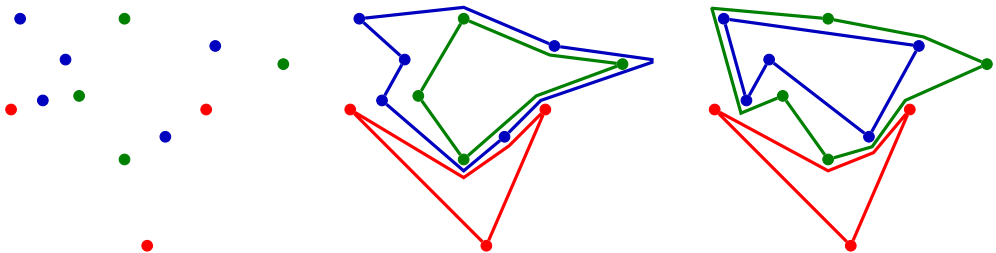
The tricolored traveling salesperson problem

Júlia Baligács

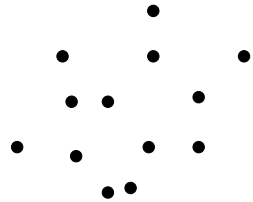


joint work with Yann Disser, Andread Emil Feldmann, Anna Zych-Pawlewicz

Euclidean traveling salesperson problem

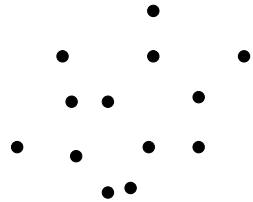
Euclidean traveling salesperson problem (ETSP)

input: points in $\mathbb{R}^2$ (called terminals)
$\rightarrow$ input size: $n :=$ # terminals



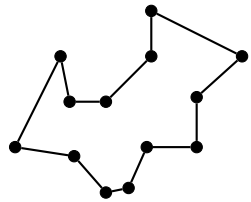
Euclidean traveling salesperson problem

Euclidean traveling salesperson problem (ETSP)	
input:	points in $\mathbb{R}^2$ (called terminals) $\rightarrow$ input size: $n :=$ # terminals
task:	find a shortest tour visiting all terminals



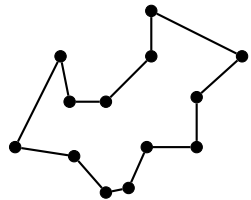
Euclidean traveling salesperson problem

Euclidean traveling salesperson problem (ETSP)	
input:	points in $\mathbb{R}^2$ (called terminals) $\rightarrow$ input size: $n :=$ # terminals
task:	find a shortest tour visiting all terminals



Euclidean traveling salesperson problem

Euclidean traveling salesperson problem (ETSP)	
input:	points in $\mathbb{R}^2$ (called terminals) $\rightarrow$ input size: $n :=$ # terminals
task:	find a shortest tour visiting all terminals

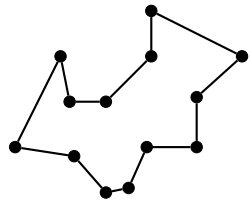


complexity of ETSP

- NP-hard [Papadimitriou (1977)]

Euclidean traveling salesperson problem

Euclidean traveling salesperson problem (ETSP)	
input:	points in $\mathbb{R}^2$ (called terminals) $\rightarrow$ input size: $n :=$ # terminals
task:	find a shortest tour visiting all terminals

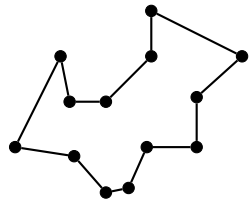


complexity of ETSP

- NP-hard [Papadimitriou (1977)]
- admits a PTAS [Arora (1998)]

Euclidean traveling salesperson problem

Euclidean traveling salesperson problem (ETSP)	
input:	points in $\mathbb{R}^2$ (called terminals) $\rightarrow$ input size: $n :=$ # terminals
task:	find a shortest tour visiting all terminals



complexity of ETSP

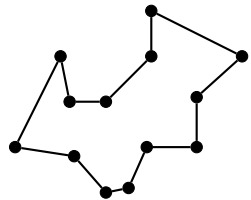
- NP-hard [Papadimitriou (1977)]
- admits a PTAS [Arora (1998)]

Definition (PTAS)

- OPT: length of shortest tour

Euclidean traveling salesperson problem

Euclidean traveling salesperson problem (ETSP)	
input:	points in $\mathbb{R}^2$ (called terminals) $\rightarrow$ input size: $n :=$ # terminals
task:	find a shortest tour visiting all terminals



complexity of ETSP

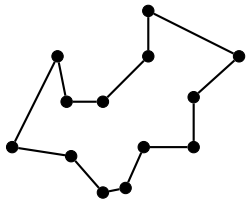
- NP-hard [Papadimitriou (1977)]
- admits a PTAS [Arora (1998)]

Definition (PTAS)

- $\textsc{Opt}$: length of shortest tour
- $(1+\varepsilon)$-approximation:
  tour of length $\leq (1+\varepsilon) \cdot \textsc{Opt}$

Euclidean traveling salesperson problem

Euclidean traveling salesperson problem (ETSP)	
input:	points in $\mathbb{R}^2$ (called terminals) $\rightarrow$ input size: $n :=$ # terminals
task:	find a shortest tour visiting all terminals



complexity of ETSP

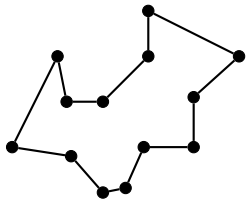
- NP-hard [Papadimitriou (1977)]
- admits a PTAS [Arora (1998)]

Definition (PTAS)

- $\textsc{Opt}$: length of shortest tour
- $(1+\varepsilon)$-approximation:
  tour of length $\leq (1+\varepsilon) \cdot \textsc{Opt}$
- polynomial time approximation scheme (PTAS): $\forall \varepsilon > 0$
  $(1+\varepsilon)$-approximation in time $n^{\mathcal{O}_\varepsilon(1)}$

Euclidean traveling salesperson problem

Euclidean traveling salesperson problem (ETSP)	
input:	points in $\mathbb{R}^2$ (called terminals) $\rightarrow$ input size: $n :=$ # terminals
task:	find a shortest tour visiting all terminals



complexity of ETSP

- NP-hard [Papadimitriou (1977)]
- admits a PTAS [Arora (1998)]
- $(1+\varepsilon)$-approx. in time $\left(\frac{1}{\varepsilon}\right)^{\mathcal{O}(1/\varepsilon^2)} \cdot n \log n$

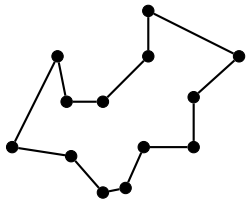
[Kisfaludi-Bak, Nederlof, Wegrzycki (2022)]

Definition (PTAS)

- $\textsc{Opt}$: length of shortest tour
- $(1+\varepsilon)$-approximation:
  tour of length $\leq (1+\varepsilon) \cdot \textsc{Opt}$
- polynomial time approximation scheme (PTAS): $\forall \varepsilon > 0$
  $(1+\varepsilon)$-approximation in time $n^{\mathcal{O}_\varepsilon(1)}$

Euclidean traveling salesperson problem

Euclidean traveling salesperson problem (ETSP)	
input:	points in $\mathbb{R}^2$ (called terminals) $\rightarrow$ input size: $n :=$ # terminals
task:	find a shortest tour visiting all terminals



complexity of ETSP

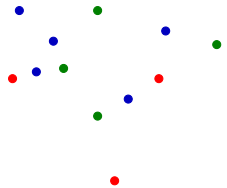
- NP-hard [Papadimitriou (1977)]
- admits a PTAS [Arora (1998)]
- $(1+\varepsilon)$-approx. in time $\left(\frac{1}{\varepsilon}\right)^{\mathcal{O}(1/\varepsilon^2)} \cdot n \log n$
  $\rightarrow$ dependency on ε optimal assuming ETH
  [Kisfaludi-Bak, Nederlof, Wegrzycki (2022)]

Definition (PTAS)

- $\textsc{Opt}$: length of shortest tour
- $(1+\varepsilon)$-approximation:
  tour of length $\leq (1+\varepsilon) \cdot \textsc{Opt}$
- polynomial time approximation scheme (PTAS): $\forall \varepsilon > 0$
  $(1+\varepsilon)$-approximation in time $n^{\mathcal{O}_\varepsilon(1)}$

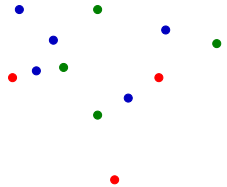
k-colored Euclidean TSP

k-colored Euclidean traveling salesperson problem (k-ETSP)
fixed: set C of k colors



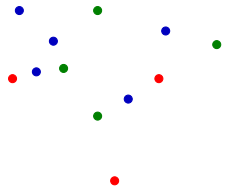
k-colored Euclidean TSP

k-colored Euclidean traveling salesperson problem (k-ETSP)	
fixed:	set C of k colors
input:	terminals (points in $\mathbb{R}^2$) colored from C

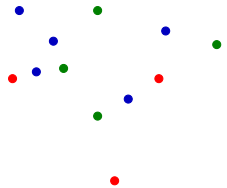


k-colored Euclidean TSP

k-colored Euclidean traveling salesperson problem (k-ETSP)	
fixed:	set C of k colors
input:	terminals (points in $\mathbb{R}^2$) colored from C
task:	find k tours $(\pi_c)_{c \in C}$ such that



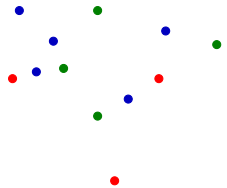
k-colored Euclidean TSP



k-colored Euclidean traveling salesperson problem (k-ETSP)

fixed:	set C of k colors
input:	terminals (points in $\mathbb{R}^2$) colored from C
task:	find k tours $(\pi_c)_{c \in C}$ such that
	▶ π_c visits all terminals colored c $(\forall c \in C)$

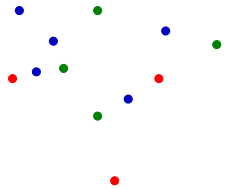
k-colored Euclidean TSP



k-colored Euclidean traveling salesperson problem (k-ETSP)

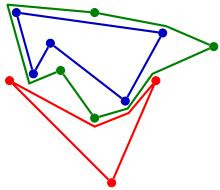
fixed:	set C of k colors
input:	terminals (points in $\mathbb{R}^2$) colored from C
task:	find k tours $(\pi_c)_{c \in C}$ such that
	▶ π_c visits all terminals colored c $(\forall c \in C)$
	▶ the tours are non-crossing, i.e., $\pi_c \cap \pi_{c'} = \emptyset \ \forall c \neq c'$

k-colored Euclidean TSP



k-colored Euclidean traveling salesperson problem (k-ETSP)	
fixed:	set C of k colors
input:	terminals (points in $\mathbb{R}^2$) colored from C
task:	find k tours $(\pi_c)_{c \in C}$ such that
	▶ π_c visits all terminals colored c $(\forall c \in C)$
	▶ the tours are non-crossing, i.e., $\pi_c \cap \pi_{c'} = \emptyset \ \forall c \neq c'$
	▶ minimize $\sum_{c \in C} l(\pi_c)$

k-colored Euclidean TSP

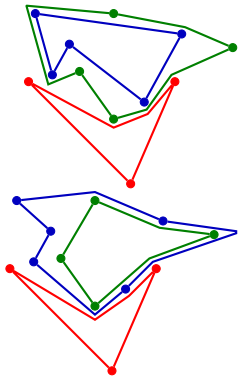


k-colored Euclidean traveling salesperson problem (k-ETSP)

fixed:	set C of k colors
input:	terminals (points in $\mathbb{R}^2$) colored from C
task:	find k tours $(\pi_c)_{c \in C}$ such that
	▶ π_c visits all terminals colored c $(\forall c \in C)$
	▶ the tours are non-crossing, i.e., $\pi_c \cap \pi_{c'} = \emptyset \ \forall c \neq c'$
	▶ minimize $\sum_{c \in C} l(\pi_c)$

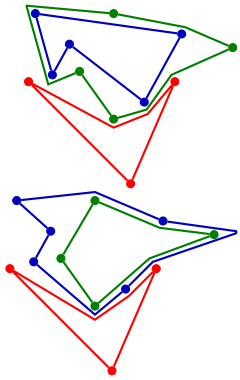
k-colored Euclidean TSP

k-colored Euclidean traveling salesperson problem (k-ETSP)	
fixed:	set C of k colors
input:	terminals (points in $\mathbb{R}^2$) colored from C
task:	find k tours $(\pi_c)_{c \in C}$ such that
	▶ π_c visits all terminals colored c $(\forall c \in C)$
	▶ the tours are non-crossing, i.e., $\pi_c \cap \pi_{c'} = \emptyset \ \forall c \neq c'$
	▶ minimize $\sum_{c \in C} l(\pi_c)$



k-colored Euclidean TSP

k-colored Euclidean traveling salesperson problem (k-ETSP)	
fixed:	set C of k colors
input:	terminals (points in $\mathbb{R}^2$) colored from C
task:	find k tours $(\pi_c)_{c \in C}$ such that
	▶ π_c visits all terminals colored c $(\forall c \in C)$
	▶ the tours are non-crossing, i.e., $\pi_c \cap \pi_{c'} = \emptyset \ \forall c \neq c'$
	▶ minimize $\sum_{c \in C} l(\pi_c)$

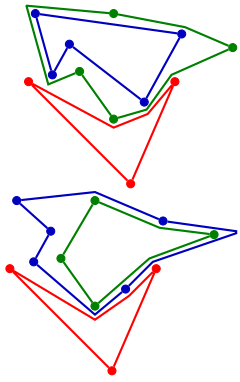


remark

- an optimum solution might not exist

k-colored Euclidean TSP

k-colored Euclidean traveling salesperson problem (k-ETSP)	
fixed:	set C of k colors
input:	terminals (points in $\mathbb{R}^2$) colored from C
task:	find k tours $(\pi_c)_{c \in C}$ such that
	▶ π_c visits all terminals colored c $(\forall c \in C)$
	▶ the tours are non-crossing, i.e., $\pi_c \cap \pi_{c'} = \emptyset \ \forall c \neq c'$
	▶ minimize $\sum_{c \in C} l(\pi_c)$

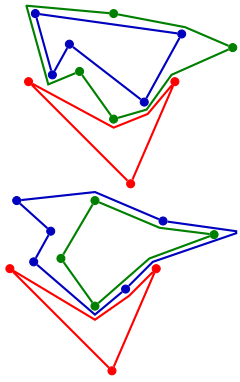


remark

- an optimum solution might not exist
- instead, set $\mathrm{OPT} := \inf \left\{ \sum_{c \in C} l(\pi_c) : (\pi_c)_{c \in C} \text{ is a solution} \right\}$

k-colored Euclidean TSP

k-colored Euclidean traveling salesperson problem (k-ETSP)	
fixed:	set C of k colors
input:	terminals (points in $\mathbb{R}^2$) colored from C
task:	find k tours $(\pi_c)_{c \in C}$ such that
	▶ π_c visits all terminals colored c ($\forall c \in C$)
	▶ the tours are non-crossing, i.e., $\pi_c \cap \pi_{c'} = \emptyset \ \forall c \neq c'$
	▶ minimize $\sum_{c \in C} l(\pi_c)$



remark

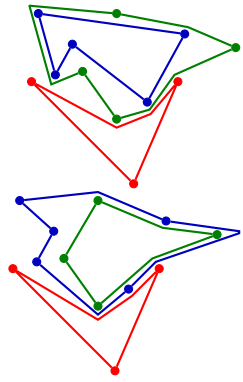
- an optimum solution might not exist
- instead, set $\mathrm{OPT} := \inf \left\{ \sum_{c \in C} l(\pi_c) : (\pi_c)_{c \in C} \text{ is a solution} \right\}$

complexity of k-ETSP

- k-ETSP is NP-hard (inherited from ETSP)

k-colored Euclidean TSP

k-colored Euclidean traveling salesperson problem (k-ETSP)	
fixed:	set C of k colors
input:	terminals (points in $\mathbb{R}^2$) colored from C
task:	find k tours $(\pi_c)_{c \in C}$ such that
	▶ π_c visits all terminals colored c ($\forall c \in C$)
	▶ the tours are non-crossing, i.e., $\pi_c \cap \pi_{c'} = \emptyset \ \forall c \neq c'$
	▶ minimize $\sum_{c \in C} l(\pi_c)$



remark

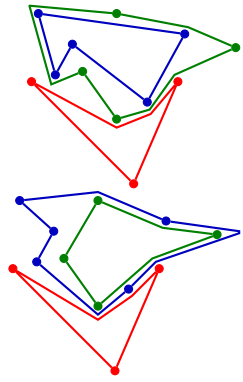
- an optimum solution might not exist
- instead, set $\mathrm{OPT} := \inf \left\{ \sum_{c \in C} l(\pi_c) : (\pi_c)_{c \in C} \text{ is a solution} \right\}$

complexity of k-ETSP

- k-ETSP is NP-hard (inherited from ETSP)
- 2-ETSP admits a PTAS [Dross, Fleszar, Wegrzycki, Zych-Pawlewicz (2023)]

k-colored Euclidean TSP

k-colored Euclidean traveling salesperson problem (k-ETSP)	
fixed:	set C of k colors
input:	terminals (points in $\mathbb{R}^2$) colored from C
task:	find k tours $(\pi_c)_{c\in C}$ such that
	▶ π_c visits all terminals colored c $(\forall c \in C)$
	▶ the tours are non-crossing, i.e., $\pi_c \cap \pi_{c'} = \emptyset \ \forall c \neq c'$
	▶ minimize $\sum_{c\in C} l(\pi_c)$



remark

- an optimum solution might not exist
- instead, set $\mathrm{OPT} := \inf\left\{\sum_{c\in C} l(\pi_c) : (\pi_c)_{c\in C} \text{ is a solution}\right\}$

complexity of k-ETSP

- k-ETSP is NP-hard (inherited from ETSP)
- 2-ETSP admits a PTAS [Dross, Fleszar, Wegrzycki, Zych-Pawlewicz (2023)]

Open problem. Does k-ETSP admit a PTAS for $k \geq 3$?

Main result

Theorem. For every $\varepsilon > 0$, there is an algorithm that computes a $\left(\frac{5}{3} + \varepsilon\right)$-approximation for 3-ETSP in time $\left(\frac{n}{\varepsilon}\right)^{O(1/\varepsilon)}$.

[B., Disser, Feldmann, Zych-Pawlewicz (2024)]

Main result

Theorem. For every $\varepsilon > 0$, there is an algorithm that computes a $\left(\frac{5}{3} + \varepsilon\right)$-approximation for 3-ETSP in time $\left(\frac{n}{\varepsilon}\right)^{O(1/\varepsilon)}$.

[B., Disser, Feldmann, Zych-Pawlewicz (2024)]

key idea: generalize Arora's algorithm (PTAS for ETSP)

Main result

Theorem. For every $\varepsilon > 0$, there is an algorithm that computes a $\left(\frac{5}{3} + \varepsilon\right)$-approximation for 3-ETSP in time $\left(\frac{n}{\varepsilon}\right)^{O(1/\varepsilon)}$.

[B., Disser, Feldmann, Zych-Pawlewicz (2024)]

key idea: generalize Arora's algorithm (PTAS for ETSP)

outline for remainder of talk

- introduction into Arora's algorithm

Main result

Theorem. For every $\varepsilon > 0$, there is an algorithm that computes a $\left(\frac{5}{3} + \varepsilon\right)$-approximation for 3-ETSP in time $\left(\frac{n}{\varepsilon}\right)^{O(1/\varepsilon)}$.

[B., Disser, Feldmann, Zych-Pawlewicz (2024)]

key idea: generalize Arora's algorithm (PTAS for ETSP)

outline for remainder of talk

- introduction into Arora's algorithm

  $\rightarrow$ Which ingredients generalize to k-ETSP?

Main result

Theorem. For every $\varepsilon > 0$, there is an algorithm that computes a $\left(\frac{5}{3} + \varepsilon\right)$-approximation for 3-ETSP in time $\left(\frac{n}{\varepsilon}\right)^{O(1/\varepsilon)}$.

[B., Disser, Feldmann, Zych-Pawlewicz (2024)]

key idea: generalize Arora's algorithm (PTAS for ETSP)

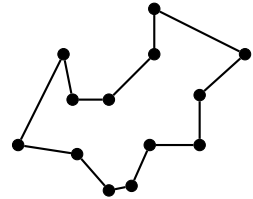
outline for remainder of talk

- introduction into Arora's algorithm

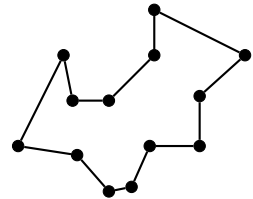
  $\rightarrow$ Which ingredients generalize to k-ETSP?
- alternative approach for $k = 3$

Arora's algorithm: outline

Theorem (Arora 1998). There is a PTAS for ETSP.



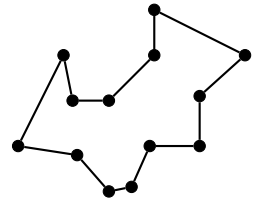
Arora's algorithm: outline



Theorem (Arora 1998). There is a PTAS for ETSP.

i.e.: $\forall \varepsilon > 0$, can compute a $(1+\varepsilon)$-approximation in polynomial time

Arora's algorithm: outline

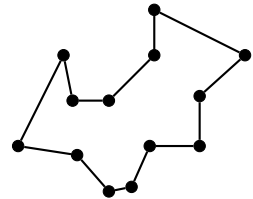


Theorem (Arora 1998). There is a PTAS for ETSP.

i.e.: $\forall \varepsilon > 0$, can compute a $(1+\varepsilon)$-approximation in polynomial time

proof outline

Arora's algorithm: outline



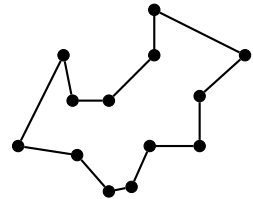
Theorem (Arora 1998). There is a PTAS for ETSP.

i.e.: $\forall \varepsilon > 0$, can compute a $(1+\varepsilon)$-approximation in polynomial time

proof outline

- restrict to the case where terminals lie in $\mathbb{Z}^2$
  with maximum distance $\mathcal{O}(n/\varepsilon)$

Arora's algorithm: outline



Theorem (Arora 1998). There is a PTAS for ETSP.

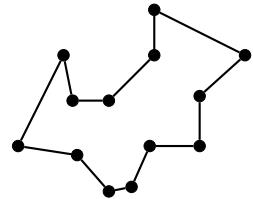
i.e.: $\forall \varepsilon > 0$, can compute a $(1+\varepsilon)$-approximation in polynomial time

proof outline

- restrict to the case where terminals lie in $\mathbb{Z}^2$
  with maximum distance $\mathcal{O}(n/\varepsilon)$

1st step: place a suitable grid in $\mathbb{R}^2$ ("dissection") and "portals" on this grid

Arora's algorithm: outline



Theorem (Arora 1998). There is a PTAS for ETSP.

i.e.: $\forall \varepsilon > 0$, can compute a $(1+\varepsilon)$-approximation in polynomial time

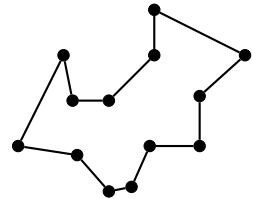
proof outline

- restrict to the case where terminals lie in $\mathbb{Z}^2$
  with maximum distance $\mathcal{O}(n/\varepsilon)$

1st step: place a suitable grid in $\mathbb{R}^2$ ("dissection") and "portals" on this grid

$\rightarrow$ "portal-respecting tour": only crosses the grid through portals

Arora's algorithm: outline



Theorem (Arora 1998). There is a PTAS for ETSP.

i.e.: $\forall \varepsilon > 0$, can compute a $(1+\varepsilon)$-approximation in polynomial time

proof outline

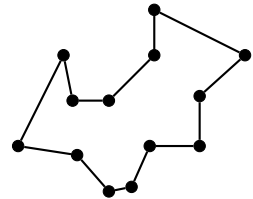
- restrict to the case where terminals lie in $\mathbb{Z}^2$
  with maximum distance $\mathcal{O}(n/\varepsilon)$

1st step: place a suitable grid in $\mathbb{R}^2$ ("dissection") and "portals" on this grid

$\rightarrow$ "portal-respecting tour": only crosses the grid through portals

2nd step: an optimal portal-respecting solution can be computed in polynomial time

Arora's algorithm: outline



Theorem (Arora 1998). There is a PTAS for ETSP.

i.e.: $\forall \varepsilon > 0$, can compute a $(1+\varepsilon)$-approximation in polynomial time

proof outline

- restrict to the case where terminals lie in $\mathbb{Z}^2$
  with maximum distance $\mathcal{O}(n/\varepsilon)$

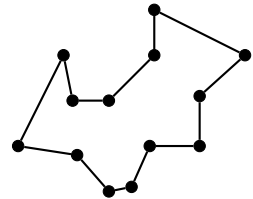
1st step: place a suitable grid in $\mathbb{R}^2$ ("dissection") and "portals" on this grid

$\rightarrow$ "portal-respecting tour": only crosses the grid through portals

2nd step: an optimal portal-respecting solution can be computed in polynomial time

3rd step: there is a portal-respecting solution of length $(1+\varepsilon) \cdot \mathrm{OPT}$

Arora's algorithm: outline



Theorem (Arora 1998). There is a PTAS for ETSP.

i.e.: $\forall \varepsilon > 0$, can compute a $(1+\varepsilon)$-approximation in polynomial time

proof outline

- restrict to the case where terminals lie in $\mathbb{Z}^2$
  with maximum distance $\mathcal{O}(n/\varepsilon)$

1st step: place a suitable grid in $\mathbb{R}^2$ ("dissection") and "portals" on this grid

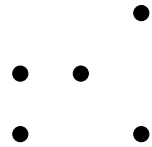
$\rightarrow$ "portal-respecting tour": only crosses the grid through portals

2nd step: an optimal portal-respecting solution can be computed in polynomial time

3rd step: there is a portal-respecting solution of length $(1+\varepsilon) \cdot \mathrm{OPT}$

Arora's algorithm: dissection and portals

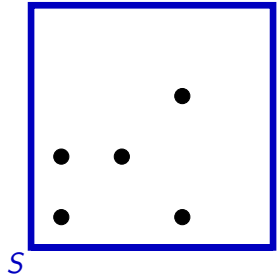
grid and portal placement



Arora's algorithm: dissection and portals

grid and portal placement

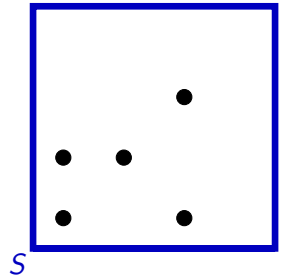
- choose a (suitable) square $S \subseteq \mathbb{R}^2$ such that



Arora's algorithm: dissection and portals

grid and portal placement

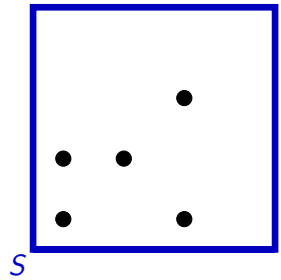
- choose a (suitable) square $S \subseteq \mathbb{R}^2$ such that
  - S contains all terminals



Arora's algorithm: dissection and portals

grid and portal placement

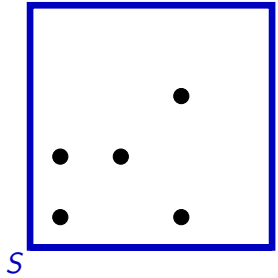
- choose a (suitable) square $S \subseteq \mathbb{R}^2$ such that
  - S contains all terminals
  - sidelength of S is $\mathcal{O}(n/\varepsilon)$ and a power of 2



Arora's algorithm: dissection and portals

grid and portal placement

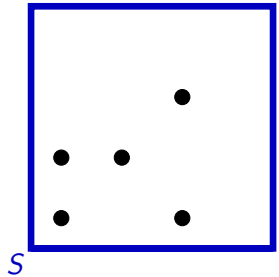
- choose a (suitable) square $S \subseteq \mathbb{R}^2$ such that
  - S contains all terminals
  - sidelength of S is $\mathcal{O}(n/\varepsilon)$ and a power of 2
  - corners of S lie in $(\mathbb{Z} + 0.5)^2$



Arora's algorithm: dissection and portals

grid and portal placement

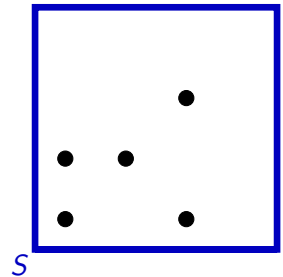
- choose a (suitable) square $S \subseteq \mathbb{R}^2$ such that
  - S contains all terminals
  - sidelength of S is $\mathcal{O}(n/\varepsilon)$ and a power of 2
  - corners of S lie in $(\mathbb{Z} + 0.5)^2$
- dissection to S: rooted tree where



Arora's algorithm: dissection and portals

grid and portal placement

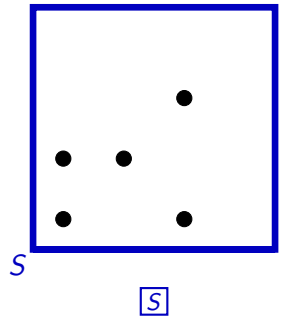
- choose a (suitable) square $S \subseteq \mathbb{R}^2$ such that
  - S contains all terminals
  - sidelength of S is $\mathcal{O}(n/\varepsilon)$ and a power of 2
  - corners of S lie in $(\mathbb{Z} + 0.5)^2$
- dissection to S: rooted tree where
  - each node is a square in $\mathbb{R}^2$ (called cells)



Arora's algorithm: dissection and portals

grid and portal placement

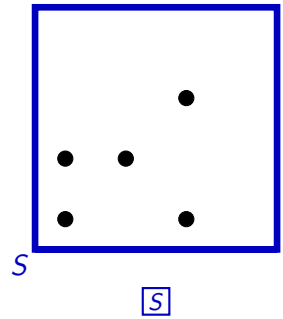
- choose a (suitable) square $S \subseteq \mathbb{R}^2$ such that
  - S contains all terminals
  - sidelength of S is $\mathcal{O}(n/\varepsilon)$ and a power of 2
  - corners of S lie in $(\mathbb{Z} + 0.5)^2$
- dissection to S: rooted tree where
  - each node is a square in $\mathbb{R}^2$ (called cells)
  - the root is S



Arora's algorithm: dissection and portals

grid and portal placement

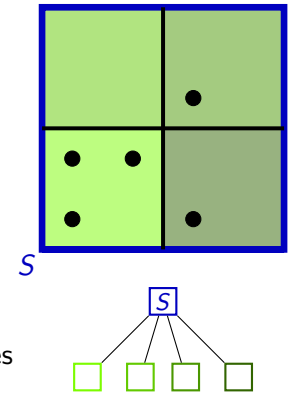
- choose a (suitable) square $S \subseteq \mathbb{R}^2$ such that
  - S contains all terminals
  - sidelength of S is $\mathcal{O}(n/\varepsilon)$ and a power of 2
  - corners of S lie in $(\mathbb{Z} + 0.5)^2$
- dissection to S: rooted tree where
  - each node is a square in $\mathbb{R}^2$ (called cells)
  - the root is S
  - each node of sidelength > 1 has 4 children: its 4 subquares



Arora's algorithm: dissection and portals

grid and portal placement

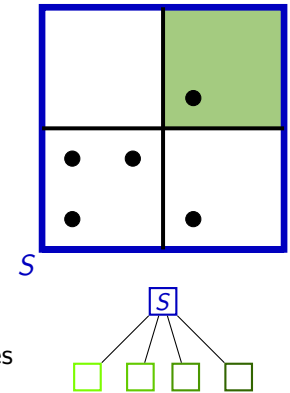
- choose a (suitable) square $S \subseteq \mathbb{R}^2$ such that
  - S contains all terminals
  - sidelength of S is $\mathcal{O}(n/\varepsilon)$ and a power of 2
  - corners of S lie in $(\mathbb{Z} + 0.5)^2$
- dissection to S: rooted tree where
  - each node is a square in $\mathbb{R}^2$ (called cells)
  - the root is S
  - each node of sidelength > 1 has 4 children: its 4 subquares



Arora's algorithm: dissection and portals

grid and portal placement

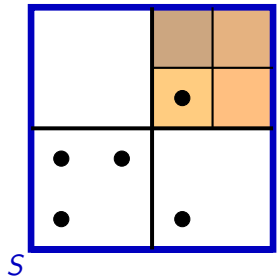
- choose a (suitable) square $S \subseteq \mathbb{R}^2$ such that
  - S contains all terminals
  - sidelength of S is $\mathcal{O}(n/\varepsilon)$ and a power of 2
  - corners of S lie in $(\mathbb{Z} + 0.5)^2$
- dissection to S: rooted tree where
  - each node is a square in $\mathbb{R}^2$ (called cells)
  - the root is S
  - each node of sidelength > 1 has 4 children: its 4 subquares



Arora's algorithm: dissection and portals

grid and portal placement

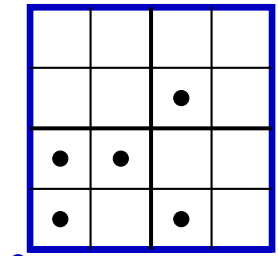
- choose a (suitable) square $S \subseteq \mathbb{R}^2$ such that
  - S contains all terminals
  - sidelength of S is $\mathcal{O}(n/\varepsilon)$ and a power of 2
  - corners of S lie in $(\mathbb{Z} + 0.5)^2$
- dissection to S: rooted tree where
  - each node is a square in $\mathbb{R}^2$ (called cells)
  - the root is S
  - each node of sidelength > 1 has 4 children: its 4 subquares

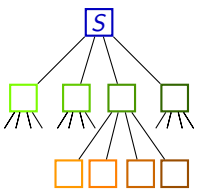


Arora's algorithm: dissection and portals

grid and portal placement

- choose a (suitable) square $S \subseteq \mathbb{R}^2$ such that
  - S contains all terminals
  - sidelength of S is $\mathcal{O}(n/\varepsilon)$ and a power of 2
  - corners of S lie in $(\mathbb{Z} + 0.5)^2$
- dissection to S: rooted tree where
  - each node is a square in $\mathbb{R}^2$ (called cells)
  - the root is S
  - each node of sidelength > 1 has 4 children: its 4 subquares

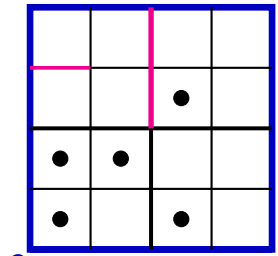


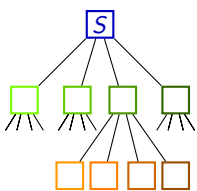


Arora's algorithm: dissection and portals

grid and portal placement

- choose a (suitable) square $S \subseteq \mathbb{R}^2$ such that
  - S contains all terminals
  - sidelength of S is $\mathcal{O}(n/\varepsilon)$ and a power of 2
  - corners of S lie in $(\mathbb{Z} + 0.5)^2$
- dissection to S: rooted tree where
  - each node is a square in $\mathbb{R}^2$ (called cells)
  - the root is S
  - each node of sidelength > 1 has 4 children: its 4 subquares
- boundary: border edge of a cell not contained in a larger cell

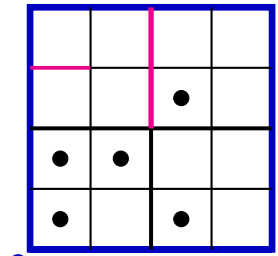


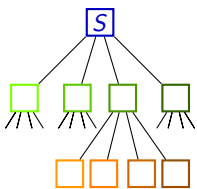


Arora's algorithm: dissection and portals

grid and portal placement

- choose a (suitable) square $S \subseteq \mathbb{R}^2$ such that
  - S contains all terminals
  - sidelength of S is $\mathcal{O}(n/\varepsilon)$ and a power of 2
  - corners of S lie in $(\mathbb{Z} + 0.5)^2$
- dissection to S: rooted tree where
  - each node is a square in $\mathbb{R}^2$ (called cells)
  - the root is S
  - each node of sidelength > 1 has 4 children: its 4 subquares
- boundary: border edge of a cell not contained in a larger cell
- portal: segment of length $\to 0$

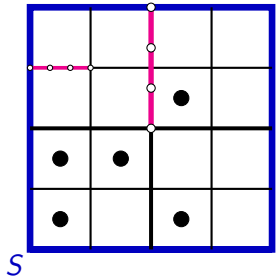




Arora's algorithm: dissection and portals

grid and portal placement

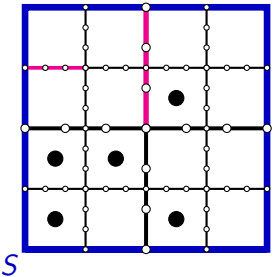
- choose a (suitable) square $S \subseteq \mathbb{R}^2$ such that
  - S contains all terminals
  - sidelength of S is $\mathcal{O}(n/\varepsilon)$ and a power of 2
  - corners of S lie in $(\mathbb{Z} + 0.5)^2$
- dissection to S: rooted tree where
  - each node is a square in $\mathbb{R}^2$ (called cells)
  - the root is S
  - each node of sidelength > 1 has 4 children: its 4 subquares
- boundary: border edge of a cell not contained in a larger cell
- portal: segment of length $\to 0$
- place $\Theta(\frac{1}{\varepsilon} \log(\frac{n}{\varepsilon}))$ equispaced portals on every boundary



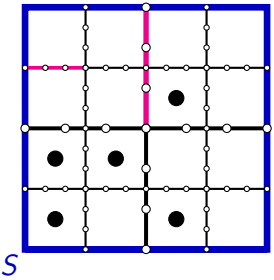
Arora's algorithm: dissection and portals

grid and portal placement

- choose a (suitable) square $S \subseteq \mathbb{R}^2$ such that
  - S contains all terminals
  - sidelength of S is $\mathcal{O}(n/\varepsilon)$ and a power of 2
  - corners of S lie in $(\mathbb{Z} + 0.5)^2$
- dissection to S: rooted tree where
  - each node is a square in $\mathbb{R}^2$ (called cells)
  - the root is S
  - each node of sidelength > 1 has 4 children: its 4 subquares
- boundary: border edge of a cell not contained in a larger cell
- portal: segment of length $\to 0$
- place $\Theta(\frac{1}{\varepsilon}\log(\frac{n}{\varepsilon}))$ equispaced portals on every boundary



Arora's algorithm: dissection and portals



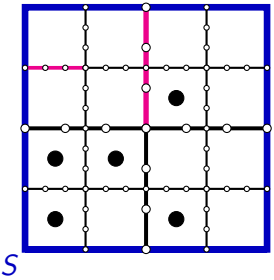
grid and portal placement

- choose a (suitable) square $S \subseteq \mathbb{R}^2$ such that
  - S contains all terminals
  - sidelength of S is $\mathcal{O}(n/\varepsilon)$ and a power of 2
  - corners of S lie in $(\mathbb{Z} + 0.5)^2$
- dissection to S: rooted tree where
  - each node is a square in $\mathbb{R}^2$ (called cells)
  - the root is S
  - each node of sidelength > 1 has 4 children: its 4 subquares
- boundary: border edge of a cell not contained in a larger cell
- portal: segment of length $\to 0$
- place $\Theta(\frac{1}{\varepsilon} \log(\frac{n}{\varepsilon}))$ equispaced portals on every boundary

remarks

- on cells of smaller sidelength, portals are placed more densely

Arora's algorithm: dissection and portals



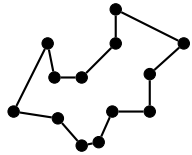
grid and portal placement

- choose a (suitable) square $S \subseteq \mathbb{R}^2$ such that
  - S contains all terminals
  - sidelength of S is $\mathcal{O}(n/\varepsilon)$ and a power of 2
  - corners of S lie in $(\mathbb{Z} + 0.5)^2$
- dissection to S: rooted tree where
  - each node is a square in $\mathbb{R}^2$ (called cells)
  - the root is S
  - each node of sidelength > 1 has 4 children: its 4 subquares
- boundary: border edge of a cell not contained in a larger cell
- portal: segment of length $\to 0$
- place $\Theta(\frac{1}{\varepsilon} \log(\frac{n}{\varepsilon}))$ equispaced portals on every boundary

remarks

- on cells of smaller sidelength, portals are placed more densely
- size of dissection: $\mathcal{O}((n/\varepsilon)^2)$ nodes

Overview of Arora's algorithm



Theorem (Arora 1998). There is a PTAS for ETSP.

i.e.: $\forall \varepsilon > 0$, can compute a $(1+\varepsilon)$-approximation in polynomial time

proof outline

- restrict to the case where terminals lie in $\mathbb{Z}^2$
  with maximum distance $\mathcal{O}(n/\varepsilon)$

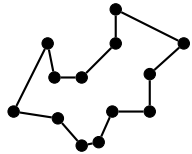
1st step: place a suitable grid in $\mathbb{R}^2$ (dissection) and portals on this grid

$\rightarrow$ "portal-respecting tour": only crosses the grid through portals

2nd step: an optimal portal-respecting solution can be computed in polynomial time

3rd step: there is a portal-respecting solution of length $(1+\varepsilon) \cdot \mathrm{OPT}$

Overview of Arora's algorithm

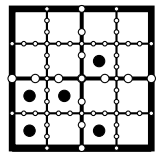


Theorem (Arora 1998). There is a PTAS for ETSP.

i.e.: $\forall \varepsilon > 0$, can compute a $(1+\varepsilon)$-approximation in polynomial time

proof outline

- restrict to the case where terminals lie in $\mathbb{Z}^2$
  with maximum distance $\mathcal{O}(n/\varepsilon)$



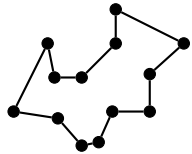
1st step: place a suitable grid in $\mathbb{R}^2$ (dissection) and portals on this grid

$\rightarrow$ "portal-respecting tour": only crosses the grid through portals

2nd step: an optimal portal-respecting solution can be computed in polynomial time

3rd step: there is a portal-respecting solution of length $(1+\varepsilon) \cdot \mathrm{OPT}$

Overview of Arora's algorithm

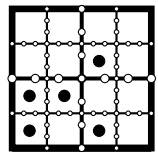


Theorem (Arora 1998). There is a PTAS for ETSP.

i.e.: $\forall \varepsilon > 0$, can compute a $(1+\varepsilon)$-approximation in polynomial time

proof outline

- restrict to the case where terminals lie in $\mathbb{Z}^2$
  with maximum distance $\mathcal{O}(n/\varepsilon)$



1st step: place a suitable grid in $\mathbb{R}^2$ (dissection) and portals on this grid

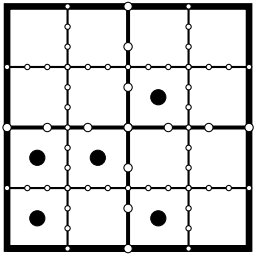
$\rightarrow$ "portal-respecting tour": only crosses the grid through portals

2nd step: an optimal portal-respecting solution can be computed in polynomial time

3rd step: there is a portal-respecting solution of length $(1+\varepsilon) \cdot \mathrm{OPT}$

Arora's algorithm: the multipath problem

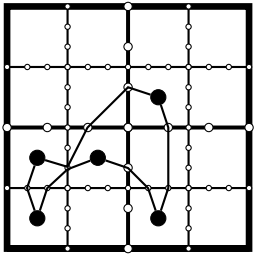
tours in dissection



Arora's algorithm: the multipath problem

tours in dissection

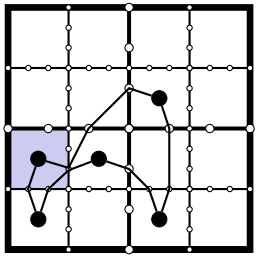
- a tour is portal-respecting if
  - it only crosses the dissection at portals
  - each portal is crossed at most a constant number of times

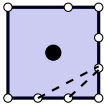


Arora's algorithm: the multipath problem

tours in dissection

- a tour is portal-respecting if
  - it only crosses the dissection at portals
  - each portal is crossed at most a constant number of times
- a cell crossing pattern is a set of pairs of portals
  (specifies through which portals a tour enters/leaves the cell)

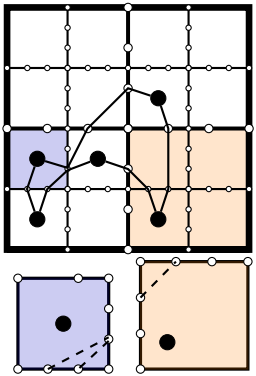




Arora's algorithm: the multipath problem

tours in dissection

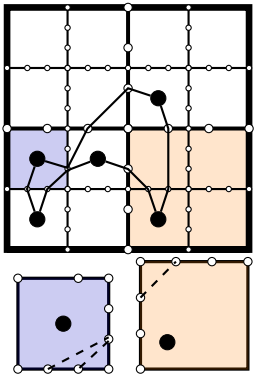
- a tour is portal-respecting if
  - it only crosses the dissection at portals
  - each portal is crossed at most a constant number of times
- a cell crossing pattern is a set of pairs of portals
  (specifies through which portals a tour enters/leaves the cell)



Arora's algorithm: the multipath problem

tours in dissection

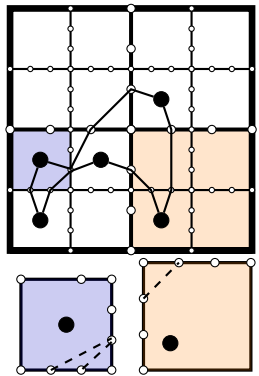
- a tour is portal-respecting if
  - it only crosses the dissection at portals
  - each portal is crossed at most a constant number of times
- a cell crossing pattern is a set of pairs of portals
  (specifies through which portals a tour enters/leaves the cell)
- recall: every cell border contains $\mathcal{O}_\varepsilon(\log n)$ portals



Arora's algorithm: the multipath problem

tours in dissection

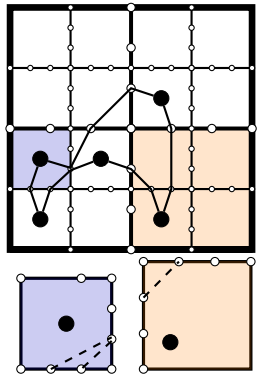
- a tour is portal-respecting if
  - it only crosses the dissection at portals
  - each portal is crossed at most a constant number of times
- a cell crossing pattern is a set of pairs of portals
  (specifies through which portals a tour enters/leaves the cell)
- recall: every cell border contains $\mathcal{O}_\varepsilon(\log n)$ portals
  $\rightarrow$ Lemma. There are poly(n) relevant cell crossing patterns.



Arora's algorithm: the multipath problem

tours in dissection

- a tour is portal-respecting if
  - it only crosses the dissection at portals
  - each portal is crossed at most a constant number of times
- a cell crossing pattern is a set of pairs of portals
  (specifies through which portals a tour enters/leaves the cell)
- recall: every cell border contains $\mathcal{O}_\varepsilon(\log n)$ portals
  $\rightarrow$ Lemma. There are poly(n) relevant cell crossing patterns.

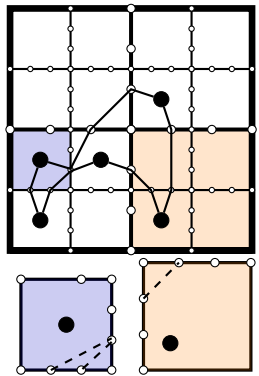


The multipath problem (informal)

Arora's algorithm: the multipath problem

tours in dissection

- a tour is portal-respecting if
  - it only crosses the dissection at portals
  - each portal is crossed at most a constant number of times
- a cell crossing pattern is a set of pairs of portals
  (specifies through which portals a tour enters/leaves the cell)
- recall: every cell border contains $\mathcal{O}_\varepsilon(\log n)$ portals
  $\rightarrow$ Lemma. There are poly(n) relevant cell crossing patterns.



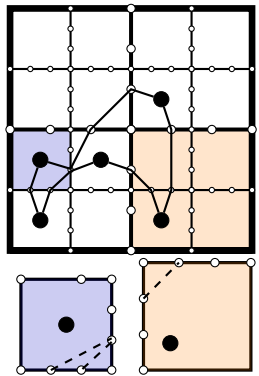
The multipath problem (informal)

input: cell A of the dissection

Arora's algorithm: the multipath problem

tours in dissection

- a tour is portal-respecting if
  - it only crosses the dissection at portals
  - each portal is crossed at most a constant number of times
- a cell crossing pattern is a set of pairs of portals
  (specifies through which portals a tour enters/leaves the cell)
- recall: every cell border contains $\mathcal{O}_\varepsilon(\log n)$ portals
  $\rightarrow$ Lemma. There are poly(n) relevant cell crossing patterns.

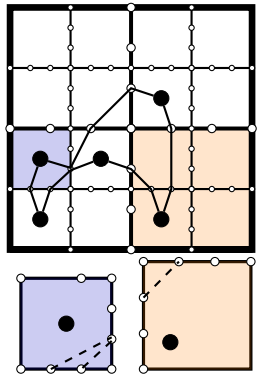


The multipath problem (informal)	
input:	cell A of the dissection
task:	for every relevant cell crossing pattern:

Arora's algorithm: the multipath problem

tours in dissection

- a tour is portal-respecting if
  - it only crosses the dissection at portals
  - each portal is crossed at most a constant number of times
- a cell crossing pattern is a set of pairs of portals
  (specifies through which portals a tour enters/leaves the cell)
- recall: every cell border contains $\mathcal{O}_\varepsilon(\log n)$ portals
  $\rightarrow$ Lemma. There are poly(n) relevant cell crossing patterns.



The multipath problem (informal)

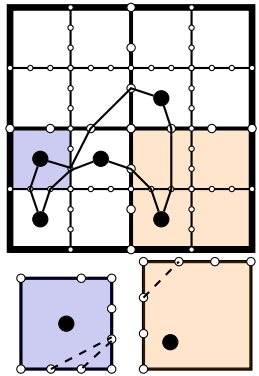
input: cell A of the dissection

task: for every relevant cell crossing pattern:
compute the shortest $\pi \cap A$ where
- π is a portal-respecting tour
- π satisfies the cell crossing pattern at A

Arora's algorithm: the multipath problem

tours in dissection

- a tour is portal-respecting if
  - it only crosses the dissection at portals
  - each portal is crossed at most a constant number of times
- a cell crossing pattern is a set of pairs of portals
  (specifies through which portals a tour enters/leaves the cell)
- recall: every cell border contains $\mathcal{O}_\varepsilon(\log n)$ portals
  $\rightarrow$ Lemma. There are poly(n) relevant cell crossing patterns.

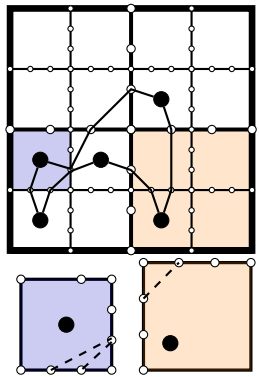


The multipath problem (informal)	
input:	cell A of the dissection
task:	for every relevant cell crossing pattern: compute the shortest $\pi \cap A$ where ▸ π is a portal-respecting tour ▸ π satisfies the cell crossing pattern at A ▸ π visits all terminals in A

Arora's algorithm: the multipath problem

tours in dissection

- a tour is portal-respecting if
  - it only crosses the dissection at portals
  - each portal is crossed at most a constant number of times
- a cell crossing pattern is a set of pairs of portals
  (specifies through which portals a tour enters/leaves the cell)
- recall: every cell border contains $\mathcal{O}_\varepsilon(\log n)$ portals
  $\rightarrow$ Lemma. There are poly(n) relevant cell crossing patterns.

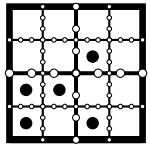


The multipath problem (informal)	
input:	cell A of the dissection
task:	for every relevant cell crossing pattern: compute the shortest $\pi \cap A$ where ▶ π is a portal-respecting tour ▶ π satisfies the cell crossing pattern at A ▶ π visits all terminals in A

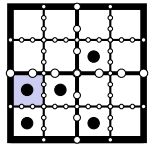
note: multipath$(S, \emptyset)$ gives optimal portal-resp. solution

Arora's algorithm: solving the multipath problem

key idea: dynamic programming

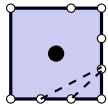


Arora's algorithm: solving the multipath problem

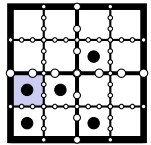


key idea: dynamic programming

for leafs (cells of sidelength 1):



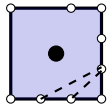
Arora's algorithm: solving the multipath problem



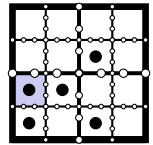
key idea: dynamic programming

for leafs (cells of sidelength 1):

- leafs contain at most 1 terminal



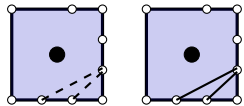
Arora's algorithm: solving the multipath problem



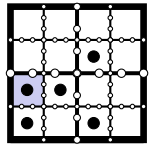
key idea: dynamic programming

for leafs (cells of sidelength 1):

- leafs contain at most 1 terminal
- realize every pair of portals by a straight line segment



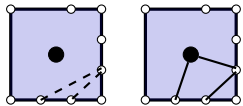
Arora's algorithm: solving the multipath problem



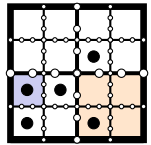
key idea: dynamic programming

for leafs (cells of sidelength 1):

- leafs contain at most 1 terminal
- realize every pair of portals by a straight line segment
  - choose a pair of portals to cover the terminal minimizing the total length



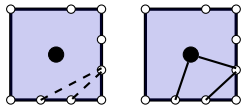
Arora's algorithm: solving the multipath problem

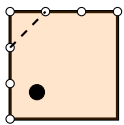


key idea: dynamic programming

for leafs (cells of sidelength 1):

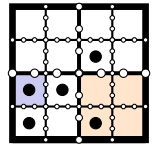
- leafs contain at most 1 terminal
- realize every pair of portals by a straight line segment
  - choose a pair of portals to cover the terminal minimizing the total length





for non-leafs

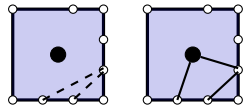
Arora's algorithm: solving the multipath problem



key idea: dynamic programming

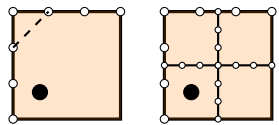
for leafs (cells of sidelength 1):

- leafs contain at most 1 terminal
- realize every pair of portals by a straight line segment
  - choose a pair of portals to cover the terminal minimizing the total length

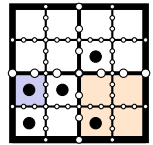


for non-leafs

- check every compatible combination of cell crossing patterns of children (already computed)



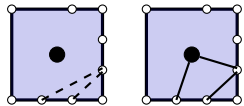
Arora's algorithm: solving the multipath problem



key idea: dynamic programming

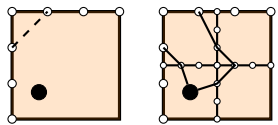
for leafs (cells of sidelength 1):

- leafs contain at most 1 terminal
- realize every pair of portals by a straight line segment
  - choose a pair of portals to cover the terminal minimizing the total length

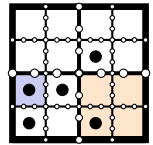


for non-leafs

- check every compatible combination of cell crossing patterns of children (already computed)



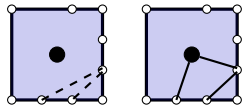
Arora's algorithm: solving the multipath problem



key idea: dynamic programming

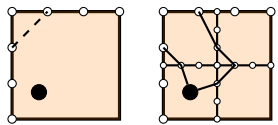
for leafs (cells of sidelength 1):

- leafs contain at most 1 terminal
- realize every pair of portals by a straight line segment
  - choose a pair of portals to cover the terminal minimizing the total length

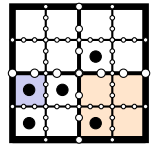


for non-leafs

- check every compatible combination of cell crossing patterns of children (already computed)
- choose the solution minimizing the total length



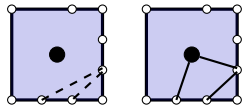
Arora's algorithm: solving the multipath problem



key idea: dynamic programming

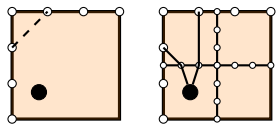
for leafs (cells of sidelength 1):

- leafs contain at most 1 terminal
- realize every pair of portals by a straight line segment
  - choose a pair of portals to cover the terminal minimizing the total length

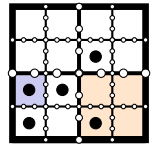


for non-leafs

- check every compatible combination of cell crossing patterns of children (already computed)
- choose the solution minimizing the total length



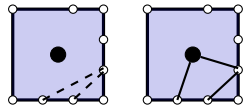
Arora's algorithm: solving the multipath problem



key idea: dynamic programming

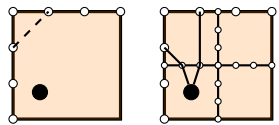
for leafs (cells of sidelength 1):

- leafs contain at most 1 terminal
- realize every pair of portals by a straight line segment
  - choose a pair of portals to cover the terminal minimizing the total length



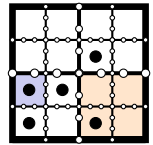
for non-leafs

- check every compatible combination of cell crossing patterns of children (already computed)
- choose the solution minimizing the total length



Theorem (Arora): The optimal portal-respecting solution for ETSP can be computed in polynomial time.

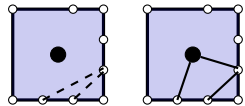
Arora's algorithm: solving the multipath problem



key idea: dynamic programming

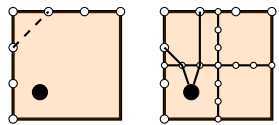
for leafs (cells of sidelength 1):

- leafs contain at most 1 terminal
- realize every pair of portals by a straight line segment
  - choose a pair of portals to cover the terminal minimizing the total length



for non-leafs

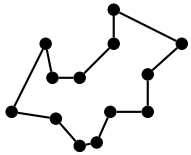
- check every compatible combination of cell crossing patterns of children (already computed)
- choose the solution minimizing the total length



Theorem (Arora): The optimal portal-respecting solution for ETSP can be computed in polynomial time.

Theorem (B., Disser, Feldmann, Zych-Pawlewicz): For all $k \in \mathbb{N}$, the optimal portal-respecting solution for k-ETSP can be computed in polynomial time (for all $k \in \mathbb{N}$).

Overview of Arora's algorithm



Theorem (Arora 1998). There is a PTAS for ETSP.

i.e.: $\forall \varepsilon > 0$, can compute a $(1+\varepsilon)$-approximation in polynomial time

proof outline

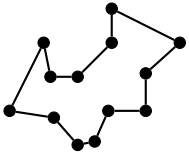
- restrict to the case where terminals lie in $\mathbb{Z}^2$
  with maximum distance $\mathcal{O}(n/\varepsilon)$

1st step: place a suitable grid in $\mathbb{R}^2$ (dissection) and portals on this grid

2nd step: optimal portal-respecting solution can be computed in polynomial time

3rd step: there is a portal-respecting solution of length $(1+\varepsilon) \cdot \mathrm{OPT}$

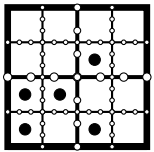
Overview of Arora's algorithm



Theorem (Arora 1998). There is a PTAS for ETSP.

i.e.: $\forall \varepsilon > 0$, can compute a $(1+\varepsilon)$-approximation in polynomial time

proof outline



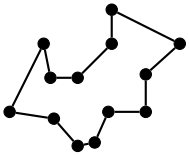
- restrict to the case where terminals lie in $\mathbb{Z}^2$
  with maximum distance $\mathcal{O}(n/\varepsilon)$

1st step: place a suitable grid in $\mathbb{R}^2$ (dissection) and portals on this grid

2nd step: optimal portal-respecting solution can be computed in polynomial time

3rd step: there is a portal-respecting solution of length $(1+\varepsilon) \cdot \mathrm{OPT}$

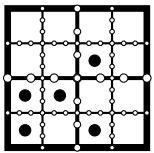
Overview of Arora's algorithm



Theorem (Arora 1998). There is a PTAS for ETSP.

i.e.: $\forall \varepsilon > 0$, can compute a $(1+\varepsilon)$-approximation in polynomial time

proof outline



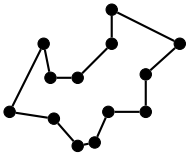
- restrict to the case where terminals lie in $\mathbb{Z}^2$
  with maximum distance $\mathcal{O}(n/\varepsilon)$

1st step: place a suitable grid in $\mathbb{R}^2$ (dissection) and portals on this grid

2nd step: optimal portal-respecting solution can be computed in polynomial time

3rd step: there is a portal-respecting solution of length $(1+\varepsilon) \cdot \textsc{Opt}$

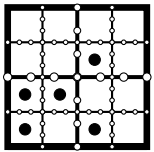
Overview of Arora's algorithm



Theorem (Arora 1998). There is a PTAS for ETSP.

i.e.: $\forall \varepsilon > 0$, can compute a $(1+\varepsilon)$-approximation in polynomial time

proof outline



- restrict to the case where terminals lie in $\mathbb{Z}^2$
  with maximum distance $\mathcal{O}(n/\varepsilon)$

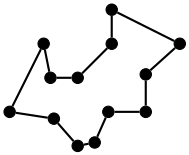
1st step: place a suitable grid in $\mathbb{R}^2$ (dissection) and portals on this grid

2nd step: optimal portal-respecting solution can be computed in polynomial time

3rd step: there is a portal-respecting solution of length $(1+\varepsilon) \cdot \mathrm{OPT}$

- 3.1: transform optimum solution such that it only crosses at portals
- 3.2: transform such that portals are crossed constantly often

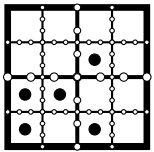
Overview of Arora's algorithm



Theorem (Arora 1998). There is a PTAS for ETSP.

i.e.: $\forall \varepsilon > 0$, can compute a $(1+\varepsilon)$-approximation in polynomial time

proof outline



- restrict to the case where terminals lie in $\mathbb{Z}^2$
  with maximum distance $\mathcal{O}(n/\varepsilon)$

1st step: place a suitable grid in $\mathbb{R}^2$ (dissection) and portals on this grid

2nd step: optimal portal-respecting solution can be computed in polynomial time

3rd step: there is a portal-respecting solution of length $(1+\varepsilon) \cdot \mathrm{OPT}$

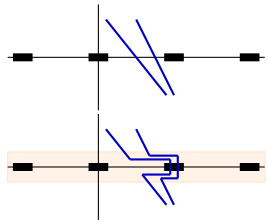
- 3.1: transform optimum solution such that it only crosses at portals
- 3.2: transform such that portals are crossed constantly often

Arora's algorithm: snapping crossings to portals

- consider an optimal solution to ETSP

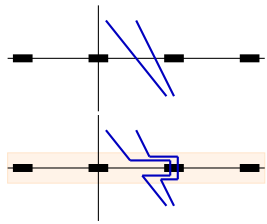
Arora's algorithm: snapping crossings to portals

- consider an optimal solution to ETSP
- snapping procedure: move all crossings with dissection to portals (in suitable order)



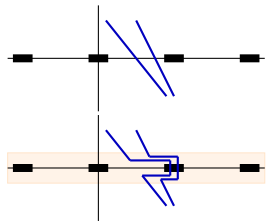
Arora's algorithm: snapping crossings to portals

- consider an optimal solution to ETSP
- snapping procedure: move all crossings with dissection to portals (in suitable order)
- recall: dissection depends on an initially chosen square S



Arora's algorithm: snapping crossings to portals

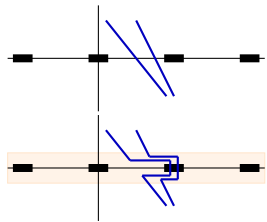
- consider an optimal solution to ETSP
- snapping procedure: move all crossings with dissection to portals (in suitable order)
- recall: dissection depends on an initially chosen square S



Theorem (Arora). There is a choice for S such that the length of the snapped tour is at most $(1+\varepsilon) \cdot \mathrm{OPT}$.

Arora's algorithm: snapping crossings to portals

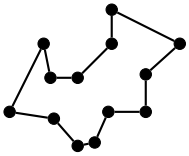
- consider an optimal solution to ETSP
- snapping procedure: move all crossings with dissection to portals (in suitable order)
- recall: dissection depends on an initially chosen square S



Theorem (Arora). There is a choice for S such that the length of the snapped tour is at most $(1+\varepsilon)\cdot \mathrm{OPT}$.

Theorem (B., Disser, Feldmann, Zych-Pawlewicz). There is a choice for S and a snapping procedure for k-ETSP preserving that tours are non-crossing such that the total length of the snapped tours is at most $(1+\varepsilon)\cdot \mathrm{OPT}$.

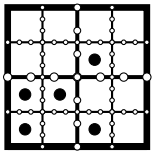
Overview of Arora's algorithm



Theorem (Arora 1998). There is a PTAS for ETSP.

i.e.: $\forall \varepsilon > 0$, can compute a $(1+\varepsilon)$-approximation in polynomial time

proof outline



- restrict to the case where terminals are distinct points in $\mathbb{Z}^2$ with maximum distance $\mathcal{O}(n/\varepsilon)$

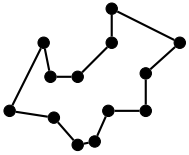
1st step: place a suitable grid in $\mathbb{R}^2$ (dissection) and portals on this grid

2nd step: optimal portal-respecting solution can be computed in poly-time

3rd step: there is a portal-respecting solution of length $(1+\varepsilon) \cdot \mathrm{OPT}$

- 3.1: transform optimum solution such that it only crosses at portals
- 3.2: transform such that portals are crossed constantly often

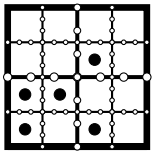
Overview of Arora's algorithm



Theorem (Arora 1998). There is a PTAS for ETSP.

i.e.: $\forall \varepsilon > 0$, can compute a $(1+\varepsilon)$-approximation in polynomial time

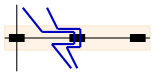
proof outline



- restrict to the case where terminals are distinct points in $\mathbb{Z}^2$ with maximum distance $\mathcal{O}(n/\varepsilon)$

1st step: place a suitable grid in $\mathbb{R}^2$ (dissection) and portals on this grid

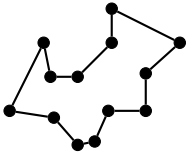
2nd step: optimal portal-respecting solution can be computed in poly-time



3rd step: there is a portal-respecting solution of length $(1+\varepsilon) \cdot \mathrm{OPT}$

- 3.1: transform optimum solution such that it only crosses at portals
- 3.2: transform such that portals are crossed constantly often

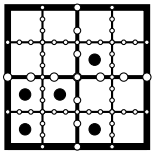
Overview of Arora's algorithm



Theorem (Arora 1998). There is a PTAS for ETSP.

i.e.: $\forall \varepsilon > 0$, can compute a $(1+\varepsilon)$-approximation in polynomial time

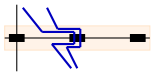
proof outline



- restrict to the case where terminals are distinct points in $\mathbb{Z}^2$ with maximum distance $\mathcal{O}(n/\varepsilon)$

1st step: place a suitable grid in $\mathbb{R}^2$ (dissection) and portals on this grid

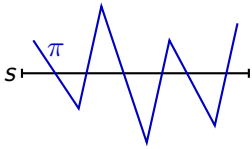
2nd step: optimal portal-respecting solution can be computed in poly-time



3rd step: there is a portal-respecting solution of length $(1+\varepsilon) \cdot \mathrm{OPT}$

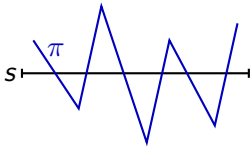
- 3.1: transform optimum solution such that it only crosses at portals
- 3.2: transform such that portals are crossed constantly often

Arora's algorithm: patching



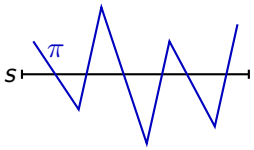
Patching Lemma (Arora). Let π be a tour and s a line segment.

Arora's algorithm: patching



Patching Lemma (Arora). Let π be a tour and s a line segment. Then, there is a tour π' such that

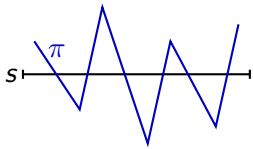
Arora's algorithm: patching



Patching Lemma (Arora). Let π be a tour and s a line segment. Then, there is a tour π' such that

- π' intersects s in at most 2 points,

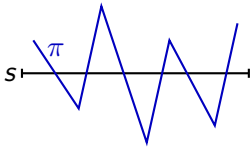
Arora's algorithm: patching



Patching Lemma (Arora). Let π be a tour and s a line segment. Then, there is a tour π' such that

- π' intersects s in at most 2 points,
- π and π' only differ in a small neighbourhood of s,

Arora's algorithm: patching



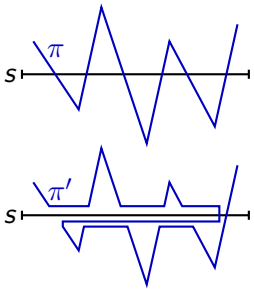
Patching Lemma (Arora). Let π be a tour and s a line segment. Then, there is a tour π' such that

- π' intersects s in at most 2 points,
- π and π' only differ in a small neighbourhood of s,
- $l(\pi') \leq l(\pi) + \mathcal{O}(l(s))$.

Arora's algorithm: patching

Patching Lemma (Arora). Let π be a tour and s a line segment. Then, there is a tour π' such that

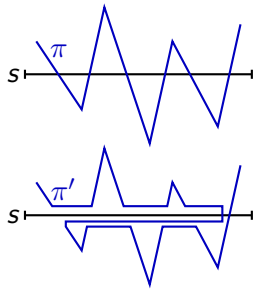
- π' intersects s in at most 2 points,
- π and π' only differ in a small neighbourhood of s,
- $l(\pi') \leq l(\pi) + \mathcal{O}(l(s))$.



Arora's algorithm: patching

Patching Lemma (Arora). Let π be a tour and s a line segment. Then, there is a tour π' such that

- π' intersects s in at most 2 points,
- π and π' only differ in a small neighbourhood of s,
- $l(\pi') \leq l(\pi) + \mathcal{O}(l(s))$.

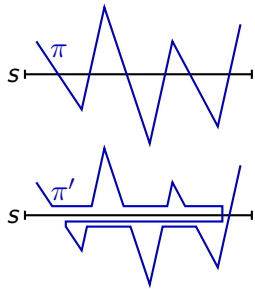


Structure Theorem (Arora). For every instance of ETSP and $\varepsilon > 0$, there is a dissection of size $\mathcal{O}_\varepsilon(n^2)$ and a portal-respecting solution (to this dissection) of length $\leq (1+\varepsilon) \cdot \mathrm{OPT}$.

Arora's algorithm: patching

Patching Lemma (Arora). Let π be a tour and s a line segment. Then, there is a tour π' such that

- π' intersects s in at most 2 points,
- π and π' only differ in a small neighbourhood of s,
- $l(\pi') \leq l(\pi) + \mathcal{O}(l(s))$.

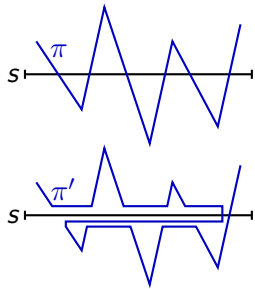


Structure Theorem (Arora). For every instance of ETSP and $\varepsilon > 0$, there is a dissection of size $\mathcal{O}_\varepsilon(n^2)$ and a portal-respecting solution (to this dissection) of length $\leq (1+\varepsilon) \cdot \mathrm{OPT}$.

proof.

- apply the portal snapping procedure to an optimum solution

Arora's algorithm: patching



Patching Lemma (Arora). Let π be a tour and s a line segment. Then, there is a tour π' such that

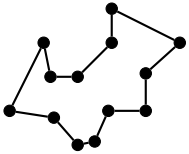
- π' intersects s in at most 2 points,
- π and π' only differ in a small neighbourhood of s,
- $l(\pi') \leq l(\pi) + \mathcal{O}(l(s))$.

Structure Theorem (Arora). For every instance of ETSP and $\varepsilon > 0$, there is a dissection of size $\mathcal{O}_\varepsilon(n^2)$ and a portal-respecting solution (to this dissection) of length $\leq (1+\varepsilon) \cdot \mathrm{OPT}$.

proof.

- apply the portal snapping procedure to an optimum solution
- apply patching inside every portal
  (in a suitable order to avoid creating new crossings)

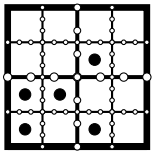
Overview of Arora's algorithm



Theorem (Arora 1998). There is a PTAS for ETSP.

i.e.: $\forall \varepsilon > 0$, can compute a $(1+\varepsilon)$-approximation in polynomial time

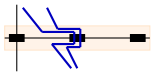
proof outline



- restrict to the case where terminals lie in $\mathbb{Z}^2$
  with maximum distance $\mathcal{O}(n/\varepsilon)$

1st step: place a suitable grid in $\mathbb{R}^2$ (dissection) and portals on this grid

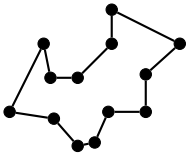
2nd step: optimal portal-respecting solution can be computed in poly-time



3rd step: there is a portal-respecting solution of length $(1+\varepsilon) \cdot \mathrm{OPT}$

- 3.1: transform optimum solution such that it only crosses at portals
- 3.2: transform such that portals are crossed constantly often (patching)

Overview of Arora's algorithm

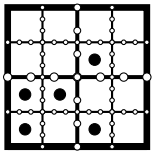


Theorem (Arora 1998). There is a PTAS for ETSP.

i.e.: $\forall \varepsilon > 0$, can compute a $(1+\varepsilon)$-approximation in polynomial time

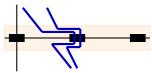
proof outline

- restrict to the case where terminals lie in $\mathbb{Z}^2$
  with maximum distance $\mathcal{O}(n/\varepsilon)$



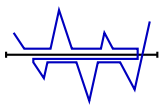
1st step: place a suitable grid in $\mathbb{R}^2$ (dissection) and portals on this grid

2nd step: optimal portal-respecting solution can be computed in poly-time

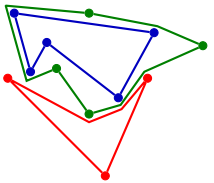


3rd step: there is a portal-respecting solution of length $(1+\varepsilon) \cdot \textsc{Opt}$

- 3.1: transform optimum solution such that it only crosses at portals
- 3.2: transform such that portals are crossed constantly often (patching)

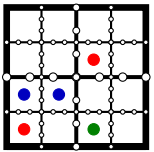


What generalizes to k-ETSP?



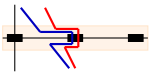
ingredients

- restrict to the case where terminals lie in $\mathbb{Z}^2$



1st step: place a suitable grid in $\mathbb{R}^2$ (dissection) and portals on this grid

2nd step: optimal portal-respecting solution can be computed in poly-time



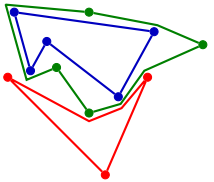
3rd step: there is a portal-respecting solution of length $(1+\varepsilon)\cdot \mathrm{OPT}$

- 3.1: transform optimum solution such that it only crosses at portals



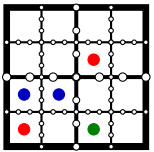
- 3.2: transform such that portals are crossed constantly often (patching)

What generalizes to k-ETSP?



ingredients

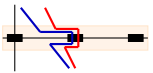
- restrict to the case where terminals lie in $\mathbb{Z}^2$



1st step: place a suitable grid in $\mathbb{R}^2$ (dissection) and portals on this grid

2nd step: optimal portal-respecting solution can be computed in poly-time

generalizes $\forall k \in \mathbb{N}$



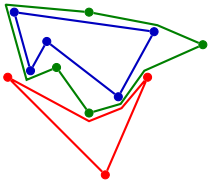
3rd step: there is a portal-respecting solution of length $(1+\varepsilon) \cdot \mathrm{OPT}$

- 3.1: transform optimum solution such that it only crosses at portals



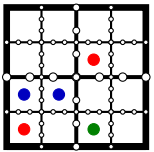
- 3.2: transform such that portals are crossed constantly often (patching)

What generalizes to k-ETSP?



ingredients

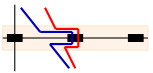
- restrict to the case where terminals lie in $\mathbb{Z}^2$



1st step: place a suitable grid in $\mathbb{R}^2$ (dissection) and portals on this grid

2nd step: optimal portal-respecting solution can be computed in poly-time

generalizes $\forall k \in \mathbb{N}$



3rd step: there is a portal-respecting solution of length $(1+\varepsilon) \cdot \mathrm{OPT}$

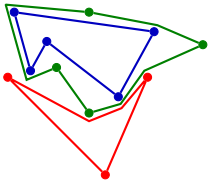
- 3.1: transform optimum solution such that it only crosses at portals

generalizes $\forall k \in \mathbb{N}$



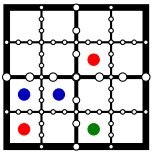
- 3.2: transform such that portals are crossed constantly often (patching)

What generalizes to k-ETSP?



ingredients

- restrict to the case where terminals lie in $\mathbb{Z}^2$

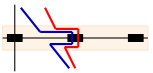


1st step: place a suitable grid in $\mathbb{R}^2$ (dissection) and portals on this grid

2nd step: optimal portal-respecting solution can be computed in poly-time

generalizes $\forall k \in \mathbb{N}$

3rd step: there is a portal-respecting solution of length $(1+\varepsilon) \cdot \mathrm{OPT}$



- 3.1: transform optimum solution such that it only crosses at portals

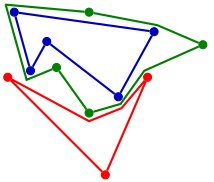
generalizes $\forall k \in \mathbb{N}$



- 3.2: transform such that portals are crossed constantly often (patching)

only generalizes to $k \leq 2$

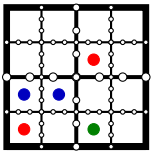
What generalizes to k-ETSP?



ingredients

- restrict to the case where terminals lie in $\mathbb{Z}^2$

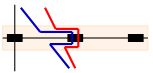
only generalizes to $k \leq 3$



1st step: place a suitable grid in $\mathbb{R}^2$ (dissection) and portals on this grid

2nd step: optimal portal-respecting solution can be computed in poly-time

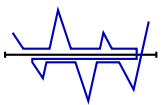
generalizes $\forall k \in \mathbb{N}$



3rd step: there is a portal-respecting solution of length $(1+\varepsilon) \cdot \mathrm{OPT}$

- 3.1: transform optimum solution such that it only crosses at portals

generalizes $\forall k \in \mathbb{N}$



- 3.2: transform such that portals are crossed constantly often (patching)

only generalizes to $k \leq 2$

Patching for multiple non-crossing tours

2-ETSP

- patching can be generalized to two non-crossing tours

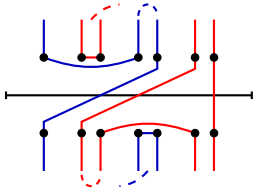
[Dross, Fleszar, Wegrzycki, Zych-Pawlewicz (2023)]

Patching for multiple non-crossing tours

2-ETSP

- patching can be generalized to two non-crossing tours

[Dross, Fleszar, Wegrzycki, Zych-Pawlewicz (2023)]

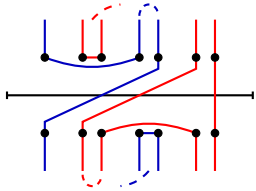


Patching for multiple non-crossing tours

2-ETSP

- patching can be generalized to two non-crossing tours

[Dross, Fleszar, Wegrzycki, Zych-Pawlewicz (2023)]



k-ETSP for $k \geq 3$

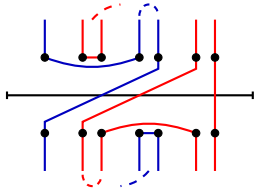
- patching cannot be generalized to more than two non-crossing tours

Patching for multiple non-crossing tours

2-ETSP

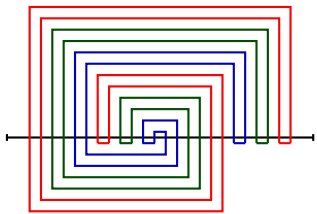
- patching can be generalized to two non-crossing tours

[Dross, Fleszar, Wegrzycki, Zych-Pawlewicz (2023)]



k-ETSP for $k \geq 3$

- patching cannot be generalized to more than two non-crossing tours



Conditional patching of three tours

Tricolored Patching Lemma (B., Disser, Feldmann, Zych-Pawlewicz).
Let $\pi_{\mathrm{R}}, \pi_{\mathrm{B}}, \pi_{\mathrm{G}}$ be non-crossing tours and s a line segment such that

Conditional patching of three tours

Tricolored Patching Lemma (B., Disser, Feldmann, Zych-Pawlewicz).
Let $\pi_{\mathrm{R}}, \pi_{\mathrm{B}}, \pi_{\mathrm{G}}$ be non-crossing tours and s a line segment such that no red crossing is next to a green crossing (along s).

Conditional patching of three tours

Tricolored Patching Lemma (B., Disser, Feldmann, Zych-Pawlewicz).
Let $\pi_{\mathrm{R}}, \pi_{\mathrm{B}}, \pi_{\mathrm{G}}$ be non-crossing tours and s a line segment such that no red crossing is next to a green crossing (along s).
Then, there are tours $\pi'_{\mathrm{R}}, \pi'_{\mathrm{B}}, \pi'_{\mathrm{G}}$ such that

- $\pi'_{\mathrm{R}}, \pi'_{\mathrm{B}}, \pi'_{\mathrm{G}}$ intersect s in at most 18 points,
- π_c and π'_c only differ in a small neighbourhood of s ($c \in \{\mathrm{R}, \mathrm{B}, \mathrm{G}\}$),
- $\sum_{c \in \{\mathrm{R},\mathrm{B},\mathrm{G}\}} l(\pi_c) \leq \sum_{c \in \{\mathrm{R},\mathrm{B},\mathrm{G}\}} l(\pi'_c) + \mathcal{O}(l(s))$.

Conditional patching of three tours

Tricolored Patching Lemma (B., Disser, Feldmann, Zych-Pawlewicz).
Let $\pi_{\mathrm{R}}, \pi_{\mathrm{B}}, \pi_{\mathrm{G}}$ be non-crossing tours and s a line segment such that no red crossing is next to a green crossing (along s).
Then, there are tours $\pi'_{\mathrm{R}}, \pi'_{\mathrm{B}}, \pi'_{\mathrm{G}}$ such that

- $\pi'_{\mathrm{R}}, \pi'_{\mathrm{B}}, \pi'_{\mathrm{G}}$ intersect s in at most 18 points,
- π_c and π'_c only differ in a small neighbourhood of s ($c \in \{\mathrm{R}, \mathrm{B}, \mathrm{G}\}$),
- $\sum_{c\in\{\mathrm{R},\mathrm{B},\mathrm{G}\}} l(\pi_c) \leq \sum_{c\in\{\mathrm{R},\mathrm{B},\mathrm{G}\}} l(\pi'_c) + \mathcal{O}(l(s))$.

proof sketch

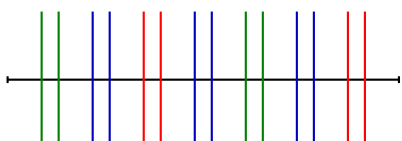
Conditional patching of three tours

Tricolored Patching Lemma (B., Disser, Feldmann, Zych-Pawlewicz).
Let $\pi_{\mathrm{R}}, \pi_{\mathrm{B}}, \pi_{\mathrm{G}}$ be non-crossing tours and s a line segment such that no red crossing is next to a green crossing (along s).
Then, there are tours $\pi'_{\mathrm{R}}, \pi'_{\mathrm{B}}, \pi'_{\mathrm{G}}$ such that

- $\pi'_{\mathrm{R}}, \pi'_{\mathrm{B}}, \pi'_{\mathrm{G}}$ intersect s in at most 18 points,
- π_c and π'_c only differ in a small neighbourhood of s ($c \in \{\mathrm{R}, \mathrm{B}, \mathrm{G}\}$),
- $\sum_{c\in\{\mathrm{R},\mathrm{B},\mathrm{G}\}} l(\pi_c) \leq \sum_{c\in\{\mathrm{R},\mathrm{B},\mathrm{G}\}} l(\pi'_c) + \mathcal{O}(l(s))$.

proof sketch

- apply 1- and 2-color patching to reduce to the case



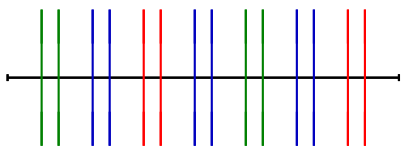
Conditional patching of three tours

Tricolored Patching Lemma (B., Disser, Feldmann, Zych-Pawlewicz).
Let $\pi_{\mathrm{R}}, \pi_{\mathrm{B}}, \pi_{\mathrm{G}}$ be non-crossing tours and s a line segment such that no red crossing is next to a green crossing (along s).
Then, there are tours $\pi'_{\mathrm{R}}, \pi'_{\mathrm{B}}, \pi'_{\mathrm{G}}$ such that

- $\pi'_{\mathrm{R}}, \pi'_{\mathrm{B}}, \pi'_{\mathrm{G}}$ intersect s in at most 18 points,
- π_c and π'_c only differ in a small neighbourhood of s ($c \in \{\mathrm{R}, \mathrm{B}, \mathrm{G}\}$),
- $\sum_{c\in\{\mathrm{R},\mathrm{B},\mathrm{G}\}} l(\pi_c) \leq \sum_{c\in\{\mathrm{R},\mathrm{B},\mathrm{G}\}} l(\pi'_c) + \mathcal{O}(l(s))$.

proof sketch

- apply 1- and 2-color patching to reduce to the case
- apply patching scheme (up to symmetry)



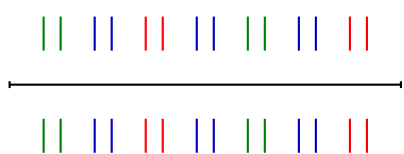
Conditional patching of three tours

Tricolored Patching Lemma (B., Disser, Feldmann, Zych-Pawlewicz).
Let $\pi_{\mathrm{R}}, \pi_{\mathrm{B}}, \pi_{\mathrm{G}}$ be non-crossing tours and s a line segment such that no red crossing is next to a green crossing (along s).
Then, there are tours $\pi'_{\mathrm{R}}, \pi'_{\mathrm{B}}, \pi'_{\mathrm{G}}$ such that

- $\pi'_{\mathrm{R}}, \pi'_{\mathrm{B}}, \pi'_{\mathrm{G}}$ intersect s in at most 18 points,
- π_c and π'_c only differ in a small neighbourhood of s ($c \in \{\mathrm{R}, \mathrm{B}, \mathrm{G}\}$),
- $\sum_{c\in\{\mathrm{R},\mathrm{B},\mathrm{G}\}} l(\pi_c) \leq \sum_{c\in\{\mathrm{R},\mathrm{B},\mathrm{G}\}} l(\pi'_c) + \mathcal{O}(l(s))$.

proof sketch

- apply 1- and 2-color patching to reduce to the case
- apply patching scheme (up to symmetry)



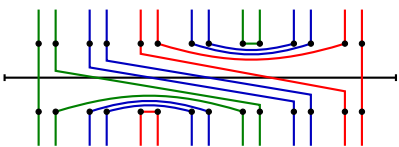
Conditional patching of three tours

Tricolored Patching Lemma (B., Disser, Feldmann, Zych-Pawlewicz).
Let $\pi_{\mathrm{R}}, \pi_{\mathrm{B}}, \pi_{\mathrm{G}}$ be non-crossing tours and s a line segment such that no red crossing is next to a green crossing (along s).
Then, there are tours $\pi'_{\mathrm{R}}, \pi'_{\mathrm{B}}, \pi'_{\mathrm{G}}$ such that

- $\pi'_{\mathrm{R}}, \pi'_{\mathrm{B}}, \pi'_{\mathrm{G}}$ intersect s in at most 18 points,
- π_c and π'_c only differ in a small neighbourhood of s ($c \in \{\mathrm{R}, \mathrm{B}, \mathrm{G}\}$),
- $\sum_{c\in\{\mathrm{R},\mathrm{B},\mathrm{G}\}} l(\pi_c) \leq \sum_{c\in\{\mathrm{R},\mathrm{B},\mathrm{G}\}} l(\pi'_c) + \mathcal{O}(l(s))$.

proof sketch

- apply 1- and 2-color patching to reduce to the case
- apply patching scheme (up to symmetry)



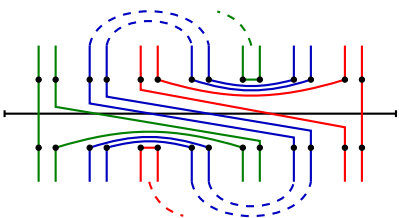
Conditional patching of three tours

Tricolored Patching Lemma (B., Disser, Feldmann, Zych-Pawlewicz).
Let $\pi_{\mathrm{R}}, \pi_{\mathrm{B}}, \pi_{\mathrm{G}}$ be non-crossing tours and s a line segment such that no red crossing is next to a green crossing (along s).
Then, there are tours $\pi'_{\mathrm{R}}, \pi'_{\mathrm{B}}, \pi'_{\mathrm{G}}$ such that

- $\pi'_{\mathrm{R}}, \pi'_{\mathrm{B}}, \pi'_{\mathrm{G}}$ intersect s in at most 18 points,
- π_c and π'_c only differ in a small neighbourhood of s ($c \in \{\mathrm{R}, \mathrm{B}, \mathrm{G}\}$),
- $\sum_{c\in\{\mathrm{R},\mathrm{B},\mathrm{G}\}} l(\pi_c) \leq \sum_{c\in\{\mathrm{R},\mathrm{B},\mathrm{G}\}} l(\pi'_c) + \mathcal{O}(l(s))$.

proof sketch

- apply 1- and 2-color patching to reduce to the case
- apply patching scheme (up to symmetry)
- argue that tours remain connected



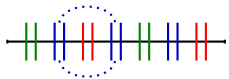
Conditional patching of three tours

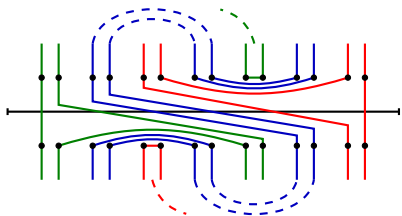
Tricolored Patching Lemma (B., Disser, Feldmann, Zych-Pawlewicz).
Let $\pi_{\mathrm{R}}, \pi_{\mathrm{B}}, \pi_{\mathrm{G}}$ be non-crossing tours and s a line segment such that no red crossing is next to a green crossing (along s).
Then, there are tours $\pi'_{\mathrm{R}}, \pi'_{\mathrm{B}}, \pi'_{\mathrm{G}}$ such that

- $\pi'_{\mathrm{R}}, \pi'_{\mathrm{B}}, \pi'_{\mathrm{G}}$ intersect s in at most 18 points,
- π_c and π'_c only differ in a small neighbourhood of s ($c \in \{\mathrm{R}, \mathrm{B}, \mathrm{G}\}$),
- $\sum_{c \in \{\mathrm{R},\mathrm{B},\mathrm{G}\}} l(\pi_c) \leq \sum_{c \in \{\mathrm{R},\mathrm{B},\mathrm{G}\}} l(\pi'_c) + \mathcal{O}(l(s))$.

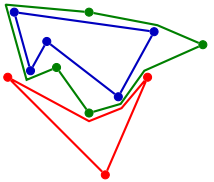
proof sketch

- apply 1- and 2-color patching to reduce to the case
- apply patching scheme (up to symmetry)
- argue that tours remain connected
  - idea: eliminate impossible connections
  - example:





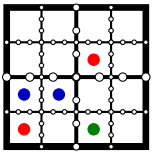
What generalizes to k-ETSP?



ingredients

- restrict to the case where terminals lie in $\mathbb{Z}^2$

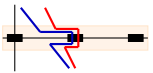
only generalizes to $k \leq 3$



1st step: place a suitable grid in $\mathbb{R}^2$ (dissection) and portals on this grid

2nd step: optimal portal-respecting solution can be computed in poly-time

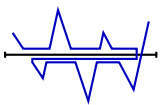
generalizes $\forall k \in \mathbb{N}$



3rd step: there is a portal-respecting solution of length $(1+\varepsilon) \cdot \mathrm{OPT}$

- 3.1: transform optimum solution such that it only crosses at portals

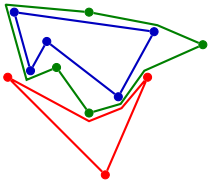
generalizes $\forall k \in \mathbb{N}$



- 3.2: transform such that portals are crossed constantly often (patching)

only generalizes to $k \leq 2$

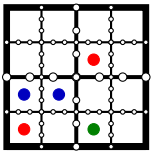
What generalizes to k-ETSP?



ingredients

- restrict to the case where terminals lie in $\mathbb{Z}^2$

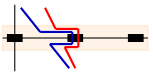
  only generalizes to $k \leq 3$



1st step: place a suitable grid in $\mathbb{R}^2$ (dissection) and portals on this grid

2nd step: optimal portal-respecting solution can be computed in poly-time

generalizes $\forall k \in \mathbb{N}$



3rd step: there is a portal-respecting solution of length $(1+\varepsilon) \cdot \mathrm{OPT}$

- 3.1: transform optimum solution such that it only crosses at portals

  generalizes $\forall k \in \mathbb{N}$



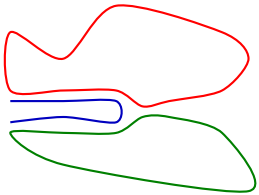
- 3.2: transform such that portals are crossed constantly often (patching)

  partly generalizes to $k = 3$

Alternative approach

non-patchable case

- let $\pi_{\mathrm{R}}, \pi_{\mathrm{G}}, \pi_{\mathrm{B}}$ be a $(1+\varepsilon)$ approximation
- assume every pair of tours "meet"
- assume $\pi_{\mathrm{R}}, \pi_{\mathrm{G}} \leq \pi_{\mathrm{B}}$, then $l(\pi_{\mathrm{R}}) + l(\pi_{\mathrm{G}}) \leq \frac{2}{3} \cdot \mathrm{OPT}$



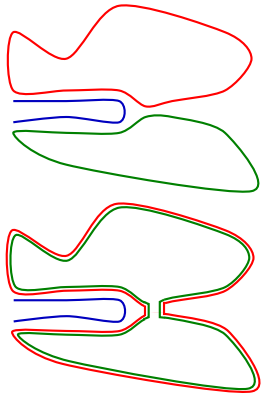
Alternative approach

non-patchable case

- let $\pi_{\mathrm{R}}, \pi_{\mathrm{G}}, \pi_{\mathrm{B}}$ be a $(1+\varepsilon)$ approximation
- assume every pair of tours "meet"
- assume $\pi_{\mathrm{R}}, \pi_{\mathrm{G}} \leq \pi_{\mathrm{B}}$, then $l(\pi_{\mathrm{R}}) + l(\pi_{\mathrm{G}}) \leq \frac{2}{3} \cdot \mathrm{OPT}$

approach

- connect and "double" red and green tour
  $\rightarrow$ solution of total length $\left(\frac{5}{3} + \mathcal{O}(\varepsilon)\right) \cdot \mathrm{OPT}$



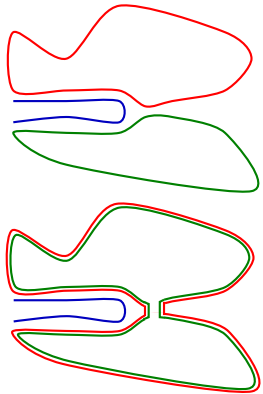
Alternative approach

non-patchable case

- let $\pi_{\mathrm{R}}, \pi_{\mathrm{G}}, \pi_{\mathrm{B}}$ be a $(1+\varepsilon)$ approximation
- assume every pair of tours "meet"
- assume $\pi_{\mathrm{R}}, \pi_{\mathrm{G}} \leq \pi_{\mathrm{B}}$, then $l(\pi_{\mathrm{R}}) + l(\pi_{\mathrm{G}}) \leq \frac{2}{3} \cdot \mathrm{OPT}$

approach

- connect and "double" red and green tour
  $\rightarrow$ solution of total length $\left(\frac{5}{3} + \mathcal{O}(\varepsilon)\right) \cdot \mathrm{OPT}$
- merging R/G points, solving 2-ETSP, "doubling" R/G-tour gives $\left(\frac{5}{3} + \mathcal{O}(\varepsilon)\right)$-approximation



Alternative approach

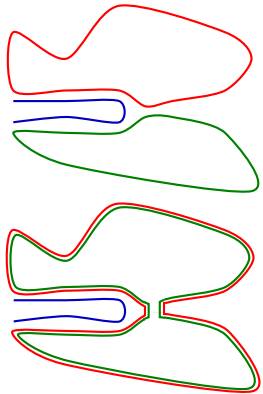
non-patchable case

- let $\pi_{\mathrm{R}}, \pi_{\mathrm{G}}, \pi_{\mathrm{B}}$ be a $(1+\varepsilon)$ approximation
- assume every pair of tours "meet"
- assume $\pi_{\mathrm{R}}, \pi_{\mathrm{G}} \leq \pi_{\mathrm{B}}$, then $l(\pi_{\mathrm{R}}) + l(\pi_{\mathrm{G}}) \leq \frac{2}{3} \cdot \mathrm{OPT}$

approach

- connect and "double" red and green tour
  $\rightarrow$ solution of total length $\left(\frac{5}{3} + \mathcal{O}(\varepsilon)\right) \cdot \mathrm{OPT}$
- merging R/G points, solving 2-ETSP, "doubling" R/G-tour gives $\left(\frac{5}{3} + \mathcal{O}(\varepsilon)\right)$-approximation

final algorithm



Alternative approach

non-patchable case

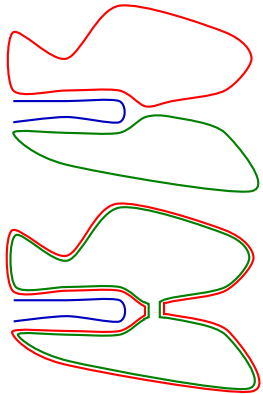
- let $\pi_{\mathrm{R}}, \pi_{\mathrm{G}}, \pi_{\mathrm{B}}$ be a $(1+\varepsilon)$ approximation
- assume every pair of tours "meet"
- assume $\pi_{\mathrm{R}}, \pi_{\mathrm{G}} \leq \pi_{\mathrm{B}}$, then $l(\pi_{\mathrm{R}}) + l(\pi_{\mathrm{G}}) \leq \frac{2}{3} \cdot \mathrm{OPT}$

approach

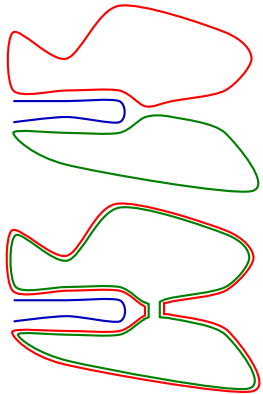
- connect and "double" red and green tour
  $\rightarrow$ solution of total length $\left(\frac{5}{3} + \mathcal{O}(\varepsilon)\right) \cdot \mathrm{OPT}$
- merging R/G points, solving 2-ETSP, "doubling" R/G-tour gives $\left(\frac{5}{3} + \mathcal{O}(\varepsilon)\right)$-approximation

final algorithm

- compute optimal portal-respecting solution



Alternative approach



non-patchable case

- let $\pi_{\mathrm{R}}, \pi_{\mathrm{G}}, \pi_{\mathrm{B}}$ be a $(1+\varepsilon)$ approximation
- assume every pair of tours "meet"
- assume $\pi_{\mathrm{R}}, \pi_{\mathrm{G}} \leq \pi_{\mathrm{B}}$, then $l(\pi_{\mathrm{R}}) + l(\pi_{\mathrm{G}}) \leq \frac{2}{3} \cdot \mathrm{OPT}$

approach

- connect and "double" red and green tour
  $\rightarrow$ solution of total length $\left(\frac{5}{3} + \mathcal{O}(\varepsilon)\right) \cdot \mathrm{OPT}$
- merging R/G points, solving 2-ETSP, "doubling" R/G-tour gives $\left(\frac{5}{3} + \mathcal{O}(\varepsilon)\right)$-approximation

final algorithm

- compute optimal portal-respecting solution
- for every pair of colors, merge colors and use PTAS for 2-ETSP

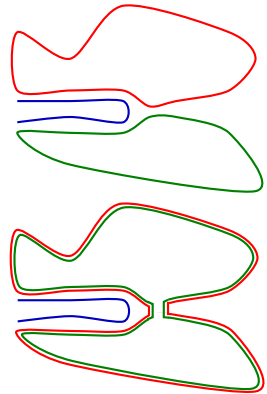
Alternative approach

non-patchable case

- let $\pi_{\mathrm{R}}, \pi_{\mathrm{G}}, \pi_{\mathrm{B}}$ be a $(1+\varepsilon)$ approximation
- assume every pair of tours "meet"
- assume $\pi_{\mathrm{R}}, \pi_{\mathrm{G}} \leq \pi_{\mathrm{B}}$, then $l(\pi_{\mathrm{R}}) + l(\pi_{\mathrm{G}}) \leq \frac{2}{3} \cdot \mathrm{OPT}$

approach

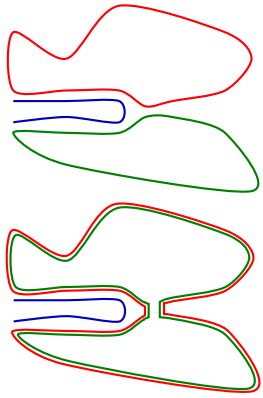
- connect and "double" red and green tour
  $\rightarrow$ solution of total length $\left(\frac{5}{3} + \mathcal{O}(\varepsilon)\right) \cdot \mathrm{OPT}$
- merging R/G points, solving 2-ETSP, "doubling" R/G-tour gives $\left(\frac{5}{3} + \mathcal{O}(\varepsilon)\right)$-approximation



final algorithm

- compute optimal portal-respecting solution
- for every pair of colors, merge colors and use PTAS for 2-ETSP
- output shortest solution

Alternative approach



non-patchable case

- let π_R, π_G, π_B be a $(1+\varepsilon)$ approximation
- assume every pair of tours "meet"
- assume $\pi_R, \pi_G \leq \pi_B$, then $l(\pi_R) + l(\pi_G) \leq \frac{2}{3} \cdot \mathrm{OPT}$

approach

- connect and "double" red and green tour
  $\rightarrow$ solution of total length $\left(\frac{5}{3} + \mathcal{O}(\varepsilon)\right) \cdot \mathrm{OPT}$
- merging R/G points, solving 2-ETSP, "doubling" R/G-tour gives $\left(\frac{5}{3} + \mathcal{O}(\varepsilon)\right)$-approximation

final algorithm

- compute optimal portal-respecting solution
  $\rightarrow$ gives $(1+\varepsilon)$-approximation if OPT is patchable
- for every pair of colors, merge colors and use PTAS for 2-ETSP
- output shortest solution

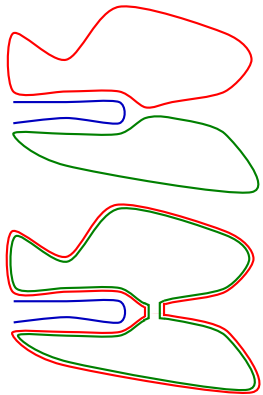
Alternative approach

non-patchable case

- let π_R, π_G, π_B be a $(1+\varepsilon)$ approximation
- assume every pair of tours "meet"
- assume $\pi_R, \pi_G \leq \pi_B$, then $l(\pi_R) + l(\pi_G) \leq \frac{2}{3} \cdot \mathrm{OPT}$

approach

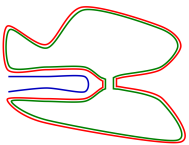
- connect and "double" red and green tour
  $\rightarrow$ solution of total length $\left(\frac{5}{3} + \mathcal{O}(\varepsilon)\right) \cdot \mathrm{OPT}$
- merging R/G points, solving 2-ETSP, "doubling" R/G-tour gives $\left(\frac{5}{3} + \mathcal{O}(\varepsilon)\right)$-approximation



final algorithm

- compute optimal portal-respecting solution
  $\rightarrow$ gives $(1+\varepsilon)$-approximation if OPT is patchable
- for every pair of colors, merge colors and use PTAS for 2-ETSP
  $\rightarrow$ gives $(5/3+\varepsilon)$-approximation if OPT is non-patchable
- output shortest solution

Summary



$\left(\frac{5}{3}+\varepsilon\right)$-approximation for 3-ETSP

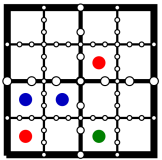
optimum patchable?

no

yes

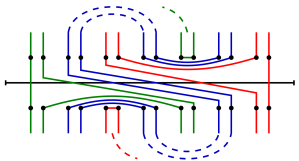
alternative approach via merging colors

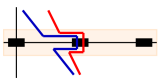
generalized Arora



dissection

multipath problem via dynamic programming





conditional patching for 3 tours

existence of portal-resp. solution

snapping crossings to portals