A Better-Than-1.6-Approximation for Prize-Collecting TSP

Jannis Blauth[*] Nathan Klein[‡] Martin Nägele[§]

Abstract

Prize-Collecting TSP is a variant of the traveling salesperson problem where one may drop vertices from the tour at the cost of vertex-dependent penalties. The quality of a solution is then measured by adding the length of the tour and the sum of all penalties of vertices that are not visited. We present a polynomial-time approximation algorithm with an approximation guarantee slightly below 1.6, where the guarantee is with respect to the natural linear programming relaxation of the problem. This improves upon the previous best-known approximation ratio of 1.774. Our approach is based on a known decomposition for solutions of this linear relaxation into rooted trees. Our algorithm takes a tree from this decomposition and then performs a pruning step before doing parity correction on the remainder. Using a simple analysis, we bound the approximation guarantee of the proposed algorithm by $(1+\sqrt{5})/2 \approx 1.618$, the golden ratio. With some additional technical care we further improve it to 1.599.

[*]Research Institute for Discrete Mathematics and Hausdorff Center for Mathematics, University of Bonn, Bonn, Germany. Email: blauth@or.uni-bonn.de.

[‡]Institute for Advanced Study, Princeton, NJ, USA. Email: nklein@ias.edu. Research supported by Air Force Office of Scientific Research grant FA9550-20-1-0212 and NSF grants DGE-1762114 and CCF-1813135. This work was done while the author was employed at University of Washington, Seattle, WA, USA.

[§]Research Institute for Discrete Mathematics and Hausdorff Center for Mathematics, University of Bonn, Bonn, Germany. Email: mnaegele@uni-bonn.de. Supported by the Swiss National Science Foundation (grant no. P500PT_206742) and the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXZ-2047/1 – 390685813.

1 Introduction

The metric traveling salesperson problem (TSP) is one of the most fundamental problems in combinatorial optimization. In an instance of this problem, we are given a set V of n vertices along with their pairwise symmetric distances, $c\colon V \times V \to \mathbb{R}_{\geq 0}$, which form a metric. The goal is to find a shortest possible Hamiltonian cycle. In the classical interpretation, there is a salesperson who needs to visit a set of cities V and wants to minimize the length of their tour. In this work, we study a variant known as *prize-collecting* TSP, in which the salesperson can decide whether or not to include each city besides the starting one[1] in their tour at the cost of a city-dependent penalty. Formally, the problem can be stated as follows.

> **Prize-Collecting TSP (PCTSP):** Given a complete undirected graph $G = (V, E)$ with metric edge lengths $c_e \geq 0$ for all $e \in E$, a root $r \in V$, and penalties $\pi_v \geq 0$ for all $v \in V \setminus \{r\}$, the task is to find a cycle $C = (V_C, E_C)$ in G that contains the root r and minimizes
>
> $$\sum_{e \in E_C} c_e + \sum_{v \in V \setminus V_C} \pi_v \ .$$

This is a very natural generalization of TSP (one can recover TSP by setting $\pi_v = \infty$ for all $v \in V \setminus \{r\}$), as from the salesperson's perspective some cities may not be worth visiting if they significantly increase the length of the tour. Indeed, in many real-world settings instances of TSP are actually prize-collecting.

As mentioned, PCTSP is at least as hard as TSP. Thus, it is NP-hard to approximate within a factor of $^{123}/_{122}$ [KLS15]. On the positive side, the first constant-factor approximation algorithm for PCTSP was shown in the early '90s [BGSW93], giving a ratio of 2.5. After a series of improvements [GW95; ABHK11; Goe09], the best approximation factor is now slightly below 1.774 [BN23].

In TSP and many of its variants, such approximation guarantees typically rely on lower bounds obtained through *linear programming relaxations*. For PCTSP, the natural such formulation is the following:[2]

$$\begin{array}{lrcll} \min & \displaystyle\sum_{e \in E} c_e x_e + \sum_{v \in V} \pi_v (1 - y_v) & & & \\ & x(\delta(v)) & = & 2y_v & \forall v \in V \setminus \{r\} \\ & x(\delta(r)) & \leq & 2 & \\ & x(\delta(S)) & \geq & 2y_v & \forall S \subseteq V \setminus \{r\}, v \in S \\ & y_r & = & 1 & \\ & x_e & \geq & 0 & \forall e \in E \\ & y_v & \geq & 0 & \forall v \in V \ . \end{array} \tag{PCTSP LP relaxation}$$

One can see that $y_v \leq 1$ is implied by the above formulation, hence the variables y_v can be interpreted as the extent to which the vertex v is visited by the fractional solution. In this paper, we prove the following.

Theorem 1. *There is a polynomial-time LP-relative* 1.599*-approximation algorithm for PCTSP.*

To obtain this result, we exploit a known decomposition of solutions (x, y) to the above relaxation into trees (see Lemma 2 for the formal statement), which can be derived from an existential result on packing branchings in a directed multigraph by Bang-Jensen, Frank, and Jackson [BFJ95, Theorem 2.6] and was—in a generalized form—first used in the context of PCTSP by Blauth and Nägele [BN23]. The decomposition

[1] "Rooted" and unrooted versions are reducible to one another while preserving approximability, as noted in, e.g., [ABHK11]. Here, we always require a root vertex r.

[2] We use $\pi_r := 0$ for convenience. For $S \subseteq V$, we denote $\delta(S) := \{e \in E\colon |e \cap S| = 1\}$; for $v \in V$, we use $\delta(v) := \delta(\{v\})$.

can be interpreted as a distribution μ over a polynomial number of trees $\mathcal{T}$ rooted at r such that for each tree $T \in \mathcal{T}$ (i) $\mathbb{E}_{T\sim\mu}[c(E[T])] \le c^\top x$ and (ii) $\mathbb{P}_{T\sim\mu}[v \in V[T]] = y_v$ for all $v \in V$.[3]

Our algorithm proceeds as follows. We apply the decomposition to a slightly modified LP solution with $y_v = 0$ or $y_v \ge \delta$ for each $v \in V$ for some parameter δ. Then, for a tree T in the support of μ and a threshold γ, we prune the tree. Concretely, we find the inclusion-wise minimal subtree of T which spans all vertices $v \in V[T]$ with $y_v \ge \gamma$. Finally, we add the minimum cost matching on the odd degree vertices of this subtree. While our algorithm simply tries all possible trees T in the support of μ and all possible thresholds $\gamma = y_v$ for $v \in V$, our analysis is randomized: We sample the tree from μ and the threshold γ from a specific distribution, and prove the main result in expectation. Clearly, the same guarantee then holds for the best choice of T and γ.

After describing the algorithm in more detail in Section 2, we bound its approximation ratio by the golden ratio $(1+\sqrt{5})/2 \approx 1.618$ through a simple analysis in Section 3. In Section 4, we show that a minor adaption of the algorithm and a slightly more involved analysis allows us to push the approximation guarantee to 1.599 as in Theorem 1.

As mentioned, this improves upon the 1.774-approximation by Blauth and Nägele [BN23]. It remains open whether there is an efficient algorithm for PCTSP that matches—in terms of the approximation factor—the $3/2$-approximation for TSP by Christofides [Chr76] and Serdyukov [Ser87] (also see [Chr22; vS20]), or the current best known approximation guarantee for TSP, which is just slightly below $3/2$ [KKO21; KKO23]. The ideal result for PCTSP would be an algorithm that given an α-approximation for TSP produces an α-approximation for PCTSP (or possibly an $(\alpha+\varepsilon)$-approximation for every $\varepsilon > 0$). Such a result was recently shown for Path TSP [TVZ22], and as approximation algorithms for PCTSP begin to approach the threshold $3/2$, this possibility feels less out of reach.

1.1 Prior work on PCTSP

While Balas [Bal89] was the first to study prize-collecting variations of TSP, the first constant-factor approximation algorithm for PCTSP was given by Bienstock, Goemans, Simchi-Levi, and Williamson [BGSW93] through a simple threshold rounding approach: Starting from a solution (x, y) of the PCTSP LP relaxation, the Christofides-Serdyukov algorithm is used to construct a tour on all vertices $v \in V$ with $y_v \ge 3/5$, giving an LP-relative $5/2$-approximation. Goemans and Williamson [GW95] later obtained a 2-approximation through a classical primal-dual approach. More precisely, they showed how to compute a tree T with $c(E[T]) \le c^\top x$ and $\pi(V \setminus V[T]) \le \pi^\top(1-y)$, so that doubling the tree yields the 2-approximation.[4]

The factor of 2 was first beaten by Archer, Bateni, Hajiaghayi, and Karloff [ABHK11]. As a black-box subroutine, they use an approximation algorithm for TSP which we assume has ratio ρ. They achieved a $2 - \left(\frac{2-\rho}{2+\rho}\right)^2$ approximation, which—for $\rho = 3/2$—is approximately 1.979. Their algorithm runs the primal-dual algorithm of Goemans and Williamson and the ρ-approximation algorithm for TSP on a carefully selected node set, and outputs the better of the two tours. Goemans [Goe09] then observed that running both threshold rounding for different thresholds and the primal-dual algorithm, and choosing the best among the computed solutions yields an approximation guarantee of $1/(1 - \frac{1}{\beta}e^{1-2/\beta})$, where β denotes the approximation guarantee of an LP-relative approximation algorithm for TSP that is used in a black-box way. For $\beta = 3/2$, the guarantee of Goemans equals approximately 1.914. Goemans was the first to exploit a randomized analysis of threshold rounding, in which the threshold γ is chosen from a specific distribution.

Blauth and Nägele [BN23] refined the threshold rounding approach by sampling a connected subgraph

[3] For a (sub-)graph $H = (V_H, E_H)$, we write $V[H] := V_H$ and $E[H] := E_H$. Moreover, for a set F of edges we abbreviate $c(F) := \sum_{e\in F} c_e$.

[4] As observed in [BN23], such a tree—and thus an algorithm matching the guarantee of the primal-dual approach—can immediately be obtained from the decomposition of solutions of the PCTSP LP relaxation that was mentioned earlier and that is also used in this paper. Therefore, there are two elementary ways to get a 2-approximation.

such that each vertex $v \in V$ with $y_v \geq \gamma$ (again, γ denotes the threshold) is always contained in the vertex set of this subgraph, whereas each vertex $v \in V$ with $y_v < \gamma$ is contained with probability at least $\exp\left(-3y_v/4\gamma\right)$. Since each vertex below the threshold is guaranteed to have even degree in this subgraph, parities can be corrected at no extra cost, yielding an approximation guarantee of slightly below 1.774 through a randomized analysis. This guarantee beats those of the previously mentioned algorithms even for $\rho = 1$ and $\beta = 4/3$ (the integrality gap of the linear programming relaxation for TSP used in [Goe09], the Held-Karp relaxation, is at least $4/3$). Although the high-level idea of [BN23] is not too complicated, it requires a good deal of technical care to sample this subgraph and analyze the expected penalty cost. The tree construction in our algorithm is much simpler, which is also reflected in the analysis.

1.2 Related results

Alongside the general version targeted here, PCTSP was studied in special metric spaces. A PTAS is known for graph metrics in planar graphs [BCEHKM11] and in metrics with bounded doubling dimension [CJJ20]. For asymmetric edge costs satisfying the triangle inequality, a $\lceil \log(|V|) \rceil$-approximation is known [Ngu13].

Besides PCTSP, there is a wide class of other prize-collecting TSP variants, most of which originate from the work of Balas [Bal89]. Although PCTSP can be seen as the main variant in this problem class, there are other variants that include a lower bound on some minimum prize money that needs to be collected [Bal89; ALM00; ABLM07; Gar05], or an upper bound on the distance that can be traveled [BCKLMM07; CKP12; PFFSW20; PFFSW22; DFPS22].

Prize-collecting versions have also been studied for other classical combinatorial optimization problems. The most prominent example is the prize-collecting Steiner tree problem, which admits a 1.968-approximation [ABHK11], thereby going beyond the integrality gap of 2 of the natural linear programming relaxation. Also the more general prize-collecting Steiner forest problem (see, e.g., [HJ06]) admits an approximation guarantee going beyond the integrality gap of the natural LP relaxation [AGHJM24]. Interestingly, for the prize-collecting Steiner forest problem, it is known that the integrality gap is strictly larger than the one of the Steiner forest problem [KOPRSV17], indicating that prize-collecting aspects may in some cases make the problem strictly harder to approximate. To date, no such separation is known for TSP and PCTSP.

2 Our algorithm

Our algorithm follows the basic idea of the Christofides-Serdyukov algorithm for TSP, which is to combine a spanning tree T with a shortest $\mathrm{odd}(T)$-join[5], and shortcut an Eulerian tour in the resulting even-degree graph to obtain a cycle. The operation of adding a shortest $\mathrm{odd}(T)$-join to a tree T is also known as *parity correction*, as it results in a graph in which every vertex has even degree. Typically, for an even cardinality set Q (in particular, for $Q = \mathrm{odd}(T)$), the cost of a shortest Q-join is bounded by the cost $c^\top z$ of a point z that is feasible for the dominant of the Q-join polytope, which is given by (see [EJ73])

$$P^{\uparrow}_{Q\text{-join}} \coloneqq \{x \in \mathbb{R}^E \colon x(\delta(S)) \geq 1 \ \forall S \subseteq V \text{ with } |S \cap Q| \text{ odd}\} \ . \tag{1}$$

We use this approach with two additional variations: First, because our setting allows to not include some vertices in the returned tour, we use trees T that may not span all vertices in V. Second, we follow what is known as a Best-of-Many approach, i.e., we construct a polynomial-size set of trees, construct a tour from each of them, and return the best. For the analysis of such an approach, one typically provides a

[5] For a graph $G = (V, E)$, we denote by $\mathrm{odd}(G) \coloneqq \{v \in V \colon \deg_G(v) \text{ odd}\}$ the set of vertices with odd degree in G. Furthermore, for $Q \subseteq V$ of even cardinality, a Q-join is a set of edges that has odd degree precisely at vertices in Q. (For metric cost functions there is always a minimum cost Q-join that is a matching.)

distribution over the involved trees and analyzes the expected cost of the returned tour—which implies the same bound on the best tour.

We base our tree construction on the following decomposition lemma, which we restate here in the form given in [BN23]. For completeness, in Section 5 we replicate the proof provided in [BN23] with minor simplifications (as the proof in [BN23] shows a generalized version of Lemma 2). As mentioned earlier, this proof very closely follows the proof of an existential statement on packing branchings in a directed multigraph [BFJ95, Theorem 2.6].

Lemma 2 ([BN23, Lemma 12]). *Let (x, y) be a feasible solution of the PCTSP LP relaxation. We can in polynomial time compute a set $\mathcal{T}$ of trees that all contain the root r, and weights $\mu \in [0,1]^{\mathcal{T}}$ such that $\sum_{T\in\mathcal{T}} \mu_T = 1$,*

$$\sum_{T\in\mathcal{T}} \mu_T \cdot \chi^{E[T]} \leq x \ , \qquad \textit{and} \qquad \forall v \in V\colon \sum_{T\in\mathcal{T}\colon v\in V[T]} \mu_T = y_v \ .$$[6]

As also mentioned in Footnote 4, this lemma gives rise to an immediate 2-approximation for PCTSP that follows the framework described above: Choosing a tree $T \in \mathcal{T}$ with probability μ_T, and performing parity correction by doubling the tree gives a tour of expected length at most $2c^\top x$, while the expected penalty incurred for vertices that are not visited can easily be seen to be $\pi^\top(1-y)$.

It is an intriguing question whether parity correction can be done at expected cost $c^\top x/2$. Such a result would immediately lead to a $3/2$-approximation algorithm for PCTSP, matching the guarantee of the Christofides-Serdyukov algorithm for TSP. While in the setting of classical TSP, we have $x/2 \in P^{\uparrow}_{Q\text{-join}}$ for any set Q of even cardinality (and can thus bound the cost of parity correction for *any* tree T by $c^\top x/2$), this is no longer true in the prize-collecting setting.

Given a tree T, we first prune it to obtain a tree T' (in a way we will shortly explain) and then perform parity correction. To analyze the cost of the parity correction, we construct a point $z \in P^{\uparrow}_{\text{odd}(T)\text{-join}}$ that is of the form

$$z = \alpha \cdot x + \beta \cdot \chi^{E[T']} \tag{2}$$

for some coefficients $\alpha, \beta \in \mathbb{R}_{\geq 0}$. (In fact, we will choose different coefficients β for different parts of the tree.) As the existence of cuts $S \subsetneq V$ for which both $x(\delta(S))$ and $\delta_{T'}(S)$ are small may require to choose α and β large, we preprocess both the solution (x, y) of the PCTSP LP relaxation that we use as well as the trees that we obtain from it through Lemma 2.

In our first preprocessing step, we get rid of cuts S for which $x(\delta(S))$ is very small. To this end, observe that our algorithm may always drop vertices $v \in V \setminus \{r\}$ for which y_v is very small. Concretely, if our tour does not visit a vertex v with $y_v \leq \delta$ for some $\delta \in [0,1)$, we pay a penalty of π_v, which is at most a factor of $1/(1-\delta)$ larger than the fractional penalty $\pi_v(1-y_v)$ occurring in the LP objective. Thus, if we aim for an α-approximation algorithm, we may safely choose $\delta = 1 - 1/\alpha$ and drop all vertices with $y_v \leq \delta$. Crucially for our analysis, we can also perform this dropping on the level of solutions of the PCTSP LP relaxation by using the so-called *splitting off* technique [Lov76; Mad78; Fra92]. For a fixed vertex $v \in V$, splitting off allows to decrease the x-weight on two well-chosen edges $\{v, s\}$ and $\{v, t\}$ incident to v (and thereby also the value y_v) while increasing the weight on the edge $\{s, t\}$ by the same amount, without affecting feasibility for the PCTSP LP relaxation.[7] Note that such a *feasible splitting* at v does not increase the cost of the solution by the triangle inequality. A sequence of feasible splittings at v that result in $y_v = 0$ is called a *complete splitting*. Complete splittings always exist [Fra92] and can be found in polynomial time through a polynomial number of minimum s-t cut computations by trying all candidate pairs of edges (see, e.g., [NI97; Nag06] for more efficient procedures). Summarizing the above directly gives the following.

[6] For a set $F \subseteq E$ of edges, we denote by $\chi^F \in \{0,1\}^E$ the incidence vector of F.

[7] Generally, one can even guarantee that minimum s-t cut sizes are preserved for all $s, t \in V \setminus \{v\}$. Feasibility for the PCTSP LP relaxation is already maintained by preserving minimum r-u cut sizes for all $u \in V \setminus \{r, v\}$.

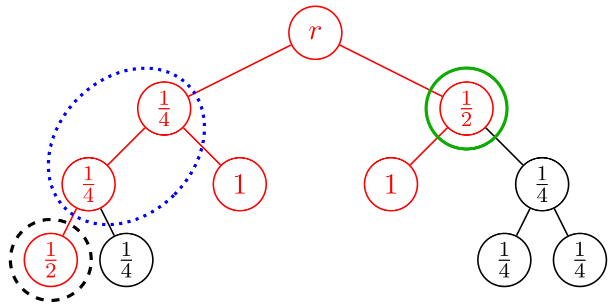

Figure 1: The core $T' = \mathrm{core}(T, 1/2)$ of the underlying tree T at threshold $\gamma = 1/2$ is highlighted in red, where y_v is shown for each node $v \in V[T]$. We emphasize the different situations that can occur in terms of cuts: For the dotted blue cut S, $x(\delta(S))$ may be small as it does not contain a vertex with $y_v \geq 1/2$, however there are at least three edges of T' in $\delta(S)$ that make up for this. The dashed black cut is a cut S with only one tree edge in $\delta(S)$, however it contains a vertex with $y_v \geq 1/2$ so $x(\delta(S))$ is large. Finally, for the solid green cut, $x(\delta(S))$ may be small and there are only two tree edges in $\delta(S)$, however for this cut, $\delta_{T'}(S)$ is even, so there is no corresponding constraint in the dominant of the $\mathrm{odd}(T')$-join polytope.

Theorem 3 (Splitting off)**.** *Let (x^*, y^*) be a feasible solution of the PCTSP LP relaxation. Let $v \in V \setminus \{r\}$. There is a deterministic algorithm that computes in polynomial time a complete splitting at v, i.e., a sequence of feasible splittings at v resulting in a feasible solution (x, y) of the PCTSP LP relaxation with $y_v = 0$, as well as*

$$c^\top x \leq c^\top x^* \qquad \text{and} \qquad \forall u \in V \setminus \{v\}\colon\ y_u = y_u^* \ .$$

Our first preprocessing step then consists of repeatedly applying Theorem 3 to vertices $v \in V \setminus \{r\}$ with $y_v < \delta$ for some parameter $\delta \in [0, 1)$ that we fix later.

Our second preprocessing step affects the trees that we obtain through Lemma 2. While the first preprocessing step guarantees that there are no non-trivial cuts S with very small $x(\delta(S))$, we also want to eliminate cuts S with moderately small $x(\delta(S))$ for which T has only a single edge in $\delta(S)$, so that the combination z defined in (2) gets a significant contribution from at least one of x or $\chi^{E[T]}$ on every non-trivial cut. To this end, we use a pruning step defined as follows (also see Figure 1).

Definition 4 (Core)**.** *For a fixed solution (x, y) of the PCTSP LP relaxation, a tree T containing the root vertex, and a threshold γ, the* core *of T with respect to γ, denoted by $\mathrm{core}(T, \gamma)$, is the inclusion-wise minimal subtree of T that spans all vertices $v \in V[T]$ with $y_v \geq \gamma$.*

Indeed, if $T' = \mathrm{core}(T, \gamma)$, we know that for any non-empty cut $S \subsetneq V[T']$, we either have $x(\delta(S)) \geq 2\gamma$, or $|\delta_T(S)| > 1$. Additionally, the only relevant such cuts S in terms of parity correction on T' are those with $|S \cap \mathrm{odd}(T')|$ odd, which is well-known to be equivalent to $|\delta_{T'}(S)|$ being odd because

$$|S \cap \mathrm{odd}(T')| \equiv \sum_{v \in S} \deg_{T'}(v) = 2 \cdot \left| E[T'] \cap \binom{S}{2} \right| + |\delta_{T'}(S)| \equiv |\delta_{T'}(S)| \pmod 2 \ . \tag{3}$$

Thus, for cuts S with $|S \cap \mathrm{odd}(T')|$ odd, $|\delta_{T'}(S)| > 1$ implies $|\delta_{T'}(S)| \geq 3$, thereby further boosting the load on $\delta(S)$ in z for this case (also see Figure 1 for examples of the different types of cuts that may appear). Altogether, we are now ready to state our new algorithm for PCTSP, Algorithm 1.

We remark that Algorithm 1 can be implemented to run in polynomial time. Indeed, by Lemma 2 and Theorem 3, Steps 2 to 4 run in polynomial time. Moreover, we can compute an optimum solution to the PCTSP LP relaxation in polynomial time as the seperation problem can be solved by computing minimum r-v cut sizes for each $v \in V \setminus \{r\}$.

We show in the next section that there is a constant δ for which Algorithm 1 is a $(1+\sqrt{5})/2$-approximation algorithm. To go beyond that and prove Theorem 1, we will later allow an instance-specific choice of δ.

Algorithm 1: Our new approximation algorithm for PCTSP

Input: *PCTSP instance* (G, r, c, π) *on* $G = (V, E)$, $\delta \in [0, 1)$.

1. Compute an optimal solution (x^*, y^*) of the PCTSP LP relaxation.
2. Perform complete splittings at all $v \in V$ with $y_v < \delta$ (see Theorem 3), resulting in a feasible solution (x, y) of the PCTSP LP relaxation.
3. Compute a set $\mathcal{T}$ of trees through Lemma 2 applied to (x, y).
4. Let
$$\mathcal{T}' = \bigcup_{\gamma \in \{y_v : v \in V\}} \{\mathrm{core}(T, \gamma) : T \in \mathcal{T}\} \enspace .$$

return *Best tour found by doing parity correction on all trees in* $\mathcal{T}'$.

3 A $(1+\sqrt{5})/2$-approximation guarantee

In this section, we prove the following result that gives the golden ratio as approximation guarantee. This is slightly weaker than what Theorem 1 claims, but the proof is simple and illustrates our main ideas.

Theorem 5. *Algorithm 1 is an* α*-approximation algorithm for PCTSP with*
$$\alpha := \max\left\{\frac{5-2\delta}{3-\delta}, \frac{3-\delta}{2-\delta}, \frac{1}{1-\delta}\right\} \enspace .$$
In particular, for $\delta = {3-\sqrt{5}}/{2} \approx 0.382$*, we get* $\alpha = {(1+\sqrt{5})}/{2} \approx 1.618$.

Throughout this section, we fix a solution (x, y) of the PCTSP LP relaxation that was obtained from an optimal solution (x^*, y^*) through complete splittings as in Step 2 of Algorithm 1, and we fix a set $\mathcal{T}$ of trees with weights $(\mu_T)_{T \in \mathcal{T}}$ that is obtained in Step 3, i.e., through Lemma 2 applied to (x, y). Moreover, we sample a random tree T from the set $\mathcal{T}'$ constructed in Step 4 of Algorithm 1 as follows: For a fixed value $\kappa \in [\delta, 1]$, sample a threshold $\gamma \in [\delta, \kappa]$ such that for any $t \in [\delta, \kappa]$, we have
$$\mathbb{P}[\gamma \leq t] = \frac{3-\delta-\kappa}{3-\delta-t} \enspace . \tag{4}$$
Independently, sample a tree $T_0 \in \mathcal{T}$ with marginals given by $(\mu_T)_{T \in \mathcal{T}}$. Then, define
$$T := \mathrm{core}(T_0, \gamma) \enspace . \tag{5}$$
By definition of the core, it is clear that $T \in \mathcal{T}'$ even if $\gamma \notin \{y_v : v \in V\}$. To prove Theorem 5, we bound the expected cost of a tour constructed from T by parity correction. We remark that for proving Theorem 5, we only need $\kappa = 1$; we nonetheless proceed in this generality here to be able to reuse some of the following statements in a proof of Theorem 1. To start with, we bound the expected tour length.

Lemma 6. *Let* $T = \mathrm{core}(T_0, \gamma)$ *be a random tree generated as described in and above (5), and let* C *be the cycle obtained through parity correction on* T *and shortcutting an Eulerian walk in the resulting graph. Then*
$$\mathbb{E}[c(E[C])] \leq \frac{7-2\delta-2\kappa}{3-\delta} \cdot c^\top x^* \enspace .$$

Proof. Let $\eta_1 > \ldots > \eta_k$ such that $\{\eta_1, \ldots, \eta_k\} = \{y_v : v \in V\}$. Define
$$E_i := \begin{cases} E[\mathrm{core}(T_0, 1)] & \text{for } i = 1 \\ E[\mathrm{core}(T_0, \eta_i)] \setminus E[\mathrm{core}(T_0, \eta_{i-1})] & \text{for } i \in \{2, \ldots, k\} \end{cases} \enspace .$$

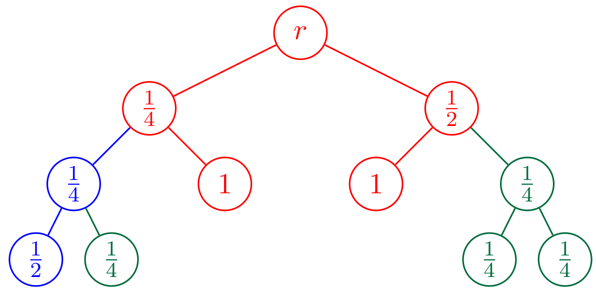

Figure 2: Here, the variables η_i are given by the sequence $1, {}^1\!/_2, {}^1\!/_4$. Edges in E_1 are drawn in red, those in E_2 in blue, and those in E_3 in green. We also color the nodes for intuition, however the sets E_i consist solely of edges.

We refer to Figure 2 for an illustration of the sets E_i. In particular, this definition implies that

$$c(E[T]) = c\big(E[\mathrm{core}(T_0, \gamma)]\big) = \sum_{i \in [k]:\, \eta_i \geq \gamma} c(E_i) \ . \tag{6}$$

To bound the cost of parity correction on T, we claim that

$$z := \frac{1}{3-\delta} \cdot x + \sum_{i \in [k]:\, \eta_i \geq \gamma} \left(1 - \frac{2\eta_i}{3-\delta}\right) \cdot \chi^{E_i}$$

lies in the dominant of the $\mathrm{odd}(T)$-join polytope. This implies that a shortest $\mathrm{odd}(T)$-join J has length

$$c(E[J])) \leq \frac{1}{3-\delta} \cdot c^\top x + \sum_{i \in [k]:\, \eta_i \geq \gamma} \left(1 - \frac{2\eta_i}{3-\delta}\right) \cdot c(E_i) \ .$$

Combining this with (6) and taking expectations immediately gives the desired bound:

$$\begin{aligned}
\mathbb{E}[c(E[C])] &\leq \frac{1}{3-\delta} \cdot c^\top x + \sum_{T_0 \in \mathcal{T}} \mu_{T_0} \sum_{i=1}^{k} \mathbb{P}[\gamma \leq \eta_i] \cdot \left(2 - \frac{2\eta_i}{3-\delta}\right) \cdot c(E_i) \\
&\leq \frac{1}{3-\delta} \cdot c^\top x + \sum_{T_0 \in \mathcal{T}} \mu_{T_0} \sum_{i=1}^{k} \frac{6 - 2\delta - 2\kappa}{3-\delta} \cdot c(E_i) && \text{(using (4))} \\
&= \frac{1}{3-\delta} \cdot c^\top x + \frac{6 - 2\delta - 2\kappa}{3-\delta} \cdot \sum_{T_0 \in \mathcal{T}} \mu_{T_0} c(E[T_0]) \\
&\leq \frac{7 - 2\delta - 2\kappa}{3-\delta} \cdot c^\top x \ . && \text{(using Lemma 2)}
\end{aligned}$$

By the construction of (x, y) from (x^*, y^*) through splitting off, we have $c^\top x \leq c^\top x^*$, hence the above implies the desired. It thus remains to show that $z \in P^{\uparrow}_{\text{odd}(T)\text{-join}}$, i.e., that $z(\delta(S)) \geq 1$ for every $S \subseteq V$ with $|S \cap \mathrm{odd}(T)|$ odd. As remarked in (3), $|S \cap \mathrm{odd}(T)|$ is odd if and only if $|\delta_T(S)|$ is. If $|\delta_T(S)| \geq 3$,

$$z(\delta(S)) = \frac{1}{3-\delta} \cdot x(\delta(S)) + \sum_{i \in [k]:\, \eta_i \geq \gamma} \left(1 - \frac{2\eta_i}{3-\delta}\right) \cdot |E_i \cap \delta_T(S)| \geq \frac{2\delta}{3-\delta} + \left(1 - \frac{2}{3-\delta}\right) \cdot 3 = 1 \ ,$$

where we used that $x(\delta(S)) \geq 2\delta$ because $y_v \geq \delta$ for all $v \in S$, and that $\eta_i \leq 1$ for each $i \in [k]$. Otherwise, we have $|\delta_T(S)| = 1$. Let $i \in [k]$ such that $\delta_T(S) \subseteq E_i$. Then $x(\delta(S)) \geq 2\eta_i$, hence also in this case

$$z(\delta(S)) \geq \frac{2\eta_i}{3-\delta} + \left(1 - \frac{2\eta_i}{3-\delta}\right) = 1 \ . \qquad \square$$

Next, we analyze the expected penalty incurred when starting with such a random tree T.

Lemma 7. *Let $T = \mathrm{core}(T_0, \gamma)$ be a random tree generated as described in and above (5). Then, for every $v \in V$, we have*

$$\mathbb{P}[v \in V[T]] \geq \begin{cases} 0 & \text{if } y_v^* \in [0, \delta) \\ y_v^* \cdot \frac{3-\delta-\kappa}{3-\delta-y_v^*} & \text{if } y_v^* \in [\delta, \kappa] \\ y_v^* & \text{if } y_v^* \in (\kappa, 1] \end{cases} .$$

Proof. By construction, the solution (x, y) has the property that for all $v \in V$, either $y_v = 0$ or $y_v = y_v^* \geq \delta$. Consequently, no tree T_0 in the family $\mathcal{T}$ generated through Lemma 2 contains vertices $v \in V$ with $y_v^* < \delta$, and thus the same holds for T. Consequently, for such vertices $v \in V$, we get $\mathbb{P}[v \in V[T]] = 0$. For vertices $v \in V$ with $y_v = y_v^* \geq \delta$, we have $\mathbb{P}[v \in V[T_0]] = y_v^*$ by Lemma 2. Hence,

$$\mathbb{P}[v \in V[T]] \geq \mathbb{P}[v \in V[T_0]] \cdot \mathbb{P}[\gamma \leq y_v^*] = y_v^* \cdot \mathbb{P}[\gamma \leq y_v^*] \ .$$

If $y_v^* > \kappa$, then $\mathbb{P}[\gamma \leq y_v^*] = 1$ and $\mathbb{P}[v \in V[T]] = y_v^*$. In the remaining case $y_v^* \in [\delta, \kappa]$, we use (4) to obtain the desired. ☐

Together, Lemmas 6 and 7 allow us to conclude Theorem 5.

Proof of Theorem 5. Let T be a random tree generated as described in and above (5), and let C be the cycle obtained through parity correction on T and shortcutting an Eulerian walk in the resulting graph.

Let $v \in V$. By Lemma 7, if $y_v^* < \delta$, then $\mathbb{P}[v \notin V[C]] = 1 \leq \frac{1}{1-\delta}(1 - y_v^*)$; if $y_v^* > \kappa$, then $\mathbb{P}[v \notin V[C]] \leq 1 - y_v^*$. If $\delta \leq y_v^* \leq \kappa$ then, again by Lemma 7,

$$\begin{aligned} \mathbb{P}[v \notin V[C]] &\leq 1 - y_v^* \cdot \frac{3-\delta-\kappa}{3-\delta-y_v^*} \\ &= \frac{(3-\delta)(1-y_v^*) - y_v^*(1-\kappa)}{3-\delta-y_v^*} \\ &\leq \frac{3-\delta}{3-\delta-\kappa} \cdot (1-y_v^*) \ . && \text{(because } y_v^* \leq \kappa \leq 1) \end{aligned}$$

Hence, together with Lemma 6 we get

$$\begin{aligned} \mathbb{E}[c(E[C]) + \pi(V \setminus V[C])] &\leq \frac{7-2\delta-2\kappa}{3-\delta} \cdot c^\top x^* + \max\left\{\frac{3-\delta}{3-\delta-\kappa}, \frac{1}{1-\delta}\right\} \cdot \pi^\top(1-y^*) \\ &\leq \max\left\{\frac{7-2\delta-2\kappa}{3-\delta}, \frac{3-\delta}{3-\delta-\kappa}, \frac{1}{1-\delta}\right\} \cdot \left(c^\top x^* + \pi^\top(1-y^*)\right) \ . \end{aligned} \tag{7}$$

Independently of the realization of the involved random variables, the cycle C is one that is generated in Algorithm 1. The maximum in (7) is minimized for $\kappa = 1$ and $\delta = (3-\sqrt{5})/2$, where it evaluates to $(1+\sqrt{5})/2$, thus giving the guarantee claimed in Theorem 5. ☐

4 Getting below 1.6

To improve upon the golden ratio approximation guarantee that we proved in Section 3, we exploit some remaining flexibility in the proof: Sampling not only the threshold γ but also δ from a distribution allows for balancing off costs better than before. The choice of distribution here is not best possible (though close to best possible, see Remark 8), but designed to demonstrate that the true approximability of PCTSP is below 1.6 in a way that reduces the use of computers to the evaluation of a "simple" function that does not involve integrals. To still obtain a deterministic algorithm, we also show how actually sampling δ can be avoided by trying polynomially many instance-dependent values.

Proof of Theorem 1. For constants κ_0 and κ to be fixed later, we sample $\delta \in [\kappa_0, \kappa]$ from a distribution with density $f(\delta) = \nu \cdot (3-\delta)(\kappa-\delta)^{2.2}$, where

$$\nu = \left(\int_{\kappa_0}^{\kappa} (3-\delta)(\kappa-\delta)^{2.2} \,\mathrm{d}\delta \right)^{-1} = \left(\frac{(3-\kappa)(\kappa-\kappa_0)^{3.2}}{3.2} + \frac{(\kappa-\kappa_0)^{4.2}}{4.2} \right)^{-1} \ .$$

Using this δ, sample a tree T as described in and above (5) (using the same κ as here), and let C be the cycle generated from parity correction on T and shortcutting an Eulerian walk in the resulting graph. Using Lemma 6, we get that the expected length of the cycle is

$$\begin{aligned}\mathbb{E}[c(E[C])] &\leq \nu \cdot \int_{\kappa_0}^{\kappa} (7-2\delta-2\kappa)(\kappa-\delta)^{2.2} \,\mathrm{d}\delta \cdot c^\top x^* \\ &= \underbrace{\nu \cdot \left(\frac{(7-4\kappa)(\kappa-\kappa_0)^{3.2}}{3.2} + \frac{2(\kappa-\kappa_0)^{4.2}}{4.2} \right)}_{=:g(\kappa,\kappa_0)} \cdot c^\top x^* \ .\end{aligned}$$

Next, we bound the expected penalty. Let $v \in V$. By Lemma 7, if $y_v^* < \kappa_0$, then $\mathbb{P}[v \notin V[C]] = 1 \leq \frac{1}{1-\kappa_0} \cdot (1-y_v^*)$; if $y_v^* > \kappa$, then $\mathbb{P}[v \notin V[C]] \leq 1 - y_v^*$. For $y_v^* \in [\kappa_0, \kappa]$, we again use Lemma 7 and obtain

$$\mathbb{P}[v \notin V[C]] \leq 1 - y_v^* \cdot \nu \cdot \int_{\kappa_0}^{y_v^*} \frac{3-\delta-\kappa}{3-\delta-y_v^*} (3-\delta)(\kappa-\delta)^{2.2} \,\mathrm{d}\delta \ .$$

Now observe that for $y_v^* \leq \kappa$ the function $\delta \mapsto \phi(\delta) := \frac{(3-\delta-\kappa)(3-\delta)}{3-\delta-y_v^*} = 3-\delta-\kappa+y_v^* - \frac{y_v^*(\kappa-y_v^*)}{3-\delta-y_v^*}$ is concave on $[\kappa_0, \kappa]$, hence for each $\delta \in [\kappa_0, \kappa]$, we have

$$\phi(\delta) \geq \phi(\kappa_0) \cdot \frac{\kappa-\delta}{\kappa-\kappa_0} + \phi(\kappa) \cdot \frac{\delta-\kappa_0}{\kappa-\kappa_0} \ .$$

Plugging in this bound and evaluating the involved integrals gives

$$\begin{aligned}\mathbb{P}[v \notin V[C]] &\leq 1 - \frac{y_v^* \cdot \nu}{\kappa-\kappa_0} \cdot \left(\phi(\kappa_0) \cdot \int_{\kappa_0}^{y_v^*} (\kappa-\delta)^{3.2} \,\mathrm{d}\delta + \phi(\kappa) \cdot \int_{\kappa_0}^{y_v^*} (\delta-\kappa_0)(\kappa-\delta)^{2.2} \,\mathrm{d}\delta \right) \\ &= 1 - \frac{y_v^* \cdot \nu}{\kappa-\kappa_0} \cdot \Bigg((\phi(\kappa_0)-\phi(\kappa)) \cdot \frac{(\kappa-\kappa_0)^{4.2} - (\kappa-y_v^*)^{4.2}}{4.2} \\ &\qquad + \phi(\kappa) \cdot (\kappa-\kappa_0) \cdot \frac{(\kappa-\kappa_0)^{3.2} - (\kappa-y_v^*)^{3.2}}{3.2} \Bigg) =: h_{y_v^*}(\kappa, \kappa_0) \ .\end{aligned}$$

Thus

$$\mathbb{E}[\pi(V \setminus V[C])] \leq \underbrace{\max\left\{ \frac{1}{1-\kappa_0}, \max_{y \in [\kappa_0, \kappa]} \frac{h_y(\kappa,\kappa_0)}{1-y} \right\}}_{=:h(\kappa,\kappa_0)} \cdot \pi^\top (1-y^*) \ , \tag{8}$$

and we get a bound on the expected total cost of the form

$$\mathbb{E}[c(E[C]) + \pi(V \setminus V[C])] \leq \max\{g(\kappa,\kappa_0), h(\kappa,\kappa_0)\} \cdot \left(c^\top x^* + \pi^\top (1-y^*) \right) \ .$$

The latter maximum evaluates to slightly below 1.599 for $\kappa_0 = 0.3724$ and $\kappa = 0.9971$, thus giving the desired guarantee. To calculate the maximum in (8), we use that the derivative of $y \mapsto \frac{h_y(\kappa,\kappa_0)}{1-y}$ on the interval $[\kappa_0, \kappa]$ can be bounded by a constant. Hence, $\max_{y\in[\kappa_0,\kappa]} \frac{h_y(\kappa,\kappa_0)}{1-y}$ can be approximated up to a minor error by evaluating the function for each y in a sufficiently fine discretization of the interval $[\kappa_0, \kappa]$.

Finally, note that the choice of δ can be derandomized by only trying the instance-specific values in the set $\{y_v^* \colon v \in V\}$ obtained from the optimal LP solution (x^*, y^*) that is used. Indeed, if $\delta \notin \{y_v^* \colon v \in V\}$, the bound on the expected cost of the cycle given in Lemma 6 improves (as long as $\kappa \geq 1/2$) by using the minimal δ' in $\{y_v^* \colon v \in V, y_v^* \geq \delta\}$, whereas the bound on the expected penalty cost does not change. $\square$

Remark 8. *Computational experiments based on discretizing a distribution over pairs* (δ, κ) *suggest that an analysis following the one in the above proof cannot achieve an approximation ratio of* 1.59*. We emphasize that this does not exclude that the actual approximation guarantee of Algorithm 1 is below* 1.59*.*

5 Proof of Lemma 2

For the sake of this paper being self-contained, we provide a proof of Lemma 2. We do very closely follow the proof in [BN23], but incorporate some minor simplifications stemming from the fact that [BN23] show a generalized version of Lemma 2.

Proof of Lemma 2. We prove the statement by induction on $|V|$. If $V = \{r, v\}$, then a feasible decomposition is given by one tree T spanning V with weight μ_T equal to $x_{\{r,v\}}/2$, and one tree containing only the root with weight $1 - \mu_T$. If $|V| > 2$, consider a vertex $s \in V \setminus \{r\}$ that minimizes y_s. By Theorem 3, we can efficiently obtain a complete splitting at s, i.e., a sequence of splitting operations that result in a solution $(x', y|_{V'})$ of the PCTSP LP relaxation over $V' := V \setminus \{s\}$. Consequently, by the inductive assumption, we can in polynomial time compute a set $\mathcal{T}$ of trees with the desired properties; in particular, for every $v \in V'$,

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

We now undo the splitting operations at s one after another and modify the trees in $\mathcal{T}$ accordingly. Before we start undoing the splitting operations, we initialize auxiliary variables $\text{spare}_v = 0$ for each $v \in V'$.

Consider a splitting operation on edges $e = \{s, u\}$ and $f = \{s, w\}$ with $e \neq f$ that reduces the weight on each of these edges by δ. Let $\mathcal{T}' := \{T \in \mathcal{T} \colon \{u, w\} \in E[T]\}$. If $\sum_{T\in\mathcal{T}'} \mu_T > \delta$, remove trees from $\mathcal{T}'$ until $\sum_{T\in\mathcal{T}'} \mu_T = \delta$ (this may require creating a copy of some tree and splitting its weight); if $\sum_{T\in\mathcal{T}'} \mu_T < \delta$, then add $\delta - \sum_{T\in\mathcal{T}'} \mu_T$ to spare_u. Now for each $T \in \mathcal{T}'$ do the following:

(i) If $s \notin V[T]$, remove $\{u, w\}$ from $E[T]$ and add $\{s, u\}$ and $\{s, w\}$ to $E[T]$.

(ii) If $s \in V[T]$, remove $\{u, w\}$ from $E[T]$ and add either $\{s, u\}$ or $\{s, w\}$ to $E[T]$ such that T remains acyclic. If $\{s, u\}$ is added to $E[T]$ increase spare_w by μ_T. Otherwise, increase spare_u by μ_T.

Note that in case (i), the total weight of trees containing s increases by μ_T. Otherwise, either spare_u or spare_w increases by μ_T. Consequently, through the above operations (including the potential initial increase of spare_u), the sum $\sum_{T\in\mathcal{T}\colon s\in V[T]} \mu_T + \sum_{v\in V'} \text{spare}_v$ increases by δ. Since the splitting operation decreased the degree of s by 2δ, we get, after reverting all splitting operations at s,

$$\sum_{T\in\mathcal{T}\colon s\in V[T]} \mu_T + \sum_{v\in V'} \text{spare}_v = \frac{x(\delta(s))}{2} = y_s \enspace .$$

Now, for each $w \in V'$ do the following: If $\text{spare}_w > 0$, let $\mathcal{T}'' = \{T \in \mathcal{T} \colon w \in V[T], s \notin V[T]\}$. Note that

$$\sum_{T\in\mathcal{T}\colon w\in V[T]} \mu_T = y_w \geq y_s = \sum_{T\in\mathcal{T}\colon s\in V[T]} \mu_T + \sum_{v\in V'} \text{spare}_v \enspace ,$$

where the inequality above holds because s minimizes y_v over all $v \in V \setminus \{r\}$, and the first equality holds by the inductive assumption and the fact that for every vertex in V' the total weight of trees covering this vertex is unchanged under splittings and their reversal.

In particular, the above implies $\sum_{T\in\mathcal{T}''}\mu_T \geq \mathrm{spare}_w$. If this inequality is strict, remove trees from $\mathcal{T}''$ until $\sum_{T\in\mathcal{T}''}\mu_T = \mathrm{spare}_w$ (again, this may require creating a copy of some tree and splitting its weight). Now, for each $T \in \mathcal{T}''$, add $\{s, w\}$ to $E[T]$. Note that this increases the total weight of the trees containing s by spare_w. Hence, after using up all spares, we get

$$\sum_{T\in\mathcal{T}\colon\, s\in V[T]} \mu_T = y_s \quad . \tag{9}$$

Note that throughout the above operations, the graphs $T \in \mathcal{T}$ are trees, and after reverting all splitting operations, $\sum_{T\in\mathcal{T}}\mu_T\chi^{E[T]} \leq x$ by construction. Moreover, as mentioned above, for every vertex in V', the total weight of trees covering this vertex is unchanged. Hence, $\sum_{T\in\mathcal{T}\colon\, v\in V[T]}\mu_T = y_v$ is still satisfied for each $v \in V'$, and also for s by (9).

Finally, we note that our construction can be executed in polynomial time. Indeed, by Theorem 3, in each step of our inductive procedure, we have to revert less than $\mathrm{poly}(|V|)$ many splitting operations, which increases the total number of trees in $\mathcal{T}$ by an additive $\mathrm{poly}(|V|)$. This implies that the size of $\mathcal{T}$ remains polynomially bounded throughout. □

References

[ABHK11] A. Archer, M. Bateni, M. Hajiaghayi, and H. Karloff. "Improved Approximation Algorithms for Prize-Collecting Steiner Tree and TSP". In: *SIAM Journal on Computing* 40.2 (2011), pp. 309–332. DOI: 10.1137/090771429 (cit. on pp. 1–3).

[ABLM07] G. Ausiello, V. Bonifaci, S. Leonardi, and A. Marchetti-Spaccamela. "Prize-Collecting Traveling Salesman and Related Problems". In: *Handbook of Approximation Algorithms and Metaheuristics*. Chapman and Hall, 2007. Chap. 40, pp. 1–13. DOI: 10.1201/9781420010749 (cit. on p. 3).

[AGHJM24] A. Ahmadi, I. Gholami, M. Hajiaghayi, P. Jabbarzade, and M. Mahdavi. "2-Approximation for Prize-Collecting Steiner Forest". In: *Proceedings of the 35th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '24)*. 2024, pp. 669–693. DOI: 10.1137/1.9781611977912 (cit. on p. 3).

[ALM00] G. Ausiello, S. Leonardi, and A. Marchetti-Spaccamela. "On Salesmen, Repairmen, Spiders, and Other Traveling Agents". In: *Proceedings of the 4th Italian Conference on Algorithms and Complexity*. 2000, pp. 1–16. DOI: 10.1007/3-540-46521-9_1 (cit. on p. 3).

[Bal89] E. Balas. "The prize collecting traveling salesman problem". In: *Networks* 19.6 (1989), pp. 621–636. DOI: 10.1002/net.3230190602 (cit. on pp. 2, 3).

[BCEHKM11] M. Bateni, C. Chekuri, A. Ene, M. Hajiaghayi, N. Korula, and D. Marx. "Prize-collecting Steiner Problems on Planar Graphs". In: *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '11)*. 2011, pp. 1028–1049. DOI: 10.1137/1.9781611973082.79 (cit. on p. 3).

[BCKLMM07] A. Blum, S. Chawla, D. Karger, T. Lane, A. Meyerson, and M. Minkoff. "Approximation algorithms for orienteering and discounted-reward TSP". In: *SIAM Journal on Computing* 37.2 (2007), pp. 653–670. DOI: 10.1137/050645464 (cit. on p. 3).

[BFJ95] J. Bang-Jensen, A. Frank, and B. Jackson. "Preserving and Increasing Local Edge-Connectivity in Mixed Graphs". In: *SIAM Journal on Discrete Mathematics* 8.2 (1995), pp. 155–178. DOI: 10.1137/S0036142993226983 (cit. on pp. 1, 4).

[BGSW93] D. Bienstock, M. X. Goemans, D. Simchi-Levi, and D. Williamson. "A note on the prize collecting traveling salesman problem". In: *Mathematical Programming* 59 (1993), pp. 413–420. DOI: 10.1007/BF01581256 (cit. on pp. 1, 2).

[BN23] J. Blauth and M. Nägele. "An improved approximation guarantee for Prize-Collecting TSP". In: *Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC '23)*. 2023, pp. 1848–1861. DOI: 10.1145/3564246.3585159 (cit. on pp. 1–4, 10).

[Chr22] N. Christofides. "Worst-Case Analysis of a New Heuristic for the Travelling Salesman Problem". In: *Operations Research Forum* 3 (2022). DOI: 10.1007/s43069-021-00101-z (cit. on p. 2).

[Chr76] 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 (cit. on pp. 2, 4).

[CJJ20] T.-H. H. Chan, H. Jiang, and S. H.-C. Jiang. "A Unified PTAS for Prize Collecting TSP and Steiner Tree Problem in Doubling Metrics". In: *ACM Transactions on Algorithms* 16.2 (2020), pp. 1–23. DOI: 10.1145/3378571 (cit. on p. 3).

[CKP12] C. Chekuri, N. Korula, and M. Pál. "Improved Algorithms for Orienteering and Related Problems". In: *ACM Transactions on Algorithms* 8.3 (2012), pp. 1–27. DOI: 10.1145/2229163.2229167 (cit. on p. 3).

[DFPS22] S. Dezfuli, Z. Friggstad, I. Post, and C. Swamy. "Combinatorial Algorithms for Rooted Prize-Collecting Walks and Applications to Orienteering and Minimum-Latency Problems". In: *Proceedings of the 23rd Conference on Integer Programming and Combinatorial Optimization (IPCO '22)*. 2022, pp. 195–209. DOI: 10.1007/978-3-031-06901-7_15 (cit. on p. 3).

[EJ73] J. Edmonds and E. Johnson. "Matching, Euler tours and the Chinese postman". In: *Mathematical Programming* 5 (1973), pp. 88–124. DOI: 10.1007/BF01580113 (cit. on p. 3).

[Fra92] A. Frank. "On a theorem of Mader". In: *Discrete Mathematics* 101.1 (1992), pp. 49–57. DOI: 10.1016/0012-365X(92)90589-8 (cit. on p. 4).

[Gar05] N. Garg. "Saving an Epsilon: A 2-Approximation for the k-MST Problem in Graphs". In: *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC '05)*. 2005, pp. 396–402. DOI: 10.1145/1060590.1060650 (cit. on p. 3).

[Goe09] M. X. Goemans. *Combining Approximation Algorithms for the Prize-Collecting TSP*. 2009. arXiv: 0910.0553 [cs.DS] (cit. on pp. 1–3).

[GW95] M. X. Goemans and D. P. Williamson. "A General Approximation Technique for Constrained Forest Problems". In: *SIAM Journal on Computing* 24.2 (1995), pp. 296–317. DOI: 10.1137/S0097539793242618 (cit. on pp. 1, 2).

[HJ06] M. T. Hajiaghayi and K. Jain. "The Prize-Collecting Generalized Steiner Tree Problem via a New Approach of Primal-Dual Schema". In: *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm (SODA '06)*. 2006, pp. 631–640. DOI: 10.1145/1109557.1109626 (cit. on p. 3).

[KKO21] A. R. Karlin, N. Klein, and S. 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)*. 2021, pp. 32–45. DOI: 10.1145/3406325.3451009 (cit. on p. 2).

[KKO23] A. R. Karlin, N. Klein, and S. 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)*. 2023, pp. 261–274. DOI: 10.1007/978-3-031-32726-1_19 (cit. on p. 2).

[KLS15] M. Karpinski, M. Lampis, and R. Schmied. “New inapproximability bounds for TSP”. In: *Journal of Computer and System Sciences* 81.8 (2015), pp. 1665–1677. DOI: 10.1016/j.jcss.2015.06.003 (cit. on p. 1).

[KOPRSV17] J. Könemann, N. Olver, K. Pashkovich, R. Ravi, C. Swamy, and J. Vygen. “On the Integrality Gap of the Prize-Collecting Steiner Forest LP”. In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM ’17)*. Vol. 81. 2017, 17:1–17:13. DOI: 10.4230/LIPIcs.APPROX-RANDOM.2017.17 (cit. on p. 3).

[Lov76] L. Lovász. “On some connectivity properties of Eulerian graphs”. In: *Acta Mathematica Academiae Scientiarum Hungarica* 28.1 (1976), pp. 129–138. DOI: 10.1007/BF01902503 (cit. on p. 4).

[Mad78] W. Mader. “A Reduction Method for Edge-Connectivity in Graphs”. In: *Annals of Discrete Mathematics* 3 (1978), pp. 145–164. DOI: 10.1016/S0167-5060(08)70504-1 (cit. on p. 4).

[Nag06] H. Nagamochi. “A Fast Edge-Splitting Algorithm in Edge-Weighted Graphs”. In: *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* E89-A.5 (2006), pp. 1263–1268. DOI: 10.1093/ietfec/e89-a.5.1263 (cit. on p. 4).

[Ngu13] V. H. Nguyen. “A Primal-Dual Approximation Algorithm for the Asymmetric Prize-Collecting TSP”. In: *Journal of Combinatorial Optimization* 25 (2013), pp. 265–278. DOI: 10.1007/s10878-012-9501-z (cit. on p. 3).

[NI97] H. Nagamochi and T. Ibaraki. “Deterministic $\tilde{O}(nm)$ Time Edge-Splitting in Undirected Graphs”. In: *Journal of Combinatorial Optimization* 1.6 (1997), pp. 5–46. DOI: 10.1023/A:1009739202898 (cit. on p. 4).

[PFFSW20] A. Paul, D. Freund, A. Ferber, D. B. Shmoys, and D. P. Williamson. “Budgeted Prize-Collecting Traveling Salesman and Minimum Spanning Tree Problems”. In: *Mathematics of Operations Research* 45.2 (2020), pp. 576–590. DOI: 10.1287/moor.2019.1002 (cit. on p. 3).

[PFFSW22] A. Paul, D. Freund, A. Ferber, D. B. Shmoys, and D. P. Williamson. “Erratum to ‘Budgeted Prize-Collecting Traveling Salesman and Minimum Spanning Tree Problems’”. In: *Mathematics of Operations Research* (2022). DOI: 10.1287/moor.2022.1340 (cit. on p. 3).

[Ser87] A. I. Serdyukov. “O nekotorykh ekstremal’nykh obkhodakh v grafakh”. In: *Upravlyaemye sistemy* 17 (1987), pp. 76–79 (cit. on pp. 2, 4).

[TVZ22] V. Traub, J. Vygen, and R. Zenklusen. “Reducing path TSP to TSP”. In: *SIAM Journal on Computing* 51.3 (2022), STOC20-24–STOC20-53. DOI: 10.1137/20M135594X (cit. on p. 2).

[vS20] R. van Bevern and V. A. Slugina. “A historical note on the 3/2-approximation algorithm for the metric traveling salesman problem”. In: *Historia Mathematica* 53 (2020), pp. 118–127. DOI: 10.1016/j.hm.2020.04.003 (cit. on p. 2).