The Constrained Minimum Spanning Tree Problem

Extended Abstract by R. Ravi and M. X. Goemans

Emma Ahrens

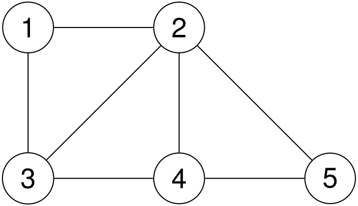
Seminar zur Diskreten Optimierung, RWTH Aachen

May 27, 2021

Table of Contents

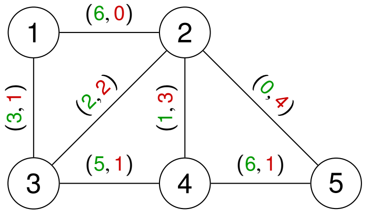
Table of Contents

Constrained Minimum Spanning Tree Problem



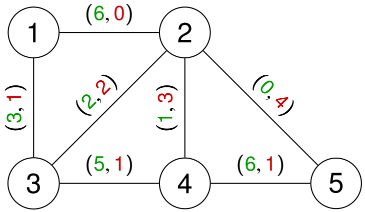
Given an undirected graph $G = (V, E)$

Constrained Minimum Spanning Tree Problem



Given an undirected graph $G = (V, E)$ and two cost functions $w, l : E \to \mathbb{N}_{\geq 0}$,

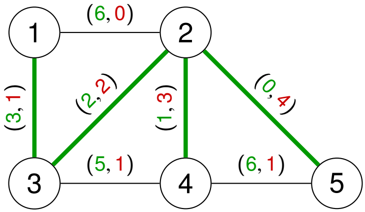
Constrained Minimum Spanning Tree Problem



Given an undirected graph $G = (V, E)$ and two cost functions $w, l : E \to \mathbb{N}_{\geq 0}$, find a spanning tree T with

- minimum total weight $w(T)$ and
- minimum total length $l(T)$.

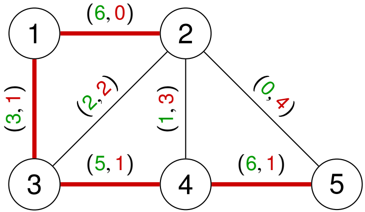
Constrained Minimum Spanning Tree Problem



Given an undirected graph $G = (V, E)$ and two cost functions $w, l : E \to \mathbb{N}_{\geq 0}$, find a spanning tree T with

- minimum total weight $w(T)$ and
- minimum total length $l(T)$.

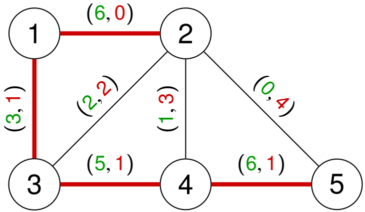
Constrained Minimum Spanning Tree Problem



Given an undirected graph $G = (V, E)$ and two cost functions $w, l : E \rightarrow \mathbb{N}_{\geq 0}$, find a spanning tree T with

- minimum total weight $w(T)$ and
- minimum total length $l(T)$.

Constrained Minimum Spanning Tree Problem



Given an undirected graph $G = (V, E)$ and two cost functions $w, l : E \to \mathbb{N}_{\geq 0}$, find a spanning tree T with

- minimum total weight $w(T)$ and
- minimum total length $l(T)$.

$\Rightarrow$ Specify budget $L \in \mathbb{N}_{\geq 0}$ for the length and minimize the weight.

Exact Solution

Integer Program (CMST)

S is the set of incidence vectors of spanning trees of G and

$$\begin{aligned} W = \min \ & \sum_{e \in E} w_e x_e \\ \text{s.t.} \ & x \in S \\ & \sum_{e \in E} l_e x_e \leq L. \end{aligned}$$

Exact Solution

Integer Program (CMST)

S is the set of incidence vectors of spanning trees of G and

$$\begin{aligned} W = \min\ & \sum_{e \in E} w_e x_e \\ \text{s.t.}\ & x \in S \\ & \sum_{e \in E} l_e x_e \leq L. \end{aligned}$$

Theorem [AAN82]

The problem is (weakly) NP-hard.

Exact Solution

Integer Program (CMST)

S is the set of incidence vectors of spanning trees of G and

$$\begin{aligned} W = \min \ & \sum_{e \in E} w_e x_e \\ \text{s.t.} \ & x \in S \\ & \sum_{e \in E} l_e x_e \leq L. \end{aligned}$$

Theorem [AAN82]

The problem is (weakly) NP-hard.

Theorem [RG96]

A (1,2)-approximation can be computed in polynomial runtime.

Exact Solution

Integer Program (CMST)

S is the set of incidence vectors of spanning trees of G and

$$\begin{aligned} W = \min \ & \sum_{e \in E} w_e x_e \\ \text{s.t.} \ & x \in S \\ & \sum_{e \in E} l_e x_e \leq L. \end{aligned}$$

Theorem [AAN82]

The problem is (weakly) NP-hard.

Theorem [RG96]

A (1,2)-approximation can be computed in polynomial runtime.

$\Rightarrow$ For fixed $\epsilon > 0$, we can even find $(1, 1+\epsilon)$-approximation.

Approximation Algorithm

1. Compute Lagrangian relaxation P_z of the IP

$$\ell(z) = \min \sum_{e \in E} (\underbrace{w_e + z l_e}_{= c_e}) x_e - zL$$

$$\text{s.t. } x \in S.$$

Approximation Algorithm

1. Compute Lagrangian relaxation P_z of the IP

$$\ell(z) = \min \sum_{e \in E} (\underbrace{w_e + z l_e}_{= c_e}) x_e - zL$$
$$\text{s.t.} \quad x \in S.$$

2. Use Megiddo's algorithm to compute value z^* which maximizes

$$\mathcal{L} = \max_{z \geq 0} \ell(z)$$

and $T_{\min}$, $T_{\max}$ optimal for P_{z^*} with min. (resp. max.) length.

Approximation Algorithm

1. Compute Lagrangian relaxation P_z of the IP

$$\ell(z) = \min \sum_{e \in E} (\underbrace{w_e + z l_e}_{= c_e}) x_e - zL$$

$$\text{s.t.} \quad x \in S.$$

2. Use Megiddo's algorithm to compute value z^* which maximizes

$$\mathcal{L} = \max_{z \geq 0} \ell(z)$$

and $T_{\min}$, $T_{\max}$ optimal for P_{z^*} with min. (resp. max.) length.

3. Compute sequence

$$T_{min} = T_0, T_1, \ldots, T_{i-1}, T_i, \ldots, T_k = T_{max}.$$

Pick first T_i such that $l(T_{i-1}) < L$ and $l(T_i) \geq L$, then

$$l(T_i) \leq L + l_{max} \leq 2L \quad \text{and} \quad w(T_i) \leq W.$$

Table of Contents

Table of Contents

Lagrangian Relaxation

$$
\begin{aligned}
W = \ & \min \sum_{e \in E} w_e x_e \\
& s.t. \sum_{e \in E} l_e x_e \leq L \\
& \qquad x \in S.
\end{aligned}
$$

Lagrangian Relaxation

$$
\begin{aligned}
W = \ & \min \sum_{e \in E} w_e x_e \\
& s.t. \sum_{e \in E} l_e x_e \leq L \\
& \quad x \in S.
\end{aligned}
$$

$$
\Rightarrow \qquad \ell(z) = \min \sum_{e \in E} w_e x_e + z\left(\sum_{e \in E} l_e x_e - L\right)
$$

$$
s.t.\ x \in S
$$

Lagrangian Relaxation

$$
\begin{aligned}
W = \ & \min \sum_{e \in E} w_e x_e \\
& s.t. \sum_{e \in E} l_e x_e \leq L \\
& \qquad x \in S.
\end{aligned}
$$

$$
\Rightarrow \qquad \ell(z) = \min \sum_{e \in E} w_e x_e + z \left(\sum_{e \in E} l_e x_e - L \right)
$$

$$
s.t. \ x \in S
$$

$$
\Leftrightarrow \qquad \ell(z) = \min \sum_{e \in E} \underbrace{(w_e + z l_e)}_{:= c_z(e)} x_e - zL
$$

$$
s.t. \ x \in S
$$

Lagrangian Relaxation

$$\ell(z) = \min \sum_{e \in E} \underbrace{(w_e + z l_e)}_{:= c_z(e)} x_e - zL$$

$$s.t.\ x \in S$$

- $\ell(z) : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is concave, piecewise linear

Lagrangian Relaxation

$$\ell(z) = \min \sum_{e \in E} \underbrace{(w_e + z l_e)}_{:= c_z(e)} x_e - zL$$
$$s.t.\ x \in S$$

- $\ell(z) : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is concave, piecewise linear
- $\ell(z) \leq W$ is a lower bound for CMST problem and $\mathcal{L} := \max_{z \in \mathbb{R}_{\geq 0}} \ell(z)$ is the greatest lower bound

Lagrangian Relaxation

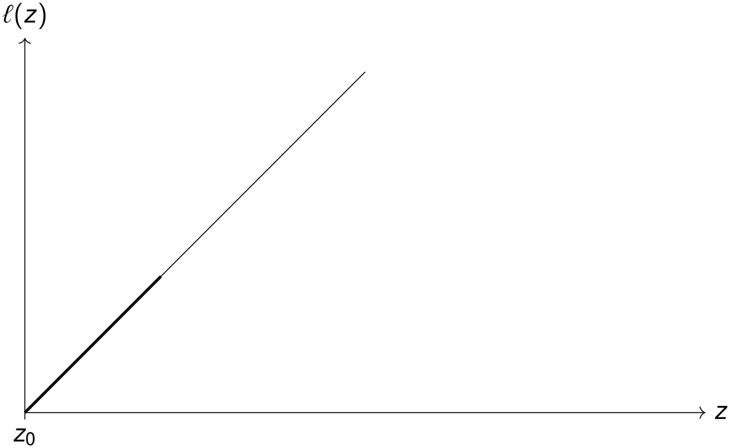
$$\ell(z) = \min \sum_{e \in E} \underbrace{(w_e + z l_e)}_{:= c_z(e)} x_e - zL$$

$$s.t.\ x \in S$$

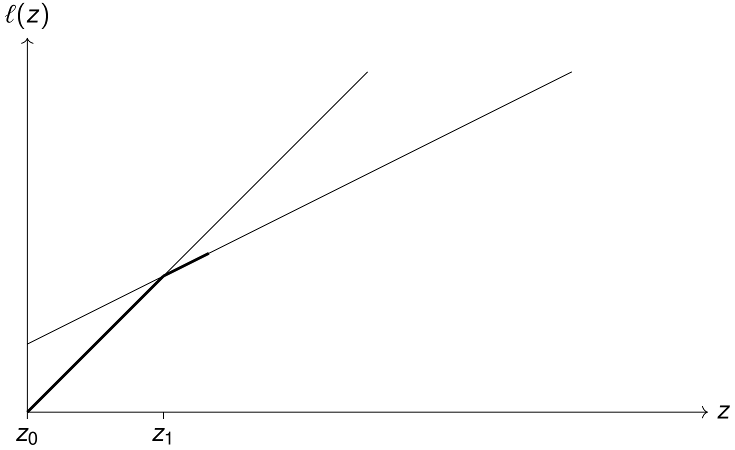
- $\ell(z) : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is concave, piecewise linear
- $\ell(z) \leq W$ is a lower bound for CMST problem and $\mathcal{L} := \max_{z \in \mathbb{R}_{\geq 0}} \ell(z)$ is the greatest lower bound

We obtain a set of *minimum spanning tree* problems for the cost function $c_z : E \to \mathbb{R}_{\geq 0}, e \mapsto w_e + z l_e$ and arbitrary $z \in \mathbb{R}_{\geq 0}$.

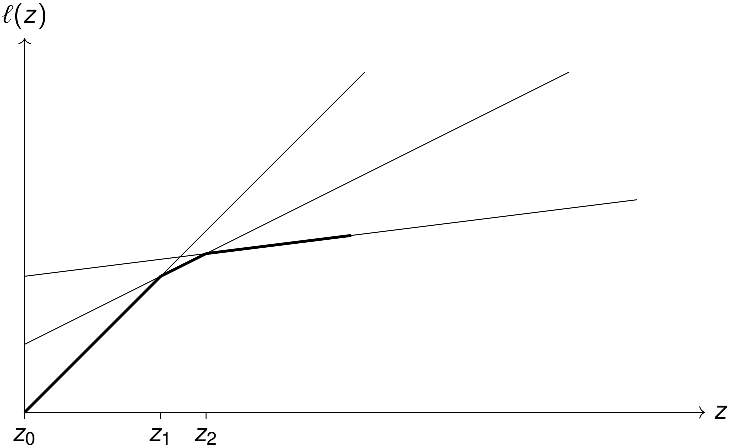
Lagrangian Relaxation



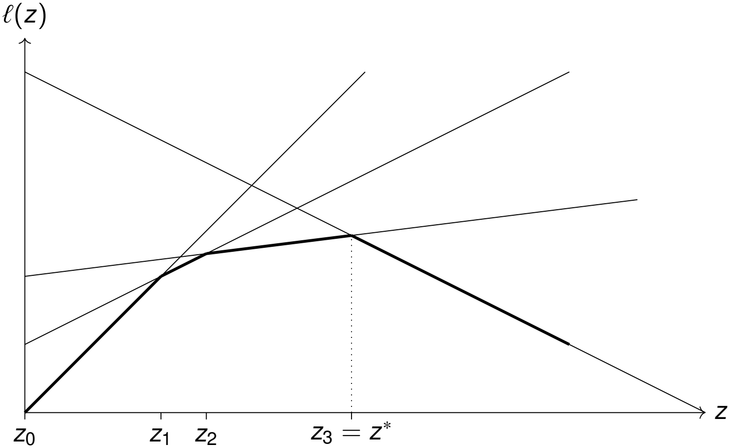
Lagrangian Relaxation



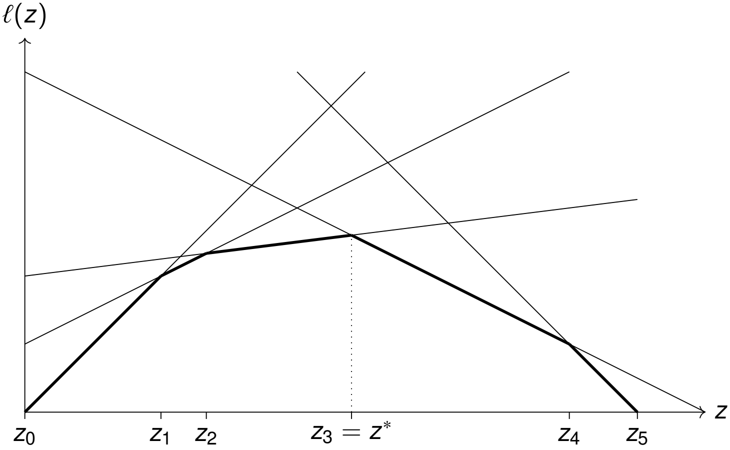
Lagrangian Relaxation



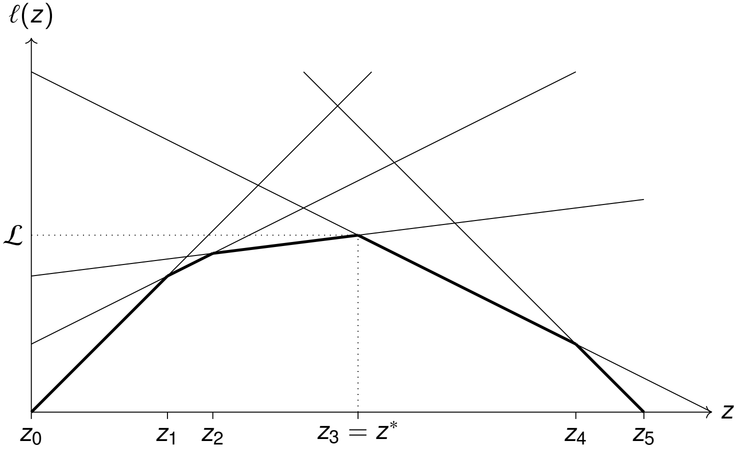
Lagrangian Relaxation



Lagrangian Relaxation



Lagrangian Relaxation



Lagrangian Relaxation

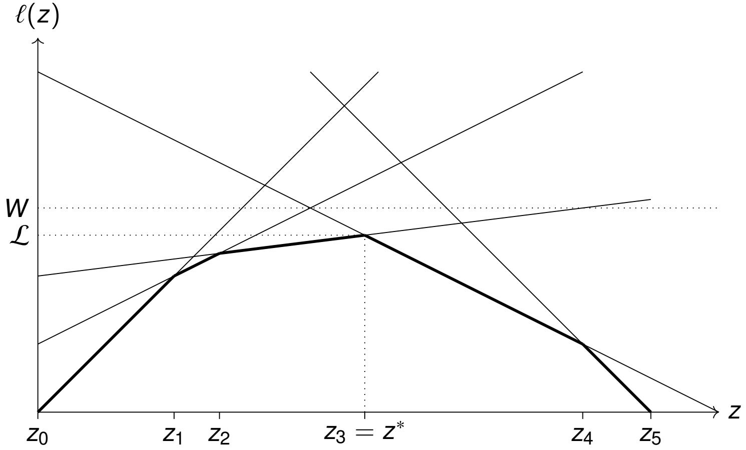


Table of Contents

Sequence of Optimal Solutions

Define $O_z \subseteq S$ as set of all MSTs for cost function c_z.

Sequence of Optimal Solutions

Define $O_z \subseteq S$ as set of all MSTs for cost function c_z.

Lemma

Let $T, T' \in O_z$. Then there exists a sequence

$$T =: T_0, T_1, \ldots, T_k, T_{k+1} := T'$$

in O_z such that T_i, T_{i+1} differ by a single edge swap, $i \in \{0, \ldots, k\}$.

Sequence of Optimal Solutions

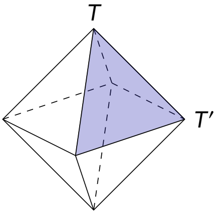
Define $O_z \subseteq S$ as set of all MSTs for cost function c_z.

Lemma

Let $T, T' \in O_z$. Then there exists a sequence

$$T =: T_0, T_1, \ldots, T_k, T_{k+1} := T'$$

in O_z such that T_i, T_{i+1} differ by a single edge swap, $i \in \{0, \ldots, k\}$.



Sequence of Optimal Solutions

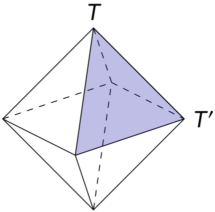
Define $O_z \subseteq S$ as set of all MSTs for cost function c_z.

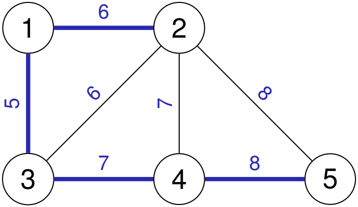
Lemma

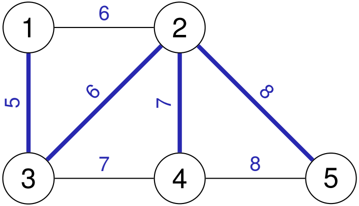
Let $T, T' \in O_z$. Then there exists a sequence

$$T =: T_0, T_1, \dots, T_k, T_{k+1} := T'$$

in O_z such that T_i, T_{i+1} differ by a single edge swap, $i \in \{0, \dots, k\}$.







Sequence of Optimal Solutions

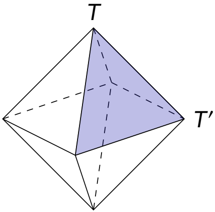
Define $O_z \subseteq S$ as set of all MSTs for cost function c_z.

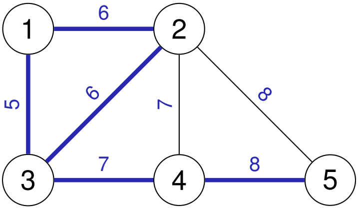
Lemma

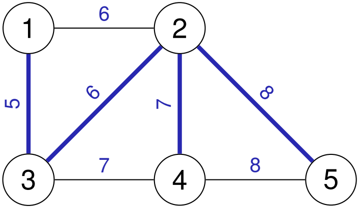
Let $T, T' \in O_z$. Then there exists a sequence

$$T =: T_0, T_1, \ldots, T_k, T_{k+1} := T'$$

in O_z such that T_i, T_{i+1} differ by a single edge swap, $i \in \{0, \ldots, k\}$.







Sequence of Optimal Solutions

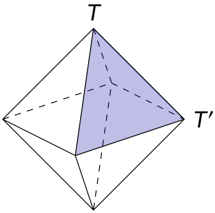
Define $O_z \subseteq S$ as set of all MSTs for cost function c_z.

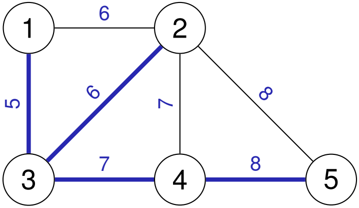
Lemma

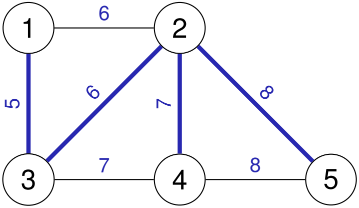
Let $T, T' \in O_z$. Then there exists a sequence

$$T =: T_0, T_1, \ldots, T_k, T_{k+1} := T'$$

in O_z such that T_i, T_{i+1} differ by a single edge swap, $i \in \{0, \ldots, k\}$.







Sequence of Optimal Solutions

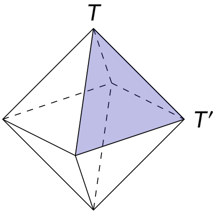
Define $O_z \subseteq S$ as set of all MSTs for cost function c_z.

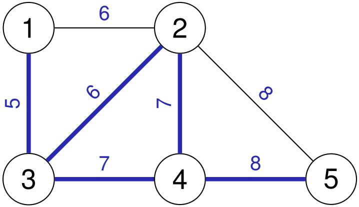
Lemma

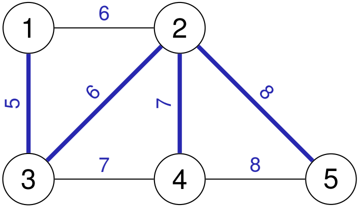
Let $T, T' \in O_z$. Then there exists a sequence

$$T =: T_0, T_1, \ldots, T_k, T_{k+1} := T'$$

in O_z such that T_i, T_{i+1} differ by a single edge swap, $i \in \{0, \ldots, k\}$.







Sequence of Optimal Solutions

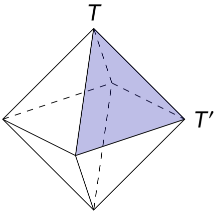
Define $O_z \subseteq S$ as set of all MSTs for cost function c_z.

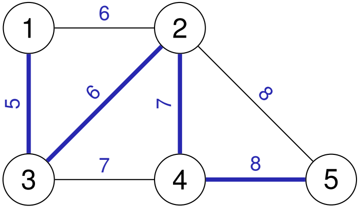
Lemma

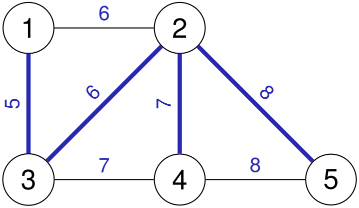
Let $T, T' \in O_z$. Then there exists a sequence

$$T =: T_0, T_1, \dots, T_k, T_{k+1} := T'$$

in O_z such that T_i, T_{i+1} differ by a single edge swap, $i \in \{0, \dots, k\}$.







Sequence of Optimal Solutions

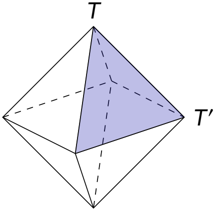
Define $O_z \subseteq S$ as set of all MSTs for cost function c_z.

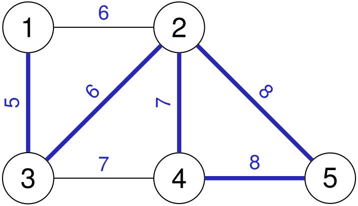
Lemma

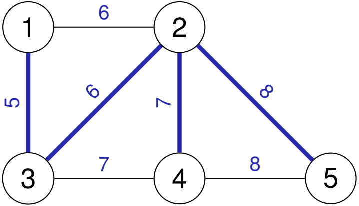
Let $T, T' \in O_z$. Then there exists a sequence

$$T =: T_0, T_1, \ldots, T_k, T_{k+1} := T'$$

in O_z such that T_i, T_{i+1} differ by a single edge swap, $i \in \{0, \ldots, k\}$.







Sequence of Optimal Solutions

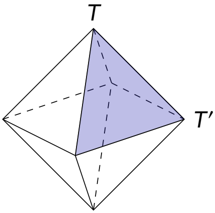
Define $O_z \subseteq S$ as set of all MSTs for cost function c_z.

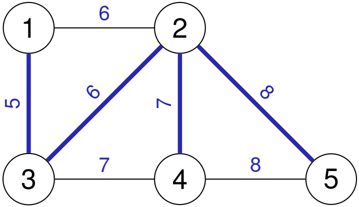
Lemma

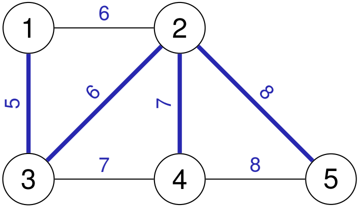
Let $T, T' \in O_z$. Then there exists a sequence

$$T =: T_0, T_1, \ldots, T_k, T_{k+1} := T'$$

in O_z such that T_i, T_{i+1} differ by a single edge swap, $i \in \{0, \ldots, k\}$.







Existence of $(1, 2)$-Approximation

Theorem

Let $z^* \in \mathbb{R}_{\geq 0}$ be optimal s.t. $\mathcal{L} = \ell(z^*)$. There exists $T \in O_{z^*}$ with

$$w(T) \leq W \quad \text{and} \quad l(T) \leq L + l_{\max}.$$

Existence of $(1, 2)$-Approximation

Theorem

Let $z^* \in \mathbb{R}_{\geq 0}$ be optimal s.t. $\mathcal{L} = \ell(z^*)$. There exists $T \in O_{z^*}$ with

$$w(T) \leq W \quad \text{and} \quad l(T) \leq L + l_{\text{max}}.$$

Proof

Let $T \in O_{z^*}$.

Existence of $(1, 2)$-Approximation

Theorem

Let $z^* \in \mathbb{R}_{\geq 0}$ be optimal s.t. $\mathcal{L} = \ell(z^*)$. There exists $T \in O_{z^*}$ with

$$w(T) \leq W \quad \text{and} \quad l(T) \leq L + l_{\max}.$$

Proof

Let $T \in O_{z^*}$. Then

$$w(T) = w(T) + z^*(l(T) - L) - z^*(l(T) - L)$$

Existence of $(1,2)$-Approximation

Theorem

Let $z^* \in \mathbb{R}_{\geq 0}$ be optimal s.t. $\mathcal{L} = \ell(z^*)$. There exists $T \in O_{z^*}$ with

$$w(T) \leq W \quad \text{and} \quad l(T) \leq L + l_{\max}.$$

Proof

Let $T \in O_{z^*}$. Then

$$\begin{aligned} w(T) &= w(T) + z^*(l(T) - L) - z^*(l(T) - L) \\ &= \ell(z^*) - z^*(l(T) - L) \end{aligned}$$

Existence of $(1,2)$-Approximation

Theorem

Let $z^* \in \mathbb{R}_{\geq 0}$ be optimal s.t. $\mathcal{L} = \ell(z^*)$. There exists $T \in O_{z^*}$ with

$$w(T) \leq W \quad \text{and} \quad l(T) \leq L + l_{\max}.$$

Proof

Let $T \in O_{z^*}$. Then

$$\begin{aligned} w(T) &= w(T) + z^*(l(T) - L) - z^*(l(T) - L) \\ &= \ell(z^*) - z^*(l(T) - L) = \mathcal{L} - z^*\underbrace{(l(T) - L)}_{\geq 0} \end{aligned}$$

Existence of $(1, 2)$-Approximation

Theorem

Let $z^* \in \mathbb{R}_{\geq 0}$ be optimal s.t. $\mathcal{L} = \ell(z^*)$. There exists $T \in O_{z^*}$ with

$$w(T) \leq W \quad \text{and} \quad l(T) \leq L + l_{\max}.$$

Proof

Let $T \in O_{z^*}$. Then

$$\begin{aligned} w(T) &= w(T) + z^*(l(T) - L) - z^*(l(T) - L) \\ &= \ell(z^*) - z^*(l(T) - L) = \mathcal{L} - z^*\underbrace{(l(T) - L)}_{\geq 0} \end{aligned}$$

and $w(T) \leq \mathcal{L} \leq W \iff l(T) \geq L$.

Existence of $(1, 2)$-Approximation

Theorem

Let $z^* \in \mathbb{R}_{\geq 0}$ be optimal s.t. $\mathcal{L} = \ell(z^*)$. There exists $T \in O_{z^*}$ with

$$w(T) \leq W \quad \text{and} \quad l(T) \leq L + l_{\max}.$$

Proof

Let $T \in O_{z^*}$. Then

$$\begin{aligned} w(T) &= w(T) + z^*(l(T) - L) - z^*(l(T) - L) \\ &= \ell(z^*) - z^*(l(T) - L) = \mathcal{L} - z^*(\underbrace{l(T) - L}_{\geq 0}) \end{aligned}$$

and $w(T) \leq \mathcal{L} \leq W \iff l(T) \geq L$.

$\Rightarrow$ Show that there exists $T \in O_{z^*}$ with $L \leq l(T) \leq L + l_{\max}$.

Existence of $(1, 2)$-Approximation

Proof (continued)

$\Rightarrow$ Show that there exists $T \in O_{z^*}$ with $L \leq l(T) \leq L + l_{\max}$.

Existence of $(1, 2)$-Approximation

Proof (continued)

$\Rightarrow$ Show that there exists $T \in O_{z^*}$ with $L \leq l(T) \leq L + l_{\max}$.

- There exists $T_{\leq} \in O_{z^*}$ s.t. $l(T_{\leq}) \leq L$:

Existence of $(1,2)$-Approximation

Proof (continued)

$\Rightarrow$ Show that there exists $T \in O_{z^*}$ with $L \leq l(T) \leq L + l_{\max}$.

- There exists $T_{\leq} \in O_{z^*}$ s.t. $l(T_{\leq}) \leq L$:
  Choose $\epsilon > 0$ with $O_{z^*+\epsilon} \subseteq O_{z^*}$ (without proof) and $T_{\leq} \in O_{z^*+\epsilon}$:

Existence of $(1,2)$-Approximation

Proof (continued)

$\Rightarrow$ Show that there exists $T \in O_{z^*}$ with $L \leq l(T) \leq L + l_{\max}$.

- There exists $T_\leq \in O_{z^*}$ s.t. $l(T_\leq) \leq L$:
  Choose $\epsilon > 0$ with $O_{z^*+\epsilon} \subseteq O_{z^*}$ (without proof) and $T_\leq \in O_{z^*+\epsilon}$:

$$\ell(z^* + \epsilon) = c_{z^*+\epsilon}(T_\leq) - (z^* + \epsilon)L \leq c_{z^*}(T_\leq) - z^* L = \ell(z^*)$$

Existence of $(1, 2)$-Approximation

Proof (continued)

$\Rightarrow$ Show that there exists $T \in O_{z^*}$ with $L \leq l(T) \leq L + l_{\max}$.

- There exists $T_{\leq} \in O_{z^*}$ s.t. $l(T_{\leq}) \leq L$:
  Choose $\epsilon > 0$ with $O_{z^*+\epsilon} \subseteq O_{z^*}$ (without proof) and $T_{\leq} \in O_{z^*+\epsilon}$:

$$\ell(z^* + \epsilon) = c_{z^*+\epsilon}(T_{\leq}) - (z^* + \epsilon)L \leq c_{z^*}(T_{\leq}) - z^* L = \ell(z^*)$$
$$\Leftrightarrow w(T_{\leq}) + (z^* + \epsilon)(l(T_{\leq}) - L) \leq w(T_{\leq}) + z^*(l(T_{\leq}) - L)$$

Existence of $(1,2)$-Approximation

Proof (continued)

$\Rightarrow$ Show that there exists $T \in O_{z^*}$ with $L \leq l(T) \leq L + l_{\max}$.

- There exists $T_{\leq} \in O_{z^*}$ s.t. $l(T_{\leq}) \leq L$:
  Choose $\epsilon > 0$ with $O_{z^*+\epsilon} \subseteq O_{z^*}$ (without proof) and $T_{\leq} \in O_{z^*+\epsilon}$:

$$\begin{aligned} \ell(z^* + \epsilon) = c_{z^*+\epsilon}(T_{\leq}) - (z^* + \epsilon)L \leq c_{z^*}(T_{\leq}) - z^* L = \ell(z^*) \\ \Leftrightarrow w(T_{\leq}) + (z^* + \epsilon)(l(T_{\leq}) - L) \leq w(T_{\leq}) + z^*(l(T_{\leq}) - L) \\ \Leftrightarrow \epsilon(l(T_{\leq}) - L) \leq 0 \Leftrightarrow l(T_{\leq}) \leq L. \end{aligned}$$

Existence of $(1,2)$-Approximation

Proof (continued)

$\Rightarrow$ Show that there exists $T \in O_{z^*}$ with $L \leq l(T) \leq L + l_{\max}$.

- There exists $T_{\leq} \in O_{z^*}$ s.t. $l(T_{\leq}) \leq L$:
  Choose $\epsilon > 0$ with $O_{z^*+\epsilon} \subseteq O_{z^*}$ (without proof) and $T_{\leq} \in O_{z^*+\epsilon}$:

$$\begin{aligned}\ell(z^* + \epsilon) = c_{z^*+\epsilon}(T_{\leq}) - (z^* + \epsilon)L \leq c_{z^*}(T_{\leq}) - z^* L = \ell(z^*)\\ \Leftrightarrow w(T_{\leq}) + (z^* + \epsilon)(l(T_{\leq}) - L) \leq w(T_{\leq}) + z^*(l(T_{\leq}) - L)\\ \Leftrightarrow \epsilon(l(T_{\leq}) - L) \leq 0 \Leftrightarrow l(T_{\leq}) \leq L.\end{aligned}$$

- There exists $T_{\geq} \in O_{z^*}$ s.t. $l(T_{\geq}) \geq L$: Analogous proof.

Existence of $(1,2)$-Approximation

Proof (continued)

$\Rightarrow$ Show that there exists $T \in O_{z^*}$ with $L \leq l(T) \leq L + l_{\max}$.

- There exists $T_\leq \in O_{z^*}$ s.t. $l(T_\leq) \leq L$:
  Choose $\epsilon > 0$ with $O_{z^*+\epsilon} \subseteq O_{z^*}$ (without proof) and $T_\leq \in O_{z^*+\epsilon}$:

$$\begin{aligned} \ell(z^* + \epsilon) = c_{z^*+\epsilon}(T_\leq) - (z^* + \epsilon)L \leq c_{z^*}(T_\leq) - z^*L = \ell(z^*) \\ \Leftrightarrow w(T_\leq) + (z^* + \epsilon)(l(T_\leq) - L) \leq w(T_\leq) + z^*(l(T_\leq) - L) \\ \Leftrightarrow \epsilon(l(T_\leq) - L) \leq 0 \Leftrightarrow l(T_\leq) \leq L. \end{aligned}$$

- There exists $T_\geq \in O_{z^*}$ s.t. $l(T_\geq) \geq L$: Analogous proof.
- Lemma: There exists sequence $T_\leq = T_0, T_1, \ldots, T_k, T_{k+1} = T_\geq$ in O_{z^*}.

Existence of $(1,2)$-Approximation

Proof (continued)

$\Rightarrow$ Show that there exists $T \in O_{z^*}$ with $L \leq l(T) \leq L + l_{\max}$.

- There exists $T_{\leq} \in O_{z^*}$ s.t. $l(T_{\leq}) \leq L$:
  Choose $\epsilon > 0$ with $O_{z^*+\epsilon} \subseteq O_{z^*}$ (without proof) and $T_{\leq} \in O_{z^*+\epsilon}$:

$$\ell(z^* + \epsilon) = c_{z^*+\epsilon}(T_{\leq}) - (z^* + \epsilon)L \leq c_{z^*}(T_{\leq}) - z^*L = \ell(z^*)$$
$$\Leftrightarrow w(T_{\leq}) + (z^* + \epsilon)(l(T_{\leq}) - L) \leq w(T_{\leq}) + z^*(l(T_{\leq}) - L)$$
$$\Leftrightarrow \epsilon(l(T_{\leq}) - L) \leq 0 \Leftrightarrow l(T_{\leq}) \leq L.$$

- There exists $T_{\geq} \in O_{z^*}$ s.t. $l(T_{\geq}) \geq L$: Analogous proof.
- Lemma: There exists sequence $T_{\leq} = T_0, T_1, \ldots, T_k, T_{k+1} = T_{\geq}$ in O_{z^*}.

$\Rightarrow$ Show that there exists element T_{i+1} such that

$$L \leq l(T_{i+1}) \leq L + l_{\max}.$$

Existence of $(1, 2)$-Approximation

Theorem

Let $z^* \in \mathbb{R}_{\geq 0}$ be optimal s.t. $\mathcal{L} = \ell(z^*)$. There exists $T \in O_{z^*}$ with

$$w(T) \leq W \quad \text{and} \quad l(T) \leq L + l_{\max}.$$

Existence of $(1, 2)$-Approximation

Theorem

Let $z^* \in \mathbb{R}_{\geq 0}$ be optimal s.t. $\mathcal{L} = \ell(z^*)$. There exists $T \in O_{z^*}$ with

$$w(T) \leq W \quad \text{and} \quad l(T) \leq L + l_{\max}.$$

Proof (continued)

We have $l(T_{\leq}) \leq L$, $T_{\leq} = T_0, T_1, \ldots, T_k, T_{k+1} = T_{\geq}$ and $l(T_{\geq}) \geq L$.

Existence of $(1, 2)$-Approximation

Theorem

Let $z^* \in \mathbb{R}_{\geq 0}$ be optimal s.t. $\mathcal{L} = \ell(z^*)$. There exists $T \in O_{z^*}$ with

$$w(T) \leq W \quad \text{and} \quad l(T) \leq L + l_{\max}.$$

Proof (continued)

We have $l(T_{\leq}) \leq L$, $T_{\leq} = T_0, T_1, \ldots, T_k, T_{k+1} = T_{\geq}$ and $l(T_{\geq}) \geq L$.
If $l(T_i) \leq L$ and $l(T_{i+1}) \geq L$ and $T_{i+1} = T_i - e + e'$ for $e, e' \in E$, then

$$l(T_{i+1}) = l(T_i - e + e') = l(T_i) - l_e + l_{e'} \leq L + l_{\max}.$$

$\square$

Existence of $(1, 2)$-Approximation

Corollary

Let $z^* \in \mathbb{R}_{\geq 0}$ be optimal s.t. $\mathcal{L} = \ell(z^*)$. There exists $T \in O_{z^*}$ with

$$w(T) \leq W \quad \text{and} \quad l(T) \leq 2L.$$

Existence of $(1, 2)$-Approximation

Corollary

Let $z^* \in \mathbb{R}_{\geq 0}$ be optimal s.t. $\mathcal{L} = \ell(z^*)$. There exists $T \in O_{z^*}$ with

$$w(T) \leq W \quad \text{and} \quad l(T) \leq 2L.$$

$\Rightarrow$ Such a spanning tree $T \in O_{z^*}$ is a $(1, 2)$-approximation.

Existence of $(1, 2)$-Approximation

Corollary

Let $z^* \in \mathbb{R}_{\geq 0}$ be optimal s.t. $\mathcal{L} = \ell(z^*)$. There exists $T \in O_{z^*}$ with

$$w(T) \leq W \quad \text{and} \quad l(T) \leq 2L.$$

$\Rightarrow$ Such a spanning tree $T \in O_{z^*}$ is a $(1, 2)$-approximation.
For a fixed $\epsilon > 0$, $\epsilon \in \mathbb{R}_{\geq 0}$, restrict edges to subset

$$E' := \{e \in E \mid l_e \leq \epsilon L\} \subseteq E.$$

Existence of $(1, 2)$-Approximation

Corollary

Let $z^* \in \mathbb{R}_{\geq 0}$ be optimal s.t. $\mathcal{L} = \ell(z^*)$. There exists $T \in O_{z^*}$ with

$$w(T) \leq W \quad \text{and} \quad l(T) \leq 2L.$$

$\Rightarrow$ Such a spanning tree $T \in O_{z^*}$ is a $(1, 2)$-approximation.
For a fixed $\epsilon > 0$, $\epsilon \in \mathbb{R}_{\geq 0}$, restrict edges to subset

$$E' := \{e \in E \mid l_e \leq \epsilon L\} \subseteq E.$$

Corollary

Let $z^* \in \mathbb{R}_{\geq 0}$ be optimal s.t. $\mathcal{L} = \ell(z^*)$. There exists $T \in O_{z^*}$ with

$$w(T) \leq W \quad \text{and} \quad l(T) \leq (1 + \epsilon)L.$$

Existence of $(1, 2)$-Approximation

Corollary

Let $z^* \in \mathbb{R}_{\geq 0}$ be optimal s.t. $\mathcal{L} = \ell(z^*)$. There exists $T \in O_{z^*}$ with

$$w(T) \leq W \quad \text{and} \quad l(T) \leq 2L.$$

$\Rightarrow$ Such a spanning tree $T \in O_{z^*}$ is a $(1, 2)$-approximation.
For a fixed $\epsilon > 0$, $\epsilon \in \mathbb{R}_{\geq 0}$, restrict edges to subset

$$E' := \{e \in E \mid l_e \leq \epsilon L\} \subseteq E.$$

Corollary

Let $z^* \in \mathbb{R}_{\geq 0}$ be optimal s.t. $\mathcal{L} = \ell(z^*)$. There exists $T \in O_{z^*}$ with

$$w(T) \leq W \quad \text{and} \quad l(T) \leq (1 + \epsilon)L.$$

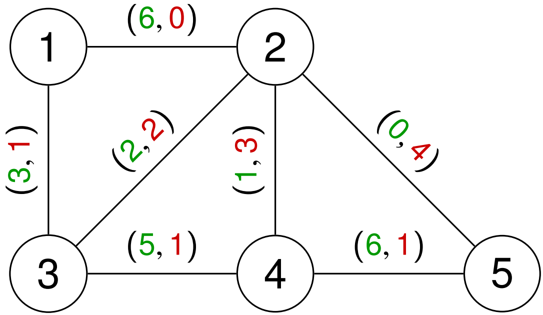
$\Rightarrow$ We can find a $(1, 1 + \epsilon)$-approximation.

Table of Contents

Table of Contents

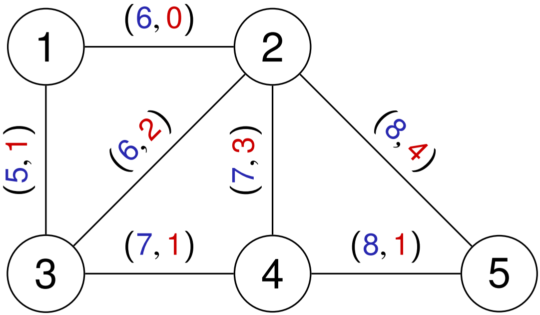
Optimum Spanning Tree with Minimum Length

$L = 8 \qquad z = 2 \qquad c_z = w_e + z l_e$



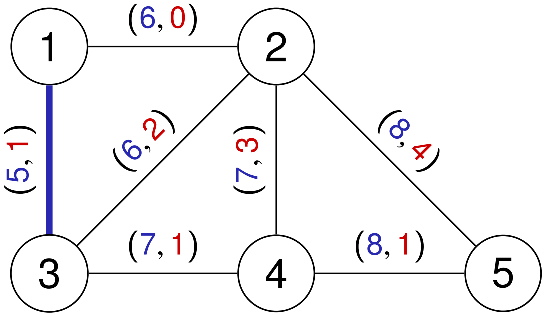
Optimum Spanning Tree with Minimum Length

$L = 8 \qquad z = 2 \qquad c_z = w_e + z l_e$



Optimum Spanning Tree with Minimum Length

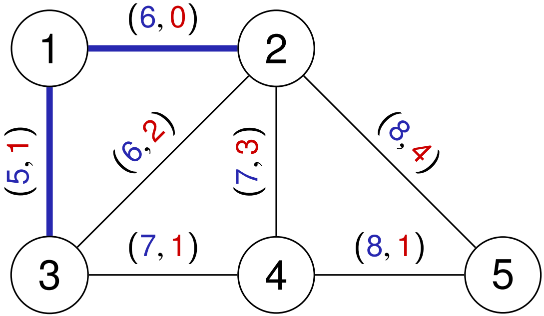
$L = 8 \qquad z = 2 \qquad c_z = w_e + z l_e$



$(5, 1)$

Optimum Spanning Tree with Minimum Length

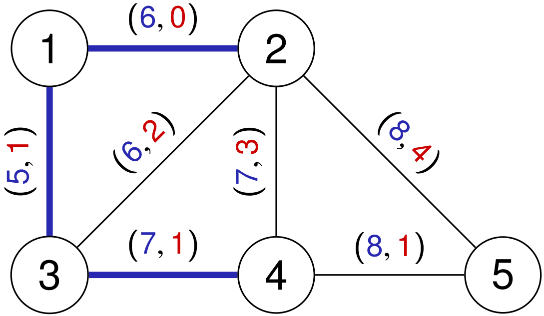
$L = 8 \qquad z = 2 \qquad c_z = w_e + z l_e$



$(5, 1) \prec_{\min} (6, 0) \prec_{\min} (6, 2)$

Optimum Spanning Tree with Minimum Length

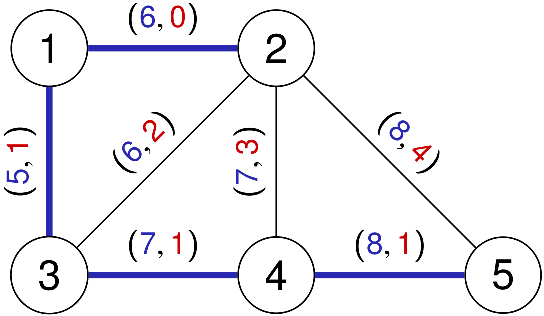
$L = 8 \qquad z = 2 \qquad c_z = w_e + z l_e$



$(5, 1) \prec_{\min} (6, 0) \prec_{\min} (6, 2) \prec_{\min} (7, 1) \prec_{\min} (7, 3)$

Optimum Spanning Tree with Minimum Length

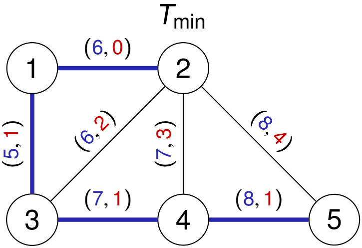
$L = 8 \qquad z = 2 \qquad c_z = w_e + z l_e$

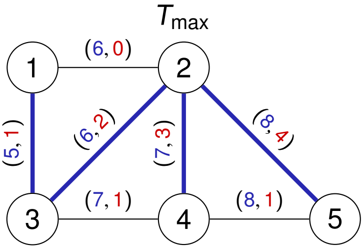


$(5,1) \prec_{\min} (6,0) \prec_{\min} (6,2) \prec_{\min} (7,1) \prec_{\min} (7,3) \prec_{\min} (8,1) \prec_{\min} (8,4)$

Lexicographic Order

$L = 8 \qquad z = 2$



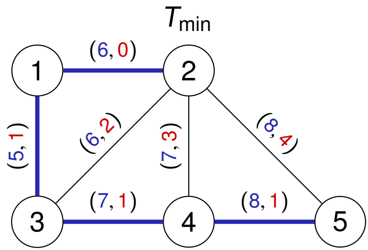


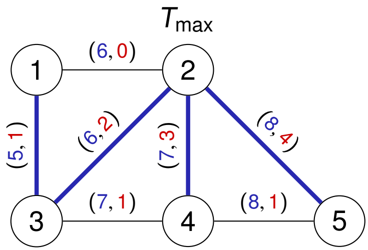
$(6, 0) \prec_{\min} (6, 2)$

$(6, 2) \prec_{\max} (6, 0)$

Lexicographic Order

$L = 8 \qquad z = 2$





$(6, 0) \prec_{\min} (6, 2)$

$(6, 2) \prec_{\max} (6, 0)$

$(c, l) \prec_{\min} (c', l')$
if $c < c'$
or $c = c'$ and $l < l'$

$(c, l) \prec_{\max} (c', l')$
if $c < c'$
or $c = c'$ and $l > l'$

Check Lagrange Multiplier z

Listing 1: check(z).

```
Input: Graph G, z ∈ ℝ≥0 and
   cost functions c_z(e), l_e : E → ℝ≥0
Output: Two spanning trees T_min and T_max
   and a valuation of z with respect to z*

1. Calculate tuples (c_z(e), l_e) for e ∈ E

2. Sort tuples based on order ≺_min
3. Find MST T_min via Prim's algorithm
4. Sort tuples based on order ≺_max
5. Find MST T_max via Prim's algorithm

6. If l(T_min) > L, then z < z*,
   if l(T_max) < L, then z > z*,
   otherwise z is an acceptable value
```

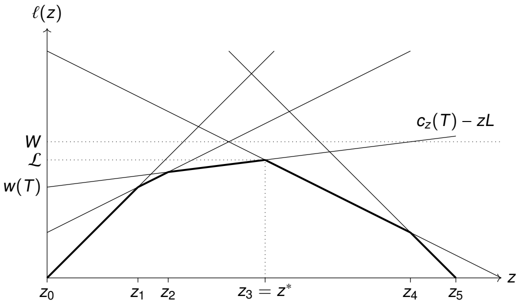
Check Lagrange Multiplier z

Listing 2: check(z).

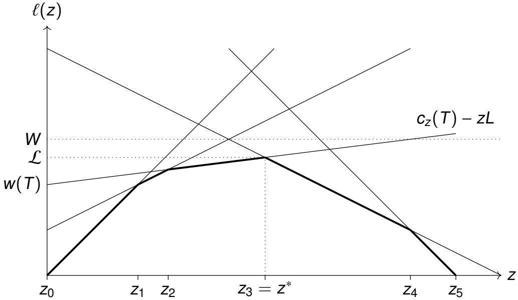
```
Input: Graph G, z ∈ ℝ≥0 and
   cost functions c_z(e), l_e : E → ℝ≥0
Output: Two spanning trees T_min and T_max
   and a valuation of z with respect to z*

1. Calculate tuples (c_z(e), l_e) for e ∈ E

2. Sort tuples based on order ≺_min
3. Find MST T_min via Prim's algorithm
4. Sort tuples based on order ≺_max
5. Find MST T_max via Prim's algorithm

6. If l(T_min) > L, then z < z*,
   if l(T_max) < L, then z > z*,
   otherwise z is an acceptable value
```

⇒ Runtime for $n = |V|$, $m = |E|$ is $O(m + n\log(n))$ [CLRS01].

Naïve Approach

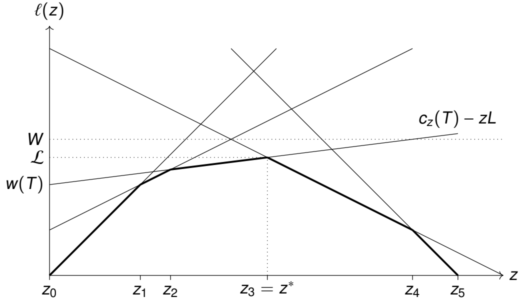


Naïve Approach



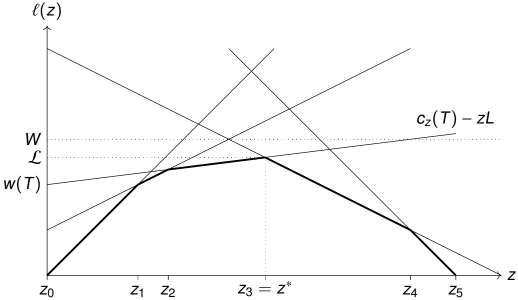
- $Z := \{z \in \mathbb{R}_{\geq 0} \mid \exists e, e' \in E \text{ s.t. } c_z(e) = c_z(e')\}$ with $|Z| \approx m^2$

Naïve Approach



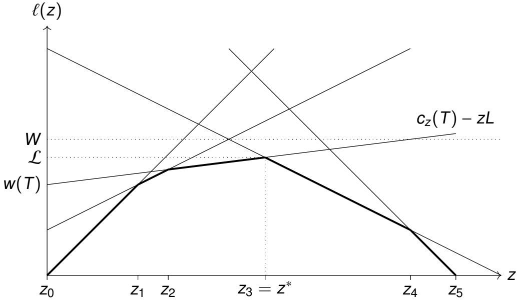
- $Z := \{z \in \mathbb{R}_{\geq 0} \mid \exists e, e' \in E \text{ s.t. } c_z(e) = c_z(e')\}$ with $|Z| \approx m^2$
- Sort in $O(m^2 \log(m^2)) = O(m^2 \log(m))$

Naïve Approach



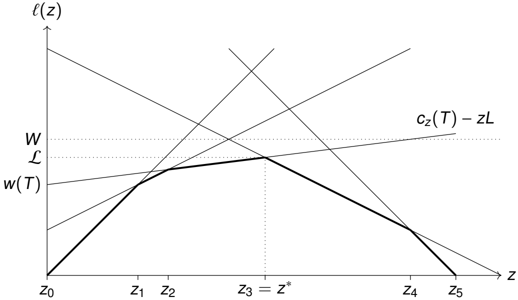
- $Z := \{z \in \mathbb{R}_{\geq 0} \mid \exists e, e' \in E \text{ s.t. } c_z(e) = c_z(e')\}$ with $|Z| \approx m^2$
- Sort in $O(m^2 \log(m^2)) = O(m^2 \log(m))$
- Binary search with $\texttt{check(z)}$ in $O(\log(m^2)(m + n\log(n))) = O(\log(m)(m + n\log(n)))$

Naïve Approach



- $Z := \{z \in \mathbb{R}_{\geq 0} \mid \exists e, e' \in E \text{ s.t. } c_z(e) = c_z(e')\}$ with $|Z| \approx m^2$
- Sort in $O(m^2 \log(m^2)) = O(m^2 \log(m))$
- Binary search with $\texttt{check}(z)$ in $O(\log(m^2)(m + n\log(n))) = O(\log(m)(m + n\log(n)))$
- Compute sequence in $O(n\log(n))$ [ST83]

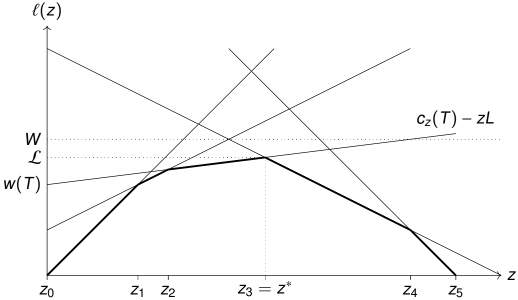
Naïve Approach



- $Z := \{z \in \mathbb{R}_{\geq 0} \mid \exists e, e' \in E \text{ s.t. } c_z(e) = c_z(e')\}$ with $|Z| \approx m^2$
- Sort in $O(m^2 \log(m^2)) = O(m^2 \log(m))$
- Binary search with $\texttt{check(z)}$ in $O(\log(m^2)(m + n\log(n))) = O(\log(m)(m + n\log(n)))$
- Compute sequence in $O(n\log(n))$ [ST83]

$\Rightarrow$ Runtime $O(m^2 \log(m) + \log(m)(m + n\log(n)) + n\log(n))$

Naïve Approach

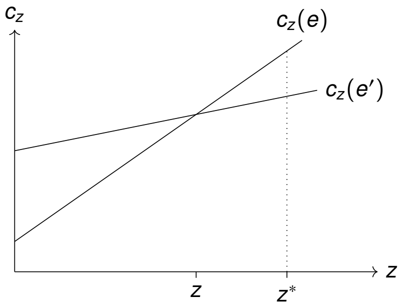


- $Z := \{z \in \mathbb{R}_{\geq 0} \mid \exists e, e' \in E \text{ s.t. } c_z(e) = c_z(e')\}$ with $|Z| \approx m^2$
- Sort in $O(m^2 \log(m^2)) = O(m^2 \log(m))$
- Binary search with $\texttt{check}(\texttt{z})$ in $O(\log(m^2)(m + n\log(n))) = O(\log(m)(m + n\log(n)))$
- Compute sequence in $O(n\log(n))$ [ST83]

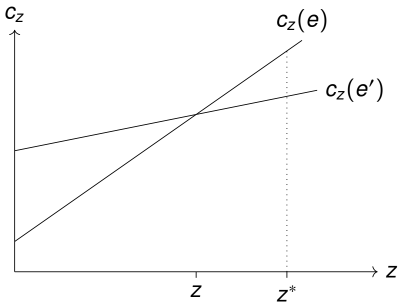
$\Rightarrow$ Runtime $O(m^2\log(m) + \log(m)(m + n\log(n)) + n\log(n))$

$$= O(m^2\log(m) + n\log(m)\log(n))$$

Compare Edges



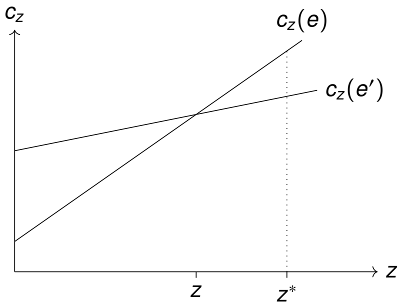
Compare Edges



Let $l_e > l_{e'}$. If the algorithm $\texttt{check}(z)$ returns

1. $z < z^*$, then $c_{z^*}(e) > c_{z^*}(e')$,
2. $z > z^*$, then $c_{z^*}(e) < c_{z^*}(e')$, and
3. z is accepted, then $c_{z^*}(e) = c_{z^*}(e')$.

Compare Edges



Let $l_e > l_{e'}$. If the algorithm $\texttt{check}(z)$ returns

1. $z < z^*$, then $c_{z^*}(e) > c_{z^*}(e')$,
2. $z > z^*$, then $c_{z^*}(e) < c_{z^*}(e')$, and
3. z is accepted, then $c_{z^*}(e) = c_{z^*}(e')$.

$\Rightarrow$ Sort edges in E by cost c_{z^*} without knowing z^*.

Table of Contents

Preparata's Algorithm for Parallel Sorting

m elements

3	4	8	2	6	1	$\cdots$	4	1	9	2	3	5

Preparata's Algorithm for Parallel Sorting

$m\log(m)$ processors

$\frac{m}{m\log(m)}$ elements

m elements

3	4	8	2	6	1	$\cdots$	4	1	9	2	3	5

Preparata's Algorithm for Parallel Sorting

$m \log(m)$ processors

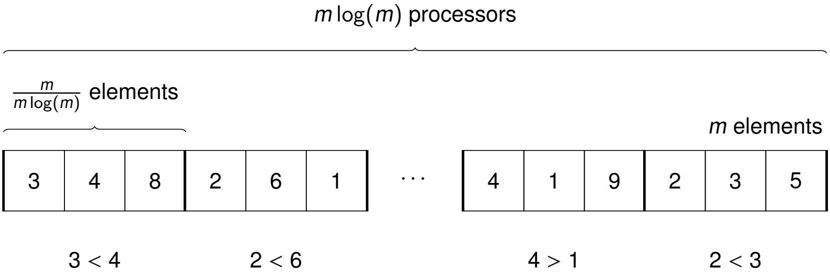
$\frac{m}{m \log(m)}$ elements

m elements

3	4	8	2	6	1	...	4	1	9	2	3	5

3 ? 4   2 ? 6   4 ? 1   2 ? 3

Preparata's Algorithm for Parallel Sorting



Preparata's Algorithm for Parallel Sorting

$m\log(m)$ processors

$\frac{m}{m\log(m)}$ elements

m elements

3	4	8	2	6	1	$\cdots$	4	1	9	2	3	5

$3 < 4$ $\quad$ $2 < 6$ $\quad$ $4 > 1$ $\quad$ $2 < 3$

$\vdots$ $\quad$ $\vdots$ $\quad$ $\vdots$ $\quad$ $\vdots$

Preparata's Algorithm for Parallel Sorting

$m\log(m)$ processors

$\frac{m}{m\log(m)}$ elements

m elements

3	4	8	2	6	1	$\cdots$	4	1	9	2	3	5

$3 < 4$ $\quad$ $2 < 6$ $\quad$ $4 > 1$ $\quad$ $2 < 3$

$\vdots$ $\quad$ $\vdots$ $\quad$ $\vdots$ $\quad$ $\vdots$

$\log(m)$ steps

Preparata's Algorithm for Parallel Sorting

$$\overbrace{\underbrace{\boxed{3}\,\boxed{4}\,\boxed{8}}_{}^{\frac{m}{m\log(m)} \text{ elements}}\,\boxed{2}\,\boxed{6}\,\boxed{1}\ \cdots\ \boxed{4}\,\boxed{1}\,\boxed{9}\,\boxed{2}\,\boxed{3}\,\boxed{5}}^{m\log(m) \text{ processors}}$$

m elements

3	4	8	2	6	1	⋯	4	1	9	2	3	5

$3 < 4$ $\quad 2 < 6$ $\quad 4 > 1$ $\quad 2 < 3$

$\vdots \quad \vdots \quad \vdots \quad \vdots$

$\log(m)$ steps $\quad \Rightarrow m \cdot \log(m)^2$ serial steps

Table of Contents

Megiddo's Algorithm

$m\log(m)$ processors

$\overbrace{e_1 \quad e_2 \quad e_3}^{\frac{m}{m\log(m)} \text{ elements}}$

m elements

e_1	e_2	e_3	e_4	e_5	e_6	$\cdots$	e_{m-5}	e_{m-4}	e_{m-3}	e_{m-2}	e_{m-1}	e_m

Megiddo's Algorithm

m elements

e_1	e_2	e_3	e_4	e_5	e_6	$\cdots$	e_{m-5}	e_{m-4}	e_{m-3}	e_{m-2}	e_{m-1}	e_m

Megiddo's Algorithm

m elements

e_1	e_2	e_3	e_4	e_5	e_6	$\cdots$	e_{m-5}	e_{m-4}	e_{m-3}	e_{m-2}	e_{m-1}	e_m

e_1 ? e_2 $\quad$ e_4 ? e_5 $\quad$ e_{m-5} ? e_{m-4} $\quad$ e_{m-2} ? e_{m-1}

Megiddo's Algorithm

m elements

e_1	e_2	e_3	e_4	e_5	e_6	$\cdots$	e_{m-5}	e_{m-4}	e_{m-3}	e_{m-2}	e_{m-1}	e_m

e_1 ? e_2 e_4 ? e_5 e_{m-5} ? e_{m-4} e_{m-2} ? e_{m-1}

$z_{1,2}$ $z_{4,5}$ $z_{m-5,m-4}$ $z_{m-2,m-1}$

Megiddo's Algorithm

m elements

e_1	e_2	e_3	e_4	e_5	e_6	$\cdots$	e_{m-5}	e_{m-4}	e_{m-3}	e_{m-2}	e_{m-1}	e_m

$e_1 \ ? \ e_2$ $\quad$ $e_4 \ ? \ e_5$ $\quad$ $e_{m-5} \ ? \ e_{m-4}$ $\quad$ $e_{m-2} \ ? \ e_{m-1}$

$z_{1,2}$ $\quad$ $z_{4,5}$ $\quad$ $z_{m-5,m-4}$ $\quad$ $z_{m-2,m-1}$

sort breakpoints

Megiddo's Algorithm

m elements

e_1	e_2	e_3	e_4	e_5	e_6	$\cdots$	e_{m-5}	e_{m-4}	e_{m-3}	e_{m-2}	e_{m-1}	e_m

$e_1 \ ? \ e_2$ $\quad$ $e_4 \ ? \ e_5$ $\quad$ $e_{m-5} \ ? \ e_{m-4}$ $\quad$ $e_{m-2} \ ? \ e_{m-1}$

$z_{1,2}$ $\quad$ $z_{4,5}$ $\quad$ $z_{m-5,m-4}$ $\quad$ $z_{m-2,m-1}$

sort breakpoints

$$z_{m-5,m-4} \leq z_{1,2} \leq \cdots \leq z_{4,5} \leq z_{m-2,m-1}$$

Megiddo's Algorithm

m elements

e_1	e_2	e_3	e_4	e_5	e_6	$\cdots$	e_{m-5}	e_{m-4}	e_{m-3}	e_{m-2}	e_{m-1}	e_m

e_1 ? e_2 $\quad$ e_4 ? e_5 $\quad$ e_{m-5} ? e_{m-4} $\quad$ e_{m-2} ? e_{m-1}

$z_{1,2}$ $\quad$ $z_{4,5}$ $\quad$ $z_{m-5,m-4}$ $\quad$ $z_{m-2,m-1}$

sort breakpoints

$$z_{m-5,m-4} \leq z_{1,2} \leq \cdots \leq z_{4,5} \leq z_{m-2,m-1}$$

$$\uparrow$$

$$z^*$$

Megiddo's Algorithm

m elements

e_1	e_2	e_3	e_4	e_5	e_6	$\cdots$	e_{m-5}	e_{m-4}	e_{m-3}	e_{m-2}	e_{m-1}	e_m

$e_1 \ ? \ e_2$ $\quad$ $e_4 \ ? \ e_5$ $\quad$ $e_{m-5} \ ? \ e_{m-4}$ $\quad$ $e_{m-2} \ ? \ e_{m-1}$

$z_{1,2}$ $\quad$ $z_{4,5}$ $\quad$ $z_{m-5,m-4}$ $\quad$ $z_{m-2,m-1}$

sort breakpoints

$$z_{m-5,m-4} \leq z_{1,2} \leq \cdots \leq z_{4,5} \leq z_{m-2,m-1}$$

$\uparrow$

z^*

Megiddo's Algorithm

m elements

e_1	e_2	e_3	e_4	e_5	e_6	$\cdots$	e_{m-5}	e_{m-4}	e_{m-3}	e_{m-2}	e_{m-1}	e_m

$e_1 \; ? \; e_2$ $\quad$ $e_4 \; ? \; e_5$ $\quad$ $e_{m-5} \; ? \; e_{m-4}$ $\quad$ $e_{m-2} \; ? \; e_{m-1}$

$\uparrow$

z^*

Megiddo's Algorithm

m elements

e_1	e_2	e_3	e_4	e_5	e_6	$\cdots$	e_{m-5}	e_{m-4}	e_{m-3}	e_{m-2}	e_{m-1}	e_m

$e_1 < e_2$ $\quad$ $e_4 > e_5$ $\quad$ $e_{m-5} > e_{m-4}$ $\quad$ $e_{m-2} < e_{m-1}$

$\uparrow$

z^*

Megiddo's Algorithm

Since $z^* \in Z$:

Megiddo's Algorithm

Since $z^* \in Z$:

$\Rightarrow$ Eventually we compare $e, e' \in E$ with $z_{e,e'} = z^*$.

Megiddo's Algorithm

Since $z^* \in Z$:

- $\Rightarrow$ Eventually we compare $e, e' \in E$ with $z_{e,e'} = z^*$.
- $\Rightarrow$ $\texttt{check}(z_{e,e'})$ returns $z_{e,e'}$ is optimal, and T_{min} and T_{max}.

Megiddo's Algorithm

Since $z^* \in Z$:

- $\Rightarrow$ Eventually we compare $e, e' \in E$ with $z_{e,e'} = z^*$.
- $\Rightarrow$ $\mathtt{check}(z_{e,e'})$ returns $z_{e,e'}$ is optimal, and T_{min} and T_{max}.
- $\Rightarrow$ Swapping edges, we obtain spanning tree $T \subseteq E$ such that T is a $(1, 2)$-approximation.

Runtime

m elements

e_1	e_2	e_3	e_4	e_5	e_6	$\cdots$	e_{m-5}	e_{m-4}	e_{m-3}	e_{m-2}	e_{m-1}	e_m

e_1 ? e_2 $\qquad$ e_4 ? e_5 $\qquad$ e_{m-5} ? e_{m-4} $\qquad$ e_{m-2} ? e_{m-1}

$z_{1,2}$ $\qquad$ $z_{4,5}$ $\qquad$ $z_{m-5,m-4}$ $\qquad$ $z_{m-2,m-1}$

sort breakpoints

- Sort breakpoints in $O(m\log(m)\cdot\log(m\log(m))) = O(m\log(m)^2)$

Runtime

m elements

e_1	e_2	e_3	e_4	e_5	e_6	$\cdots$	e_{m-5}	e_{m-4}	e_{m-3}	e_{m-2}	e_{m-1}	e_m

$e_1 \ ? \ e_2$ $\quad$ $e_4 \ ? \ e_5$ $\quad$ $e_{m-5} \ ? \ e_{m-4}$ $\quad$ $e_{m-2} \ ? \ e_{m-1}$

$z_{1,2}$ $\quad$ $z_{4,5}$ $\quad$ $z_{m-5,m-4}$ $\quad$ $z_{m-2,m-1}$

sort breakpoints

$z_{m-5,m-4} \leq z_{1,2} \leq \cdots \leq z_{4,5} \leq z_{m-2,m-1}$

- Sort breakpoints in $O(m\log(m) \cdot \log(m\log(m)) = O(m\log(m)^2)$
- Binary search with $\mathtt{check}(z)$ in $O(\log(m\log(m)) \cdot (m + n\log(n))) = O(\log(m)(m + n\log(n)))$

Runtime

m elements

e_1	e_2	e_3	e_4	e_5	e_6	$\cdots$	e_{m-5}	e_{m-4}	e_{m-3}	e_{m-2}	e_{m-1}	e_m

$e_1 \;?\; e_2$ $\qquad$ $e_4 \;?\; e_5$ $\qquad$ $e_{m-5} \;?\; e_{m-4}$ $\qquad$ $e_{m-2} \;?\; e_{m-1}$

$z_{1,2}$ $\qquad$ $z_{4,5}$ $\qquad$ $z_{m-5,m-4}$ $\qquad$ $z_{m-2,m-1}$

sort breakpoints

$z_{m-5,m-4} \leq z_{1,2} \leq \cdots \leq z_{4,5} \leq z_{m-2,m-1}$

- Sort breakpoints in $O(m\log(m)\cdot\log(m\log(m)) = O(m\log(m)^2)$
- Binary search with $\texttt{check}(z)$ in $O(\log(m\log(m))\cdot(m+n\log(n))) = O(\log(m)(m+n\log(n)))$

$\Rightarrow$ $\log(m)$ parallel steps, hence the runtime is

$$O(m\log(m)^3 + \log(m)^2(m+n\log(n)))$$

Runtime

m elements

e_1	e_2	e_3	e_4	e_5	e_6	$\cdots$	e_{m-5}	e_{m-4}	e_{m-3}	e_{m-2}	e_{m-1}	e_m

$e_1 \; ? \; e_2$ $\quad$ $e_4 \; ? \; e_5$ $\quad$ $e_{m-5} \; ? \; e_{m-4}$ $\quad$ $e_{m-2} \; ? \; e_{m-1}$

$z_{1,2}$ $\quad$ $z_{4,5}$ $\quad$ $z_{m-5,m-4}$ $\quad$ $z_{m-2,m-1}$

sort breakpoints

$$z_{m-5,m-4} \leq z_{1,2} \leq \cdots \leq z_{4,5} \leq z_{m-2,m-1}$$

- Sort breakpoints in $O(m \log(m) \cdot \log(m \log(m)) = O(m \log(m)^2)$
- Binary search with $\texttt{check}(z)$ in $O(\log(m \log(m)) \cdot (m + n \log(n))) = O(\log(m)(m + n \log(n)))$

$\Rightarrow$ $\log(m)$ parallel steps, hence the runtime is

$$O(m \log(m)^3 + \log(m)^2(m + n \log(n)))$$

- Compute sequence in $O(n \log(n))$ [ST83]

Runtime

m elements

e_1	e_2	e_3	e_4	e_5	e_6	$\cdots$	e_{m-5}	e_{m-4}	e_{m-3}	e_{m-2}	e_{m-1}	e_m

$e_1 \ ? \ e_2$ $\quad$ $e_4 \ ? \ e_5$ $\quad$ $e_{m-5} \ ? \ e_{m-4}$ $\quad$ $e_{m-2} \ ? \ e_{m-1}$

$z_{1,2}$ $\quad$ $z_{4,5}$ $\quad$ $z_{m-5,m-4}$ $\quad$ $z_{m-2,m-1}$

sort breakpoints

$$z_{m-5,m-4} \leq z_{1,2} \leq \cdots \leq z_{4,5} \leq z_{m-2,m-1}$$

- Sort breakpoints in $O(m\log(m) \cdot \log(m\log(m)) = O(m\log(m)^2)$
- Binary search with $\mathtt{check}(z)$ in
  $O(\log(m\log(m)) \cdot (m + n\log(n))) = O(\log(m)(m + n\log(n)))$

$\Rightarrow$ $\log(m)$ parallel steps, hence the runtime is

$$O(m\log(m)^3 + \log(m)^2(m + n\log(n)))$$

- Compute sequence in $O(n\log(n))$ [ST83]

$\Rightarrow$ Total runtime is $O(m\log(m)^3 + \log(m)^2(m + n\log(n)) + n\log(n)) =$

$$O(m\log(m)^3 + n\log(m)^2\log(n)))$$

Table of Contents

Conclusion

Approximation algorithm for the CMST problem:

Conclusion

Approximation algorithm for the CMST problem:

1. Compute Lagrangian relaxation

Conclusion

Approximation algorithm for the CMST problem:

1. Compute Lagrangian relaxation
2. Check Lagrange multiplier via algorithm $\mathtt{check}(z)$

Conclusion

Approximation algorithm for the CMST problem:

1. Compute Lagrangian relaxation
2. Check Lagrange multiplier via algorithm `check`(z)
3. Use Megiddo's algorithm based on parallel-sorting algorithm by Preparata

Conclusion

Approximation algorithm for the CMST problem:

1. Compute Lagrangian relaxation
2. Check Lagrange multiplier via algorithm $\mathtt{check}(z)$
3. Use Megiddo's algorithm based on parallel-sorting algorithm by Preparata
4. Obtain z^* and spanning trees T_{min}, T_{max}

Conclusion

Approximation algorithm for the CMST problem:

1. Compute Lagrangian relaxation
2. Check Lagrange multiplier via algorithm $\texttt{check}(z)$
3. Use Megiddo's algorithm based on parallel-sorting algorithm by Preparata
4. Obtain z^* and spanning trees T_{min}, T_{max}
5. Find approximation via swapping edges

Conclusion

Approximation algorithm for the CMST problem:

1. Compute Lagrangian relaxation
2. Check Lagrange multiplier via algorithm $\texttt{check}(z)$
3. Use Megiddo's algorithm based on parallel-sorting algorithm by Preparata
4. Obtain z^* and spanning trees $T_{\min}$, $T_{\max}$
5. Find approximation via swapping edges

$\Rightarrow$ The result is a $(1, 1+\epsilon)$-approximation with runtime

$$O(m \log(m)^3 + n \log(m)^2 \log(n)))$$

Conclusion

Approximation algorithm for the CMST problem:

1. Compute Lagrangian relaxation
2. Check Lagrange multiplier via algorithm $\texttt{check}(z)$
3. Use Megiddo's algorithm based on parallel-sorting algorithm by Preparata
4. Obtain z^* and spanning trees $T_{\min}$, $T_{\max}$
5. Find approximation via swapping edges

$\Rightarrow$ The result is a $(1, 1+\epsilon)$-approximation with runtime

$$O(m\log(m)^3 + n\log(m)^2\log(n)))$$

Thank you for your attention!

References I

V. Aggarwal, Yash P. Aneja, and K. P. K. Nair.
Minimal spanning tree subject to a side constraint.
Comput. Oper. Res., 9(4):287–296, 1982.

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein.
Introduction to Algorithms, Second Edition.
The MIT Press and McGraw-Hill Book Company, 2001.

Marshall L. Fisher.
The lagrangian relaxation method for solving integer programming problems.
Manag. Sci., 50(12-Supplement):1861–1871, 2004.

References II

Nimrod Megiddo.
Applying parallel computation algorithms in the design of serial algorithms.
J. ACM, 30(4):852–865, 1983.

Franco P. Preparata.
New parallel-sorting schemes.
IEEE Trans. Computers, 27(7):669–673, 1978.

R. Ravi and Michel X. Goemans.
The constrained minimum spanning tree problem (extended abstract).
Algorithm Theory — SWAT, 1097:66–75, 1996.

Daniel Dominic Sleator and Robert Endre Tarjan.
A data structure for dynamic trees.
J. Comput. Syst. Sci., 26(3):362–391, 1983.