Assignment 4: Multi agent search
DD2438 at KTH

GROUP 2

Tanay Rastogi	Niklas Rolleberg
1991-07-21	1993-06-15
tanay@kth.se	nrol@kth.se


Abstract

Multi agent search is a hard problem made even harder with distributed AI, We present an approach to this problem suitable for unreliable networks. The approach was tested on ROS kinetic with lawnmowers as agents with good results.

Contents

1 Introduction

With the advancement of the capabilities of various sensors, advanced algorithms and computational resources, the technological capabilities of artificial intelligence is ever increasing. Today, the robots can not only follow instructions, but can also sense, interpret and react to uncertain knowledge by autonomously exploit the resources and making useful interpretation of the information collected in the environment. These technical capabilities can not only provide comfort solutions to us humans, but can also be a support system in situations that are dangerous or hazardous. [5] One of the domain where this kind of technological is best put to use is in a Search and Rescue (SAR) operation and disaster management. A modern search and rescue scenario poses various challenges for the teams doing the search and rescue operations. The primary goals of these teams are the fast exploration of the disaster area and the fast rescue of the victims. Since in many cases the searching procedure endangers the search team members, which may be the case in unstable buildings after an earthquake or in the wake of a nuclear disaster. This lead to the introduction of robots in these kinds of scenarios. SAR may be generally characterized through multiple dimensions and attributes including: one-sided search in which targets are non-responsive toward searcher's actions, two-sided, describing target behavior diversity (cooperative, non-cooperative or anti-cooperative), stationary vs. moving target search, discrete vs. continuous time and space search (efforts indivisibility/divisibility), observation model, static/dynamic as well as open and closed-loop decision models, pursued objectives, target and searcher multiplicity and diversity[7]

For computational analysis, a SAR problem can be divided into categories: search problem, rescue operation and formation keeping. A multi-agent cooperative search operation is a NP hard problem. Most of the algorithms available are generally restricted to special cases where reliable information is either available or assumed to be relatively simple to do collaborative tasks. Usually in a search operation, one does not know the environment or have a limited information. The robot are supposed to negotiate compromised and collapsed structure, locate victim and also provide practical sensor map of the environment so that the first responders can approach the victim

The rescue operations are activated once the victim is found in the environment. The robots are supposed to establish communication with the victim, identify the victim condition, provide essential first aid, medicines, nourishment and fluids. The robots communicates the position of the victim to other robot and possible mark the best path for the rescue for the victim.

The agents have to analyze the situation of the victims. Based upon the situation, the agents have to do some formation such that it is possible to carry the victim and move towards the risk free zone as soon as possible.

In the paper we investigate and develop an approach to do a SAR operation for a scenario such that there is one victim trapped inside the bounded area and we run an operation with the help of N robot agents. The problem statement is:

- The floor map of the environment is known for the operation. An example map can be like on given in Fig1. In the environment, we know the bounding wall of the environment and the obstacles position.

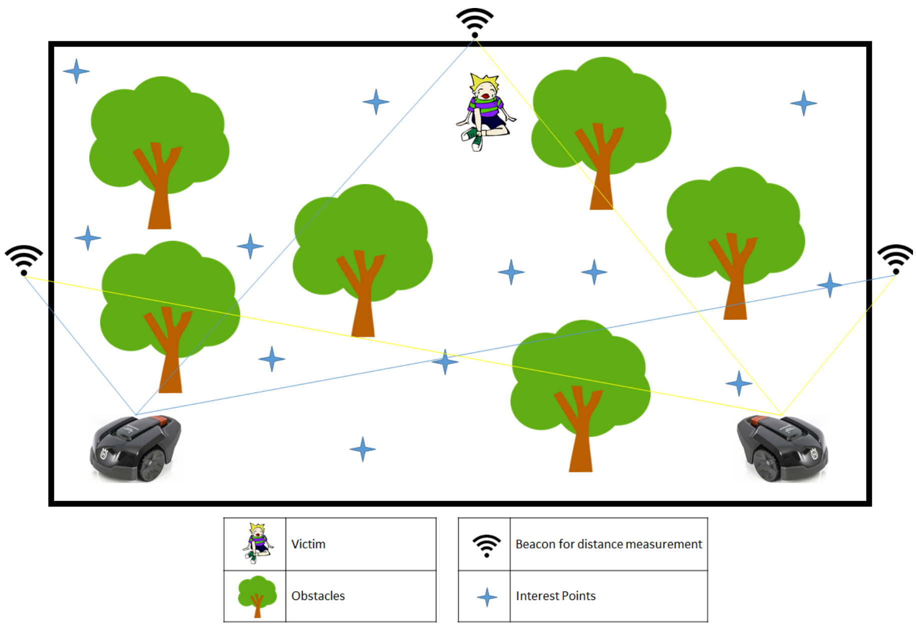


Figure 1: *Environment*

- We will assign some interest points in the environment which has to be visited by the agents. The intuition behind the placement of interest points is to cover the search complete environment.

- There is a victim whose position is unknown to the agents. The agents have to search for the victim and locate it in the environment.

- Once the victim position is located, the position has to be communicated to all the robots.

1.1 Contribution

In order to run a SAR operation for the scenario presented in the fig1, we sub divide the task, namely,

- **Localization of the Agents:** In order to know where the agents are, we did data fusion on the sensor information from the lawnmower odometry and beacon measurement. The fusion was achieved by average filter.
- **Path Planning:** All the agents have to cover all the interest points in the environment. The cooperative search for the agents is treated as a multi traveling salesman problem and is solved by hybrid A* algorithm.
- **Object Detection:** In order to identify the victim in the environment, we use image processing techniques. A pink coloured football is used as dummy victim and is identified using Hough transform.

We only investigate the search operation on the scenario and then try to locate the victim position. The agents has to cover all the interest points assigned to it, even if it locates the victim in the bounded area. For the purpose of testing, we will use lawnmowers, as agents, from Husqvarna Research Platform (HRP) which are controlled and communicated using Robotic Operating System (ROS), running on Raspberry pi 3 for networking, processing and control.

The idea here is to test the limitation and complexities of running system on limited processing capabilities and to check the reliability of the system.

1.2 Outline

Existing Search algorithm related to SAR are are discussed in section 2.

Afterwards the methods implemented for the SAR operation are discussed in the section 3 and then how they are implemented to run on the Lawnmowers using ROS is explained in section 4

For the purpose of the evaluation of the system, some maps are designed are are discussed in the section 5

Finally, the experimental results are presented in 6, and corresponding conclusions exposed in 7.

2 Related work

The SAR robots are the one that are developed for doing rescue operations either doing support missions, working with the humans or completely autonomously. Although robots are still not used widely in actual SAR missions at the moment but the development in this field is promising, thanks to the international events like RoboCup Rescue[2]. This is also well supported by the fact that today we have enough sensor suit and computational resources to run complex algorithm for the SAR. A SAR mission can be divided to search and then rescue. The most important part of the search mission for a robot is to map and then localize the environment. The environment can be known or unknown. In the case of unknown environments, multiple SLAM approaches have been studied using different sensor suits based on RFID[3], RGB-D sensors[6], Acoustic sensor[11], Thermal infrared sensor[1], and others. In a known environment, the approach can be treated as a segment of multi-traveling salesman problem, where each agent has to travel to certain interest points in the know environment such that all the agents visit all the points at-least once in least minimum time. There are several methods proposed to solve the mTSP problem like 2-opt[10], simulating the problem as a gravitational emulation[4], using K-means [9], or using genetic algorithm[8] among others.

3 Method

The three objectives defined in section 1.1 are achieved on the lawnmowers using ROS as follows:

3.1 Network Structure

The network is a distributed network between the agents and the master node. Each of the nodes has their own local network with local nodes. Communication between the agents are kept to a minimum to improve reliability on unreliable networks. A interpretation of the ROS network running is shown in figure 2.

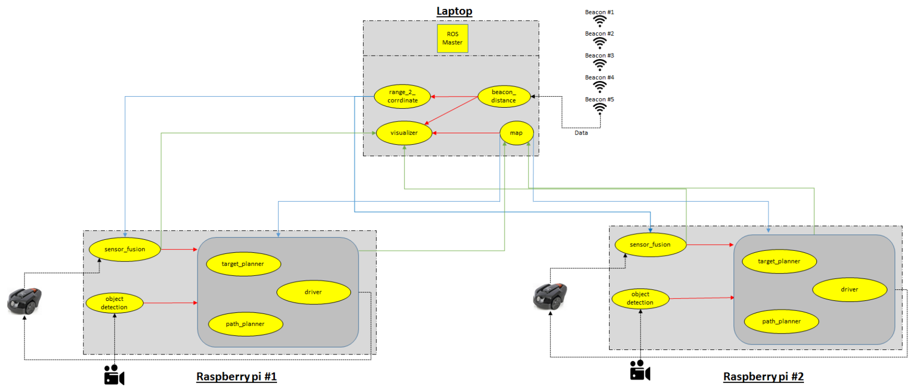


Figure 2: *ROS network structure*

3.2 Information Flow

A simplified image of the information flow in the network is shown in figure 3. The figure illustrates the flow of information and commands between the master (laptop) and one of the agents (lawnmower).

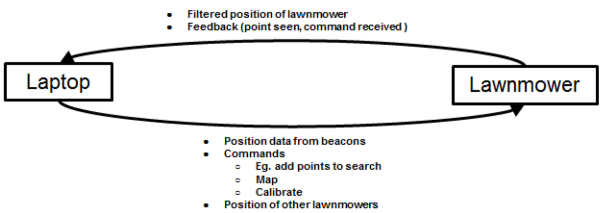


Figure 3: *Information flow in ROS structure*

The laptop receives data from radio beacons, calculates positions of the lawnmowers and sends it to the lawnmowers, the lawnmowers receive this position and combines it with data from local sensors and sends the new position back to the laptop. The combined position is much more accurate than the position given by the beacons. This position is used for visualization and is also distributed among the other lawnmowers.

Command and information messages, e.g. how the map looks and points of interest there is are also provided by the laptop.

3.3 Localization and sensor fusion

The localization of the agents are done by combining data from radio beacons located at fixed points in the environment and on the lawnmowers, and odometry sensors. The reason for this is firstly to get a common coordinate system for all the agents, and also to be able to counteract drift in the odometry data. In order to combine the data it must first be transformed to a common form. The data provided by the beacons are distances, and the data provided by the lawnmower is a position and heading relative the the lawnmowers starting point and linear and angular velocities. Firstly beacons are used to define a coordinate system with one beacon in the origin, another on the x-axis and one on the positive side of the y-axis in the xy-plane. The positions of the lawnmowers in this coordinate system are calculated with trilateration. An image of this is shown in figure 4.

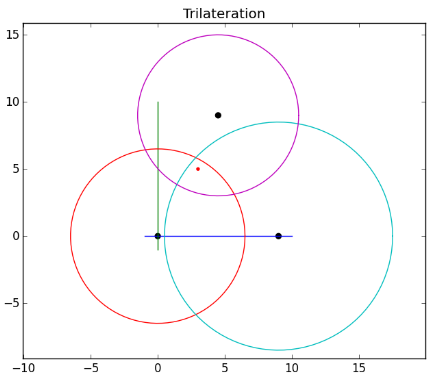


Figure 4: *Localization of agents via trilateration.*

The positions in the coordinate system defined by the beacons are sent to the lawnmowers to be combined with the odometry data. In order to do this a calibration is needed to find out the orientation and position of the lawnmower. This is done by collecting data from two positions on a line in front of the lawnmower, preferably by driving the lawnmower in a straight line from point one to point two. The second point is then used as the current position of the lawnmower, and the angle between the points is used as the heading. When new odometry data becomes available the difference between the the new data and the previous data is added to the position and heading. This is done instead of integrating the velocities because it is not time sensitive. New beacon data is added to the position via exponential moving average.

3.4 Target selection

The agent is given several points of interest to visit, and to save time these points should be visited in a good order. For this application we chose a greedy approach where the agent always goes towards the closest point. This is not optimal in terms distance traveled, but it is a very computationally inexpensive way of selecting the targets, which is important because the computational power available is limited and must be shared with other demanding processes like object detection.

3.5 Path planning

When a target has been selected a path is planned towards it using a modified version of Hybrid A*. The modifications consists of a user definable limit for the number of nodes in each element, and moving obstacles. The moving obstacles are the other lawnmowers in this application. This makes it possible for the lawnmowers to avoid each other without the need for some reactive avoidance algorithm. In order to keep the path up to date it must be re planned after a while since the moving object might change direction. The obstacle avoidance are done in two ways, firstly by making all states inside obstacles invalid, and secondly by altering the heuristic function to increase the cost for states close to obstacles. Thus making these states less desirable to expand. The effect of this can be seen in figure 5 where the path keeps away from the obstacles even though the path would be shorter if it went closer.

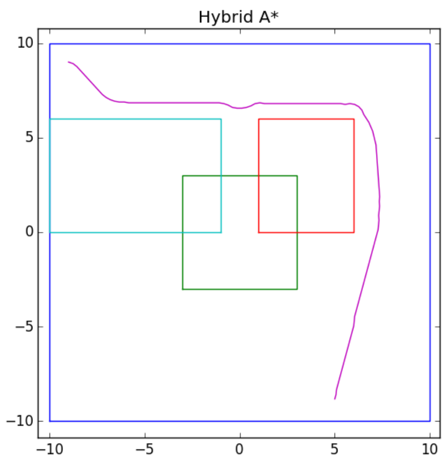


Figure 5: *Path generated with Hybrid A**

3.6 Object Detection

For the purpose of the experiment, a dummy victim is used, a pink colored football. In order to identify the ball in the given environment, we convert the camera image to HSV form and then isolate the color of the football and masking all other color to zero values. The resultant image is smoothed to remove noise in the image. We use a canny edge detector on the smoothed image to find the edges and then run Hough Transform on those edges to find the circle in the image. The Hough transform is restricted by the radius of the circle it can detect and also how far the center of the circle should be. This will reduce the number of false circle detected.

4 Implementation

This project was implemented in ROS kinetic. We created custom packages and messages for all tasks. Since we did not have a reliable network communication we could not use a regular ROS network with a single roscore. Instead each of the agents and the laptop had their own roscore and local ROS-network with nodes needed locally, and all communication between the agents were done via a ROS package called nimbro_topic_transport. This package allows for topics to be sent from one ROS-network to another. To make the information flow as simple as possible there were no direct communication between the lawnmowers, but instead messages were sent to the laptop and then distributed to the other lawnmowers.
A graph of the ROS network on the laptop is shown in figure 6, and the network on one of the lawnmowers is shown in figure 7

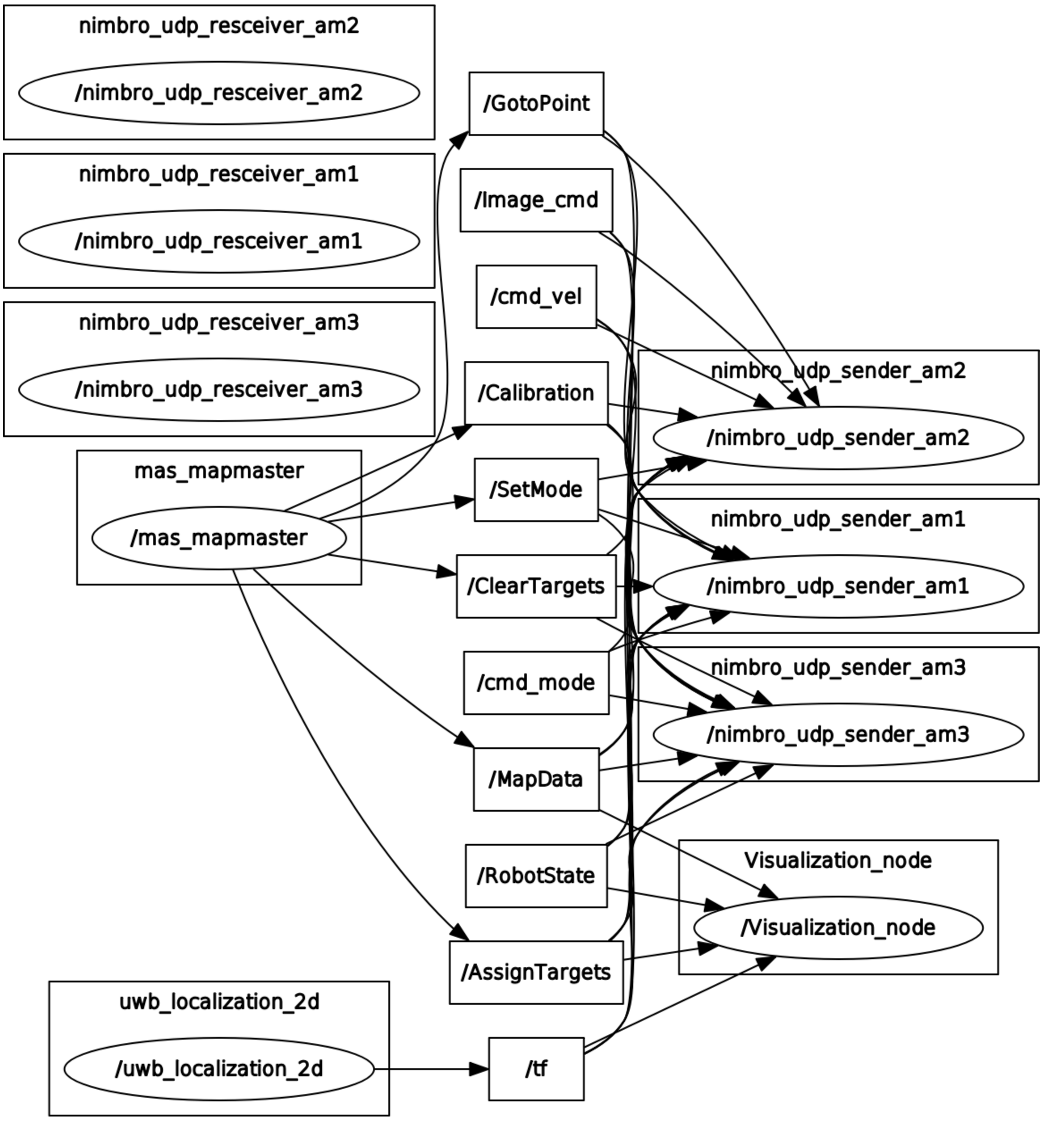


Figure 6: *ROS network on the laptop*

The nodes running on the laptop handles localization and filtering of beacon data, visualization, sending of commands to the lawnmowers and forwarding of messages between the lawnmowers.

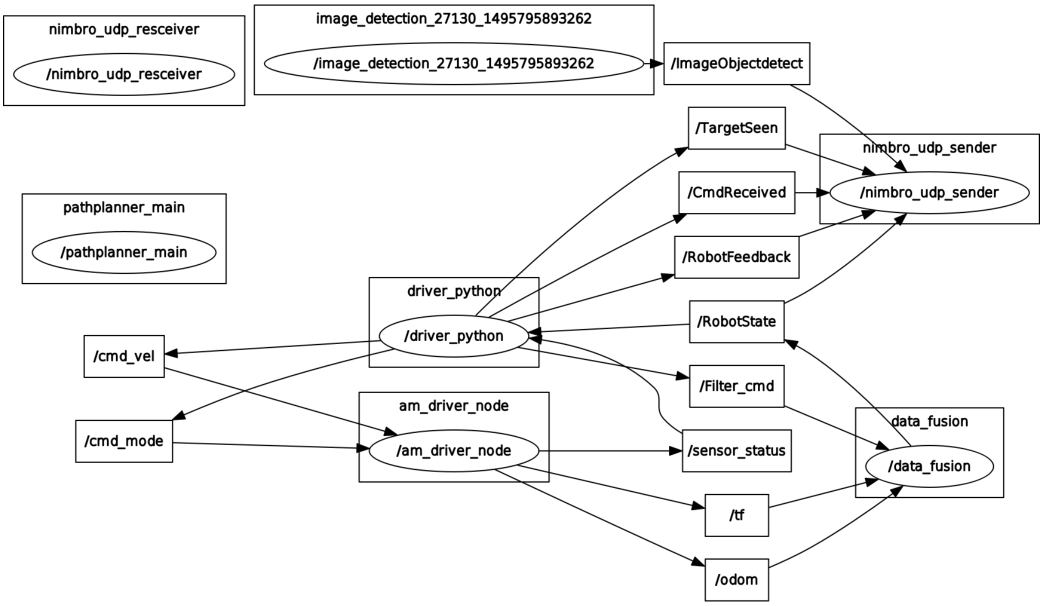


Figure 7: *ROS network on the lawnmower*

The nodes on the lawnmowers handles sensor fusion between odometry data and positions calculated from the beacons, target selection, path planning and the actual driving of the lawnmowers. All communications between the nodes are done via topics except for the path planning, where a service is used.

5 Evaluation

For the purpose of evaluation of the algorithms developed to do SAR, 4 different maps were created. Each map is to test certain elements of the algorithm to make sure that it is successfully implemented.

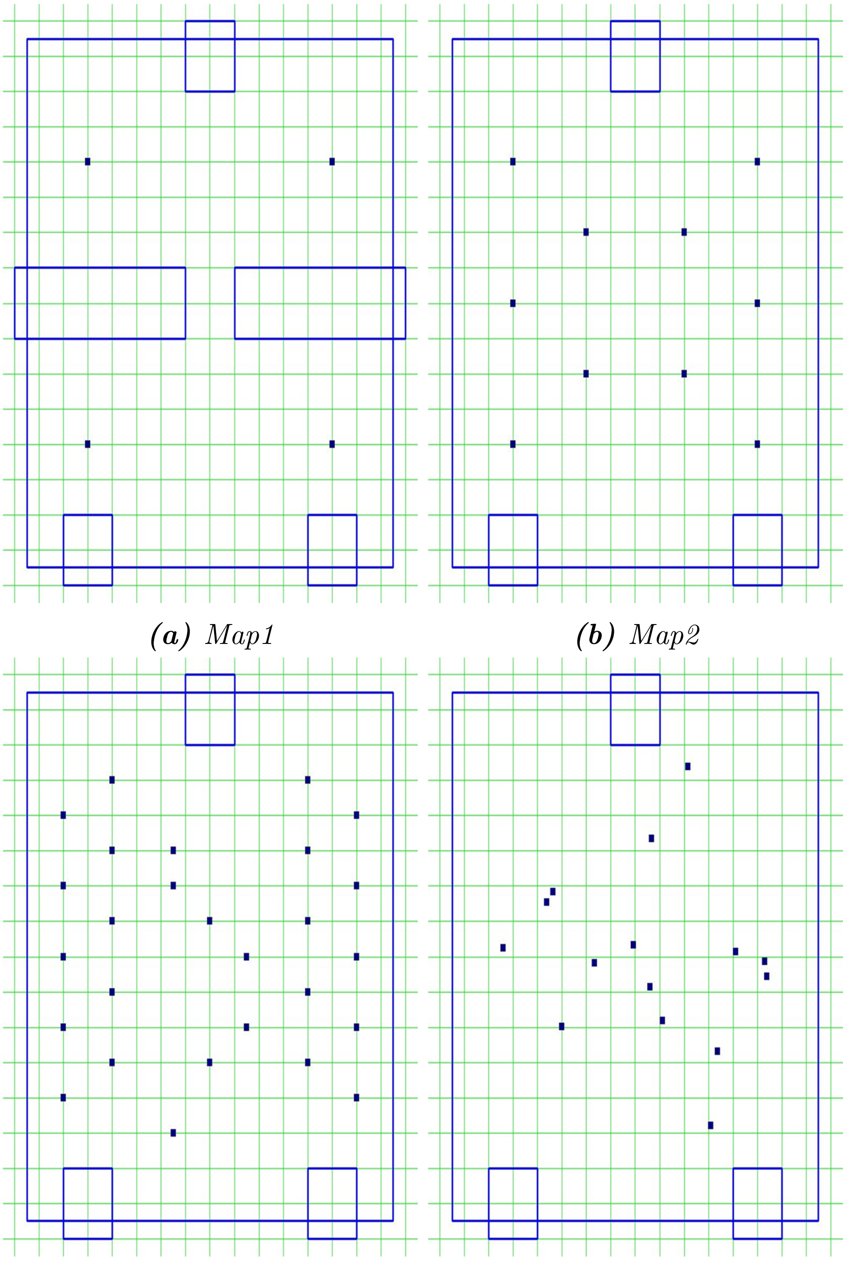
1. **MAP 1:** The map is shown in the fig. 8a. On the map, only one lawnmower is driven. The map contains 4 interest points placed in a square formation. There are two opaque obstacles. The path planning have to plan the path avoiding the obstacles such that it visits each interest points. The main aim of the map is to test is path planning algorithm in a static environment. The map helps to test the reliability of the network and if the setup is correct or not.

2. **MAP 2:** The map is shown in the fig. 8b. On the map, two lawnmowers are driven. The map contains 10 interest points placed in a horizontal hourglass formation. The interest points are assigned to the

lawnmowers in such a way that there will be cross paths. The aim of the map is to test the goodness of the obstacles avoidance algorithm. Also, to check if the system works if scaled for two agents.

3. **MAP 3:** The map is shown in the fig. 8c. For this map we drive all three lawnmowers. The maps contain three different path, one fore each lawnmower. The interest points are allocated in such a way that each lawnmower travels in its own path, not crossing with the other. The idea here is to check the scalability of the system.

4. **MAP 4:** The map is shown in the fig. 8d. For this map we drive all three lawnmowers. The interest points are generated randomly, and each lawnmower is assigned three points to visit. The idea here is to test the system when subjected to unplanned maps and how good the algorithms work in these kind of situations.



(a) *Map1* *(b)* *Map2*

(c) *Map3* *(d)* *Map4*

Figure 8: *Maps used for evaluation of SAR*

6 Results

The overall result of the project was good. We were able to make the lawn-mowers drive autonomously in the predefined maps and visit all assigned

points of interest while avoiding obstacles and each other. All tests were not perfect however, sometimes the lawnmowers did collide and other times they would just stop for some unknown reason. We also noticed that the ball we had chosen as our victim had the same color as high visibility jackets used by some children who played close to our test area. This resulted in several false positive identification of the victim.
All components of the project was also tested in isolation with good results.

7 Discussion

Overall the test results were good, but as stated in Section 6 it did not always work as intended. We believe the reason for this is mostly the result of unreliable communications. If the connection to a lawnmower is lost it's position will not be updated in the other lawnmowers, which makes it impossible to avoid. Also the approach we use to combine data from the beacons and odometry can only counteract drift in position, so if there is a drift in heading it will cause a problem. In our case this was not a big problem and we could run the lawnmowers for more than 10 minutes before seeing any effects of this.

References

[1]

[2] Robocup Rescue. `http://robocup2016.org/en/leagues/robocup-rescue/`. Accessed: 2017-06-01.

[3] Bernhard Nebel Alexander Kleiner, Johann Prediger. Rfid technology-based exploration and slam for search and rescue. *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1 2007.

[4] Hengameh Keshavarz Ali Shokouhi Rostami, Farahnaz Mohanna and Ali Asghar Rahmani Hosseinabadi. Solving multiple traveling salesman problem using the gravitational emulation local search algorithm. 2014.

[5] Szczerbicka H Becker M., Blatt F. A multi-agent flooding algorithm for search and rescue operations in unknown terrain. *Multiagent System Technologies. MATES 2013. Lecture Notes in Computer Science, vol 8076. Springer, Berlin, Heidelberg*, 2013.

[6] Skrzypczyński P. Walas K. Wietrzykowski J Belter D., Nowicki M. Lightweight rgb-d slam system for search and rescue robots. *Progress in Automation, Robotics and Measuring Techniques. Advances in Intelligent Systems and Computing, vol 351*, 2015.

[7] Nassirou Lo Jean Berger. An innovative multi-agent search-and-rescue path planning approach. *Computers and Operations Research Volume 53*, 1 2015.

[8] MengChu Zhou Xianzhong Dai Jun Li, Qirui Sun. A new multiple traveling salesman problem and its genetic algorithm-based solution. *IEEE International Conference on Systems, Man, and Cybernetics.*

[9] Majd Latah. Solving multiple tsp problem by k-means and crossover based modified aco algorithm.

[10] DiaBo Liu MengShu. A novel method for solving the multiple traveling salesmen problem with multiple depots with 2-opt. *IEEE*, 2012.

[11] José Antonio Sánchez Fernández Xin Yuan, José-Fernán Martínez-Ortega and Martina Eckert. Aekf-slam: A new algorithm for robotic underwater navigation. *Multidisciplinary Digital Publishing Institute (MDPI)*, 5 2017.