Multi-Agent Hybrid Approach for Environment Discovery

Marco Perini, *229203*

***Abstract*—In this paper, we introduce a multi-agent system designed for the exploration and mapping of unknown environments.**

Leveraging Voronoi tessellation and k-means clustering, we efficiently partition space and assign regions to agents. To ensure optimal navigation within these regions, a heuristic approach to the Traveling Salesman Problem (TSP) is adopted. The agents, modeled as differential drive robots and equipped with a comprehensive sensor suite, traverse their respective regions to scan and construct a map of the unknown environment. An Extended Kalman Filter (EKF) is implemented for state estimation and sensor fusion.

Through simulated experiments, we demonstrate the system's effectiveness and highlight its potential for real-world multi-agent robotic applications.

I. Introduction

As robotic systems grow in both complexity and capability, there emerges an increasing demand for efficient, scalable solutions that leverage the collective potential of multiple robots. This collaborative approach, often referred to as multi-agent systems, taps into a unique synergy where individual agents work together to achieve what would be insurmountable tasks for solo entities.

Multi-agent robotic systems have garnered significant attention in the research community and industry. Their potential to perform tasks more efficiently, be it in large scale agricultural automation, search and rescue operations, or space exploration, offers avenues that single-agent systems simply can't address. In many of these scenarios, the environment to be explored is unknown or dynamic. Mapping such an environment in real-time is a daunting task, further exacerbated by the inherent challenges of multi-agent coordination, path optimization, and data fusion.

One of the fundamental questions in deploying multi-agent systems for exploration revolves around the distribution of the environment among the agents. How do we ensure that each agent is provided with a section of the environment to scan, ensuring minimal overlap and redundancy? Furthermore, once these regions are designated, how do agents chart their paths in a manner that is both efficient and comprehensive?

This paper delves into these challenges, introducing an integrated approach to multi-agent exploration. We focus on space partitioning using Voronoi tessellation and k-means clustering, ensuring that each robot is given a unique, significant region to explore. Following this, the challenge of efficient navigation within these regions is addressed using a heuristic approach to the Traveling Salesman Problem (TSP). The agents, modeled as differential drive robots, come equipped with different kind of sensors, each offering different modalities of perception, ultimately aiding in a holistic understanding of the unknown environment. To cope with the uncertainties of these sensor readings, we implement an Extended Kalman Filter (EKF) for state estimation and data fusion.

The remainder of this paper presents a detailed account of our methods, the underlying principles driving them, and the experimental results that validate our approach. Through this research, we aim to advance the field of multi-agent exploration and offer a robust solution for real-world challenges.

II. Related Work

The K-Means algorithm has been thoroughly examined in the context of clustering, with Ahmed et al. presenting a comprehensive survey on its performance [1]. On the topic of route optimization, the Traveling Salesman Problem (TSP) stands out with significant contributions like that of Jünger, who dedicated an extensive chapter to its exploration [2], and Abdulkarim's comparison of TSP-solving algorithms [3]. In robot navigation, the role of LIDAR-based systems and path-planning algorithms have been highlighted in past studies, showcasing their effectiveness in complex navigation scenarios [4], [5].

III. Description of the project

This project aims to develop a comprehensive multi-agent robotic system, integrating various techniques and methodologies to enhance efficiency in environmental mapping tasks. Here's a brief rundown of the elements used:

- **Multi-agent Systems**: The foundation is built on deploying multiple robots, designed to collaborate and meet common goals. Using several robots offers advantages like covering larger terrains, achieving tasks quickly, and ensuring redundancy. This ensures that if one robot fails, the mission can still proceed with minimal interruption. The agents are located inside the environment using Ultra-Wideband (UWB) anchors and a central server makes all the needed calculation to provide the agents with an optimal path to follow.
- **Voronoi Tessellations**: This method is used for efficient spatial partitioning of the environment. It helps in dividing the area so that individual robots can effectively focus on their allocated regions, ensuring thorough scanning and mapping.
- **K-means Clustering**: For optimal task allocation and robot deployment, the K-means clustering algorithm is implemented. It helps by grouping adjacent Voronoi cells in regions and assigning robots to them, reducing overlap and maximizing efficiency.

- **Traveling Salesman Problem (TSP)**: A heuristic approach to the TSP is employed. It finds the shortest path that connects all the cells in a specific region, thus allowing the agent to efficiently scan the surroundings.
- **Differential Drive Robots**: Agents are defined with a differential drive dynamics which allows them to follow the trajectories using a proportional controller.
- **Sensors and Extended Kalman Filter (EKF)**: A range of sensors, including Encoders, Magnetometers, Lidars, and Stereo Cameras, are incorporated. The data from these sensors undergoes processing through the Extended Kalman Filter, refining the input by reducing noise and enhancing the accuracy of the environmental mapping.

In summary, the project combines these tools and techniques with the primary objective of achieving detailed and accurate environmental mapping using a team of robotic agents.

A. Environment

The environment is represented as a 2D polygonal shape (Figure 1), serving as the workspace for the robotic agents. Each corner of this polygon hosts a UWB (Ultra-Wideband) anchor. These UWB anchors, with their fixed and known positions, offer a reliable reference for the agents that allows them to achieve precise navigation within the environment, ensuring effective task execution with a clear spatial context.

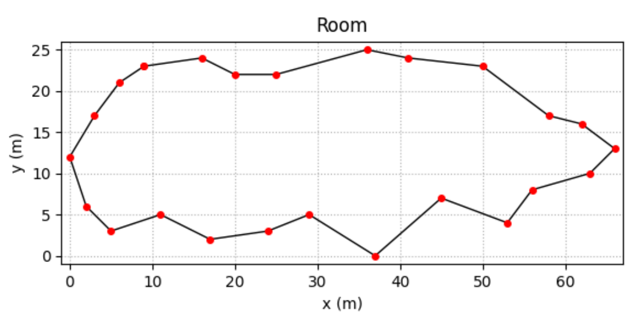

Fig. 1: Polygonal shaped room

B. Bounded Voronoi Tessellation

Voronoi tessellation provides a methodical approach to subdivide a given space into distinct regions based on certain criteria. For the purposes of this project, the 2D polygonal room is partitioned into Voronoi cells. These cells play a pivotal role in both path planning for the robotic agents and the allocation of specific regions for individual agents to scan and map.

Given a set of distinct points $P = \{p_1, p_2, ..., p_n\}$ in a plane, the Voronoi cell $V(p_i)$ corresponding to the point p_i is defined as:

$$V(p_i) = \{x \in R^2 | \forall j \neq i, \|x - p_i\| \leq \|x - p_j\|\}$$

Here, $\|x - p_i\|$ is the Euclidean distance between the point x and p_i. The Voronoi tessellation is the set of all Voronoi cells $V(p_i)$ for all points p_i in P.

C. K-Means Clustering

After dividing the room into Voronoi cells, we then group these cells into larger sections, or clusters, using the K-Means clustering algorithm (Figure 2). Each robot is assigned to one of these clusters which are formed by joining together neighboring Voronoi cells. This way, every robot gets a connected piece of the room to work on, avoiding scattered or disjointed areas.

This connected assignment helps robots move smoothly within their sections, making sure they don't bump into each other or go over areas that have already been scanned by another robot.

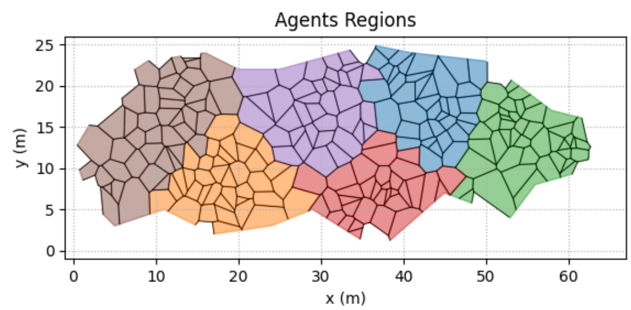

Fig. 2: Voronoi tesselation with k-means clustering

D. TSP Heuristic Problem

In the context of our project, the Traveling Salesman Problem (TSP) plays a pivotal role in optimizing the navigation paths for agents within their assigned regions. Once regions are delineated for each agent using Voronoi tessellation and K-Means clustering, the TSP is employed to generate a path that allows the agent to scan its region efficiently, visiting each portion of its region exactly once (Figure **??**).

Given an agent's region with certain points or locations $L = \{l_1, l_2, \ldots, l_n\}$ derived from the Voronoi cells, and a distance $d(l_i, l_j)$ between every pair of points l_i and l_j, our goal is to determine the most efficient path that ensures each part of the region is covered without redundancy.

Since the TSP is NP-hard, a heuristic approach is taken into consideration to ensure timely computation. The problem can be formally defined as:

$$\min \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} d(l_i, l_j) x_{ij}$$

subject to:

$$\sum_{i=1, i \neq j}^{n} x_{ij} = 1 \quad \text{for all } j \textbf{ and } \sum_{j=1, j \neq i}^{n} x_{ij} = 1 \quad \text{for all } i$$

Where x_{ij} is a binary decision variable, which is 1 if the agent moves from point l_i to point l_j and 0 otherwise.

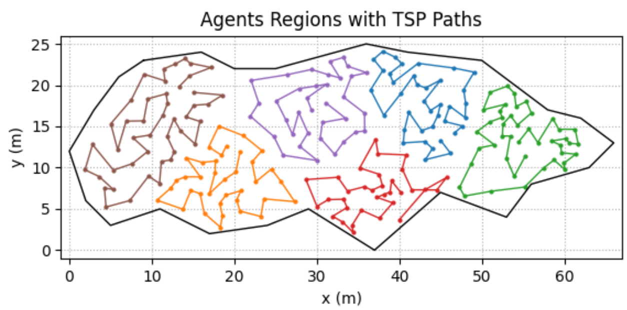

Fig. 3: TSP path planning

E. Differential Drive Robot

The robotic agents employed in this project are based on the differential drive mechanism. This is a common configuration for mobile robots due to its simplicity and ease of control. A differential drive robot has two separately driven wheels placed on either side of the robot body. It can change its direction by varying the relative rate of rotation of its wheels and hence doesn't require an additional steering motion.

Given the wheel radius r and the distance L between the two wheels, the linear and angular velocities v and ω respectively, of the robot can be related to the wheel speeds ω_{left} and ω_{right} as:

$$\begin{cases} v = r \times \dfrac{\omega_{left} + \omega_{right}}{2} \\ \omega = r \times \dfrac{\omega_{left} - \omega_{right}}{L} \end{cases} \tag{1}$$

The robot's position and orientation updates can be described using the equations:

$$\begin{cases} \delta s = v \times \Delta t \\ \delta \theta = \omega \times \Delta t \end{cases} \tag{2}$$

Where Δt is the time step. To ensure the robot accurately follows the designated path within its region, a proportional controller is integrated into its control loop. This controller adjusts the robot's speed and turning rate based on the deviation from the desired path, ensuring precise and smooth navigation within its assigned region.

F. UWB Multilateration

UWB technology provides a means for precise agent localization within the environment through the use of UWB anchors. By measuring the time of flight it is possible to determine the distances between the agents and therefore calculate their spatial positions by trilateration.

While it is sufficient to have three measurements to get the position, in our case there could be more UWB anchors, making the problem over-constrained (Figure 4).

To tackle this challenge and improve the accuracy of localization (multilateration), a minimization problem is set up. Let p be the position of an agent, and a_i be the position of the i-th UWB anchor. The distance measurement from the i-th anchor is denoted by d_i, and the actual distance between the agent and the anchor is $\|p - a_i\|$. We can also associate a weight w_i for each distance measurement to reflect its reliability.

The minimization problem can be formulated as:

$$\min_p \sum_{i=1}^{N} w_i \left(\|p - a_i\| - d_i\right)^2$$

where N is the number of UWB anchors.

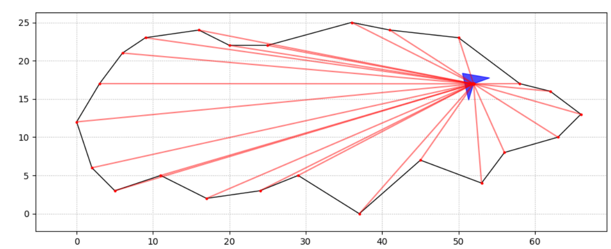

Fig. 4: UWB Multilateration

G. Sensors

Robotic agents can be equipped with a variety of sensory devices. Each sensor offers unique capabilities and advantages that allow the agent to perceive specific aspects of its environment. For the current system, the primary sensors under investigation are:

- **Encoders**: Provide feedback on the movement of the robot by measuring the rotation of wheels. They are crucial for odometry calculations, allowing the robot to estimate its relative position over time.
- **Magnetometer**: Measures the strength and direction of the magnetic field in its vicinity. It can be particularly useful for determining the robot's orientation relative to the Earth's magnetic north.
- **Lidar**: Uses laser beams to measure distances to nearby objects, producing a detailed map of the environment in real-time. It's particularly adept at detecting obstacles and can be essential for navigation and collision avoidance.
- **Stereo Camera**: Utilizes two slightly apart cameras to capture 3D depth information of the scene. By comparing the differences between the images of the two cameras, depth and distance to objects can be inferred. In 2D can be considered as a Lidar sensor with limited Field of View (FOV).

H. Sensor Noise

In real-world scenarios, the readings obtained from sensors are never perfectly accurate due to a multitude of factors, ranging from the environment's interference to the internal electronics' imprecision. Replicating these inaccuracies in simulations is crucial to ensure that developed algorithms can cope with these uncertainties when deployed in real environments.

Mathematically, sensor noise can be modeled as a random variable typically drawn from a normal (or Gaussian) distribution. If s represents the true value of a sensor reading and ϵ represents the noise, the observed reading s_{obs} can be described as:

$$s_{obs} = s + \epsilon$$

where

$$\epsilon \sim \mathcal{N}(0, \sigma^2)$$

Here, $\mathcal{N}(0, \sigma^2)$ denotes a normal distribution with mean 0 and variance σ^2, which is determined by the standard deviation σ of the sensor noise.

By incorporating this noise model into the simulation, one can account for the unpredictable variations that sensors experience, ensuring a more authentic and robust testing environment.

I. Extended Kalman Filter

The Extended Kalman Filter (EKF) is an evolution of the Kalman Filter, designed to handle nonlinear systems. It linearizes the system dynamics and measurements about the current state estimate using the Jacobian matrices, allowing it to provide optimal estimates for nonlinear systems under the Gaussian noise assumption.

Given the state vector for the differential drive robot as:

$$\mathbf{x}_t = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

and the control input as:

$$\mathbf{u}_t = \begin{bmatrix} \Delta s \\ \Delta\theta \end{bmatrix}$$

The state transition model can be expressed as:

$$\mathbf{x}_t = \mathbf{f}(\mathbf{x}_{t-1}, \mathbf{u}_t) + \mathbf{w}_{t-1}$$

Where:

$$\mathbf{f}(\mathbf{x}_{t-1}, \mathbf{u}_t) = \begin{bmatrix} x_{t-1} + \Delta s \cos(\theta_{t-1} + \frac{1}{2}\Delta\theta) \\ y_{t-1} + \Delta s \sin(\theta_{t-1} + \frac{1}{2}\Delta\theta) \\ \theta_{t-1} + \Delta\theta \end{bmatrix}$$

and $\mathbf{w}_{t-1}$ is the process noise.

For the EKF update, the Jacobians of the function $\mathbf{f}$ with respect to the state $\mathbf{x}$ and control input $\mathbf{u}$ are required. These Jacobians will linearize the nonlinear functions around the current state estimate.

In the prediction step:

$$\hat{\mathbf{x}}_t^- = \mathbf{f}(\hat{\mathbf{x}}_{t-1}, \mathbf{u}_t)$$

$$\mathbf{P}_t^- = \mathbf{F}_t \mathbf{P}_{t-1} \mathbf{F}_t^T + \mathbf{Q}_t$$

Where $\mathbf{F}_t$ is the Jacobian of $\mathbf{f}$ with respect to the state $\mathbf{x}$, and $\mathbf{Q}_t$ is the covariance of the process noise.

In the correction step:

$$\mathbf{K}_t = \mathbf{P}_t^- \mathbf{H}_t^T (\mathbf{H}_t \mathbf{P}_t^- \mathbf{H}_t^T + \mathbf{R}_t)^{-1}$$

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_t^- + \mathbf{K}_t(\mathbf{z}_t - \mathbf{h}(\hat{\mathbf{x}}_t^-))$$

$$\mathbf{P}_t = (I - \mathbf{K}_t \mathbf{H}_t)\mathbf{P}_t^-$$

Where $\mathbf{H}_t$ is the Jacobian of the measurement model $\mathbf{h}$, $\mathbf{R}_t$ is the measurement noise covariance, and $\mathbf{z}_t$ is the observed measurement.

Incorporating the EKF in the robot dynamics allows for more accurate state estimation in the presence of nonlinearities and uncertainties.

J. Simulation

Multiple agents navigate through a virtual environment, equipped with various sensing capabilities to aid in mapping and path planning. The main flow of the simulation can be outlined as follows:

1) Load elements data from a CSV file;
2) Implement Voronoi tessellation using the vertices of the room;
3) Determine the number of agents to deploy and assign sensors to them;
4) Equip each agent with a set of sensors: Encoders, Accelerometers, Gyroscopes, Magnetometers, Lidar, Stereo Cameras, etc. Every sensor is initialized with its inherent uncertainty;
5) Use K-means clustering to partition the map into regions. Each robot is assigned a set of cells within a region to explore;
6) For each designated region, compute the shortest path to traverse all cells using a heuristic approach to the Traveling Salesman Problem (TSP);
7) Send the computed path coordinates to the respective robots;
8) Robots commence their navigation. As they follow the designated paths, they simultaneously determine their position using UWB multilateration and wheel encoders. Concurrently, the robots scan their environment using sensors like Lidar, Stereo Cameras, and Ultrasonic devices to construct individual maps;
9) As each sensor possesses intrinsic uncertainties, sensor fusion is achieved using an Extended Kalman Filter;
10) Once the robots finish the scanning, the environment will send new target coordinates to organize them in pairs in order to share map information.

IV. Results

We conducted a series of simulations to assess the performance of our multi-agent system under various parameters. These simulations aimed to evaluate key performance indicators such as tessellation quality, TSP convergence velocity, Kalman filter accuracy, and the fidelity of the robot's trajectory to the target. The ensuing sections detail our findings and insights.

A. TSP Heuristic Problem

1) Convergence Speed: We observed that the TSP convergence speed was inversely proportional to the number of points in the region. However, by dividing the environment into smaller regions, the time required for TSP to converge decreased substantially. This underscores the advantage of our approach, which utilizes region segmentation to optimize TSP execution. In Figure 5 the TSP convergency velocity is computed iterating through 5 agents with voronoi cells number going from 50 to 300.

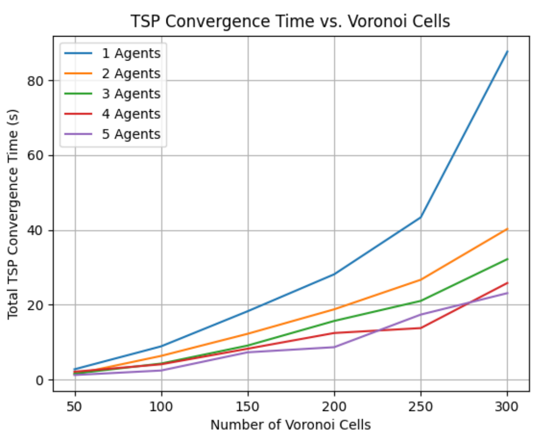

Fig. 5: TSP convergency rate

2) Path quality: In Figure 6 it is possible to see the comparison between the path generated for a single agent against multiple ones generated for five agents, using 300 centroids. The paths in both are well-generated and in the latter one they are also well-organized for multiple agent exploration.

B. Trajectory Tracking

The accuracy of a robot's path following its target trajectory is a critical metric for ensuring efficient and precise navigation. The tests showcased that, by employing a proportional controller, the robots could closely track the intended path, also smoothing the turning manoeuvre by jumping to the next target when it was relatively close to the corner. Deviations occurred occasionally due to sensor noise, but these were promptly corrected, ensuring consistent and reliable trajectory tracking throughout the simulation (Figure 7).

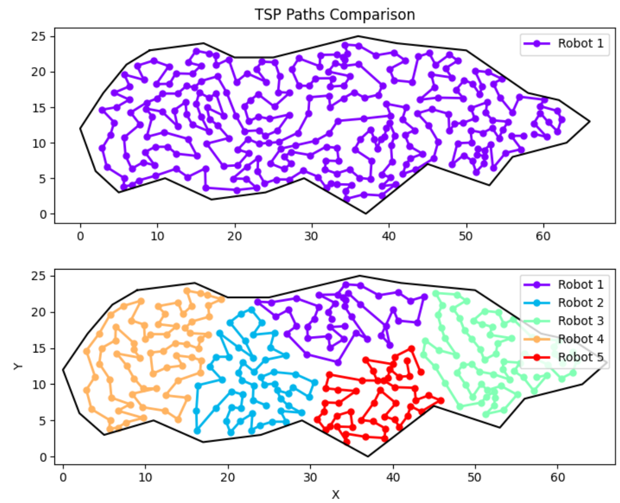

Fig. 6: TSP paths comparison

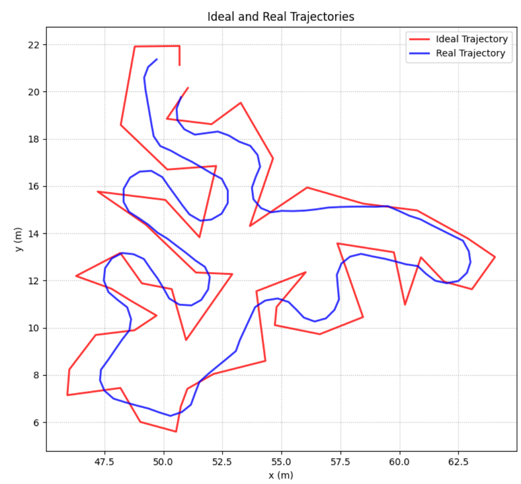

Fig. 7: Trajectory comparison

C. Environment Map Scanning

Using LIDAR and other different kind of sensors, the multi-agent system was tasked with scanning the environment. We found that as the number of agents increased, the coverage rate improved considerably. In Figure 8 it is possible to see a comparison between a random low noise mapping against a high noise one. The EKF estimations of the agent states are pretty good, having to deal with low drift.

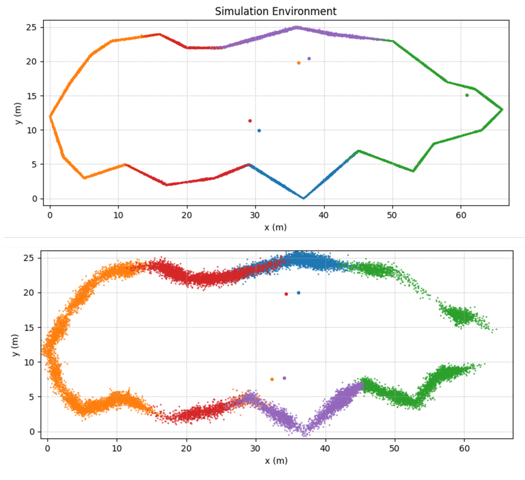

Fig. 8: Scanned Map comparison

V. Conclusions

In this paper we presented the development and simulation of a multi-agent system operating in a defined environment. Agents, endowed with sensor capabilities and their inherent uncertainties, moved within this environment, effectively generating a detailed map.

The core achievements of this study include:

- Establishment of a simulation environment capable of representing real-world challenges.
- Effective movement and interaction of agents equipped with a diverse range of sensors.
- Map generation with associated uncertainties, providing a foundational framework for more advanced navigational tasks.

VI. Future Work

While the simulation provides a strong foundation for multi-agent operations in a shared environment, there remains potential for further enhancement and refinement:

- **Obstacle Avoidance**: Integration of algorithms such as the Vector Field Histogram (VFH) can be investigated to enhance the agents' ability to dynamically avoid obstacles.
- **Advanced Path Planning**: Upon completing the environment scanning, implementation of algorithms like A* can facilitate intelligent and efficient navigation, further enhancing the agents' capabilities.
- **Physical Prototyping**: It would be interesting creating a differential drive robot prototype. This would allow for the real-world application and testing of the developed algorithms.

References

[1] M. Ahmed, R. Seraj, and S. M. S. Islam, "The k-means algorithm: A comprehensive survey and performance evaluation," 08 2020. [Online]. Available: https://www.mdpi.com/2079-9292/9/8/1295

[2] M. Jünger, *Chapter 4 The traveling salesman problem*, 04 2005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0927050705801215

[3] H. Abdulkarim. Comparison of algorithms for solving traveling salesman problem. [Online]. Available: https://www.researchgate.net/profile/Haider-Abdulkarim/publication/280597707_Comparison_of_Algorithms_for_Solving_Traveling-Salesman-Problem/links/55bcab9808ae9289a0968a31/Comparison-of-Algorithms-for-Solving-Traveling-Salesman-Problem.pdf

[4] "Mobile robot navigation based on lidar." [Online]. Available: https://ieeexplore.ieee.org/document/8407319

[5] "An overview of autonomous mobile robot path planning algorithms." [Online]. Available: https://ieeexplore.ieee.org/abstract/document/4339335