The Traveling Salesman: Classical Tools and Recent Advances

Lecture 4: Symmetric s-t Path TSP

Anke van Zuylen
XV Summer School in Discrete Mathematics
Valparaiso, January 6-10, 2020

William & Mary
Chartered 1693

Symmetric s-t Path TSP

Recall: s-t Path TSP

Input:

- A complete undirected graph $G = (V, E)$;
- Start vertex $s \in V$, end vertex $t \in V$;
- Edge costs $c(e) \equiv c(i, j) \geq 0$ for all $e = (i, j) \in E$;
- Edge costs satisfy the **triangle inequality**: $c(i, j) \leq c(i, k) + c(k, j)$ for all i, j, k.

Goal: Find a min-cost path from s to t that visits all other vertices in between.

Recall: LP relaxation

Let $\delta(S)$ be the set of edges with exactly one endpoint in S, and $x(E') \equiv \sum_{e \in E'} x(e)$.

Call $\delta(S)$ an <u>s-t cut</u> if $s \in S, t \notin S$ (or $s \notin S, t \in S$). Call $\delta(S)$ a <u>non s-t cut</u> if $s, t \notin S$ (or $s, t \in S$).

$$\text{Min} \quad \sum_{e \in E} c(e)x(e)$$

subject to:

$$x(\delta(i)) = \begin{cases} 1, & \forall i = s, t, \\ 2, & \forall i \neq s, t, \end{cases}$$

$$x(\delta(S)) \geq \begin{cases} 1, & \forall s\text{-}t \text{ cuts } \delta(S), \\ 2, & \forall \text{non } s\text{-}t \text{ cuts } \delta(S), \end{cases}$$

$$0 \leq x(e) \leq 1, \qquad \forall e \in E.$$

Recall: No Even Narrow Cuts – No Problem

Let x^* be an optimal solution for the s-t path TSP LP.

Definition

A cut $\delta(S)$ is narrow if $x^*(\delta(S)) < 2$.

Observation

Let T be a spanning tree, and $W_T = Odd_T \triangle \{s, t\}$.

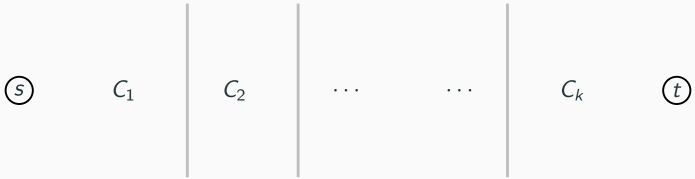
If $c(T) \leq OPT_{LP}$, and there is no narrow cut $\delta(S)$ for which $|\delta(S) \cap T|$ is even, then adding a minimum-cost W_T-matching to T gives an s-t traveling salesman path of cost at most $\frac{3}{2} OPT_{LP}$.

Recall: Narrow Cuts Are Nested

Theorem (An, Kleinberg, Shmoys (2012))

If $\delta(S_1), \delta(S_2)$ are narrow cuts, $S_1 \neq S_2$, then either $S_1 \subset S_2$ or $S_2 \subset S_1$.

So the narrow cuts look like $s \in S_1 \subset S_2 \subset \cdots \subset S_k \subset V$.



Each narrow cut $\delta(S_i)$ is indicated by a gray line; $S_i = \bigcup_{j=1}^{i} C_j$ is all nodes to the left of the line.

An idea!?

Can we just use Gao's algorithm again??!?

✓ Yes, we can find a Gao-tree T_{Gao}:

$$|T_{\mathrm{Gao}} \cap \delta(S)| \text{ is odd for all cuts } \delta(S) \text{ with } x^*(\delta(S)) < 2.$$

An idea!?

Can we just use Gao's algorithm again??!?

✓ Yes, we can find a Gao-tree T_{Gao}:

$$|T_{\mathrm{Gao}} \cap \delta(S)| \text{ is odd for all cuts } \delta(S) \text{ with } x^*(\delta(S)) < 2.$$

✓ That implies that the "cost of parity correction" is at most $\frac{1}{2} OPT_{LP}$:

$$c(M) \leq \frac{1}{2} OPT_{LP},$$

for a minimum-cost $\mathrm{W}_{T_{\mathrm{Gao}}}$-matching M.

An idea!?

Can we just use Gao's algorithm again??!?

✓ Yes, we can find a Gao-tree T_{Gao}:

$$|T_{\mathrm{Gao}} \cap \delta(S)| \text{ is odd for all cuts } \delta(S) \text{ with } x^*(\delta(S)) < 2.$$

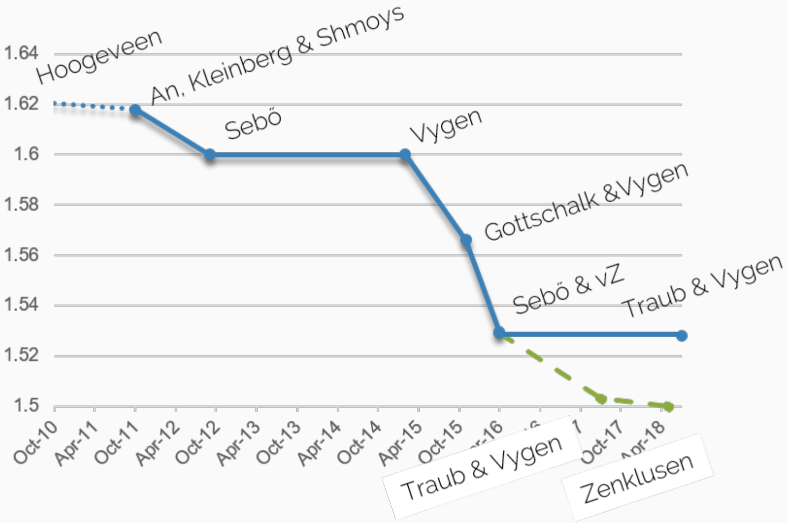
✓ That implies that the "cost of parity correction" is at most $\frac{1}{2} OPT_{LP}$:

$$c(M) \leq \frac{1}{2} OPT_{LP},$$

for a minimum-cost $\mathrm{W}_{T_{\mathrm{Gao}}}$-matching M.

✗ But, unfortunately, Gao (2015) shows that $c(T_{\mathrm{Gao}}) \not\leq OPT_{LP}$...

Recent Developments



Many new ideas

Many interesting ideas in these recent developments:

- An, Kleinberg, Shmoys (2012): Best-of-many trees algorithm;
- Gottschalk, Vygen (2018): Choosing better trees;
- Sebő, vZ (2019): Best-of-many with deletion;
- Traub, Vygen (2019), Zenklusen (2019): Dynamic programming.

We will talk about the first three ideas in this lecture, giving high level ideas and simplified proofs. In tomorrow's lecture, we will describe the last result (giving a full analysis).

Best-of-Many Trees

Convex Combination of Spanning Trees[1]

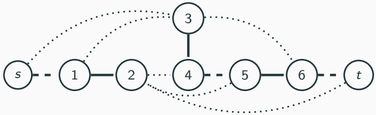
Because an optimal LP solution x^* is in the spanning tree polytope (feasible for the spanning tree LP), we can compute a convex combination of spanning trees

$$x^* = \sum_{i=1}^{k} \lambda_i \chi_{T_i}.$$

$(\sum_{i=1}^{k} \lambda_i = 1, \lambda_i \geq 0$ for $i = 1, \ldots, k.)$

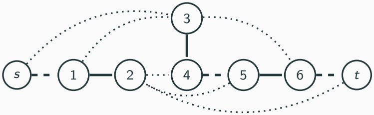
[1]Recall: The characteristic vector of T has $\chi_T(e) = 1$ if $e \in T$, $\chi_T(e) = 0$ if $e \notin T$.

Example: Convex Combination of Spanning Trees



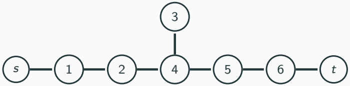
—	$x^*(e) = 1$
- -	$x^*(e) = 2/3$
$\cdots$	$x^*(e) = 1/3$

Example: Convex Combination of Spanning Trees

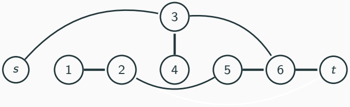


—	$x^*(e) = 1$
- -	$x^*(e) = 2/3$
· · ·	$x^*(e) = 1/3$

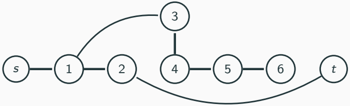
$\lambda_1 = \frac{1}{3}$, $T_1 =$



$\lambda_2 = \frac{1}{3}$, $T_2 =$



$\lambda_3 = \frac{1}{3}$, $T_3 =$



Best-of-Many Algorithm

An, Kleinberg, Shmoys (2012) propose the Best-of-Many Christofides' algorithm: given optimal LP solution x^*, compute convex combination of spanning trees

$$x^* = \sum_{i=1}^{k} \lambda_i \chi_{T_i}.$$

For each spanning tree T_i:

- Let $\mathrm{W}_{T_i} = \mathrm{Odd}_{T_i} \triangle \{s, t\}$ be the set of vertices whose degree parity needs fixing.

Best-of-Many Algorithm

An, Kleinberg, Shmoys (2012) propose the Best-of-Many Christofides' algorithm: given optimal LP solution x^*, compute convex combination of spanning trees

$$x^* = \sum_{i=1}^{k} \lambda_i \chi_{T_i}.$$

For each spanning tree T_i:

- Let $\mathrm{W}_{T_i} = \mathrm{Odd}_{T_i} \triangle \{s, t\}$ be the set of vertices whose degree parity needs fixing.
- Let M_i be a minimum-cost W_{T_i}-matching.

Best-of-Many Algorithm

An, Kleinberg, Shmoys (2012) propose the Best-of-Many Christofides' algorithm: given optimal LP solution x^*, compute convex combination of spanning trees

$$x^* = \sum_{i=1}^{k} \lambda_i \chi_{T_i}.$$

For each spanning tree T_i:

- Let $\mathrm{W}_{T_i} = \mathrm{Odd}_{T_i} \triangle \{s, t\}$ be the set of vertices whose degree parity needs fixing.
- Let M_i be a minimum-cost W_{T_i}-matching.
- Find s-t traveling salesman path by shortcutting Eulerian path of $(V, T_i \sqcup M_i)$.

Best-of-Many Algorithm

An, Kleinberg, Shmoys (2012) propose the Best-of-Many Christofides' algorithm: given optimal LP solution x^*, compute convex combination of spanning trees

$$x^* = \sum_{i=1}^{k} \lambda_i \chi_{T_i}.$$

For each spanning tree T_i:

- Let $\mathrm{W}_{T_i} = \mathrm{Odd}_{T_i} \triangle \{s, t\}$ be the set of vertices whose degree parity needs fixing.
- Let M_i be a minimum-cost W_{T_i}-matching.
- Find s-t traveling salesman path by shortcutting Eulerian path of $(V, T_i \sqcup M_i)$.

Return the shortest traveling salesman path found over all i.

Probabilistic View

Since the algorithm returns the best solution among the solutions based on $T_1, \ldots, T_k$, the cost of the algorithm's solution is at most the (weighted) average cost of these solutions.

For convenience, we view the weighted average cost as an expected value, by considering a random spanning tree $\mathbb{T}$ where $P(\mathbb{T} = T_i) = \lambda_i$ for $i = 1, \ldots, k$, and adding $\mathbb{M}$, a min-cost $\mathrm{W}_{\mathbb{T}}$-matching for the random spanning tree $\mathbb{T}$.

The cost of the algorithm's solution is at most

$$\mathbb{E}(c(\mathbb{T})) + \mathbb{E}(c(\mathbb{M})),$$

where $\mathbb{E}(\cdot)$ indicates the expectation.

Probabilistic View

Since the algorithm returns the best solution among the solutions based on $T_1, \ldots, T_k$, the cost of the algorithm's solution is at most the (weighted) average cost of these solutions.

For convenience, we view the weighted average cost as an expected value, by considering a random spanning tree $\mathbb{T}$ where $P(\mathbb{T} = T_i) = \lambda_i$ for $i = 1, \ldots, k$, and adding $\mathbb{M}$, a min-cost $\mathrm{W}_{\mathbb{T}}$-matching for the random spanning tree $\mathbb{T}$.

The cost of the algorithm's solution is at most

$$\mathbb{E}(c(\mathbb{T})) + \mathbb{E}(c(\mathbb{M})),$$

where $\mathbb{E}(\cdot)$ indicates the expectation.

Observation: $\mathbb{P}(e \in \mathbb{T}) = x^*(e)$, and $\mathbb{E}(|\mathbb{T} \cap \delta(S)|) = x^*(\delta(S))$.

Theorem

Theorem

The Best-of-Many algorithm returns a solution of cost at most $\frac{13}{8} OPT_{LP} = 1.625 OPT_{LP}$.

We will prove the theorem, by proving the following two lemmas.

Lemma (Connectivity Cost)

$$\mathbb{E}(c(\mathbb{T})) = OPT_{LP}.$$

Lemma (Parity Correction Cost)

$$\mathbb{E}(c(\mathbb{M})) \leq \frac{5}{8} OPT_{LP}.$$

Connectivity Cost

$$\mathbb{E}(c(\mathbb{T})) = OPT_{LP}.$$

$$\mathbb{E}(c(\mathbb{T})) = \sum_{e \in E} c(e) \cdot \mathbb{P}(e \in \mathbb{T})$$

$$= \sum_{e \in E} c(e)\, x^*(e) = OPT_{LP}$$

$\square$

Analyzing the Parity Correction Cost

Remember that the obstacle in our analysis is even narrow cuts, i.e., s-t cuts $\delta(S)$ with $x^*(\delta(S)) < 2$ and $|T \cap \delta(S)|$ even.

Analyzing the Parity Correction Cost

Remember that the obstacle in our analysis is <u>even narrow cuts</u>, i.e., s-t cuts $\delta(S)$ with $x^*(\delta(S)) < 2$ and $|T \cap \delta(S)|$ even.

Lemma

For a narrow cut $\delta(S)$,

$$P(|\mathbb{T} \cap \delta(S)| \text{ is odd}) \geq \underline{2 - x^*(\delta(S))}.$$

$$x^*(\delta(S)) = \mathbb{E}(|\mathbb{T} \cap \delta(S)|) = \sum_{k=1}^{|V|-1} k \cdot \mathbb{P}(|\mathbb{T} \cap \delta(S)| = k)$$

$$\geq 1 \cdot \mathbb{P}(|\mathbb{T} \cap \delta(S)| = 1) + 2 \cdot \mathbb{P}(|\mathbb{T} \cap \delta(S)| > 1)$$

$$= \mathbb{P}(|\mathbb{T} \cap \delta(S)| = 1) + 2(1 - \mathbb{P}(|\mathbb{T} \cap \delta(S)| = 1))$$

$$= 2 - \mathbb{P}(|\mathbb{T} \cap \delta(S)| = 1)$$

$$\mathbb{P}(|\mathbb{T} \cap \delta(S)| \text{ is odd}) \geq \mathbb{P}(|\mathbb{T} \cap \delta(S)| = 1) \geq 2 - x^*(\delta(S))$$

$\square$

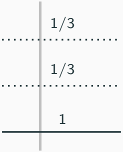
Analyzing the Parity Correction Cost

Let the narrow cuts be $\delta(S_1), \delta(S_2), \ldots, \delta(S_\ell)$.

Approach: Given a tree T, construct a vector z_T that is feasible for the W_T-matching LP:

$$z_T = \frac{1}{2}x^* + \text{ additional vectors,}$$

one for each narrow cut $\delta(S_j)$ such that $|T \cap \delta(S_j)|$ is even



$x^*(\delta(S_j)) = 5/3$

Let $f_j = 2 - x^*(\delta(S_j))$

Then we are

$\frac{1}{2} f_j$ "short"

Analyzing the Parity Correction Cost

Let the narrow cuts be $\delta(S_1), \delta(S_2), \ldots, \delta(S_\ell)$. For each narrow cut, let $f_j = 2 - x^*(\delta(S_j))$.

Let e_j be the cheapest edge in $\delta(S_j)$. For a tree T, we define

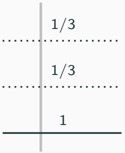
$$z_T = \frac{1}{2}x^* + \sum_{j:|T \cap \delta(S_j)| \text{ is even}} \frac{1}{2} f_j \chi_{e_j}.$$

Analyzing the Parity Correction Cost

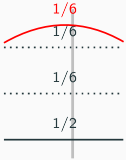
Let the narrow cuts be $\delta(S_1), \delta(S_2), \ldots, \delta(S_\ell)$. For each narrow cut, let $f_j = 2 - x^*(\delta(S_j))$.

Let e_j be the cheapest edge in $\delta(S_j)$. For a tree T, we define

$$z_T = \frac{1}{2}x^* + \sum_{j:|T\cap\delta(S_j)| \text{ is even}} \frac{1}{2} f_j \chi_{e_j}.$$



$x^*(\delta(S)) = 5/3$



$z_T(\delta(S)) = \frac{1}{2} \cdot \frac{5}{3} + \frac{1}{2}(2 - \frac{5}{3}) = 1$
(if $|T \cap \delta(S)|$ is even)

Proof of Second Lemma: $\mathbb{E}(c(\mathbb{M})) \leq \frac{5}{8} OPT_{LP}$

Since

$$z_T = \frac{1}{2}x^* + \sum_{j:|T \cap \delta(S_j)| \text{ is even}} \frac{1}{2} f_j \chi_{e_j}$$

is feasible for the W_T-matching LP for any tree T, we have

$$\mathbb{E}(c(\mathbb{M})) \quad \leq \mathbb{E}(\sum_e c(e) z_{\mathbb{T}}(e))$$

Proof of Second Lemma: $\mathbb{E}(c(\mathbb{M})) \leq \frac{5}{8} OPT_{LP}$

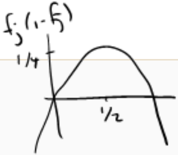
Since

$$z_T = \frac{1}{2}x^* + \sum_{j:|T \cap \delta(S_j)| \text{ is even}} \frac{1}{2} f_j \chi_{e_j}$$

is feasible for the W_T-matching LP for any tree T, we have

$$\begin{aligned} \mathbb{E}(c(\mathbb{M})) &\leq \mathbb{E}(\sum_e c(e) z_{\mathbb{T}}(e)) \\ &= \frac{1}{2} \sum_{e \in E} c(e) x^*(e) + \sum_{j=1}^{\ell} \frac{1}{2} f_j c(e_j) P(|\mathbb{T} \cap \delta(S_j)| \text{ even}). \end{aligned}$$

Proof of Second Lemma: $\mathbb{E}(c(\mathbb{M})) \leq \frac{5}{8} OPT_{LP}$



Since

$$z_T = \frac{1}{2}x^* + \sum_{j:|T\cap\delta(S_j)| \text{ is even}} \frac{1}{2} f_j \chi_{e_j}$$

is feasible for the W$_T$-matching LP for any tree T, we have

$\leq 1 - f_j$

$$\begin{aligned} \mathbb{E}(c(\mathbb{M})) &\leq \mathbb{E}(\sum_e c(e) z_{\mathbb{T}}(e)) \\ &= \frac{1}{2}\sum_{e\in E} c(e)x^*(e) + \sum_{j=1}^{\ell} \frac{1}{2} f_j c(e_j) P(|\mathbb{T}\cap\delta(S_j)| \text{ even}). \end{aligned}$$

We showed that $P(|\mathbb{T}\cap\delta(S_j)| \text{ is odd}) \geq 2 - x^*(\delta(S_j)) = f_j$, so $\mathbb{E}(c(\mathbb{M})) \leq$

$$\frac{1}{2} OPT_{LP} + \sum_{j=1}^{\ell} \frac{1}{2} f_j c(e_j)(1 - f_j) \leq$$

Proof of Second Lemma: $\mathbb{E}(c(\mathbb{M})) \le \frac{5}{8} OPT_{LP}$

Since

$$z_T = \frac{1}{2}x^* + \sum_{j:|T \cap \delta(S_j)| \text{ is even}} \frac{1}{2} f_j \chi_{e_j}$$

is feasible for the W_T-matching LP for any tree T, we have

$$\begin{aligned}
\mathbb{E}(c(\mathbb{M})) &\le \mathbb{E}(\sum_e c(e) z_{\mathbb{T}}(e)) \\
&= \frac{1}{2}\sum_{e \in E} c(e)x^*(e) + \sum_{j=1}^{\ell} \frac{1}{2} f_j c(e_j) P(|\mathbb{T} \cap \delta(S_j)| \text{ even}).
\end{aligned}$$

We showed that $P(|\mathbb{T} \cap \delta(S_j)| \text{ is odd}) \ge 2 - x^*(\delta(S_j)) = f_j$, so
$\mathbb{E}(c(\mathbb{M})) \le$

$$\frac{1}{2} OPT_{LP} + \sum_{j=1}^{\ell} \frac{1}{2} f_j c(e_j)(1 - f_j) \le \frac{1}{2} OPT_{LP} + \frac{1}{8}\sum_{j=1}^{\ell} c(e_j).$$

One Last Lemma

We need one last ingredient.

Lemma (Gao (2015))

$$\sum_{j=1}^{\ell} c(e_j) \leq OPT_{LP}.$$

Proof.

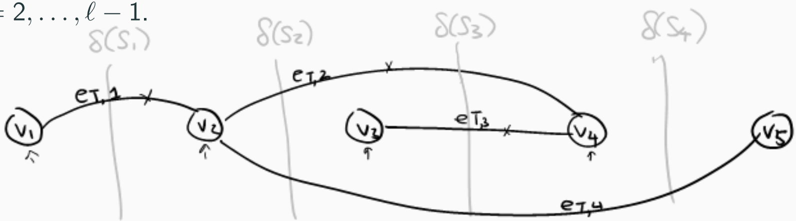
Consider the MST T. We will show that we can assign each narrow cut $\delta(S_j)$ an edge $e_{T,j} \in T \cap \delta(S_j)$ in such a way that no edge in T is assigned to more than one cut.

So $\sum_{j=1}^{\ell} c(e_j) \leq \sum_{j=1}^{\ell} c(e_{T,j}) \leq c(T) \leq OPT_{LP}$.

□

Assigning Edges of T to Narrow Cuts

Consider (V, T). Contract S_1 to v_1, $V \setminus S_\ell$ to v_ℓ, and $S_j \setminus S_{j-1}$ to v_j, for $j = 2, \ldots, \ell - 1$.



Graph is connected; remove edges from T if necessary to ensure T is a spanning tree of contracted graph.

For $j = 1, \ldots, \ell$:

- Let $e_{T,j}$ be the edge incident on v_j on the unique path from v_j to v_{j+1}. Remove $e_{T,j}$ from T, and contract v_j, v_{j+1}.

Improved Analysis

We showed that the solution returned by the Best-of-Many algorithm has cost at most $\frac{13}{8} OPT_{LP} = 1.625 OPT_{LP}$.

An, Kleinberg and Shmoys give a more refined analysis, showing the following result.

Theorem (An, Kleinberg, Shmoys (2012))

The Best-of-Many algorithm returns a solution of cost at most $\frac{1+\sqrt{5}}{2} OPT_{LP} \leq 1.618 OPT_{LP}$.

Improved Analysis

We showed that the solution returned by the Best-of-Many algorithm has cost at most $\frac{13}{8} OPT_{LP} = 1.625 OPT_{LP}$.

An, Kleinberg and Shmoys give a more refined analysis, showing the following result.

Theorem (An, Kleinberg, Shmoys (2012))

The Best-of-Many algorithm returns a solution of cost at most $\frac{1+\sqrt{5}}{2} OPT_{LP} \leq 1.618 OPT_{LP}$.

Their analysis was further improved by Sebő:

Theorem (Sebő (2013))

The Best-of-Many algorithm returns a solution of cost at most $1.6 OPT_{LP}$.

Choosing Better Trees

Reassembling Trees

Vygen shows that exchanging edges in pairs of spanning trees of the convex combination can improve their properties under certain conditions.

Theorem (Vygen (2015))

The Best-of-Many algorithm "with Reassembling of Trees" returns a solution of cost at most $1.599 OPT_{LP}$.

Analysis is complicated, but the idea of reassembly led to the next idea: a Gao-like (or layered) convex combination.

Gao-like Convex Combinations

Given a convex combination $x^* = \sum_{i=1}^{k} \lambda_i \chi_{T_i}$, and a narrow cut $\delta(S)$, we previously showed that

$$P(|\mathbb{T} \cap \delta(S)| \text{ is odd}) \geq \sum_{i:|T_i \cap \delta(S)|=1} \lambda_i \geq 2 - x^*(\delta(S)).$$

Gao-like Convex Combinations

Given a convex combination $x^* = \sum_{i=1}^{k} \lambda_i \chi_{T_i}$, and a narrow cut $\delta(S)$, we previously showed that

$$P(|\mathbb{T} \cap \delta(S)| \text{ is odd}) \geq \sum_{i:|T_i \cap \delta(S)|=1} \lambda_i \geq 2 - x^*(\delta(S)).$$

Call a narrow cut <u>lonely</u> in tree T if $|T \cap \delta(S)| = 1$. Let $f_S = 2 - x^*(\delta(S))$.

The above says that each narrow cut is lonely in an at least an "f_S fraction" of the trees in the convex combination.

Gao-like Convex Combinations

Given a convex combination $x^* = \sum_{i=1}^{k} \lambda_i \chi_{T_i}$, and a narrow cut $\delta(S)$, we previously showed that

$$P(|\mathbb{T} \cap \delta(S)| \text{ is odd}) \geq \sum_{i:|T_i \cap \delta(S)|=1} \lambda_i \geq 2 - x^*(\delta(S)).$$

Call a narrow cut lonely in tree T if $|T \cap \delta(S)| = 1$. Let $f_S = 2 - x^*(\delta(S))$.

The above says that each narrow cut is lonely in an at least an "f_S fraction" of the trees in the convex combination.

Gottschalk and Vygen showed that we can find a convex combination such that $\delta(S)$ is lonely in the "first" f_S fraction of the trees; and that this holds simultaneously for all narrow cuts $\delta(S)$.

Layered Convex Combination

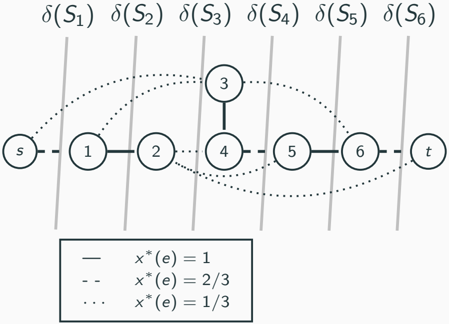
Theorem (Gottschalk, Vygen (2018), Schalekamp, Sebő, Traub, vZ (2018))

There exists spanning trees $T_1, \ldots, T_k$ and multipliers $\lambda_1, \ldots, \lambda_k \geq 0$ such that

$$x^* = \sum_{i=1}^{k} \lambda_i \chi_{T_i},$$

and for any narrow cut $\delta(S)$, there exists ℓ such that $|T_i \cap \delta(S)| = 1$ for $1 \leq i \leq \ell$ and $\sum_{j=1}^{\ell} \lambda_i \geq 2 - x^*(\delta(S))$.

Example



Narrow cuts $\delta(S)$ indicated by gray lines; S is the set of vertices to the left of the line.

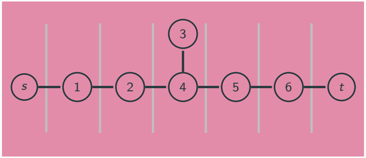
$x^*(\delta(S_j)) = \frac{5}{3}$ for $j = 2, 3, 4, 5$ $\rightarrow$ must be lonely in first $2 - \frac{5}{3} = \frac{1}{3}$ fraction of trees,

$x^*(\delta(S_1)) = x^*(\delta(S_6)) = 1$ $\rightarrow$ must be lonely in first $2 - 1 = 1$ fraction of trees.

Layered Convex Combination

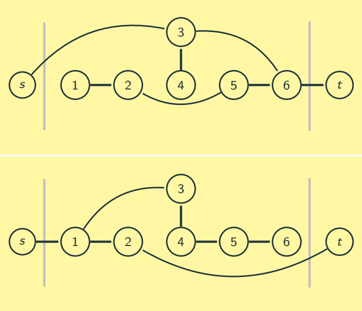
"Layer 1":

- All cuts $\delta(S)$ with $x^*(\delta(S)) < 2$ are lonely in trees in layer 1.
- Weight of layer 1 is ϕ_1.



"Layer 2":

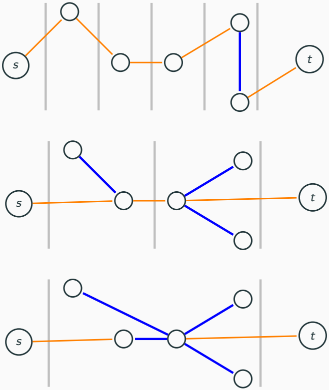
- All cuts $\delta(S)$ with $x^*(\delta(S)) < 2 - \phi_1$ are lonely in layer 2.
- Weight of layer 2 is ϕ_2.



. . .

Layered Trees and Matroids

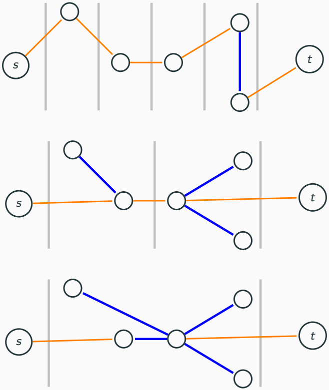
Schalekamp, Sebő, Traub and vZ: Trees in a given layer are bases of a matroid:



- Spanning tree in each "level set", plus
- one edge per lonely cut.

Layered Trees and Matroids

Schalekamp, Sebő, Traub and vZ: Trees in a given layer are bases of a matroid:

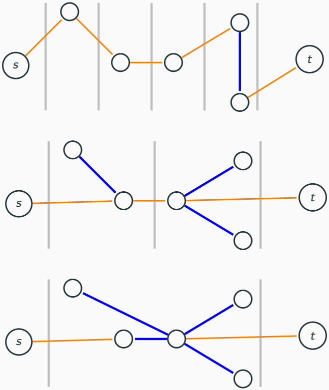


- Spanning tree in each "level set", plus
- one edge per lonely cut.

$\Rightarrow$ simpler proof of the theorem of Gottschalk and Vygen.

Layered Trees and Matroids

Schalekamp, Sebő, Traub and vZ: Trees in a given layer are bases of a matroid:

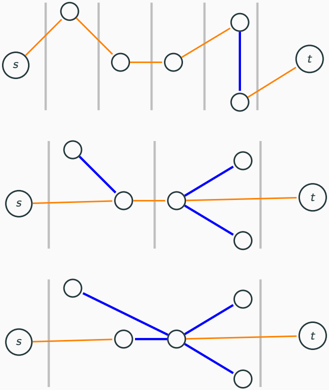


- Spanning tree in each "level set", plus
- one edge per lonely cut.

⇒ simpler proof of the theorem of Gottschalk and Vygen.

⇒ we can use greedy algorithm to find minimum-cost tree for each layer (instead of computing convex combination).

Why Layered Set of Trees May Help in the Analysis



As we go down the layered set:

- Trees are less restrictive $\rightarrow$ tree cost decreasing;
- More narrow cuts may be even $\rightarrow$ parity correction cost increasing.

Best-of-Many on Layered Trees

Theorem (Gottschalk, Vygen (2018))

The Best-of-Many algorithm on a Layered Set of Trees returns a solution of cost at most $1.566 OPT_{LP}$.

Best-of-Many with Deletion

Best-of-Many with Deletion

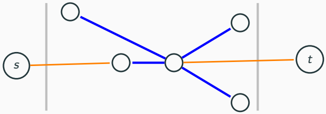
Sebő and vZ (2016) propose the Best-of-Many with Deletion (BOMD) algorithm: given optimal LP solution x^*, and a layered set of trees for x^*, for each spanning tree T_i:

- Delete the edges in the layer's lonely cuts to get a forest F_i.
- Let $\mathrm{W}_{F_i} = \mathrm{Odd}_{F_i} \triangle \{s, t\}$ be the set of vertices whose degree parity needs fixing, and let M_i be a minimum-cost W_{F_i}-matching.
- Add doubled edges D_i in lonely cuts if needed to reconnect $(V, F_i \sqcup M_i)$.
- Find s-t traveling salesman path by shortcutting Eulerian path of $(V, F_i \sqcup M_i \sqcup D_i)$.

Return the shortest traveling salesman path found over all i.

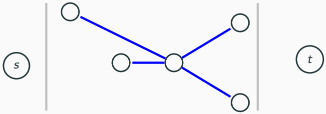
First Example of BOMD

Forest F_i.



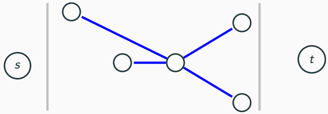
First Example of BOMD

Forest F_i.

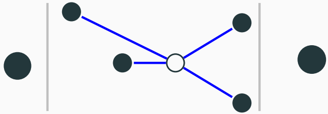


First Example of BOMD

Forest F_i.

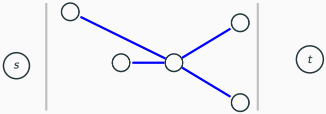


Add parity correction M_i.

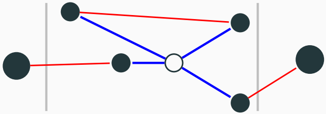


First Example of BOMD

Forest F_i.

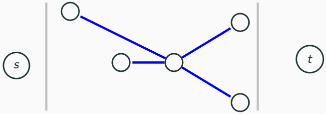


Add parity correction M_i.

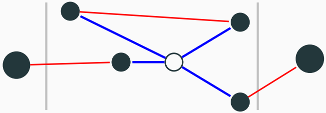


First Example of BOMD

Forest F_i.



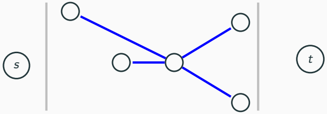
Add parity correction M_i.



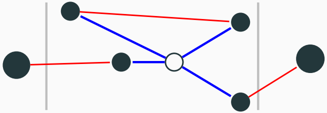
Reconnect if needed.

First Example of BOMD

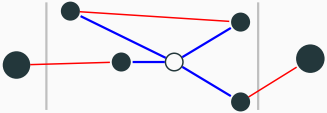
Forest F_i.



Add parity correction M_i.



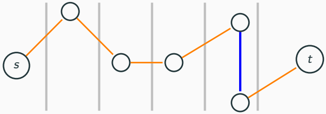
Reconnect if needed.



Parity correction reconnected the forest! (and we show this happens often)

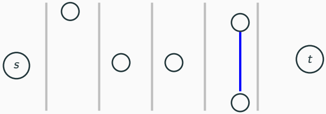
Second Example of BOMD

Forest F_i.



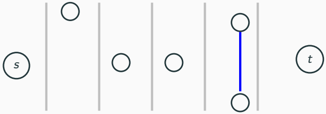
Second Example of BOMD

Forest F_i.

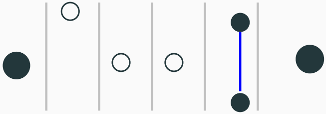


Second Example of BOMD

Forest F_i.

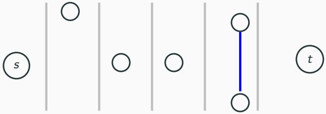


Add parity correction M_i.

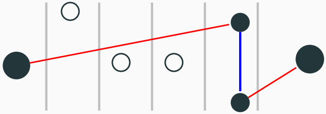


Second Example of BOMD

Forest F_i.

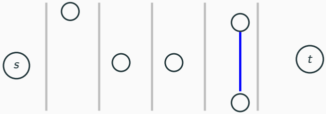


Add parity correction M_i.

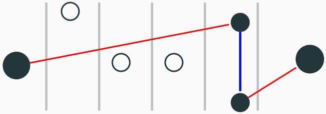


Second Example of BOMD

Forest F_i.



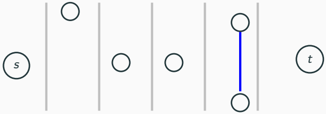
Add parity correction M_i.



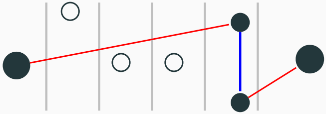
Reconnect if needed.

Second Example of BOMD

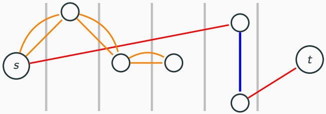
Forest F_i.



Add parity correction M_i.



Reconnect if needed.



Analysis of BOMD

Since we start with a forest instead of a tree, we "save" compared to starting with a spanning tree. The analysis of the cost of parity correction is similar to before. We can prove that parity correction often reconnects the forest, so that the cost of reconnection is small on average.

Theorem (Sebő, vZ (2016))

The Best-of-Many with Deletion algorithm returns a solution of cost at most $\left(\frac{3}{2}+\frac{1}{34}\right) OPT_{LP} < 1.5294 OPT_{LP}$.

Analysis of BOMD

Since we start with a forest instead of a tree, we "save" compared to starting with a spanning tree. The analysis of the cost of parity correction is similar to before. We can prove that parity correction often reconnects the forest, so that the cost of reconnection is small on average.

Theorem (Sebő, vZ (2016))

The Best-of-Many with Deletion algorithm returns a solution of cost at most $\left(\frac{3}{2} + \frac{1}{34}\right) OPT_{LP} < 1.5294 OPT_{LP}$.

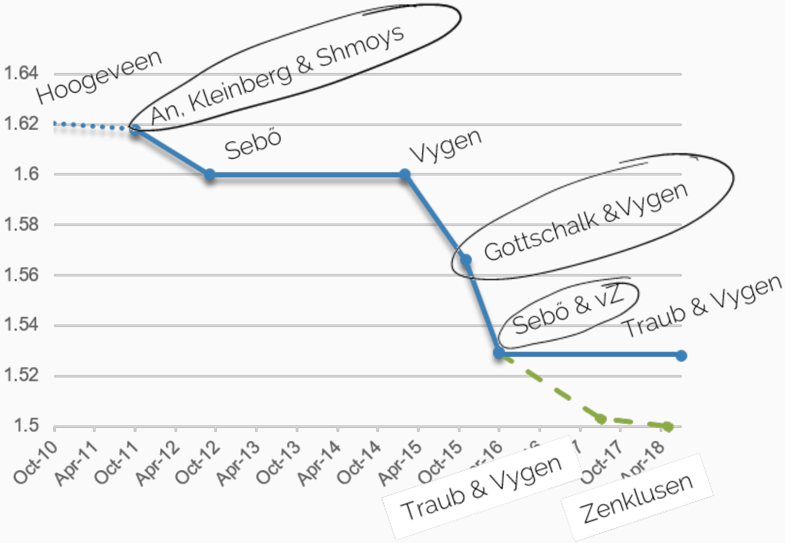
The analysis was improved to:

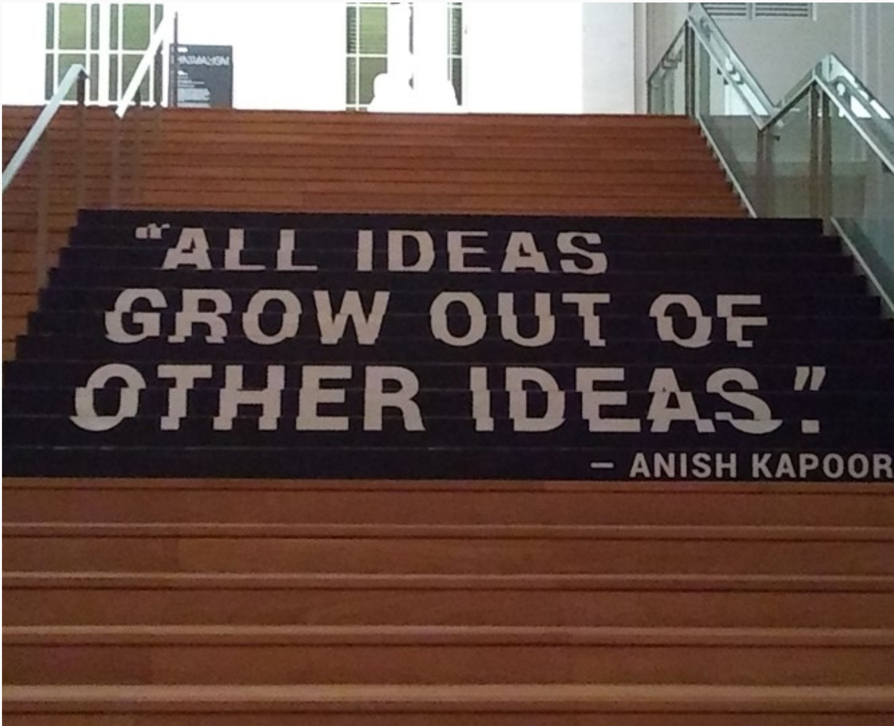
Theorem (Traub, Vygen (2016))

The Best-of-Many with Deletion algorithm returns a solution of cost at most $1.5284 OPT_{LP}$.

Summary

Recent Developments




"ALL IDEAS
GROW OUT OF
OTHER IDEAS."
– ANISH KAPOOR