A $(3/2 + 1/e)$-Approximation Algorithm for Ordered TSP

Susanne Armbruster
Research Institute for Discrete Mathematics and Hausdorff Center for Mathematics,
University of Bonn, Germany

Matthias Mnich
Hamburg University of Technology, Institute for Algorithms and Complexity, Hamburg, Germany

Martin Nägele
Department of Mathematics, ETH Zurich, Zurich, Switzerland[1]

Abstract

We present a new $(3/2 + 1/e)$-approximation algorithm for the Ordered Traveling Salesperson Problem (Ordered TSP). Ordered TSP is a variant of the classic metric Traveling Salesperson Problem (TSP) where a specified subset of vertices needs to appear on the output Hamiltonian cycle in a given order, and the task is to compute a cheapest such cycle. Our approximation guarantee of approximately 1.868 holds with respect to the value of a natural new linear programming (LP) relaxation for Ordered TSP. Our result significantly improves upon the previously best known guarantee of $5/2$ for this problem and thereby considerably reduces the gap between approximability of Ordered TSP and metric TSP. Our algorithm is based on a decomposition of the LP solution into weighted trees that serve as building blocks in our tour construction.

2012 ACM Subject Classification Theory of computation → Discrete optimization; Theory of computation → Routing and network design problems; Theory of computation → Rounding techniques

Keywords and phrases Travelling Salesperson Problem, precedence constraints, linear programming, approximation algorithms

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2024.1

Category APPROX

Related Version *Full Version*: https://arxiv.org/abs/2405.06244 [1]

Funding *Matthias Mnich*: Partially supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), project MN 59/4-1.
Martin Nägele: Partially supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXZ-2047/1 – 390685813, and the Swiss National Science Foundation (grant no. P500PT_206742).

1 Introduction

The classic metric Traveling Salesperson Problem (TSP) is one of the most fundamental and well-studied problems in Combinatorial Optimization and has a large number of applications. A metric TSP instance is given by a complete undirected graph $G = (V, E)$ with metric edge cost $c: E \to \mathbb{R}_{\geq 0}$. The task is to find a cycle of minimum cost that visits each vertex exactly once, where the cost of a cycle equals the sum of the edge costs over all edges it contains. Metric TSP is highly relevant in many practical applications and thus, a lot of different

[1] Most of this work was done while M. Nägele was employed at University of Bonn.

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2024).
Editors: Amit Kumar and Noga Ron-Zewi; Article No. 1; pp. 1:1–1:18
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

variants are studied (see, e.g., [34]). The problem is NP-hard and APX-hard [32]; concretely, assuming $\mathsf{P} \neq \mathsf{NP}$, it is known that no polynomial-time algorithm can guarantee to find a cycle of cost at most $^{123}/_{122}$ times the cost of a cheapest cycle [24]. For a long time, the best-known approximation algorithm for metric TSP was the Christofides-Serdyukov $^3/_2$-approximation algorithm [10,11,35]. This was recently improved to a breakthrough ($^3/_2 - \varepsilon$)-approximation algorithm, for some $\varepsilon > 10^{-36}$, by Karlin, Klein, and Oveis Gharan [22,23].

In this work, we focus on a generalization of metric TSP known as *Ordered TSP*, in which some of the vertices must be visited in a given order.

Ordered TSP (OTSP): Given a complete undirected graph $G = (V, E)$ with metric edge cost $c\colon E \to \mathbb{R}_{\geq 0}$ and pairwise distinct vertices $d_1, \dots, d_k \in V$, the task is to find a cheapest spanning cycle C in G that contains the vertices d_1, $\dots$, d_k in this order.

We typically refer to an input of OTSP as an *OTSP instance* $(G, c, (d_1, \dots, d_k))$; solutions are often called *tours*. Our goal in this paper is to further the understanding of the approximability of OTSP, i.e., we aim to design α-approximation algorithms for OTSP with α as small as possible.

Clearly, OTSP is at least as hard as metric TSP, and therefore APX-hard. Surprisingly, not much more is known on the approximability of OTSP. Böckenhauer, Hromkovič, Kneis, and Kupke [6] observed that a $^5/_2$-approximate solution can be readily obtained by first traversing $d_1, \dots, d_k$ in this order and subsequently appending a tour on $V \setminus \{d_1, \dots, d_k\}$ constructed through the Christofides-Serdyukov algorithm. The black-box use of a metric TSP approximation algorithm allows to reduce this guarantee by the same additive improvement of $\varepsilon > 10^{-36}$ as in the ($^3/_2 - \varepsilon$)-approximation by Karlin, Klein, and Oveis Gharan. Besides that, Böckenhauer, Mömke, and Steinová [7] gave a ($^5/_2 - ^2/_k$)-approximation algorithm, where $k \geq 2$ is the number of ordered vertices in the OTSP input. Note that their result does not directly inherit the improvement achieved for metric TSP, making its approximation ratio asymptotically inferior to the earlier approach of Böckenhauer, Hromkovič, Kneis, and Kupke. Finally, the intuition that OTSP should become easier once k approaches n is confirmed by a dynamic programming approach of Deĭneko, Hoffmann, Okamoto, and Woeginger [14] that runs in $O(2^r r^2 n)$ time and $O(2^r rn)$ space, i.e., in polynomial time and space if $r := n - k$, the number of vertices that are not in the input order, is of magnitude $O(\log n)$.

OTSP is in fact a special case of a the following significantly more general TSP variation termed *TSP with Precedence Constraints.*

TSP with Precedence Constraints (TSP-PC): Given a complete undirected graph $G = (V, E)$ with metric edge cost $c\colon E \to \mathbb{R}_{\geq 0}$ and a partial order $\prec$ on V, the task is to find a cheapest spanning cycle C in G that respects $\prec$, i.e., C can be traversed such that whenever $u \prec v$ for two vertices $u, v \in V$, then u appears earlier on C than v.

Compared to the total order constraints in OTSP, general partial orders allow for modeling a much wider range of problems. One among many applications of TSP-PC is, e.g., tour planning for mixed pickup and delivery services, where one needs to make sure that a pickup happens before a delivery (but apart from that, pickups and deliveries can be intertwined arbitrarily). There is a considerable body of research on the structure of the

TSP-PC polyhedron, different dynamic programming algorithms, enhanced branch-and-bound methods, and various other exact and heuristic approaches, typically even for the more general version of TSP-PC with asymmetric edge cost (see, e.g., [2, 18, 25, 33] and references therein). Despite that, essentially no positive results on the approximability of TSP-PC are known, which is possibly explained by an influential hardness result of Charikar, Motwani, Raghavan, and Silverstein [9]: By relating the problem to the *Shortest Common Supersequence Problem*, they are able to show that there is no $(\log n)^{\delta}$-approximation for the path version of TSP-PC for any constant δ, unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{O(\log\log n)})$, even if the underlying metric space is a line. This motivates our study of the approximability of TSP-PC on general metric spaces with special partial orders, i.e., OTSP.

1.1 Our results and techniques

Our main contribution is to significantly improve the state of the art for OTSP by giving an LP-relative approximation guarantee of $3/2 + 1/e \approx 1.868$, as stated in the following theorem.

▶ **Theorem 1.** *There is a polynomial-time* $(3/2 + 1/e)$*-approximation algorithm for OTSP.*

This constitutes a significant improvement over the previous $(5/2 - \varepsilon)$-approximation algorithm. We achieve this improvement by introducing a new linear programming (LP) relaxation for OTSP and devising a suitable rounding procedure. The LP relaxation is based on the Held-Karp relaxation that is typically leveraged in the context of TSP, but allows for taking the prescribed order of the vertices $d_1, \dots, d_k$ into account by using disjoint sets of variables to represent the d_i-d_{i+1} strolls[2] that a solution is composed of. Our rounding procedure crucially relies on a result on decomposing (fractional) s-t strolls into a convex combination of trees. This decomposition resembles an existential result by Bang-Jensen, Frank, and Jackson [3, Theorem 2.6] on packing branchings in a directed multigraph. Variations thereof have recently been used for advances on another variant of TSP, namely *Prize-Collecting TSP* [4, 5], and motivate the application here. (See Lemma 5 for the precise statement of the decomposition result.) The trees obtained from stroll decompositions enable the construction of a subgraph that spans a reasonably large part of V at cost no more than the LP solution cost, and contains a walk with visits at $d_1, \dots, d_k$ in this order. Our tour construction is completed by connecting the remaining isolated vertices in a cheapest possible way, and applying a parity correction step as typical for TSP-like problems.

Our approach crucially relies on being able to split a solution into d_i-d_{i+1} strolls upfront, hence it is not directly suitable for handling arbitrary precedence constraints other than total orders. While one can always try to guess a suitable total order that is compatible with the given partial order, and then apply Theorem 1, this is generally not efficient. We can, though, obtain approximation algorithms for some special cases of precedence constraints, as for example in the following result that is a direct generalization of Theorem 1.

▶ **Theorem 2.** *Consider a TSP-PC instance* $(G, c, \prec)$ *on a complete graph* $G = (V, E)$ *with a partial order* $\prec$ *that can be equivalently given as independent total orders on disjoint subsets* $D_1, \dots, D_\ell \subseteq V$*. There is a polynomial-time* $(\ell + 1/2 + 1/e^{\ell})$*-approximation algorithm for this class of TSP-PC problems.*

[2] We use the term *s-t stroll* instead of *s-t path* for a path from s to t in the underlying graph to emphasize that we do not require all vertices to be covered. Also, for convenience of notation, we use $d_{k+1} := d_1$ throughout the paper.

The total orders on the sets D_i are also called *chains*. We remark that losing a factor of ℓ in Theorem 2 is intrinsic to our approach: We never merge the given chains, but traverse them one after another. Still, the result of Theorem 2 is superior to a black-box algorithm that independently applies the algorithm from Theorem 1 to the ℓ chains and concatenates the resulting tours (while shortcutting to avoid repeated visits).

1.2 Related Work

Variations of OTSP and TSP-PC are also studied in the context of scheduling with precedence constraints. In a classic setup, denoted by $Pm|\text{prec}|C_{\max}$ in the scheduling literature, one needs to find a schedule for a set $\mathcal{J}$ of n jobs on m identical machines subject to precedence constraints between the jobs. Formally, each job $j \in \mathcal{J}$ is characterized by a processing time $p_j \in \mathbb{Z}_{\geq 0}$, and a schedule $\sigma\colon \mathcal{J} \to \mathbb{Z}_{\geq 0} \times \{1, \ldots, m\}$ assigns each job $j \in \mathcal{J}$ to a pair $(\sigma_1(j), \sigma_2(j))$ consisting of an integer start time $\sigma_1(j)$ and a machine $\sigma_2(j)$ such that no other job scheduled on that machine has their start time in the time interval $[\sigma_1(j), \sigma_1(j) + p_j]$, and for any two jobs $j, j' \in \mathcal{J}$ related as $j \prec j'$ it holds that $\sigma_1(j) < \sigma_2(j')$. The makespan objective $C_{\max}$ of a schedule σ is the maximum completion time $C_j = \sigma_1(j) + p_j$ over all jobs $j \in \mathcal{J}$. Generally, precedence constraints of this type are studied extensively in a wide range of scheduling problems, including different settings and objectives (see, e.g., [13, 19, 28, 36]). The three-machine problem $P3|\text{prec}, p_j \equiv 1|C_{\max}$ is one of the few famous open problems by Garey and Johnson [17] whose computational complexity has not yet been resolved.

The complexity of many scheduling problems with precedence constraints that are chains has been well-investigated. An influential paper of Lenstra and Rinnooy Kan [27] shows strong NP-hardness for minimizing the number of chain-constrained unit-size jobs that miss their deadline on a single machine. Several other works with chain constraints have appeared [15, 21, 26, 37].

Towards analogues of TSP-PC, we may consider the problem $Pm|\text{prec}|C_{\max}$ on a single machine, but add sequence-dependent setup times $s_{ij} \in \mathbb{Z}_{\geq 0}$ between any two jobs i and j, which add to the makespan of the schedule. This problem, which is denoted as $1|\text{prec}, s_{ij}|C_{\max}$, was discussed by Liaee and Emmons [29, Section 3.1.2]. In case of TSP-PC, the setup times are metric (i.e., $s_{ij} \leq s_{ik} + s_{kj}$ for any triple (i, j, k) of distinct jobs), and all jobs have equal processing time $p_j \equiv 0$. To be precise, the objective function for TSP-PC takes into account the cost for returning to the origin city whereas no such cost occurs in the objective function for $1|\text{prec}, s_{ij}|C_{\max}$, hence the latter in fact models a path version of TSP-PC.

We remark that shortly after the first dissemination of this work [1], Böhm, Friggstad, Mömke, and Spoerhase [8] provided an independent paper on TSP variants, including a result matching the guarantee in Theorem 1. Albeit different at first sight, their approach is very similar at its core to the one presented here: They use a directed analogue of the LP relaxation that we also build our work on here, and thereby facilitate a direct application of the original result by Bang-Jensen, Frank, and Jackson on packing branchings in a directed multigraph.

1.3 Organization of the paper

In Section 2, we introduce our new linear programming formulation for OTSP (Section 2.1) and analyze a randomized algorithm giving the guarantee of Theorem 1 in expectation (Section 2.2). We show how this algorithm can be derandomized in Section 2.3. Finally, Section 3 extends our framework to yield Theorem 2, and Section 4 shows how our main technical lemma is implied by a closely related known result.

2 Our algorithm

2.1 The LP relaxation and polyhedral basics

The most commonly used LP relaxation in approximation algorithms for classic TSP is the so-called *Held-Karp relaxation*. It was first introduced by Dantzig, Fulkerson, and Johnson [12] and is given by

$$P_{\mathrm{HK}}(G) \coloneqq \left\{ x \in \mathbb{R}^E_{\geq 0} \colon \begin{array}{rcll} x(\delta(v)) & = & 2 & \forall v \in V \\ x(\delta(S)) & \geq & 2 & \forall S \subsetneq V, S \neq \emptyset \end{array} \right\} ,$$

where $G = (V, E)$ is the underlying complete graph.[3] While TSP simply asks for a spanning cycle, OTSP requires that the vertices $d_1, \ldots, d_k$ appear on the cycle in this order. Thus, a solution is naturally composed of k strolls, namely a d_i-d_{i+1} stroll for every $i \in \{1, \ldots, k\}$. For a polyhedral description of s-t strolls in a complete graph $G = (V, E)$, we modify the Held-Karp relaxation for *s-t path TSP*[4] to allow partial coverage of vertices. Concretely, the variables $y \in \mathbb{R}^V_{\geq 0}$ in the following formulation indicate the extent at which vertices are covered:[5]

$$P_{s\text{-}t \text{ stroll}}(G) \coloneqq \left\{ (x,y) \in \mathbb{R}^E_{\geq 0} \times \mathbb{R}^V_{\geq 0} \colon \begin{array}{rcll} x(\delta(v)) & = & 2y_v & \forall v \in V \\ x(\delta(S)) & \geq & 1 & \forall S \subseteq V \setminus \{t\}, s \in S \\ x(\delta(S)) & \geq & 2y_v & \forall S \subseteq V \setminus \{s,t\}, v \in S \\ y_s = y_t & = & 1/2 & \end{array} \right\} . \quad (1)$$

Note that setting $y_s = y_t = 1/2$ corresponds to s and t having degree 1 in an s-t stroll, while all interior vertices of an integral stroll have degree 2, which corresponds to a y-value of 1. Using the above polyhedral relaxation (1) for all d_i-d_{i+1} strolls, it remains to link the strolls by requiring full joint coverage of every $v \in V$. This results in the following LP relaxation for OTSP:

$$\begin{array}{rrcll} \min & \displaystyle\sum_{e \in E} c_e \sum_{i=1}^{k} x^i_e & & & \\ & \displaystyle\sum_{i=1}^{k} y^i_v & = & 1 & \forall v \in V \\ & (x^i, y^i) & \in & P_{d_i\text{-}d_{i+1} \text{ stroll}}(G) & \forall i \in \{1, \ldots, k\} \ . \end{array} \qquad \text{(OTSP LP relaxation)}$$

Here and throughout the paper, note that we use superscripts to distinguish different strolls. It is clear that any OTSP solution can be turned into a feasible solution to the above LP of the same objective value, hence the above LP is indeed a relaxation of OTSP. We first observe that this OTSP LP relaxation strengthens the Held-Karp relaxation in the following sense.

▶ **Observation 3.** *Let $(x^i, y^i)_{i \in \{1,\ldots,k\}}$ be feasible for the OTSP LP relaxation. Then $x \coloneqq \sum_{i=1}^{k} x^i \in P_{HK}(G)$.*

[3] For $S \subseteq V$ we denote by $\delta(S)$ the set of edges with exactly one endpoint in S. For $v \in V$, we abbreviate $\delta(v) \coloneqq \delta(\{v\})$.

[4] Given a complete graph $G = (V, E)$ with metric edge costs and vertices $s, t \in V$, *s-t path TSP* is the variant of TSP that seeks a path of smallest total cost from s to t while visiting every vertex exactly once.

[5] The constraints of $P_{s\text{-}t \text{ stroll}}$ imply that for $v \in V \setminus \{s,t\}$, we have $2y_v \leq x(\delta(V \setminus \{s,t\})) \leq x(\delta(s)) + x(\delta(t)) \leq 2$, and thus $y_v \leq 1$, legitimating the proposed interpretation.

Proof. To see that x satisfies the degree constraints in $P_{\mathrm{HK}}(G)$, note that for all $v \in V$, we have

$$x(\delta(v)) = \sum_{i=1}^{k} x^i(\delta(v)) = 2 \cdot \sum_{i=1}^{k} y_i = 2 \ .$$

To verify the cut constraints, let $S \subsetneq V$ be a non-empty set of vertices. If both $S \cap \{d_1, \dots, d_k\} \neq \emptyset$ and $(V \setminus S) \cap \{d_1, \dots, d_k\} \neq \emptyset$, then there exist two distinct indices $i_1, i_2 \in \{1, \dots, k\}$ such that $d_{i_1} \in S$ but $d_{i_1+1} \notin S$, and $d_{i_2} \notin S$ but $d_{i_2+1} \in S$. This implies that $x^{i_1}(\delta(S)) \geq 1$ and $x^{i_2}(\delta(V \setminus S)) \geq 1$, so we get

$$x(\delta(S)) = \sum_{i=1}^{k} x^i(\delta(S)) \geq x^{i_1}(\delta(S)) + x^{i_2}(\delta(S)) = x^{i_1}(\delta(S)) + x^{i_2}(\delta(V \setminus S)) \geq 2 \ .$$

Otherwise, assume without loss of generality that $S \cap \{d_1, \dots, d_k\}$ is empty (if not, $V \setminus S$ has this property) and fix a vertex $v \in S$. We then know that $x^i(\delta(S)) \geq 2y_v$ for all $i \in \{1, \dots, k\}$, hence

$$x(\delta(S)) = \sum_{i=1}^{k} x^i(\delta(S)) \geq 2 \cdot \sum_{i=1}^{k} y_v = 2 \ . \qquad ◀$$

The point $x \in P_{\mathrm{HK}}(G)$ constructed in Observation 3 has the property that its cost $c^\top x$ equals the objective value c_{LP} of the feasible point of the OTSP LP relaxation that we started with. Thus, following the arguments of Wolsey's polyhedral analysis [38] of the Christofides-Serdyukov algorithm, we immediately obtain the following.

▶ **Corollary 4.** *Let c_{LP} denote the optimal objective value of the OTSP LP relaxation. Then in the underlying graph G with edge costs c, the following holds true.*

(i) *A shortest spanning tree T satisfies $c(T) \leq c_{LP}$.*

(ii) *For any even cardinality set $Q \subseteq V$, a shortest Q-join J satisfies $c(J) \leq \frac{1}{2} \cdot c_{LP}$.*

Proof. Let $(x^i, y^i)_{i \in \{1,\dots,k\}}$ be an optimal solution of the OTSP LP relaxation. By Observation 3, $x := \sum_{i=1}^{k} x^i \in P_{\mathrm{HK}}(G)$. It is well-known due to Held and Karp [20] that then, $\frac{|V|-1}{|V|} \cdot x$ is feasible for the spanning tree polytope, and due to Wolsey [38] that $\frac{1}{2}x$ is feasible for the dominant of the Q-join polytope, hence $c(T) \leq \frac{|V|-1}{|V|} \cdot c^\top x < c^\top x$ and $c(J) \leq \frac{1}{2} c^\top x$. Using that $c^\top x = c_{\mathrm{LP}}$, the result follows. ◀

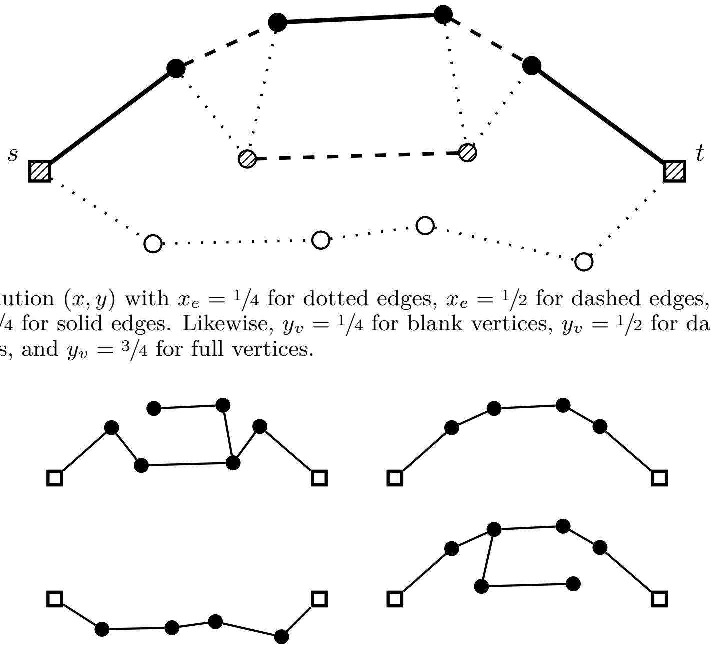
2.2 Rounding an LP solution

At its core, our algorithm for rounding a typically fractional solution $(x^i, y^i)_{i \in \{1,\dots,k\}}$ of the OTSP LP relaxation is based on leveraging a decomposition result for each of the points $(x^i, y^i) \in P_{d_i\text{-}d_{i+1}\text{ stroll}}$. By scaling up (x^i, y^i) by a large enough factor M such that Mx^i is integral, this decomposition can be viewed as a result on packing trees into the multigraph that has Mx^i_e copies of every edge $e \in E$ and such that every vertex v appears in My^i many of the trees. While most results of this type deal with packing spanning trees (or, in the directed case, arborescences), i.e., consider uniform packings, Bang-Jensen, Frank, and Jackson [3] gave one of few results in a non-uniform setting as we are facing here. Their splitting-off based construction was revised by Blauth and Nägele [5] to obtain more fine-grained control over the output components of the decomposition when starting from a solution of a Held-Karp-type relaxation that allows partial coverage of vertices (similar to what we allow in $P_{s\text{-}t\text{ stroll}}$). We observe that these findings can be immediately carried over to solutions of $P_{s\text{-}t\text{ stroll}}$, giving Lemma 5 below. We defer a formal proof to Section 4.

▶ **Lemma 5.** *Let $G = (V, E)$ be an undirected graph, $s, t \in V$, and let $(x, y) \in P_{s\text{-}t\ stroll}(G)$. We can in polynomial time compute a family $\mathcal{T}$ of subtrees of G that each contain the vertices s and t, and weights $\mu \in [0,1]^{\mathcal{T}}$ with $\sum_{T\in\mathcal{T}} \mu_T = 1$ such that*[6]

$$\sum_{T\in\mathcal{T}} \mu_T \chi^{E[T]} = x \qquad \textit{and} \qquad \sum_{T\in\mathcal{T}\colon v\in V[T]} \mu_T = y_v \quad \forall v \in V \setminus \{s,t\} \ .$$

In other words, Lemma 5 allows to decompose a fractional s-t stroll into a convex combination of trees in a family $\mathcal{T}$ that all connect s and t, and such that for every other vertex $v \in V \setminus \{s,t\}$, the weighted number of trees that contain v equals the coverage y_v of v in the stroll. An example of a feasible solution (x, y) and a decomposition satisfying the properties of Lemma 5 is given in Figure 1.



(a) Solution (x, y) with $x_e = 1/4$ for dotted edges, $x_e = 1/2$ for dashed edges, and $x_e = 3/4$ for solid edges. Likewise, $y_v = 1/4$ for blank vertices, $y_v = 1/2$ for dashed vertices, and $y_v = 3/4$ for full vertices.

(b) A decomposition of the solution (x, y) given in (a) into four trees with uniform weight $\mu \equiv 1/4$ satisfying the properties of Lemma 5.

Figure 1 A solution $(x, y) \in P_{s\text{-}t\ \text{stroll}}$ along with a decomposition into trees, exemplifying Lemma 5.

After applying Lemma 5 to all strolls $(x^i, y^i) \in P_{d_i\text{-}d_{i+1}\ \text{stroll}}$ obtained from an optimal solution of the OTSP LP relaxation, we choose one tree from each of the decompositions and consider the (multi-)union of all edges obtained this way. This results in a graph that already contains a closed walk with visits at $d_1, \dots, d_k$ in this order, giving the basis for our construction of an OTSP solution. Also, we can easily bound the expected cost of the edge set obtained in this way by randomly choosing the trees with marginals given by the weights μ from Lemma 5.

[6] For a graph H we denote by $V[H]$ the set of vertices and by $E[H]$ the set of edges of H. Furthermore, for an edge set $F \subseteq E$, we denote by $\chi^F \in \{0,1\}^E$ the characteristic vector of F, i.e., the $\{0,1\}$-vector with, for all $e \in E$, $\chi^F_e = 1$ if and only if $e \in F$.

To obtain an actual OTSP solution, the missing steps are to

(i) connect vertices that are not covered by any of the trees T_i,
(ii) perform parity correction to guarantee that there exists an Eulerian tour, and
(iii) shortcut appropriately to obtain an actual OTSP solution.

Altogether, this leads to the randomized Algorithm 1 as laid out below; also see Figure 2 for an example illustration of the different edge sets that are constructed in Algorithm 1.

Algorithm 1 A randomized approximation algorithm for OTSP.

```
Input: OTSP instance $(G, c, \{d_1, \dots, d_k\})$ on graph $G = (V, E)$.
Compute an optimal solution $(x^i, y^i)_{i \in \{1,\dots,k\}}$ to the OTSP LP relaxation.
foreach $i \in \{1, \dots, k\}$ do
    Apply Lemma 5 to decompose $(x^i, y^i)$ into trees $\mathcal{T}_i$ with weights $\mu^i$.
    Sample one tree $T_i$ from $\mathcal{T}_i$ with marginals given by $\mu^i$.
Compute a minimum-cost edge set $F \subseteq E$ such that the multigraph
    $$H := \left(V, F \cup \dot{\bigcup}_{i \in \{1,\dots,k\}} E[T_i]\right)$$
  is connected.
Let $Q = \mathrm{odd}(H)$ and compute a minimum cost $Q$-join $J$ in $G$.
return Spanning cycle $C$ in $G$ obtained from $H \dot\cup J$ through Lemma 10.
```

We first show that Algorithm 1 gives the guarantees claimed by Theorem 1 in expectation and – even stronger – with respect to the value c_{LP} of the OTSP LP relaxation, as stated in the subsequent theorem. In Section 2.3, we show that Algorithm 1 admits an immediate derandomization using the method of conditional expectation, thereby completing the proof of Theorem 1.

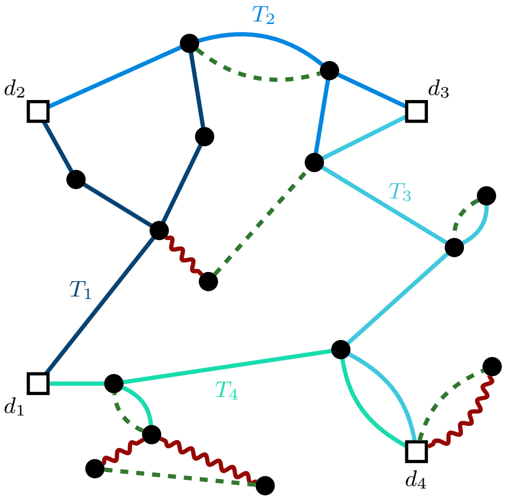


Figure 2 Exemplifying the construction of Eulerian graph $H \dot\cup J$ from Algorithm 1: Trees T_1, T_2, T_3, T_4 drawn as solid blueish edges, the edge set F connecting all vertices to the trees drawn as curly red edges, and the odd(H)-join J drawn as dashed green edges.

▶ **Theorem 6.** *Let c_{LP} be the cost of an optimal solution of the OTSP LP relaxation. Algorithm 1 returns in polynomial time an OTSP solution C satisfying*

$$\mathbb{E}[c(E[C])] \leq \left(\frac{3}{2} + \frac{1}{\mathrm{e}}\right) \cdot c_{LP} \ .$$

To prove Theorem 6, we first study the random graph $H_0 := (V, \dot{\bigcup}_{i\in\{1,\ldots,k\}} E[T_i])$ obtained from taking the union of trees $T_i \in \mathcal{T}_i$ for all $i \in \{1,\ldots,k\}$ as sampled in Algorithm 1. In order for the following statements to also be applicable in a proof of Theorem 2, we refer to the tree distributions of the type generated in Algorithm 1 as *connecting tree distributions.*

▶ **Definition 7** (Connecting tree distribution). *Let $G = (V,E)$ be a graph and let $d_1,\ldots,d_k \in V$. A connecting tree distribution $(\mathcal{T}_i, \mu^i)_{i\in\{1,\ldots,k\}}$ consists of a family $\mathcal{T}_i$ of subtrees of G and marginals $\mu^i \colon \mathcal{T}_i \to (0,1]$ for every $i \in \{1,\ldots,k\}$ with the following properties.*

(i) $\sum_{T\in\mathcal{T}_i} \mu^i_T = 1$ *for all* $i \in \{1,\ldots,k\}$.

(ii) $V[T] \cap \{d_1,\ldots,d_k\} = \{d_i, d_{i+1}\}$ *for all* $T \in \mathcal{T}_i$ *and* $i \in \{1,\ldots,k\}$.

(iii) $\sum_{i=1}^k \sum_{T\in\mathcal{T}_i \colon v\in V[T]} \mu^i_T = 1$ *for all* $v \in V \setminus \{d_1,\ldots,d_k\}$.

The distributions $(\mathcal{T}_i, \mu^i)$ obtained in Algorithm 1 by applying Lemma 5 indeed satisfy the constraints of the above definition; in particular, Item iii is fulfilled because

$$\sum_{i=1}^k \sum_{T\in\mathcal{T}_i \colon v\in V[T]} \mu^i_T = \sum_{i=1}^k y^i_v = 1 \quad \forall v \in V \setminus \{d_1,\ldots,d_k\} \ ,$$

where the first equality follows from Lemma 5, and the second one is implied by constraints of $P_{d_i\text{-}d_{i+1}\text{ stroll}}$.

▶ **Lemma 8.** *Let $G = (V,E)$ be a graph, $d_1,\ldots,d_k \in V$, and let $(\mathcal{T}_i, \mu^i)$ be a connecting tree distribution.*

(i) *For any choice of trees $T_i \in \mathcal{T}_i$ for $i \in \{1,\ldots,k\}$, the multigraph $H_0 := (V, \dot{\bigcup}_{i\in\{1,\ldots,k\}} E[T_i])$ consists of one large connected component and potentially several isolated vertices. The large connected component contains a walk with visits at $d_1,\ldots,d_k$ in this order that can be constructed efficiently from the trees T_i.*

(ii) *If, in the above construction, the trees T_i are sampled with marginals μ^i, we have that for all $v \in V \setminus \{d_1,\ldots,d_k\}$,*

$$\mathbb{P}[v \textit{ is isolated in } H_0] \leq \frac{1}{\mathrm{e}} \ .$$

Proof. For Item i observe that each tree T_i is connected within itself by definition and, as it contains d_i and d_{i+1}, the union of all trees form one large connected component, while all other components must be isolated vertices. Also, because each tree T_i contains a d_i-d_{i+1} path, we may concatenate these paths to obtain the desired walk with visits at d_1, …, d_k in this order.

To prove Item ii, we calculate the probability that a vertex $v \in V \setminus \{d_1,\ldots,d_k\}$ is isolated in H_0. First, note that for any such vertex v and any $i \in \{1,\ldots,k\}$, we have

$$\mathbb{P}\left[v \notin V\left[T_i\right]\right] = 1 - \sum_{T\in\mathcal{T}_i \colon v\in V[T]} \mu^i_T \ .$$

Thus, the probability that a vertex $v \in V \setminus \{d_1, \dots, d_k\}$ is not contained in any tree T_i for $i \in \{1, \dots, k\}$, and hence is isolated in H_0, can be bounded as follows:

$$\mathbb{P}\left[v \notin \bigcup_{i=1}^{k} V[T_i]\right] = \prod_{i=1}^{k} \mathbb{P}[v \notin V[T_i]] = \prod_{i=1}^{k} \left(1 - \sum_{T \in \mathcal{T}_i \colon v \in V[T]} \mu_T^i\right)$$
$$\leq \exp\left(-\sum_{i=1}^{k} \sum_{T \in \mathcal{T}_i \colon v \in V[T]} \mu_T^i\right) = \frac{1}{\mathrm{e}} \ ,$$

where we used that $1 - t \leq \exp(-t)$ for all $t \in \mathbb{R}$, and $\sum_{i=1}^{k} \sum_{T \in \mathcal{T}_i \colon v \in V[T]} \mu_T^i = 1$ because $(\mathcal{T}_i, \mu^i)$ is a connecting tree distribution. ◀

Next, we bound the cost of the minimum-cost connector F computed in Line 1 of Algorithm 1.

▶ **Lemma 9.** *Let $G = (V, E)$ be a graph with metric edge costs c, let $d_1, \dots, d_k \in V$, and let T be a minimum-cost spanning tree of (G, c).*

(i) *For all $v \in V \setminus \{d_1\}$, let e_v denote the unique edge outgoing of v when orienting T towards d_1. For every graph H_0 on the vertex set V with components that are – up to possibly the component containing d_1 – singleton vertices, the minimum-cost edge set F that connects H_0 satisfies*

$$c(F) \leq \sum_{v \text{ isolated in } H_0} c(e_v) \ .$$

(ii) *Let $(\mathcal{T}_i, \mu^i)$ for $i \in \{1, \dots, k\}$ be a connecting tree distribution. If the trees $T_i \in \mathcal{T}_i$ are sampled with marginals μ^i and $H_0 := (V, \dot\bigcup_{i \in \{1,\dots,k\}} E[T_i])$, we obtain*

$$\mathbb{E}[c(F)] \leq \frac{1}{\mathrm{e}} c(T) \ .$$

Proof. In order to prove Item i, we construct a feasible connecting edge set F' as the set of all edges e_v for which v is an isolated vertex. Then $H_0 \cup F'$ is indeed connected, because each isolated vertex of H_0 is connected to its predecessor in T by an edge of F', hence inductively, the component of H_0 containing d_1 can be reached along edges of F'. As the minimum-cost connector F has cost at most $c(F')$, we have

$$c(F) \leq c(F') \leq \sum_{v \text{ isolated in } H_0} c(e_v) \ .$$

To prove Item ii, we note that in this case, H_0 consists of one large connected component and some isolated vertices by Item i of Lemma 8. Using Item ii of Lemma 8 on top of the above, we get

$$\mathbb{E}[c(F)] \leq \mathbb{E}[c(F')] = \sum_{v \in V \setminus \{d_1\}} \mathbb{P}[v \text{ isolated in } H_0] \cdot c(e_v) \leq \frac{1}{\mathrm{e}} \sum_{v \in V \setminus \{d_1\}} c(e_v) = \frac{1}{\mathrm{e}} c(E[T]) \ . ◀$$

The cost of the odd(H)-join J constructed in Line 1 of Algorithm 1 can be bounded by $\frac{1}{2} c^\top x$ by Item ii of Corollary 4. Hence, to complete the analysis of Algorithm 1, it is left to show that from the Eulerian graph $H \dot\cup J$ constructed in Line 1 of Algorithm 1, we can obtain an OTSP solution of no larger cost. We remark that such a step has also been used by Böckenhauer, Mömke, and Steinová [7]; we repeat it here explicitly and give a slightly different proof for completeness. In the proof, we repeatedly use the operation of *shortcutting* a vertex v on a walk, which is the following: If the predecessor and successor of v on the walk are u and w, respectively, we delete the edges $\{u, v\}$ and $\{v, w\}$ from the walk and add the direct edge $\{u, w\}$ instead. It is clear that this operation results in a walk again; furthermore, by the triangle inequality, the costs of the walk do not increase under such operations.

▶ **Lemma 10.** *Let $G = (V, E)$ be a complete graph with metric edge costs c, and let $d_1, \dots, d_k \in V$ be distinct. Given an undirected connected Eulerian multigraph $M = (V, E_M)$ together with a closed walk in M with visits at $d_1, \dots, d_k$ in this order, we can in polynomial time determine a spanning cycle C in G with visits at $d_1, \dots, d_k$ in this order of cost at most $c(E_M)$.*

Proof. Let C be the given closed walk on which $d_1, \dots, d_k$ appear in this order, delete C from M and partition the remaining Eulerian graph into a set $\mathcal{W}$ of closed walks. Shortcut C to a cycle while maintaining visits at $d_1, \dots, d_k$ in this order. This can, for example, be done by traversing C starting at d_1, and shortcutting

(i) vertices that have already been visited, and

(ii) vertices d_i that are not yet to be visited due to the order constraint.

Afterwards, as long as $\mathcal{W}$ is non-empty, pick a closed walk W from $\mathcal{W}$ that intersects C, and let v be a vertex in the intersection. Traversing W starting from v, shortcut W to a cycle by skipping, except for v itself, all vertices that are already contained in C. Then, merge W into C by first traversing C up to (and including) v, then completely traversing W until (but not including) v before continuing on C, thereby including only one visit at v in the updated C. It is immediate that C is still a cycle after any such operation, and the vertices $d_1, \dots, d_k$ still appear on C once and in this order. By connectivity of M, this procedure only terminates once $\mathcal{W}$ is empty, and in that case, C is a spanning cycle of G. Also, all steps can be implemented to run in polynomial time. Clearly, the final length of C with respect to c is at most $c\,(E_M)$ because c is metric. ◀

From the above ingredients, we can readily prove Theorem 6.

Proof of Theorem 6. The solution returned by Algorithm 1 is a spanning cycle C in G obtained from $H \dot\cup J$ through Lemma 10, hence it is feasible and of cost at most $c(E[H \dot\cup J])$. Note that the required closed walk in $H \dot\cup J$ with visits at $d_1, \dots, d_k$ in this order is guaranteed and can be constructed efficiently from the trees T_i by Item i of Lemma 8. Furthermore, by Item ii of Lemma 9 and Corollary 4, we know that $\mathbb{E}[c(F)] \leq {}^1\!/\!_\mathrm{e} \cdot c(T) \leq {}^1\!/\!_\mathrm{e} \cdot c_{\mathrm{LP}}$, where T is a minimum-cost spanning tree. In addition, Corollary 4 also implies that $c(E[J]) \leq \frac{1}{2} \cdot c_{\mathrm{LP}}$. Last but not least, we can express the expected cost of each T_i as

$$\mathbb{E}[c(E[T_i])] = \sum_{T \in \mathcal{T}_i} \mu_T^i c(E[T]) = \sum_{T \in \mathcal{T}_i} \mu_T^i c^\top \chi^{E[T]} = c^\top x^i \qquad \forall i \in \{1, \dots, k\} \enspace .$$

Thus, by summing over all constructed trees, we obtain $\sum_{i=1}^k \mathbb{E}[c(E[T_i])] = \sum_{i=1}^k c^\top x^i = c_{\mathrm{LP}}$. Together, this yields the proclaimed bound

$$\mathbb{E}[c(C)] \leq \mathbb{E}\left[c(E[H \dot\cup J])\right] \leq \left(\frac{3}{2} + \frac{1}{\mathrm{e}}\right) \cdot c_{\mathrm{LP}} \enspace .$$

It remains to note that Algorithm 1 can be implemented to run in polynomial time. To start with, an optimal solution of the OTSP LP relaxation can be found in polynomial time because $P_{s\text{-}t \text{ stroll}}$ admits a polynomial-time separation oracle through polynomially many calls to a minimum-cut algorithm. Next, the decomposition in Line 1 is obtained in polynomial time, finding an optimal edge set F in Line 1 can be implemented by Prim's algorithm, and the odd(H)-join is well-known to be computable in polynomial time. Finally, also the computation of the cycle C in Line 1 is polynomial due to Lemma 10, concluding the proof. ◀

2.3 Derandomizing Algorithm 1

To complete a proof of our main result, Theorem 1, we now show how to derandomize Algorithm 1 using the method of conditional expectations, which results in the following proof.

Proof of Theorem 1. By the construction of the solution C in Algorithm 1, using Item i of Lemma 9 to bound the cost of F, and Item ii of Corollary 4 to bound the cost of J, we know that

$$\begin{aligned} c(C) &\leq \sum_{i=1}^{k} c(E[T_i]) + c(F) + c(E[J]) \\ &\leq \underbrace{\sum_{i=1}^{k} c(E[T_i]) + \sum_{v \notin \bigcup_{i=1}^{k} V[T_i]} c(e_v) + \frac{1}{2} \cdot c_{\mathrm{LP}}}_{=:g(T_1,\ldots,T_k)} \ , \end{aligned} \tag{2}$$

where we recall that e_v, for $v \in V \setminus \{d_1\}$, is the unique outgoing edge at v when orienting a minimum-cost spanning tree of G towards d_1. For Theorem 6, we showed that $\mathbb{E}[g(T_1, \ldots, T_k)] \leq (3/2 + 1/e) \cdot c_{\mathrm{LP}}$. Following the method of conditional expectations, in order to derandomize the choices of the trees T_i in Line 1 of Algorithm 1 while maintaining the upper bound on the solution cost, we sequentially choose trees S_i for $i \in \{1, \ldots, k\}$ such that

$$S_i = \operatorname*{arg\,min}_{S \in \mathcal{T}_i} \ \mathbb{E}\left[g(T_1, \ldots, T_k) \mid T_1 = S_1, \ldots, T_{i-1} = S_{i-1}, T_i = S\right] \ . \tag{3}$$

Note that feasibility of the cycle C and the bound of (2) on its cost are unaffected by fixing $T_i = S_i$. By definition of conditional expectation, we know that

$$\begin{aligned} &\mathbb{E}\left[g(T_1, \ldots, T_k) \mid T_1 = S_1, \ldots, T_{i-1} = S_{i-1}\right] \\ &\qquad = \sum_{S \in \mathcal{T}_i} \mu_S^i \cdot \mathbb{E}\left[g(T_1, \ldots, T_k) \mid T_1 = S_1, \ldots, T_{i-1} = S_{i-1}, T_i = S\right] \ , \end{aligned}$$

hence the sequence of conditional expectations

$$(\mathbb{E}[g(T_1, \ldots, T_k) \mid T_1 = S_1, \ldots, T_i = S_i])_{i \in \{1,\ldots,k\}}$$

is non-increasing by the choice in (3), because $\sum_{S \in \mathcal{T}_i} \mu_S = 1$. Thus, it remains to observe that the conditional expectations in (3) can be computed. To this end, observe that

$$\begin{aligned} &\mathbb{E}\left[g(T_1, \ldots, T_k) \mid T_1 = S_1, \ldots, T_\ell = S_\ell\right] \\ &\quad = \sum_{i=1}^{\ell} c(E[S_i]) + \sum_{i=\ell+1}^{k} \mathbb{E}[c(E[T_i])] + \sum_{v \notin \bigcup_{i=1}^{\ell} V[S_i]} \mathbb{P}\left[v \notin \bigcup_{i=\ell+1}^{k} V[T_i]\right] c(e_v) + \frac{1}{2} \cdot c_{\mathrm{LP}} \ , \end{aligned}$$

and we can readily compute

$$\mathbb{E}[c(E[T_i])] = \sum_{T \in \mathcal{T}_i} \mu_T^i c(E[T]) \qquad \text{and} \qquad \mathbb{P}\left[v \notin \bigcup_{i=\ell+1}^{k} V[T_i]\right] = \prod_{i=\ell+1}^{k} (1 - y_v^i) \ .$$ ◀

3 Extending to several independent total orders: Proving Theorem 2

In this section, we show how our approach can be extended to TSP-PC with a specific structure of precedence constraints that corresponds to having total orders on disjoint subsets $D_1, \dots, D_\ell \subseteq V$ of the input graph $G = (V, E)$.

As mentioned in the introduction, our approach is inherently tied to handle total orders – which is why, in the aforementioned setup, our solutions will not interleave vertices from different chains D_j, but rather treat the chains D_j one after another. Still, our approach allows to do better than simply constructing OTSP solutions for all subinstances (G, c, D_j) in a black-box way and concatenating them with appropriate shortcutting. The latter would lead to an immediate $(3/2 + 1/e)\ell$-approximate solution by using Algorithm 1 on each subinstance. Instead, we observe that after solving the OTSP LP relaxation and sampling trees for each subinstance as in Algorithm 1, we may join all edges obtained this way and only *once* need to connect remaining singletons and do parity correction. This leads to Algorithm 2 as stated below.

Note that, deviating from the above outline, Algorithm 2 starts by guessing a root node d_0 among the minimal nodes in all sets D_j with respect to $\prec$; this node is used as a common anchor of the given partial orders and results in connectivity of the multigraph containing all sampled trees. To be able to compare the obtained solution to an optimal solution, we need d_0 to be, among the minimal nodes in all sets D_j, the first one to appear on an optimal solution. We remark that for one $j \in \{1, \dots, \ell\}$, we already have $d_0 \in D_j$. For the sake of uniform notation, we still add a copy of d_0 to D_j in Line 2 of Algorithm 2.

Algorithm 2 Approximating a special case of TSP-PC.

```
Input: TSP-PC instance $(G, c, \prec)$ on graph $G = (V, E)$, where $\prec$ precisely induces
       total orders on disjoint subsets $D_1, \dots, D_\ell \subseteq V$.

1 Guess a root node $d_0$ among the minimal nodes in $D_i$ with respect to $\prec$.
2 foreach $j \in \{1, \dots, \ell\}$ do
3     Compute an optimal solution $(x^{ji}, y^{ji})_{i \in \{0,1,\dots,|D_j|\}}$ to the OTSP LP relaxation
      for the OTSP instance $(G, c, \{d_0\} \dot\cup D_j)$ with an order given by $\prec$ extended by
      $d_0 \prec D_j$.
4     foreach $i \in \{0, 1, \dots, |D_j|\}$ do
5         Apply Lemma 5 to decompose $(x^{ji}, y^{ji})$ into trees $\mathcal{T}_{ji}$ with weights $\mu^{ji}$.
6         Sample one tree $T_{ji}$ from $\mathcal{T}_{ji}$ with marginals given by $\mu^{ji}$.
7 Compute a minimum-cost edge set $F \subseteq E$ such that the multigraph
      $$H := \left(V, F \cup \dot\bigcup_{j=1}^{\ell} \dot\bigcup_{i \in \{1,\dots,|D_j|\}} E[T_{ji}]\right)$$
  is connected.
8 Let $Q = \mathrm{odd}(H)$ and compute a minimum cost $Q$-join $J$ in $G$.
9 return Shortest spanning cycle $C$ in $G$ (over all guesses of $d_0$) that visits $d_0$,
  $D_1 \setminus \{d_0\}$, ..., $D_\ell \setminus \{d_0\}$ in this order (while respecting $\prec$ in each $D_i$) and is
  obtained from $H \dot\cup J$ through Lemma 10.
```

We show that this algorithm gives the guarantee claimed by Theorem 2 in expectation, and that it can be derandomized using the method of conditional expectations in a way analogous to the derandomization of Algorithm 1.

Proof of Theorem 2. Let c_{OPT} denote the cost of an optimal solution of the given TSP-PC instance. For every $j \in \{1, \dots, \ell\}$, note that the value c^j_{LP} of the optimal solution $(x^{ji}, y^{ji})_{i \in \{1,\dots,|D_j|\}}$ to the OTSP instance $(G, c, \{d_0\} \dot\cup D_j)$ generated in Line 2 of Algorithm 2 satisfies $c^j_{\mathrm{LP}} \le c_{\mathrm{OPT}}$. For every $j \in \{1, \dots, \ell\}$, denote

$$H_j := \left(V, \dot\bigcup_{i=0}^{|D_j|} E[T_{ji}] \right) \quad .$$

Every such graph is composed of trees from a connecting tree distribution. Hence, by Item i of Lemma 8, H_j consists of a large connected component that contains a walk with visits at d_0 and all vertices of D_j in the desired order, and potentially isolated vertices. For all $v \notin \{d_0\} \cup D_j$, Item ii of Lemma 8 implies that

$$\mathbb{P}[v \text{ is isolated in } H_j] \le \frac{1}{\mathrm{e}} \quad .$$

Also, observe that

$$\mathbb{E}[c(E[H_j])] = \sum_{i=0}^{|D_j|} \mathbb{E}[c(T_{ji})] = \sum_{i=0}^{|D_j|} \sum_{T \in \mathcal{T}_{ji}} \mu^{ji}_T c(E[T]) = \sum_{i=0}^{|D_j|} c^\top x^{ji} = c^j_{\mathrm{LP}} \le c_{\mathrm{OPT}} \quad .$$

Consequently, the multigraph $H_0 := \dot\bigcup_{j \in \{1,\dots,\ell\}} H_j$ has total edge cost at most $\ell \cdot c_{\mathrm{OPT}}$. Furthermore, H_0 has one large connected component that contains a walk with visits at d_0, $D_1 \setminus \{d_0\}$, …, $D_j \setminus \{d_0\}$ in this order (obtained by concatenating the walks obtained in the graphs H_j above), i.e., a walk that respects $\prec$. Also, because the graphs $H_1, \dots, H_\ell$ are independent,

$$\mathbb{P}[v \text{ is isolated in } H_0] = \prod_{j=1}^{\ell} \mathbb{P}[v \text{ is isolated in } H_j] \le \frac{1}{\mathrm{e}^\ell} \quad .$$

Hence, by Item i of Lemma 9, the cost of the minimum-cost edge set F connecting H_0, as constructed in Line 2 of Algorithm 2, can be bounded as follows:

$$\mathbb{E}[c(F)] \le \sum_{v \text{ isolated in } H_0} \mathbb{P}[v \text{ is isolated in } H_0] \cdot c(e_v) \le \frac{1}{\mathrm{e}^\ell} \cdot c(T) \le \frac{1}{\mathrm{e}^\ell} \cdot c_{\mathrm{OPT}} \quad .$$

Here, we used that for any $j \in \{1, \dots, \ell\}$, we have $c(T) \le c^j_{\mathrm{LP}}$ by Item i of Corollary 4, and $c^j_{\mathrm{LP}} \le c_{\mathrm{OPT}}$ as mentioned above. Similarly, by Item ii of Corollary 4, we know that the cost of a cheapest odd(H)-join J in the multigraph $H = H_0 \cup F$ can be bounded by $c(E[J]) \le \frac{1}{2} \cdot c^j_{\mathrm{LP}}$ for any $j \in \{1, \dots, \ell\}$, hence $c(E[J]) \le \frac{1}{2} \cdot c_{\mathrm{OPT}}$.

Altogether, we obtain a connected Eulerian multigraph $H \dot\cup J$ together with a walk that has visits at d_0, $D_1 \setminus \{d_0\}$, …, $D_j \setminus \{d_0\}$ in the order given by $\prec$, and

$$\mathbb{E}[c(E[H \dot\cup J])] \le \left(\ell + \frac{1}{2} + \frac{1}{\mathrm{e}^\ell} \right) \cdot c_{\mathrm{OPT}} \quad .$$

Thus, by Lemma 10, we can efficiently find a cycle with visits at d_0, $D_1 \setminus \{d_0\}$, …, $D_j \setminus \{d_0\}$ in the order given by $\prec$ of at most the above expected cost.

Finally, to derandomize the random selection of trees T_{ji} in Algorithm 2, we observe that the present randomized analysis relies on a bound of the form

$$c(C) \leq \sum_{j=1}^{\ell} \sum_{i=0}^{|D_j|} c(T_{ji}) + \sum_{v \notin \dot{\bigcup}_{j\in\{1,\dots,\ell\}} \dot{\bigcup}_{i\in\{1,\dots,|D_j|\}} V[T_{ji}]} c(e_v) + \frac{1}{2} \cdot c_{\mathrm{OPT}} \quad .$$

The conditional expectations of this bound with respect to fixing any subset of the trees T_{ji} can be readily computed. Thus, the derandomization works analogously to Algorithm 1 by the method of conditional expectations, in each iteration fixing one of the T_{ji}. To complete the proof of Theorem 6, we observe that all steps of Algorithm 2 can be implemented to run in polynomial time. ◀

▶ Remark 11. We remark that the analysis of Algorithm 2 above is with respect to the actual cost c_{OPT} of an optimal TSP-PC solution. Alternatively, after guessing a root node d_0, one could also write an LP relaxation generalizing the OTSP LP relaxation by introducing independent copies of the variables for each chain $\{d_0\} \cup D_j$ and minimizing the cost of a point $x \in P_{\mathrm{HK}}(G)$ that dominates the edge usage within each of the copies. For the ease of presentation, though, we decided to present the above analysis only.

4 Proof of Lemma 5

As mentioned earlier, we derive Lemma 5 from a closely related result used by Blauth and Nägele [5, Lemma 4.2]. We restate their result here in a slightly simplified form that follows immediately from the original formulation.

▶ **Lemma 12** ([5, Lemma 4.2]). *Let $G = (V, E)$ be a graph with $r \in V$, let $(x, y) \in \mathbb{R}^E_{\geq 0} \times \mathbb{R}^V_{\geq 0}$ be feasible for the system*

$$\begin{array}{rcll} x(\delta(v)) & = & 2y_v & \forall v \in V \\ x(\delta(S)) & \geq & 2y_v & \forall S \subseteq V \setminus \{r\}, v \in S \\ y_r & = & 1 \ , & \end{array} \tag{4}$$

and assume that there is a vertex $u \in V \setminus \{r\}$ such that $y_u = 1$ and $e_0 = \{u, r\}$ satisfies $x_{e_0} \geq 1$. We can in polynomial time construct a set $\mathcal{T}$ of trees that all contain the vertices r and u, and weights $\mu \in [0, 1]^{\mathcal{T}}$ with $\sum_{T\in\mathcal{T}} \mu_T = 1$ and the following properties.

(i) *The point $x \in \mathbb{R}^E_{\geq 0}$ is a conic combination of the trees in $\mathcal{T}$ with weights μ and the edge e_0, i.e.,*

$$x = \sum_{T\in\mathcal{T}} \mu_T \chi^{E[T]} + \chi^{e_0} \quad .$$

(ii) *For every $v \in V \setminus U$,*

$$\sum_{T\in\mathcal{T}\colon v\in V[T]} \mu_T = y_v \quad .$$

The proof of Lemma 12 relies on the well-known *splitting-off* technique (see, e.g., [16,30,31]) applied in the graph G with weights x. Indeed, the constraints in the system (4) can be interpreted as r-v connectivity requirements for all $v \in V \setminus \{r\}$, hence splitting-off allows to remove a vertex from the graph while preserving the connectivity properties of the remaining graph. An inductive construction of the desired family of trees is then achieved by reverting the splitting-off operations and extending trees appropriately. For a complete proof, we refer to Blauth and Nägele [5].

To deduce Lemma 5 from Lemma 12, we note that a point $(x, y) \in P_{s\text{-}t \text{ stroll}}$ can be easily transformed into a point (x', y') satisfying the assumptions of Lemma 12 by adding one unit to $x_{\{s,t\}}$ and adjusting y_s and y_t accordingly. Note that intuitively, this corresponds to closing an *s*-*t* stroll to obtain a tour by adding a copy of the edge $\{s, t\}$.

Proof of Lemma 5. Given $(x, y) \in P_{s\text{-}t \text{ stroll}}$, we assume without loss of generality that $e_0 := \{s, t\} \in E$ and define $x' := x + \chi^{\{s,t\}}$ and $y' = y + \frac{1}{2}(\chi^s + \chi^t)$. We claim that (x', y') with $r = s$ and $u = t$ satisfy the assumptions of Lemma 12. Indeed, $y'_s = y'_t = 1$, and $x'_{e_0} = x_{e_0} + 1 \geq 1$. Moreover, for $v \notin \{s, t\}$, we have $x'(\delta(v)) = x(\delta(v)) = 2y_v$; for $v \in \{s, t\}$, we have $x'(\delta(v)) = x(\delta(v)) + 1 = 2 = 2y'_v$, hence the degree constraints in (4) are satisfied. Finally, to verify that the cut constraints of (4) are satisfied, too, let $S \subseteq V \setminus \{r\}$ and $v \in S$. If $t \notin S$, then $x'(\delta(S)) = x(\delta(S)) \geq 2y_v = 2y'_v$ follows from the corresponding constraint of $P_{s\text{-}t \text{ stroll}}$. If otherwise $t \in S$, we know that $x(\delta(S)) \geq 1$, hence

$$x'(\delta(S)) = x(\delta(S)) + 1 \geq 2 \geq 2y'_v \enspace ,$$

where we use that $y'_v = y_v \leq 1$ is implied by the constraints of $P_{s\text{-}t \text{ stroll}}$ for $v \in V \setminus \{s, t\}$ (see Footnote 5), and $y'_s = y'_t = 1$.

Consequently, by applying Lemma 12 to (x', y'), we obtain in polynomial time a set $\mathcal{T}$ of trees that all contain s and t, and weights $\mu \in [0, 1]^{\mathcal{T}}$ with $\sum_{T \in \mathcal{T}} \mu_T = 1$ such that

$$x + \chi^{e_0} = x' = \sum_{T \in \mathcal{T}} \mu_T \chi^{E[T]} + \chi^{e_0} \enspace ,$$

i.e., $x = \sum_{T \in \mathcal{T}} \mu_T \chi^{E[T]}$, and, for every $v \in V \setminus \{s, t\}$,

$$\sum_{T \in \mathcal{T} \colon v \in V[T]} \mu_T = y'_v = y_v \enspace . \qquad ◀$$

References

1 Susanne Armbruster, Matthias Mnich, and Martin Nägele. A (3/2 + 1/e)-approximation algorithm for Ordered TSP, 2024. arXiv:2405.06244v1.

2 Egon Balas, Matteo Fischetti, and William R. Pulleyblank. The precedence-constrained asymmetric traveling salesman polytope. *Mathematical Programming*, 68(1):241–265, 1995. doi:10.1007/BF01585767.

3 Jørgen Bang-Jensen, András Frank, and Bill Jackson. Preserving and increasing local edge connectivity in mixed graphs. *SIAM Journal on Discrete Mathematics*, 8(2):155–178, 1995. doi:10.1137/S0036142993226983.

4 Jannis Blauth, Nathan Klein, and Martin Nägele. A better-than-1.6-approximation for prize-collecting TSP. In *Proceedings of the 23rd Conference on Integer Programming and Combinatorial Optimization (IPCO '24)*, pages 28–42, 2024. doi:10.1007/978-3-031-59835-7_3.

5 Jannis Blauth and Martin Nägele. An improved approximation guarantee for prize-collecting TSP. In *Proceedings of the 55th Annual ACM SIGACT Symposium on Theory of Computing (STOC '23)*, pages 1848–1861, 2023. doi:10.1145/3564246.3585159.

6 Hans-Joachim Böckenhauer, Juraj Hromkovič, Joachim Kneis, and Joachim Kupke. On the approximation hardness of some generalizations of TSP. In *Proceedings of the 10th Scandinavian Workshop on Algorithm Theory (SWAT '06)*, pages 184–195, 2006. doi:10.1007/11785293_19.

7 Hans-Joachim Böckenhauer, Tobias Mömke, and Monika Steinová. Improved approximations for TSP with simple precedence constraints. *Journal of Discrete Algorithms*, 21:32–40, 2013. doi:10.1016/j.jda.2013.04.002.

8 Martin Böhm, Zachary Friggstad, Tobias Mömke, and Joachim Spoerhase. Approximating TSP variants using a bridge lemma, 2024. arXiv:2405.12876.

9 Moses Charikar, Rajeev Motwani, Prabhakar Raghavan, and Craig Silverstein. Constrained TSP and low-power computing. In *Proceedings of the 5th International Workshop on Algorithms and Data Structures (WADS '97)*, pages 104–115, 1997. doi:10.1007/3-540-63307-3_51.

10 N. Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical Report 388, Graduate School of Industrial Administration, Carnegie Mellon University, 1976.

11 N. Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. *Operations Research Forum*, 3, 2022. doi:10.1007/s43069-021-00101-z.

12 G. Dantzig, R. Fulkerson, and S. Johnson. Solution of a large-scale traveling-salesman problem. *Journal of the Operations Research Society of America*, 2(4):393–410, 1954. doi:10.1287/opre.2.4.393.

13 Sami Davies, Janardhan Kulkarni, Thomas Rothvoss, Jakub Tarnawski, and Yihao Zhang. Scheduling with communication delays via LP hierarchies and clustering. In *Proceedings of the 61st Annual IEEE Symposium on Foundations of Computer Science (FOCS '20)*, pages 822–833, 2020. doi:10.1109/FOCS46700.2020.00081.

14 Vladimir G. Deĭneko, Michael Hoffmann, Yoshio Okamoto, and Gerhard J. Woeginger. The traveling salesman problem with few inner points. *Operations Research Letters*, 34(1):106–110, 2006. doi:10.1016/j.orl.2005.01.002.

15 Jianzhong Du, Joseph Y-T. Leung, and Gilbert H. Young. Scheduling chain-structured tasks to minimize makespan and mean flow time. *Information and Computation*, 92(2):219–236, 1991. doi:10.1016/0890-5401(91)90009-Q.

16 András Frank. On a theorem of Mader. *Discrete Mathematics*, 101(1):49–57, 1992. doi:10.1016/0012-365X(92)90589-8.

17 M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, first edition edition, 1979.

18 L. Gouveia and P. Pesneau. On extended formulations for the precedence constrained asymmetric traveling salesman problem. *Networks*, 48(2):77–89, 2006. doi:10.1002/net.20122.

19 R. L. Graham. Bounds on multiprocessing timing anomalies. *SIAM Journal on Applied Mathematics*, 17(2):416–429, 1969. doi:10.1137/0117039.

20 Michael Held and Richard M. Karp. The traveling-salesman problem and minimum spanning trees. *Operations Research*, 18(6):1138–1162, 1970. doi:10.1287/opre.18.6.1138.

21 Klaus Jansen and Roberto Solis-Oba. Approximation schemes for scheduling jobs with chain precedence constraints. *International Journal of Foundations of Computer Science*, 21(01):27–49, 2010. doi:10.1142/S0129054110007118.

22 Anna R. Karlin, Nathan Klein, and Shayan Oveis Gharan. A (slightly) improved approximation algorithm for metric TSP. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC '21)*, pages 32–45, 2021. doi:10.1145/3406325.3451009.

23 Anna R. Karlin, Nathan Klein, and Shayan Oveis Gharan. A deterministic better-than-3/2 approximation algorithm for metric TSP. In *Proceedings of the 22nd Conference on Integer Programming and Combinatorial Optimization (IPCO '23)*, pages 261–274, 2023. doi:10.1007/978-3-031-32726-1_19.

24 Marek Karpinski, Michael Lampis, and Richard Schmied. New inapproximability bounds for TSP. *Journal of Computer and System Sciences*, 81(8):1665–1677, 2015. doi:10.1016/j.jcss.2015.06.003.

25 Daniil Khachai, Ruslan Sadykov, Olga Battaia, and Michael Khachay. Precedence constrained generalized traveling salesman problem: Polyhedral study, formulations, and branch-and-cut algorithm. *European Journal of Operational Research*, 309(2):488–505, 2023. doi:10.1016/j.ejor.2023.01.039.

26 Manfred Kunde. Nonpreemptive LP-scheduling on homogeneous multiprocessor systems. *SIAM Journal on Computing*, 10(1):151–173, 1981. doi:10.1137/0210012.

27 J.K. Lenstra and A.H.G. Rinnooy Kan. Complexity results for scheduling chains on a single machine. *European Journal of Operational Research*, 4(4):270–275, 1980. doi:10.1016/0377-2217(80)90111-3.

28 Elaine Levey and Thomas Rothvoss. A $(1+\varepsilon)$-approximation for makespan scheduling with precedence constraints using LP hierarchies. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing (STOC '16)*, pages 168–177, 2016. doi:10.1145/2897518.2897532.

29 Mohammad Mehdi Liaee and Hamilton Emmons. Scheduling families of jobs with setup times. *International Journal of Production Economics*, 51(3):165–176, 1997. doi:10.1016/S0925-5273(96)00105-3.

30 L. Lovász. On some connectivity properties of Eulerian graphs. *Acta Mathematica Academiae Scientiarum Hungarica*, 28(1):129–138, 1976. doi:10.1007/BF01902503.

31 W. Mader. A reduction method for edge-connectivity in graphs. *Annals of Discrete Mathematics*, 3:145–164, 1978. doi:10.1016/S0167-5060(08)70504-1.

32 Christos H. Papadimitriou and Mihalis Yannakakis. The traveling salesman problem with distances one and two. *Mathematics of Operations Research*, 18(1):1–11, 1993. doi:10.1287/moor.18.1.1.

33 Yaroslav Salii. Revisiting dynamic programming for precedence-constrained traveling salesman problem and its time-dependent generalization. *European Journal of Operational Research*, 272(1):32–42, 2019. doi:10.1016/j.ejor.2018.06.003.

34 Sophia Saller, Jana Koehler, and Andreas Karrenbauer. A systematic review of approximability results for traveling salesman problems leveraging the TSP-T3CO definition scheme, 2023. arXiv:2311.00604.

35 A. I. Serdyukov. O nekotorykh ekstremal'nykh obkhodakh v grafakh. *Upravlyaemye sistemy*, 17:76–79, 1987. URL: http://nas1.math.nsc.ru/aim/journals/us/us17/us17_007.pdf.

36 Ola Svensson. Conditional hardness of precedence constrained scheduling on identical machines. In *Proceedings of the 42nd Annual ACM SIGACT Symposium on Theory of Computing (STOC '10)*, pages 745–754, 2010. doi:10.1145/1806689.1806791.

37 Gerhard J. Woeginger. A comment on scheduling on uniform machines under chain-type precedence constraints. *Operations Research Letters*, 26(3):107–109, 2000. doi:10.1016/S0167-6377(99)00076-0.

38 Laurence A. Wolsey. *Heuristic analysis, linear programming and branch and bound*, pages 121–134. Springer Berlin Heidelberg, 1980. doi:10.1007/BFb0120913.