Getting Better All the Time? Harmonic and Cumulative Traveling Salesman Problems

Sándor P. Fekete[1], Dominik Krupke[1], Christian Rieck[2], Arne Schmidt[1], and Tobias Wallner[1]

1 **Department of Computer Science, TU Braunschweig, Braunschweig, Germany**
`s.fekete@tu-bs.de, {krupke, aschmidt, wallner}@ibr.cs.tu-bs.de`

2 **Department of Discrete Mathematics, University of Kassel, Kassel, Germany**
`christian.rieck@mathematik.uni-kassel.de`

Abstract

We consider versions of the Traveling Salesman Problem in which the cost of travel changes along the way: In the Harmonic TSP (HTSP), the cost of each traversed edge is its length divided by the number of previously visited vertices, while in the Cumulative TSP (CTSP), it is its length multiplied by this number. Both problems are related to the Minimum Latency Problem, which aims at minimizing the average arrival time of a tour; as we show (along with geometric properties), these problems are distinct. In addition to observations placing these variants into context, our main contribution is a constant-factor approximation for the HTSP on general metric instances.

1 Introduction

The Traveling Salesman Problem (TSP) is a classical problem of combinatorial optimization, seeking a cheapest round trip that visits each of a given set of points. In its original formulation, the cost of a TSP tour is simply the sum of edge lengths; however, in many applications from transportation and logistics, the price of traversing an edge also depends on the context within a tour: In ride-sharing or group travel, the cost per passenger may decrease as additional riders join the route and share expenses. Conversely, in some pick-up and freight services, the total load grows over the course of the route, increasing fuel consumption and handling costs. This leads to variants in which the travel cost changes along the way, either *inversely proportionally*, referred to as the Harmonic variant (HTSP), or *proportionally* (the Cumulative variant (CTSP)) to the number of previously visited vertices.

Numerous variants with travel cost that change dynamically have been studied before, including the Discounted-Reward TSP [9], the Time-dependent TSP, and the Time-dependent Vehicle Routing Problem [1, 2, 3, 11, 16, 21]. Most closely related to the variants of this paper is the well-studied Minimum Latency Problem (MLP) of minimizing the average arrival time of a Hamiltonian path that visits a given set of points. Blum et al. [8] gave a 144-approximation algorithm for metric instances. Goemans and Kleinberg [15] improved the factor to 10.78, Archer and Williamson [4] to 9.28, and Chaudhuri et al. [10] to 3.59 with run-time $\widetilde{\mathcal{O}}(n^4)$. Arora and Karakostas [6] also presented a quasi-polynomial time approximation scheme for the special case of traveling on a tree.

As part of our contribution, we observe that the MLP, the HTSP and the CTSP are distinct. Furthermore, optimal solutions for the Cumulative TSP and the Minimum Latency Problem are always within constant factors of each other, whereas this is generally not true for the Harmonic TSP and the MLP.

Formally, we define the Harmonic and Cumulative TSP variants as follows.

41st European Workshop on Computational Geometry, Liblice, Czech republic, April 9–11, 2025.
This is an extended abstract of a presentation given at EuroCG'25. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear eventually in more final form at a conference with formal proceedings and/or in a journal.

Harmonic Traveling Salesman Problem (HTSP)

Given: A connected, weighted graph $G = (V, E)$ with n vertices, a starting vertex $v_0 \in V$, and a weight function $c\colon E \to \mathbb{R}_0^+$.
Wanted: A permutation $\pi\colon \{0, ..., n-1\} \to V$, with $\pi(0) = v_0$, such that the total cost

$$C = \sum_{i=1}^{n-1} \left[\frac{\ell(\pi(i-1), \pi(i))}{i} \right] + \frac{\ell(\pi(n-1), \pi(0))}{n}$$

is minimized, with $\ell(v_i, v_j)$ denoting the weight of a shortest path from v_i to v_j.

Cumulative Traveling Salesman Problem (CTSP)

Given: A connected, weighted graph $G = (V, E)$ with n vertices, a starting vertex $v_0 \in V$, and a weight function $c\colon E \to \mathbb{R}_0^+$.
Wanted: A permutation $\pi\colon \{0, ..., n-1\} \to V$, with $\pi(0) = v_0$, such that the total cost

$$C = \sum_{i=1}^{n-1} [i \cdot \ell(\pi(i-1), \pi(i))] + n \cdot \ell(\pi(n-1), \pi(0))$$

is minimized, with $\ell(v_i, v_j)$ denoting the weight of a shortest path from v_i to v_j.

In both variants, any solution corresponds to a sequence of edges obtained by concatenating the shortest paths from $\pi(i)$ to $\pi(i+1)$ for $i = 0, ..., n-1$. When traversing an edge e of weight $c(e)$ as part of a shortest path from $\pi(k)$ to $\pi(k+1)$, we say that the vertices $\pi(0), ..., \pi(k)$ have already been *collected* at this time. Traversing e increases the cost of the current tour by an amount of $1/k \cdot c(e)$ or $k \cdot c(e)$, respectively. We refer to $1/k$ or k as the current *speed factor*, which depends on the number k of previously collected vertices.

Our Contributions We provide the following insights and results.

1. We discuss the relationship to the Minimum Latency Problem, as well as geometric properties of optimal tours, and show that both CTSP and HTSP can be solved by a simple dynamic programming approach on trees with a fixed number of leaves.
2. We establish a constant-factor approximation algorithm for the HTSP.

Throughout the remainder of this paper, we assume that the triangle inequality holds.

2 Basic Observations

Relationship to the Minimum Latency Problem Like the HTSP and the CTSP, the MLP asks for a permutation π of the vertices minimizing a cost function, the *latency* $L = \sum_{i=1}^{n-1} (n-i) \cdot \ell(v_{i-1}, v_i)$; closing the tour by returning to v_0 does not incur any further latency. Thus, any solution to the MLP can also be considered as a solution to the HTSP or the CTSP. We now evaluate the respective quality of the solutions.

The instance depicted in Figure 1 shows that no constant or logarithmic approximation factor can be guaranteed for the HTSP by using an optimal MLP solution: For both problems, only two options are reasonable: (a) first visiting all the vertices on the left, followed by those on the right, or (b) vice versa. For the HTSP, the total cost of (a) is in $\Theta(\ln k)$, while that of (b) is in $\Theta(k)$, making (a) optimal for large k. However, (b) is optimal for the MLP for any k: in particular, the total latency of (a) is $7k^2/2 + 9k/2$ whereas that of (b) is $7k^2/2 + 3k/2$.

Figure 1 An instance showing that the MLP provides no approximation for the HTSP.

An optimal MLP solution does not yield an optimal CTSP solution either, even though the CTSP cost of a permutation and the MLP latency of the reversed permutation only differ by the simple tour length: Consider the permutation $(v_0, ..., v_{n-1})$ and its inverse as solutions for the CTSP and the MLP, respectively. The cost of the CTSP solution is $C = \sum_{i=1}^{n-1} (i \cdot \ell(v_{i-1}, v_i)) + n \cdot \ell(v_{n-1}, v_0)$. The total latency of the MLP solution is $L = (n-1) \cdot \ell(v_0, v_{n-1}) + \sum_{j=1}^{n-1} (n-j-1) \cdot \ell(v_{n-j}, v_{n-j-1}) = \sum_{i=1}^{n-1} (i-1) \cdot \ell(v_{i-1}, v_i) + (n-1) \cdot \ell(v_{n-1}, v_0) = C - \left(\sum_{i=1}^{n-1} \ell(v_{i-1}, v_i) + \ell(v_{n-1}, v_0)\right)$.

Figure 2 illustrates an instance for which the optimal CTSP and the inverted MLP solutions are distinct: Out of the 24 possible permutations, v_0, v_4, v_2, v_3, v_1 with $C = 75$ is optimal for the CTSP; the optimal MLP solution is v_0, v_1, v_2, v_3, v_4 with $L = 47$.

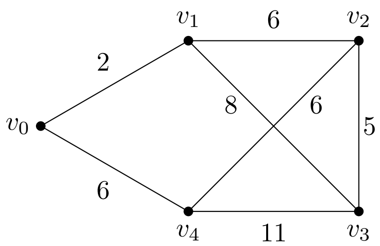

Figure 2 A simple instance showing that the MLP and the CTSP are not equivalent.

However, the inverse of an optimal solution for the MLP yields a 3-approximation for the CTSP. Based on the MLP solution, from $L = C - \left(\sum_{i=1}^{n-1} \ell(v_{i-1}, v_i) + \ell(v_{n-1}, v_0)\right)$, $\sum_{i=1}^{n-1} \ell(v_{i-1}, v_i) \leq L$, $L \leq C$, and $\ell(v_{n-1}, v_0) \leq L$, we may infer $L \leq C \leq 3L$. Assume that $(v_0, ..., v_{n-1})$ is an optimal solution for the MLP with total latency L_{OPT}. Inverting this permutation yields a CTSP solution with cost $C \leq 3 \cdot L_{\text{OPT}} \leq 3 \cdot C_{\text{OPT}}$.

Dynamic Program for Trees with Bounded Number of Leaves Paths as in Figure 1 and trees with a bounded number of leaves can be solved in polynomial time via dynamic programming. For the HTSP, consider a path $v_l, ..., v_0 (= w_0), ..., w_r$ with l vertices on the left and r on the right of the root $v_0 = w_0$. Define $\ell_m(u, u')$ as the cost of moving from u to u' after m vertices have been visited (with no additional speed factor changes), with the exact form of ℓ_m depending on the problem variant.

For $0 \leq i \leq l$ and $0 \leq j \leq r$, let $s_{\underline{i},j}$ be the minimum accumulated cost of a partial tour that has visited the first i vertices on the left and the first j vertices on the right of the root, ending at (and thus newly visiting) v_i. Define $s_{i,\underline{j}}$ analogously for tours ending at w_j. To compute $s_{\underline{i},j}$, we consider the two possible predecessors v_{i-1} and w_j of v_i:

$$s_{\underline{i},j} = \begin{cases} 0 & \text{if } i = 0 \text{ and } j = 0, \\ \perp, & \text{if } i < 0 \text{ or } j < 0, \\ \min\left\{s_{\underline{i-1},j} + \ell_{i+j-1}(v_{i-1}, v_i),\ s_{i-1,\underline{j}} + \ell_{i+j-1}(w_j, v_i)\right\}, & \text{otherwise.} \end{cases}$$

We can compute $s_{i,\underline{j}}$ analogously, yielding $\mathcal{O}(n^2)$ states in total, each requiring $\mathcal{O}(1)$ time to compute. Having visited all $l+r$ non-root vertices, we return to $v_0 = w_0$, so the total cost is

$$\min\Big\{ s_{\underline{l},r} + \ell_{l+r}(v_l, v_0), \quad s_{l,\underline{r}} + \ell_{l+r}(w_r, w_0) \Big\}.$$

A standard backtracking procedure (storing which minimum was chosen) recovers the tour in linear time. Analogous arguments are applicable to the CTSP.

More generally, for a tree with k leaves, one extends the state to k dimensions (one per leaf), each corresponding to a branch of the tree. The first i vertices collected by an optimal HTSP solution will form an interval (including the root vertex) in each of the branches. In the CTSP, after the first i vertices have been collected by an optimal solution, the vertices not yet collected will form an interval in each branch. These properties can be used to formulate dynamic programs, considering the cost of visiting an additional vertex in any of the branches, and leading to a run-time of $\mathcal{O}(k \cdot n^k)$.

Edge Crossings for Geometric Instances of the HTSP Due to the triangle inequality, optimal solutions for the EUCLIDEAN TSP never contain any crossing edges, reducing possible solutions to the set of simple polygonizations, whose worst-case cardinality is between $\Omega(4.642^n)$ [12] and $\mathcal{O}(54.543^n)$ [19, 20], and thus considerably less than the full set of $(n-1)!$ permutations. Simple connectedness is also a prerequisite for several geometric approximation techniques, such as the PTASs by Mitchell [17] and Arora [5].

However, simplicity does not hold for geometric instances of the HTSP; in fact, there may be $\Theta(n^2)$ edge crossings in an optimal solution. We demonstrate this by constructing a set of n points in a polar coordinate system as follows. We define the polar radius r_i of v_i as $r_i := 2^{i^2}$, forcing the optimal solution to visit the vertices in the order $v_0, ..., v_{n-1}, v_0$, regardless of their respective polar angles. It is straightforward to show that visiting any vertex v_j with $j > i$ before v_i results in a suboptimal total cost, as the additional cost incurred by collecting v_i first is overcompensated by the resulting speed factor improvement for the exponentially larger distance to v_j. Now that the sequence of the vertices v_i in the optimal solution has been determined based on their polar radius, we construct their polar angles φ_i, ensuring that the solution contains $\Theta(n^2)$ crossings. For $i = 1, 2, 3, 4$, we set φ_i to $\frac{3\pi}{2}$, 0, $\frac{\pi}{2}$, and π. For $i > 4$, we alternatingly set φ_i to $0 + \varepsilon_i$ and $\pi + \varepsilon_i$ with increasing $\varepsilon_i > 0$, such that every edge $v_{i-1}v_i$, each chosen in the optimal solution, crosses every edge $v_0v_1, v_1v_2, ..., v_{i-3}v_{i-2}$. An example of this construction is shown in Figure 3.

3 Constant-Factor Approximation Algorithm for the HTSP

We propose an approximation technique for the HTSP by applying existing approximation algorithms for the k-TSP and the k-MST problem; this has some resemblance to the approach used by Blum et al. [8] for the MLP, but yields a different analysis and approximation factor.

Let $\pi = (v_0, v_1, ..., v_n)$ be a permutation of V with optimal total weight, and let ℓ_i abbreviate the weight $\ell(v_{i-1}, v_i)$ of a shortest path p_i from v_{i-1} to v_i. Consider an alternative solution π' formed by altering π in the following way: After reaching the 2^j-th (new) vertex of π, $j = 0, 1, ..., \log n$, we retrace our path back to v_0. We then repeat the process, following the same sequence of vertices, until we reach the 2^{j+1}-th vertex of π. The arrows in Figure 4 illustrate this construction.

▶ **Lemma 3.1.** *The total cost of a solution π' constructed from an optimal solution π as described above is at most 6 times as high as the cost of π.*

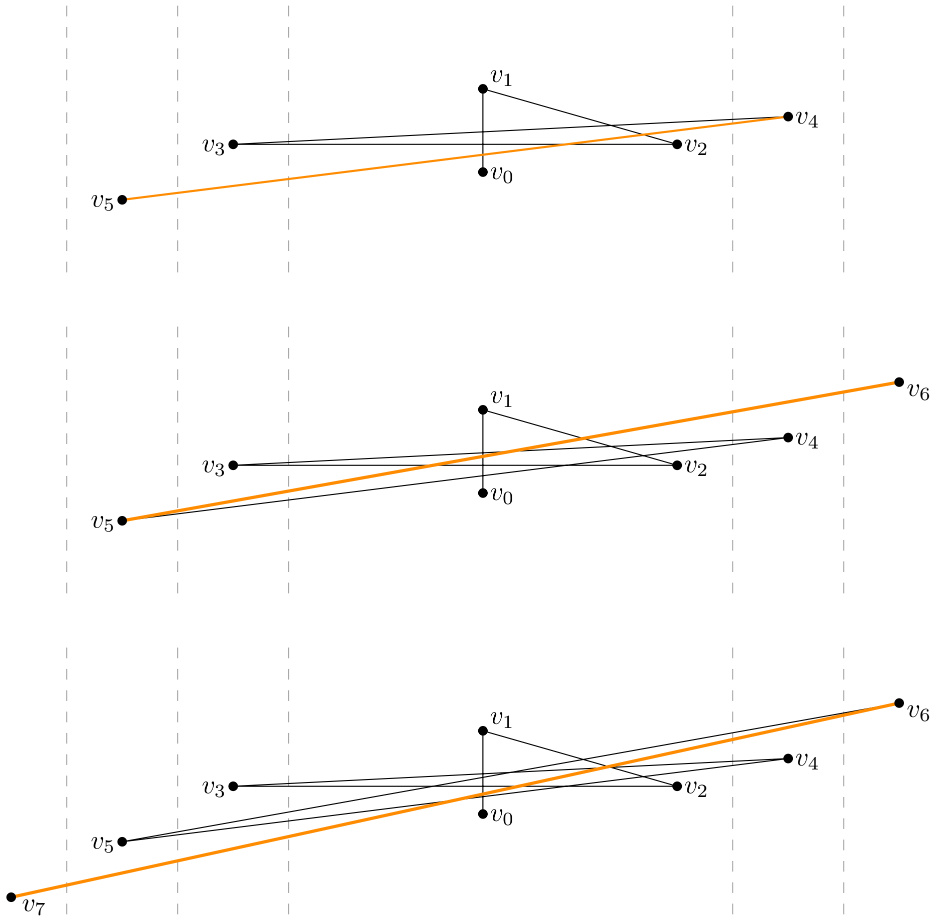

Figure 3 Instance (not to scale) with $\Theta(n^2)$ crossings in an optimal solution. The latest chosen edge is highlighted in orange, and the dashed lines represent large distances, growing like 2^{i^2}.

Proof. For each path p_i with length ℓ_i, we compare the sum of the speed factors with which p_i is traversed in π' to the speed factor of p_i in an optimal solution.

Given the decreasing speed factor, we may bound the total cost $C_{\pi'}$ of π'. Summing up the speed factors of the p_i (as illustrated in Figure 4) yields:

$$\begin{aligned}
C_{\pi'} \leq\ & \ell_1\Big(3 \cdot 1 + 2 \cdot \frac{1}{2} + 2 \cdot \frac{1}{4} + 2 \cdot \frac{1}{8} + \dots + 2 \cdot \frac{1}{2^{\lceil \log n \rceil}}\Big) \\
&+ \ell_2\Big(1 + 2 \cdot \frac{1}{2} + 2 \cdot \frac{1}{4} + 2 \cdot \frac{1}{8} + \dots + 2 \cdot \frac{1}{2^{\lceil \log n \rceil}}\Big) \\
&+ \ell_3\Big(\quad 1 \cdot \frac{1}{2} + 2 \cdot \frac{1}{4} + 2 \cdot \frac{1}{8} + \dots + 2 \cdot \frac{1}{2^{\lceil \log n \rceil}}\Big) \\
&+ \ell_4\Big(\quad 1 \cdot \frac{1}{2} + 2 \cdot \frac{1}{4} + 2 \cdot \frac{1}{8} + \dots + 2 \cdot \frac{1}{2^{\lceil \log n \rceil}}\Big) \\
&+ \ell_5\Big(\qquad 1 \cdot \frac{1}{4} + 2 \cdot \frac{1}{8} + \dots + 2 \cdot \frac{1}{2^{\lceil \log n \rceil}}\Big) \\
&+ \dots
\end{aligned}$$

Rewriting the geometric series yields

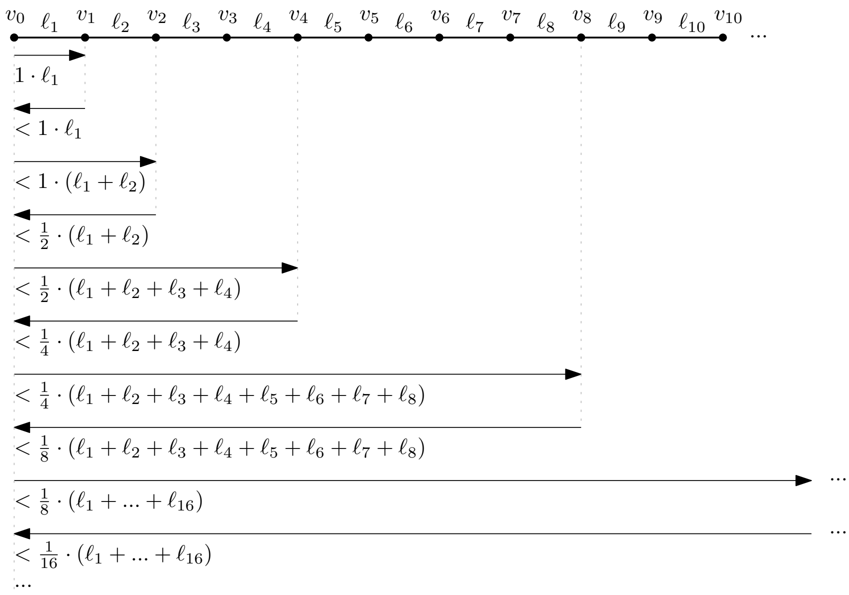

Figure 4 Illustration for the cost analysis of π'.

$$\begin{aligned}
C_{\pi'} \leq \ell_1 + \ell_1 \cdot \left(2 \sum_{k=0}^{\lceil \log n \rceil} \left(\frac{1}{2}\right)^k\right) + \ell_2 \cdot \left(-1 + 2 \sum_{k=0}^{\lceil \log n \rceil} \left(\frac{1}{2}\right)^k\right) \\
+ \ell_3 \cdot \left(-\frac{1}{2} + 2 \sum_{k=0}^{\lceil \log n \rceil} \left(\frac{1}{2}\right)^k - 2 \sum_{k=0}^{0} \left(\frac{1}{2}\right)^k\right) + \ldots + \\
+ \ell_j \cdot \left(-\frac{1}{2^{\lceil \log j \rceil - 1}} + 2 \sum_{k=0}^{\lceil \log n \rceil} \left(\frac{1}{2}\right)^k - 2 \sum_{k=0}^{\lceil \log j \rceil - 2} \left(\frac{1}{2}\right)^k\right) + \ldots
\end{aligned}$$

With the convergence of the geometric series, the coefficient of ℓ_j in this equation becomes

$$\begin{aligned}
& -\frac{1}{2^{\lceil \log j \rceil - 1}} + 2 \sum_{k=0}^{\lceil \log n \rceil} \left(\frac{1}{2}\right)^k - 2 \sum_{k=0}^{\lceil \log j \rceil - 2} \left(\frac{1}{2}\right)^k \\
& \leq -\left(\frac{1}{2}\right)^{\lceil \log j \rceil - 1} + 2 \cdot 2 - 2 \cdot \left(2 - \left(\frac{1}{2}\right)^{\lceil \log j \rceil - 2}\right) \\
& \leq -2 \left(\frac{1}{2}\right)^{\lceil \log j \rceil} + 8 \left(\frac{1}{2}\right)^{\lceil \log j \rceil} \leq 6 \left(\frac{1}{2}\right)^{\lceil \log j \rceil} \leq 6 \cdot \frac{1}{j}.
\end{aligned}$$

Thus, each path weight ℓ_j is weighted at most 6 times as much as in π (in which it is weighted with $1/j$). It is straightforward to establish this bound for the coefficients of ℓ_1 and ℓ_2. ◀

For each round k, we need to visit 2^{k-1} vertices and return to v_0 while bounding the cost with respect to the k-th round of π'. Let C_k be its total cost (accounting for speed factors) and λ_k its simple weight (only totaling edge weights). We analyze two options for finding a round-trip from v_0 via k additional vertices.

The first variant uses the related k-MST problem, asking for a tree of minimum total length spanning exactly k vertices. In the *rooted* variant, we are given a vertex v_0 that has to be included in the spanning tree. Clearly, the (unrooted) k-MST problem can be reduced to the rooted k-MST problem by going through all of the n possible roots. The problem has been proven to be NP-complete [7, 18]. Garg [14] provides a polynomial-time 2-approximation for this problem in metric graphs.

▶ **Theorem 3.2.** *There is a 24α-approximation for the HTSP in metric graphs, with α denoting an approximation factor for the rooted k-MST problem.*

Proof. Clearly, the total weight $\lambda_{\text{MST},k}$ of a v_0-rooted $(2^{k-1}+1)$-MST is not higher than λ_k. Each rooted $(2^{k-1}+1)$-MST yields a tour by traveling every edge exactly twice. Consider the solution resulting from traversing, one after the other, the v_0-rooted $(2^{k-1}+1)$-MST for $k = 1, 2, \ldots$. Then, when traversing the $(2^{k-1}+1)$-MST, at least $2^{k-2}+1$ (1 in case of the 2-MST, i.e., $k = 1$) have been collected before. Hence, the cost $C_{\text{MST},k}$ of traversing this particular tree is $C_{\text{MST},k} \leq \frac{1}{2^{k-2}+1} \cdot 2\lambda_{\text{MST},k}$. In the k-th round of π', the best speed factor is at least $\frac{1}{2^{k-1}+1}$. So, the cost $C_{\pi',k}$ of the round is $C_{\pi',k} \geq \frac{1}{2^{k-1}+1}\lambda_k \geq \frac{1}{2^{k-1}+1}\lambda_{\text{MST},k}$. Thus, we obtain $C_{\text{MST},k} \leq \frac{1}{2^{k-2}+1} \cdot 2\lambda_{\text{MST},k} \leq 2 \cdot \frac{2^{k-1}+1}{2^{k-2}+1} \cdot C_{\pi',k} \leq 4 \cdot C_{\pi',k}$. So, we can 4-approximate π'. With that, Lemma 3.1 yields the theorem. ◀

▶ **Corollary 3.3.** *There is a 48-approximation for the HTSP in metric graphs.*

The arguments for using a k-MST algorithm to approximate a round of π' can be adapted to using an algorithm for k-TSP, asking for a minimum cost tour visiting exactly k vertices; clearly the problem is NP-hard [7]. Garg [13] gave a 3-approximation for the rooted variant of the k-TSP in metric graphs.

By applying the same arguments as in the k-MST approach, but without the factor 2 for traversing the spanning trees, we obtain the following.

▶ **Theorem 3.4.** *There is a 12α-approximation for the HTSP in metric graphs, with α denoting an approximation factor for the rooted k-TSP problem.*

Proof. Because the v_0-rooted $(2^{k-1}+1)$-TSP solution is the shortest tour visiting $2^{k-1}+1$ vertices including v_0, its total weight $\lambda_{\text{TSP},k}$ is not higher than λ_k. We concatenate the v_0-rooted $(2^{k-1}+1)$-TSP tours, with cost $C_{\text{TSP},k}$ each, for $k = 1, 2, \ldots$.

By $C_{\text{TSP},k} \leq \frac{1}{2^{k-2}+1} \cdot \lambda_{\text{TSP},k}$ and $C_{\pi',k} \geq \frac{1}{2^{k-1}+1} \cdot \lambda_k \geq \frac{1}{2^{k-1}+1} \cdot \lambda_{\text{TSP},k}$, we obtain $C_{\text{TSP},k} \leq \frac{1}{2^{k-2}+1} \cdot \lambda_{\text{TSP},k} \leq \frac{2^{k-1}+1}{2^{k-2}+1} \cdot C_{\pi',k} \leq 2 \cdot C_{\pi',k}$. By combining this with Lemma 3.1, the theorem follows. ◀

▶ **Corollary 3.5.** *There is a 36-approximation for the HTSP in metric graphs.*

4 Conclusion

We introduced both the Harmonic Traveling Salesman Problem and the Cumulative Traveling Salesman Problem, in which the cost of traveling an edge depends inversely proportional (or proportional, respectively) on the number of previously visited vertices. We provided a number of new results, including exact solutions in trees with a fixed number of leaves, and approximation for general metric instances.

There are numerous open questions, including the existence of better approximation algorithms, possibly making use of special geometric properties, as well as exact methods for computing provably optimal solutions for benchmark instances of interesting size.

References

1 Hernán Abeledo, Ricardo Fukasawa, Artur Pessoa, and Eduardo Uchoa. The Time Dependent Traveling Salesman Problem: Polyhedra and branch-cut-and-price algorithm. In *Symposium on Experimental Algorithms (SEA)*, pages 202–213, 2010. doi:10.1007/978-3-642-13193-6_18.

2 Tommaso Adamo, Gianpaolo Ghiani, Pierpaolo Greco, and Emanuela Guerriero. Learned upper bounds for the Time Dependent Traveling Salesman Problem. *IEEE Access*, 11:2001–2011, 2023. doi:10.1109/ACCESS.2022.3233852.

3 Ali Fuat Alkaya and Ekrem Duman. A new generalization of the Traveling Salesman Problem. *Applied and Computational Mathematics*, 9(2):162–175, 2010.

4 Aaron Archer and David P. Williamson. Faster approximation algorithms for the Minimum Latency Problem. In *Symposium on Discrete Algorithms (SODA)*, pages 88–96, 2003. URL: http://dl.acm.org/citation.cfm?id=644108.644122.

5 Sanjeev Arora. Polynomial time approximation schemes for Euclidean Traveling Salesman and other geometric problems. *Journal of the ACM*, 45(5):753–782, 1998. doi:10.1145/290179.290180.

6 Sanjeev Arora and George Karakostas. Approximation schemes for minimum latency problems. *SIAM Journal on Computing*, 32(5):1317–1337, 2003. doi:10.1137/S0097539701399654.

7 Sanjeev Arora and George Karakostas. A $2+\varepsilon$ approximation algorithm for the k-MST problem. *Math. Program.*, 107(3):491–504, 2006. doi:10.1007/S10107-005-0693-1.

8 Avrim Blum, Prasad Chalasani, Don Coppersmith, Bill Pulleyblank, Prabhakar Raghavan, and Madhu Sudan. The minimum latency problem. In *Symposium on Theory of Computing (STOC)*, pages 163–171, 1994. doi:10.1145/195058.195125.

9 Avrim Blum, Shuchi Chawla, David R. Karger, Terran Lane, Adam Meyerson, and Maria Minkoff. Approximation algorithms for orienteering and discounted-reward TSP. *SIAM Journal on Computing*, 37(2):653–670, 2007. doi:10.1137/050645464.

10 Kamalika Chaudhuri, Brighten Godfrey, Satish Rao, and Kunal Talwar. Paths, trees, and minimum latency tours. In *Symposium on Foundations of Computer Science (FOCS)*, pages 36–45, 2003. doi:10.1109/SFCS.2003.1238179.

11 Jean-François Cordeau, Gianpaolo Ghiani, and Emanuela Guerriero. Analysis and branch-and-cut algorithm for the Time Dependent Traveling Salesman Problem. *Transportation Science*, 48(1):46–58, 2014. doi:10.1287/trsc.1120.0449.

12 Alfredo Garcıa, Marc Noy, and Javier Tejel. Lower bounds on the number of crossing-free subgraphs of K_N. *Computational Geometry*, 16(4):211–221, 2000. doi:10.1016/S0925-7721(00)00010-9.

13 Naveen Garg. A 3-approximation for the minimum tree spanning k vertices. In *Conference on Foundations of Computer Science (FOCS)*, pages 302–309, 1996. doi:10.1109/SFCS.1996.548489.

14 Naveen Garg. Saving an epsilon: A 2-approximation for the k-MST problem in graphs. In *Symposium on Theory of Computing (STOC)*, page 396–402, 2005. doi:10.1145/1060590.1060650.

15 Michel Goemans and Jon Kleinberg. An improved approximation ratio for the Minimum Latency Problem. *Mathematical Programming*, 82(1):111–124, 1998. doi:10.1007/BF01585867.

16 Chryssi Malandraki and Mark S. Daskin. Time dependent vehicle routing problems: Formulations, properties and heuristic algorithms. *Transp. Sci.*, 26(3):185–200, 1992. doi:10.1287/TRSC.26.3.185.

17 Joseph S. B. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric TSP, k-MST, and related problems. *SIAM Journal on Computing*, 28(4):1298–1309, 1999. doi:10.1137/S0097539796309764.

18 R. Ravi, Ravi Sundaram, Madhav V. Marathe, Daniel J. Rosenkrantz, and S. S. Ravi. Spanning trees—short or small. *SIAM Journal on Discrete Mathematics*, 9(2):178–200, 1996. doi:10.1137/S0895480194266331.

19 Micha Sharir, Adam Sheffer, and Emo Welzl. Counting plane graphs: perfect matchings, spanning cycles, and Kasteleyn's technique. In *Symposium on Computational Geometry (SoCG)*, pages 189–198, 2012. doi:10.1145/2261250.2261277.

20 Micha Sharir, Adam Sheffer, and Emo Welzl. Counting plane graphs: Perfect matchings, spanning cycles, and Kasteleyn's technique. *Journal of Combinatorial Theory, Series A*, 120(4):777–794, 2013. doi:10.1016/J.JCTA.2013.01.002.

21 Russ J. Vander Wiel and Nikolaos V. Sahinidis. An exact solution approach for the Time Dependent Traveling Salesman Problem. *Naval Research Logistics (NRL)*, 43(6):797–820, 1996.