EAI Endorsed Transactions
on AI and Robotics
Research Article EAI.EU

Efficiently Guiding K-Robots Along Pathways with Minimal Turns

Hamid Hoorfar[1,*], Nedasadat Taheri[2], Houman Kosarirad[3], Alireza Bagheri[4]

[1]Department of Epidemiology and Public Health, University of Maryland, 10 S Pine St., Baltimore, MD 21201, USA.
[2]School of Computing, University of Nebraska–Lincoln, 256 Avery Hall, 1144 T, Lincoln, NE, 68588, USA.
[3]Durham School of Architectural Engineering and Construction, University of Nebraska-Lincoln, 122 NH, Lincoln, NE, 68588, USA.
[4]Department of Computer Engineering, Amirkabir University of Technology, 424 Hafez St., Tehran, Iran.

Abstract

This paper addresses the navigation of a team of k protector robots within pathways, focusing on minimizing the total number of turns. These robots utilize orthogonal routes known as watchman routes, which prioritize finding the shortest path while maintaining visibility of all points in the environment from at least one robot on its designated route. The main objective of this research is to optimize robot navigation by reducing the overall number of turns. The primary objective of this study is to develop a linear-time algorithm that efficiently processes and determines routes for k robots within a specified area. By minimizing the number of turns, this algorithm aims to enhance the navigation capabilities of watchman robots, enabling them to effectively traverse complex environments. This research employs techniques derived from computational geometry to investigate the navigation of protector robots. The focus is on developing an algorithm that can efficiently process and determine the optimal routes for the robots, considering factors such as visibility and shortest path length. The algorithm is designed to minimize the number of turns while ensuring efficient coverage of the environment. The main results of this paper include the development of a linear-time algorithm for determining routes for a team of k protector robots. The algorithm efficiently processes the input data and produces separate routes for each robot. By minimizing the number of turns, the algorithm improves the overall navigation efficiency of the robots. The results demonstrate the effectiveness of the algorithm in optimizing robot paths and reducing the complexity of navigation in real-world scenarios. In conclusion, this research contributes to the field of robotic systems by addressing the navigation challenges faced by a team of protector robots. The introduced linear-time algorithm optimizes the routes for k robots, aiming to minimize the total number of turns. The outcomes of this study have significant implications for the advancement of watchman robots, enhancing their coverage and surveillance capabilities in real-world applications. The algorithm's efficiency and effectiveness in minimizing turns open new opportunities for developing efficient navigation strategies in complex environments.

Received on 27 June 2023; accepted on 08 July 2023; published on 12 July 2023

Keywords: robot navigation, watchman robot, guarding, algorithm, motion planning

doi:10.4108/airo.3492

1. Introduction

In recent years, algorithms have gained paramount significance, expanding their influence and becoming ingrained in various aspects of our lives [1, 2]. They shape our daily routines and significantly impact decision-making processes [3, 4], system control [5], artificial intelligence [6, 7], sampling methods [8, 9], motion planning, robot navigation [10, 11], graph theory[16–19], social networks [13, 14], big data [15] and especially robotics [20–24]. Robot navigation is a prime example of how algorithms have become

*Corresponding author. Email: hhoorfar@som.umaryland.edu

indispensable components of our modern existence. This paper focuses specifically on the intricacies and significance of robot navigation. Orthogonal watchman robot navigation plays a crucial role in various applications, including surveillance, monitoring, and security systems. Efficiently traversing a group of robots within a given environment is essential to ensure comprehensive coverage and effective monitoring. One robot alone cannot efficiently achieve this task. One key aspect of optimizing robot navigation is minimizing the number of turns or bends in the robot's path. Reducing turns not only improves movement efficiency but also minimizes the time required to complete tasks. This paper addresses the challenge of minimizing turns in orthogonal watchman robot navigation for a team of k robots. The goal is to develop effective strategies and propose practical solutions that significantly reduce the number of bends in the robots' trajectories while maintaining complete visibility of the environment. By minimizing turns, the objective is to improve the overall effectiveness, scope, and monitoring capacities of watchman robots. The Watchman Route Problem is an optimization problem in computational geometry that involves finding the shortest route for a watchman or robot to guard an entire area, given only the layout of the area. The problem is typically defined using a simple polygon to represent the area, and the objective is to find the shortest closed curve such that all points within the polygon and on its boundary are visible to at least one point on the curve. There are two different variants of this problem: the fixed variant, where the watchman passes through a specified boundary point, and the float variant, where there is no predetermined starting point for the watchman.

1.1. Background

Previous research has investigated various algorithms to address the Watchman Route Problem. Chin and Ntafos [25] developed an algorithm with a time complexity of $O(n^4)$ for the fixed variant, while Xuehou Tan [26] introduced an incremental solution with a time complexity of $O(n^3)$ that constructs the shortest route. For the float variant, Tan [27] and later Sagheer [28] contributed a solution with a worst-case time complexity of $O(n^5)$. Tan also proposed a linear-time 2-approximation algorithm for the fixed watchman route. In practical applications, watchmen are often implemented as robots, which have motion restrictions due to their structural limitations. Orthogonal robots, capable of movement in two perpendicular directions, offer cost-effective solutions and are commonly employed. Consequently, minimizing the number of bends in the route becomes crucial, as additional bends lead to increased costs. In the orthogonal variant of the Watchman Route Navigation, the area under consideration is an orthogonal polygon, and the shortest route must consist of an orthogonal polygonal curve with the fewest possible bends. Extensive research has been conducted in the field of guarding and securing orthogonal polygons [12, 29–33], considering the prevalence of real-world scenarios that can be effectively represented using orthogonal polygonal environments [34–38]. These research efforts have significantly contributed to the development of more efficient solutions for such applications. This paper proposes a linear-time algorithm to solve the float watchman route problem for pathways in collaboration with a k-member team of robots. Our algorithm achieves an exact linear-time solution and addresses the challenge of minimizing turns in orthogonal watchman robot navigation by presenting tailored strategies and solutions for the Watchman Route Problem. Through the reduction of bends, the goal is to optimize the efficiency and coverage capabilities of k watchman robots in real-world scenarios.

1.2. Motivation

The motivation behind the problem of navigating a team of k robots in orthogonal pathways with minimal turns is rooted in practical applications that require efficient and effective robot navigation. By utilizing multiple robots, the coverage and surveillance capabilities can be significantly enhanced compared to using a single robot. The involvement of k robots allows for comprehensive monitoring of the environment, ensuring that all critical areas are adequately observed. Minimizing the number of turns in the robots' paths is crucial for several reasons. Firstly, reducing turns improves the overall efficiency of the robots' movement, enabling them to navigate through pathways more swiftly and effectively. This efficiency is particularly important in time-sensitive tasks where rapid response and completion are desired. Secondly, minimizing turns helps to optimize the utilization of resources and energy. Each turn or bend in a robot's trajectory requires additional time, energy, and potentially introduces mechanical wear and tear. By reducing the number of turns, the robots can conserve their resources, leading to prolonged operation times and decreased maintenance requirements. Thirdly, minimizing turns in the robots' paths enhances their maneuverability and flexibility. In complex environments with narrow pathways or obstacles, excessive turns can limit the robots' ability to navigate smoothly. By reducing turns, the robots can navigate more efficiently, maneuvering around obstacles and reaching their destinations more effectively. By employing a team of k robots, the problem of minimizing turns in their paths becomes even more relevant. Each robot's trajectory must be optimized to minimize turns while ensuring comprehensive coverage of the environment. Coordinating the paths of multiple robots introduces additional complexity and challenges, such as avoiding collisions and optimizing their routes collectively. Therefore, the use of k robots in orthogonal pathways necessitates addressing the problem of minimizing turns. By developing strategies, algorithms, and solutions specifically tailored for k robot navigation, researchers and practitioners can unlock the potential for more efficient, flexible, and collaborative robot systems, contributing to advancements in various domains, including surveillance, monitoring, security systems, and other real-world applications.

2. Preliminaries

In the field of robot navigation, the use of geometric objects, particularly polygons, has proven to be highly beneficial for modeling the surrounding environment. Among polygons, orthogonal ones offer a simplified $2D$ representation that greatly aids in the development of navigation algorithms, particularly in urban settings. By leveraging the concept of Minkowski summation, which eliminates the influence of dimensionality, robots can be treated as points or other geometric objects, regardless of their dimensional properties. This paper focuses on an environment characterized by an orthogonal path polygon denoted as P. An orthogonal polygon is referred to as a "path polygon" when the dual graph of its vertical decomposition (specifically, not the general decomposition) forms a path. The process of extending the vertical edges connected to the reflex vertices of P leads to the decomposition of the polygon into rectangles. This decomposition, known as the vertical decomposition, results in $\frac{n-2}{2}$ rectangles, where n represents the number of vertices. Let's denote this set of rectangles as $R = R_1, R_2, ..., R_m$, ordered from left to right based on the x-coordinate of their left edges when P is horizontal.

The upper and lower horizontal edges of R_i are denoted as u_i and l_i respectively. Additionally, $U = u_1, u_2, ..., u_m$ and $L = l_1, l_2, ..., l_m$ represent lists of upper and lower horizontal edges respectively, where $1 \leq i \leq m = \frac{n-2}{2}$.

For a horizontal line segment s, $x(s)$ represents the x-coordinate of the left vertex of s, while $y(s)$ denotes the y-coordinate of the line segment. Notably, for all $1 \leq i \leq m-1$, it holds true that $y(u_i) = y(u_{i+1})$ or $y(l_i) = y(l_{i+1})$.

Expanding on this observation, $e(u_i)$ is defined as the edge of P that contains u_i, and $e(l_i)$ as the edge that contains l_i.

To arrange these edges in a left-to-right manner, two sets are introduced:

$$EU = \{e(u_i) | 1 \leq i \leq m\} \tag{1}$$

$$EL = \{e(l_i) | 1 \leq i \leq m\} \tag{2}$$

. Within the list of horizontal edges E of P, an edge e_j is classified as a local maximum if $y(e_j) \geq y(e_{j-1})$ and $y(e_j) \geq y(e_{j+1})$. Similarly, an edge e_k is considered a local minimum if $y(e_k) \leq y(e_{k-1})$ and $y(e_k) \leq y(e_{k+1})$. The edge $e(u_m)$ (or $e(l_m)$) is designated as a local maximum if u_m (or l_m) represents a local maximum. Likewise, u_n (or l_n) is termed a local minimum if $e(u_n)$ (or $e(l_n)$) is a local minimum. Among the set R, R_1 is specifically identified as a local maximum if u_1 and l_1 represent a local maximum and a local minimum, respectively. The concept of weak visibility is introduced for two axis-parallel segments, l and k, which are considered weakly visible if an axis-parallel line segment can be drawn from a point on l to a point on k without intersecting the boundary of P. Covering an ortho-convex polygon P can be achieved by passing through a point located within its kernel. The kernel refers to a region that allows for a comprehensive view of the entire shape. An ortho-convex polygon is defined as a polygon that maintains vertical and horizontal monotonicity, ensuring that its edges follow a consistent ascending or descending order along both axes.

3. Balanced sub-polygon identification algorithm

Our objective is to determine k orthogonal routes with the least number of bends in total while ensuring that every point within P remains visible along one of these k routes. Put simply, the routes should provide unobstructed visibility of both the interior and boundary of P. An orthogonal polygon is considered x-monotone if it lacks any dent edges in the directions perpendicular to the y-axis. A balanced polygon, a subtype of x-monotone orthogonal polygons, can be traversed using an orthogonal route represented by a horizontal line segment. It possesses a distinctive horizontal line segment called the "align" segment, which connects the leftmost and rightmost vertical edges without intersecting any other edges. It is important to note that multiple align-segments can exist within a balanced polygon. Additionally, each balanced polygon contains an orthogonal corridor that extends from the leftmost to the rightmost edge, encompassing the align-segment. Ortho-convex polygons also fall under the category of balanced polygons. Conversely, an orthogonal polygon that fails to meet the criteria of a balanced polygon is referred to as an "unbalanced" polygon. This paper presents an algorithm for decomposing x-monotone orthogonal polygons into balanced sub-polygons, relying on the identification of corridors within the polygon. Let ϵ denote the leftmost vertical edge of polygon P. The algorithm for identifying balanced sub-polygons within polygon P begins by starting at the leftmost vertical edge, ϵ. From ϵ, a light beam is projected perpendicular to ϵ while remaining collinear with the X-axis. The rectilinear path of the light beam intersects a subset of rectangles, denoted as R, obtained from the vertical decomposition of P. This subset, referred to as R_ρ, collectively forms a sub-polygon ρ, representing the first balanced part of the polygon. Within sub-polygon ρ, each rectangle R_i has upper and lower horizontal edges denoted as u_i and l_i respectively. It can be established that:

$$\min_{u_i \in P}(y(u_i)) \geq \max_{l_j \in P}(y(l_j)) \tag{3}$$

The existence of a horizontal line segment σ connecting the leftmost and rightmost vertical edges of sub-polygon ρ implies that σ is an integral part of the balanced structure of ρ. By utilizing the align-segment technique, it becomes possible to determine an optimal orthogonal route for a balanced x-monotone polygon that minimizes its length.

3.1. Decomposing Polygon P into Balanced Sub-polygons

To decompose polygon P into multiple balanced x-monotone polygons, Subsequently, sub-polygon ρ is eliminated from P, and the aforementioned steps are reiterated. By iterating through this process, A linear-time algorithm for decomposing P into balanced sub-polygons is obtained as described in the Algorithm 1. In Algorithm 1, when the condition in line 3 is satisfied, it indicates the presence of a balanced sub-polygon ρ. The next step involves the removal of ρ from polygon P and the iterative application of the algorithm to the remaining portion of P, referred to as $P - \rho$. The rectangles belonging to ρ are removed from the set R, resulting in an updated set $R = R - R_\rho$. The elements in R are initially ordered and labeled from left to right. Upon removing ρ, the remaining elements are relabeled, commencing from 1. The same actions are performed on the sets U and L. The number of iterations in this process is equal to the size of R at the beginning, ensuring a linear time complexity for decomposing P into balanced polygons. Each balanced polygon ρ has an align line-segment σ connecting its leftmost and rightmost edges. This align-segment provides a comprehensive view of ρ from at least one point on σ, making ρ weakly visible from σ. Hence, σ emerges as a viable candidate for the orthogonal watchman route problem within ρ. When P is decomposed into balanced sub-polygons $\rho_1, \rho_2, \ldots, \rho_k$ with corresponding align line segments $\sigma_1, \sigma_2, \ldots, \sigma_k$, it becomes possible to connect these align line segments using $k - 1$ vertical line segments. These vertical line segments establish a connection between the right endpoint of σ_i and the left endpoint of σ_{i+1} for each i ranging from 1 to $k - 1$. This resulting orthogonal path, denoted as Π, represents the primary solution for the orthogonal floating watchman route problem in P.

Algorithm 1 Decomposition P into the balanced sub-polygons

Require: an x-monotone polygon with n vertices.
Ensure: a list of balanced sub-polygon.
(1) Set $j = 0$, $min_u = u_1$ and $max_l = l_1$.
while *there is rectangle R_i belongs to R* **do**
  if *$u_i > max_l$ or $l_i < min_u$* **then**
    remove R_1, ... , R_{i-1} from R .
    $P_j = R_1$, ... , R_{i-1}.
    $j = j + 1$.
    refresh the index of R starting with 1.
    go to 1.
  end
  Compute min_u =min(min_u, u_i) and max_l = max(max_l, l_i).
end
return $last$ = number of sub-polygons.

3.2. Adjusting the Route for Minimum Orthogonal Path

However, some unnecessary portions can be trimmed from the beginning and end of Π to obtain the shortest possible orthogonal route. The watchman can ignore these trimmed parts and begin guarding from the leftmost point of the kernel of P_1 intersecting σ_1 to the rightmost point of the kernel of P_{last} intersecting σ_{last}. In order to obtain the shortest orthogonal route while preserving the structure and properties of Π, Algorithm 2 is introduced. This algorithm operates in linear time and effectively trims Π. This algorithm focuses on removing unnecessary portions from Π to optimize the route length.

In most scenarios, the algorithm operates in constant time. However, in the worst-case scenario, its complexity reaches $O(n)$. Nevertheless, the resulting route requires further adjustments to achieve the minimum possible orthogonal path. These modifications primarily focus on optimizing the lengths of the vertical line segments that connect consecutive align segments. Let $\Sigma = \{\sigma_1, \sigma_2, ..., \sigma_k\}$ represent the align segments, and $y(\sigma_i)$ denote the y-coordinate of σ_i for each i between 1 and k. Each sub-polygon P_i has its align segment, and within the corridor of P_i, there can be multiple horizontal line segments that could serve as its align segment. However, certain align segments are more advantageous than others, as they contribute to a shorter overall route. To determine the favorable align segments, the y-coordinate of an align segment is evaluated in relation to two parameters of the balanced sub-polygon P_i.

Algorithm 2 Trimming path π to the shortest orthogonal route

Require: an orthogonal path polygon π.
Ensure: the shortest route in π.
Set $j = 0$, $min_u = u_1$ and $max_l = l_1$.
while *there is rectangle* R_i *belongs to* R, *from* $i = 1$ *to* m **do**
  if u_i *is a local maximum or* l_i *is a local minimum* **then**
    remove $R_1, \dots, R_i$ from R.
    go to 1.
  end
end
(1) **while** *there is rectangle* R_i *belongs to* R, *from* $i = m$ *down to* 1 **do**
  if u_i *is a local maximum or* l_i *is a local minimum* **then**
    remove $R_1, \dots, R_i$ from R.
    go to 2.
  end
end
(2) return $\pi = \pi \cap R$.

These parameters are denoted as follows:

$$M_i = \min_{u_j \in R_{P_i}} y(u_j) \tag{4}$$

and

$$m_i = \max_{l_j \in R_{P_i}} y(l_j) \tag{5}$$

3.3. Optimizing the Vertical Line Segments for Consecutive Align Segments

Considering each i from 1 to k, the following conditions are taken into consideration: $M_i < m_{i+1}$ or $m_i > M_{i+1}$. In cases where $M_{i-1} < M_i < M_{i+1}$ or $M_{i-1} > M_i > M_{i+1}$, the selection of $y(\sigma_i)$ (whether it equals M_i or m_i) does not impact the total length of the vertical line segments connecting consecutive align segments σ_{i-1} σ_i, and σ_i σ_{i+1}. The length remains constant in such instances. In the scenario where $M_{i-1} < M_i > M_{i+1}$ or $M_{i-1} > M_i < M_{i+1}$, the total length of the orthogonal route varies. In the former case, the align segment with $y(\sigma_i) = m_i$ is chosen, while in the latter case, the align segment with $y(\sigma_i) = M_i$ is selected. This selection minimizes the route length by reducing the total length of v_{i-1} and v_i. Here, v_{i-1} represents the vertical line segment connecting the right endpoint of σ_{i-1} to the left endpoint of σ_i, and v_i denotes the vertical line segment connecting the right endpoint of σ_i to the left endpoint of σ_{i+1}. For the initial balanced sub-polygon, if $M_1 < M_2$, the align segment with $y(\sigma_1) = M_1$ is chosen. Conversely, if $M_1 > M_2$, the align segment with $y(\sigma_1) = m_1$ is selected. Similarly, for the last sub-polygon P_k, if $M_k < M_{k-1}$, the align segment with $y(\sigma_k) = M_k$ is chosen. Conversely, if $M_k > M_{k-1}$, the align segment with $y(\sigma_k) = m_k$ is selected. These choices effectively lead to a shorter route by minimizing the total length of the vertical line segments v_{i-1} and v_i. Through optimizing the selection of align segments, an improved route can be achieved. For visual reference, please refer to 1. We introduce Algorithm 3, a linear-time algorithm designed to select the most suitable align segments for each balanced sub-polygon, with the objective of obtaining the shortest orthogonal route possible. The pseudo code for this algorithm is Algorithm 3.

Algorithm 3 Align Selection

Require: a set of available aligns.
Ensure: a set of appropriate aligns.
while *there is sub-polygon* P_1 **do**
  if $M_1 < M_2$ **then**
    set $y(\sigma_1) = M_1$.
  else
    set $y(\sigma_1) = m_1$.
  end
end
while *sub-polygon* $P_i i$ *belongs to* P, *from* $i = 1$ *to* m **do**
  if $M_{i-1} < M_i$ *and* $M_i > M_{i+1}$ **then**
    set $y(\sigma_i) = m_i$.
  else
    set $y(\sigma_i) = M_i$.
  end
end
while *there is sub-polygon* P_m **do**
  if $M_m < M_{m-1}$ **then**
    set $y(\sigma_m) = M_m$.
  else
    set $y(\sigma_m) = m_m$.
  end
end

3.4. Time Complexity

We have introduced an algorithm that effectively discovers the shortest orthogonal path while minimizing the number of bends. The algorithm is structured into three distinct sections, each of which has been accompanied by comprehensive explanations. To attain the desired outcome, it is crucial to execute all three sections sequentially. Now, let's delve into the algorithm's complexity. In the worst-case scenario, our algorithm exhibits a time complexity of $O(n)$. This indicates that the running time of the algorithm increases linearly with the number of edges present in the polygon. The efficient time complexity of the algorithm has been accomplished by meticulously designing it to process each section in a linear fashion. As a result, the overall time complexity of the complete algorithm remains linear as well. This linear time complexity is considered optimal since, at the very least, examining every vertex of the polygon is necessary to compute an orthogonal route. Thus, our algorithm is highly optimized, and it is unlikely that a better time complexity can be attained. Moreover, the space complexity of the algorithm is $O(n)$, indicating that the amount of memory required by the algorithm also grows linearly with the size of the input.

4. Partitioning the Single Optimal Route into k Separate Routes for k Robots

In the previous section, an algorithm was discussed to determine a single optimal orthogonal route encompassing all the routes for k robots. The current objective is to partition this single route into k separate routes, allowing each robot to utilize a designated portion of the pathway. The aim is to minimize the total number of bends on these individual routes.

Let R represent the single route obtained from the previous section, which consists of t turns. For a balanced distribution of turns among the k robots, each route should ideally have an average of $\lfloor t/k \rfloor$ turns. In R, the vertical segments of the route pass through rectangles that serve as connectivity points along the path. Given the presence of k robots, certain rectangles are no longer necessary for connecting the paths of the robots. Once an adjacent robot on its respective route reaches such a rectangle, the mission of covering that specific rectangle is achieved, even without intersecting the vertical line passing through it. Consequently, these rectangles can be disregarded in subsequent routes.

Additionally, Algorithm 2, introduced earlier, can be employed to trim the path of the neighboring robot and reduce its route length. This step contributes to the overall efficiency and optimization of each robot's route. By partitioning the single optimal route into k separate routes, each robot is assigned a specific section of the pathway. This approach minimizes the total number of bends and ensures that the entire route is covered collectively by the k robots. Through this strategy, a certain number of turns can be disregarded, resulting in the total number of turns in all robot routes being equal to or less than t.

This process involves decomposing the single route, eliminating unnecessary rectangles, and employing trimming techniques to optimize the routes for each individual robot. See Algorithm 4

The algorithm aims to partition the single optimal route R into k separate routes, each designated for a robot. It begins by initializing an empty list called robot-routes to store the separate routes. The average number of turns per route is calculated to ensure an approximately equal distribution of turns among the robots. Next, the algorithm iteratively selects vertical line-segments from the leftmost end of the route R and divides it into separate routes. Each route, denoted as R_i, is optimized using trimming techniques (Algorithm 2) to reduce its length and eliminate unnecessary portions. The optimized routes, R_i, are added to the robot-routes list, and the corresponding processed route is removed from R. This process continues until all parts of the original route R have been partitioned and optimized. Finally, the algorithm returns the robot-routes list, which contains k separate routes. Each route is tailored for a robot, ensuring minimized bends and optimized length. Note: The reference to Algorithm 2 pertains to the trimming algorithm mentioned earlier. Algorithm 2 focuses on removing unnecessary portions from the route while preserving its overall structure and properties.

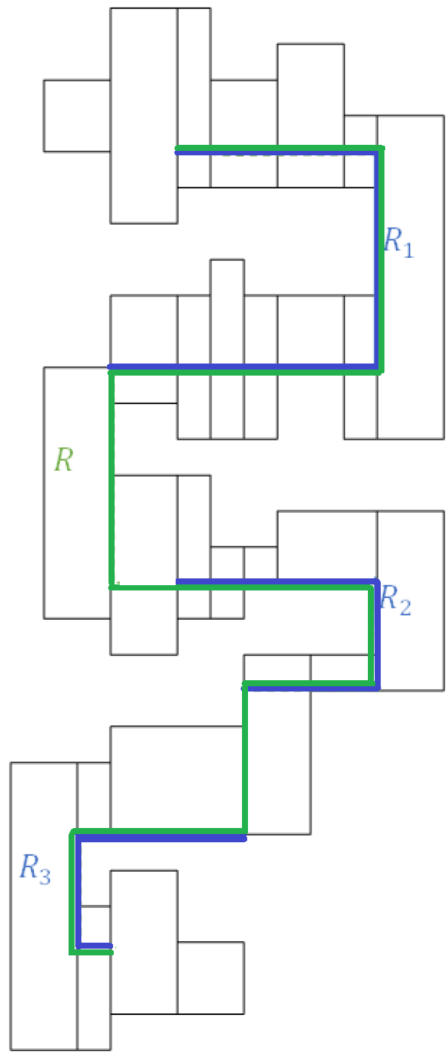

Figure 1. Partitioning the Single Optimal Route into $k = 3$ Separate Routes.

Algorithm 4 finding k routes for k robots with the optimal total number of turns.

Require: Route R (with t turns), Number of robots k.
Ensure: k separate routes for the robots.
Initialize an empty list denoted as robot-routes.
Calculate the average number of turns per route, a = avg-turns-per-route, by integer dividing t by k.
while *R is not empty* **do**
    Find the ath vertical line-segment of R from the leftmost end-point.
    Call the part of R before the ath vertical line-segment as a route R_1.
    Use Algorithm 2 on R_1 to trim the route.
    Add R_1 to the robot-routes list.
    Remove R_1 from R.
end
Return the robot-routes list containing k separate routes for the robots.

4.1. Time complexity

The time complexity of the algorithm 4 can be analyzed as follows:
Initialization: Initializing an empty list of k routes and calculating the average number of turns per route can be done in constant time, $O(1)$.
While loop: The loop iterates until route R is empty. The number of iterations depends on the size of route R and the number of robots k. (1) Finding the ath vertical line-segment of R from the leftmost endpoint can be done in linear time, $O(n)$, where n is the number of line-segments in the route R. (2) Calling Algorithm 2 on R_i to trim the route takes time proportional to the size of R_i. (3) Adding R_i to the robots-routes list and removing R_i from R can be done in constant time, $O(1)$. The overall time complexity of the operations within the while loop depends on the time complexity of Algorithm 2 .
Return: Returning the robot-routes list containing k separate routes is a constant-time operation, $O(1)$.

Therefore, the time complexity of the algorithm primarily relies on the time complexity of Algorithm 2 and the size of the input route R. The overall time complexity can be approximated as $O(n \times T)$, where n represents the number of line-segments in the route R and T denotes the time complexity of Algorithm 2. Since Algorithm 2 processes each part of R once, its total time complexity over all parts is $O(n)$. Consequently, if Algorithm 2 operates in linear time or better, the overall time complexity of the algorithm can be considered linear, specifically $O(n)$.

5. Conclusion

In conclusion, this paper explores the navigation of a team of k protector robots along orthogonal watchman routes, aiming to minimize the total number of turns. An algorithm is provided that operates in linear time and linear space of the input size to find k routes for k orthogonal robots. This algorithm efficiently processes and determines separate routes for each robot, resulting in significant advancements in optimizing their navigation within complex environments. The introduced algorithm successfully partitions a single optimal route into k separate routes, tailored for each robot, while considering visibility requirements. By applying trimming techniques, unnecessary portions are eliminated, reducing bends and improving efficiency. With its linear time complexity, the algorithm can be applied in real-time scenarios, enhancing the overall capabilities of watchman robots. The outcomes of this research have significant implications for the field of robotic systems, particularly in the domain of watchman robots. By reducing the number of turns, the coverage and surveillance capabilities of these robots are greatly enhanced, making them highly effective in real-world applications. The contributions of this paper advance the development of efficient and versatile robotic systems, enabling their deployment in various domains such as security, monitoring, and exploration. By minimizing the number of turns, the proposed algorithm contributes to the design of more efficient and effective watchman robots, furthering advancements in robotic navigation and expanding the scope of their applications. In summary, the findings presented in this paper demonstrate the importance of optimizing navigation by minimizing turns for watchman robots. The developed algorithm offers a practical solution to this problem, facilitating the advancement of robotic systems and paving the way for future research and development in this field.

Future works in this area could focus on enhancing the algorithm to handle dynamic environments and obstacles, allowing the watchman robots to adapt and navigate in real-time. Additionally, investigating methods to optimize the routes further while considering energy consumption and battery life would be valuable. Exploring the integration of advanced sensing and perception technologies could also improve the overall navigation capabilities and decision-making of the watchman robots. These avenues of research will contribute to the continuous advancement of robotic navigation and its applications in various domains.

References

[1] Sadeghi M, Andani ME, Parnianpour M, Fattah A. *A bio-inspired modular hierarchical structure to plan the sit-to-stand transfer under varying environmental conditions*. Neurocomputing. 2013 Oct 22;118:311-21.

[2] Moshayedi AJ, Hosseinzadeh M, Joshi BP, Emadi Andani M. *Recognition System for Ergonomic Mattress and Pillow: Design and Fabrication*. IETE Journal of Research. 2023 Jan 10:1-9.

[3] Shahparvari S, Hassanizadeh B, Mohammadi A, Kiani B, Lau KH, Chhetri P, Abbasi B. *A decision support system for prioritised COVID-19 two-dosage vaccination allocation and distribution*. Transportation Research Part E: Logistics and Transportation Review. 2022 Mar 1;159:102598.

[4] Maydanchi M, Ziaei A, Basiri M, Azad AN, Pouya S, Ziaei M, Haji F, Sargolzaei S. *Comparative Study of Decision Tree, AdaBoost, Random Forest, Naïve Bayes, KNN, and Perceptron for Heart Disease Prediction*. InSoutheastCon 2023 2023 Apr 1 (pp. 204-208). IEEE.

[5] Toragay O, Silva DF. *Fast heuristic approach for control of complex authentication systems*. Applied Stochastic Models in Business and Industry. 2021 Jul;37(4):744-66.

[6] Moshayedi AJ, Roy AS, Taravet A, Liao L, Wu J, Gheisari M. *A secure traffic police remote sensing approach via a deep learning-based low-altitude vehicle speed detector through uavs in smart cites: Algorithm, implementation and evaluation*. Future Transportation. 2023 Feb 3;3(1):189-209.

[7] Liu J, Prabuwono AS, Abulfaraj AW, Miniaoui S, Taheri N. *Cognitive cloud framework for waste dumping analysis using deep learning vision computing in healthy environment*. Computers and Electrical Engineering. 2023 Sep 1;110:108814.

[8] Moshtaghi Largani S, Lee S. *Efficient Sampling for Big Provenance*. InCompanion Proceedings of the ACM Web Conference 2023 2023 Apr 30 (pp. 1508-1511).

[9] Kashgarani H, Kotthoff L. *Is algorithm selection worth it? Comparing selecting single algorithms and parallel execution*. InAAAI Workshop on Meta-Learning and MetaDL Challenge 2021 Aug 18 (pp. 58-64). PMLR.

[10] Hoorfar H, Fathi F, Moshtaghi Largani S, Bagheri A. *Securing Pathways with Orthogonal Robots*. The 21st International Conference on Scientific Computing; Las Vegas, USA2023.

[11] Hoorfar H, Moshtaghi Largani S, Rahimi R, Bagheri A. *Minimizing Turns in Watchman Robot Navigation: Strategies and Solutions*. The 21st International Conference on Scientific Computing; Las Vegas, USA2023.

[12] Hoorfar H, Bagheri A. *Minimum hidden guarding of histogram polygons*. arXiv preprint arXiv:1708.05815. 2017 Aug 19.

[13] Kazemzadeh F, Safaei AA, Mirzarezaee M, Afsharian S, Kosarirad H. *Determination of influential nodes based on the Communities' structure to maximize influence in social networks*. Neurocomputing. 2023 May 14;534:18-28.

[14] Liu Q, Kosarirad H, Meisami S Alnowibet KA Hoshyar AN. *An Optimal Scheduling Method in IoT-Fog-Cloud Network Using Combination of Aquila Optimizer and African Vultures Optimization*. Processes. 2023 Apr 10;11(4):1162.

[15] Kosarirad H, Ghasempour Nejati M, Saffari A, Khishe M, Mohammadi M. *Feature Selection and Training Multilayer Perceptron Neural Networks Using Grasshopper Optimization Algorithm for Design Optimal Classifier of Big Data Sonar*. Journal of Sensors. 2022 Nov 14;2022.

[16] Mahmudi F, Soleimani M, Naderi MH. *Some Properties of the Maximal Graph of a Commutative Ring*. Southeast Asian Bulletin of Mathematics. 2019 Jul 1;43(4).

[17] Moshayedi AJ, Khan AS, Shuxin Y, Kuan G, Jiandong H, Soleimani M, Razi A. *E-Nose design and structures from statistical analysis to application in robotic: a compressive review*. EAI Endorsed Transactions on AI and Robotics. 2023 Apr 20;2(1):e1-.

[18] Soleimani M, Mahmudi F, Naderi MH. *Some results on the maximal graph of commutative rings*. Advanced Studies: Euro-Tbilisi Mathematical Journal. 2023 Mar;16(supp1):21-6.

[19] Gaur A, Sharma A. *Maximal graph of a commutative ring*. Int. J. Algebra. 2013;7(12):581-8.

[20] Pourghorban A, Dorothy M, Shishika D, Von Moll A, Maity D. *Target Defense against Sequentially Arriving Intruders*. In2022 IEEE 61st Conference on Decision and Control (CDC) 2022 Dec 6 (pp. 6594-6601). IEEE.

[21] Pourghorban A, Maity D. *Target defense against periodically arriving intruders*. arXiv preprint arXiv:2303.05577. 2023 Mar 9.

[22] KhazeniFard A, Bahrami F, Andani ME, Ahmadabadi MN. *An energy efficient gait trajectory planning algorithm for a seven linked biped robot using movement elements*. In2015 23rd Iranian Conference on Electrical Engineering 2015 May 10 (pp. 1006-1011). IEEE.

[23] Moshayedi AJ, Roy AS, Sambo SK, Zhong Y, Liao L. *Review on: The service robot mathematical model*. EAI Endorsed Transactions on AI and Robotics. 2022 Feb 23;1(1).

[24] Moshayedi AJ, Reza KS, Khan AS, Nawaz A. *Integrating Virtual Reality and Robotic Operation System (ROS) for AGV Navigation*. EAI Endorsed Transactions on AI and Robotics. 2023 Apr 21;2(1):e3-.

[25] Chin WP, Ntafos S. *Shortest watchman routes in simple polygons*. Discrete and Computational Geometry. 1991 Mar, 6(1):9-31.

[26] Tan X, Hirata T, Inagaki Y. *An incremental algorithm for constructing shortest watchman routes*. International Journal of Computational Geometry and Applications. 1993 Dec;3(04):351-65.

[27] Tan X. *A linear-time 2-approximation algorithm for the watchman route problem for simple polygon*. Theoretical Computer Science. 2007 Sep 24;384(1):92-103.

[28] Chin WP, Ntafos S. *Optimum watchman routes*. In Proceedings of the second annual symposium on Computational geometry 1986 Aug 1 (pp. 24-33).

[29] Hoorfar H, Bagheri A. *NP-completeness of chromatic orthogonal art gallery problem*. The Journal of Supercomputing. 2021 Mar;77(3):3077-109.

[30] Biedl T, Mehrabi S. *On r-Guarding Thin Orthogonal Polygons*. arXiv preprint arXiv:1604.07100. 2016 Apr 25.

[31] Hoorfar H, Mohades A. *Special guards in chromatic art gallery*. 31st European Conference on Computational Geometry, 2015. 16

[32] Schuchardt D, Hecker HD. *Two NP-Hard Art-Gallery Problems for Ortho-Polygons*. Mathematical Logic Quarterly. 1995;41(2):261-7.

[33] Hoorfar H, Bagheri A. *Guarding Path Polygons with Orthogonal Visibility*. arXiv preprint arXiv:1709.01569. 2017 Sep 5.

[34] Hoorfar H, Bagheri A. *A linear-time algorithm for orthogonal watchman route problem with minimum bends*. arXiv preprint arXiv:1708.01461. 2017 Aug 4.

[35] Gewali LP. *Placing guards inside orthogonal polygons*. Information sciences. 1996 Jan 1;88(1-4):1-4.

[36] Hoorfar H, Bagheri A. *Geometric Embedding of Path and Cycle Graphs in Pseudo-convex Polygons*. arXiv preprint arXiv:1708.01457. 2017 Aug 4.

[37] Hoorfar H, Bagheri A. *A New Optimal Algorithm for Computing the Visibility Area of a simple Polygon from a Viewpoint*. arXiv preprint arXiv:1803.10184. 2018 Mar 27.

[38] Froncek D, Miller M, Kratochvíl J. *Combinatorial Algorithms*: 25th International Workshop, IWOCA 2014 Duluth, MN, USA, October 15-17, 2014 Revised Selected Papers. In25th International Workshop on Combinatorial Algorithms, IWOCA 2014 2015. Springer Verlag.