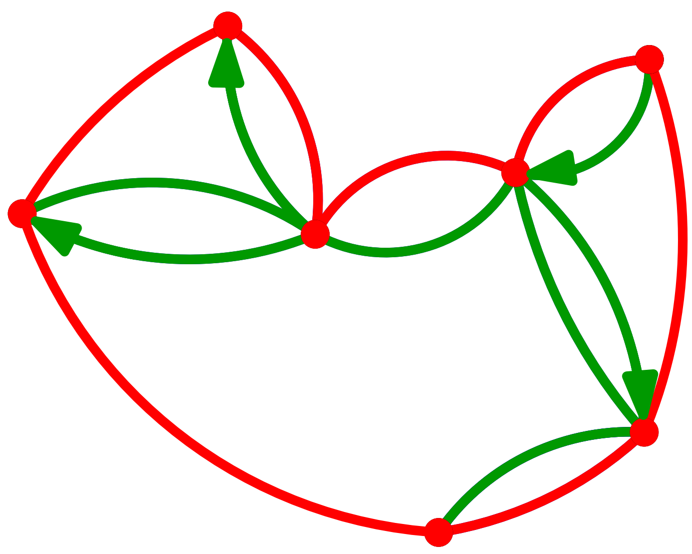
The Metric/Euclidean Case ($\alpha = 1$)

Theorem. [folklore]
The MST yields a 2-approximation for metric TSP.



The Metric/Euclidean Case ($\alpha = 1$)

Theorem. [folklore]
The MST yields a 2-approximation for metric TSP.



The Metric/Euclidean Case ($\alpha = 1$)

Theorem. [folklore]
The MST yields a 2-approximation for metric TSP.

Theorem. [Christofides'76]
There is a 3/2-approximation for metric TSP.

The Metric/Euclidean Case ($\alpha = 1$)

Theorem. [folklore]
The MST yields a 2-approximation for metric TSP.

Theorem. [Christofides'76]
There is a 3/2-approximation for metric TSP.

Theorem. [Papadimitriou'77]
Euclidean TSP is NP-hard in any dimension $d \geq 1$

The Metric/Euclidean Case ($\alpha = 1$)

Theorem. [folklore]
The MST yields a 2-approximation for metric TSP.

Theorem. [Christofides'76]
There is a 3/2-approximation for metric TSP.

Theorem. [Papadimitriou'77]
Euclidean TSP is NP-hard in any dimension $d \geq 1$

The Metric/Euclidean Case ($\alpha = 1$)

Theorem. [folklore]
The MST yields a 2-approximation for metric TSP.

Theorem. [Christofides'76]
There is a 3/2-approximation for metric TSP.

Theorem. [Papadimitriou'77]
Euclidean TSP is NP-hard in any dimension $d \geq 1$

Theorem. [Arora'96, Mitchell'96, RaoSmith'98]
Euclidean TSP admits a PTAS for any fixed $d \geq 1$.

The Metric/Euclidean Case ($\alpha = 1$)

Theorem. [folklore]
The MST yields a 2-approximation for metric TSP.

Theorem. [Christofides'76]
There is a 3/2-approximation for metric TSP.

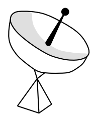
Theorem. [Papadimitriou'77]
Euclidean TSP is NP-hard in any dimension $d \geq 1$

Theorem. [Arora'96, Mitchell'96, RaoSmith'98]
Euclidean TSP admits a PTAS for any fixed $d \geq 1$.

But what about TSP(d, α)
for $\alpha \neq 1$?

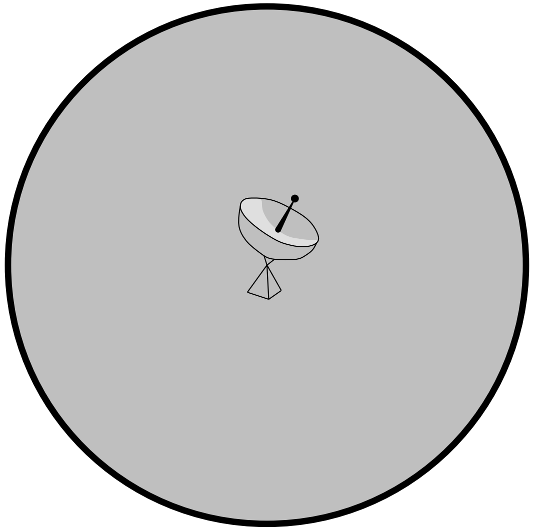
Motivation

1. Range assignment for wireless networks



Motivation

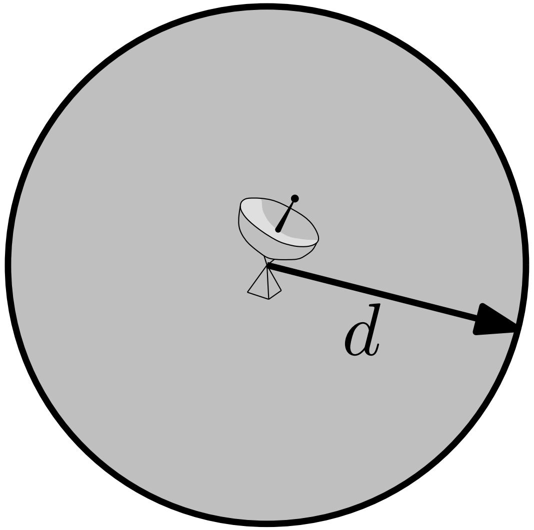
1. Range assignment for wireless networks



- transmission range depends on power

Motivation

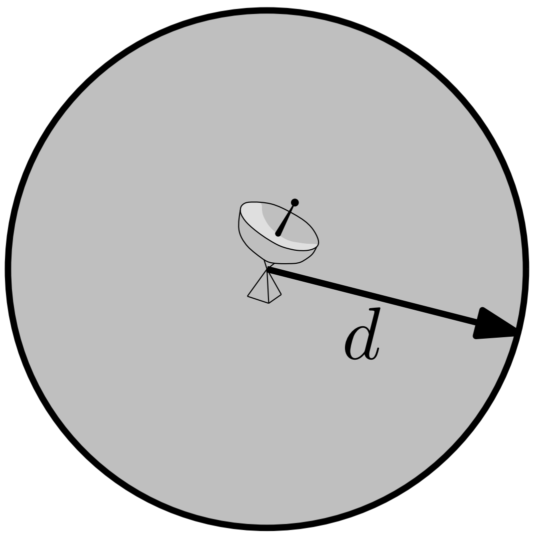
1. Range assignment for wireless networks



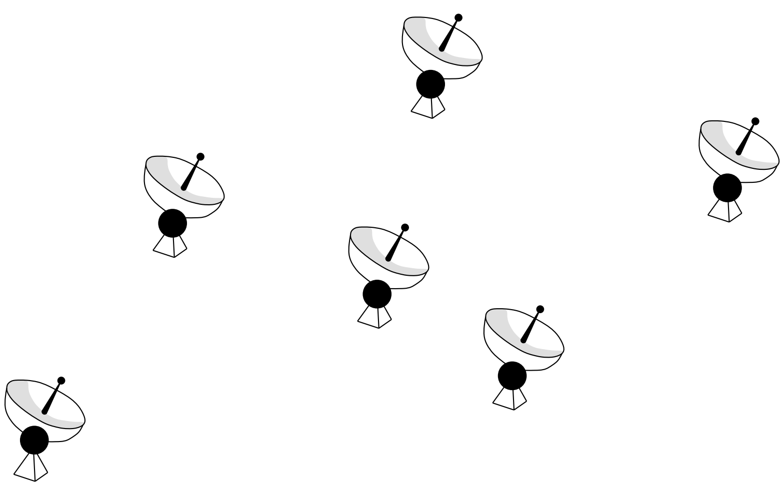
- transmission range depends on power
- energy consumption $\sim d^\alpha$
  for some $\alpha \in [2, 6]$
  (*"distance-power gradient"*)

Motivation

1. Range assignment for wireless networks

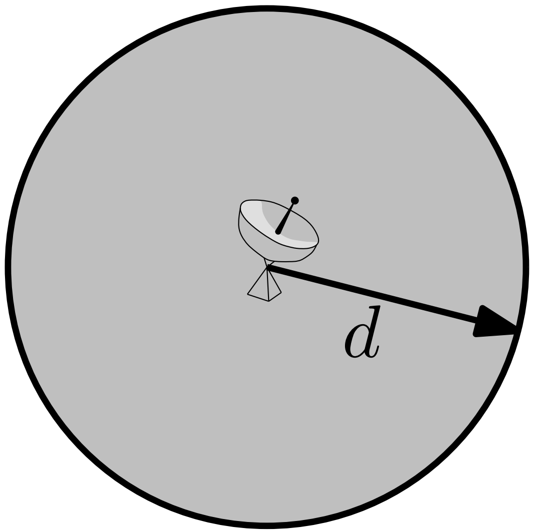


- transmission range depends on power
- energy consumption $\sim d^{\alpha}$
  for some $\alpha \in [2, 6]$
  (*"distance-power gradient"*)

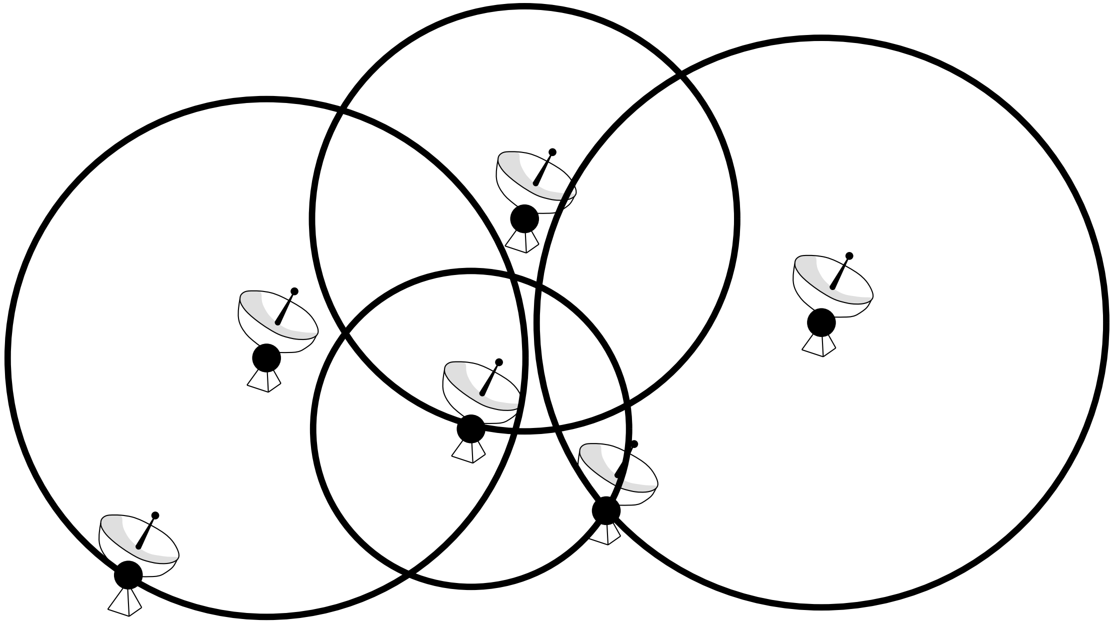


Motivation

1. Range assignment for wireless networks

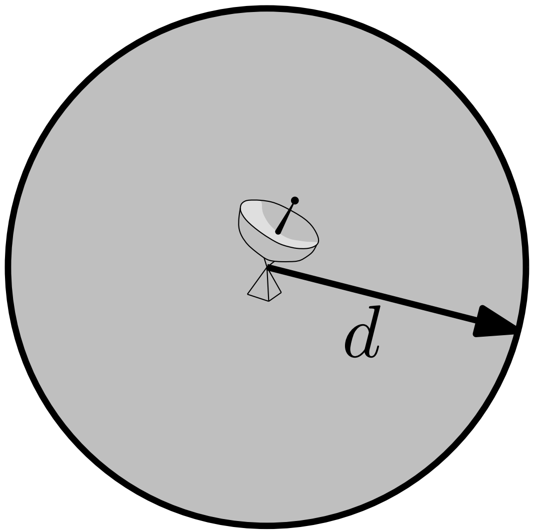


- transmission range depends on power
- energy consumption $\sim d^{\alpha}$
  for some $\alpha \in [2, 6]$
  (*"distance-power gradient"*)

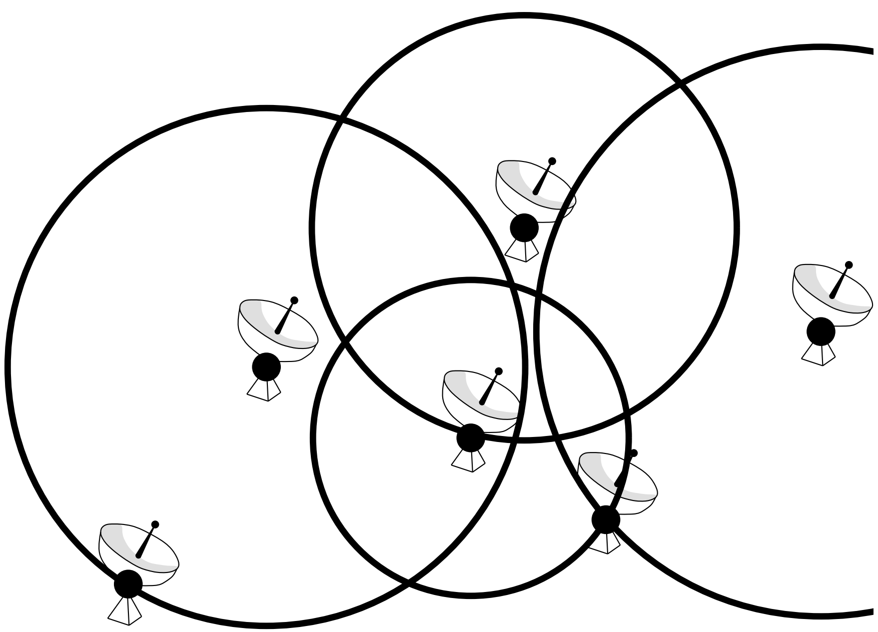


Motivation

1. Range assignment for wireless networks



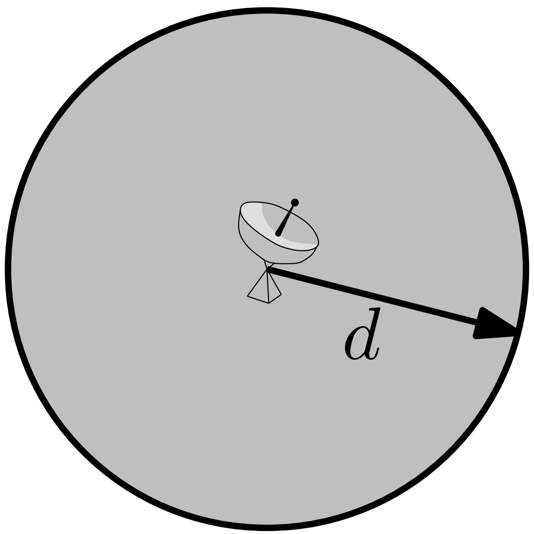
- transmission range depends on power
- energy consumption $\sim d^\alpha$
  for some $\alpha \in [2, 6]$
  (*"distance-power gradient"*)



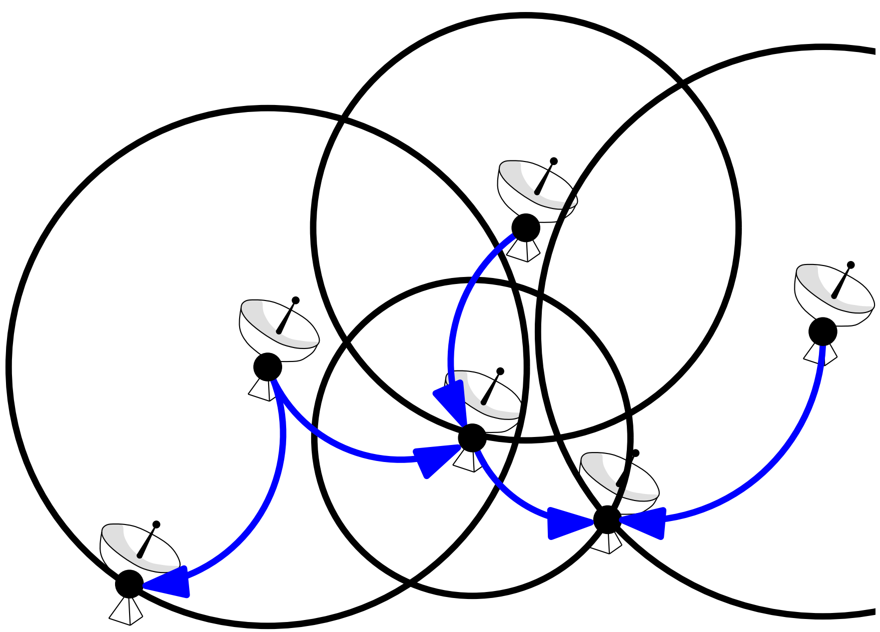
- range assignment ρ

Motivation

1. Range assignment for wireless networks



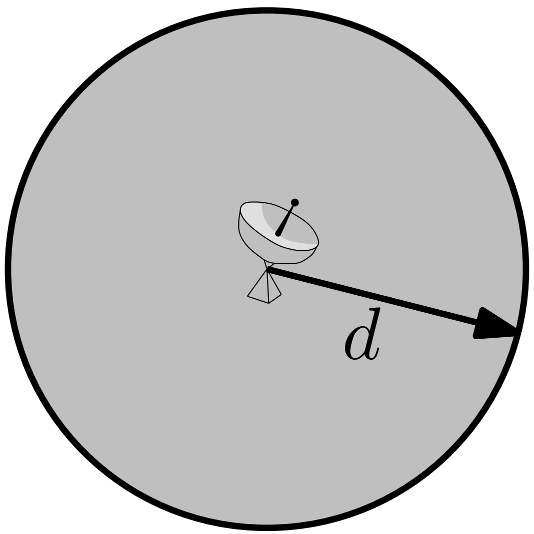
- transmission range depends on power
- energy consumption $\sim d^\alpha$ for some $\alpha \in [2, 6]$ (*"distance-power gradient"*)



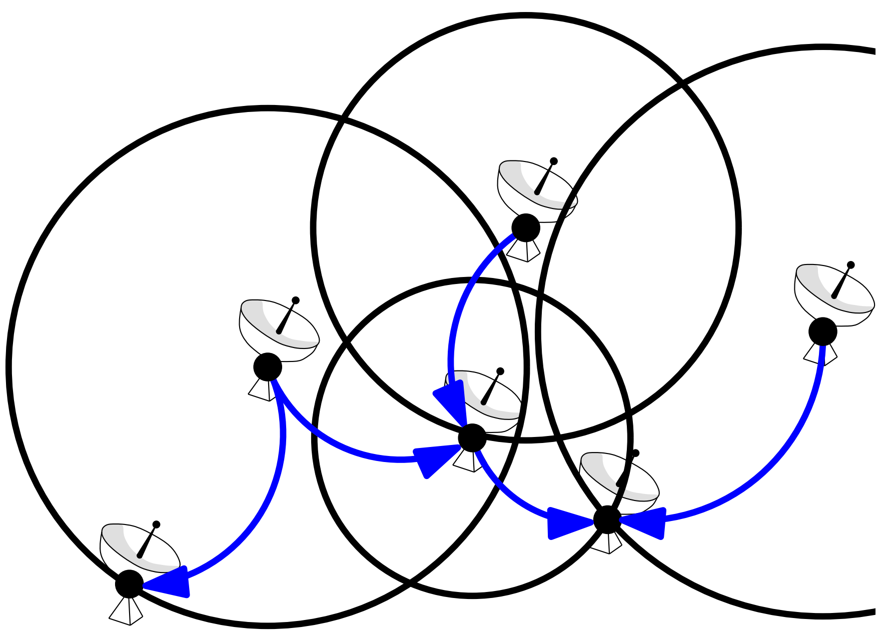
- range assignment ρ induces dir. communication graph G_ρ

Motivation

1. Range assignment for wireless networks



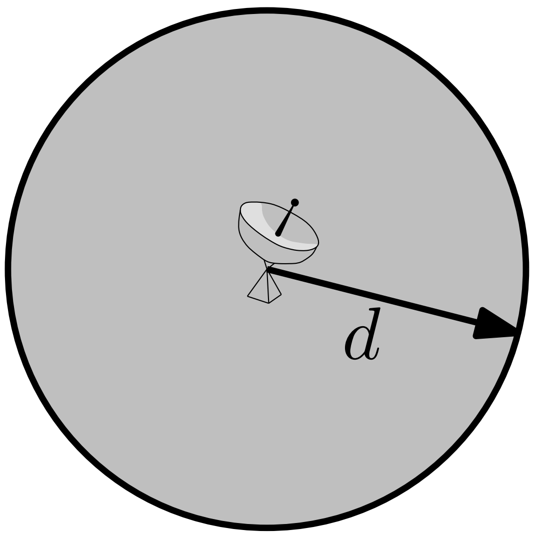
- transmission range depends on power
- energy consumption $\sim d^\alpha$ for some $\alpha \in [2, 6]$ (*"distance-power gradient"*)



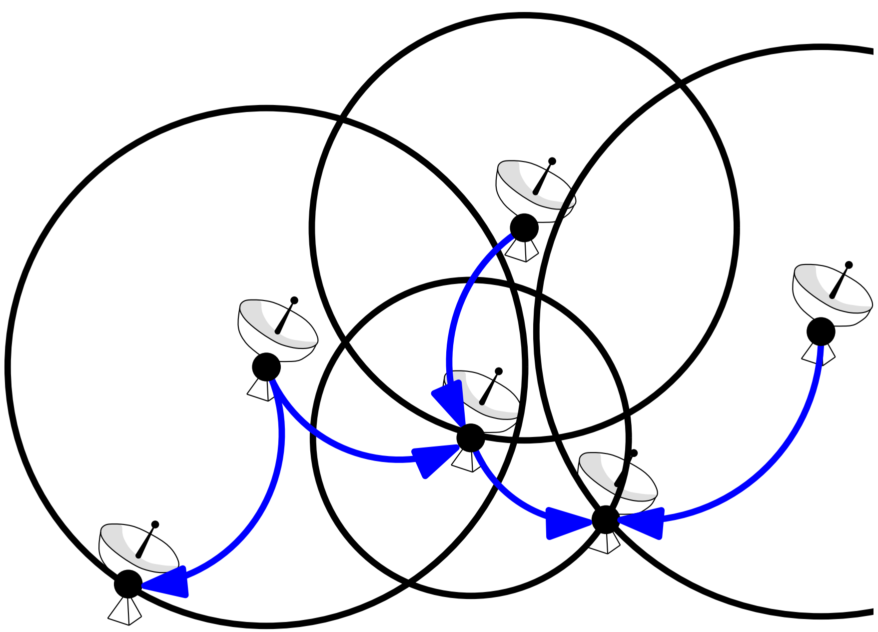
- range assignment ρ induces dir. communication graph G_ρ
- optimization problem: compute *min-energy* range assignment ρ s.t. G_ρ ...

Motivation

1. Range assignment for wireless networks



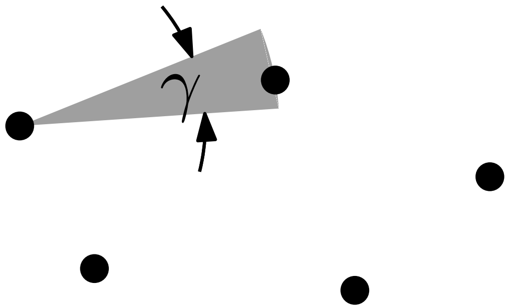
- transmission range depends on power
- energy consumption $\sim d^{\alpha}$ for some $\alpha \in [2, 6]$ (*"distance-power gradient"*)



- range assignment ρ induces dir. communication graph G_ρ
- optimization problem: compute *min-energy* range assignment ρ s.t. G_ρ ...
  – is strongly connected
  – contains broadcast tree
  – *contains tour* [Funke...'08]

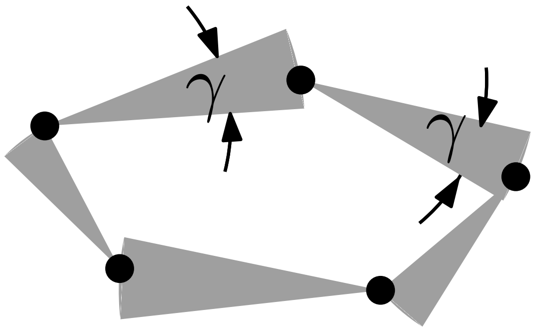
Motivation

2. Directional antennas with circular sectors [Caragiannis...'08]



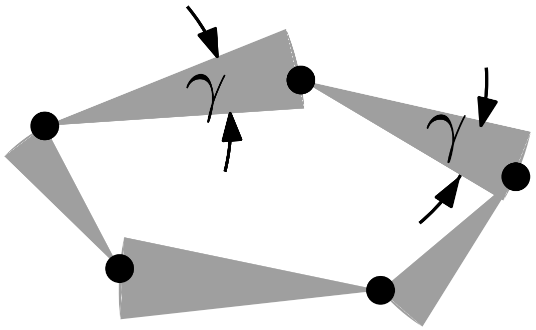
Motivation

2. Directional antennas with circular sectors [Caragiannis...'08]



Motivation

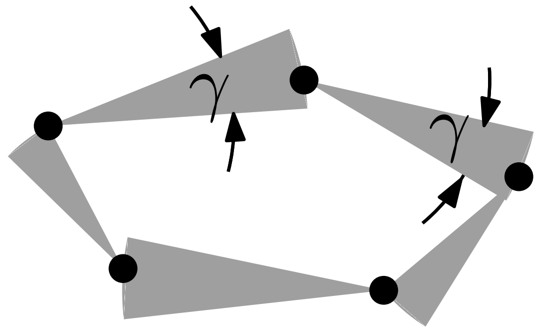
2. Directional antennas with circular sectors [Caragiannis…'08]



G_ρ strongly connected

Motivation

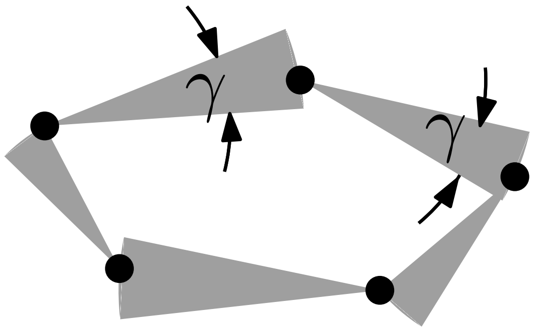
2. Directional antennas with circular sectors [Caragiannis...'08]



G_ρ strongly connected $\xrightarrow{\gamma \to 0}$

Motivation

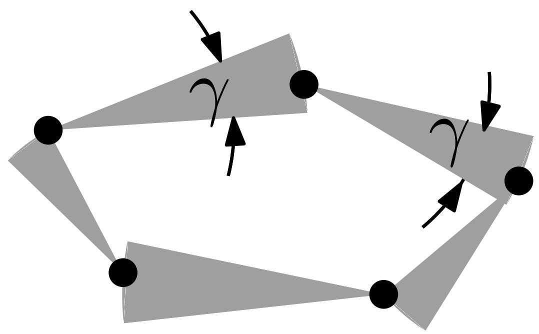
2. Directional antennas with circular sectors [Caragiannis...'08]



$$G_\rho \text{ strongly connected} \xrightarrow{\gamma \to 0} G_\rho \text{ tour}$$

Motivation

2. Directional antennas with circular sectors [Caragiannis...'08]



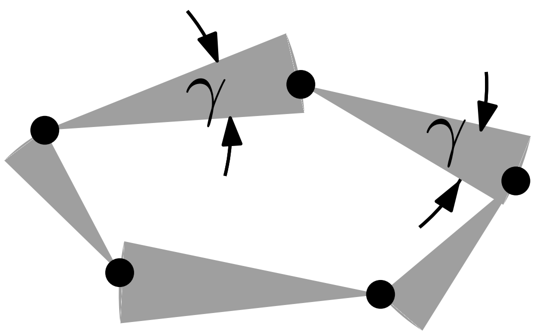
G_ρ strongly connected $\xrightarrow{\gamma \to 0}$ G_ρ tour

3. Complexity

Are things becoming simpler or harder?

Motivation

2. Directional antennas with circular sectors [Caragiannis...'08]



G_ρ strongly connected $\xrightarrow{\gamma \to 0}$ G_ρ tour

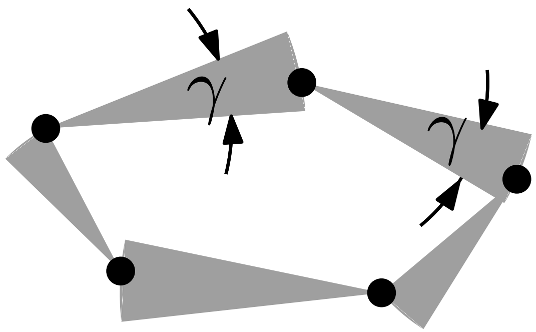
3. Complexity

Are things becoming simpler or harder?

Is Arora's PTAS for Euclidean TSP a "lucky coincidence"?

Motivation

2. Directional antennas with circular sectors [Caragiannis...'08]



G_ρ strongly connected $\xrightarrow{\gamma \to 0}$ G_ρ tour

3. Complexity

Are things becoming simpler or harder?

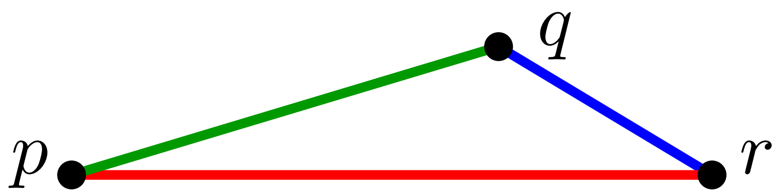
Is Arora's PTAS for Euclidean TSP a "lucky coincidence"?

If it is, how well can we approximate, say, TSP$(2, 2)$?

Previous Work

Definition. $\text{dist}(\cdot,\cdot)$ fulfills the τ-*relaxed triangle inequality* if any three points p, q, r satisfy

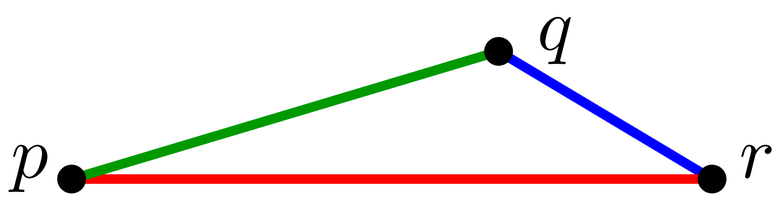
$$\text{dist}(p, r) \leq \tau \cdot (\text{dist}(p, q) + \text{dist}(q, r)).$$



Previous Work

Definition. $\text{dist}(\cdot,\cdot)$ fulfills the τ*-relaxed triangle inequality* if any three points p, q, r satisfy

$$\text{dist}(p,r) \leq \tau \cdot (\text{dist}(p,q) + \text{dist}(q,r)).$$



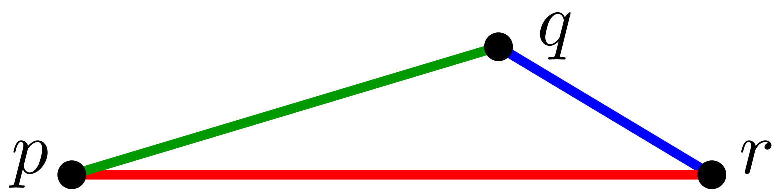
Lemma. [Funke...'08]

$|\cdot|^2$ fulfills the 2-relaxed triangle inequality.

Previous Work

Definition. dist$(\cdot,\cdot)$ fulfills the τ*-relaxed triangle inequality* if any three points p, q, r satisfy

$$\text{dist}(p,r) \leq \tau \cdot (\text{dist}(p,q) + \text{dist}(q,r)).$$

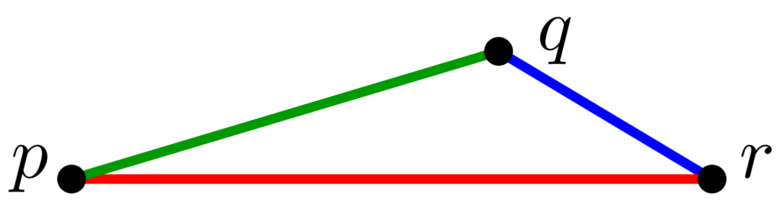


Lemma. [Funke...'08] For $\alpha \geq 1$, $|\cdot|^\alpha$ fulfills the $2^{\alpha-1}$-relaxed triangle inequality.

Previous Work

Definition. dist$(\cdot,\cdot)$ fulfills the τ*-relaxed triangle inequality* if any three points p, q, r satisfy

$$\text{dist}(p,r) \leq \tau \cdot (\text{dist}(p,q) + \text{dist}(q,r)).$$



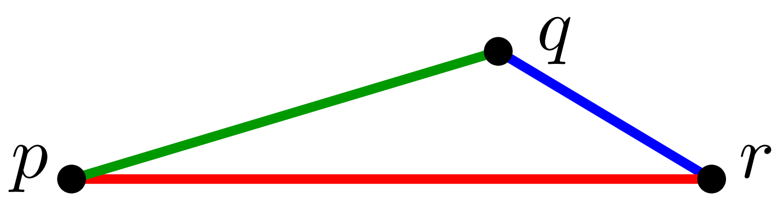
Lemma. [Funke...'08] For $\alpha \geq 1$, $|\cdot|^\alpha$ fulfills the $2^{\alpha-1}$-relaxed triangle inequality.

Good news. Can apply algorithms for Δ_τ-TSP to TSP$(\,\cdot\,,\alpha)$!

Previous Work

Definition. dist$(\cdot,\cdot)$ fulfills the τ-*relaxed triangle inequality* if any three points p, q, r satisfy

$$\text{dist}(p,r) \leq \tau \cdot (\text{dist}(p,q) + \text{dist}(q,r)).$$



Lemma. [Funke...'08] For $\alpha \geq 1$, $|\cdot|^\alpha$ fulfills the $2^{\alpha-1}$-relaxed triangle inequality.

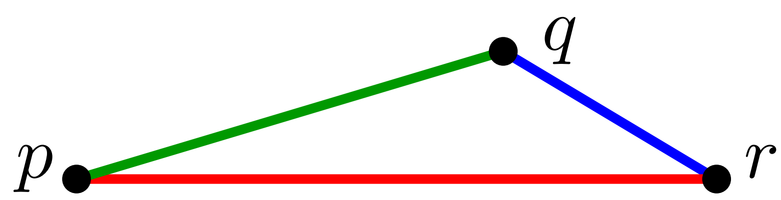
Good news. Can apply algorithms for Δ_τ-TSP to TSP$(\cdot, \alpha)$!

Δ_τ	TSP$(\cdot, \alpha)$	
$\tau^2 + \tau$	$4^{\alpha-1} + 2^{\alpha-1}$	[Andreae'01]
4τ	$2^{\alpha+1}$	[BenderChekuri'00]

Previous Work

Definition. dist$(\cdot,\cdot)$ fulfills the τ-*relaxed triangle inequality* if any three points p, q, r satisfy

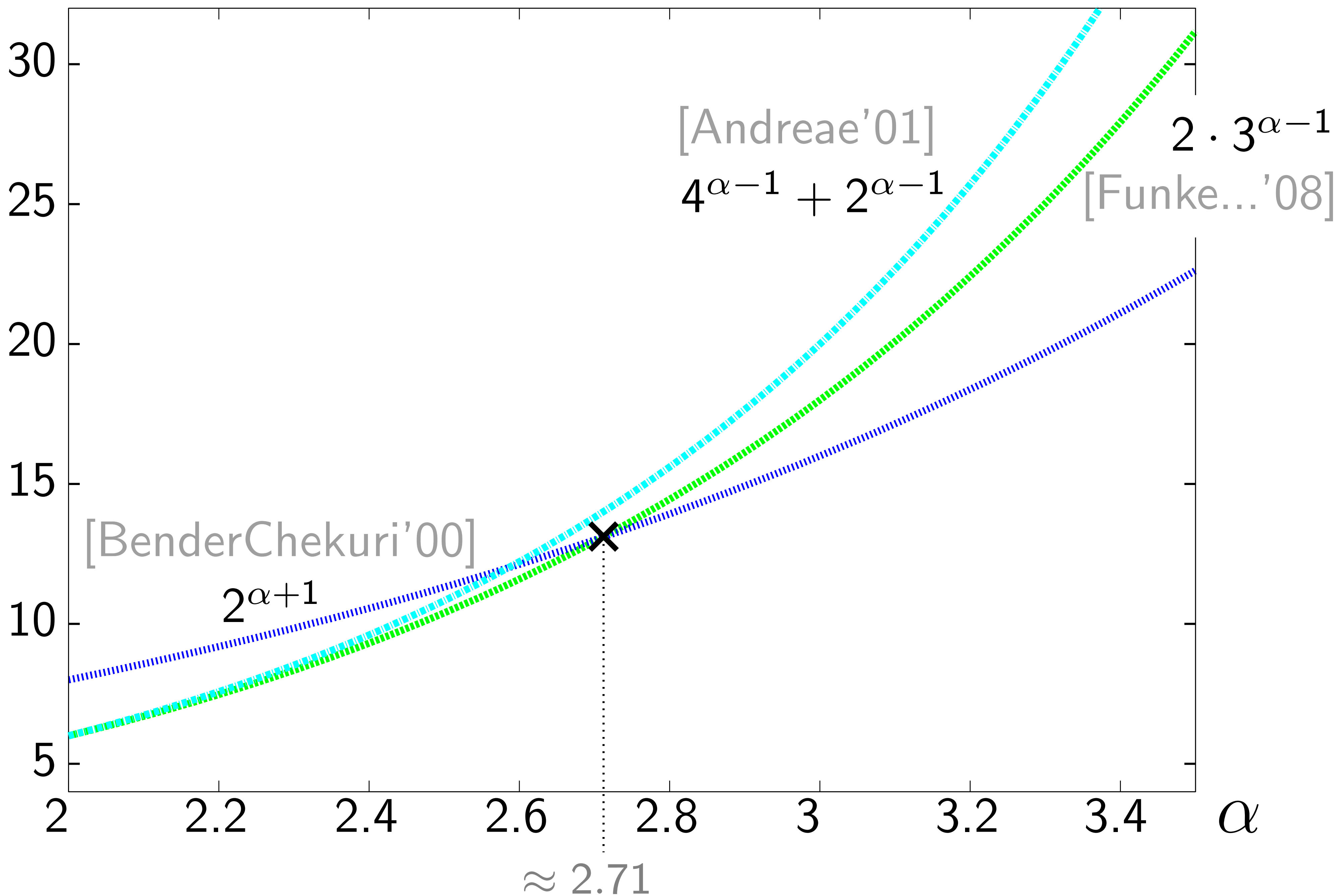
$$\text{dist}(p,r) \leq \tau \cdot (\text{dist}(p,q) + \text{dist}(q,r)).$$



Lemma. [Funke...'08] For $\alpha \geq 1$, $|\cdot|^\alpha$ fulfills the $2^{\alpha-1}$-relaxed triangle inequality.

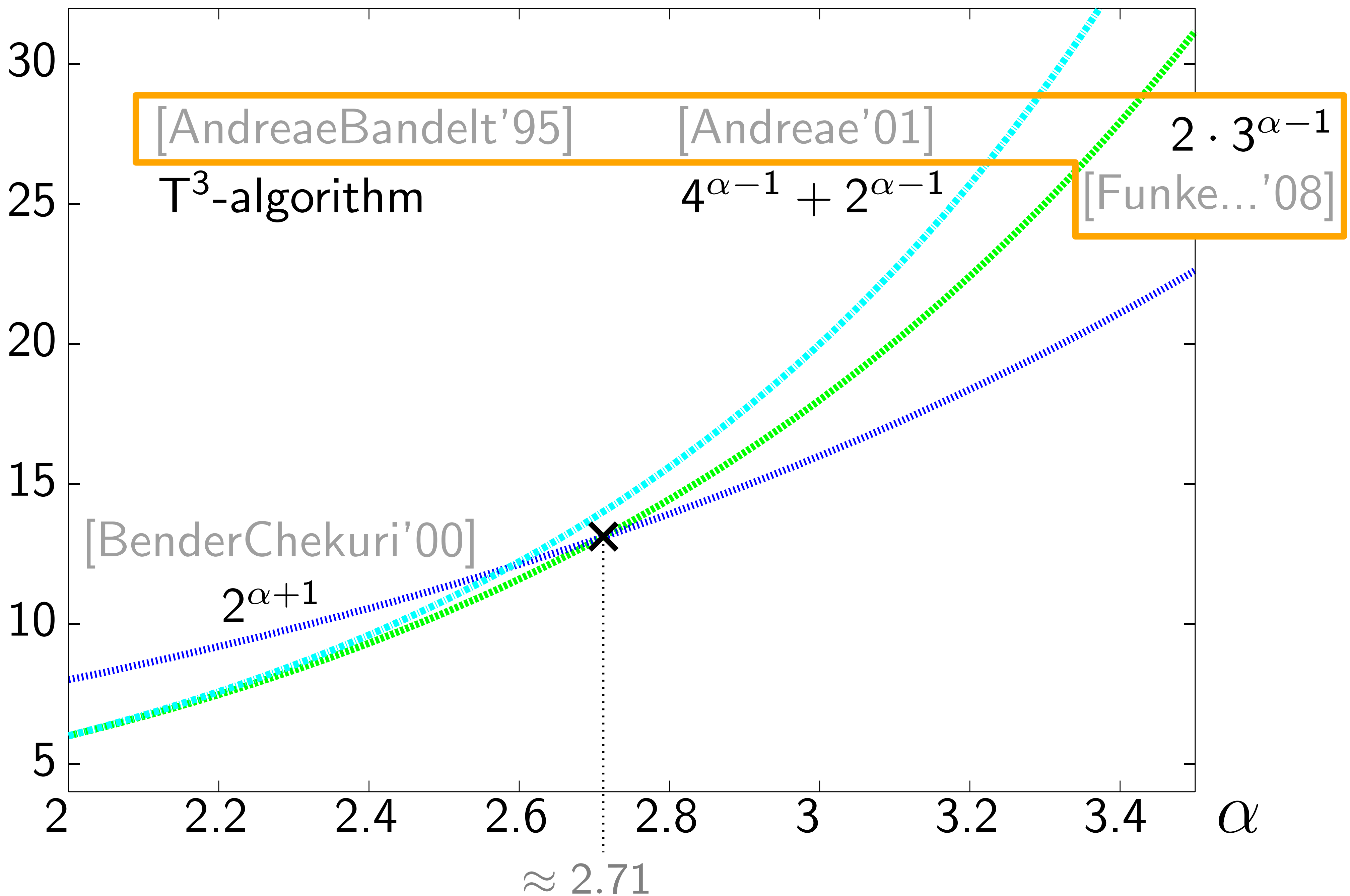
Good news. Can apply algorithms for Δ_τ-TSP to TSP$(\cdot, \alpha)$!

Δ_τ	TSP$(\cdot, \alpha)$	
$\tau^2 + \tau$	$4^{\alpha-1} + 2^{\alpha-1}$	[Andreae'01]
4τ	$2^{\alpha+1}$	[BenderChekuri'00]
—	$3 \cdot 2^{\alpha-1}$	[Funke...'08]



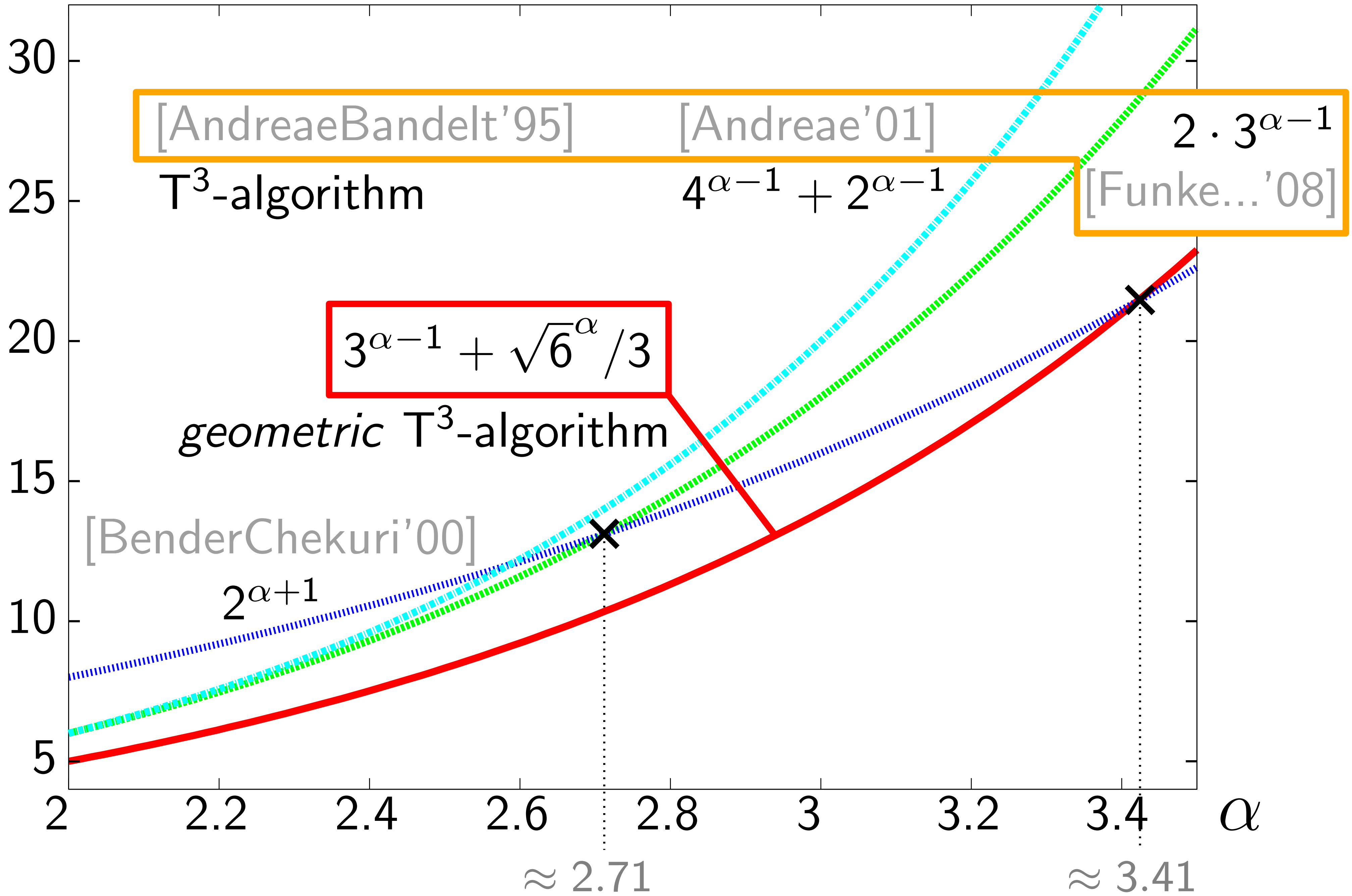
[Andreae'01]
$4^{\alpha-1} + 2^{\alpha-1}$
$2 \cdot 3^{\alpha-1}$
[Funke...'08]
[BenderChekuri'00]
$2^{\alpha+1}$
30
25
20
15
10
5
2
2.2
2.4
2.6
2.8
3
3.2
3.4
α
≈ 2.71

Our Results



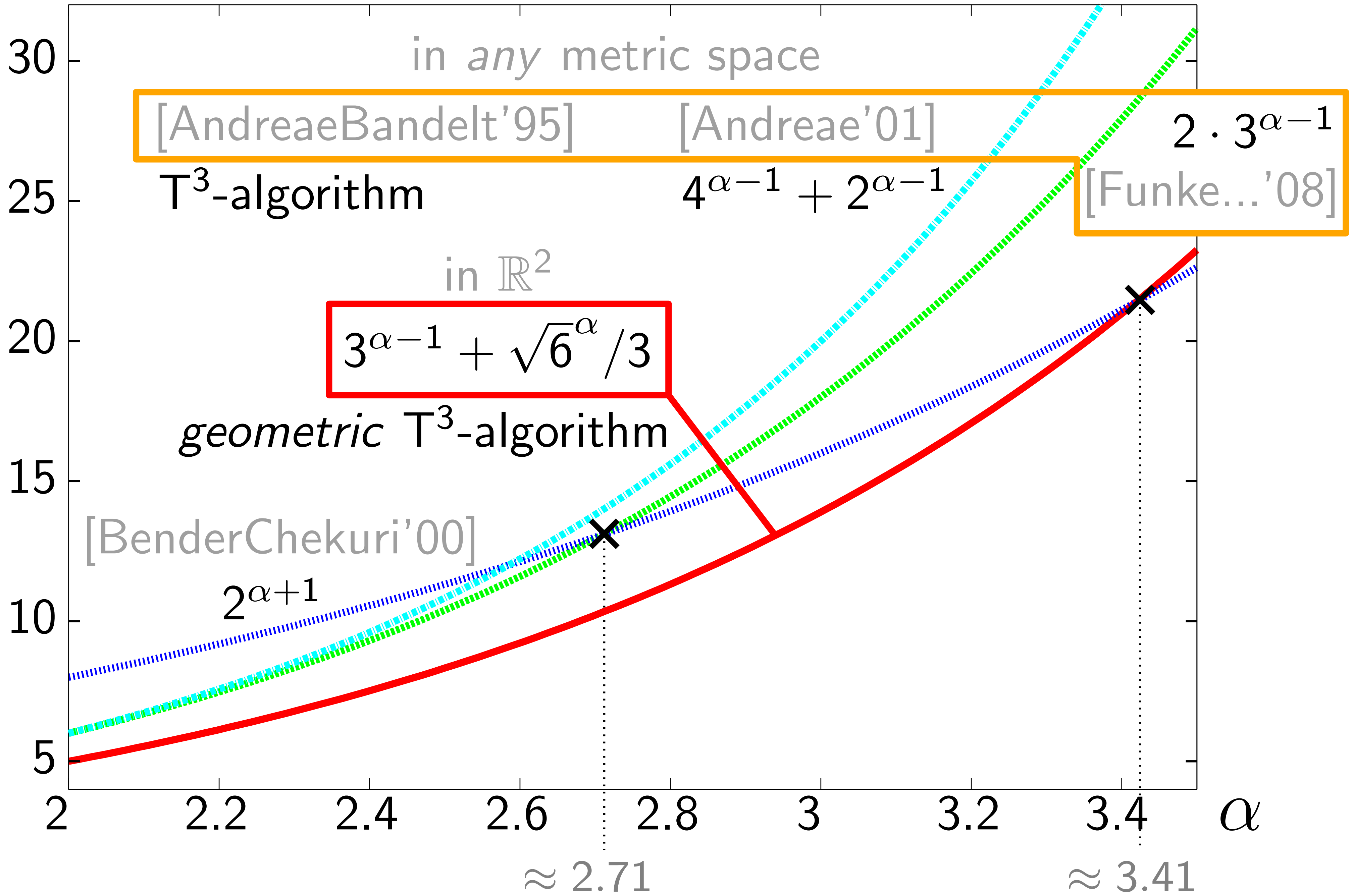
30
25
20
15
10
5
2
2.2
2.4
2.6
2.8
3
3.2
3.4
α
[AndreaeBandelt'95]
[Andreae'01]
$2 \cdot 3^{\alpha-1}$
T^3-algorithm
$4^{\alpha-1} + 2^{\alpha-1}$
[Funke...'08]
[BenderChekuri'00]
$2^{\alpha+1}$
≈ 2.71

Our Results



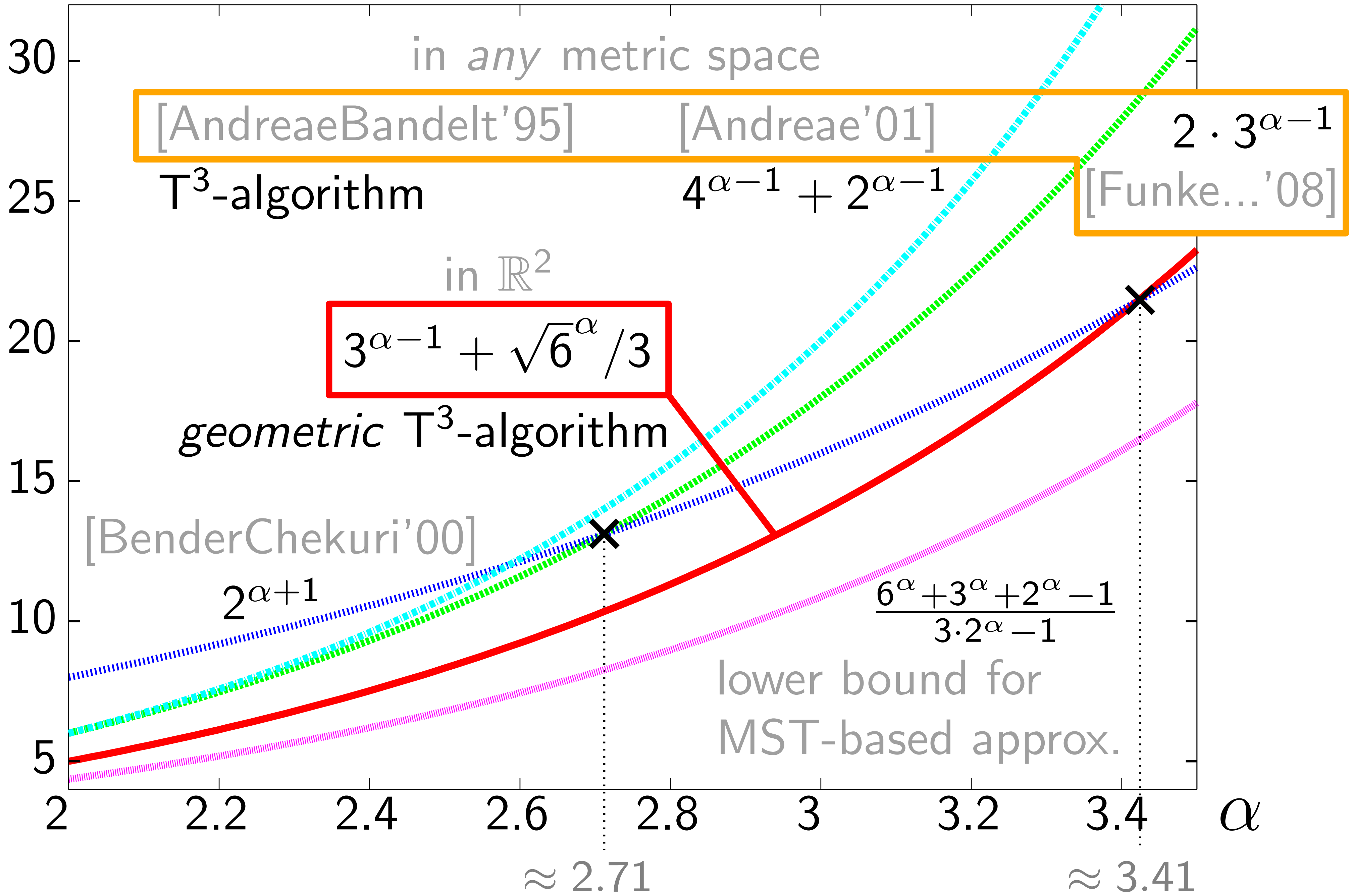
[AndreaeBandelt'95]
T^3-algorithm
[Andreae'01]
$4^{\alpha-1} + 2^{\alpha-1}$
$2 \cdot 3^{\alpha-1}$
[Funke...'08]
$3^{\alpha-1} + \sqrt{6}^{\alpha}/3$
geometric T^3-algorithm
[BenderChekuri'00]
$2^{\alpha+1}$
30
25
20
15
10
5
2
2.2
2.4
2.6
2.8
3
3.2
3.4
α
≈ 2.71
≈ 3.41

Our Results



in *any* metric space
[AndreaeBandelt'95]
[Andreae'01]
$2 \cdot 3^{\alpha-1}$
T3-algorithm
$4^{\alpha-1} + 2^{\alpha-1}$
[Funke...'08]
in $\mathbb{R}^2$
$3^{\alpha-1} + \sqrt{6}^{\alpha}/3$
geometric T3-algorithm
[BenderChekuri'00]
$2^{\alpha+1}$
30
25
20
15
10
5
2
2.2
2.4
2.6
2.8
3
3.2
3.4
α
≈ 2.71
≈ 3.41

Our Results

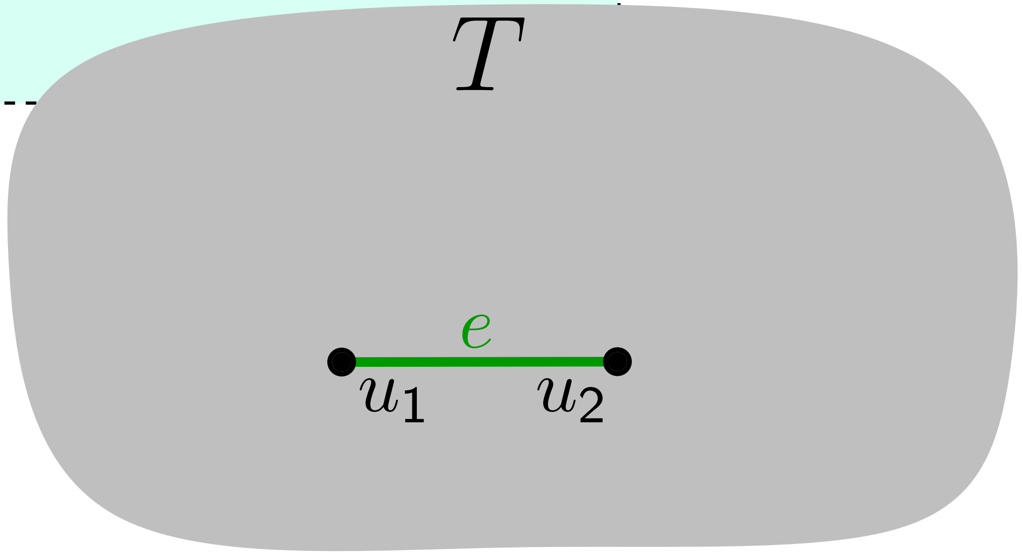


in *any* metric space
[AndreaeBandelt'95]
T^3-algorithm
[Andreae'01]
$4^{\alpha-1} + 2^{\alpha-1}$
$2 \cdot 3^{\alpha-1}$
[Funke...'08]
in $\mathbb{R}^2$
$3^{\alpha-1} + \sqrt{6}^{\alpha}/3$
geometric T^3-algorithm
[BenderChekuri'00]
$2^{\alpha+1}$
$\frac{6^\alpha+3^\alpha+2^\alpha-1}{3\cdot 2^\alpha-1}$
lower bound for
MST-based approx.
30
25
20
15
10
5
2
2.2
2.4
2.6
2.8
3
3.2
3.4
α
≈ 2.71
≈ 3.41

The T^3-Algorithm

[Sekanina'60, AndreaeBandelt'95]

$\textsc{CycleInCube}(T, e = u_1u_2)$

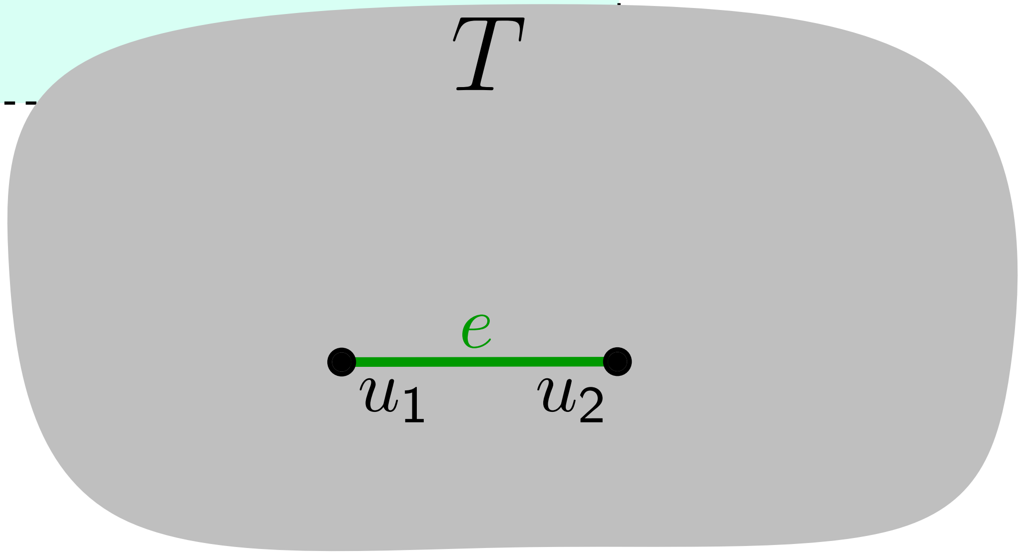


The T^3-Algorithm

[Sekanina'60, AndreaeBandelt'95]

$\textsc{CycleInCube}(T, e = u_1u_2)$

Take MST of given point set!



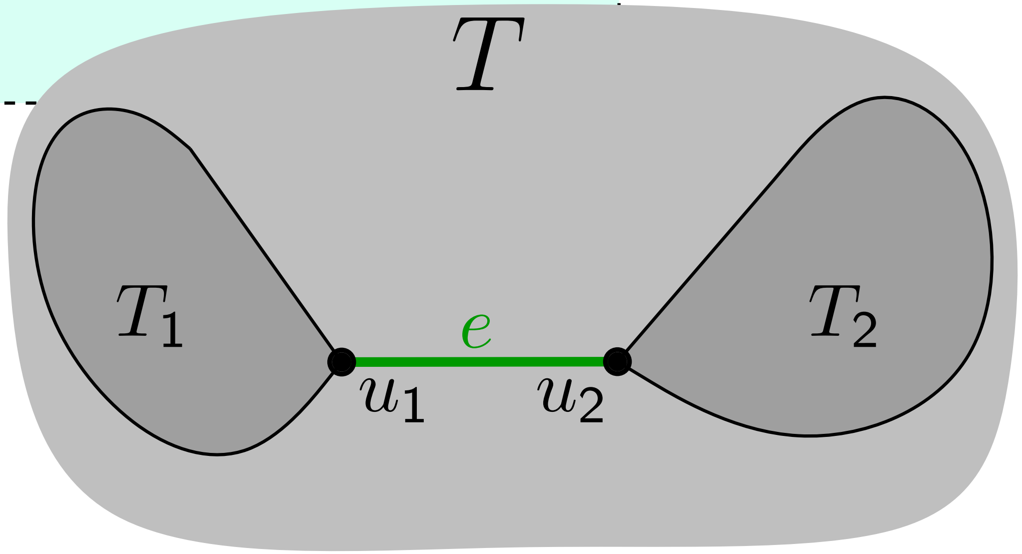
The T^3-Algorithm

[Sekanina'60, AndreaeBandelt'95]

CycleInCube(T, $e = u_1 u_2$)

for $i \leftarrow 1$ **to** 2 **do**

$T_i \leftarrow$ component of $T - e$ that contains u_i



The T^3-Algorithm

[Sekanina'60, AndreaeBandelt'95]

$\textsc{CycleInCube}(T, e = u_1 u_2)$

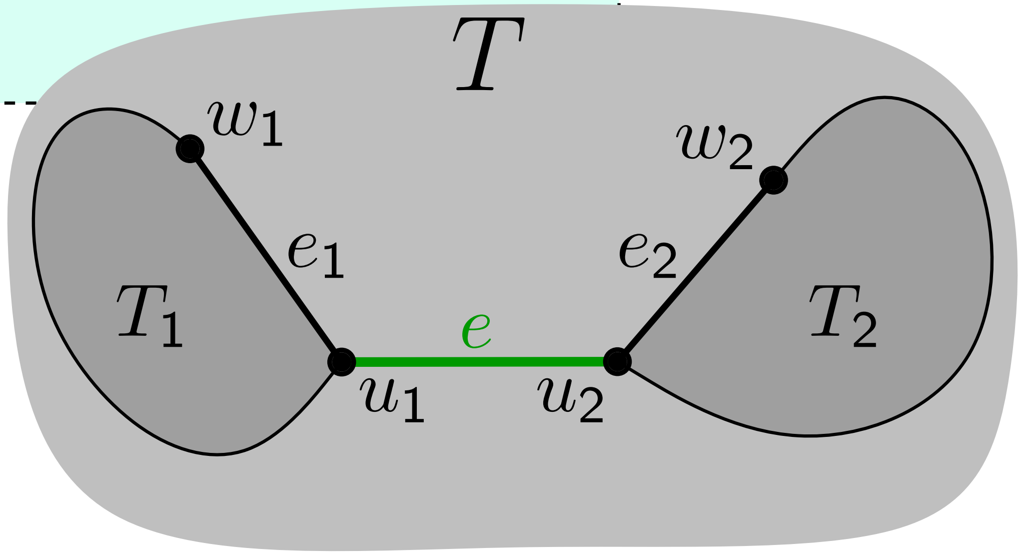
for $i \leftarrow 1$ **to** 2 **do**

$T_i \leftarrow$ component of $T - e$ that contains u_i

if $|T_i| = 1$ **then** $\Pi_i \leftarrow \emptyset$; $w_i \leftarrow u_i$

else

pick an edge $e_i = u_i w_i$ incident to u_i in T_i



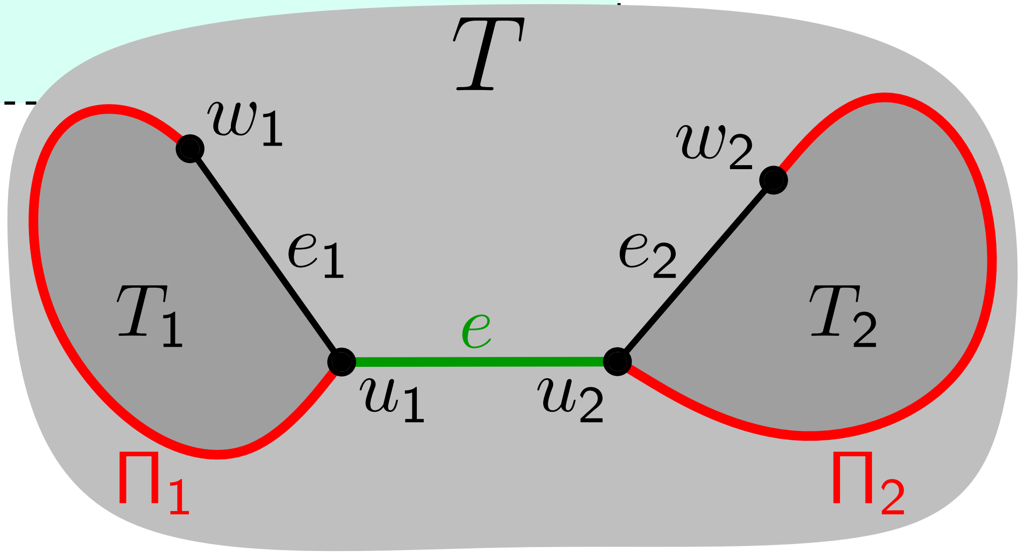
The T^3-Algorithm

[Sekanina'60, AndreaeBandelt'95]

$\textsc{CycleInCube}(T, e = u_1u_2)$

for $i \leftarrow 1$ **to** 2 **do**

- $T_i \leftarrow$ component of $T - e$ that contains u_i
- **if** $|T_i| = 1$ **then** $\Pi_i \leftarrow \emptyset$; $w_i \leftarrow u_i$
- **else**
  - pick an edge $e_i = u_iw_i$ incident to u_i in T_i
  - **if** $|T_i| = 2$ **then** $\Pi_i \leftarrow e_i$
  - **else** $\Pi_i \leftarrow \textsc{CycleInCube}(T_i, e_i) - e_i$



The T^3-Algorithm

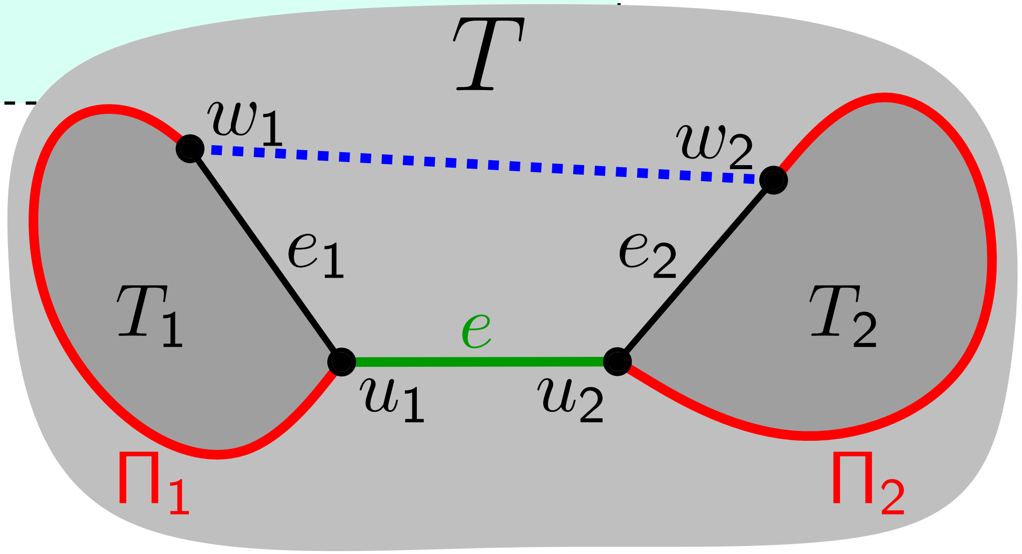
[Sekanina'60, AndreaeBandelt'95]

CycleInCube(T, $e = u_1u_2$)

for $i \leftarrow 1$ **to** 2 **do**

- $T_i \leftarrow$ component of $T - e$ that contains u_i
- **if** $|T_i| = 1$ **then** $\Pi_i \leftarrow \emptyset$; $w_i \leftarrow u_i$
- **else**
  - pick an edge $e_i = u_iw_i$ incident to u_i in T_i
  - **if** $|T_i| = 2$ **then** $\Pi_i \leftarrow e_i$
  - **else** $\Pi_i \leftarrow$ CycleInCube$(T_i, e_i) - e_i$

return $\Pi_1 + e + \Pi_2 + w_1w_2$



The T^3-Algorithm

[Sekanina'60, AndreaeBandelt'95]

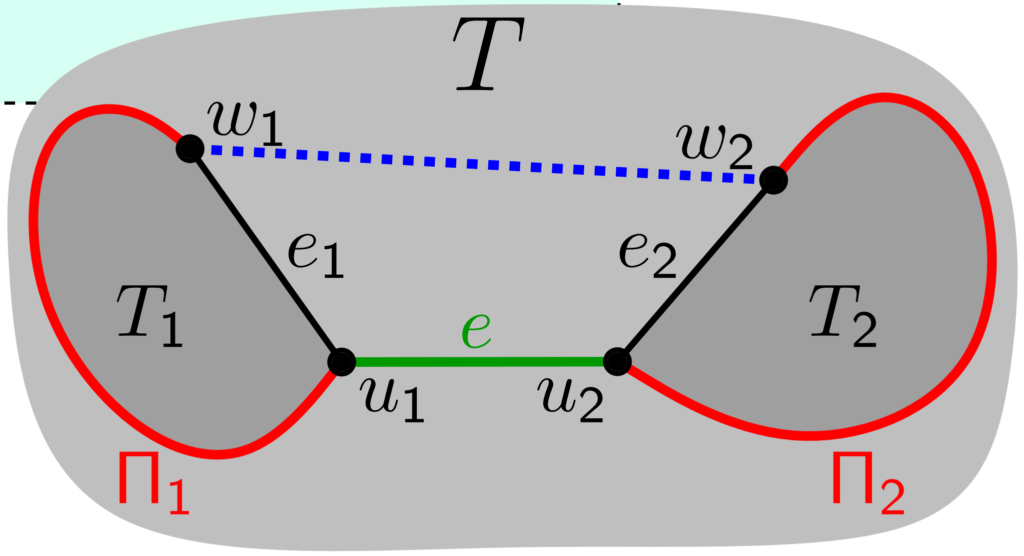
$\textsc{CycleInCube}(T, e = u_1u_2)$

for $i \leftarrow 1$ **to** 2 **do**

- $T_i \leftarrow$ component of $T - e$ that contains u_i
- **if** $|T_i| = 1$ **then** $\Pi_i \leftarrow \emptyset$; $w_i \leftarrow u_i$
- **else**
  - pick an edge $e_i = u_iw_i$ incident to u_i in T_i
  - **if** $|T_i| = 2$ **then** $\Pi_i \leftarrow e_i$
  - **else** $\Pi_i \leftarrow \textsc{CycleInCube}(T_i, e_i) - e_i$

return $\Pi_1 + e + \Pi_2 + w_1w_2$

3-*shortcut*



The T^3-Algorithm

[Sekanina'60, AndreaeBandelt'95]

CycleInCube(T, $e = u_1 u_2$)

for $i \leftarrow 1$ **to** 2 **do**

  $T_i \leftarrow$ component of $T - e$ that contains u_i

  if $|T_i| = 1$ **then** $\Pi_i \leftarrow \emptyset$; $w_i \leftarrow u_i$

  else

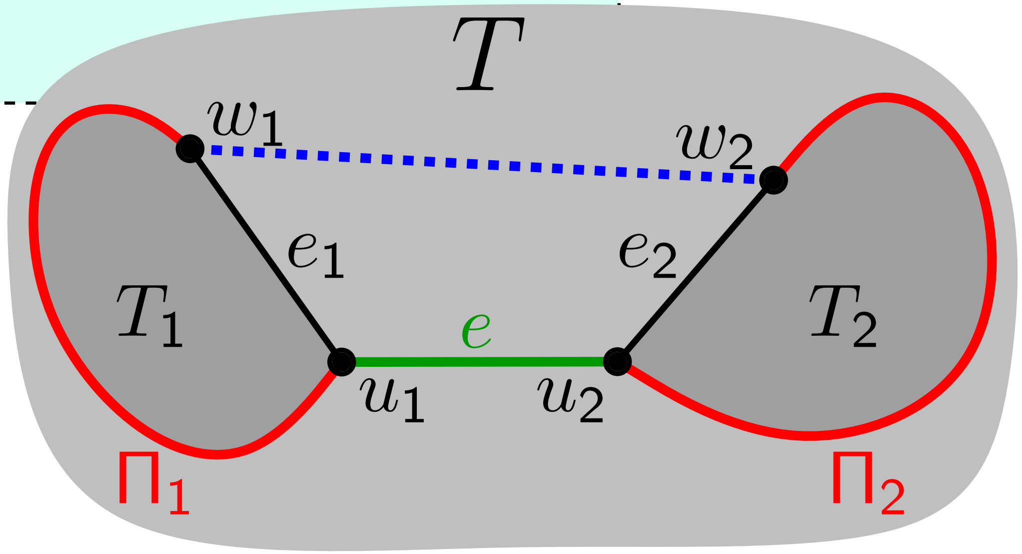
    pick an edge $e_i = u_i w_i$ incident to u_i in T_i

    if $|T_i| = 2$ **then** $\Pi_i \leftarrow e_i$

    else $\Pi_i \leftarrow$ CycleInCube(T_i, e_i) $- e_i$

return $\Pi_1 + e + \Pi_2 + w_1 w_2$

3-*shortcut*



The T^3-Algorithm

[Sekanina'60, AndreaeBandelt'95]

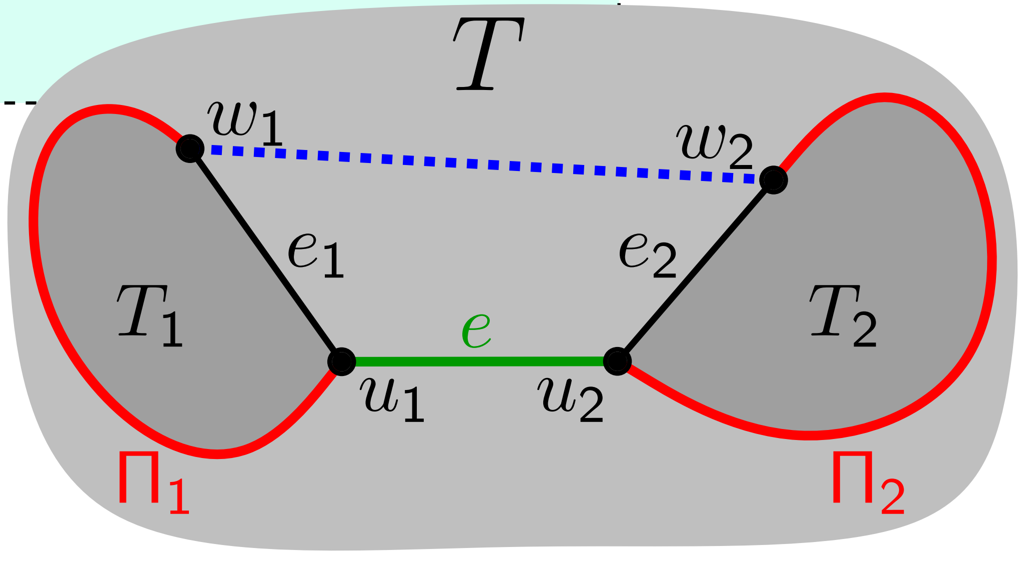
$\textsc{CycleInCube}(T, e = u_1u_2)$

for $i \leftarrow 1$ **to** 2 **do**

- $T_i \leftarrow$ component of $T - e$ that contains u_i
- **if** $|T_i| = 1$ **then** $\Pi_i \leftarrow \emptyset$; $w_i \leftarrow u_i$
- **else**
  - pick an edge $e_i = u_iw_i$ incident to u_i in T_i
  - **if** $|T_i| = 2$ **then** $\Pi_i \leftarrow e_i$
  - **else** $\Pi_i \leftarrow \textsc{CycleInCube}(T_i, e_i) - e_i$

return $\Pi_1 + e + \Pi_2 + w_1w_2$

3-*shortcut*



The T^3-Algorithm

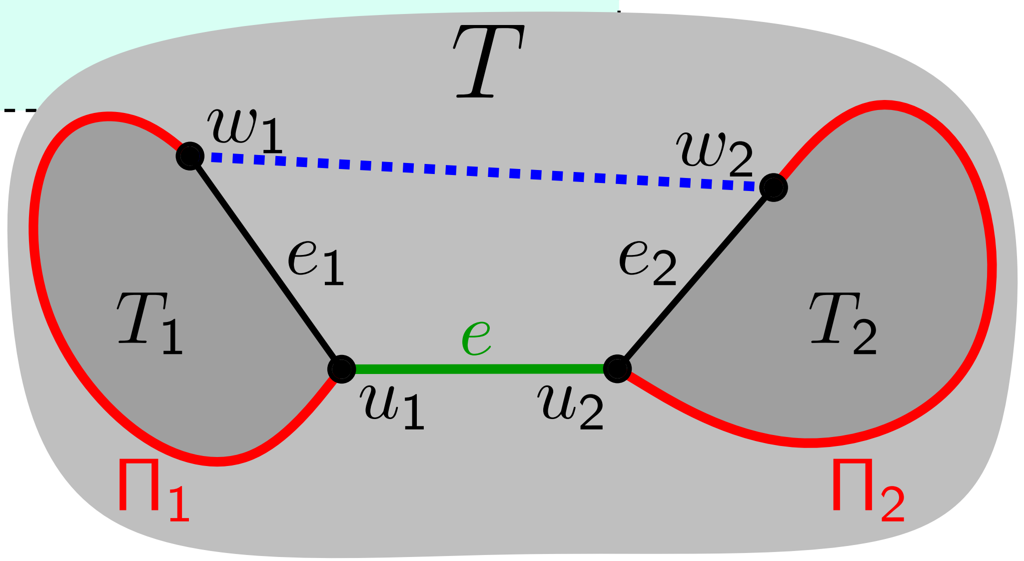
[Sekanina'60, AndreaeBandelt'95]

$\textsc{CycleInCube}(T, e = u_1 u_2)$

for $i \leftarrow 1$ **to** 2 **do**
- $T_i \leftarrow$ component of $T - e$ that contains u_i
- **if** $|T_i| = 1$ **then** $\Pi_i \leftarrow \emptyset$; $w_i \leftarrow u_i$
- **else**
  - pick an edge $e_i = u_i w_i$ incident to u_i in T_i
  - **if** $|T_i| = 2$ **then** $\Pi_i \leftarrow e_i$
  - **else** $\Pi_i \leftarrow \textsc{CycleInCube}(T_i, e_i) - e_i$

return $\Pi_1 + e + \Pi_2 + w_1 w_2$

(2- or) 3-*shortcut*



The T^3-Algorithm [Sekanina'60, AndreaeBandelt'95]

$\textsc{CycleInCube}(T, e = u_1 u_2)$

for $i \leftarrow 1$ **to** 2 **do**

  $T_i \leftarrow$ component of $T - e$ that contains u_i

  if $|T_i| = 1$ **then** $\Pi_i \leftarrow \emptyset$; $w_i \leftarrow u_i$

  else

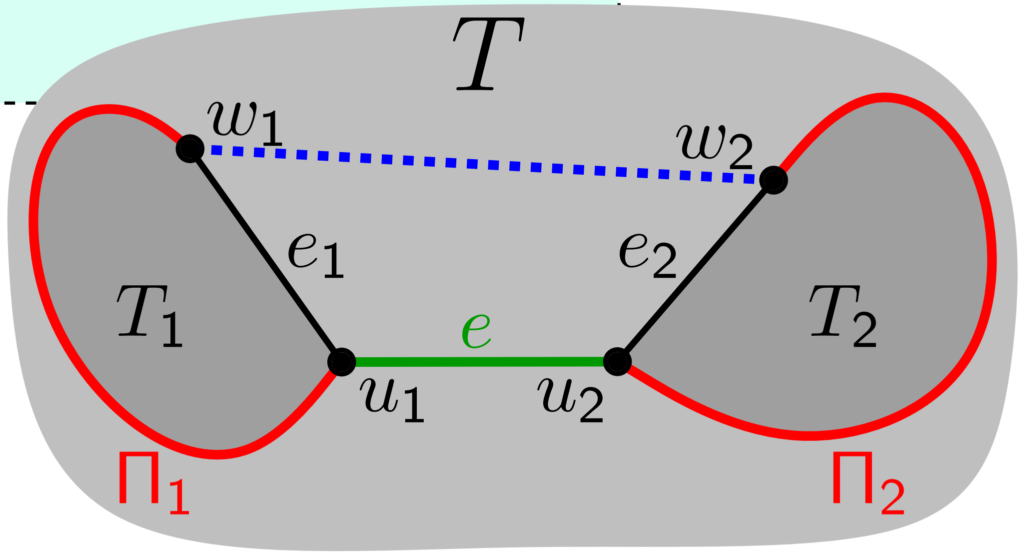
    pick an edge $e_i = u_i w_i$ incident to u_i in T_i

    if $|T_i| = 2$ **then** $\Pi_i \leftarrow e_i$

    else $\Pi_i \leftarrow \textsc{CycleInCube}(T_i, e_i) - e_i$

return $\Pi_1 + e + \Pi_2 + w_1 w_2$

(2- or) *3-shortcut* uses edges e, e_1, and e_2

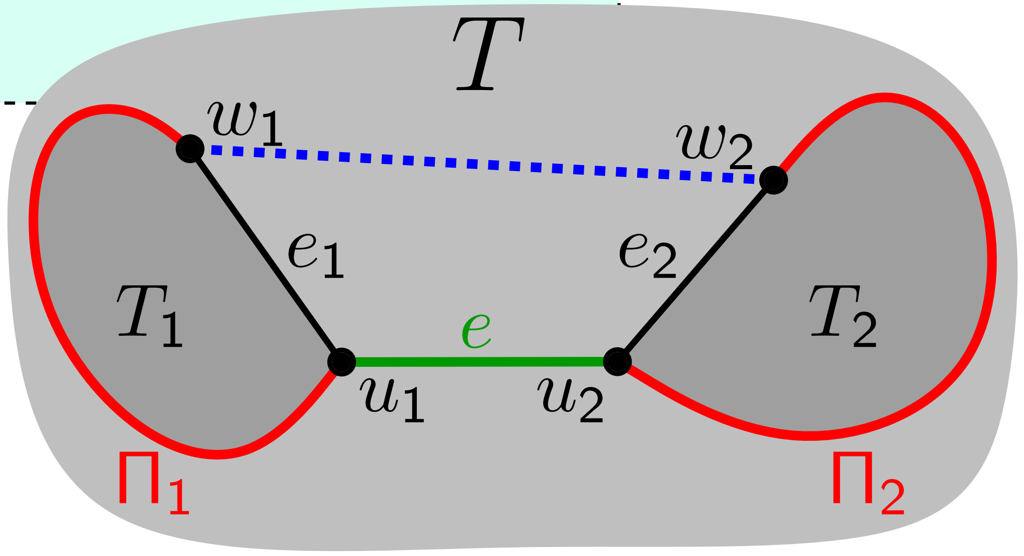


The T^3-Algorithm

[Sekanina'60, AndreaeBandelt'95]

```
CycleInCube(T, e = u_1 u_2)
  for i ← 1 to 2 do
    T_i ← component of T − e that contains u_i
    if |T_i| = 1 then Π_i ← ∅; w_i ← u_i
    else
      pick an edge e_i = u_i w_i incident to u_i in T_i
      if |T_i| = 2 then Π_i ← e_i
      else Π_i ← CycleInCube(T_i, e_i) − e_i
  return Π_1 + e + Π_2 + w_1 w_2
```

(2- or) 3-*shortcut*
uses edges e, e_1, and e_2



Observation.
Every edge is used at most **...?**

The T³-Algorithm

[Sekanina'60, AndreaeBandelt'95]

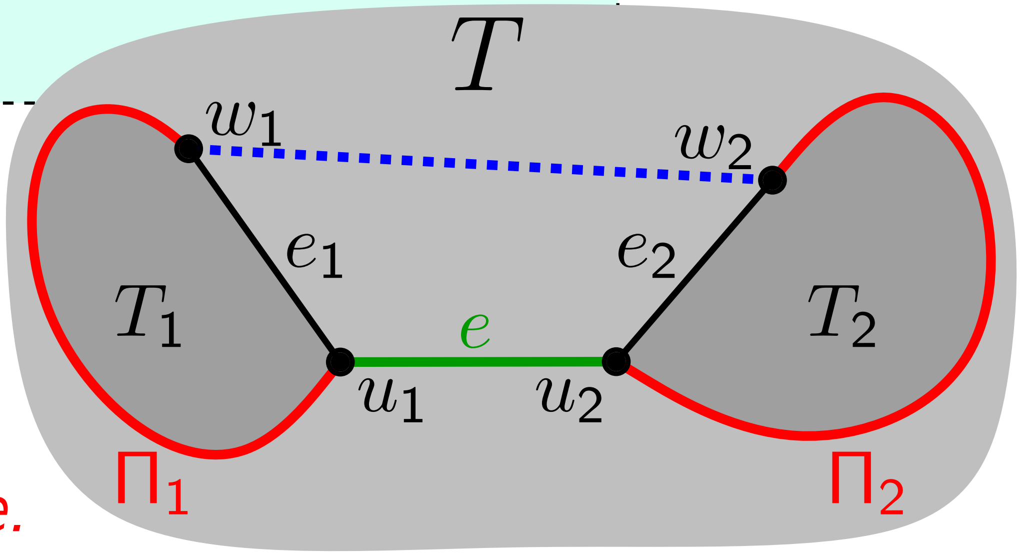
CycleInCube(T, $e = u_1u_2$)

for $i \leftarrow 1$ **to** 2 **do**

- $T_i \leftarrow$ component of $T - e$ that contains u_i
- **if** $|T_i| = 1$ **then** $\Pi_i \leftarrow \emptyset$; $w_i \leftarrow u_i$
- **else**
  - pick an edge $e_i = u_iw_i$ incident to u_i in T_i
  - **if** $|T_i| = 2$ **then** $\Pi_i \leftarrow e_i$
  - **else** $\Pi_i \leftarrow$ CycleInCube$(T_i, e_i) - e_i$

return $\Pi_1 + e + \Pi_2 + w_1w_2$

(2- or) 3-*shortcut* *uses* edges e, e_1, and e_2



Observation.
Every edge is used at most *twice.*

Result #1

Observation. Every edge is used at most twice.

Result #1

Observation. Every edge is used at most twice.

Lemma. Let e be a 3-shortcut using a, b, c. Let $\alpha \geq 1$. Then $|e|^\alpha \leq 3^{\alpha-1}\big(|a|^\alpha + |b|^\alpha + |c|^\alpha\big)$.

Result #1

Observation. Every edge is used at most twice.

Lemma. Let e be a 3-shortcut using a, b, c. Let $\alpha \geq 1$. Then $|e|^\alpha \leq 3^{\alpha-1}\big(|a|^\alpha + |b|^\alpha + |c|^\alpha\big)$.

Corollary. For $\alpha \geq 2$, the T^3-algorithm yields a $(2 \cdot 3^{\alpha-1})$-approximation for TSP$(\,\cdot\,, \alpha)$.

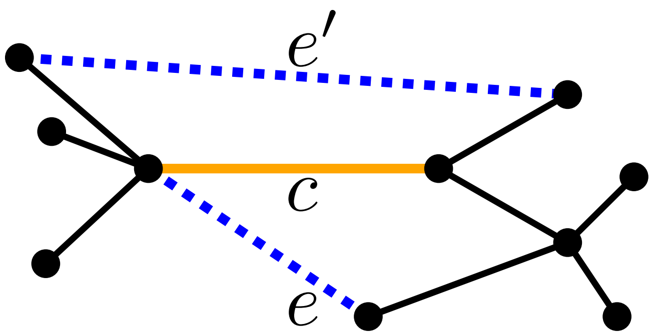
Result #1

Observation. Every edge is used at most twice.

Lemma. Let e be a 3-shortcut using a, b, c. Let $\alpha \geq 1$. Then $|e|^\alpha \leq 3^{\alpha-1}\big(|a|^\alpha + |b|^\alpha + |c|^\alpha\big)$.

Corollary. For $\alpha \geq 2$, the T^3-algorithm yields a $(2 \cdot 3^{\alpha-1})$-approximation for TSP$(\,\cdot\,, \alpha)$.

Proof. Every edge c of the MST (w.r.t. $|\cdot|^\alpha$) contributes at most twice at most $3^{\alpha-1}|c|^\alpha$ to the T^3-tour.



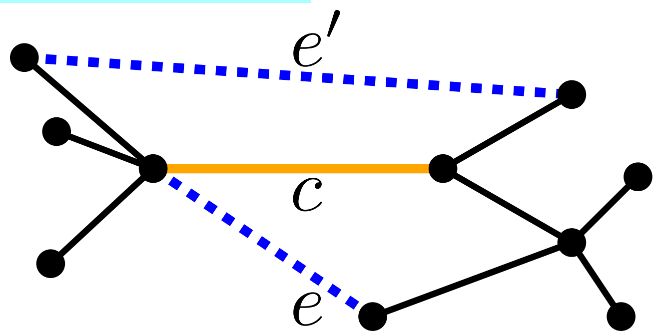
Result $\#1$

Observation. Every edge is used at most twice.

Lemma. Let e be a 3-shortcut using a, b, c. Let $\alpha \geq 1$. Then $|e|^\alpha \leq 3^{\alpha-1}\big(|a|^\alpha + |b|^\alpha + |c|^\alpha\big)$.

Corollary. For $\alpha \geq 2$, the T^3-algorithm yields a $(2 \cdot 3^{\alpha-1})$-approximation for TSP$(\,\cdot\,, \alpha)$.

Proof. Every edge c of the MST (w.r.t. $|\cdot|^\alpha$) contributes at most twice at most $3^{\alpha-1}|c|^\alpha$ to the T^3-tour.



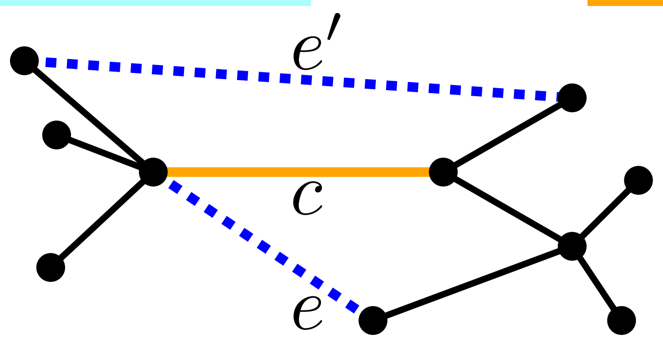
Result #1

Observation. Every edge is used at most twice.

Lemma. Let e be a 3-shortcut using a, b, c. Let $\alpha \geq 1$. Then $|e|^\alpha \leq 3^{\alpha-1}\big(|a|^\alpha + |b|^\alpha + |c|^\alpha\big)$.

Corollary. For $\alpha \geq 2$, the T[3]-algorithm yields a $(2 \cdot 3^{\alpha-1})$-approximation for TSP$(\,\cdot\,, \alpha)$.

Proof. Every edge c of the MST (w.r.t. $|\cdot|^\alpha$) contributes at most twice at most $3^{\alpha-1}|c|^\alpha$ to the T[3]-tour.



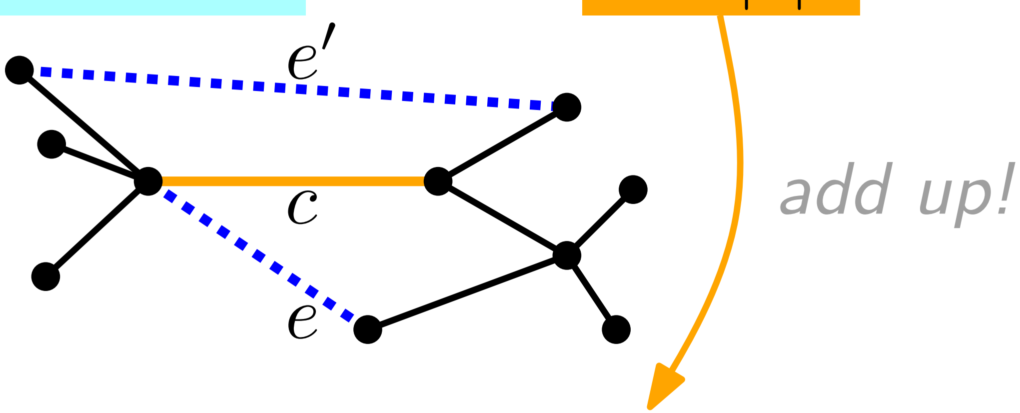
Result #1

Observation. Every edge is used at most twice.

Lemma. Let e be a 3-shortcut using a, b, c. Let $\alpha \geq 1$. Then $|e|^\alpha \leq 3^{\alpha-1}\big(|a|^\alpha + |b|^\alpha + |c|^\alpha\big)$.

Corollary. For $\alpha \geq 2$, the T^3-algorithm yields a $(2 \cdot 3^{\alpha-1})$-approximation for TSP$(\,\cdot\,, \alpha)$.

Proof. Every edge c of the MST (w.r.t. $|\cdot|^\alpha$) contributes at most twice at most $3^{\alpha-1}|c|^\alpha$ to the T^3-tour.



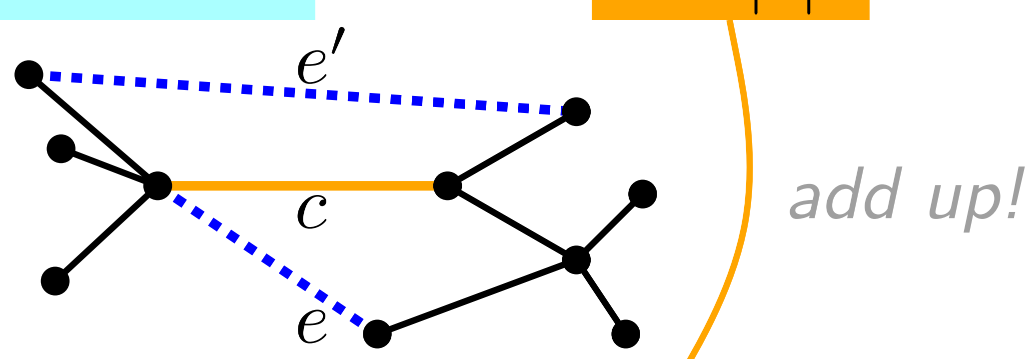
Result #1

Observation. Every edge is used at most twice.

Lemma. Let e be a 3-shortcut using a, b, c. Let $\alpha \geq 1$. Then $|e|^\alpha \leq 3^{\alpha-1}\big(|a|^\alpha + |b|^\alpha + |c|^\alpha\big)$.

Corollary. For $\alpha \geq 2$, the T^3-algorithm yields a $(2 \cdot 3^{\alpha-1})$-approximation for TSP$(\,\cdot\,,\alpha)$.

Proof. Every edge c of the MST (w.r.t. $|\cdot|^\alpha$) contributes at most twice at most $3^{\alpha-1}|c|^\alpha$ to the T^3-tour.



$3^{\alpha-1}\,|\text{MST}|^\alpha$

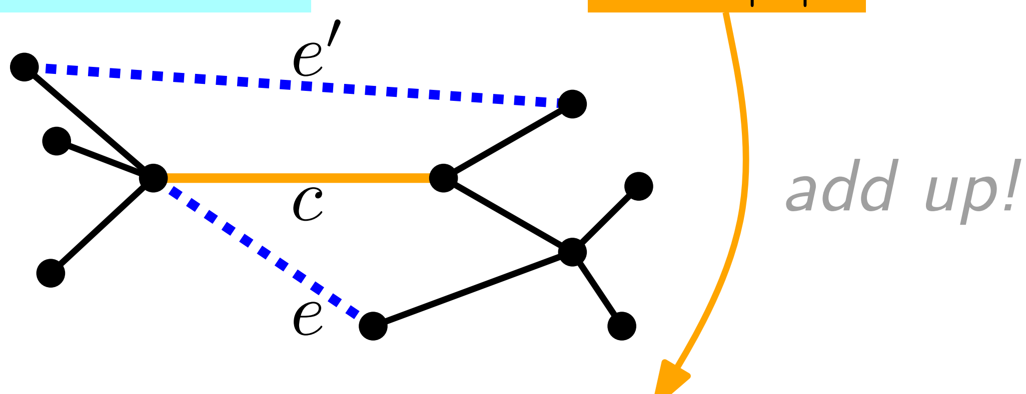
Result #1

Observation. Every edge is used at most twice.

Lemma. Let e be a 3-shortcut using a, b, c. Let $\alpha \geq 1$. Then $|e|^\alpha \leq 3^{\alpha-1}\left(|a|^\alpha + |b|^\alpha + |c|^\alpha\right)$.

Corollary. For $\alpha \geq 2$, the T^3-algorithm yields a $(2 \cdot 3^{\alpha-1})$-approximation for TSP$(\,\cdot\,, \alpha)$.

Proof. Every edge c of the MST (w.r.t. $|\cdot|^\alpha$) contributes at most twice at most $3^{\alpha-1}|c|^\alpha$ to the T^3-tour.



$2 \cdot 3^{\alpha-1}\,|\mathrm{MST}|^\alpha$

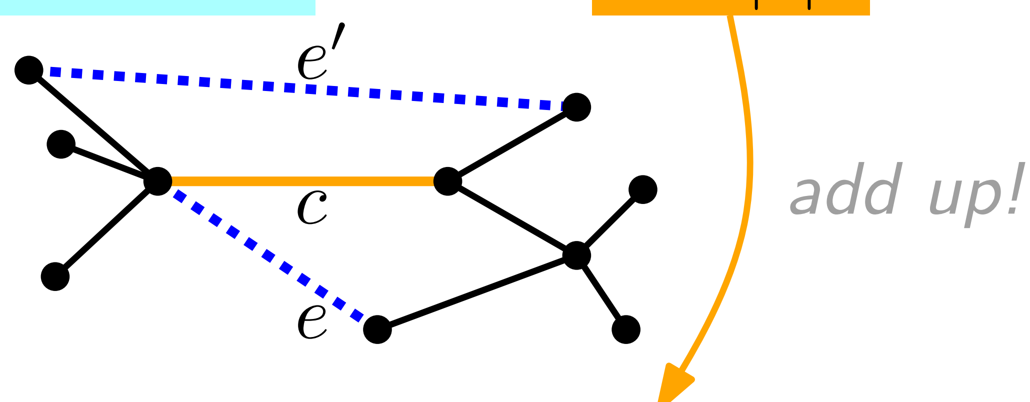
Result #1

Observation. Every edge is used at most twice.

Lemma. Let e be a 3-shortcut using a, b, c. Let $\alpha \geq 1$. Then $|e|^\alpha \leq 3^{\alpha-1}\big(|a|^\alpha + |b|^\alpha + |c|^\alpha\big)$.

Corollary. For $\alpha \geq 2$, the T^3-algorithm yields a $(2 \cdot 3^{\alpha-1})$-approximation for TSP$(\,\cdot\,, \alpha)$.

Proof. Every edge c of the MST (w.r.t. $|\cdot|^\alpha$) contributes at most twice at most $3^{\alpha-1}|c|^\alpha$ to the T^3-tour.



$\Rightarrow |\text{T}^3\text{-tour}|^\alpha \leq 2 \cdot 3^{\alpha-1}\,|\text{MST}|^\alpha$

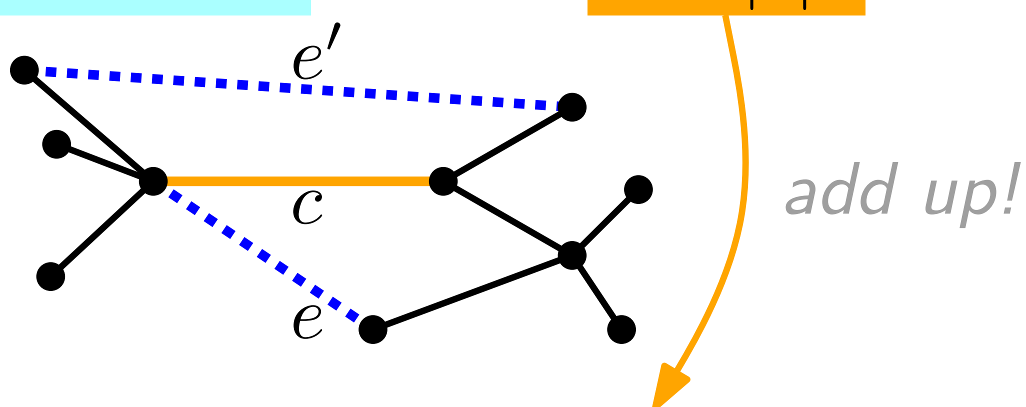
Result #1

Observation. Every edge is used at most twice.

Lemma. Let e be a 3-shortcut using a, b, c. Let $\alpha \geq 1$. Then $|e|^\alpha \leq 3^{\alpha-1}\left(|a|^\alpha + |b|^\alpha + |c|^\alpha\right)$.

Corollary. For $\alpha \geq 2$, the T^3-algorithm yields a $(2 \cdot 3^{\alpha-1})$-approximation for TSP$(\,\cdot\,, \alpha)$.

Proof. Every edge c of the MST (w.r.t. $|\cdot|^\alpha$) contributes at most twice at most $3^{\alpha-1}|c|^\alpha$ to the T^3-tour.



$\Rightarrow |\text{T}^3\text{-tour}|^\alpha \leq 2 \cdot 3^{\alpha-1}\,|\text{MST}|^\alpha \leq 2 \cdot 3^{\alpha-1} \cdot \text{OPT}$

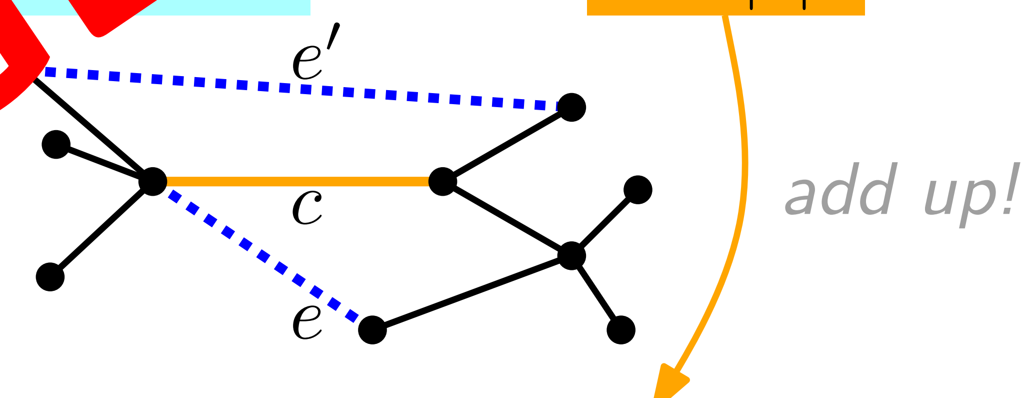
Result #1

Observation. Every edge is used at most twice.

Lemma. Let e be a 3-shortcut using a, b, c. Let $\alpha \geq 1$.
Then $|e|^\alpha \leq 3^{\alpha-1}\big(|a|^\alpha + |b|^\alpha + |c|^\alpha\big)$.

Corollary. For $\alpha \geq 2$, the T^3-algorithm yields a $(2 \cdot 3^{\alpha-1})$-approximation for TSP$(\,\cdot\,, \alpha)$.

Proof. Every edge c of the MST (w.r.t. $|\cdot|^\alpha$) contributes at most twice at most $3^{\alpha-1}|c|^\alpha$ to the T^3-tour.



$$\Rightarrow |\mathrm{T}^3\text{-tour}|^\alpha \leq 2 \cdot 3^{\alpha-1}\,|\mathrm{MST}|^\alpha \leq 2 \cdot 3^{\alpha-1} \cdot \mathrm{OPT}$$

NO GEOMETRY!

Result #2

Corollary. The T^3-algorithm yields a $(2 \cdot 3^{\alpha - 1})$-approximation for TSP$(\,\cdot\,, \alpha)$ if $\alpha \geq 2$.

Result #2

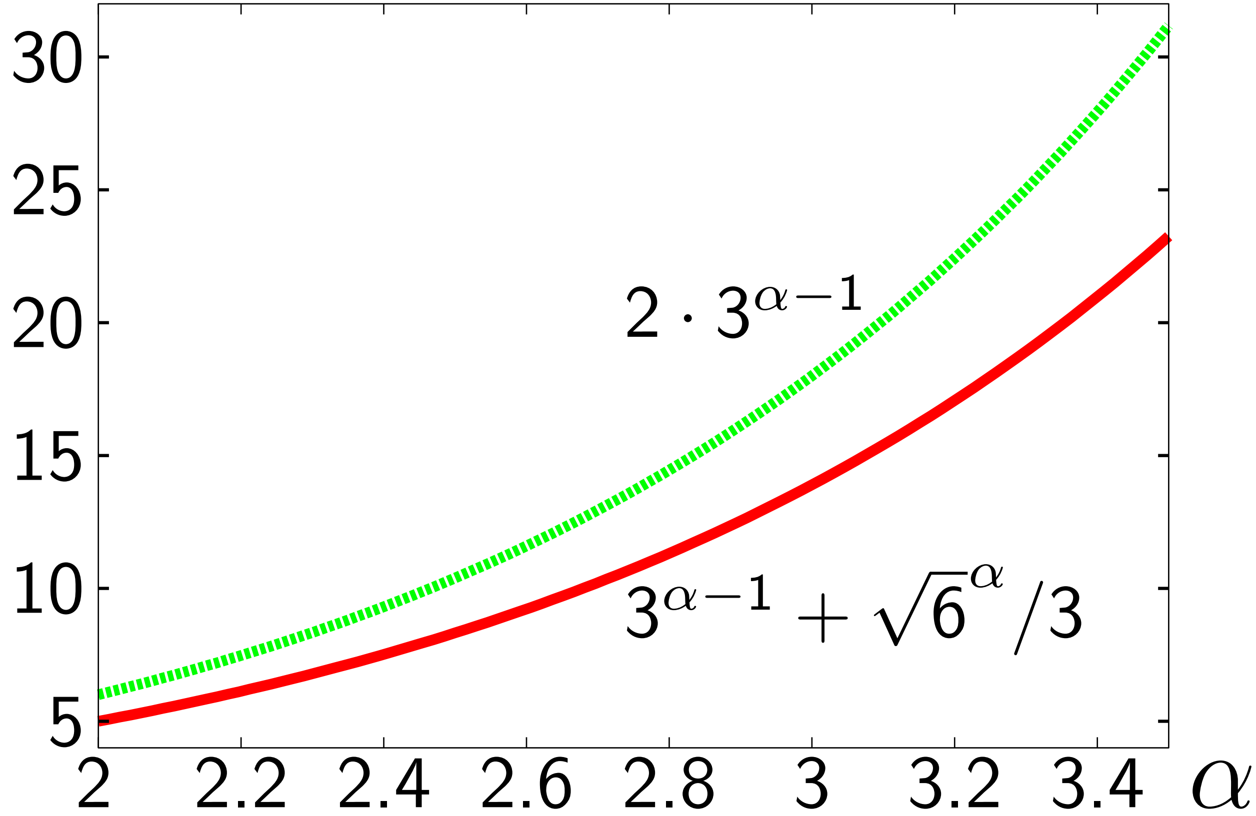
Corollary. The T^3-algorithm yields a $(2 \cdot 3^{\alpha-1})$-approximation for TSP$(\,\cdot\,, \alpha)$ if $\alpha \geq 2$.

Theorem. For $\alpha \geq 2$, the *geometric* T^3-algorithm yields a $\left(3^{\alpha-1} + \sqrt{6}^{\alpha}/3\right)$-approximation for TSP$(2, \alpha)$.

Result #2

Corollary. The T^3-algorithm yields a $(2\cdot 3^{\alpha-1})$-approximation for TSP$(\,\cdot\,,\alpha)$ if $\alpha \geq 2$.

Theorem. For $\alpha \geq 2$, the *geometric* T^3-algorithm yields a $\left(3^{\alpha-1} + \sqrt{6}^{\alpha}/3\right)$-approximation for TSP$(2, \alpha)$.



Result #2

Corollary. The T^3-algorithm yields a $(2 \cdot 3^{\alpha-1})$-approximation for TSP$(\cdot\,, \alpha)$ if $\alpha \geq 2$.

Theorem. For $\alpha \geq 2$, the *geometric* T^3-algorithm yields a $\left(3^{\alpha-1} + \sqrt{6}^{\alpha}/3\right)$-approximation for TSP$(2, \alpha)$.

```
GeometricT³(tree T, e = u₁u₂ of T)
  for i ← 1 to 2 do
    | Tᵢ ← component of T − e that contains uᵢ
    | ...
    | pick an edge eᵢ = uᵢwᵢ incident to uᵢ in Tᵢ
    | ...
    | Πᵢ ← GeometricT³(Tᵢ, eᵢ) − eᵢ
  return Π₁ + e + Π₂ + w₁w₂
```

Result #2

Corollary. The T^3-algorithm yields a $(2{\cdot}3^{\alpha-1})$-approximation for TSP$(\cdot\,,\alpha)$ if $\alpha \geq 2$.

Theorem. For $\alpha \geq 2$, the *geometric* T^3-algorithm yields a $\left(3^{\alpha-1} + \sqrt{6}^{\alpha}/3\right)$-approximation for TSP$(2, \alpha)$.

MST w.r.t. $|\cdot|^{\alpha}$

```
GeometricT³(tree T, e = u₁u₂ of T)
  for i ← 1 to 2 do
    | Tᵢ ← component of T − e that contains uᵢ
    | ...
    | pick an edge eᵢ = uᵢwᵢ incident to uᵢ in Tᵢ
    | ...
    | Πᵢ ← GeometricT³(Tᵢ, eᵢ) − eᵢ
  return Π₁ + e + Π₂ + w₁w₂
```

Result #2

Corollary. The T^3-algorithm yields a $(2 \cdot 3^{\alpha-1})$-approximation for TSP$(\,\cdot\,,\alpha)$ if $\alpha \geq 2$.

Theorem. For $\alpha \geq 2$, the *geometric* T^3-algorithm yields a $\left(3^{\alpha-1} + \sqrt{6}^{\alpha}/3\right)$-approximation for TSP$(2,\alpha)$.

MST w.r.t. $|\cdot|^\alpha$

GeometricT^3(tree T, $e = u_1u_2$ of T)
for $i \leftarrow 1$ **to** 2 **do**
- $T_i \leftarrow$ component of $T - e$ that contains u_i
- ...
- pick an edge $e_i = u_iw_i$ incident to u_i in T_i — s.t. the angle $\angle ee_i$ is min!
- ...
- $\Pi_i \leftarrow$ GeometricT$^3(T_i, e_i) - e_i$

return $\Pi_1 + e + \Pi_2 + w_1w_2$

Result #2

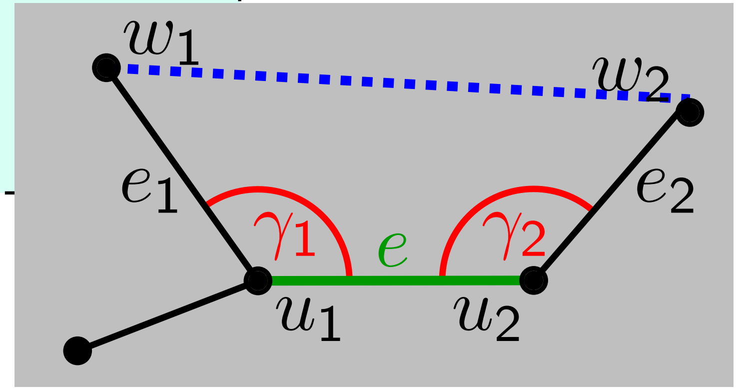
Corollary. The T^3-algorithm yields a $(2 \cdot 3^{\alpha-1})$-approximation for TSP$(\,\cdot\,, \alpha)$ if $\alpha \geq 2$.

Theorem. For $\alpha \geq 2$, the *geometric* T^3-algorithm yields a $\left(3^{\alpha-1} + \sqrt{6}^{\alpha}/3\right)$-approximation for TSP$(2, \alpha)$.

MST w.r.t. $|\cdot|^\alpha$

GeometricT^3(tree T, $e = u_1u_2$ of T)
for $i \leftarrow 1$ **to** 2 **do**
  $T_i \leftarrow$ component of $T - e$ that contains u_i
  ...
  pick an edge $e_i = u_i w_i$ incident to u_i in T_i
  ...
  $\Pi_i \leftarrow$ GeometricT$^3(T_i, e_i) - e_i$
return $\Pi_1 + e + \Pi_2 + w_1 w_2$

s.t. the angle $\angle e e_i$ is min!

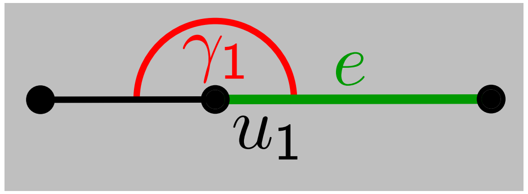


Main Insight

Why can we bound γ_1 (and γ_2) from above?

Main Insight

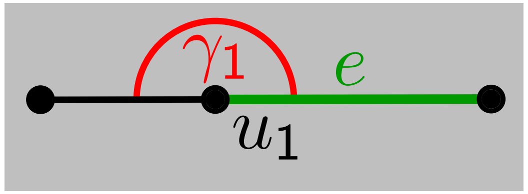
Why can we bound γ_1 (and γ_2) from above?



- If $\deg u_1 \leq 2$, then we can't.

Main Insight

Why can we bound γ_1 (and γ_2) from above?

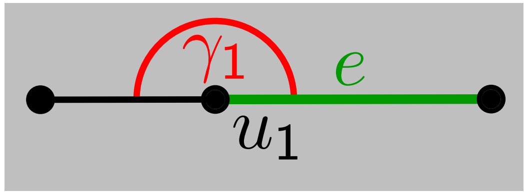


- If $\deg u_1 \leq 2$, then we can't.

  But then e is used by at most *one* 3-shortcut and maybe a 2-shortcut

Main Insight

Why can we bound γ_1 (and γ_2) from above?

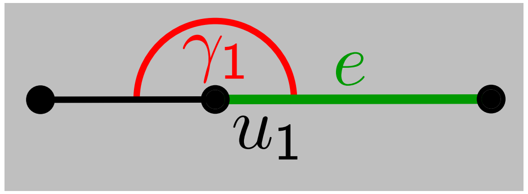


- If $\deg u_1 \leq 2$, then we can't.

  But then e is used by at most *one* 3-shortcut and maybe a 2-shortcut—but those are cheap.

Main Insight

Why can we bound γ_1 (and γ_2) from above?



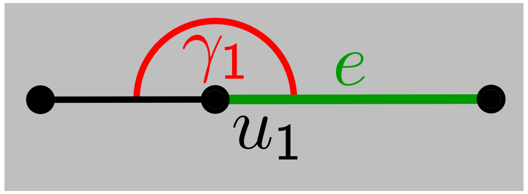
- If $\deg u_1 \leq 2$, then we can't.

  But then e is used by at most *one* 3-shortcut and maybe a 2-shortcut—but those are cheap.

- Otherwise recall that in the MST (w.r.t. $|\cdot|$ and w.r.t. $|\cdot|^\alpha$) edges incident to the same vertex make angles $\geq 60^\circ$.

Main Insight

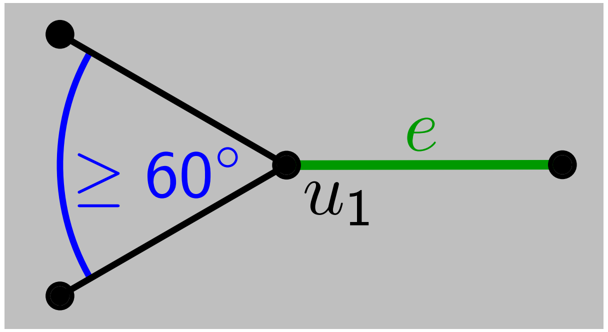
Why can we bound γ_1 (and γ_2) from above?



- If $\deg u_1 \leq 2$, then we can't.

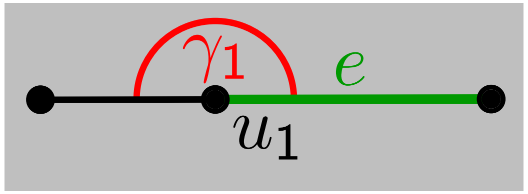
  But then e is used by at most *one* 3-shortcut and maybe a 2-shortcut—but those are cheap.

- Otherwise recall that in the MST (w.r.t. $|\cdot|$ and w.r.t. $|\cdot|^\alpha$) edges incident to the same vertex make angles $\geq 60°$.



Main Insight

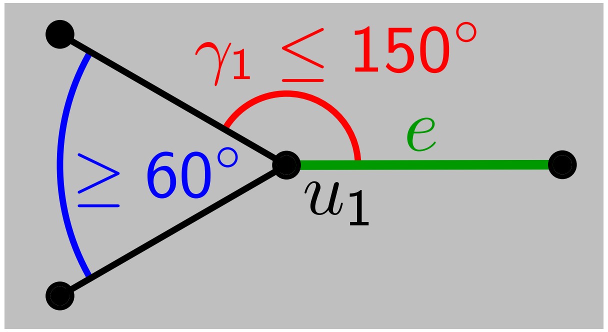
Why can we bound γ_1 (and γ_2) from above?



- If $\deg u_1 \leq 2$, then we can't.

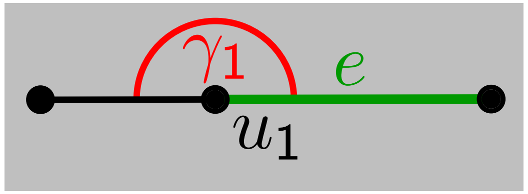
  But then e is used by at most *one* 3-shortcut and maybe a 2-shortcut—but those are cheap.

- Otherwise recall that in the MST (w.r.t. $|\cdot|$ and w.r.t. $|\cdot|^\alpha$) edges incident to the same vertex make angles $\geq 60°$.



Main Insight

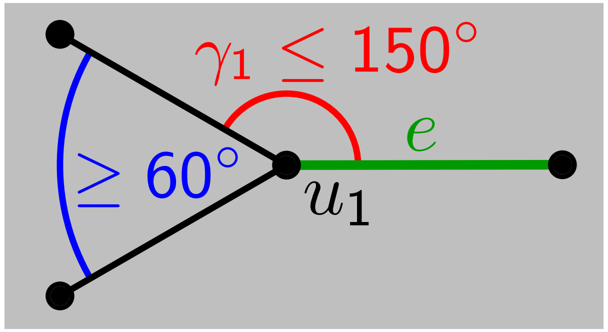
Why can we bound γ_1 (and γ_2) from above?



- If $\deg u_1 \leq 2$, then we can't.

  But then e is used by at most *one* 3-shortcut and maybe a 2-shortcut—but those are cheap.

- Otherwise recall that in the MST (w.r.t. $|\cdot|$ and w.r.t. $|\cdot|^\alpha$) edges incident to the same vertex make angles $\geq 60°$.



Thus, there is an edge e_1 incident to u_1 with $\angle e e_1 \leq 150°$.

Conclusions

- *Geometry helps!*

Conclusions

- *Geometry helps!*

  We have improved approx. of TSP(2,2) from factor 6 to 5.
  There is a lower bound of $4\frac{4}{11}$ for MST-based methods.

Conclusions

- *Geometry helps!*

  We have improved approx. of TSP(2,2) from factor 6 to 5.
  There is a lower bound of $4\frac{4}{11}$ for MST-based methods.

- *What about complexity?*

Conclusions

- *Geometry helps!*

  We have improved approx. of TSP(2,2) from factor 6 to 5.
  There is a lower bound of $4\frac{4}{11}$ for MST-based methods.

- *What about complexity?*

  TSP(d, α) is APX-hard for any $d \geq 3$ and $\alpha > 1$!
  This is in sharp contrast with Euclidean TSP.

Conclusions

- *Geometry helps!*

  We have improved approx. of TSP(2,2) from factor 6 to 5.
  There is a lower bound of $4\frac{4}{11}$ for MST-based methods.

- *What about complexity?*

  TSP(d, α) is APX-hard for any $d \geq 3$ and $\alpha > 1$!
  This is in sharp contrast with Euclidean TSP.

- *What about allowing the salesperson to revisit cities?*

Conclusions

- *Geometry helps!*

  We have improved approx. of TSP(2,2) from factor 6 to 5.
  There is a lower bound of $4\frac{4}{11}$ for MST-based methods.

- *What about complexity?*

  TSP(d, α) is APX-hard for any $d \geq 3$ and $\alpha > 1$!
  This is in sharp contrast with Euclidean TSP.

- *What about allowing the salesperson to revisit cities?*
  - Rev-TSP(d, α) is APX-hard for any $d \geq 3$ and $\alpha > 1$.
  - Rev-TSP($2, \alpha$) has a PTAS for any $\alpha \geq 2$.
  - Rev-TSP($\cdot\,, \alpha$) has a quasi-PTAS for any $0 < \alpha < 1$.

Conclusions

- *Geometry helps!*

  We have improved approx. of TSP(2,2) from factor 6 to 5. There is a lower bound of $4\frac{4}{11}$ for MST-based methods.

- *What about complexity?*

  TSP(d, α) is APX-hard for any $d \geq 3$ and $\alpha > 1$! This is in sharp contrast with Euclidean TSP.

- *What about allowing the salesperson to revisit cities?*
  - Rev-TSP(d, α) is APX-hard for any $d \geq 3$ and $\alpha > 1$.
  - Rev-TSP($2, \alpha$) has a PTAS for any $\alpha \geq 2$.
  - Rev-TSP($\cdot\,, \alpha$) has a quasi-PTAS for any $0 < \alpha < 1$.
  - What about Rev-TSP($2, \alpha$) with $1 < \alpha < 2$?

Conclusions

- *Geometry helps!*

  We have improved approx. of TSP(2,2) from factor 6 to 5.
  There is a lower bound of $4\frac{4}{11}$ for MST-based methods.

- *What about complexity?*

  TSP(d, α) is APX-hard for any $d \geq 3$ and $\alpha > 1$!
  This is in sharp contrast with Euclidean TSP.

- *What about allowing the salesperson to revisit cities?*
  - Rev-TSP(d, α) is APX-hard for any $d \geq 3$ and $\alpha > 1$.
  - Rev-TSP($2, \alpha$) has a PTAS for any $\alpha \geq 2$.
  - Rev-TSP($\cdot$, α) has a quasi-PTAS for any $0 < \alpha < 1$.
  - What about Rev-TSP($2, \alpha$) with $1 < \alpha < 2$?

    (At least as hard as TSP in weighted planar graphs!)

Conclusions

- *Geometry helps!*

  We have improved approx. of TSP(2,2) from factor 6 to 5.
  There is a lower bound of $4\frac{4}{11}$ for MST-based methods.

- *What about complexity?*

  TSP(d, α) is APX-hard for any $d \geq 3$ and $\alpha > 1$!
  This is in sharp contrast with Euclidean TSP.

- *What about allowing the salesperson to revisit cities?*
  - Rev-TSP(d, α) is APX-hard for any $d \geq 3$ and $\alpha > 1$.
  - Rev-TSP($2, \alpha$) has a PTAS for any $\alpha \geq 2$.
  - Rev-TSP($\cdot\,, \alpha$) has a quasi-PTAS for any $0 < \alpha < 1$.
  - What about Rev-TSP($2, \alpha$) with $1 < \alpha < 2$?

    (At least as hard as TSP in weighted planar graphs!)

Result #3

Theorem. TSP(d, α) and Rev-TSP(d, α) are APX-hard for any $d \geq 3$ and any $\alpha > 1$.

Result #3

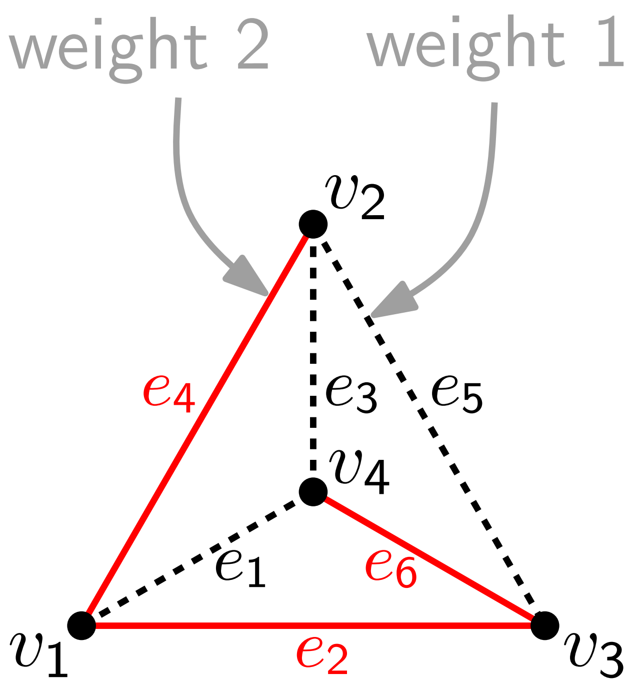
Theorem. TSP(d, α) and Rev-TSP(d, α) are APX-hard for any $d \geq 3$ and any $\alpha > 1$.

Proof sketch. Reduction from $\{1, 2\}$-TSP, which is APX-hard.

Result #3

Theorem. TSP(d, α) and Rev-TSP(d, α) are APX-hard for any $d \geq 3$ and any $\alpha > 1$.

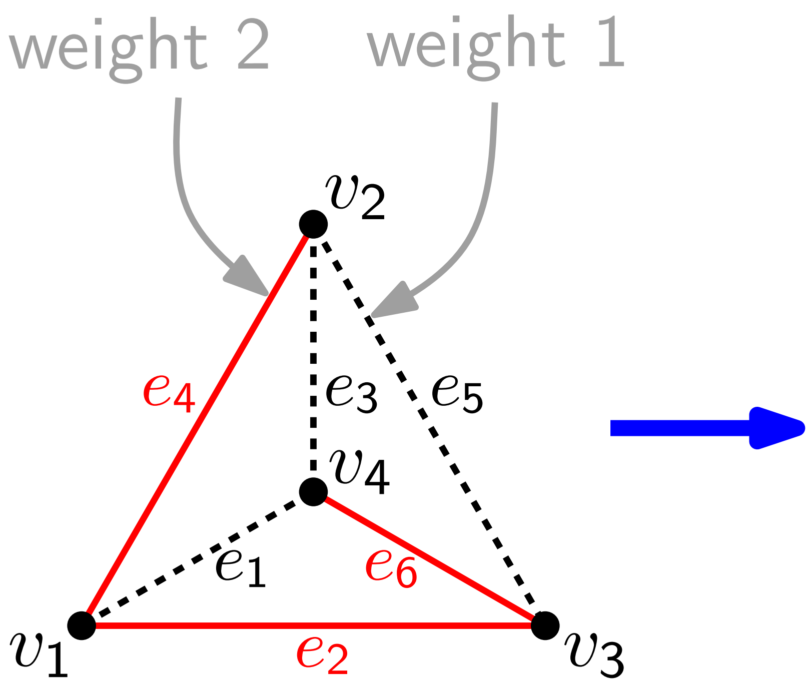
Proof sketch. Reduction from $\{1, 2\}$-TSP, which is APX-hard.



Result #3

Theorem. TSP(d, α) and Rev-TSP(d, α) are APX-hard for any $d \geq 3$ and any $\alpha > 1$.

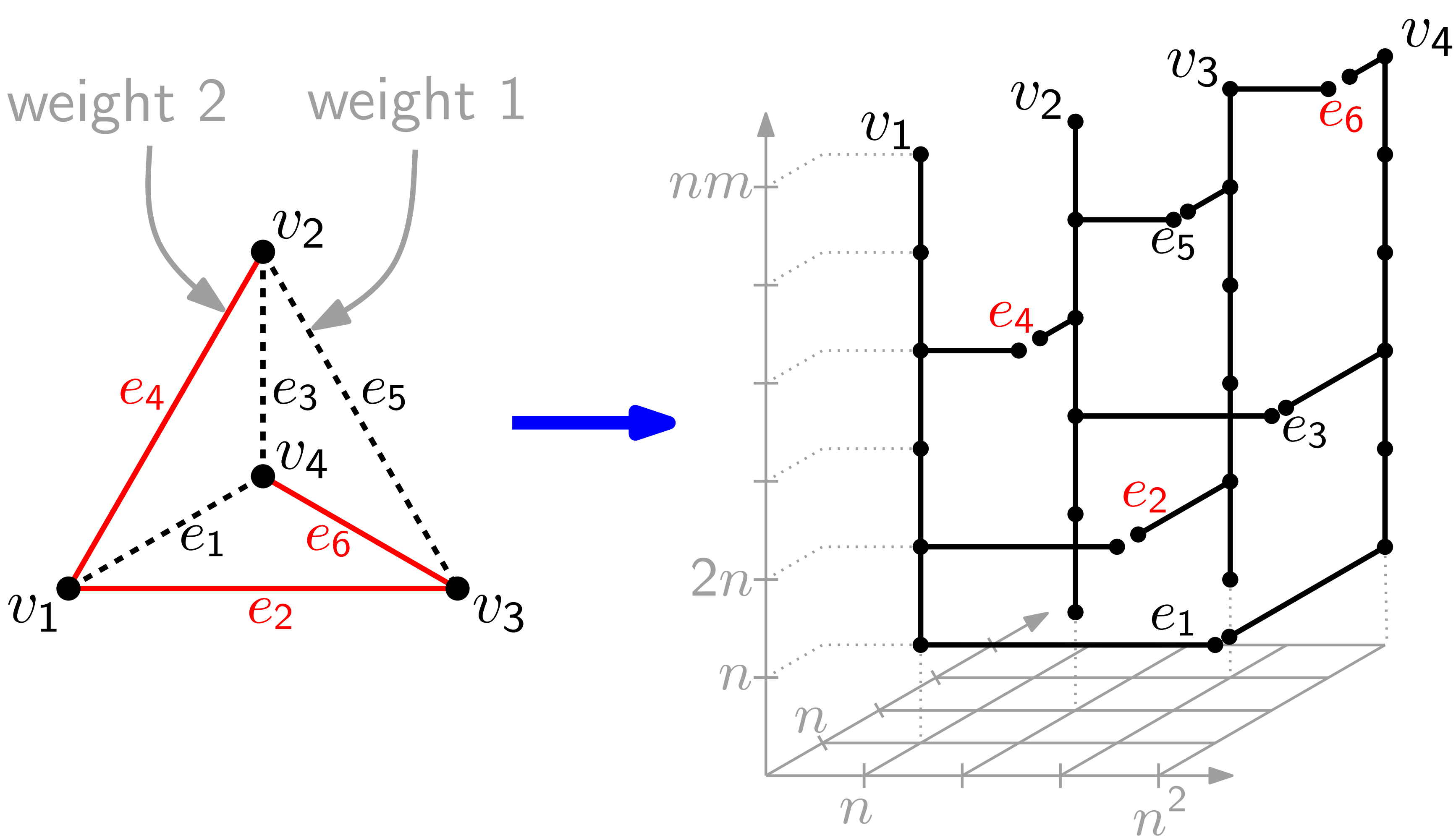
Proof sketch. Reduction from $\{1, 2\}$-TSP, which is APX-hard.



Result #3

Theorem. TSP(d, α) and Rev-TSP(d, α) are APX-hard for any $d \geq 3$ and any $\alpha > 1$.

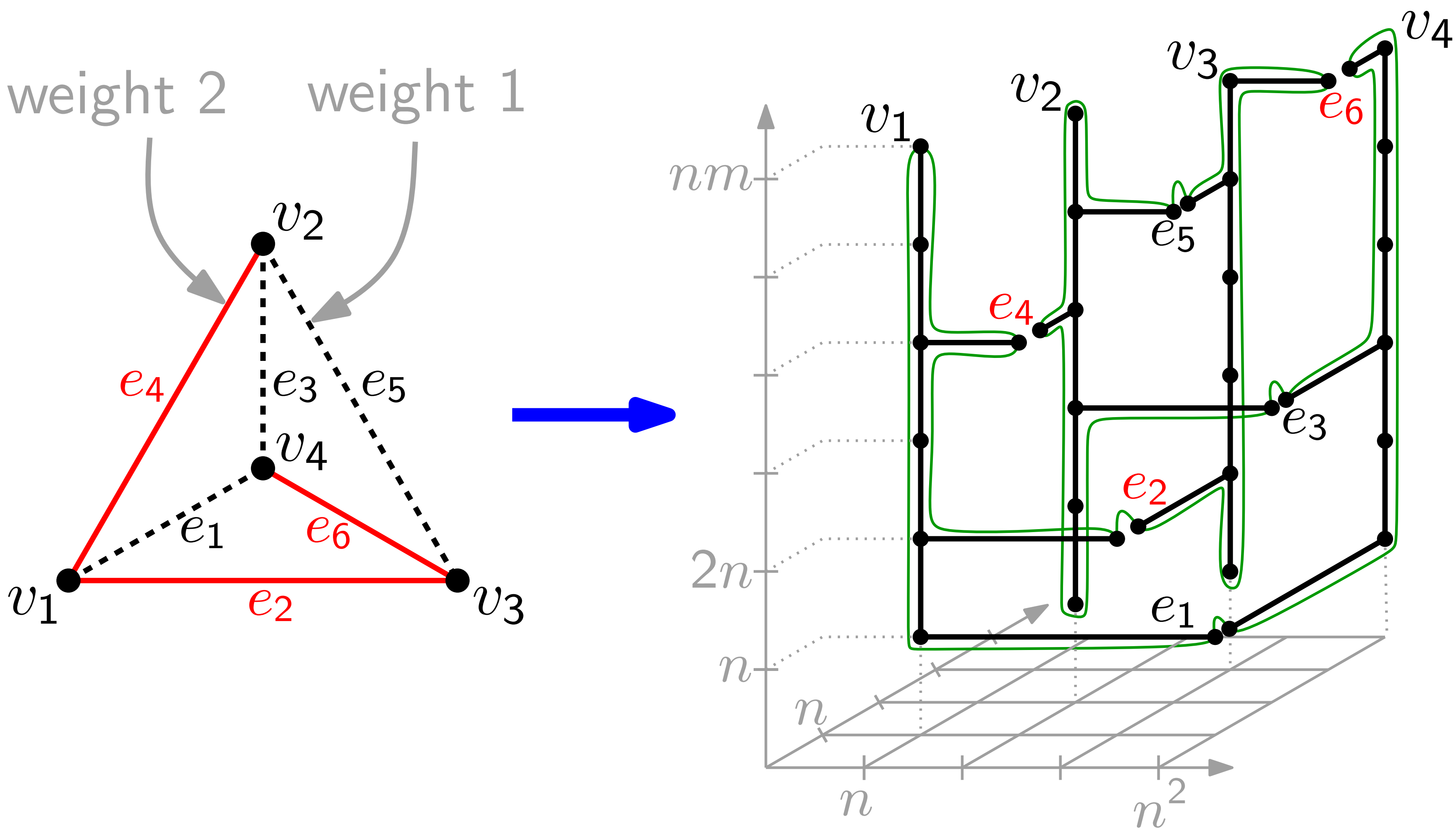
Proof sketch. Reduction from $\{1, 2\}$-TSP, which is APX-hard.



Result #3

Theorem. TSP(d, α) and Rev-TSP(d, α) are APX-hard for any $d \geq 3$ and any $\alpha > 1$.

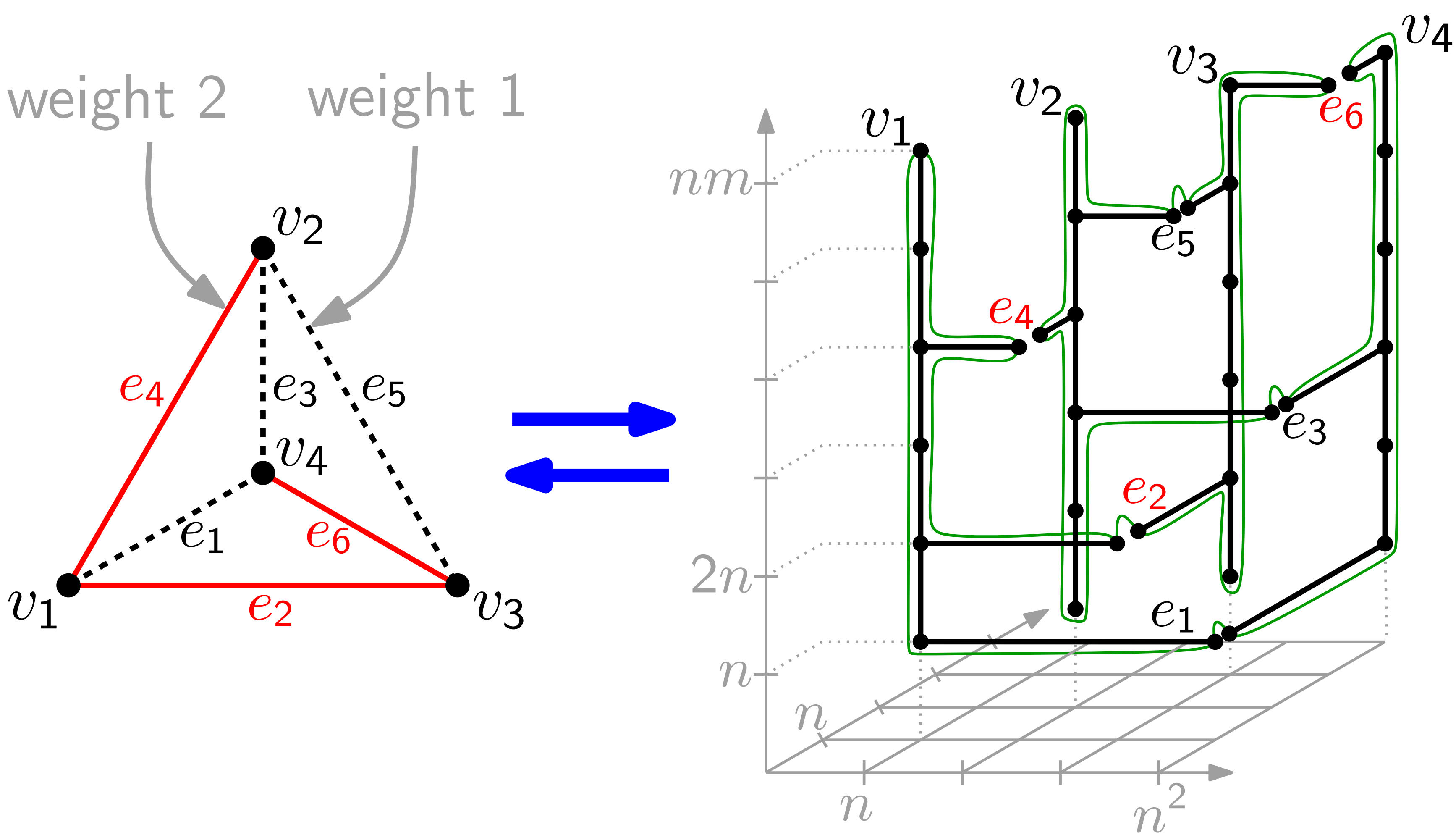
Proof sketch. Reduction from $\{1, 2\}$-TSP, which is APX-hard.



Result #3

Theorem. TSP(d,α) and Rev-TSP(d,α) are APX-hard for any $d \geq 3$ and any $\alpha > 1$.

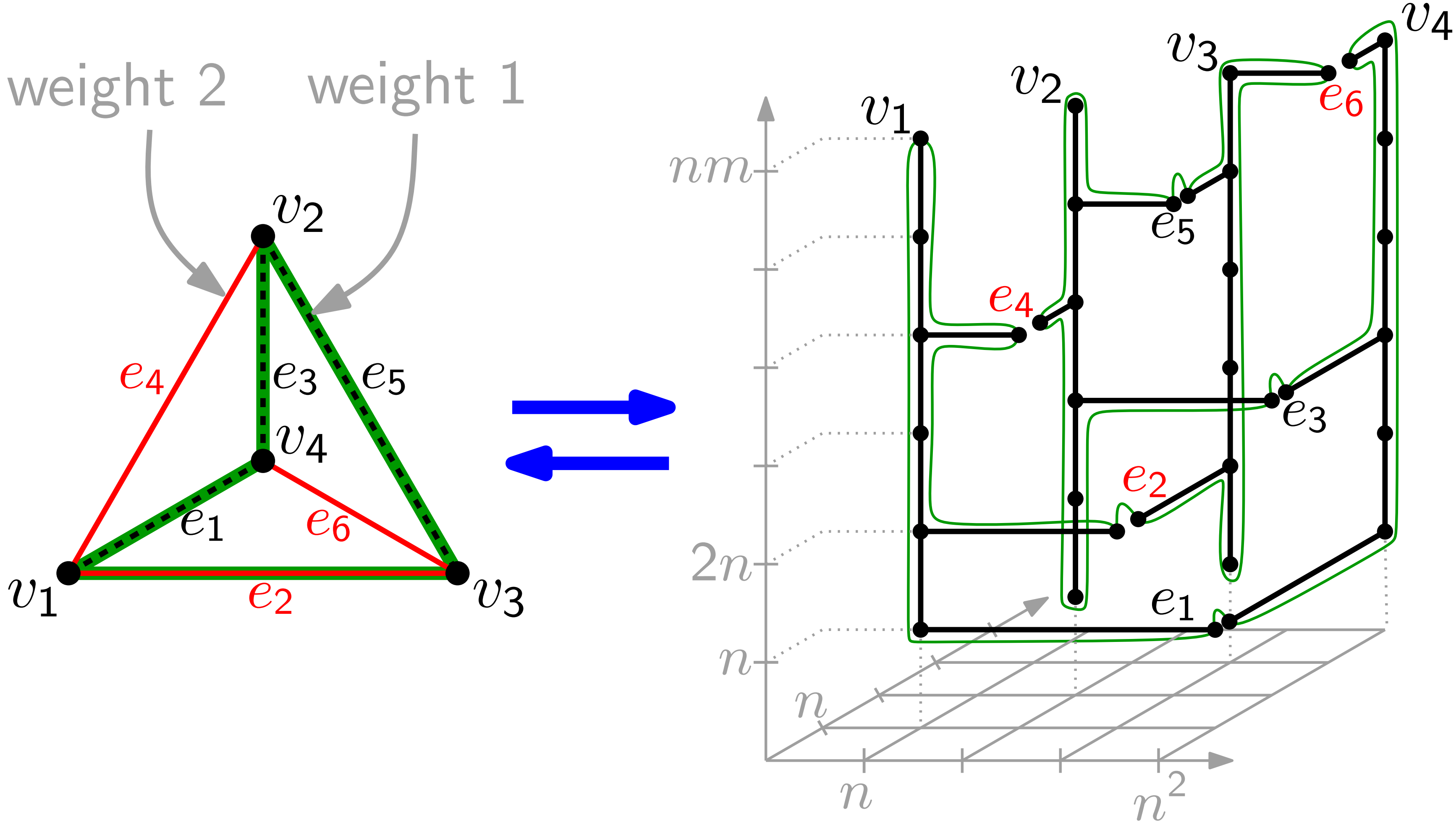
Proof sketch. Reduction from $\{1,2\}$-TSP, which is APX-hard.



Result #3

Theorem. TSP(d, α) and Rev-TSP(d, α) are APX-hard for any $d \geq 3$ and any $\alpha > 1$.

Proof sketch. Reduction from $\{1, 2\}$-TSP, which is APX-hard.



Conclusion (once more :-)

- *Geometry helps!*

  We have improved approx. of TSP(2,2) from factor 6 to 5.
  There is a lower bound of $4\frac{4}{11}$ for MST-based methods.

- *What about complexity?*

  TSP(d, α) is APX-hard for any $d \geq 3$ and $\alpha > 1$!
  This is in sharp contrast with Euclidean TSP.

- *What about allowing the salesman to revisit cities?*
  - Rev-TSP(d, α) is APX-hard for any $d \geq 3$ and $\alpha > 1$.
  - Rev-TSP($2, \alpha$) has a PTAS for any $\alpha \geq 2$.
  - Rev-TSP($\cdot$, α) has a quasi-PTAS for any $0 < \alpha < 1$.
  - What about Rev-TSP($2, \alpha$) with $1 < \alpha < 2$?

    (At least as hard as TSP in weighted planar graphs!)