Transactions on Combinatorics
ISSN (print): 2251-8657, ISSN (on-line): 2251-8665
Vol. 15 No. 4 (2026), pp. 305–315.

https://toc.ui.ac.ir
https://www.ui.ac.ir

COLORED POINTS TRAVELING SALESMAN PROBLEM

SAEED ASAEEDI

Abstract. The Colored Points Traveling Salesman Problem (Colored Points TSP) is introduced in this work as a novel variation of the traditional Euclidean Traveling Salesman Problem (TSP) in which the set of points is partitioned into multiple classes, each of which is represented by a distinct color (or label). The goal is to find a minimum cost cycle that visits all the colors so that each color appears only once. This problem finds diverse applications across various fields, including transportation, goods distribution, postal services, inspection, insurance, and banking. By reducing the traditional TSP to it, we can demonstrate that Colored Points TSP is NP-hard. Here, we offer a $\frac{2\pi r}{3}$-approximation algorithm for this problem when the points are placed on a unit grid, where r denotes the radius of the points' smallest color-spanning circle. The algorithm has been implemented, executed on random datasets, and compared against the brute force method.

1. Introduction

The Traveling Salesman Problem (TSP) is a famous classical problem in combinatorial optimization. Given a set of points in the plane, this problem finds the minimal cost cycle that visits each point exactly once. TSP was investigated by Laporte in [5], and several exact and approximate algorithms were surveyed for this problem. This classic problem has numerous variants. One notable example is the Chromatic Traveling Salesman Problem [11], which aims to minimize the number of salespeople needed while ensuring each salesman's tour does not exceed a predefined length limit.

MSC(2010): Primary: 05-08; Secondary: 68W25.

Keywords: Colored Points TSP, Colored TSP, TSP, Approximation Algorithm, Computational Geometry.

Article Type: Research Paper.

Communicated by Manouchehr Zaker.

Received: 24 February 2024, Accepted: 01 December 2025, Published Online: 04 December 2025.

Cite this article: Saeed Asaeedi, Colored points traveling salesman problem, *Trans. Comb.*, **15** no. 4 (202x) 305–315. https://dx.doi.org/10.22108/toc.2025.140814.2153 .

The Colorful Traveling Salesman Problem, introduced by Xiong et al. [14], is described on a connected undirected graph with colored edges. The objective is to find a Hamiltonian cycle on the graph with the minimum number of distinct colors. The Multicolor Traveling Salesman Problem is defined on a complete graph whose vertices are colored [4]. It aims to identify the optimal Hamiltonian cycle while bounding the number of vertices of the sub tour between two consecutive vertices of the same color. The Colored Traveling Salesman Problem, introduced by Li et al. [7], deals with a set of colored points. This problem represents a variant of the multiple traveling salesman problem, wherein each salesperson is associated with a distinct color and is permitted to visit points of the same color, along with shared points that are common to all salesperson. In [13], the colored traveling salesman problem is considered on the points with varying colors. The Labeled Traveling Salesman Problem [2] is defined as the problem of finding a tour on a complete graph with colored edges, aiming to maximize or minimize the number of distinct colors used.

We introduce the Colored Points TSP, a novel variation of the traditional Traveling Salesman Problem, in this paper. This problem is defined on a set of colored points in the plane, with the objective being to determine the shortest cycle that visits points of all distinct colors, ensuring that each color is visited exactly once. We prove the NP-hardness of the problem and propose an $\frac{2\pi r}{3}$-approximation algorithm for its solution, where r represents the radius of the smallest color-spanning circle containing the points.

The minimum spanning circle problem was introduced by Sylvester [6] in 1857. A linear time algorithm is presented by Megiddo [12] to solve this problem. Abellanas et al. [9] present two algorithms for computing the colored version of this problem, known as the Minimum Color-Spanning Circle Problem: Given n points with k different colors, the objective is to discover the smallest spanning circle that contains at least one point of each color. The first algorithm operates in $O(nk)$ time following the computation of the Farthest Color Voronoi Diagram [9] of the points in $O(n^2\alpha(k)\log k)$ time. Meanwhile, the second algorithm, faster for small values of k, runs in $O(k^3 n \log n)$.

Abellanas et al. [10] compute the smallest color-spanning objects, specifically axis-parallel rectangles and the narrowest strips. Recently, Acharyya et al. [1] have delved into considering the minimum color-spanning circle on imprecise points.

The rest of the paper is organized as follows. Section 2 formally defines the Colored Points TSP, presenting an exact and an approximation algorithm to address this problem. Section 3 delves into the numerical results of the presented algorithms. Section 4 concludes the paper by highlighting its achievements.

2. **Colored Points TSP**

Let $S = \{s_1, \ldots, s_n\}$ be a set of points in the plane and $C = \{c_1, \ldots, c_k\}$ be a set of distinct colors. We assume that each point s_i is associated with only one color c_j, which we indicate as $C(s_i) = c_j$.

https://dx.doi.org/10.22108/toc.2025.140814.2153

Consequently, the points are categorized into k classes, each containing points of the same color. We represent the number of points in the i'th class as x_i, where $i \in \{1, 2, \dots, k\}$ and $\Sigma_{i=1}^{k} x_i = n$. For every $i \in \{1, 2, \dots, k\}$, we consider CS_i as a vector of points with the color c_i, i.e., $|CS_i| = x_i$. Definition 2.1 introduces *Colored Points TSP* as follows:

Definition 2.1. *Colored Points TSP is the problem of finding a polygon* $P = (p_1, p_2, \dots, p_k, p_1)$ *with the shortest perimeter, ensuring that* $p_i \in S$ *for all* $i \in \{1, 2, \dots, k\}$*, and* $C(p_i) \neq C(p_j)$ *for all* $i \neq j \in \{1, 2, \dots, k\}$.

The Colored Points TSP presents a variation of the classic TSP, wherein the salesman visits all the colors (not all the points) exactly once. This concept finds applications across diverse domains, including transportation, goods distribution networks, postal systems, inspections, insurance, banking, and more. Wherever one encounters a scenario involving various branches of different service providers, and the primary aim is to navigate through precisely one branch of each service provider, optimizing for the shortest path, the Colored Points TSP emerges as a significant tool. We assume that the requirement of visiting a single branch of each provider is adequate, and that each branch of a provider is indistinguishable from the others.

In Fig. 1, we observe an instance of the problem for $n = 70$ and $k = 7$. It is clear that this problem is a generalization of the traditional TSP. When $k = n$, the Colored Points TSP reduces to the standard TSP. As shown in Theorem 2.2, the Colored Points TSP is NP-hard.

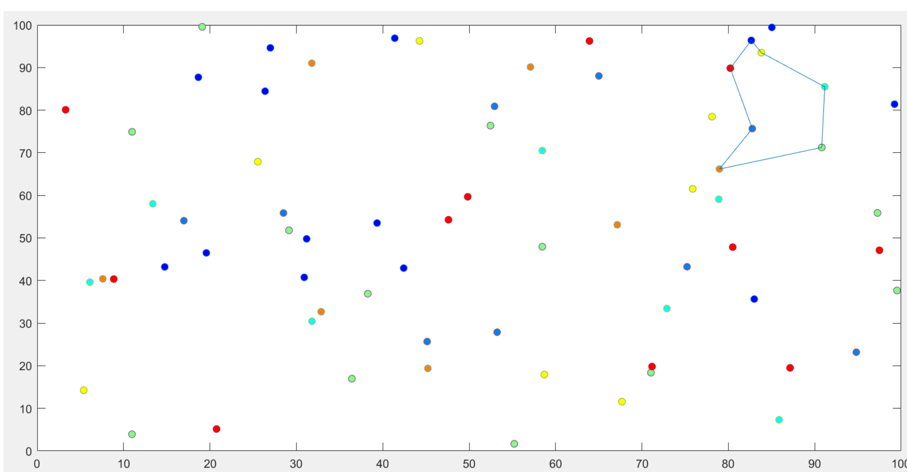

FIGURE 1. All 70 points are associated with one of seven distinct colors, and the line segments collectively form a polygon with the smallest possible perimeter, encompassing all 7 unique colors.

Theorem 2.2. *The Colored Points TSP is NP-hard.*

Proof. We reduce the classic TSP to the Colored Points TSP by transforming each instance of TSP, denoted as $A = \{s_1, s_2, \ldots, s_n\}$, into an instance of Colored Points TSP represented as $S = \{s_1, s_2, \ldots, s_n\}$, $C = \{c_1, c_2, \ldots, c_n\}$, and $C(s_i) = c_i$ for all $i \in \{1, 2, \ldots, n\}$. Notably, the polygon with the minimal perimeter traversing through all colors is equivalent to that with the minimal perimeter traversing through all points. □

Papadimitriou [3] introduced the Euclidean TSP and proved that it remains NP-hard even when all points have integer coordinates. From this result, the following corollary follows.

Corollary 2.3. *Colored Points TSP is NP-hard when the points are embedded in the plane with integer-coordinate positions.*

Theorem 2.4 shows that the problem retains its NP-hardness even under the restriction to grid-point inputs.

Theorem 2.4. *The Colored Points TSP is NP-hard over grid point inputs.*

Proof. We refer to the Colored Points TSP restricted to grid-point inputs as the Colored Grid-Points TSP. Given a set of points $S = \{s_1, s_2, \ldots, s_n\}$ in the plane with integer coordinates, we reduce the Colored Points TSP on S to the Colored Grid-Points TSP on a grid-aligned set S'. Based on Corollary 2.3, the Colored Points TSP on integer-coordinate point sets is NP-hard. Let $x(s_i)$ and $y(s_i)$ denote the $x-$ and $y-$coordinates of s_i, respectively. The following steps transform S into the grid-aligned set S':

(1) Find the GCD of all coordinate differences: For all $i \neq j \in \{1, \ldots, n\}$, compute $dx_{i,j} =\mid x(s_i) - x(s_j) \mid$, and $dy_{i,j} =\mid y(s_i) - y(s_j) \mid$. Let b be the greatest common divisor of all such $dx_{i,j}$ and $dy_{i,j}$.
(2) Compute the grid-cell size: The grid-cell size is given by $u = b$.

The set S' is then constructed by aligning the points of S to a grid with cell size u. □

In the following, we introduce the brute-force algorithm and an approximation algorithm to address the Colored Points TSP. In the brute-force algorithm, we examine all potential combinations of k distinct colors from n points and determine the one with the smallest possible perimeter. For a detailed description of this approach, refer to Algorithm 1.

https://dx.doi.org/10.22108/toc.2025.140814.2153

Algorithm 1 The brute-force algorithm for the Colored Points TSP

Require: $CS_i \ \forall i \in \{1, 2, \ldots, k\}$

Ensure: $P = (p_1, p_2, \ldots, p_k, p_1)$ as the polygon with the shortest perimeter that encompasses all distinct colors.

$P \leftarrow ()$
$min \leftarrow$ MAXINT
for $i_1 \leftarrow 1$ to x_1 **do**
  for $i_2 \leftarrow 1$ to x_2 **do**
    $\vdots$
    for $i_k \leftarrow 1$ to x_k **do**
      $Q = (CS[i_1], CS[i_2], \ldots, CS[i_k], CS[i_1])$
      if Perimeter of Q is less than min **then**
        $P \leftarrow Q$
        $min \leftarrow$ Perimeter of Q
      else
        Break
      end if
    end for
   $\vdots$
  end for
end for

The time complexity of Algorithm 1 is $O(x_1 * x_2 * \cdots * x_k)$. In the worst case, when $x_1 = x_2 = \cdots = x_k = \frac{n}{k}$, the time complexity is $O(\frac{n}{k}^k)$. In the following, we present an approximation algorithm for the Colored Points TSP, achieving a factor of $\frac{2\pi r}{3}$, where r represents the radius of the smallest color-spanning circle on S. This approach is detailed in Algorithm 2, which outlines the steps of the approximation algorithm utilizing the onion peeling technique and the minimum color-spanning circle algorithm. The onion peeling technique is defined as follows:

Definition 2.5. [8] *Start with the outermost points defining the convex hull and then iteratively remove these points and find the next layer of points that form the convex hull of the remaining set. The process of peeling away the layers is called onion peeling, and the count of layers is called onion depth.*

Algorithm 2 The approximation algorithm for the Colored Points TSP

Require: $S = \{s_1, s_2, \ldots, s_n\}$ as a set of n points on the unit grid, each assigned a color from $C = \{c_1, \ldots, c_k\}$

Ensure: $P = (p_1, p_2, \ldots, p_k, p_1)$ as the approximated polygon with the shortest perimeter that encompasses all distinct colors.

$R \leftarrow$ Minimum Color-Spanning Circle on S and C

$MSP \leftarrow$ The set of points inside R

$MSP \leftarrow$ Remove duplicate colors from MSP

$P \leftarrow ()$

while MSP is not empty **do**

  $L \leftarrow$ Points on the boundary of the convex hull of MSP

  $P \leftarrow P.L$ ▷ Concatenation of P and L

  $MSP \leftarrow MSP - L$ ▷ Remove L from MSP

end while

$P \leftarrow P.P(1)$ ▷ Add first point of P to P

In Algorithm 2, we begin by employing the algorithm outlined in [9] to compute R, representing the minimum color-spanning circle of the colored points S, within an efficient runtime of $O(k^3 n \log n)$. Subsequently, let MSP denote the set of points within the circle R, each of which contains at least one point of each color. To avoid visiting the same color multiple times, we eliminate points with duplicate colors from MSP. As a result, the cardinality of MSP is $|MSP| = k$.

In the next steps, we calculate the layers of onion peeling on MSP. For each layer L, we traverse the points of L clockwise, commencing with the leftmost point. In the worst-case scenario, where the maximum onion depth is $\frac{k}{3}$, the time complexity of the while loop is $O(k^2 \log k)$. Let $P = (p_1, p_2, \ldots, p_x)$ be the polygon computed after some iterations, and let $L = (l_1, l_2, \ldots, l_y)$ be the next layer of the onion peeling. The next iteration's polygon is formed by concatenating P and L: $P = (p_1, p_2, \ldots, p_x, l_1, l_2, \ldots, l_y)$. Subsequently, Theorem 2.7 accompanied by a lemma demonstrates that the approximation factor of algorithm 2 amounts to $\frac{2\pi r}{3}$. According to Pick's theorem, the lower bound for the area of each polygon within the set of n points is $\frac{n}{2} - 1$. Lemma 2.6 provides a lower bound for the perimeter of these polygons.

Lemma 2.6. *Given a set S of n points on a grid with cell size u, let P be the polygon with the minimum perimeter that crosses all the points of S. Then, the perimeter of P is greater than or equal to un.*

Proof. If P is a polygon with n edges, and we assume by contradiction that the perimeter of P is less than un, then it would imply that the length of an edge of P is less than u. However, since the points of S are on the grid, the distance between two points cannot be less than u. □

Theorem 2.7. *Let S be a set of points on a grid with cell size u. The approximation factor of Algorithm 2 is $\frac{2\pi r}{3u}$, where r is the radius of the smallest color-spanning circle on S.*

Proof. Given the points of MSP inside the circle R, each layer L of onion peeling has a length less than or equal to the circle's circumference, $2\pi r$. In the worst case, if we have $\frac{k}{3}$ layers, the perimeter of P will be less than or equal to $\frac{k}{3}2\pi r$. On the other hand, if we assume that OPT represents the perimeter of the optimal polygon P^* on MSP, as per Lemma 2.6, OPT is greater than or equal to uk. Therefore, the perimeter of P will be less than or equal to $\frac{2\pi r}{3u} * OPT$ □

Corollary 2.8. *Let S be a set of points on a unit grid. The approximation factor of Algorithm 2 on S is $\frac{2\pi r}{3}$, where r is the radius of the smallest color-spanning circle on S.*

Corollary 2.9. *Let S be a set of n points on a grid with cell size u where the points form a gapless arrangement (i.e., each row and column contain at least one point). The approximation factor of Algorithm 2 on S is $\frac{\sqrt{2}\pi n}{3}$. Since this arrangement guarantees $r \leq \frac{\sqrt{2}un}{2}$, substituting into the bound $P \leq \frac{2\pi r}{3u} * OPT$ yields $P \leq \frac{\sqrt{2}\pi n}{3}$ for the perimeter P.*

3. Implementation

We have implemented algorithms 1 and 2 in MATLAB and executed them on multiple datasets. The results shown in Fig. 2 pertain to sets of 50 and 70 random points, each with 5 and 7 distinct colors, respectively. It is important to note that due to its time-consuming nature, Algorithm 1 was not viable for running on larger datasets. Subsequently, Fig. 3 demonstrates the results of Algorithm 2 when applied to sets of 500 and 1000 random points, with 10 and 20 distinct colors, respectively.

As previously discussed, Algorithm 1 has a time complexity of $O(\frac{n}{k}^k)$ in the worst case, while Algorithm 2 operates at a time complexity of $O(k^3 n \log n)$. The comparison of the execution times for these algorithms is presented in Table 1, Fig. 4 and Fig. 5. Table 1 provides the execution times and the corresponding perimeters for both the exact algorithm (Exact) and the approximation algorithm (APX) across various values of n and k.

4. Conclusion

In this paper, the Colored Points TSP is defined as a variation of the TSP where points are colored with a set of distinct colors, and the objective is to find the minimum cycle to visit each color once. The NP-hardness of this problem is proven, and two exact and approximate algorithms are presented to solve this problem. The approximation factor is $\frac{2\pi r}{3u}$ for points on a grid of cell size u, where r is the radius of the smallest color-spanning circle of the points, and $\frac{\sqrt{2}\pi n}{3}$ when the points form a gapless arrangement, where n denotes the number of points. The presented algorithms have been implemented, and numerical results have been obtained on random datasets. The experimental validation confirms the efficiency of the presented approximation algorithm.

https://dx.doi.org/10.22108/toc.2025.140814.2153

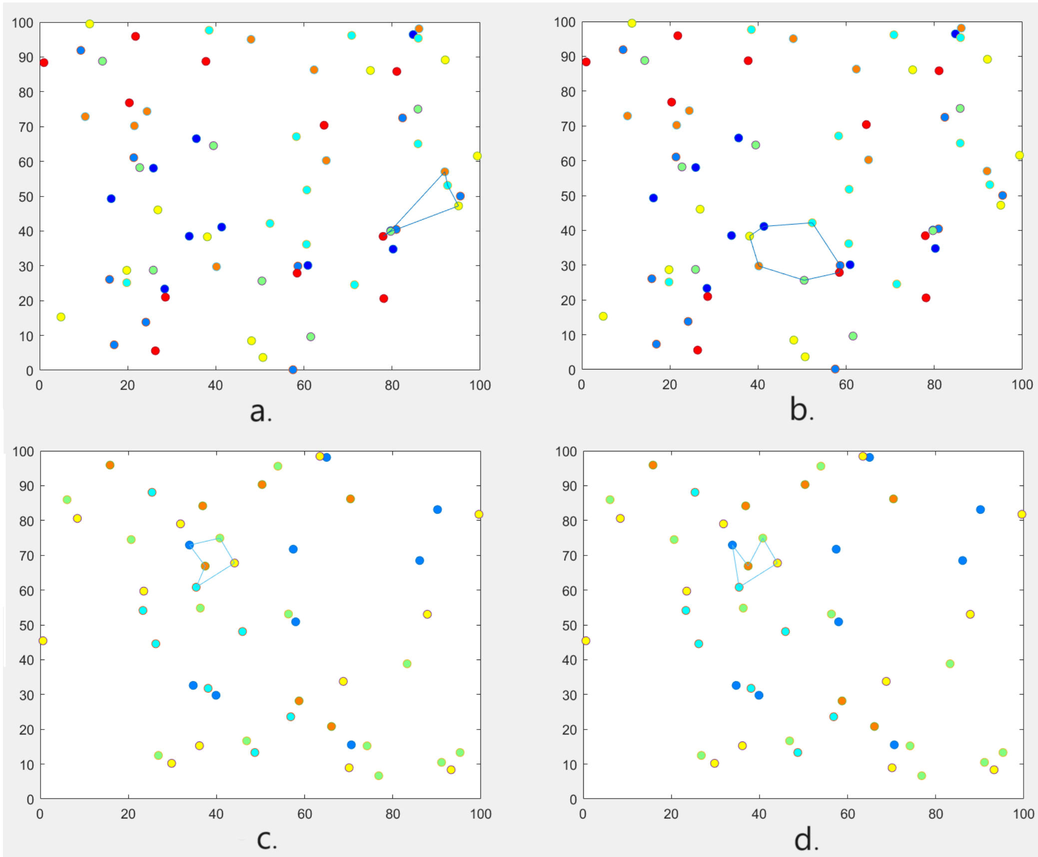

FIGURE 2. Figures (a) and (b) display the results of both the exact and approximate algorithms for the problem with $n = 70$ and $k = 7$, while figures (c) and (d) show the results for the problem with $n = 50$ and $k = 5$.

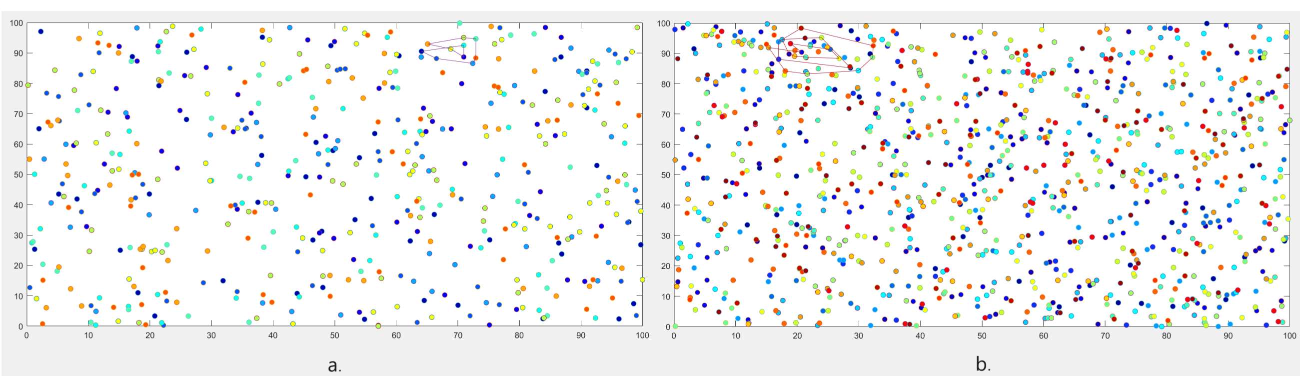

FIGURE 3. The results of the approximation algorithm are presented for (a) a set of 500 points with 10 distinct colors and (b) a set of 1000 points with 20 distinct colors.

https://dx.doi.org/10.22108/toc.2025.140814.2153

TABLE 1. Results of some samples for varying values of n and k

			$k=4$	$k=5$	$k=6$	$k=7$	$k=8$
$n=10$	Time	Exact	1.665056	0.815696	0.673525	0.191644	2.578316
		APX	0.269556	0.086585	0.074245	0.107769	0.087482
	Perimeter	Exact	134.9572	66.407	170.7678	162.7868	100.9232
		APX	134.9572	66.407	170.7678	162.7868	105.152
$n=15$	Time	Exact	3.242028	3.53374	4.085118	4.143144	5.221325
		APX	0.101243	0.089292	0.093642	0.091797	0.109786
	Perimeter	Exact	95.94496	125.3841	63.1264	217.3375	218.7068
		APX	95.94496	147.0142	63.1264	262.5696	251.6334
$n=20$	Time	Exact	7.94314	11.63913	11.13294	18.85075	12.58581
		APX	0.139876	0.12601	0.12412	0.125416	0.133383
	Perimeter	Exact	55.81232	124.1199	104.1315	112.4648	206.2783
		APX	78.39269	127.2698	158.5983	112.4648	288.5908
$n=25$	Time	Exact	14.82877	28.86919	62.3261	57.34854	84.27023
		APX	0.203443	0.174234	0.179336	0.157593	0.153333
	Perimeter	Exact	85.45303	93.87832	91.25952	131.4472	110.7979
		APX	101.6047	94.74419	105.7901	167.6262	132.0743
$n=30$	Time	Exact	30.30081	111.3685	168.3525	171.341	383.6654
		APX	0.377235	0.388495	0.362932	0.30269	0.329382
	Perimeter	Exact	43.08548	46.68444	71.69952	116.0164	186.5542
		APX	45.94552	68.19333	111.9154	187.6612	233.8898
$n=35$	Time	Exact	73.85159	58.34555	455.2198	589.9313	622.9517
		APX	0.539233	0.433459	0.534906	0.512461	0.433698
	Perimeter	Exact	36.97732	112.902	77.66447	85.63003	120.2107
		APX	36.97732	169.5268	132.1911	85.63003	193.1477
$n=40$	Time	Exact	106.644	387.588	580.1023	2345.168	3685.009
		APX	0.743369	0.719214	0.687392	0.651942	0.705154
	Perimeter	Exact	41.00075	38.47183	102.8376	103.4379	89.85302
		APX	52.00889	41.79108	125.3398	189.4559	152.3103

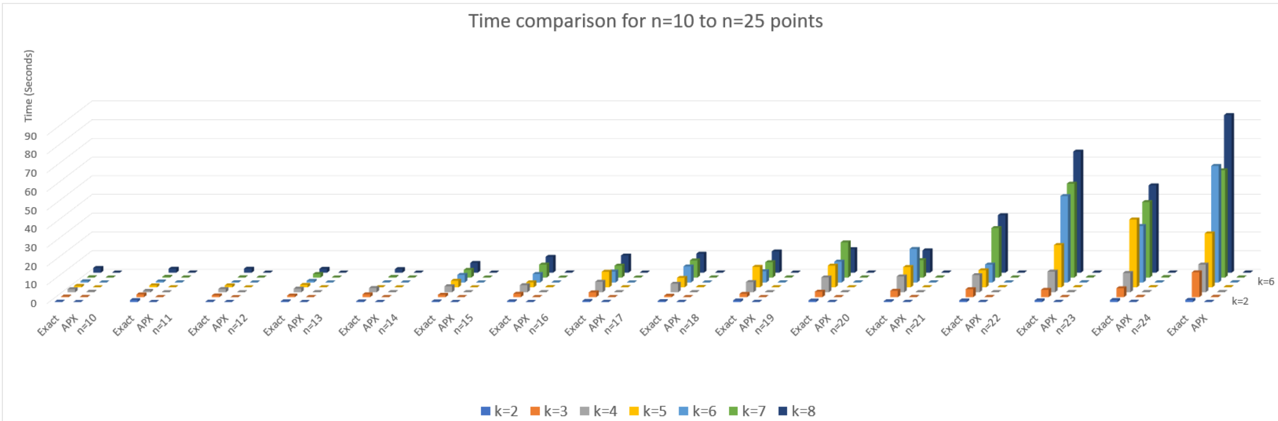

FIGURE 4. Comparison of the algorithms' execution time for n ranging from 10 to 25 points and k ranging from 2 to 8 colors.

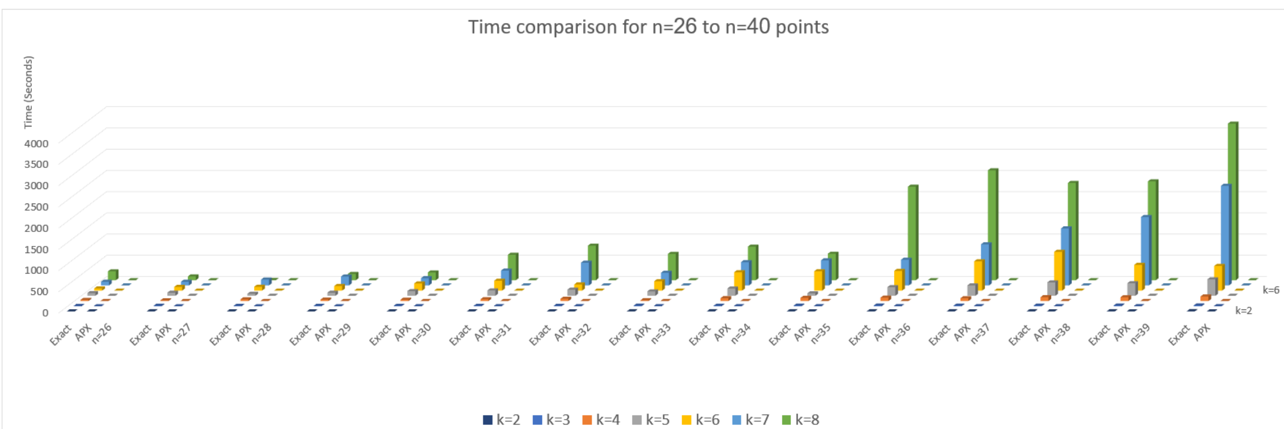

FIGURE 5. Comparison of the algorithms' execution time for n ranging from 26 to 40 points and k spanning from 2 to 8 colors.

REFERENCES

[1] A. Acharyya, R. K. Jallu, V. Keikha, M. Löffler and M. Saumell, Minimum color spanning circle of imprecise points, *Theoret. Comput. Sci.*, **930** (2022) 116–127.

[2] B. Couëtoux, L. Gourves, J. Monnot and O. A. Telelis, Labeled traveling salesman problems: complexity and approximation, *Discrete Optim.*, **7** no. 1-2 (2010) 74–85.

[3] C. H. Papadimitriou, The Euclidean traveling salesman problem is NP-complete, *Theoret. Comput. Sci.*, **4** no. 3 (1977) 237–244.

https://dx.doi.org/10.22108/toc.2025.140814.2153

[4] E. Tresoldi, R. Wolfler Calvo and S. Borne, Solving the multicolor tsp, in: *24th European Conference on Operational Research (EURO XXIV)*, Lisbon, Portugal, (2010).

[5] G. Laporte, The traveling salesman problem: An overview of exact and approximate algorithms, *European J. Oper. Res.*, **59** no. 2 (1992) 231–247.

[6] J. J. Sylvester, A question in the geometry of situation, *Q. J. Pure Appl. Math.*, **1** no. 1 (1857) 79–80.

[7] J. Li, M. Zhou, Q. Sun, X. Dai and X. Yu, Colored traveling salesman problem, *IEEE transactions on cybernetics*, **45** no. 11 (2014) 2390–2401.

[8] J. O'Rourke, *Computational geometry in C*, Second edition. Cambridge University Press, Cambridge, 1998.

[9] M. Abellanas, F. Hurtado, C. Icking, R. Klein, E. Langetepe, L. Ma, B. Palop and V. Sacristán, The farthest color voronoi diagram and related problems, in: *Abstracts 17th European Workshop Comput. Geom*, Freie Universität, Berlin, (2001) 113–116.

[10] M. Abellanas, F. Hurtado, C. Icking, R. Klein, E. Langetepe, L. Ma, B. Palop and V. Sacristán, Smallest color-spanning objects, in: *Algorithms—ESA 2001: 9th Annual European Symposium Århus*, Denmark, August 28–31, 2001 Proceedings 9, Springer, (2001) 278–289.

[11] M. E. Harvey, R. T. Hocking and J. R. Brown, The chromatic traveling-salesmen problem and its application to planning and structuring geographic space, *Geographical Analysis*, **6** no. 1 (1974) 33–52.

[12] N. Megiddo, Linear-time algorithms for linear programming in $\mathbf{R}^3$ and related problems, *SIAM J. Comput.*, **12** no. 4 (1983) 759–776.

[13] X. Meng, J. Li and M. Zhou, A colored traveling salesman problem with varying city colors, *Discrete Dynamics in Nature and Society* **2021** (2021) 1–14.

[14] Y. Xiong, B. Golden and E. Wasil, The colorful traveling salesman problem, *Extending the Horizons: Advances in Computing, Optimization, and Decision Technologies*, Chapter, (2007) 115–123.

Saeed Asaeedi

Department of Computer Science, Faculty of Mathematical Sciences, University of Kashan, P.O.Box 87317-53153, Kashan, I. R. Iran

Email: asaeedi@kashanu.ac.ir

https://dx.doi.org/10.22108/toc.2025.140814.2153