Approximation and Hardness of Polychromatic TSP

Thomas Schibler* Subhash Suri* Jie Xue†

Abstract

We introduce the *Polychromatic Traveling Salesman Problem* (PCTSP), where the input is an edge weighted graph whose vertices are partitioned into k equal-sized color classes, and the goal is to find a minimum-length Hamiltonian cycle that visits the classes in a fixed cyclic order. This generalizes the Bipartite TSP (when $k = 2$) and the classical TSP (when $k = n$). We give a polynomial-time $(3-2\cdot 10^{-36})$-approximation algorithm for metric PCTSP. Complementing this, we show that Euclidean PCTSP is APX-hard even in $\mathbb{R}^2$, ruling out the existence of a PTAS unless P = NP.

1 Introduction

The classic Traveling Salesman Problem (TSP) takes an edge-weighted graph G as input and aims to find a minimum-weight Hamiltonian cycle in G (which by definition is a simple cycle that visits all vertices of G). As one of the most fundamental algorithmic problems, TSP has stood as a testing ground for algorithmic developments in the theory community since its introduction as one of Karp's original NP-Complete problems. In 1976, Christofides showed a 1.5-approximation for any metric graph G, and this bound resisted progress for decades until Karlin et al. [6] improved the bound (slightly) to $1.5 - 10^{-36}$, which remains the best known. In the case that G contains only edge weights 1 and 2, Papadimitriou and Yannakakis obtained a better approximation ratio of 7/6, and showed that even this restricted case is APX-Hard, and thus does not admit a PTAS unless P=NP [8]. However, if the edge weights correspond to the Euclidean distances between points in some fixed $\mathbb{R}^d$, then a PTAS is known from the celebrated work of Arora [2] and Mitchell.

In this paper, we consider a colored version of TSP, called *polychromatic TSP* (PCTSP), in which the vertices of G are partitioned into color classes of the same size and the goal is to find a TSP tour that repeatedly visits the color classes in a consistent order. Formally, a *polychromatic graph* consists of a graph G together with a partition $\mathcal{P} = \{V_1, \cdots, V_k\}$ of $V(G)$ into subsets (called *color classes*) of equal size (i.e., $|V_1| = \cdots = |V_k|$) and a weight function $w : E(G) \to \mathbb{R}_{\geq 0}$. Let $(G, \mathcal{P}, w)$ be a polychromatic graph with $|V(G)| = n$ and $|\mathcal{P}| = k$. The *weight* of a path/cycle π in G, denoted by $w(\pi)$, is the sum of the weights of the edges in π under the weight function w. For a permutation $\sigma = (\sigma(0), \sigma(1), \ldots, \sigma(k-1))$ of $[k] = \{1, \cdots, k\}$, we say a cycle C in G is a *σ-cycle* if it can be written as $C = (v_0, v_1, \cdots, v_r)$ where $v_r = v_0$ such that $v_i \in V_{\sigma(i \bmod k)}$ for all $i \in \{0\} \cup [r]$; the weight of C is equal to $\sum_{i=1}^{n} w((v_{i-1}, v_i))$. Note that the length of a σ-cycle is always a multiple of k. Intuitively, a σ-cycle repeatedly visits the color classes of G in the order specified by σ. The goal of PCTSP is to find a minimum-weight Hamiltonian cycle in G that is a σ-cycle for some permutation σ of $[k]$. Clearly, when $k = n$, PCTSP is exactly the classic TSP. As such, PCTSP is a generalization of TSP and is thus NP-hard even when G is a metric graph. The main focus of this paper is to study the approximability of PCTSP in metric graphs, and more specifically, Euclidean graphs in a fixed dimension.

We remark that PCTSP is a natural extension of Bipartite TSP, where a tour must alternate between red and blue nodes. This problem has long been studied in the robotics literature as an appropriate model for "pick and place" route planning [1, 4, 10]. (In this setting, a robot with unit capacity must move objects from a set of sources to a set of destinations, necessitating a route that alternates between the two node types.) For metric graphs, Anily and Hassin gave a 2.5-approximation [1], and Chalasani et al. [4] later improved the ratio to 2.

Lastly, many other lines of research consider chromatic variants of the TSP. For example, Dross et al. [5] give a Gap-ETH tight approximation scheme for the bicolored noncrossing Euclidean TSP, where the goal is to find a separate tour for the points of each color, such that the two do not intersect in $\mathbb{R}^2$. Baligács et al. give a constant factor approximation for the 3-color version [3]. Other settings involve multiple salesman that can only visit given subsets of the colors, or seek the shortest cycle that visits each color once. In contrast, we aim for a single polychromatic tour that visits every node of G while cycling through a large number of colors.

1.1 Main results

We initiate the study of metric PCTSP and Euclidean PCTSP in a fixed dimension. Our first result is a polynomial-time constant-approximation algorithm for

*University of California, Santa Barbara, CA, USA, {tschibler,suri}@ucsb.edu

†New York University Shanghai, China, jiexue@nyu.edu

PCTSP in metric graphs.

Theorem 1 *There is a polynomial-time* $(3-2\cdot 10^{-36})$*-approximation algorithm for metric PCTSP.*

To complement the above algorithmic result, we rule out the existence of (Q)PTASes for metric PCTSP and even Euclidean PCTSP, by proving the following APX-hardness result.

Theorem 2 *Euclidean PCTSP in* $\mathbb{R}^2$ *does not admit a PTAS unless P=NP. In particular, Euclidean PCTSP is APX-Hard.*

Organization. The rest of the paper is organized as follows. In Section 2, we give the basic notions and preliminaries that are required for our results. Section 3 presents our approximation algorithm for metric PCTSP, and Section 4 presents our hardness result for Euclidean PCTSP.

2 Preliminaries

For a graph G, we use $V(G)$ and $E(G)$ to denote the vertex set and edge set of G, respectively. For $V \subseteq V(G)$, we denote by $G[V]$ the graph induced in G by V. Let $(G, \mathcal{P}, w)$ be a polychromatic graph where $\mathcal{P} = \{V_1, \dots, V_k\}$. We call a cycle in G a *polychromatic cycle* in $(G, \mathcal{P}, w)$ if it is a σ-cycle for some permutation of $[k]$. We denote by $\mathcal{H}_G$ the set of all Hamiltonian σ-cycles of G: $\mathcal{H}_G = \{\pi : \pi \text{ is a polychromatic cycle and } V(\pi) = V(G)\}$.

In metric PCTSP, we are given a polychromatic graph $(G, \mathcal{P}, w)$ such that G is complete and $w((v, v')) \leq w((v, v'')) + w((v'', v'))$ for all $v, v', v'' \in V(G)$. We are to compute $\arg\min_{\pi\in\mathcal{H}_G} w(\pi)$. In Euclidean PCTSP, the input is $(P, \mathcal{P})$ for a set of points $P \subset \mathbb{R}^d$, and partition of the points $\mathcal{P} = \{P_1, \cdots, P_k\}$. The task is to solve metric PCTSP on input $(G, \mathcal{P}', w)$, where G is the complete graph with vertices $V(G) = \{v_p : p \in P\}$, $\mathcal{P}' = \{P'_1, \cdots, P'_k\}$ s.t. $P'_i = \{v_p : p \in P_i\}$ for all $1 \leq i \leq k$, and $w((v_p, v_{p'})) = ||p - p'||_2$. $||\cdot||_2$ denotes the Euclidean norm. For convenience, we similarly refer to a cycle of the points $\pi = (p_0, \cdots, p_r)$ where $p_r = p_0$ as a σ-cycle if $p_i \in P_{\sigma(i \bmod k)}$ for all $i \in \{0\} \cup [r]$. The length of π is $||\pi|| = \sum_{i\in[r]} ||p_i - p_{i+1}||_2$. Finally, we use the term σ-tour to mean a Hamiltonian σ-cycle.

3 Constant approximation for metric PCTSP

In this section, we give our polynomial-time constant-approximation algorithm for metric PCTSP. Consider an instance $(G, \mathcal{P}, w)$ of metric PCTSP, where $|V(G)| = n$ and $\mathcal{P} = \{V_1, \dots, V_k\}$.

At a high level, we solve the problem via two steps. First, we give an algorithm that can compute in polynomial time, for any given permutation σ of $[k]$, an $O(1)$-approximation for the optimal σ-tour. When $k = O(1)$, this already solves metric PCTSP since we can simply run the algorithm on all $k!$ possibilities of σ. However, we do not have the assumption that k is a constant in PCTSP. Therefore, in the second step, we show how to approximate the optimal ordering σ in polynomial time.

3.1 Computing the tour for a fixed order

We first consider how to compute a good σ-tour in G for a given ordering σ of $[k]$. Without loss of generality, we can assume that $\sigma = (1, \dots, k)$. For convenience, define $V_{k+1} = V_1$. Our algorithm is presented in Algorithm 1. The sub-routine $\textsc{MinMatching}(G, w, V_i, V_{i+1})$ in line 2 computes a minimum-weight perfect matching M_i between V_i and V_{i+1} under the weight function w, which must exist because $|V_i| = |V_{i+1}|$ and G is a complete graph. Define H as the subgraph of G consisting of the edges in $\bigcup_{i=1}^{k} M_i$ (line 3), and let $\mathcal{C}$ be the set of connected components of H (line 4). Note that each $C \in \mathcal{C}$ is a σ-cycle in G. Next, we arbitrarily pick a vertex $v_C \in V(C) \cap V_1$ in each $C \in \mathcal{C}$ (line 6) and let $V'_1 \subseteq V_1$ be the set of these vertices (line 7). We then call the sub-routine $\textsc{ApproxTSP}(G[V'_1], w)$ in line 8, which computes a $(1.5 - 10^{-36})$-approximation TSP T of the subgraph $G[V'_1]$ under the weight function w, which can be done using the algorithm of Karlin et al. [6]. Finally, we "glue" the cycles in $\mathcal{C}$ along T as follows to obtain a Hamiltonian σ-cycle S (line 9). Write $\mathcal{C} = \{C_1, \dots, C_t\}$ such that $T = (v_{C_1}, \dots, v_{C_k}, v_{C_1})$. For $i \in [t]$, let u_{C_i} be the neighbor of v_{C_i} in C_t that belongs to V_k. Then we take the disjoint σ-cycles $C_1, \dots, C_t$, remove the edges $(u_{C_1}, v_{C_1}), \dots, (u_{C_k}, v_{C_k})$, and add the edges $(u_{C_1}, v_{C_2}), \dots, (u_{C_{k-1}}, v_{C_k}), (u_{C_k}, v_{C_1})$. This results in a Hamiltonian cycle S in G, and one can easily verify that S is a σ-cycle. We return S as the output of our algorithm (line 10).

Algorithm 1 $\textsc{FixedOrderTour}(G, \{V_1, \cdots, V_k\}, w)$

1: **for** every $i \in [k]$ **do**
2: $\quad M_i \leftarrow \textsc{MinMatching}(G, w, V_i, V_{i+1})$
3: $H \leftarrow (V(G), \bigcup_{i=1}^{k} M_i)$
4: $\mathcal{C} \leftarrow$ set of connected components of H
5: **for** every $C \in \mathcal{C}$ **do**
6: $\quad$ pick an arbitrary vertex $v_C \in V(C) \cap V_1$
7: $V'_1 \leftarrow \{v_C : C \in \mathcal{C}\}$
8: $T \leftarrow \textsc{ApproxTSP}(G[V'_1], w)$
9: $S \leftarrow \textsc{Glue}(T, \mathcal{C})$
10: **return** S

In the rest of this section, we show that S is a $(2.5 - 10^{-36})$-approximation for the optimal σ-tour of G.

Lemma 3 *$\sum_{i=1}^{k}\sum_{e\in M_i} w(e) \le \sum_{e\in E(R)} w(e)$ for any Hamiltonian σ-cycle R in G.*

Proof. Note that the edges of R between V_i and V_{i+1} form a perfect matching between V_i and V_{i+1}, for $i \in [k]$. Since M_i is a minimum-cost perfect matching between V_i and V_{i+1}, we have $\sum_{i=1}^{k}\sum_{e\in M_i} w(e) \le \sum_{e\in E(R)} w(e)$. □

Lemma 4 *$\sum_{e\in E(S)} w(e) \le \sum_{i=1}^{k}\sum_{e\in M_i} w(e) + (1.5 - 10^{-36})\cdot \mathsf{tsp}(G)$, where $\mathsf{tsp}(G)$ is the minimum weight of a Hamiltonian cycle of G.*

Proof. For convenience, let us write $C_{t+1} = C_1$. We observe that $\sum_{e\in E(S)} w(e) = \sum_{i=1}^{k}\sum_{e\in M_i} w(e) + \sum_{i=1}^{t}(w((u_{C_i}, v_{C_{i+1}})) - w((u_{C_i}, v_{C_i})))$. Since G is a metric graph, we have $w((u_{C_i}, v_{C_{i+1}})) - w((u_{C_i}, v_{C_i})) \le w((v_{C_i}, v_{C_{i+1}}))$. It follows that $\sum_{i=1}^{t}(w((u_{C_i}, v_{C_{i+1}})) - w((u_{C_i}, v_{C_i}))) \le \sum_{e\in E(T)} w(e)$. Recall that T is a $(1.5 - 10^{-36})$-approximation TSP T of $G[V_1']$. As $G[V_1']$ is an induced subgraph of G and G is a metric graph, we have $\sum_{e\in E(T)} w(e) \le (1.5 - 10^{-36})\mathsf{tsp}(G)$. □

Since $\mathsf{tsp}(G) \le \sum_{e\in E(R)} w(e)$ for any Hamiltonian σ-cycle R in G, the above lemmas imply that $\sum_{e\in E(S)} w(e) \le (2.5 - 10^{-36})\sum_{e\in E(R)} w(e)$ for any Hamiltonian σ-cycle R in G.

3.2 Solving metric PCTSP

In this section, we show how to compute a good ordering σ. Together with the algorithm in the previous section, this shall give us the final algorithm for Theorem 1.

For $i, j \in [k]$ with $i \neq j$, we compute a minimum-cost perfect matching $M_{i,j}$ between V_i and V_j in G. Let Σ_k denote the set of all permutations of $[k]$. For $\sigma = (r_1, \dots, r_k) \in \Sigma_k$, let H_σ denote the subgraph of G consisting of the edges in $\bigcup_{i=1}^{k} M_{r_i, r_{i+1}}$; here we set $r_{k+1} = r_1$ for convenience. Define $w_\sigma = \sum_{e\in E(H_\sigma)} w(e)$.

Suppose opt is the optimum of the PCTSP instance, i.e., the weight of an optimal solution. Note that $\mathsf{opt} \ge \mathsf{tsp}(G)$. By Lemma 3, we have $\mathsf{opt} \ge \min_{\sigma\in\Sigma_k} w_\sigma$. If $\hat{\sigma} \in \Sigma_k$ satisfies that $w_{\hat{\sigma}} \le c \cdot \min_{\sigma\in\Sigma_k} w_\sigma$, then Lemma 4 further implies that when running the algorithm in the previous section on $\hat{\sigma}$, the output tour S satisfies $\sum_{e\in E(S)} w(e) \le (c + 1.5 - 10^{-36}) \cdot \mathsf{opt}$. Based on this observation, it suffices to find $\hat{\sigma} \in \Sigma_k$ that (approximately) minimizes $w_{\hat{\sigma}}$.

We achieve this goal as follows. Build a complete graph G' with $V(G') = \{V_1, \dots, V_k\}$, and define a weight function $w' : E(G') \to \mathbb{R}_{\ge 0}$ by setting $w'((V_i, V_j)) = \sum_{e\in E(M_{i,j})} w(e)$. One can easily check that G' is also a metric graph. Observe that for each $\sigma = (r_1, \dots, r_k) \in \Sigma_k$, w_σ is just equal to the weight of the Hamiltonian cycle $(V_{r_1}, \dots, V_{r_k}, V_{r_1})$ in G' under w'. As such, we simply compute a $(1.5 - 10^{-36})$-approximation TSP $(V_{\hat{r}_1}, \dots, V_{\hat{r}_k}, V_{\hat{r}_1})$ in G'. Then $\hat{\sigma} = (\hat{r}_1, \dots, \hat{r}_k)$ satisfies that $w_{\hat{\sigma}} \le (1.5 - 10^{-36}) \cdot \min_{\sigma\in\Sigma_k} w_\sigma$. Therefore, if we apply the algorithm in the previous section on $\hat{\sigma}$, the output tour S satisfies $\sum_{e\in E(S)} w(e) \le (3 - 10^{-36}) \cdot \mathsf{opt}$. This completes the proof of Theorem 1.

Theorem 1 *There is a polynomial-time $(3 - 2\cdot 10^{-36})$-approximation algorithm for metric PCTSP.*

Remark. As one can verify from our algorithm, the approximation factor and the running time of the algorithm in Theorem 1 in fact depend on the best matching algorithm and metric TSP algorithm. Specifically, let $M(n)$ be the time for computing a minimum-cost bipartite matching in a complete bipartite graph with n vertices and $T(n)$ be the time for computing an α-approximate TSP in a metric graph with n vertices. Then we get a 2α-approximation algorithm for PCTSP with running time $O(k^2 M(2n/k) + T(k) + T(n/k))$.

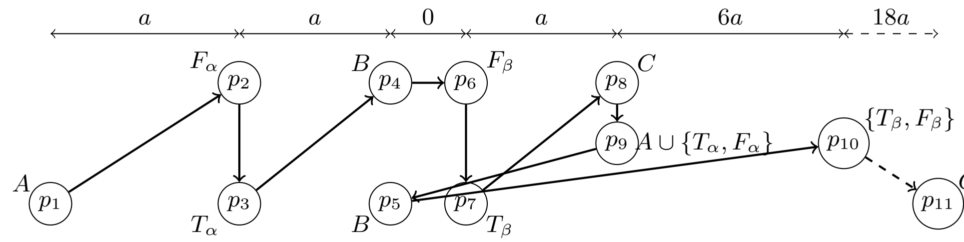
4 APX-Hardness for Euclidean PCTSP in $\mathbb{R}^2$

In this section, we prove Theorem 2 by a reduction from Max 2-SAT, which is known to be APX-hard [7]. The input to Max 2-SAT is a set of n variables $\{x_1, \cdots, x_n\}$ and m clauses $\{c_1, \cdots, c_m\}$. Each clause is a conjunction of two literals; a literal is a variable x_i or its negation $\bar{x}_i$. The task is to find a boolean assignment of the variables that satisfies the maximum number of clauses.

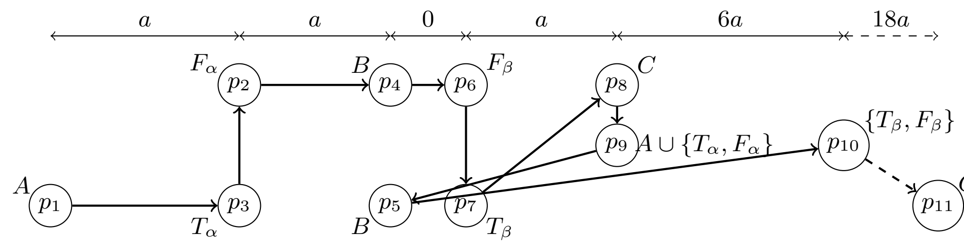
The general idea behind our reduction is to encode the truth assignment of SAT variables in the choice of permutation σ. We then test a corresponding tour with a series of clause gadgets; informally, if σ fails to "satisfy" a clause, then the clause gadget will penalize the tour by a fixed small amount. The structure of the proof is as follows. We first describe how to encode a truth assignment using σ. We then introduce our clause gadget and reason about its structure in Lemma 5. Finally, we build the full construction using a series of clause gadgets and an auxiliary set of points S. The points of S serve two purposes: (1) they rule out the possibility of σ that does not properly encode a truth assignment, and (2) they divert the return trip of the tour away from the clause gadgets so that we may reason about a single pass through our gadgets.

Our construction uses $k = 3n + 1$ color classes (point sets), which we will label R_α, T_α, F_α for $\alpha \in [1, n]$, and the additional R_{n+1}. For a permutation σ, let $\prec_\sigma$ be the total order of the classes under σ, i.e. $V_{\sigma(0)} \prec_\sigma V_{\sigma(1)} \cdots \prec_\sigma V_{\sigma(k-1)}$. We say that a permutation σ is *valid* if it satisfies:

1. $R_\alpha \prec_\sigma T_\alpha$, $R_\alpha \prec_\sigma F_\alpha$, for all $1 \le \alpha \le n$ and
2. $T_\alpha \prec_\sigma R_\beta$, $F_\alpha \prec_\sigma R_\beta$, for all $1 \le \alpha < \beta \le n + 1$.



(a) Optimal σ-path visiting C_i when σ does not satisfy the clause, i.e. $F_\alpha \prec_\sigma T_\alpha$, $F_\beta \prec_\sigma T_\beta$



(b) Optimal σ-path visiting C_i where σ satisfies x_α but not x_β, i.e. $T_\alpha \prec_\sigma F_\alpha$, $F_\beta \prec_\sigma T_\beta$

Figure 1: The clause gadget C_i for clause $c_i = x_\alpha \vee x_\beta$ showing two possible σ-paths. The gadget consists of $6m+2$ points (two of each color) placed at 9 distinct locations. (Notice that p_4, p_6 and p_5, p_7 coincide, but we distinguish them to make it clear when a path revisits a location.) At an intuitive level, the gadget tests whether σ satisfies x_α (i.e. that $T_\alpha \prec_\sigma F_\alpha$) in the triangle (p_1, p_2, p_3). Observe that path (a) does not satisfy x_α, and must take the longer leg (p_1, p_2), while path (b) takes the shorter leg (p_2, p_3) by satisfying x_α. Similarly, triangle (p_6, p_7, p_8) tests x_β. However, between these two tests, the tour has the choice of which set B to visit, either at p_4 or p_5, such that satisfying exactly one of x_α or x_β allows the tour to take an edge of rectangle (p_2, p_3, p_5, p_4) while satisfying both (or neither) of x_α, x_β forces the path to take a diagonal. This avoids over-rewarding σ for satisfying both x_α and x_β, and is the main idea behind the gadget.

Observe that there are exactly 2^n valid permutations that naturally correspond to truth assignments of the variables $\{x_1, \cdots, x_n\}$. For each $\alpha \in [1, n]$, we only have the freedom to choose whether $T_\alpha \prec_\sigma F_\alpha$ or $F_\alpha \prec_\sigma T_\alpha$. We say that σ *satisfies* the literal x_α if $T_\alpha \prec_\sigma F_\alpha$; likewise σ satisfies $\bar{x}_\alpha$ if $F_\alpha \prec_\sigma T_\alpha$. σ satisfies a clause c_i if it satisfies either of the literals of c_i.

4.1 Clause gadget

The crux of our reduction lies in the creation of a clause gadget that tests σ on whether it satisfies a given clause. This test can only depend on σ's encoding of the two variables in that clause, so the gadget will need to allow the tour to "jump through" all of the colors that precede $\{T_\alpha, F_\alpha\}$, that fall between $\{T_\alpha, F_\alpha\}$, $\{T_\beta, F_\beta\}$, and that come after $\{T_\beta, F_\beta\}$, respectively. Formally, we define these color sets for use in the gadget, for all α, β, s.t. $1 \leq \alpha \leq \beta \leq n$:

- $A_{\alpha,\beta} = \{R_{\alpha'}, T_{\alpha'}, F_{\alpha'} | 0 < \alpha' < \alpha\} \cup \{R_\alpha\}$
- $B_{\alpha,\beta} = \{R_{\alpha'}, T_{\alpha'}, F_{\alpha'} | \alpha < \alpha' < \beta\} \cup \{R_\beta\}$
- $C_{\alpha,\beta} = \{R_{\alpha'}, T_{\alpha'}, F_{\alpha'} | \beta < \alpha' < n+1\} \cup \{R_{n+1}\}$

When the indices α and β are clear, we abbreviate the sets as A, B, C. These sets are nonempty, and any valid σ satisfies (by definition):

$$A \prec_\sigma \{T_\alpha, F_\alpha\}, \prec_\sigma B \prec_\sigma \{T_\beta, F_\beta\} \prec_\sigma C \qquad (1)$$

Figure 1 shows the gadget, along with (optimal) example paths for when σ does, and does not satisfy the clause. We describe the position of the points of the gadget (and later the spacing between gadgets) in terms of constants a and b, with the intuition that $b \gg a \gg 1$. Let $x > 0$. A clause gadget C_i for the clause $c_i = x_\alpha \vee x_\beta$ anchored at x is the following collection of points.[1] (When we refer to a set of colors X at a location, we mean a set of $|X|$ coincident points at that location, one of each color in the set X.)

- Set A at $(x, -1)$
- Points of color F_α, T_α at $(x+a, 1), (x+a, -1)$, resp.

[1] For clauses that include negative literals (e.g. $\bar{x}_\alpha$ instead of x_α), we swap the corresponding colors (T_α and F_α) in the gadget construction. The key is that the satisfying color appears on the line $y = -1$, and the other on the line $y = 1$. Without loss of generality, we will assume these colors are labeled as T_α and T_β in our analysis.

- Two sets B at $(x+2a, \pm 1)$
- Points of F_β, T_β at $(x+2a, 1), (x+2a, -1)$, resp.
- Set C at $(x+3a, 1)$
- Set $A \cup \{T_\alpha, F_\alpha\}$ at $(x+3a, 0)$
- Set $\{T_\alpha, F_\alpha\}$ at $(x+9a, 0)$
- Set C at $(x+27a, -1)$

Let us first establish the length of the optimal path through the gadget. To do so, we need to clarify what it means for a path to be polychromatic. A path is a *σ-path* if it can be written as $(v_1, \cdots, v_{r-1})$ such that for some j, $v_i \in V_{\sigma(i+j \mod k)}$ for all $i \in [r]$. Note that a σ-path may start at a point of any color. We will also need the following definitions. Let a permutation σ be *α, β-valid* if equation 1 holds for σ, α, β. Two permutations σ, σ' *agree* on x_α if $T_\alpha \prec_\sigma F_\alpha$ iff $T_\alpha \prec_{\sigma'} F_\alpha$.

Lemma 5 *Let C_i be the clause gadget for $c_i = x_\alpha \vee x_\beta$. For any valid σ, let T_i^* be the shortest σ-path that visits all points of C_i. Then,*

$$\|T_i^*\| = \begin{cases} c + 2a & \text{if } \sigma \text{ satisfies } c_i \\ c + 2\sqrt{a^2+4} & \text{otherwise} \end{cases}$$

where $c = 5 + \sqrt{a^2+4} + \sqrt{a^2+1} + \sqrt{49a^2+1} + \sqrt{(18a)^2+1}$. Furthermore, there is no σ' and σ'-path shorter than T_i^* visiting C_i such that σ' is α, β-valid and agrees with σ on x_α, x_β.

Proof. Due to space constraints, the proof is included in the full version. [9] □

4.2 Full construction

We now describe the full construction, depicted in Figure 2. Given a Max 2-SAT instance $\Pi = (\{x_1, \cdots, x_n\}, \{c_1, \cdots, c_m\})$ we create a set of points $P_\Pi = S \cup C$. S will be a special set containing exactly one point from each color class. C is a collection of clause gadgets $C_1 \cup \cdots \cup C_m$. For each clause c_i, C_i is anchored at $(i-1)(9a+b)$. Fix W as the maximum x-coordinate of a point in C, i.e. $W = 27ma + (m-1)b$.

We place the points of S along the horizontal line $y = -(2W+1)$ from East to West following a valid permutation. Points of the colors T_i, F_i are coincident for each i, so that *any* valid tour may visit S via a straight line, while any invalid tour will have to backtrack. The spacing around the points is proportional to the number of clauses that x_i appears in. The idea behind this spacing is to only penalize invalid tours relative to the number of clause gadgets they could "cheat" in (precisely those for which they are not α, β-valid), since we cannot make the total spacing too large while preserving approximate solutions. Specifically, for each variable x_i, let n_i be the number of clauses that it or its negation appears in. Clearly $\sum_{i=1}^n n_i = 2m$, since each clause contains exactly two literals. Let $N_i = \sum_{j=1}^i n_j$. S contains the following points for $(1 \le i \le n)$, where $l = \frac{27}{4}a + (\frac{1}{4} - \frac{1}{m})b$:

- Point r_i of color R_i at $(W - 2lN_{i-1}, -2W-1)$
- Points t_i, f_i of colors T_i, F_i resp. both located at $(W - 2lN_{i-1} - ln_i, -2W-1)$

Finally, S contains a point r_{n+1} of color R_{n+1} at $(0, -2W-1) = (W - 2lN_n, -2W-1)$. We can now state the length of the optimal σ-tour for any valid σ.

Lemma 6 *Let σ be a valid permutation satisfying k clauses of Π. Then there is a σ-tour T^* of $P_\pi = S \cup C$ s.t.*

$$\|T^*\| = 5W + (m-1)b + k(c+2a) + (m-k)(c + 2\sqrt{a^2+4})$$

where $c = f(a)$ is the constant defined in lemma 5.

Proof. We construct a tour T^* that first visits the points of S via a straight line and then visits each of the clauses in ascending order $C_1, \cdots, C_m$, using the σ-path T_i^* of Lemma 5 to visit each C_i. First, write $S = \{s_1, \cdots, s_{3n+1}\}$ s.t. $s_1 \prec_\sigma \cdots \prec_\sigma s_{3n+1}$, and let $T_S = (s_1, \cdots, s_{3n+1})$ be the unique σ-path visiting S. By construction, $||T_S|| = W$ for any valid σ. Now, let u_i, v_i be the first and last points of T_i^*, respectively. Then T^* consists of the edges in T_S, the edges in $T_1^*, \cdots, T_m^*$, and the additional edges $(s_{3n+1}, u_1), (v_m, s_1)$, and (v_i, u_{i+1}) for all $1 \le i < m$. By Lemma 1, $\sum_{i \in [1,m]} T_i^* = k(c+2a) + (m-k)(c + 2\sqrt{a^2+4})$. Furthermore, $||s_{3n+1} - u_1|| = ||v_m, s_1|| = 2W$, and $\sum_{i=1}^m ||v_i - u_{i+1}|| = (m-1)b$. Summing yields the desired total length. □

We now argue that the existence of a σ-cycle with length close to that of T^* in Lemma 6 implies that σ satisfies close to k clauses of Π. To do so, we will need to break apart an arbitrary tour for a finer-grained analysis. With this goal in mind, let T^σ be a σ-tour (for possibly invalid σ), and let $T^\sigma[S]$ and $T^\sigma[C]$ be the portions of T^σ induced on S and C respectively. It is worth pointing out that, in general, $T^\sigma[S]$ and $T^\sigma[C]$ may not be connected. However, we have chosen a suitable separation of distance $2W$ between S and C so that any tour that alternates between S and C multiples times has length that well exceeds that of T^*. Thus, we can safely ignore these tours, and assume that both $T^\sigma[S]$ and $T^\sigma[C]$ are connected.

Now, let us further decompose $T^\sigma[C]$: we will use T_i^σ to denote the portion of $T^\sigma[C]$ spent inside the clause

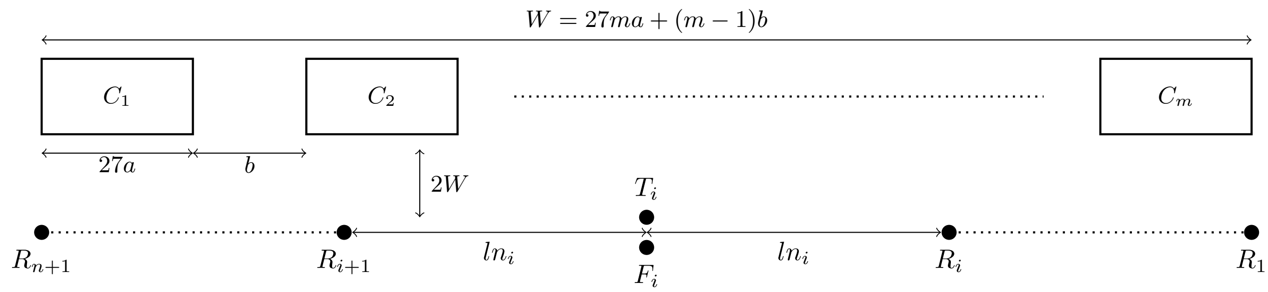


Figure 2: The instance $P_\Pi = S \cup C$. C (top) is the union of m clause gadgets $C_1 \cdots C_m$. S (bottom) contains exactly one point of each color placed along the line $y = -2W - 1$.

gadget C_i. More specifically, consider the set of vertical lines L given by the equations $x = (27a + b)(i - 1)$ and $x = (27a + b)(i - 1) + 27a$ for $i \in [1, m]$, and let us subdivide each leg of $T^\sigma[C]$ at each crossing of a line in L. After subdivision, let T_i^σ be the set of line segments of $T^\sigma[C]$ contained (inclusively) between the vertical lines $x = (27a+b)(i-1)$ and $x = (27a+b)(i-1)+27a$. (T_i^σ may not be connected, and may include partial legs of $T^\sigma[C]$.) Collectively, the T_i^σ's do not account for the remaining portion of $T^\sigma[C]$ between gadgets, which has length at least $(m-1)b$. We get the following bound:

Observation 1 $\|T^\sigma\| > \sum_{i=1}^m \|T_i^\sigma\| + 5W + (m-1)b$.

If σ is valid, we can use Lemma 5 to bound the number of clauses σ satisfies based on the length of an approximate σ-tour. Lemma 8 will account for invalid σ.

Lemma 7 *Let T^σ be a σ-tour of P_Π for valid σ. Let $c = f(a)$ be the constant given in Lemma 5. If*

$$\|T^\sigma\| \leq 5W+(m-1)b+k(c+2a)+(m-k)(c+2\sqrt{a^2+4}),$$

then σ satisfies at least k clauses of Π.

Proof. For the sake of contradiction, let us assume that σ satisfies k' clauses for some $k' < k$. If every T_i^σ is connected, then by Lemma 5,

$$\sum_{i=1}^m \|T_i^\sigma\| > k'(c + 2a) + (m - k')(c + 2\sqrt{a^2 + 4}).$$

For each disconnected T_i^σ, T^σ incurs an additional length b to enter and leave the gadget C_i. We charge this overhead to T_i^σ, such that the above bound always holds as long as $b >> a$, e.g. $b > 30a$. However, from Observation 1 and the assumption of the lemma, we have that

$$\sum_{i=1}^m \|T_i^\sigma\| < 2 + k(c + 2a) + (m - k)(c + 2\sqrt{a^2 + 4}),$$

which implies that $(k - k')(2\sqrt{a^2 + 4} - 2a) < 0$, a contradiction since $k - k' > 0$. □

We lastly rule out the possibility of a σ-tour for invalid σ. Essentially, we are able to show that any gains made by "cheating" in the clause gadgets are offset by an equal increase in overhead to visit S. The proof is included in the full version. [9]

Lemma 8 *Let T^σ be a σ-tour of P_Π. There exists valid σ' and σ'-tour $T^{\sigma'}$ such that $\|T^{\sigma'}\| \leq \|T^\sigma\|$. Furthermore, σ' can be computed in polynomial time given σ.*

We are ready to prove theorem 2, which we restate here.

Theorem 2 *Euclidean PCTSP in $\mathbb{R}^2$ does not admit a PTAS unless P=NP. In particular, Euclidean PCTSP is APX-Hard.*

Proof. Assuming a PTAS for Euclidean PCTSP, we can derive a PTAS for Max 2-SAT, which contradicts its hardness [7]. Given an instance to Max 2-SAT Π with optimal solution value k and target error ϵ, we first construct the instance P_Π of Figure 2 in polynomial time. Let $f(k)$ be the length of the tour in Lemma 6. For an appropriate ϵ', we apply the PTAS for EPCTSP to recover a σ-tour of P_Π that has length at most $(1 + \epsilon')f(k)$, since OPT of P_Π is at most $f(k)$ by Lemma 6. We can assume σ is valid, else we can compute a valid σ' that admits a σ'-tour with equal or better cost in polynomial time by Lemma 8. By Lemma 7, σ satisfies at least $(1 - \epsilon)k$ clauses of Π for a sufficiently small choice of constant $\epsilon' = f(\epsilon, a, b)$ (see full version for details [9]). □

5 Conclusion

In this paper, we introduced the Polychromatic TSP and studied its metric and Euclidean variants. In the metric case, we gave a constant factor approximation that remains polynomial time for any number of colors. We complemented this algorithmic result with the nonexistence of a PTAS, even for points in the plane as long as the number of colors is unbounded. An interesting open question remains: does Euclidean PCTSP admit a PTAS for a constant number of colors?

References

[1] Shoshana Anily and Refael Hassin. The swapping problem. *Networks*, 22(4):419–433, 1992.

[2] S. Arora. Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems. *J. ACM*, 45(5):753–782, September 1998.

[3] J. Baligács, Y. Disser, A. E. Feldmann, and A. Zych-Pawlewicz. A $(5/3 + \epsilon)$-approximation for tricolored non-crossing euclidean tsp. In *32nd Annual European Symposium on Algorithms (ESA 2024)*, volume 308, pages 15:1–15:15, 2024.

[4] P. Chalasani, R. Motwani, and A. Rao. Algorithms for robot grasp and delivery. In *2nd International Workshop on Algorithmic Foundations of Robotics*, 1996.

[5] F. Dross, K. Fleszar, K. Węgrzycki, and A. Zych-Pawlewicz. Gap-eth-tight approximation schemes for red-green-blue separation and bicolored non-crossing euclidean travelling salesman tours. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1433–1463, 2023.

[6] A. R. Karlin, N. Klein, and S. O. Gharan. A (slightly) improved approximation algorithm for metric tsp. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, page 32–45. Association for Computing Machinery, 2021.

[7] C. H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43(3):425–440, 1991.

[8] C. H. Papadimitriou and M. Yannakakis. The traveling salesman problem with distances one and two. *Mathematics of Operations Research*, 18(1):1–11, 1993.

[9] T. Schibler, S. Suri, and J. Xue. Full version: Approximation and hardness of polychromatic tsp, 2025. https://arxiv.org/abs/2507.04974.

[10] A. Srivastav, H. Schroeter, and C. Michel. Approximation Algorithms for Pick-and-Place Robots. *Annals of Operations Research*, 107(1):321–338, October 2001.