Traveling Salesman Problem

Approximation Algorithms

Juan Da Silva - Ruprecht Karls Universität Heidelberg

Quotes of the day

“Problem solving is hunting. It is savage pleasure and we are born to it.”

Thomas Harris

“Algorithms must be seen to be believed.”

Donald Knuth

Outline

- Metric TSP
  - Double-Tree Algorithm
  - Christofides Algorithm
- Euclidean TSP
  - Arora’s Algorithm
- Real-world application

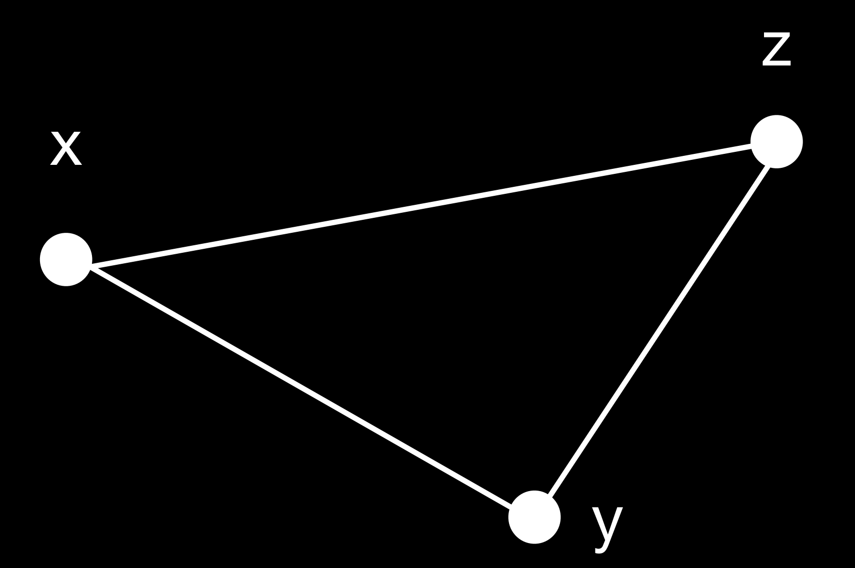
Goals of this talk

- Understand presented approximation algorithms
- Implement presented algorithms to solve real world problems

Metric TSP

$d(\cdot,\cdot) : X \times X \to \mathbb{R}, d(x,y) \mapsto \|x-y\|$

- $d(x,y) \geq 0$, for $d(x,y)=0 \Rightarrow x=y$
- $d(x,y) = d(y,x)$
- $d(x,z) \leq d(x,y) + d(y,z)$



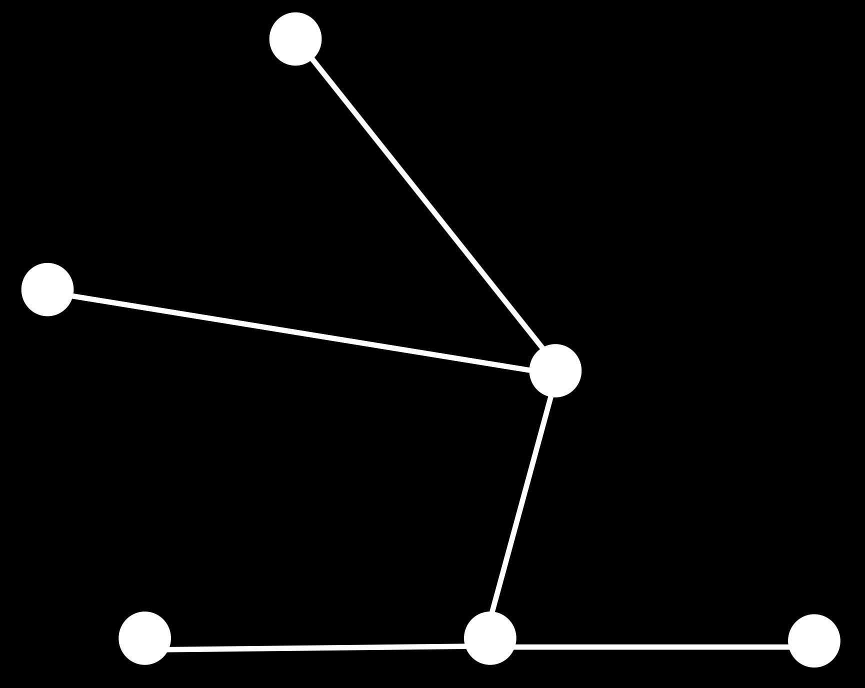
<u>Instance:</u> A complete graph G with weights $c : E(G) \to \mathbb{R}_+$ such that metric properties hold

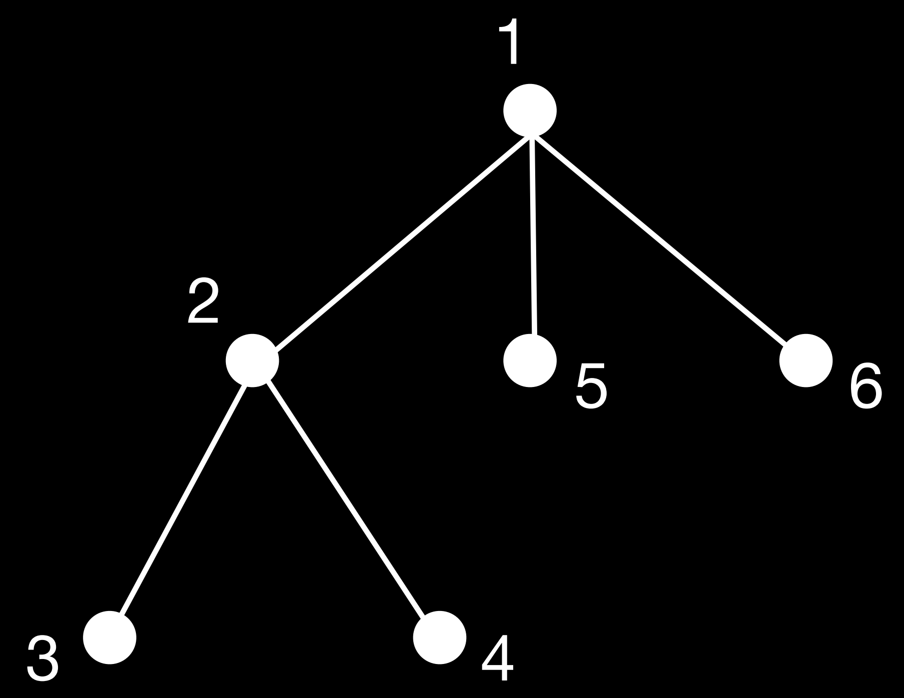
<u>Goal:</u> Find Hamiltonian cycle in G of minimum weight

Theorem 1. The Metric TSP is NP-hard

Approximation algorithms

Double-Tree Algorithm



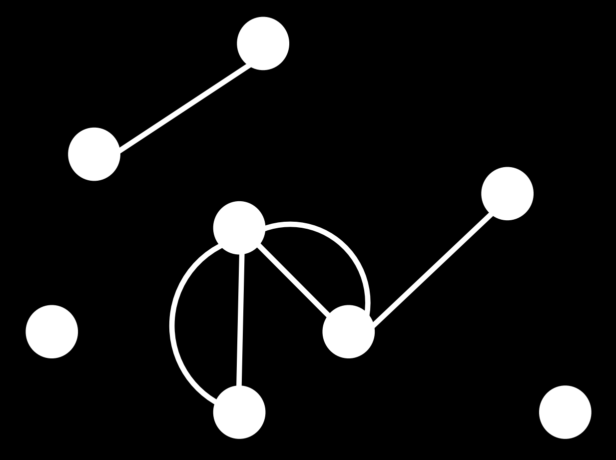


DFS traversal: 1, 2, 3, 2, 4, 2, 1, 5, 1, 6, 1

Instance: An instance (G, c) of the Metric TSP

Goal: A cycle

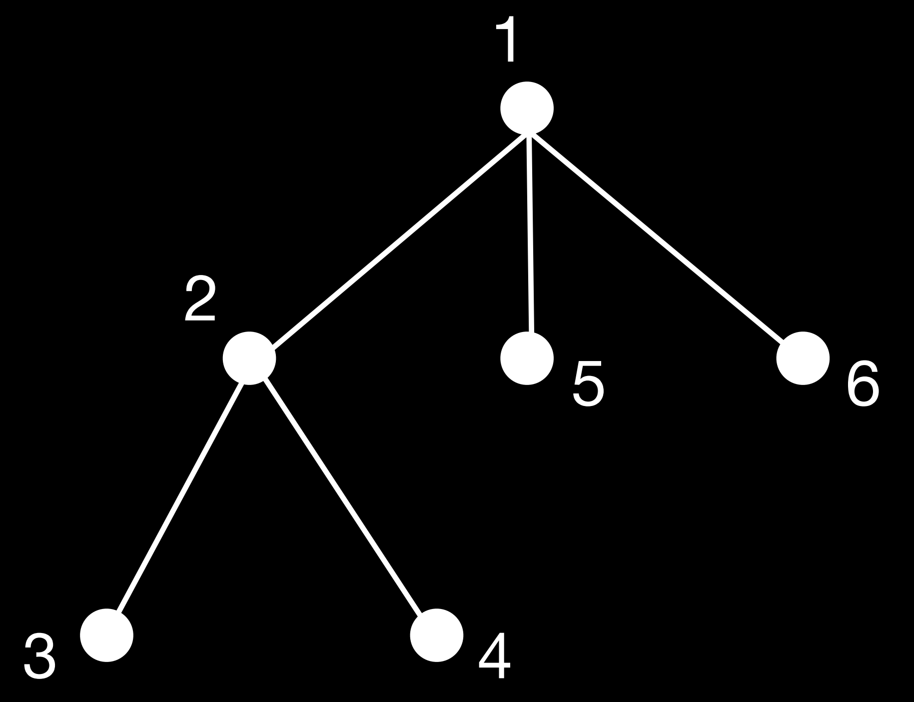
Theorem 2. The Double-Tree Algorithm is a 2-factor approximation algorithm for the Metric TSP



S: (Multi)-set of edges
c(S): Sum of cost of all edges in S
$c(H_G^*)$: cost of optimal Hamiltonian cycle

Approximation algorithms

Double-Tree Algorithm



DFS traversal: 1, 2, 3, 2, 4, 2, 1, 5, 1, 6, 1

DFS traversal: 1, 2, 3, ~~2,~~ 4, ~~2,~~ ~~1,~~ 5, ~~1,~~ 6, ~~1~~

$$T \longrightarrow G \longrightarrow G'$$

$2 \cdot c(T) = c(G)$ $\quad$ $c(G') \leq c(G)$

(1) $\Rightarrow c(G') \leq 2 \cdot c(T)$

<u>Instance:</u> An instance (G, c) of the Metric TSP

Goal: A cycle

Theorem 2. The Double-Tree Algorithm is a 2-factor approximation algorithm for the Metric TSP

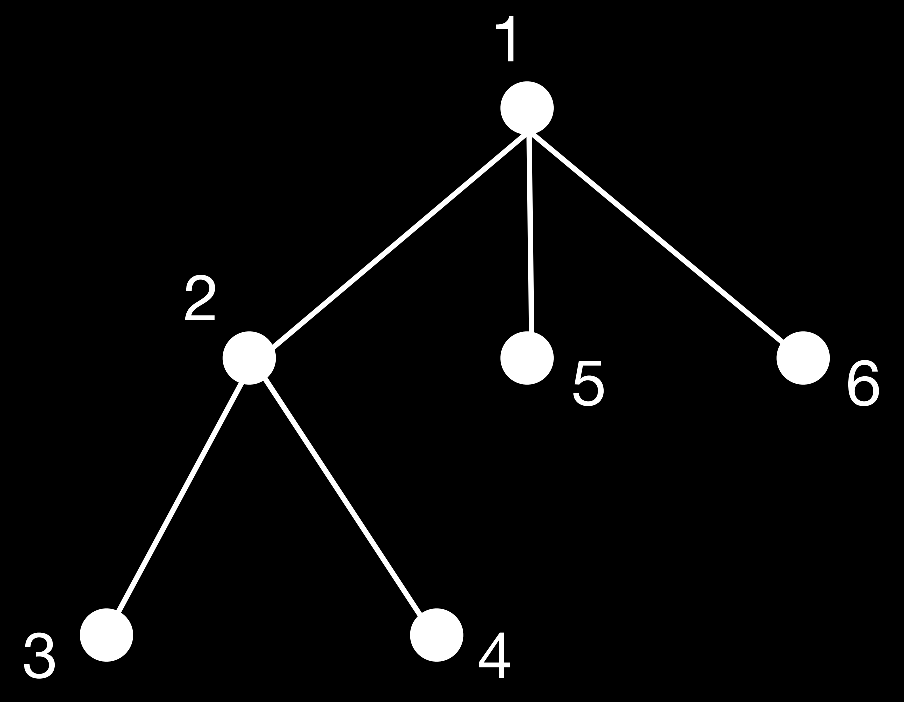
$1 \rightarrow 2 \rightarrow 3 \rightarrow 2 \rightarrow 4 \rightarrow \ldots$

$1 \rightarrow 2 \rightarrow 3 \rightarrow 2 \rightarrow 4 \rightarrow \ldots$

- $d(x,z) \leq d(x,y) + d(y,z)$

Approximation algorithms

Double-Tree Algorithm



DFS traversal: 1, 2, 3, 2, 4, 2, 1, 5, 1, 6, 1

DFS traversal: 1, 2, 3, ~~2,~~ 4, ~~2,~~ ~~1,~~ 5, ~~1,~~ 6, ~~1~~

$$T \longrightarrow G \longrightarrow G'$$

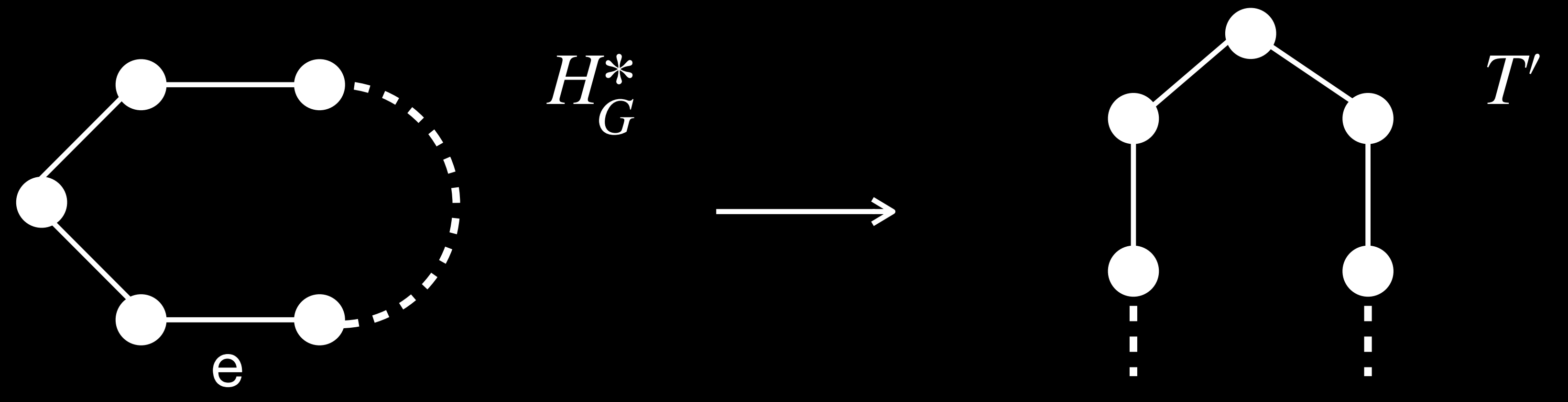
$2 \cdot c(T) = c(G)$ $\quad$ $c(G') \leq c(G)$

(1) $\Rightarrow c(G') \leq 2 \cdot c(T)$

<u>Instance:</u> An instance (G, c) of the Metric TSP

Goal: A cycle

Theorem 2. The Double-Tree Algorithm is a 2-factor approximation algorithm for the Metric TSP



(2) $c(H_G^*) \geq c(H_G^* - e) \geq c(T)$

$\Longrightarrow$ From (1) and (2) $c(G') \leq 2 \cdot c(H_G^*)$

Approximation algorithms

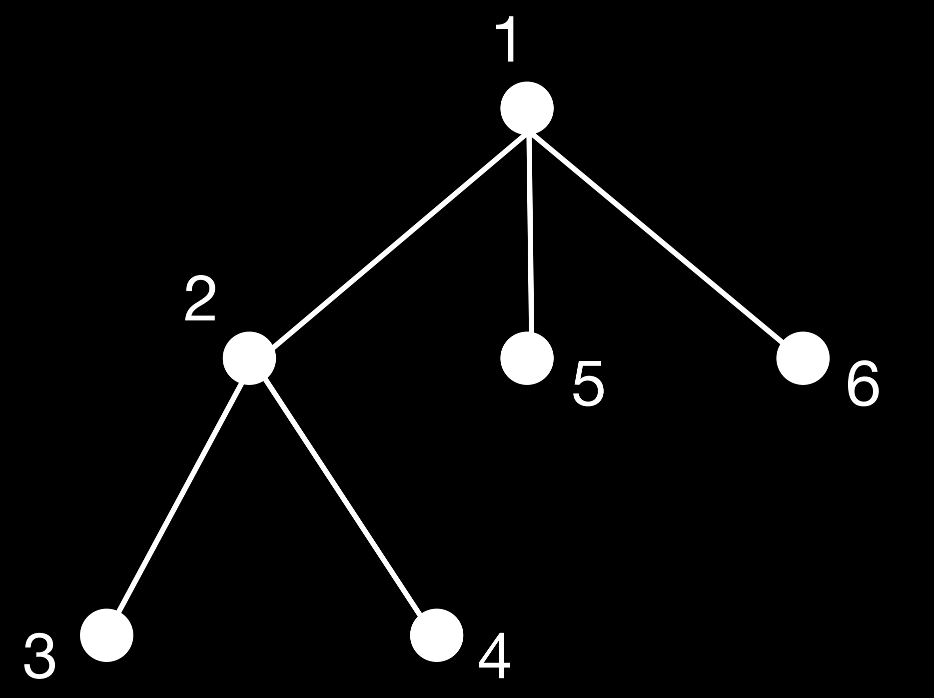
Double-Tree Algorithm - Recap

<u>Instance:</u> An instance (G, c) of the Metric TSP

Goal: A cycle

Theorem 2. The Double-Tree Algorithm is a 2-factor approximation algorithm for the Metric TSP

1.



DFS traversal: 1, 2, 3, 2, 4, 2, 1, 5, 1, 6, 1

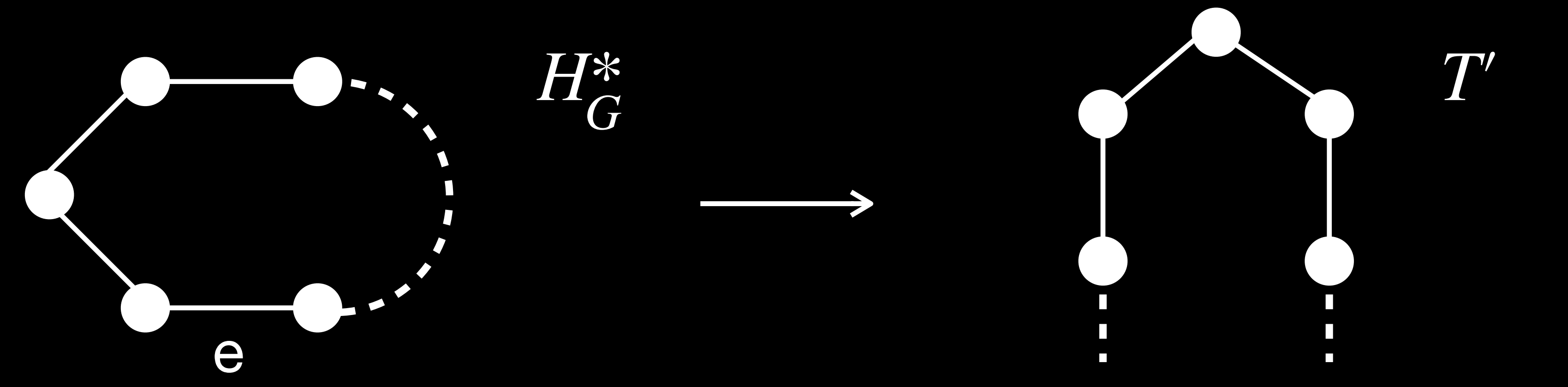
2. DFS traversal: 1, 2, 3, ~~2,~~ 4, ~~2,~~ ~~1,~~ 5, ~~1,~~ 6, ~~1~~

3. $$T \longrightarrow G \longrightarrow G'$$

$2 \cdot c(T) = c(G)$ $\quad c(G') \leq c(G)$

(1) $\Rightarrow c(G') \leq 2 \cdot c(T)$

4.

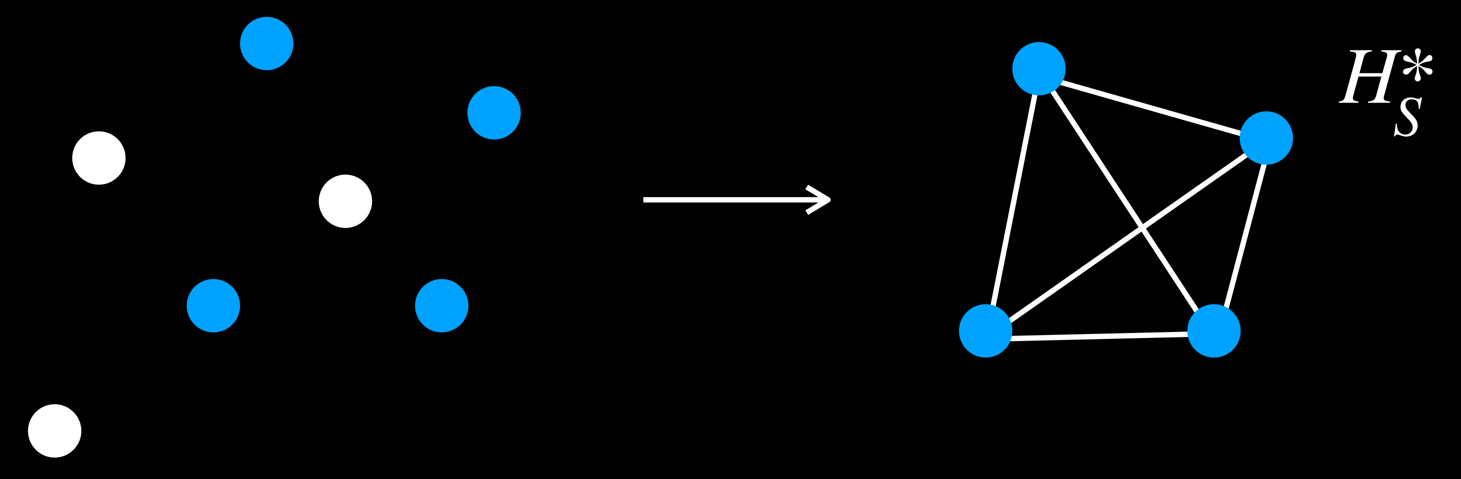


(2) $c(H_G^*) \geq c(H_G^* - e) \geq c(T)$

$\Longrightarrow$ From (1) and (2) $c(G') \leq 2 \cdot c(H_G^*)$

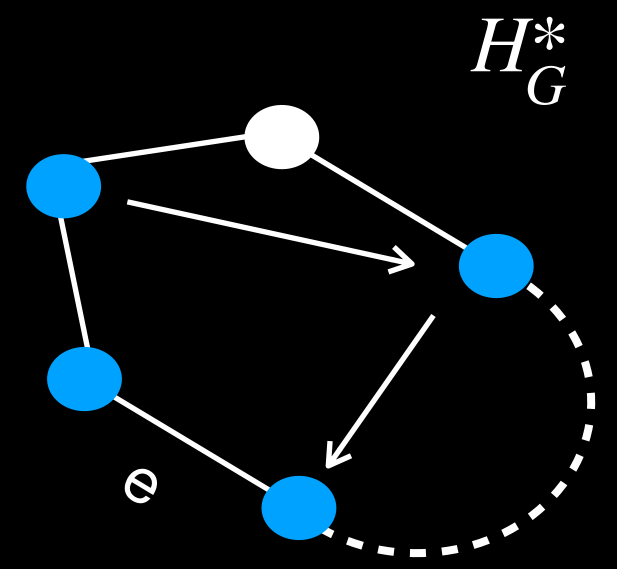
Some useful concepts and lemmas

$S \subseteq V$

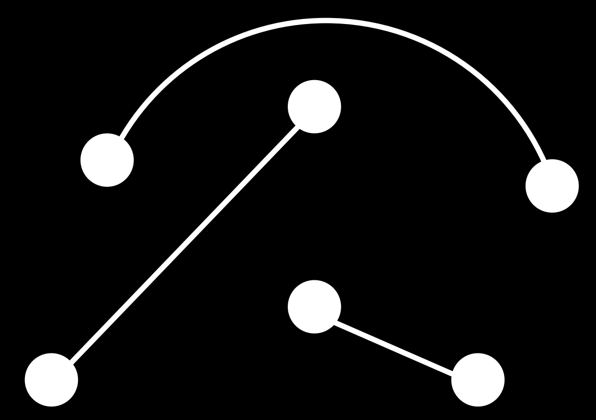


Claim: $c(H_S^*) \leq c(H_G^*)$

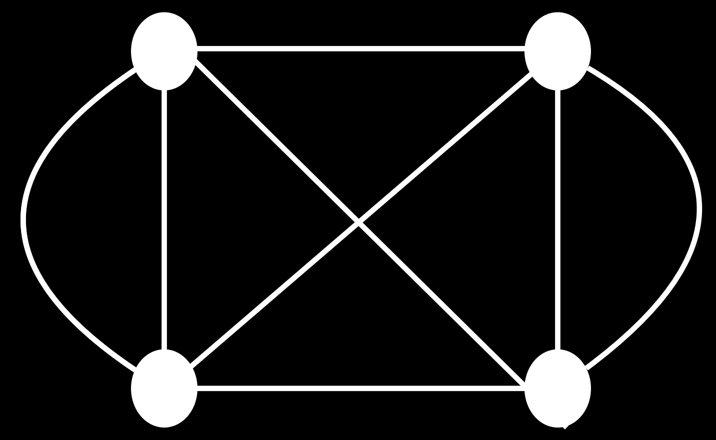
Say $c(H_S^*) > c(H_G^*)$



Perfect Matchings

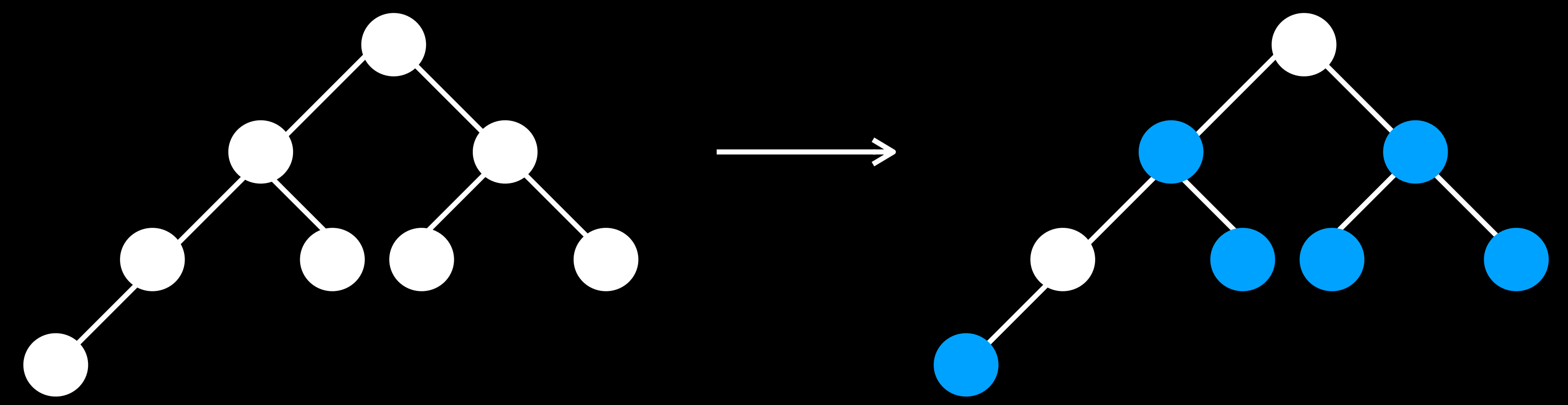


Eulerian Graph

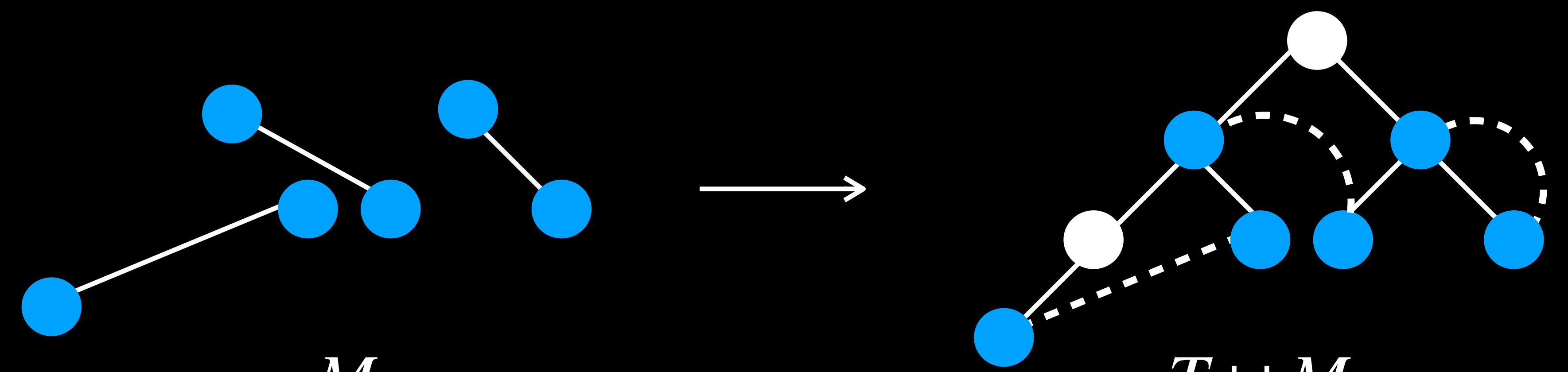


Approximation algorithms

Christofides' Algorithm



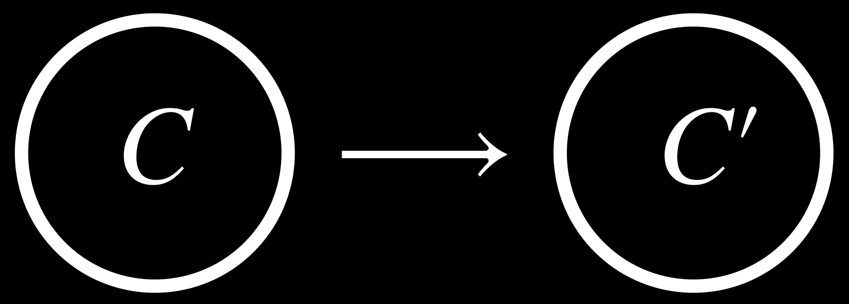
T



M $\quad$ $T \cup M$

$c(C) = c(T) + c(M)$

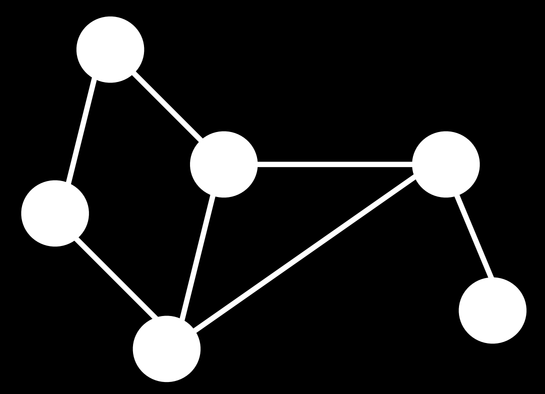
Duplication argument:



<u>Instance:</u> An instance (G, c) of the Metric TSP

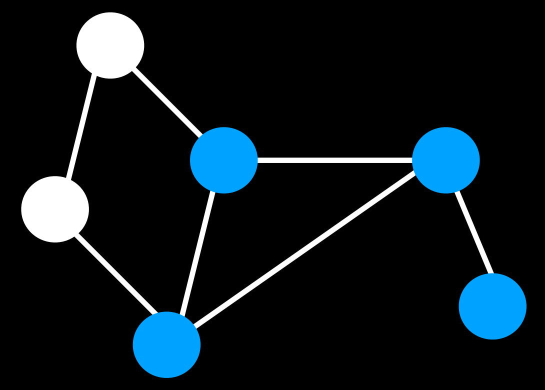
Goal: A cycle

Theorem 3. Christofides' Algorithm is a 3/2-factor approximation algorithm for the Metric TSP



$$\sum_{v_i \in V} d_i = 2 \cdot |E|$$

$\sum_{v_i \in V}^{r} d_i$ is even, $\forall i = 1,\ldots,r : v_i$ has even degree



$$\sum_{v_i \in V} d_i = \sum_{v_i \in V:\ v_i \text{ even}} d_i + \sum_{v_i \in V:\ v_i \text{ odd}} d_i$$

$\Longrightarrow$ Number of vertices with odd degree (number of terms in $\sum_{v_i \in V:\ v_i \text{ odd}} d_i$) must be even

Approximation algorithms

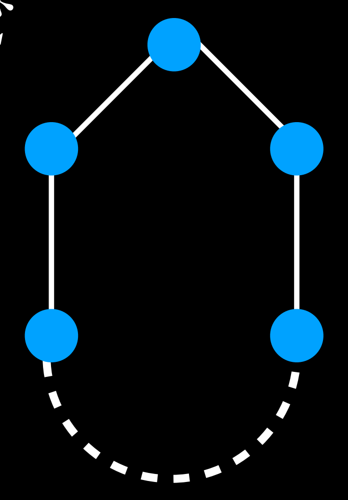
Christofides' Algorithm

$c(C) = c(T) + c(M)$

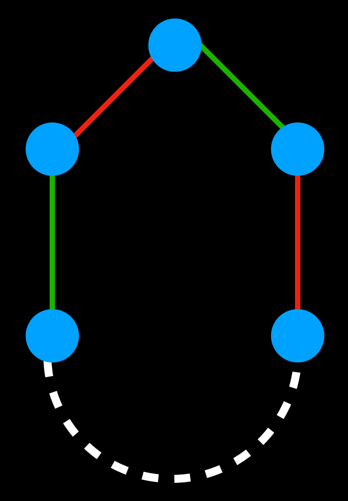
Duplication argument: $C \longrightarrow C'$

From Double-Tree algorithm: $c(T) \leq c(H_G^*)$

H_S^*



From previous lemma: $c(H_S^*) \leq c(H_G^*)$



M_1 $\quad c(M) \leq c(M_1)$

M_2 $\quad c(M) \leq c(M_2)$

<u>Instance:</u> An instance (G, c) of the Metric TSP

Goal: A cycle

Theorem 3. Christofides' Algorithm is a 3/2-factor approximation algorithm for the Metric TSP

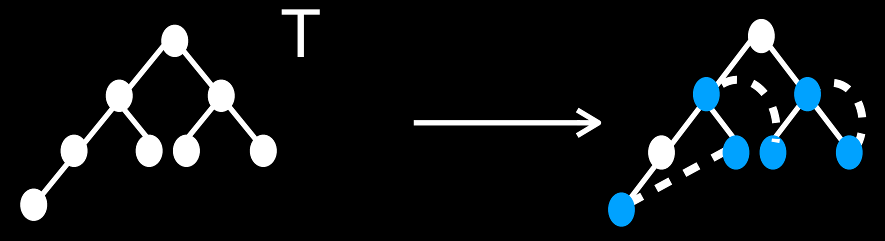
$$\begin{aligned} \Rightarrow c(M) &\leq \frac{1}{2}(c(M_1) + c(M_2)) \\ &= \frac{1}{2}c(H_S^*) \\ &\leq \frac{1}{2}c(H_G^*) \end{aligned}$$

$$\begin{aligned} c(C) &= c(T) + c(M) \\ &\leq c(H_G^*) + \frac{1}{2}c(H_G^*) \end{aligned}$$

$$\Longrightarrow c(C') \leq \frac{3}{2}c(H_G^*)$$

Approximation algorithms

Christofides' Algorithm - Recap

1. 

2. $C \longrightarrow C'$ Duplication argument

3. Claim: $c(C) = c(T) + c(M)$

4. M bounded by H_S^* $\to M_1$, $\to M_2$

<u>Instance:</u> An instance (G, c) of the Metric TSP

Goal: A cycle

Theorem 3. Christofides' Algorithm is a 3/2-factor approximation algorithm for the Metric TSP

5.

$$\begin{aligned} c(M) &\leq \frac{1}{2}(c(M_1) + c(M_2)) \\ &= \frac{1}{2}c(H_S^*) \\ &\leq c(H_G^*) \end{aligned}$$

$$\begin{aligned} c(C) &= c(T) + c(M) \\ &\leq c(H_G^*) + \frac{1}{2}c(H_G^*) \end{aligned}$$

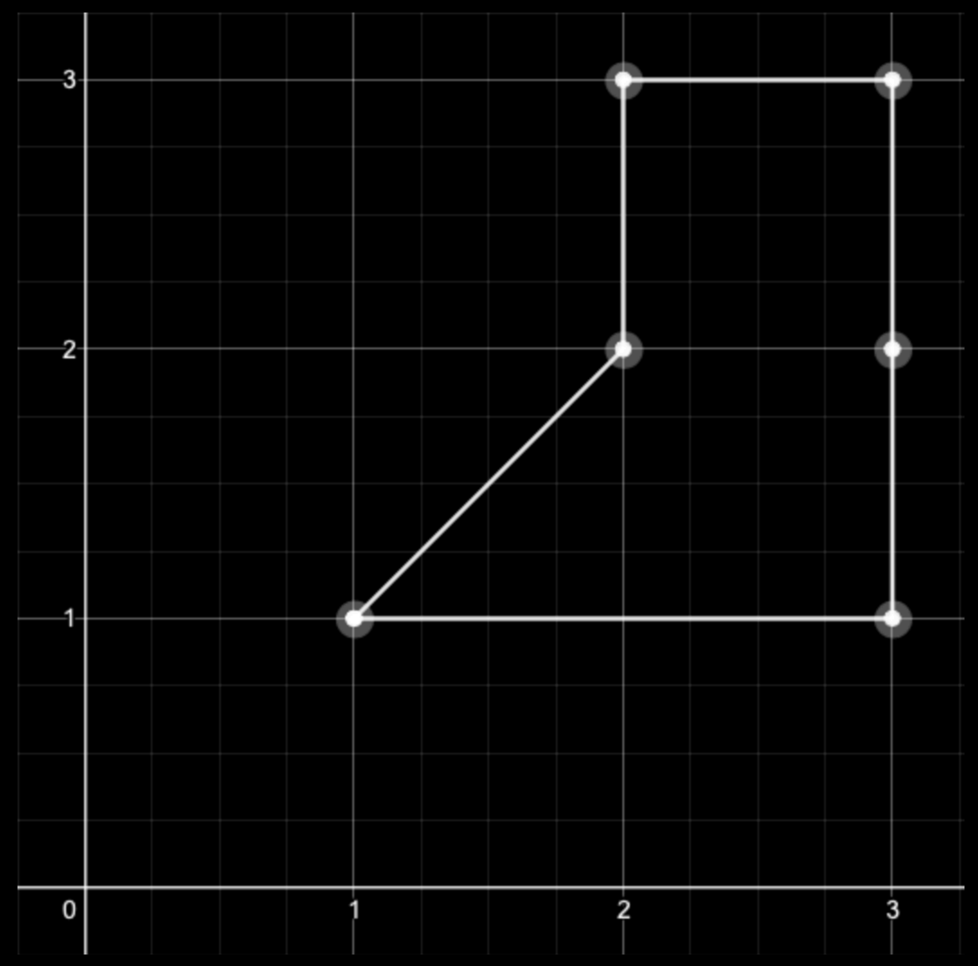
$$\implies c(C') \leq \frac{3}{2}c(H_G^*)$$

Up to this point

Euclidean TSP

$d(\cdot,\cdot): X \times X \to \mathbb{R}, d(x,y) \mapsto \|x-y\|$

- $d(x,y) \geq 0$, for $d(x,y) = 0 \Rightarrow x = y$
- $d(x,y) = d(y,x)$
- $d(x,z) \leq d(x,y) + d(y,z)$



<u>Instance:</u> Finite set $V \subseteq \mathbb{R}^2, |V| > 3$ with euclidean distances, i.e., $d(x,y) = \|x-y\|$

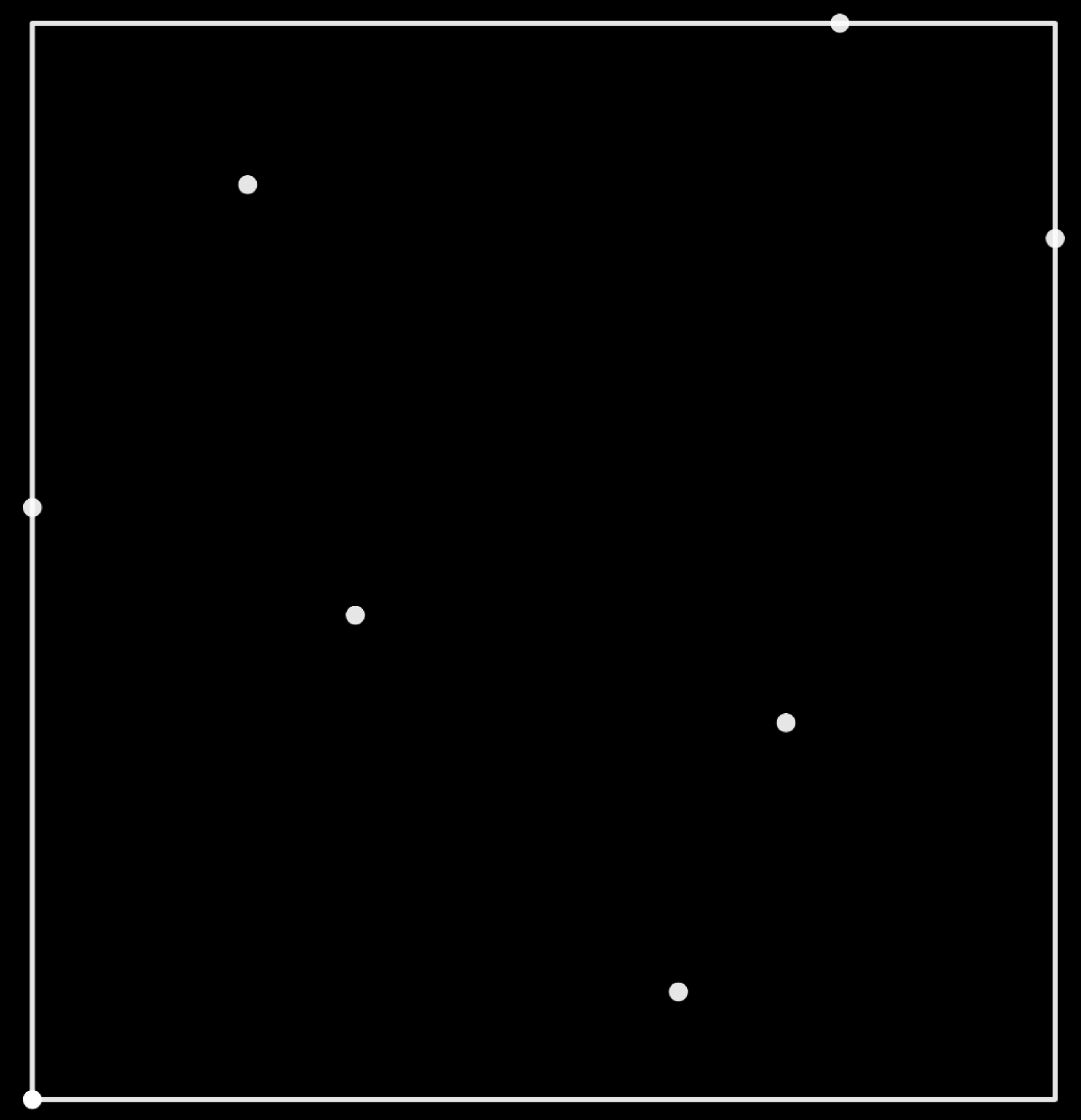
<u>Goal:</u> Find Hamiltonian cycle in G of minimum weight

Theorem 4. The Euclidean TSP is NP-hard

Some useful definitions and lemmas

Definition. (ϵ-nice instances): An instance of euclidean TSP is called ϵ-nice if the following conditions hold:

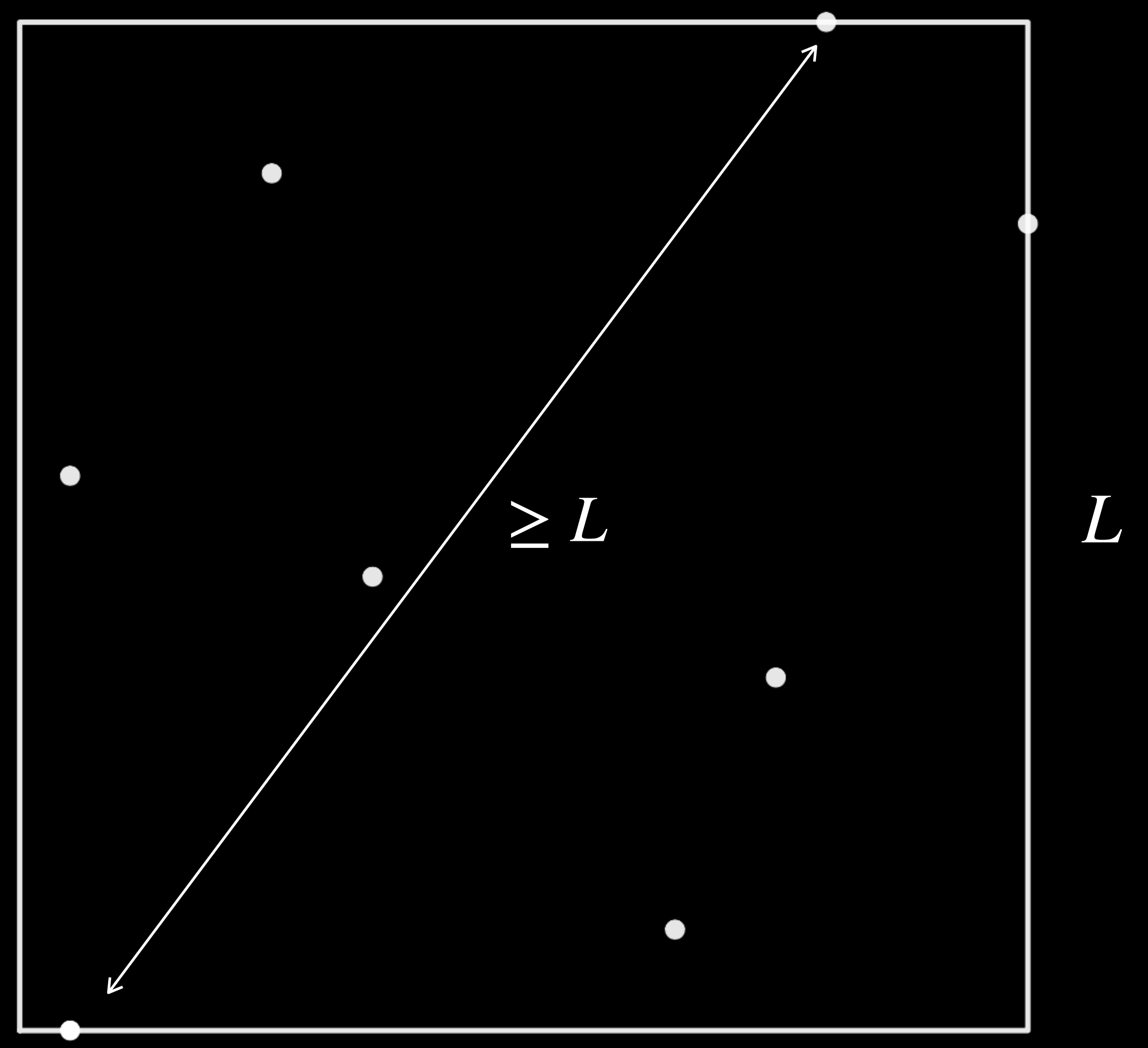
- Every point has integral coordinates in the interval $\left[0,\ O\left(\frac{n}{\epsilon}\right)\right]^2$
- Any two different points have distances at least 4.



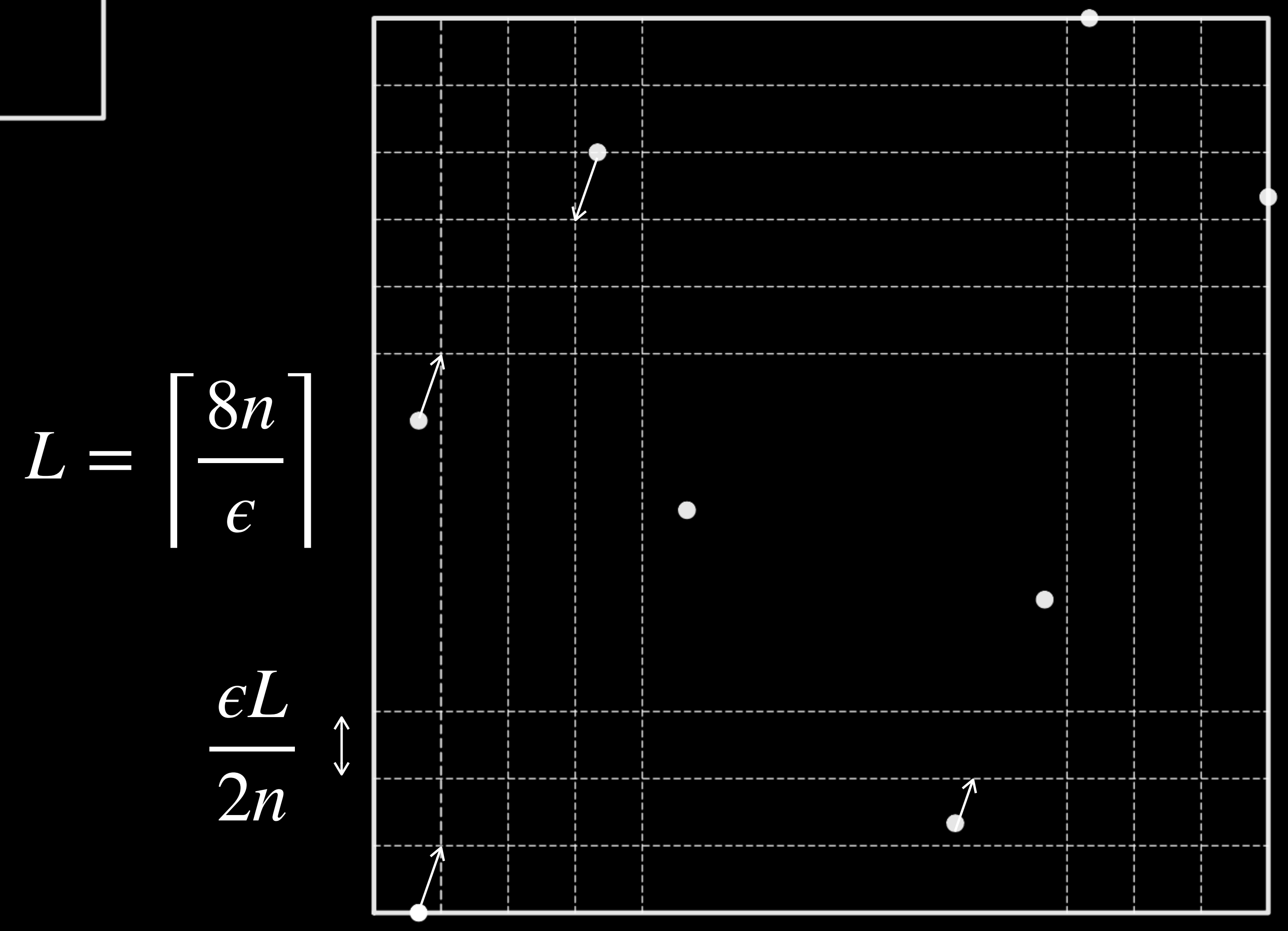
$$L \longrightarrow L = \left\lceil \frac{8n}{\epsilon} \right\rceil$$

Some useful definitions and lemmas

Arora's Algorithm



$OPT_I \geq 2L \geq L$



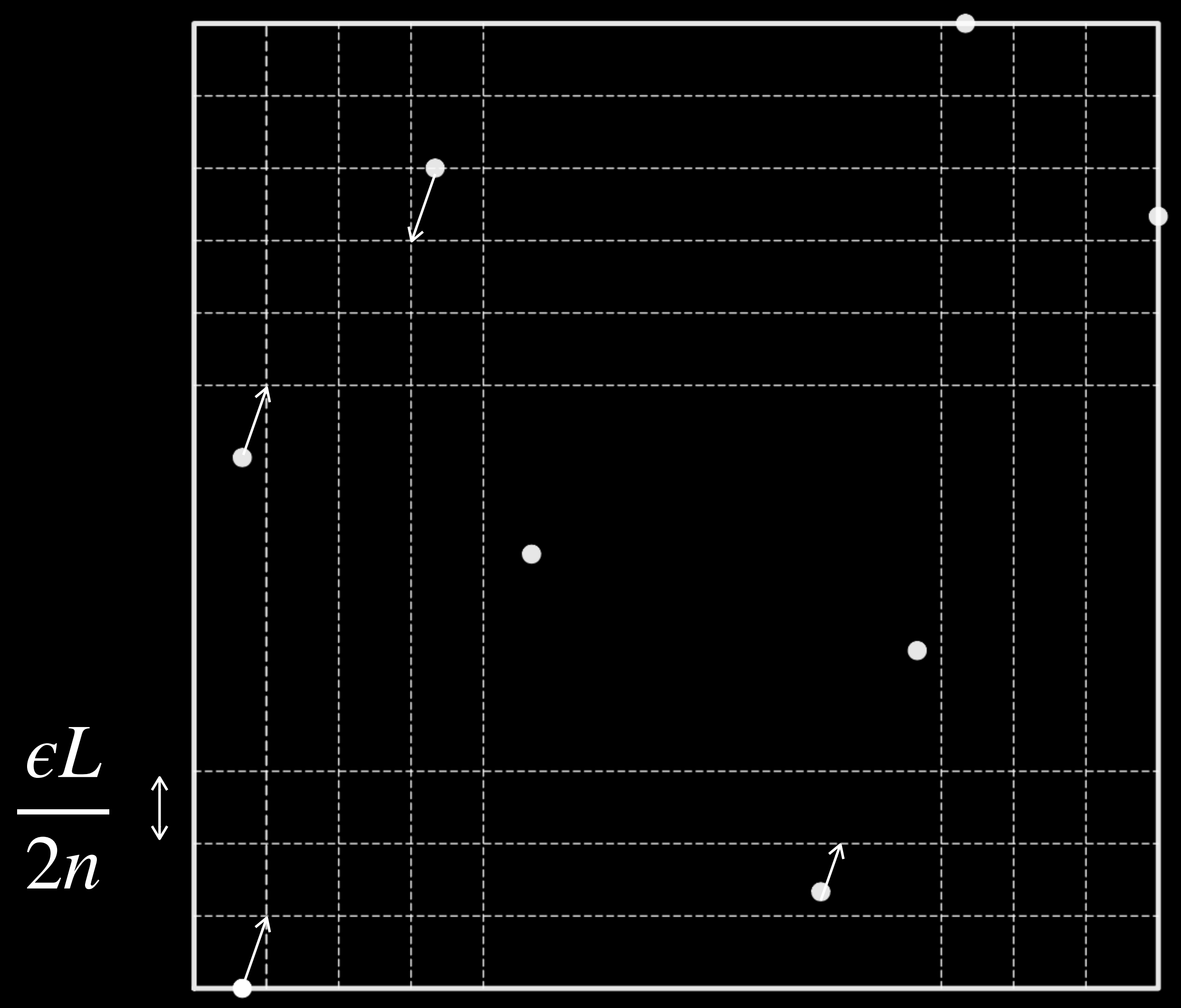
Lemma. Let I be an arbitrary instance to euclidean TSP. Let OPT_I denote the length of the optimum tour in I. We can transform I into an ϵ-nice instance I' such that $OPT_{I'} \leq (1+\epsilon)OPT_I$

- Every point has integral coordinates in the respective range since $L = \lceil 8n/\epsilon \rceil \in O(n/\epsilon)$
- Grid spacing is $\frac{\epsilon L}{2n} \geq \frac{\epsilon}{2n}\frac{8n}{\epsilon} \Rightarrow d(x,y) \geq 4$

$\Longrightarrow I'$ is ϵ-nice

Some useful definitions and lemmas

Arora's Algorithm



$$L = \left\lceil \frac{8n}{\epsilon} \right\rceil$$

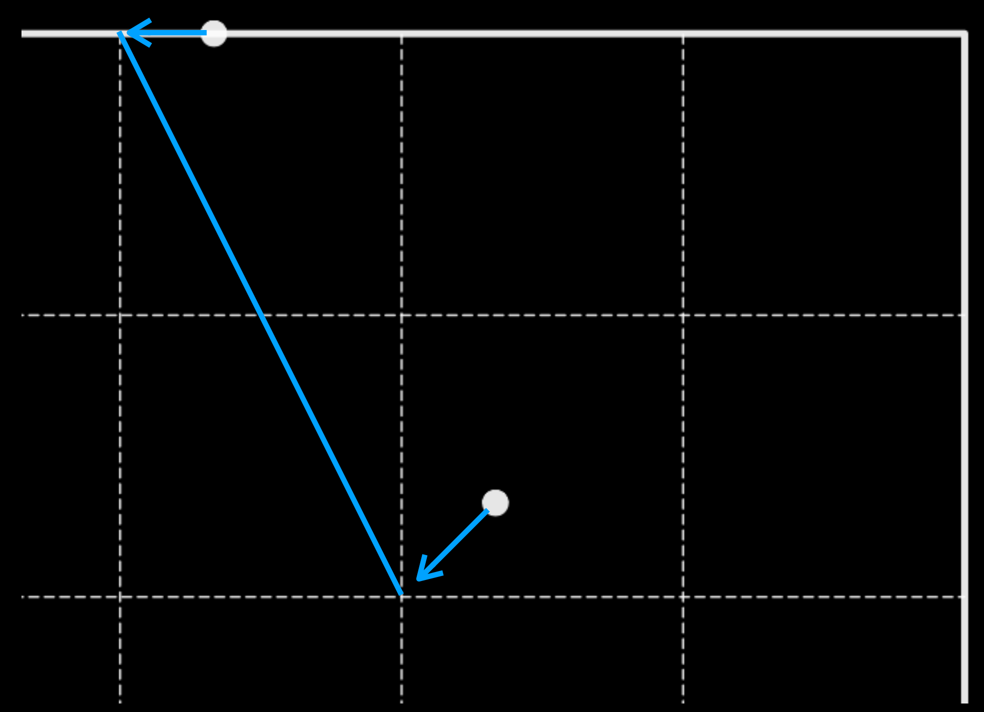
Mapping points from I to I': distance $\leq \frac{\epsilon L}{2n}$

$\Rightarrow$ Mapping edges from I to I': length edge $\leq \frac{\epsilon L}{n}$

Lemma. Let I be an arbitrary instance to euclidean TSP. Let OPT_I denote the length of the optimum tour in I. We can transform I into an ϵ-nice instance I' such that $OPT_{I'} \leq (1+\epsilon)OPT_I$

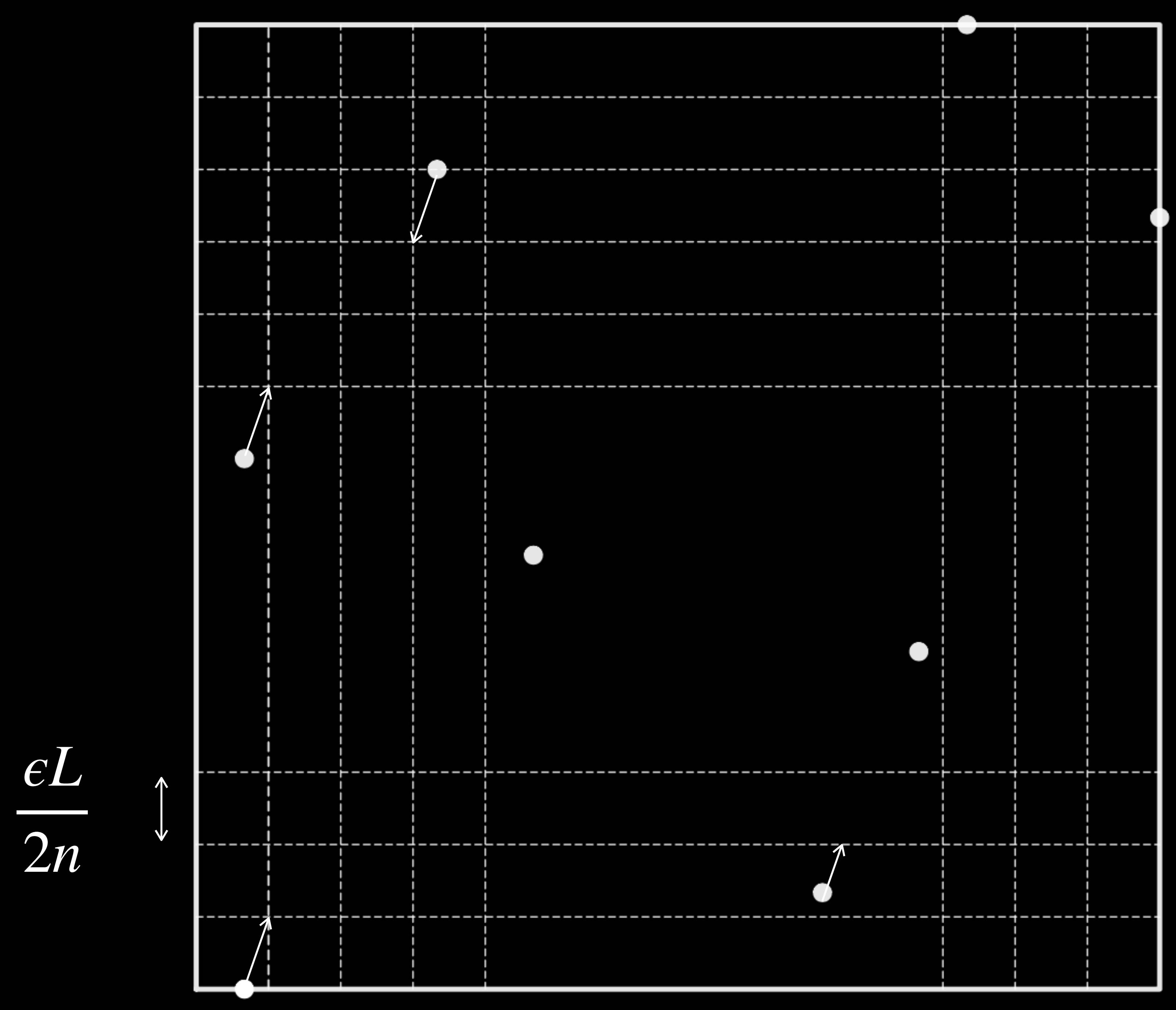
$$c^2 = \left(\frac{\epsilon L}{2n}\right)^2 + 2\left(\frac{\epsilon L}{2n}\right)^2$$

$$\Leftrightarrow c = \sqrt{3}\left(\frac{\epsilon L}{2n}\right)^2 \leq \frac{\epsilon L}{n}$$



Some useful definitions and lemmas

Arora's Algorithm



$$L = \left\lceil \frac{8n}{\epsilon} \right\rceil$$

Lemma. Let I be an arbitrary instance to euclidean TSP. Let OPT_I denote the length of the optimum tour in I. We can transform I into an ϵ-nice instance I' such that $OPT_{I'} \leq (1+\epsilon)OPT_I$

Mapping points from I to I': distance $\leq \frac{\epsilon L}{2n}$

$\Rightarrow$ Mapping edges from I to I': length edge $\leq \frac{\epsilon L}{n}$

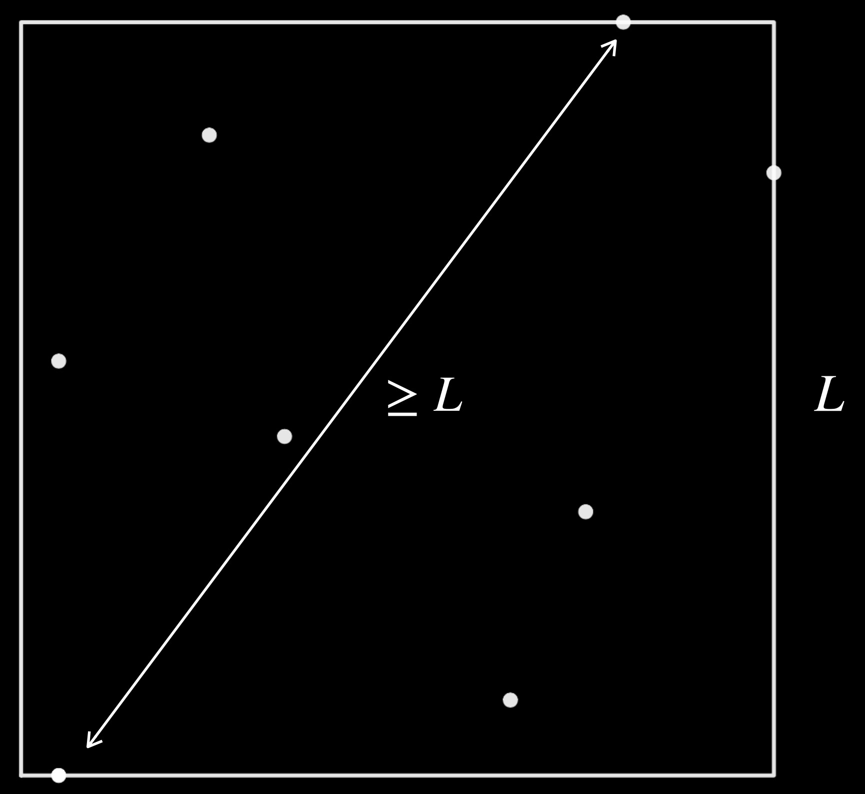
$\Rightarrow OPT_{I'} \leq OPT_I + \epsilon L$

$\implies OPT_{I'} \leq OPT_I + \epsilon L \leq (1+\epsilon)\ OPT_I$

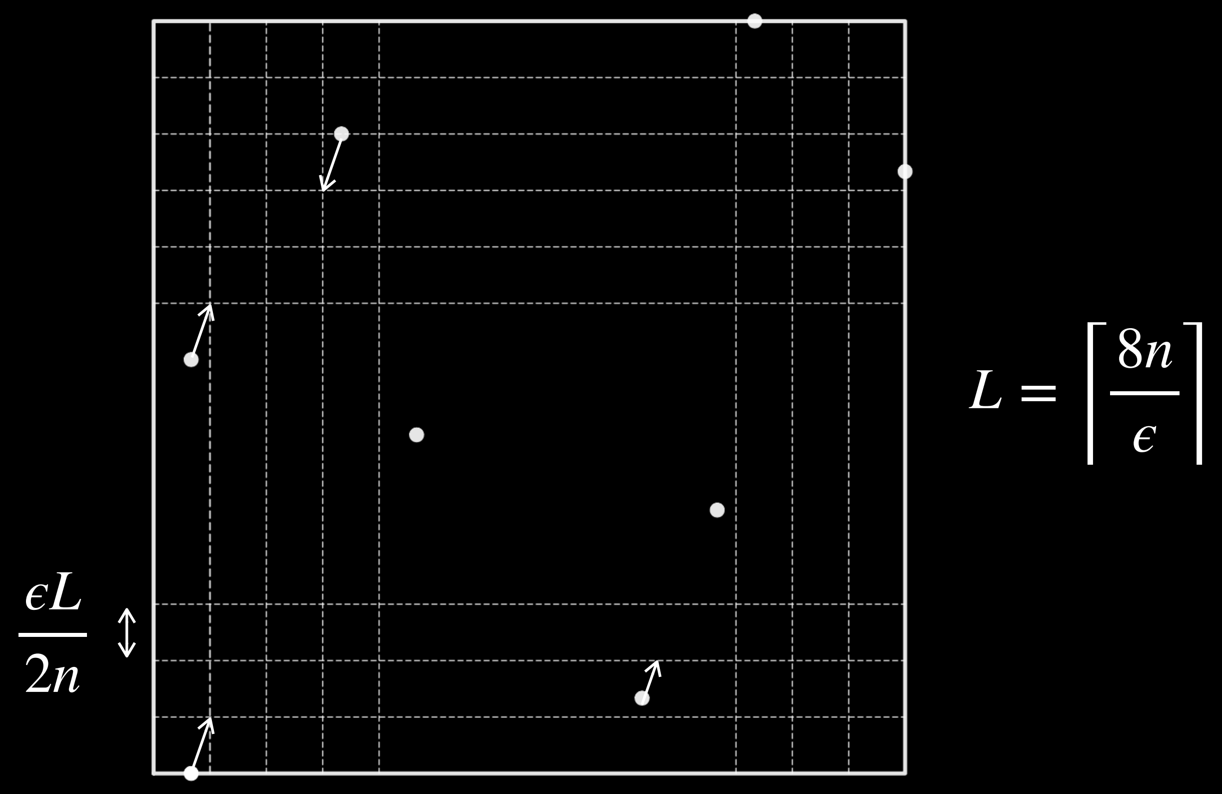
Euclidean TSP

Arora's Algorithm - Recap

1.



2.



Mapping points from I to I': distance $\leq \dfrac{\epsilon L}{2n}$

$\Rightarrow$ Mapping edges from I to I': length edge $\leq \dfrac{\epsilon L}{n}$

$\Rightarrow OPT_{I'} \leq OPT_I + \epsilon L$

$\Longrightarrow OPT_{I'} \leq OPT_I + \epsilon L \leq (1+\epsilon)\ OPT_I$

Real world application

Thank you for your attention!

References

- Demaine, E., Devadas, S., & Lynch, N. (2015). Approximation Algorithms: Traveling Salesman Problem. Retrieved from https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-046j-design-and-analysis-of-algorithms-spring-2015/recitation-notes/MIT6_046JS15_Recitation9.pdf
- Svensson Ola (2013). Approxiimation algorithms: Euclidean TSP. Retrieved from https://theory.epfl.ch/osven/courses/Approx13/Notes/lecture4-5.pdf
- Korte, B. H., & Vygen, J. (2010). Combinatorial optimization: Theory and algorithms. Berlin: Springer.Korte, B. H., & Vygen, J. (2010). Combinatorial optimization: Theory and algorithms. Berlin: Springer.

Github: https://github.com/juan190199/TravelingSalesmanProblem