The Salesman's Improved Paths

A $\frac{3}{2} + \frac{1}{34}$ approximation

Anke van Zuylen

based on joint work with András Sebő

Workshop on Flexible Network Design, Amsterdam

(Metric) $s - t$ Path TSP

Given:

- vertex set V
- special vertices $s, t \in V$
- distance metric $c : \binom{V}{2} \to \mathbb{Q}^{\geq 0}$

Find:

Minimum length path from s to t that visits every vertex in V.

(Metric) $s - t$ Path TSP

Given:

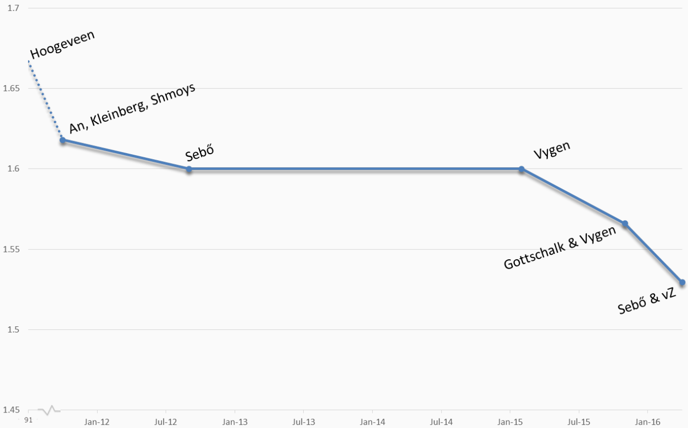
- vertex set V
- special vertices $s, t \in V$
- distance metric $c : \binom{V}{2} \to \mathbb{Q}^{\geq 0}$

Find:

Minimum length path from s to t that visits every vertex in V.

Equivalent to solving $s - t$ path TSP:
Find a minimum cost "$s - t$ tour": a connected multigraph G in which s, t have odd degree and every other vertex has even degree.

$s-t$ Path TSP: Progress for General Metrics



Why does Christofides's algorithm not immediately give a 3/2-approximation?

Christofides for $s - t$ path TSP

Algorithm	Analysis
• Connectivity: find a Minimum Spanning Tree S.	
• Degree parities: add a Min Cost T_S-join J_S:	
$s, t \in T_S$ if they have even degree in S,	
$v \in T_S$, if $v \in V \setminus \{s, t\}$ has odd degree in S.	

Christofides for $s-t$ path TSP

Algorithm	Analysis
• Connectivity: find a Minimum Spanning Tree S.	$c(S) \leq OPT$
• Degree parities: add a Min Cost T_S-join J_S: $s, t \in T_S$ if they have even degree in S, $v \in T_S$, if $v \in V \setminus \{s, t\}$ has odd degree in S.	

Christofides for $s - t$ path TSP

Algorithm	Analysis
• Connectivity: find a Minimum Spanning Tree S.	$c(S) \leq OPT$
• Degree parities: add a Min Cost T_S-join J_S: $s, t \in T_S$ if they have even degree in S, $v \in T_S$, if $v \in V \setminus \{s, t\}$ has odd degree in S.	$c(J_S) \not\leq \frac{1}{2} OPT$!!
	Total ???

LP Based Analysis: Subtour Elimination LP for $s-t$ Path TSP

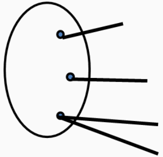
- $x(e)$ is a decision variable indicating if edge e is used in the $s-t$ path TSP

LP Based Analysis: Subtour Elimination LP for $s-t$ Path TSP

- $x(e)$ is a decision variable indicating if edge e is used in the $s-t$ path TSP
- Objective: minimize cost $c(x) = \sum_e c(e)x(e)$

LP Based Analysis: Subtour Elimination LP for $s - t$ Path TSP

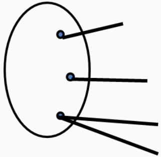
- $x(e)$ is a decision variable indicating if edge e is used in the $s - t$ path TSP
- Objective: minimize cost $c(x) = \sum_e c(e)x(e)$
- Subject to: subtour elimination constraints



Notation: $x(Q) = \sum_{e \in Q} x(e)$

LP Based Analysis: Subtour Elimination LP for $s-t$ Path TSP

- $x(e)$ is a decision variable indicating if edge e is used in the $s-t$ path TSP
- Objective: minimize cost $c(x) = \sum_e c(e)x(e)$
- Subject to: subtour elimination constraints

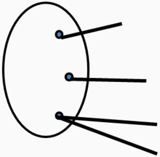


Notation: $x(Q) = \sum_{e \in Q} x(e)$

- At least one edge in an $s-t$ cut

LP Based Analysis: Subtour Elimination LP for $s-t$ Path TSP

- $x(e)$ is a decision variable indicating if edge e is used in the $s-t$ path TSP
- Objective: minimize cost $c(x) = \sum_e c(e)x(e)$
- Subject to: subtour elimination constraints



Notation: $x(Q) = \sum_{e \in Q} x(e)$

- At least one edge in an $s-t$ cut
- At least two edges in a non $s-t$ cut

LP Based Analysis I: Connectivity

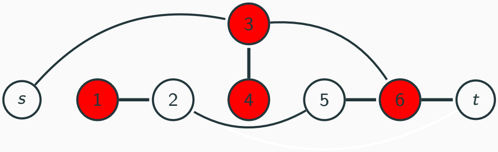
Subtour LP

$$
\begin{aligned}
\min \quad & c(x) \\
\text{s.t.} \quad & x(Q) \geq 2, \text{ for all non } s-t \text{ cuts } Q, \\
& x(Q) \geq 1, \text{ for all } s-t \text{ cuts } Q, \\
& x(\delta(v)) = \begin{cases} 2, & \text{if } v \notin \{s,t\}, \\ 1, & \text{if } v \in \{s,t\}, \end{cases} \\
& x(e) \geq 0, \text{ for all } e.
\end{aligned}
$$

Note: Feasible region is a subset of the spanning tree polytope $\Rightarrow$ $c(S) \leq c(x^*)$.

LP Based Analysis: T_S-Join Polyhedron

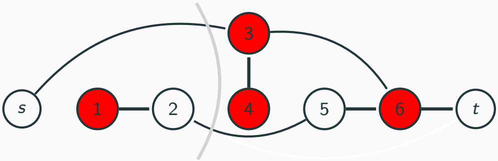
- $y(e)$ is a decision variable indicating whether e is used in the T_S-join, $y(e) \geq 0$
- Objective: minimize cost $c(y) = \sum_e c(e)y(e)$
- Subject to:



at least one edge in $\delta(U)$ if $U \subset V$ has an odd number of vertices in T_S

LP Based Analysis: T_S-Join Polyhedron

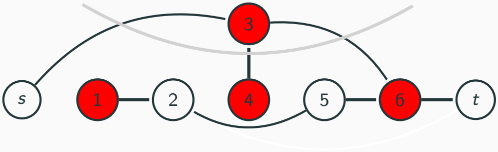
- $y(e)$ is a decision variable indicating whether e is used in the T_S-join, $y(e) \geq 0$
- Objective: minimize cost $c(y) = \sum_e c(e)y(e)$
- Subject to:



at least one edge in $\delta(U)$ if $U \subset V$ has an odd number of vertices in T_S

LP Based Analysis: T_S-Join Polyhedron

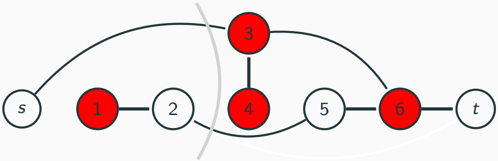
- $y(e)$ is a decision variable indicating whether e is used in the T_S-join, $y(e) \geq 0$
- Objective: minimize cost $c(y) = \sum_e c(e)y(e)$
- Subject to:



at least one edge in $\delta(U)$ if $U \subset V$ has an odd number of vertices in T_S

LP Based Analysis: T_S-Join Polyhedron

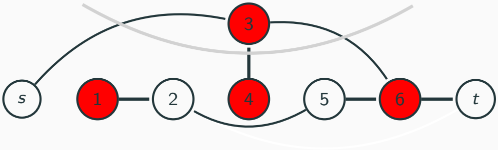
- $y(e)$ is a decision variable indicating whether e is used in the T_S-join, $y(e) \geq 0$
- Objective: minimize cost $c(y) = \sum_e c(e)y(e)$
- Subject to:



at least one edge in cut Q when the tree S does not have the parity in Q that an $s - t$ path TSP solution should have.

LP Based Analysis: T_S-Join Polyhedron

- $y(e)$ is a decision variable indicating whether e is used in the T_S-join, $y(e) \geq 0$
- Objective: minimize cost $c(y) = \sum_e c(e)y(e)$
- Subject to:



at least one edge in cut Q when the tree S does not have the parity in Q that an $s - t$ path TSP solution should have.

LP Based Analysis II: Parity

T_S-join polyhedron

$$\begin{aligned}
\min \quad & c(y) \\
\text{s.t.} \quad & y(Q) \geq 1, \ \text{if } |S \cap Q| \text{ is odd, and } Q \text{ is a non } s-t \text{ cut,} \\
& y(Q) \geq 1, \ \text{if } |S \cap Q| \text{ is even, and } Q \text{ is an } s-t \text{ cut,} \\
& y(e) \geq 0, \ \text{for all } e.
\end{aligned}$$

LP Based Analysis II: Parity

T_S-join polyhedron

$$
\begin{aligned}
\min \quad & c(y) \\
\text{s.t.} \quad & y(Q) \geq 1, \text{ if } |S \cap Q| \text{ is odd, and } Q \text{ is a non } s-t \text{ cut}, \\
& y(Q) \geq 1, \text{ if } |S \cap Q| \text{ is even, and } Q \text{ is an } s-t \text{ cut}, \\
& y(e) \geq 0, \text{ for all } e.
\end{aligned}
$$

Note: T_S-join polyhedron has integer extreme points $\Rightarrow$ Can upper bound cost of minimum T_S-join by cost of any feasible y-vector.

LP Based Analysis: Summary

Subtour LP

$$
\begin{aligned}
\min \quad & c(x) \\
\text{s.t.} \quad & x(Q) \geq 2, \text{ for all non } s-t \text{ cuts } Q, \\
& x(Q) \geq 1, \text{ for all } s-t \text{ cuts } Q, \\
& x(\delta(v)) = \begin{cases} 2, & \text{if } v \notin \{s,t\}, \\ 1, & \text{if } v \in \{s,t\}, \end{cases} \\
& x(e) \geq 0, \text{ for all } e.
\end{aligned}
$$

T_S-join polyhedron

$$
\begin{aligned}
\min \quad & c(y) \\
\text{s.t.} \quad & y(Q) \geq 1, && \text{if } |S \cap Q| \text{ is odd,} \\
& && \text{and } Q \text{ is a non } s-t \text{ cut,} \\
& y(Q) \geq 1, && \text{if } |S \cap Q| \text{ is even,} \\
& && \text{and } Q \text{ is an } s-t \text{ cut,} \\
& y(e) \geq 0, && \text{for all } e.
\end{aligned}
$$

LP Based Analysis: Summary

Subtour LP

$$\begin{aligned} \min \quad & c(x) \\ \text{s.t.} \quad & x(Q) \geq 2, \text{ for all non } s-t \text{ cuts } Q, \\ & x(Q) \geq 1, \text{ for all } s-t \text{ cuts } Q, \\ & x(\delta(v)) = \begin{cases} 2, & \text{if } v \notin \{s,t\}, \\ 1, & \text{if } v \in \{s,t\}, \end{cases} \\ & x(e) \geq 0, \text{ for all } e. \end{aligned}$$

T_S-join polyhedron

$$\begin{aligned} \min \quad & c(y) \\ \text{s.t.} \quad & y(Q) \geq 1, \quad \text{if } |S \cap Q| \text{ is odd, and } Q \text{ is a non } s-t \text{ cut}, \\ & y(Q) \geq 1, \quad \text{if } |S \cap Q| \text{ is even, and } Q \text{ is an } s-t \text{ cut}, \\ & y(e) \geq 0, \quad \text{for all } e. \end{aligned}$$

Connectivity: Cost of Minimum Spanning Tree S is upper bounded by $c(x^*)$.

LP Based Analysis: Summary

Subtour LP

$$\begin{aligned} \min \quad & c(x) \\ \text{s.t.} \quad & x(Q) \geq 2, \text{ for all non } s-t \text{ cuts } Q, \\ & x(Q) \geq 1, \text{ for all } s-t \text{ cuts } Q, \\ & x(\delta(v)) = \begin{cases} 2, & \text{if } v \notin \{s,t\}, \\ 1, & \text{if } v \in \{s,t\}, \end{cases} \\ & x(e) \geq 0, \text{ for all } e. \end{aligned}$$

T_S-join polyhedron

$$\begin{aligned} \min \quad & c(y) \\ \text{s.t.} \quad & y(Q) \geq 1, && \text{if } |S \cap Q| \text{ is odd,} \\ & && \text{and } Q \text{ is a non } s-t \text{ cut,} \\ & y(Q) \geq 1, && \text{if } |S \cap Q| \text{ is even,} \\ & && \text{and } Q \text{ is an } s-t \text{ cut,} \\ & y(e) \geq 0, && \text{for all } e. \end{aligned}$$

Connectivity: Cost of Minimum Spanning Tree S is upper bounded by $c(x^*)$.

Parity: Cost of minimum T_S-join J_S is equal to $c(y^*)$.

LP Based Analysis: Summary

Subtour LP

$$
\begin{aligned}
\min \quad & c(x) \\
\text{s.t.} \quad & x(Q) \geq 2, \text{ for all non } s-t \text{ cuts } Q, \\
& x(Q) \geq 1, \text{ for all } s-t \text{ cuts } Q, \\
& x(\delta(v)) = \begin{cases} 2, & \text{if } v \notin \{s,t\}, \\ 1, & \text{if } v \in \{s,t\}, \end{cases} \\
& x(e) \geq 0, \text{ for all } e.
\end{aligned}
$$

T_S-join polyhedron

$$
\begin{aligned}
\min \quad & c(y) \\
\text{s.t.} \quad & y(Q) \geq 1, && \text{if } |S \cap Q| \text{ is odd,} \\
& && \text{and } Q \text{ is a non } s-t \text{ cut,} \\
& y(Q) \geq 1, && \text{if } |S \cap Q| \text{ is even,} \\
& && \text{and } Q \text{ is an } s-t \text{ cut,} \\
& y(e) \geq 0, && \text{for all } e.
\end{aligned}
$$

Connectivity: Cost of Minimum Spanning Tree S is upper bounded by $c(x^*)$.

Parity: Cost of minimum T_S-join J_S is equal to $c(y^*)$.

Wishful thinking: if $y = x^*/2$ is feasible for the T_S-join polyhedron, then $c(y^*) \leq \frac{1}{2}c(x^*)$.

Conclusion: no $3/2$ yet because...

$x^*/2$ will violate the join constraints for $s-t$ cuts Q such that

- Parity of tree S in Q is "wrong": S has an *even* number of edges in Q, and
- Q is "narrow": $x^*(Q) < 2$.

Conclusion: no $3/2$ yet because...

$x^*/2$ will violate the join constraints for $s - t$ cuts Q such that

- Parity of tree S in Q is "wrong": S has an *even* number of edges in Q, and
- Q is "narrow": $x^*(Q) < 2$.

An, Kleinberg and Shmoys noted that narrow cuts form a "chain".

Conclusion: no $3/2$ yet because...

$x^*/2$ will violate the join constraints for $s - t$ cuts Q such that

- Parity of tree S in Q is "wrong": S has an *even* number of edges in Q, and
- Q is "narrow": $x^*(Q) < 2$.

An, Kleinberg and Shmoys noted that narrow cuts form a "chain".

Gottschalk and Vygen showed that x^* can be decomposed into spanning trees S such that *for each narrow cut* Q, the "first" $2 - x^*(Q)$ fraction of the trees has a single edge in $S \cap Q$.

Conclusion: no $3/2$ yet because...

$x^*/2$ will violate the join constraints for $s - t$ cuts Q such that

- Parity of tree S in Q is "wrong": S has an *even* number of edges in Q, and
- Q is "narrow": $x^*(Q) < 2$.

An, Kleinberg and Shmoys noted that narrow cuts form a "chain".

Gottschalk and Vygen showed that x^* can be decomposed into spanning trees S such that *for each narrow cut* Q, the "first" $2 - x^*(Q)$ fraction of the trees has a single edge in $S \cap Q$.

Let the *deficit* for a narrow cut Q be $\left(\frac{2-x^*(Q)}{2}\right)$ "parts-of-an-edge" if $|S \cap Q|$ is even,

Conclusion: no $3/2$ yet because...

$x^*/2$ will violate the join constraints for $s-t$ cuts Q such that

- Parity of tree S in Q is "wrong": S has an *even* number of edges in Q, and
- Q is "narrow": $x^*(Q) < 2$.

An, Kleinberg and Shmoys noted that narrow cuts form a "chain".

Gottschalk and Vygen showed that x^* can be decomposed into spanning trees S such that *for each narrow cut* Q, the "first" $2 - x^*(Q)$ fraction of the trees has a single edge in $S \cap Q$.

Let the *deficit* for a narrow cut Q be $\left(\frac{2-x^*(Q)}{2}\right)$ "parts-of-an-edge" if $|S \cap Q|$ is even, e.g. an $s-t$ cut Q with $x^*(Q) = \frac{3}{2}$ has deficit $\frac{1}{4}$ if S is even across Q.

How to deal with even narrow cuts?

[AKS'12, S'12, V'14, GV'15]: **Best-of-Many Christofides**

- ✓ Connectivity: Decompose x^* into a collection of spanning trees $\mathcal{S}$ (with multipliers $\lambda_S > 0, \sum_{S\in\mathcal{S}} \lambda_S = 1$).
- ✗ Parity: For each $S \in \mathcal{S}$: add a min cost T_S-join

[Gao'13]: **Gao tree** (for graph $s - t$ path TSP)

- ✗ Connectivity: Compute a minimum cost tree S_{Gao} that has $|S_{Gao} \cap Q| = 1$ *for each narrow cut* Q.
- ✓ Parity: Add a $T_{S_{Gao}}$-join of cost at most $c(x^*)/2$.

How to deal with even narrow cuts?

[AKS'12, S'12, V'14, GV'15]: **Best-of-Many Christofides**

✓ Connectivity: Decompose x^* into a collection of spanning trees $\mathcal{S}$ (with multipliers $\lambda_S > 0, \sum_{S \in \mathcal{S}} \lambda_S = 1$).

✗ Parity: For each $S \in \mathcal{S}$: add a min cost T_S-join

[Gao'13]: **Gao tree** (for graph $s - t$ path TSP)

✗ Connectivity: Compute a minimum cost tree S_{Gao} that has $|S_{Gao} \cap Q| = 1$ *for each narrow cut* Q.

✓ Parity: Add a $T_{S_{Gao}}$-join of cost at most $c(x^*)/2$.

[This work]: **Best-of-Many with Deletion**

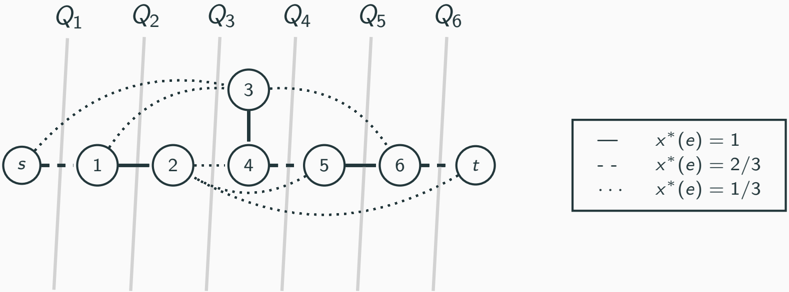
Main idea: For each cut Q, we

- "save" (compared to $c(S) + c(x^*)/2$) on trees S when $|S \cap Q| = 1$
- use the "savings" to make up for Q's deficit in trees S' when $|S' \cap Q|$ is even.

Best-of-Many with Deletion

- Decompose x^* into a collection of spanning trees $\mathcal{S}$ (with multipliers $\lambda_S > 0, \sum_{S \in \mathcal{S}} \lambda_S = 1$) using the Gottschalk-Vygen decomposition.
  - Gottschalk-Vygen guarantees: for narrow cut Q, $|S \cap Q| = 1$ in the "first" $2 - x^*(Q)$ fraction of the trees in the decomposition.
  - In those trees, we will call the edge in $S \cap Q$ "lonely"
- For each $S \in \mathcal{S}$:
  - Delete the "lonely edges"
  - Correct parity (with a join chosen from a specific distribution)
  - Correct connectivity: add back doubled lonely edges if needed.
- Output the solution with minimum cost.

Illustration: LP Solution



Example from Gao'15:

- Narrow cuts are indicated by gray lines and labeled $Q_1, \ldots, Q_6$.
- $x^*(Q_1) = x^*(Q_6) = 1, x^*(Q_i) = \frac{5}{3}$ for $i = 2, 3, 4, 5$.

Illustration: Gottschalk-Vygen Decomposition into Trees

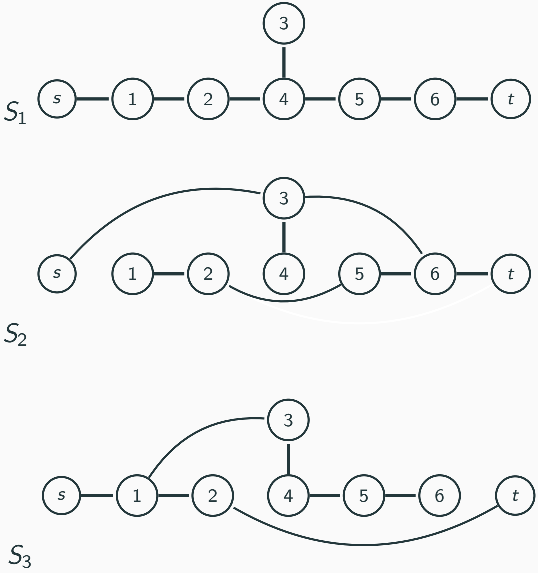


Illustration: Algorithm Executed on Gao-Tree S_1

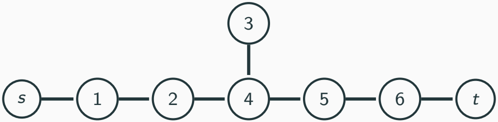
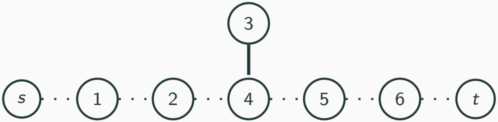
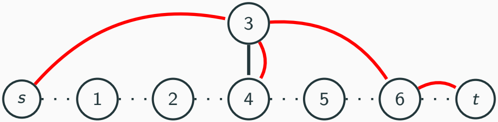


Illustration: Algorithm Executed on Gao-Tree S_1



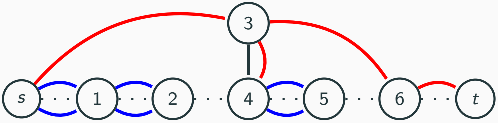
- Black edges form the forest after deletion (dotted edges are deleted lonely edges).

Illustration: Algorithm Executed on Gao-Tree S_1



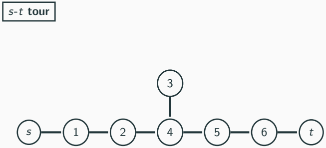
- Black edges form the forest after deletion (dotted edges are deleted lonely edges).
- Red edges correct parity.

Illustration: Algorithm Executed on Gao-Tree S_1



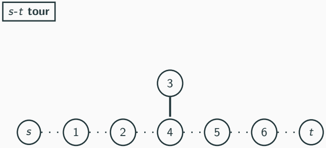
- Black edges form the forest after deletion (dotted edges are deleted lonely edges).
- Red edges correct parity.
- Blue edges correct connectivity.

Analysis of Solution Based on S_1



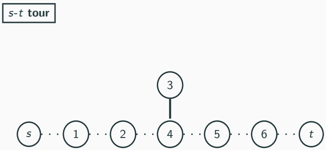
Forest	S_1
Parity Correction	
Connectivity Correction	
Total	

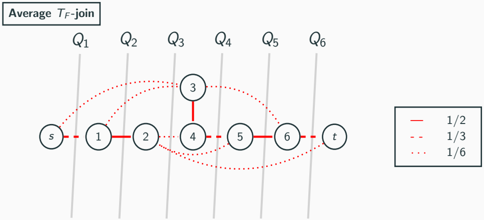
Analysis of Solution Based on S_1



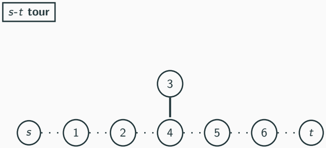
Forest	S_1	$-L_1$
Parity Correction		
Connectivity Correction		
Total		

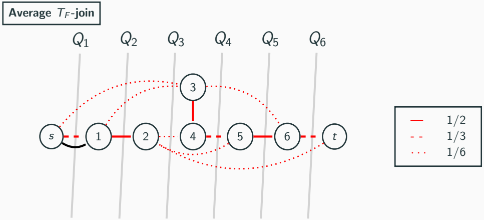
Analysis of Solution Based on S_1





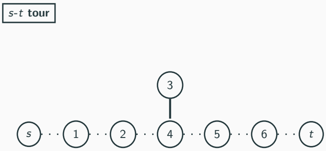
Forest	S_1	$-L_1$
Parity Correction (average)	$\frac{x^*}{2}+$	
Connectivity Correction		
Total		

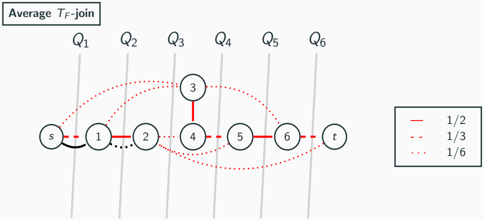




Forest	S_1		$-L_1$
Parity Correction (average)		$\frac{x^*}{2}+$	$\frac{1}{2}(L_1 \cap Q_1)+$
Connectivity Correction			
Total			

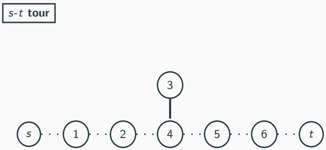
Analysis of Solution Based on S_1

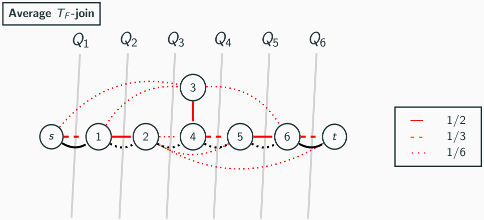




Forest	S_1		$-L_1$
Parity Correction (average)		$\frac{x^*}{2}+$	$\frac{1}{2}(L_1 \cap Q_1)+$ $\frac{1}{6}(L_1 \cap Q_2)+\dots$
Connectivity Correction			
Total			

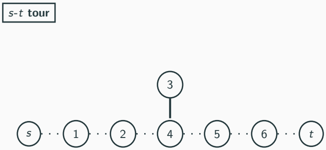
Analysis of Solution Based on S_1

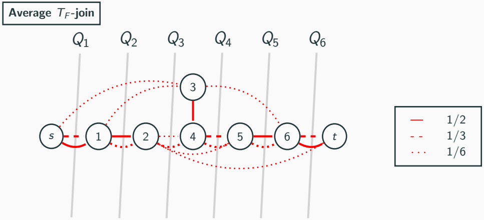




Forest	S_1	$-L_1$
Parity Correction (average)	$\frac{x^*}{2}+$	$\frac{1}{2}(L_1 \cap Q_1)+ \ \frac{1}{6}(L_1 \cap Q_2)+\dots$
Connectivity Correction		
Total		

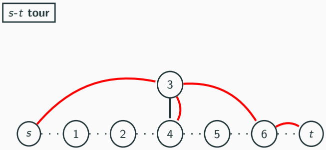
Analysis of Solution Based on S_1

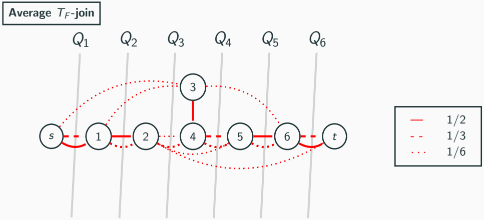




Forest	S_1	$-L_1$
Parity Correction (average)	$\frac{x^*}{2}+$	$\sum_{i=1}^{6} \frac{2-x^*(Q_i)}{2}(L_1 \cap Q_i)$
Connectivity Correction		
Total		

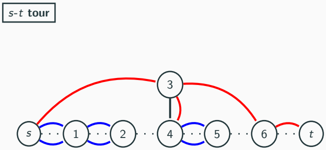
Analysis of Solution Based on S_1

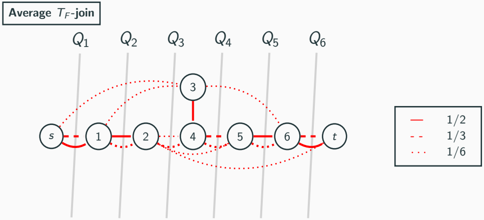




Forest	S_1	$-L_1$
Parity Correction (average)	$\frac{x^*}{2}+$	$\sum_{i=1}^{6} \frac{2-x^*(Q_i)}{2}(L_1 \cap Q_i)$
Connectivity Correction		
Total		

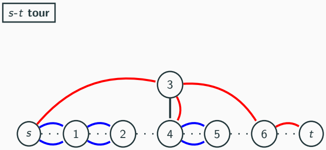
Analysis of Solution Based on S_1

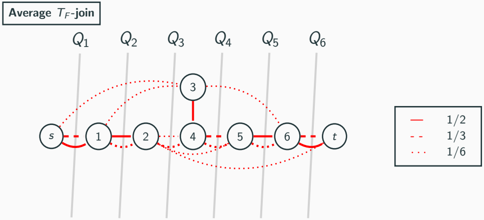




Forest	S_1	$-L_1$
Parity Correction (average)	$\frac{x^*}{2}+$	$\sum_{i=1}^{6} \frac{2-x^*(Q_i)}{2}(L_1 \cap Q_i)$
Connectivity Correction		
Total		

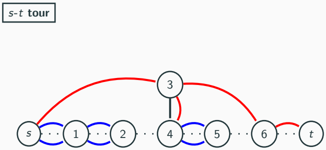
Analysis of Solution Based on S_1

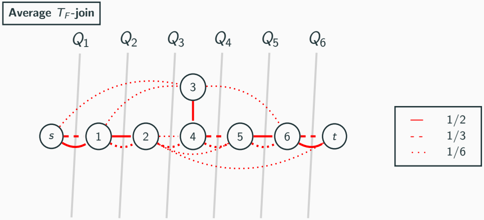




Forest	S_1		$-L_1$
Parity Correction (average)		$\frac{x^*}{2}+$	$\sum_{i=1}^{6} \frac{2-x^*(Q_i)}{2}(L_1 \cap Q_i)$
Connectivity Correction			$\sum_{i=1}^{6} (x^*(Q_i)-1)(L_1 \cap Q_i)$
Total			

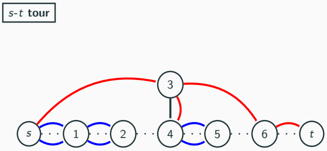
Analysis of Solution Based on S_1

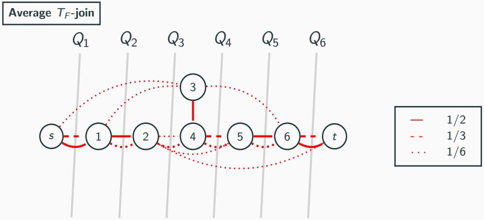




Forest	S_1		$-\sum_{i=1}^{6}(L_1 \cap Q_i)$
Parity Correction (average)		$\frac{x^*}{2}+$	$\sum_{i=1}^{6} \frac{2-x^*(Q_i)}{2}(L_1 \cap Q_i)$
Connectivity Correction			$\sum_{i=1}^{6}\left(x^*(Q_i)-1\right)(L_1 \cap Q_i)$
Total			

Analysis of Solution Based on S_1





Forest	S_1		$-\sum_{i=1}^{6}(L_1 \cap Q_i)$
Parity Correction (average)		$\frac{x^*}{2}+$	$\sum_{i=1}^{6}\frac{2-x^*(Q_i)}{2}(L_1 \cap Q_i)$
Connectivity Correction			$\sum_{i=1}^{6}(x^*(Q_i)-1)(L_1 \cap Q_i)$
Total	$S_1+\frac{x^*}{2}$		$-\sum_{i=1}^{6}\frac{2-x^*(Q_i)}{2}(L_1 \cap Q_i)$

How deletion helps

Consider a given narrow cut Q:

- Can show in general:
  If S has a lonely edge in Q, deletion allows us to "save" $\left(\frac{2-x^*(Q)}{2}\right)$-parts-of-an-edge from the total $S + \frac{x^*}{2}$.

How deletion helps

Consider a given narrow cut Q:

- Can show in general:
  If S has a lonely edge in Q, deletion allows us to “save” $\left(\frac{2-x^*(Q)}{2}\right)$-parts-of-an-edge from the total $S + \frac{x^*}{2}$.
- Recall:
  If S has an even number of edges, $x^*/2$ has “deficit” $\left(\frac{2-x^*(Q)}{2}\right)$-parts-of-an-edge for parity correction.

How deletion helps

Consider a given narrow cut Q:

- Can show in general:
  If S has a lonely edge in Q, deletion allows us to "save" $\left(\frac{2-x^*(Q)}{2}\right)$-parts-of-an-edge from the total $S + \frac{x^*}{2}$.
- Recall:
  If S has an even number of edges, $x^*/2$ has "deficit" $\left(\frac{2-x^*(Q)}{2}\right)$-parts-of-an-edge for parity correction.
- Hence, if the (weighted) fraction of trees in which Q has a lonely edge is at most the (weighted) fraction of trees in which Q has an even number of edges, the "saved edge-parts" can pay for the "deficits".

How deletion helps

Consider a given narrow cut Q:

- Can show in general:
  If S has a lonely edge in Q, deletion allows us to "save" $\left(\frac{2-x^*(Q)}{2}\right)$-parts-of-an-edge from the total $S + \frac{x^*}{2}$.
- Recall:
  If S has an even number of edges, $x^*/2$ has "deficit" $\left(\frac{2-x^*(Q)}{2}\right)$-parts-of-an-edge for parity correction.
- Hence, if the (weighted) fraction of trees in which Q has a lonely edge is at most the (weighted) fraction of trees in which Q has an even number of edges (guaranteed if $x^*(Q) \leq \frac{3}{2}$), the "saved edge-parts" can pay for the "deficits".

How deletion helps

Consider a given narrow cut Q:

- Can show in general:
  If S has a lonely edge in Q, deletion allows us to "save" $\left(\frac{2-x^*(Q)}{2}\right)$-parts-of-an-edge from the total $S + \frac{x^*}{2}$.
- Recall:
  If S has an even number of edges, $x^*/2$ has "deficit" $\left(\frac{2-x^*(Q)}{2}\right)$-parts-of-an-edge for parity correction.
- Hence, if the (weighted) fraction of trees in which Q has a lonely edge is at most the (weighted) fraction of trees in which Q has an even number of edges (guaranteed if $x^*(Q) \leq \frac{3}{2}$), the "saved edge-parts" can pay for the "deficits".

Theorem (SvZ'16)

Let x^* be an optimal solution to the subtour LP, such that no cut Q has $x^*(Q) \in (\frac{3}{2}, 2)$. Then there exists an $s - t$ tour of cost at most $\frac{3}{2}c(x^*)$.

How deletion helps

Consider a given narrow cut Q:

- Can show in general:
  If S has a lonely edge in Q, deletion allows us to "save" $\left(\frac{2-x^*(Q)}{2}\right)$-parts-of-an-edge from the total $S + \frac{x^*}{2}$.
- Recall:
  If S has an even number of edges, $x^*/2$ has "deficit" $\left(\frac{2-x^*(Q)}{2}\right)$-parts-of-an-edge for parity correction.

Problem: Q may save less often than it has a deficit...

How deletion helps

Consider a given narrow cut Q:

- Can show in general:
  If S has a lonely edge in Q, deletion allows us to "save" $\left(\frac{2-x^*(Q)}{2}\right)$-parts-of-an-edge from the total $S + \frac{x^*}{2}$.
- Recall:
  If S has an even number of edges, $x^*/2$ has "deficit" $\left(\frac{2-x^*(Q)}{2}\right)$-parts-of-an-edge for parity correction.

Problem: Q may save less often than it has a deficit...

Solution: Increase savings and decrease deficits by increasing our base line:
$S + \frac{x^*}{2} \rightarrow S + \frac{x^*}{2} + \gamma$ times the incidence vector of the $s - t$ path in S.

How deletion helps

Consider a given narrow cut Q:

- Can show in general:
  If S has a lonely edge in Q, deletion allows us to "save" $\left(\frac{2-x^*(Q)}{2}\right)$-parts-of-an-edge from the total $S + \frac{x^*}{2} + \gamma S(s,t)$.
- Recall:
  If S has an even number of edges, $x^*/2$ has "deficit" $\left(\frac{2-x^*(Q)}{2}\right)$-parts-of-an-edge for parity correction.

Problem: Q may save less often than it has a deficit...

Solution: Increase savings and decrease deficits by increasing our base line:
$S + \frac{x^*}{2} \rightarrow S + \frac{x^*}{2} + \gamma$ times the incidence vector of the $s - t$ path in S.

How deletion helps

Consider a given narrow cut Q:

- Can show in general:
  If S has a lonely edge in Q, deletion allows us to "save" $\left(\frac{2-x^*(Q)}{2} + \gamma\right)$-parts-of-an-edge from the total $S + \frac{x^*}{2} + \gamma S(s,t)$.
- Recall:
  If S has an even number of edges, $x^*/2$ has "deficit" $\left(\frac{2-x^*(Q)}{2}\right)$-parts-of-an-edge for parity correction.

Problem: Q may save less often than it has a deficit...

Solution: Increase savings and decrease deficits by increasing our base line:
$S + \frac{x^*}{2} \rightarrow S + \frac{x^*}{2} + \gamma$ times the incidence vector of the $s - t$ path in S.

How deletion helps

Consider a given narrow cut Q:

- Can show in general:
  If S has a lonely edge in Q, deletion allows us to "save" $\left(\frac{2-x^*(Q)}{2} + \gamma\right)$-parts-of-an-edge from the total $S + \frac{x^*}{2} + \gamma S(s,t)$.
- Recall:
  If S has an even number of edges, $x^*/2$ has "deficit" $\left(\frac{2-x^*(Q)}{2} - \gamma\right)$-parts-of-an-edge for parity correction.

Problem: Q may save less often than it has a deficit...

Solution: Increase savings and decrease deficits by increasing our base line:
$S + \frac{x^*}{2} \to S + \frac{x^*}{2} + \gamma$ times the incidence vector of the $s - t$ path in S.

How deletion helps

Consider a given narrow cut Q:

- Can show in general:
  If S has a lonely edge in Q, deletion allows us to "save" $\left(\frac{2-x^*(Q)}{2} + \gamma\right)$-parts-of-an-edge from the total $S + \frac{x^*}{2} + \gamma S(s,t)$.
- Recall:
  If S has an even number of edges, $x^*/2$ has "deficit" $\left(\frac{2-x^*(Q)}{2} - \gamma\right)$-parts-of-an-edge for parity correction.

Problem: Q may save less often than it has a deficit...

Solution: Increase savings and decrease deficits by increasing our base line:
$S + \frac{x^*}{2} \to S + \frac{x^*}{2} + \gamma$ times the incidence vector of the $s-t$ path in S.

Can show: If $\gamma = \frac{1}{16}$, then savings can always pay for the deficits.

Improved Result for General Case

Lemma (SvZ'16)
Let x^* be an optimal solution to the subtour LP, and given a decomposition of x^* into spanning trees $\mathcal{S}$, let p^* be the average of the incidence vectors of the $s-t$ path in $\mathcal{S}$. There exists an $s-t$ tour of cost at most $\frac{3}{2}c(x^*) + \frac{1}{16}c(p^*)$.

Improved Result for General Case

Lemma (SvZ'16)

Let x^* be an optimal solution to the subtour LP, and given a decomposition of x^* into spanning trees $\mathcal{S}$, let p^* be the average of the incidence vectors of the $s-t$ path in $\mathcal{S}$. There exists an $s-t$ tour of cost at most $\frac{3}{2}c(x^*) + \frac{1}{16}c(p^*)$.

Observation (Hoogeveen'91, Sebő'12)

Let x^* be an optimal solution to the subtour LP, and given a decomposition of x^* into spanning trees $\mathcal{S}$, let p^* be the average of the incidence vectors of the $s-t$ path in $\mathcal{S}$. There exists an $s-t$ tour of cost at most $2c(x^*) - c(p^*)$.

Improved Result for General Case

Lemma (SvZ'16)

Let x^* be an optimal solution to the subtour LP, and given a decomposition of x^* into spanning trees $\mathcal{S}$, let p^* be the average of the incidence vectors of the $s-t$ path in $\mathcal{S}$. There exists an $s-t$ tour of cost at most $\frac{3}{2}c(x^*) + \frac{1}{16}c(p^*)$.

Observation (Hoogeveen'91, Sebő'12)

Let x^* be an optimal solution to the subtour LP, and given a decomposition of x^* into spanning trees $\mathcal{S}$, let p^* be the average of the incidence vectors of the $s-t$ path in $\mathcal{S}$. There exists an $s-t$ tour of cost at most $2c(x^*) - c(p^*)$.

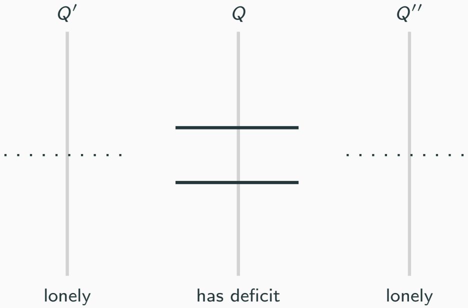
The minimum of the above two tours has cost at most $\left(\frac{3}{2} + \frac{1}{34}\right) c(x^*)$.

What I did not show you (but did use):

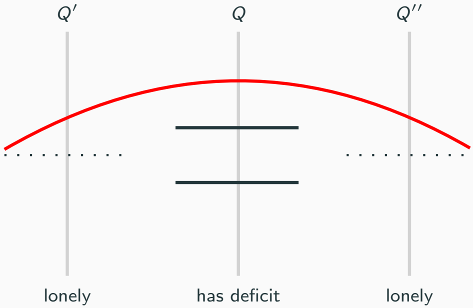
how to prove that Connectivity Correction can be done using at most

$$\sum_{Q \text{ lonely in S}} (x^*(Q) - 1)(L \cap Q)$$

A few details on the proof: need for Gottschalk-Vygen

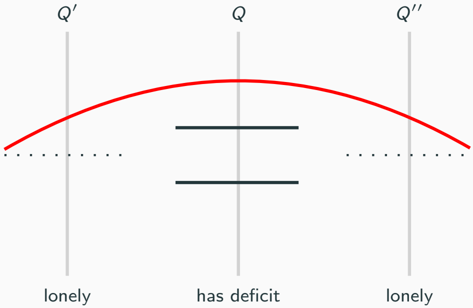


A few details on the proof: need for Gottschalk-Vygen



red edge is bad
(leads to reconnection)

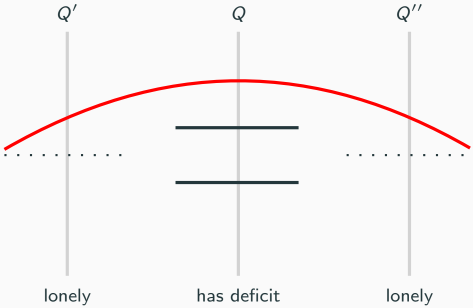
A few details on the proof: need for Gottschalk-Vygen



red edge is bad
(leads to reconnection)

GV guarantees that Q's *saved* edge-parts are not bad:

A few details on the proof: need for Gottschalk-Vygen

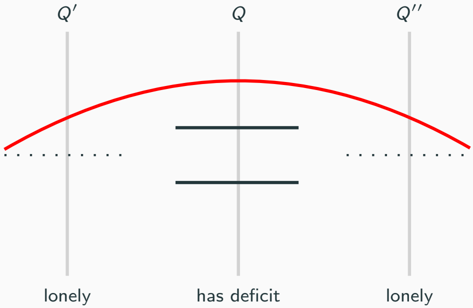


red edge is bad
(leads to reconnection)

GV guarantees that Q's *saved* edge-parts are not bad:

- Q's saved edge-parts come from trees were Q was lonely.

A few details on the proof: need for Gottschalk-Vygen

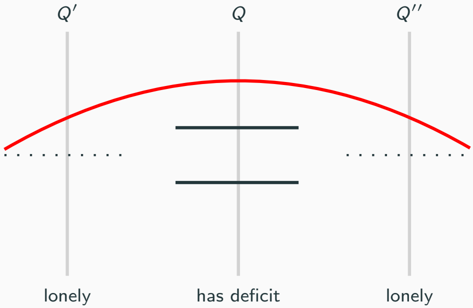


red edge is bad
(leads to reconnection)

GV guarantees that Q's *saved* edge-parts are not bad:

- Q's saved edge-parts come from trees were Q was lonely.
- In those trees Q', Q'' where also lonely.

A few details on the proof: need for Gottschalk-Vygen

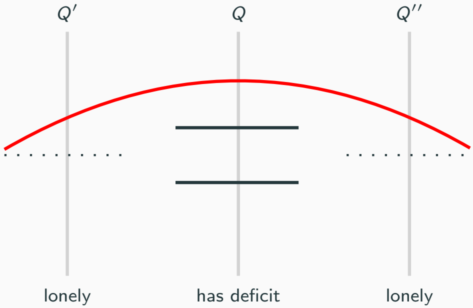


red edge is bad
(leads to reconnection)

GV guarantees that Q's *saved* edge-parts are not bad:

- Q's saved edge-parts come from trees were Q was lonely.
- In those trees Q', Q'' where also lonely.
- But then Q's saved edge-parts cannot cross Q' and Q''.

A few details on the proof: need for Gottschalk-Vygen



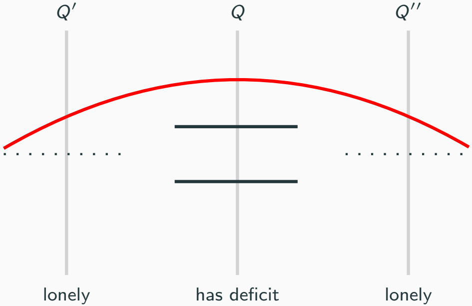
red edge is bad
(leads to reconnection)

GV guarantees that Q's *saved* edge-parts are not bad:

- Q's saved edge-parts come from trees were Q was lonely.
- In those trees Q', Q'' where also lonely.
- But then Q's saved edge-parts cannot cross Q' and Q''.

$\Rightarrow$ Q's saved edge-parts are not bad.

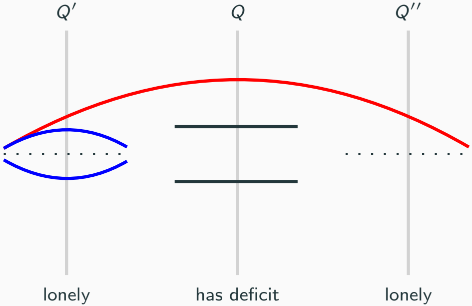
A few details on the proof: LP for reconnection



red edge is bad;
may still appear
because of $x^*/2$ part
of parity correction

For each bad edge (that crosses more than one cut with a lonely edge), we double all but one of the lonely edges to reconnect.

A few details on the proof: LP for reconnection

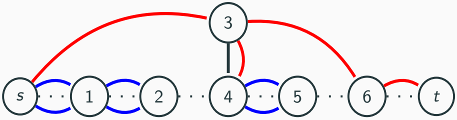


red edge is bad;
may still appear
because of $x^*/2$ part
of parity correction

For each bad edge (that crosses more than one cut with a lonely edge), we double all but one of the lonely edges to reconnect.

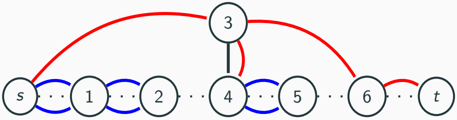
The solution to a linear program is used to decide *which* lonely edges are used for reconnection (only needed to "spread things out" for the analysis: the algorithm may use the cheapest lonely edges).

Conclusion



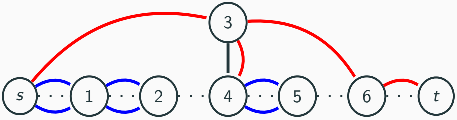
- Approximation guarantee and integrality gap bound of $\frac{3}{2} + \frac{1}{34}$

Conclusion



- Approximation guarantee and integrality gap bound of $\frac{3}{2} + \frac{1}{34}$
- Mixing Parity and Connectivity stages in new ways leads to benefits!

Conclusion



- Approximation guarantee and integrality gap bound of $\frac{3}{2} + \frac{1}{34}$
- Mixing Parity and Connectivity stages in new ways leads to benefits!
- Open question: Shave off the remaining $+\frac{1}{34}$ from the ratio.

Questions??