A $\left(\frac{3}{2}+\frac{1}{e}\right)$-Approximation Algorithm for Ordered TSP

Susanne Armbruster[1] Matthias Mnich[2] Martin Nägele[1]

[1]University of Bonn [2]TU Hamburg



Swiss National Science Foundation

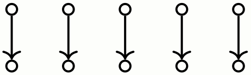
DFG Deutsche Forschungsgemeinschaft
German Research Foundation

Motivation: TSP with Precedence Constraints

Precedence Constraints:

Partial order $\prec$ on ground set
that needs to be respected in tour construction

- Pickup-delivery constraints:
  $p_i \prec d_i$



- Total order on subset:
  $d_1 \prec d_2 \prec \ldots \prec d_k$

Ordered TSP

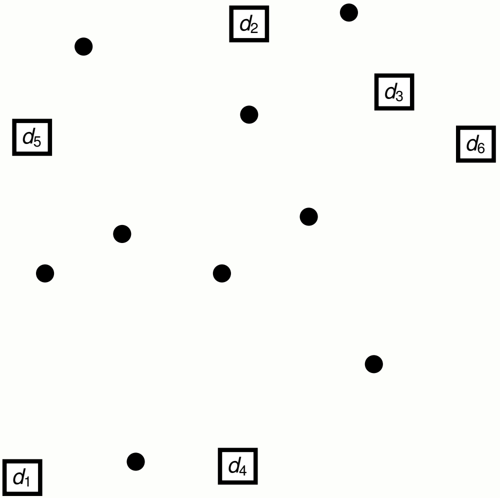
Input:

Complete $G = (V, E)$, metric $c \colon E \to \mathbb{R}_{\geq 0}$.

Distinct vertices $d_1, \ldots, d_k$.

Task:

Find a cheapest Hamiltonian cycle C in G that visits $d_1, \ldots, d_k$ in this order.



Ordered TSP

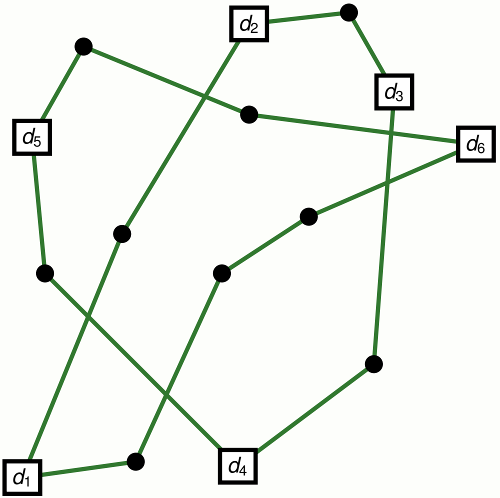
Input:

Complete $G = (V, E)$, metric $c \colon E \to \mathbb{R}_{\geq 0}$.

Distinct vertices $d_1, \ldots, d_k$.

Task:

Find a cheapest Hamiltonian cycle C in G that visits $d_1, \ldots, d_k$ in this order.

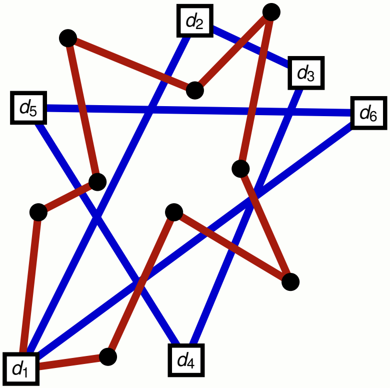


Known and new results

- Immediate $\frac{5}{2}$-approximation:
  Direct $d_1, \ldots, d_k$ tour $+$ Christofides TSP tour
  [Böckenhauer, Hromkovič, Kneis, Kupke 2006]

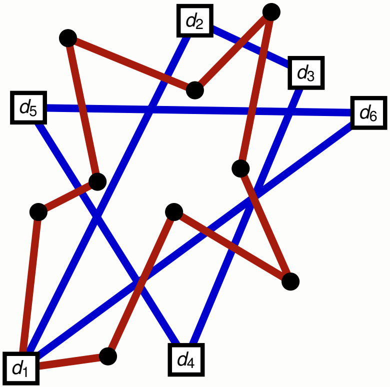
- Improved to $\frac{5}{2} - \frac{2}{k}$ for $k \geq 2$
  [Böckenhauer, Mömke, Steinova 2013]

- Exact DP in $O(2^r r^2 n)$ time and $O(2^r rn)$ space for $r = n - k$
  [Deineko, Hoffmann, Okamoto, Woeginger 2006]



Known and new results

- Immediate $\frac{5}{2}$-approximation:
  Direct $d_1, \ldots, d_k$ tour $+$ Christofides TSP tour
  [Böckenhauer, Hromkovič, Kneis, Kupke 2006]
- Improved to $\frac{5}{2} - \frac{2}{k}$ for $k \geq 2$
  [Böckenhauer, Mömke, Steinova 2013]
- Exact DP in $O(2^r r^2 n)$ time and $O(2^r rn)$ space for $r = n - k$
  [Deineko, Hoffmann, Okamoto, Woeginger 2006]

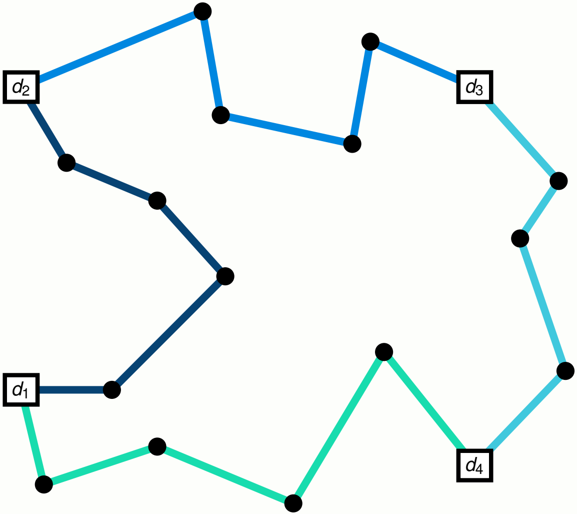


Our main result

There is a polynomial-time α-approximation algorithm for Ordered TSP, where $\alpha = \frac{3}{2} + \frac{1}{e} < 1.868$.

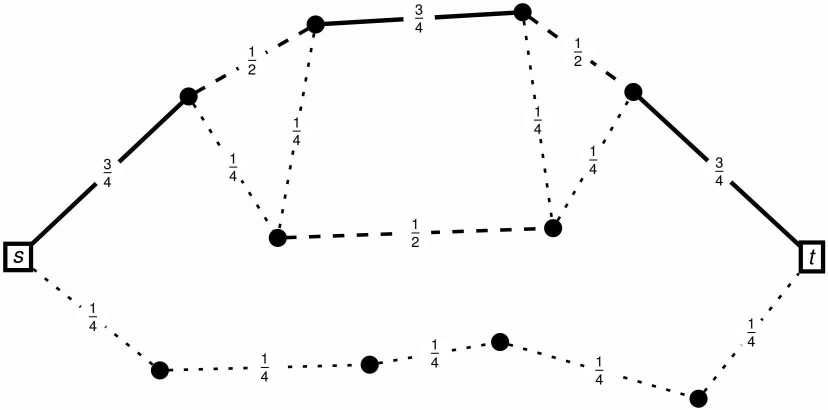
[Armbruster, Mnich, Nägele 2024]

Our high-level approach

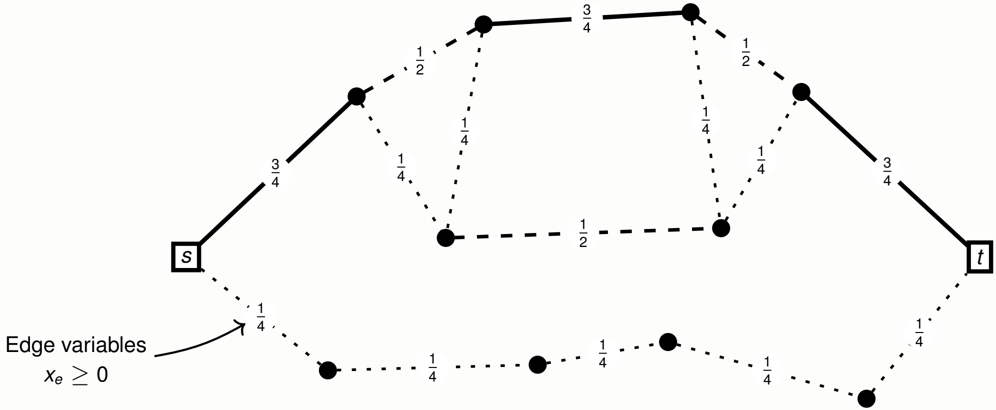


- Split a solution into d_i-d_{i+1} *strolls*
- Obtain an LP relaxation through polyhedral description of s-t strolls
- Round an optimal LP solution

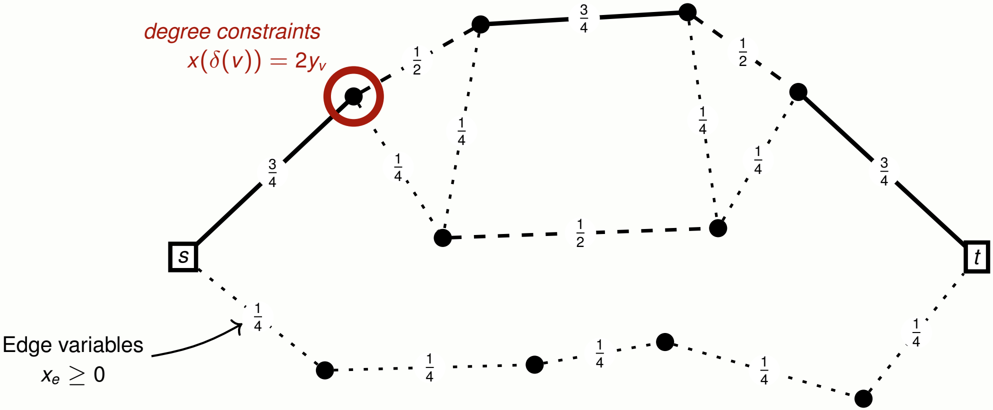
A Held-Karp type relaxation of s-t strolls: The polytope $P_{s\text{-}t}$



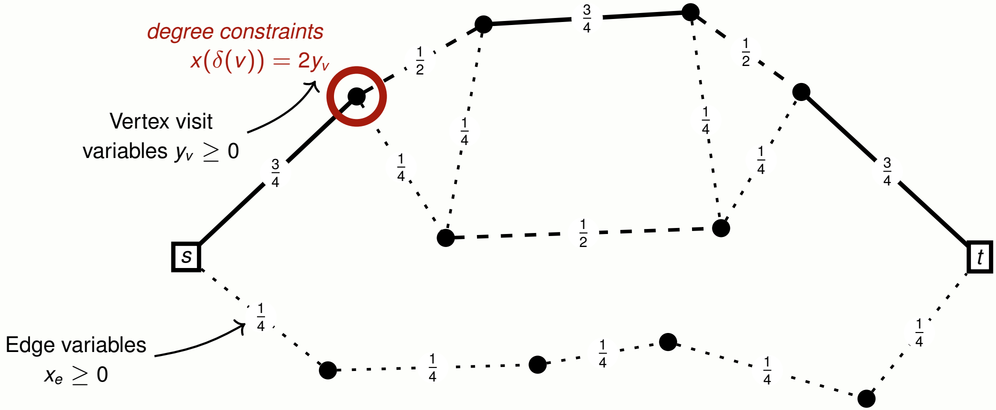
A Held-Karp type relaxation of *s*-*t* strolls: The polytope $P_{s\text{-}t}$



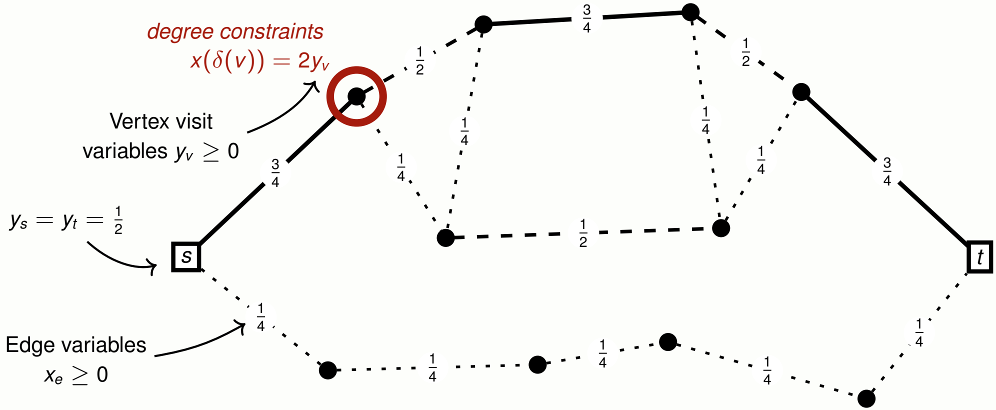
A Held-Karp type relaxation of s-t strolls: The polytope $P_{s\text{-}t}$



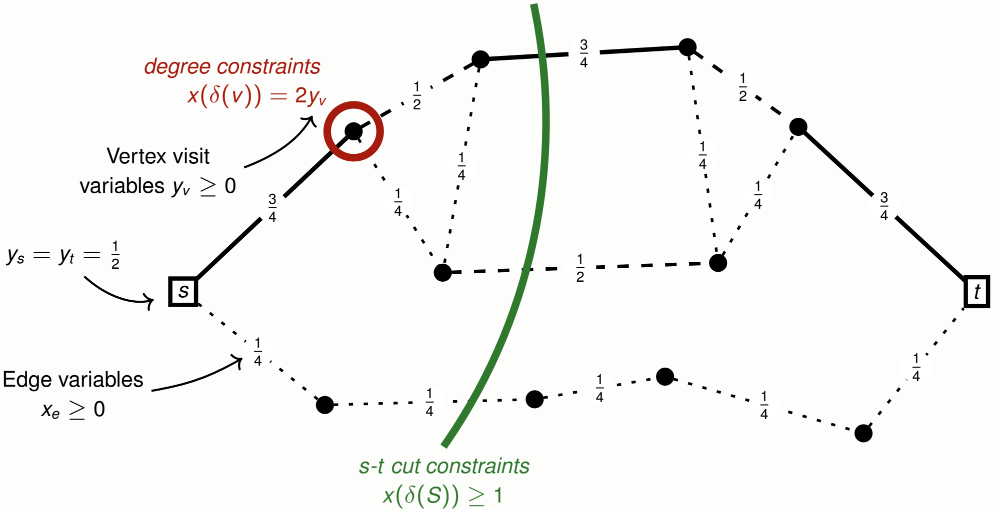
A Held-Karp type relaxation of s-t strolls: The polytope $P_{s\text{-}t}$



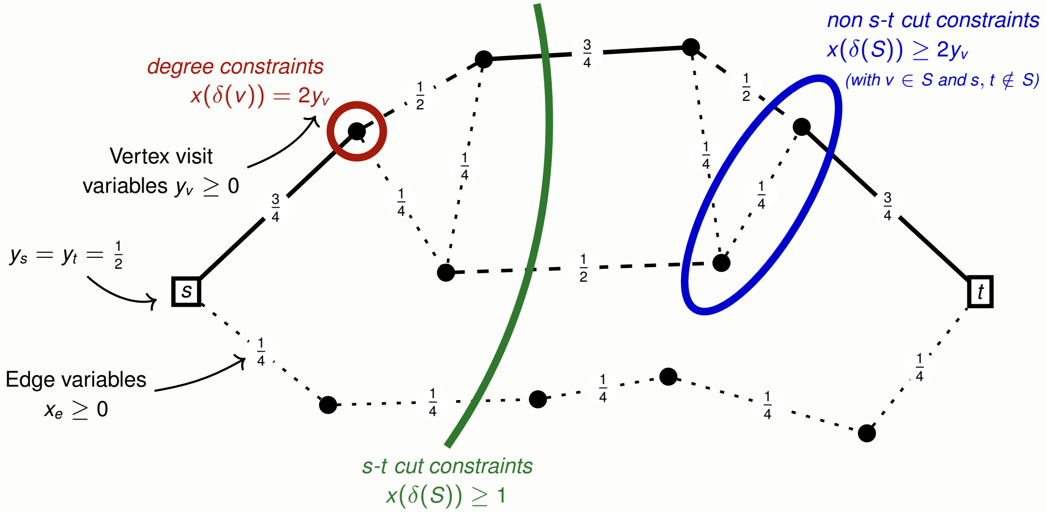
A Held-Karp type relaxation of s-t strolls: The polytope $P_{s\text{-}t}$



A Held-Karp type relaxation of s-t strolls: The polytope $P_{s\text{-}t}$



A Held-Karp type relaxation of s-t strolls: The polytope $P_{s\text{-}t}$



Our LP-based algorithm

1 Solve the LP relaxation

$$\min \quad \sum_{e \in E} c_e \sum_{i=1}^{k} x_e^i$$

$$\sum_{i=1}^{k} y_v^i = 1 \qquad \forall v \in V$$

$$(x^i, y^i) \in P_{d_i\text{-}d_{i+1}} \qquad \forall i \in \{1, \ldots k\}$$

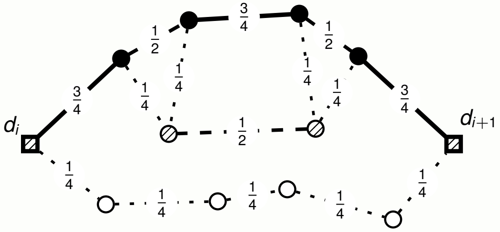
joint coverage of all vertices

fractional d_i-d_{i+1} strolls

Our LP-based algorithm

1 Solve the LP relaxation

Obtain covering fractional d_i-d_{i+1} strolls (x^i, y^i).



Our LP-based algorithm

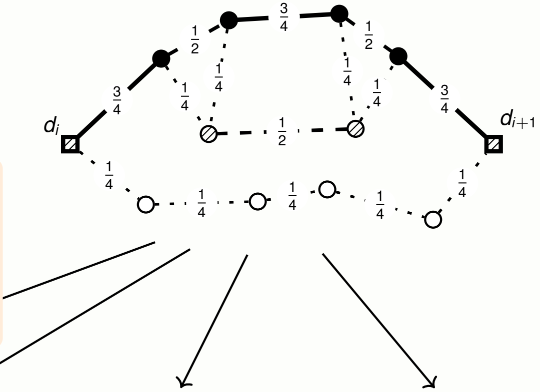
1 Solve the LP relaxation

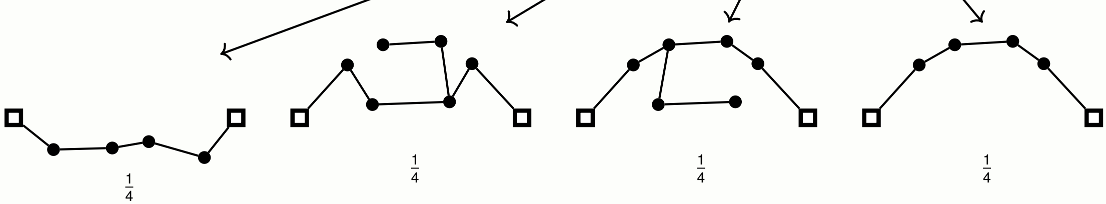
Obtain covering fractional d_i-d_{i+1} strolls (x^i, y^i).

2 Sample trees from LP solution

Decompose (x^i, y^i) and sample a tree T_i connecting d_i and d_{i+1} with

$$\mathbb{E}[c(E[T_i])] = c^\top x^i \quad \text{and}$$
$$\mathbb{P}[v \in V[T_i]] = y^i_v \quad \text{for } v \in V \setminus \{d_i, d_{i+1}\} \ .$$





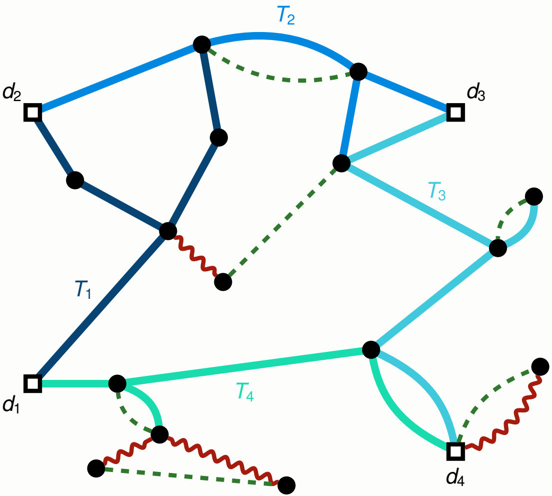
Our LP-based algorithm

1 Solve the LP relaxation

Obtain covering fractional d_i-d_{i+1} strolls (x^i, y^i).

2 Sample trees from LP solution

Expected total cost c_{LP}, preserve marginal coverage.



Our LP-based algorithm

1 Solve the LP relaxation

Obtain covering fractional d_i-d_{i+1} strolls (x^i, y^i).

2 Sample trees from LP solution

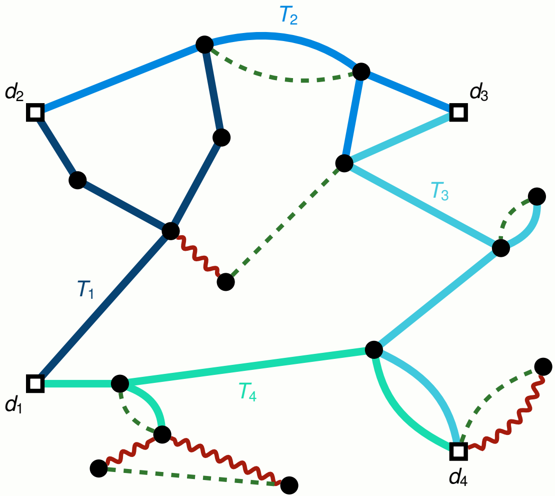
Expected total cost c_{LP}, preserve marginal coverage.

3 Ensure connectivity

Vertex $v \neq d_j$ is not covered by *any* T_i with probability

$$\prod_{i=1}^{k}(1 - y_v^i) \leq \exp\left(-\sum_{i=1}^{k} y_v^i\right) = \frac{1}{\mathrm{e}} \; .$$

$\implies$ can find a connector at expected cost $\leq \frac{1}{\mathrm{e}} \cdot c_{\mathrm{LP}}$.



Our LP-based algorithm

1 Solve the LP relaxation

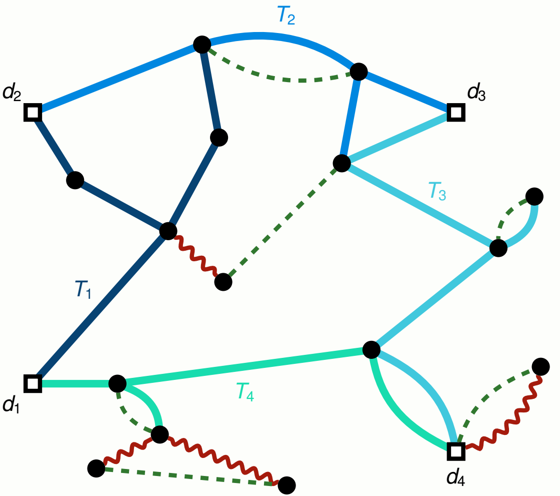
Obtain covering fractional d_i-d_{i+1} strolls (x^i, y^i).

2 Sample trees from LP solution

Expected total cost c_{LP}, preserve marginal coverage.

3 Ensure connectivity

Possible at expected extra cost $\leq \frac{1}{e} \cdot c_{LP}$.



Our LP-based algorithm

1 Solve the LP relaxation

Obtain covering fractional d_i-d_{i+1} strolls (x^i, y^i).

2 Sample trees from LP solution

Expected total cost c_{LP}, preserve marginal coverage.

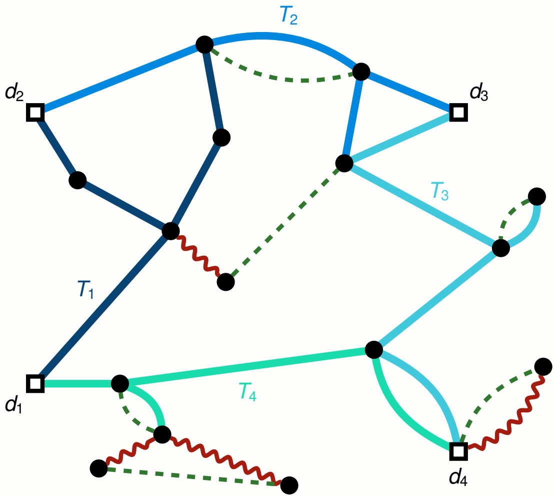
3 Ensure connectivity

Possible at expected extra cost $\leq \frac{1}{e} \cdot c_{LP}$.

4 Parity correction

We have $x := \sum_{i=1}^{k} x^i \in P_{HK}$.

$\implies$ any T-join costs $\leq \frac{1}{2} \cdot c_{LP}$.



Our LP-based algorithm

1 Solve the LP relaxation

Obtain covering fractional d_i-d_{i+1} strolls (x^i, y^i).

2 Sample trees from LP solution

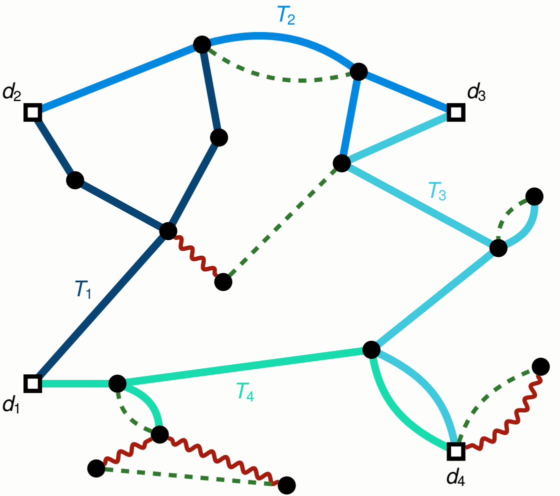
Expected total cost c_{LP}, preserve marginal coverage.

3 Ensure connectivity

Possible at expected extra cost $\leq \frac{1}{e} \cdot c_{LP}$.

4 Parity correction

Obtain even degrees at extra cost $\leq \frac{1}{2} \cdot c_{LP}$.



Our LP-based algorithm

1 Solve the LP relaxation

Obtain covering fractional d_i-d_{i+1} strolls (x^i, y^i).

2 Sample trees from LP solution

Expected total cost c_{LP}, preserve marginal coverage.

3 Ensure connectivity

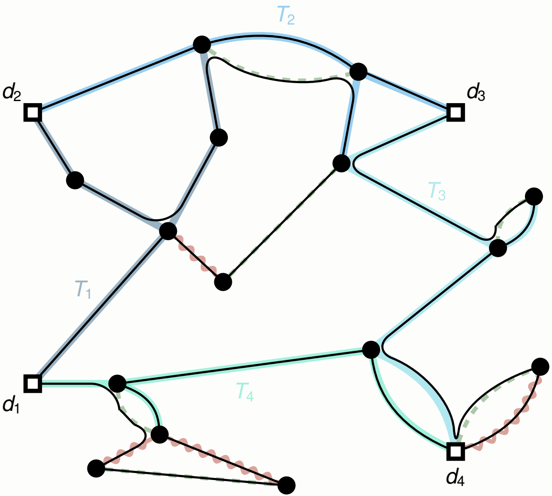
Possible at expected extra cost $\leq \frac{1}{e} \cdot c_{LP}$.

4 Parity correction

Obtain even degrees at extra cost $\leq \frac{1}{2} \cdot c_{LP}$.

5 Tour construction

Shortcut an Euler tour respecting the order $d_1, \ldots, d_k$.



Conclusion

Theorem

Randomized $\left(\frac{3}{2} + \frac{1}{e}\right)$-approximation algorithm for Ordered TSP.

- Efficient implementation: LP can be solved in polynomial time via separation
- Derandomization: Following the method of conditional expectations

Related open problems

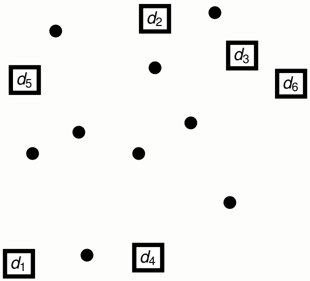
"Ordered Tree" problem

Input:

OTSP instance

Task:

Find a cheapest spanning tree T in G that contains a path visiting $d_1, \ldots, d_k$ in this order.



Related open problems

"Ordered Tree" problem

Input:

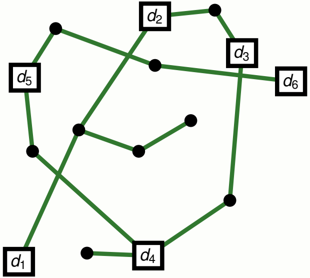
OTSP instance

Task:

Find a cheapest spanning tree T in G that contains a path visiting $d_1, \ldots, d_k$ in this order.

Our algorithm:

Ordered tree solution of cost $(1 + \frac{1}{e}) \cdot c_{LP}$.



Related open problems

"Ordered Tree" problem

Input:

OTSP instance

Task:

Find a cheapest spanning tree T in G that contains a path visiting $d_1, \ldots, d_k$ in this order.

Deadline TSP problem

Input:

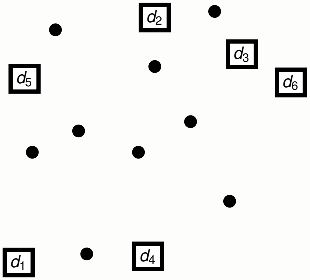
OTSP instance with deadlines $\delta_1, \ldots, \delta_k$.

Task:

Find a cheapest Hamiltonian cycle C in G such that the tour length before d_i is at most δ_i.

Our algorithm:

Ordered tree solution of cost $(1 + \frac{1}{e}) \cdot c_{LP}$.



Related open problems

"Ordered Tree" problem

Input:

OTSP instance

Task:

Find a cheapest spanning tree T in G that contains a path visiting $d_1, \ldots, d_k$ in this order.

Deadline TSP problem

Input:

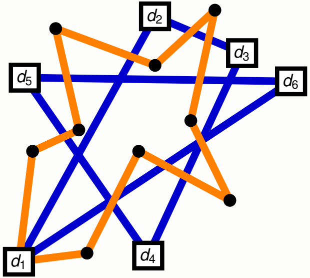
OTSP instance with deadlines $\delta_1, \ldots, \delta_k$.

Task:

Find a cheapest Hamiltonian cycle C in G such that the tour length before d_i is at most δ_i.

Our algorithm:

Ordered tree solution of cost $(1 + \frac{1}{e}) \cdot c_{LP}$.



Immediate 2.5-approximation:
direct $d_1, \ldots, d_k$ tour + Christofides' tour