UNIVERSITY OF YORK

Exploiting Semantic Knowledge in Swarm Robotic Systems for Target Searching

by

Guangsha Xu

Doctor of Philosophy

Physics, Engineering and Technology

August 2022

UNIVERSITY OF YORK

Abstract

Physics, Engineering and Technology

Doctor of Philosophy

by Guangsha Xu

Robotic systems have long been used for search and rescue tasks in hazardous environments. The prevailing solutions which utilize delicate units for sensing and positioning show their reliance on globalized information when multiple robots are deployed. To employ multiple robots (especially swarm robots in this thesis) in a searching task, the local perceptual ability and local communication range demand a new strategy for environmental information recording and exchanging, to promote searching efficiencies of the robots.

This thesis presents a semantic knowledge-based mechanism for environmental information storage and communication in swarm robotic systems. Human expert knowledge about the environment can be utilized by such a mechanism for promoting searching efficiency. Robots without the knowledge provided in advance could learn knowledge in a task-oriented way, and help other robots in the swarm find the target faster by sharing the knowledge.

Acknowledgements

First and foremost, I would like to thank my wife Mingjia Zhang, for all of her support throughout my PhD studies. Without her meticulous care and constant encouragement, this thesis surely would not have been completed.

Next, I would like to thank my parents and mother-in-law for their support in helping me to pursue my studies in the UK. Their foresight made it possible for me to get started, and their trust has carried me through the whole process.

I must also thank my supervisors Martin Trefzer and Jon Timmis for their support and advice over the past few years. They have been beacons of light guiding me through the darkness during my PhD studies.

I would like to thank my examiners David Halliday and Tony Pipe for their time and guidance in refining this thesis. Thanks to David, also for his motivating encouragement as my thesis advisor.

Finally, I would like to thank my colleagues in the department, Wei Li, Yuyuan Zhang and other people. They have generously provided me with advice and help whenever needed.

Declaration of Authorship

I declare that this thesis is a presentation of original work and I am the sole author. This work has not previously been presented for a degree or other qualification at this University or elsewhere. All sources are acknowledged as references.

Contents

List of Figures

List of Tables

Abbreviations

FCM	**F**uzzy **C**ognitive **M**ap
HMM	**H**idden **M**arkov **M**odel
SLAM	**S**imultaneous **L**ocalization **A**nd **M**apping

Chapter 1

Introduction

1.1 Motivation

The initial idea which inspires the research in this thesis is to build a robotic system to search for a hazardous gas source in an unknown environment. The prevailing solution is to deploy the robot, which is equipped with delicate gas sensors to timely detect the gas density, the perceptional units, e.g. Kinect depth camera or LIDAR to map the environment, and the positioning sensor to record the accurate X-Y-Z coordinates in a globalized frame. On robots like this, works have been done to make contributions to optimizing the path planning algorithms in accordance with the gas dispersion and wind direction. However, those researches show weak resistance against the propagation turbulence. When multiple robots with such configuration are deployed, the system relies heavily on the accurate and globalized X-Y-Z coordinates to communicate.

In this thesis, the robots to be deployed will be relatively simple in hardware and low-cost, following the common trend in the research of swarm robotics. The robot has its abilities of locomotion and environmental perception. This means, it can go through the searching place to reach the gas source and confirm the source only in the vicinity of it. Useful information about the environment will be recorded and communicated. In order to amend the weakness in searching efficiency caused by lacking the complex sensors, the proposed system will parallelly run on robots in a swarm. Therefore, the strategy for environmental information storage and communication is urgently required. Being inspired by the knowledge learning and communicating process of our humans, the semantic abstraction of the information is proposed. Here, semantic abstraction means linking pieces of information (e.g., of objects) through their contextual purpose or meaning within the environment. Also, the expert knowledge of humans about the searching environment could be provided to the robots for promoting searching efficiency.

1.1.1 Challenges of Swarm Robotic Systems

To deploy a swarm of robots working for the searching task, the challenges of swarm robotic systems are raised as follows,

- The hardware platforms embodied in the representative robots from the swarm robotics community are designed to be low-cost and simple, so that a large number of robots can be deployed.
- The robot only has access to information within a local range.
- The robot can only communicate with other robots at limited distance.
- The searching task should be assigned to each robot to guide the behaviour design of it.

1.1.2 Core Topics

In order to perform the searching task, the robot being deployed should be able to: store searching results from past experience, aka mapping the environment; plan a path by combining the real-time sensor inputs and the mapping result; communicate with its neighbouring robots and use information received from other robots. The mechanisms related to semantic knowledge/information are designed to fulfil these requirements in this thesis. They can be summarised as follows:

- Swarm robotic controller architecture for incorporating semantic knowledge.
- Semantic knowledge model for swarm robots.
- Semantic knowledge learning of swarm robots.
- Semantic information exchanging of swarm robots.

In this thesis, semantic knowledge refers to the connections between concepts that represent entities in the environment, e.g., objects and rooms. And semantic information is the robot's interpretation of collected environmental data.

1.2 Hypothesis and Objectives

1.2.1 Hypothesis

Hypothesis: *The application of semantic knowledge based mechanism and decentralised communication in a swarm robotic system increases its speed, efficiency when performing search tasks. Also, an expert semantic knowledge about the searching environment could further increase the efficiency if provided in advance.*

The following sub-hypotheses support the main hypothesis:

Sub-hypothesis 1: *The prior semantic knowledge about environment can promote searching efficiency of a single robot.*

Sub-hypothesis 2: *Without prior semantic knowledge, to deploy a swarm of robots exchanging the self-learned semantic knowledge can promote the searching efficiency compared with a single robot.*

Sub-hypothesis 3: *Without prior semantic knowledge, the efficiency of the searching can be promoted by increasing the swarm size when the robot in the swarm exchanges the self-learned semantic knowledge.*

Sub-hypothesis 4: *The prior semantic knowledge about environment can promote searching efficiency of a swarm of robots.*

1.2.2 Objectives

- To show that the searching speed of a robot with semantic knowledge in advance using the proposed system architecture outperforms a robot without the semantic knowledge.
- To show that without providing semantic knowledge in advance, deploying a swarm of robots which learn and communicate the semantic knowledge

about the environment, find the target faster than only deploying a single robot.

- To show that without providing semantic knowledge in advance, the searching time of a swarm of robots which learn and communicate the semantic knowledge about the environment, can be reduced by increasing the swarm size.
- To show that the searching speed of a swarm of robots with prior expert knowledge is better than the swarm of robots without the semantic knowledge.

1.3 Research Contributions

This research makes the following contributions to technology, methodology and body of knowledge in the field of swarm robotics.

- Technical
  1. Application of semantic information storage and reasoning in a swarm robot system.
  2. The construction of robot controller for searching tasks deploying fuzzy cognitive map.
  3. The formation of semantic map follows the local restrictions of sensors and communications in a swarm robot system.
- Methodology
  1. Explores the method of knowledge abstraction and utilization for recording information about the environment in a searching task.
  2. Explores the application of semantic knowledge communication between robots to helps promote searching task performance.
  3. A framework of combining semantic information and numeric sensor inputs to build a robot controller.
- Body of knowledge

1. Proposes a low-cost,task-oriented mapping method for the physically-simple robot in swarm robot system.
2. Explores how the semantic knowledge is communicated in a swarm robot system.
3. A bottom-up robot behaviour design pattern for swarm robotic system executing searching task.

1.4 Thesis Organization

This thesis is organized and divided into seven chapters as follows:

- Chapter 2 Background and Related Works

  This chapter reviewed the topic of swarm robotics and the techniques required by searching tasks. The searching problems are discussed in details. At last, the approaches for statistics analysis are reviewed.

- Chapter 3 Searching Task and The Fuzzy Cognitive Map Inspired Robot Controller

  This chapter defines the searching task to be solved in this thesis. The Fuzzy Cognitive Map inspired robot controller is built for the incorporation of semantic knowledge. The Hidden Markov Model is deployed for recording semantic knowledge and utilizing semantic knowledge.

- Chapter 4 Prior Semantic Knowledge Utilized by A Single Robot

  This chapter testifies the effectiveness of prior semantic knowledge for promoting the searching efficiency on a single robot. The semantic knowledge describing more generalized object to room relation is proposed. Experiments with different set-ups of objects are run to validate the contribution of various semantic knowledge. The reliance on the specific object in a room is explored.

- Chapter 5 Robots Learning The Semantic Knowledge

  This chapter shows the mechanism of the semantic knowledge learning, communication and utilization for the robot in a swarm without providing semantic knowledge in advance. The searching time of robots in a swarm if compared with a single robot in the absence of semantic knowledge. Comparisons are made between robots in different sizes of swarm. The correlation between searching time and knowledge learning time is explored.

- Chapter 6 The Advantage of Prior Semantic Knowledge on Swarms

  This chapter presents an enrichment of semantic information transmitted between robots with prior semantic knowledge. The advantage of proving prior semantic knowledge is tested in different sizes of swarm. The advantage is compared crossing different sizes of swarm as well.

- Chapter 7 Evaluation and Conclusions

  This chapter summarises the work presented in this thesis, and discusses the limitations. The potential areas of future work is then suggested. The conclusion is made by testifying the main hypothesis.

Chapter 2

Background and Related Works

This chapter begins with a review of swarm robotics, which is an irreplaceable part of robotic research inspired by the observations of animal colony behaviours. Then research on exploration tasks with swarm robots is reviewed for designing robots' space coverage strategy. To find the solution for recording information about the searching environment, we review the works on SLAM(Simultaneously Localization and Mapping). Work on the knowledge model is then carried out in order to develop the semantic knowledge management strategy. After the wide review of related research topics, we focus our attention on discussing the searching problems. At last, methods for statistically analyzing the data collected from experiments are presented.

2.1 Swarm Robotics

2.1.1 From Biological Inspiration

As the inspiration of robot research coming from imitating behaviors of human beings and animals, researchers found answers of solving self-adaptive multiple robots systems from observing self-organized colony animals. Swarm robotics can be described as "the study of how large numbers of relatively simple physically embodied agents can be designed such that a desired collective behavior emerges from the local interactions among agents and between the agents and the environment" [3]. Contrary to the initial guess about hierarchical and centralized control in animal colonies, stigmergy behaviors are used to describe how individuals make stimulus-response elementary decisions and interact locally with other individuals or environment like the example of building activity in termites[4]. Studying the ant colonies choosing shortest path [5] from binary bridges leading to food source with different length lead the findings of ACO(ant colony optimization) and revealed principles and properties behind. When observe from the colony level, emergent behaviors, like choosing shortest path, comes from individual level interactions. Behind this, a positive feedback of leaving pheromone is strengthening individual decision makings about the binary bridge, a negative feedback is responding to natural processes like food source exhaust, amplification of fluctuation models stochastic and brings dynamic to deal with changes the colony facing, at last stigmergic interactions interchange information among individuals.

According to [6], collective behaviors of social insects can be categorized as,

- Coordination, tasks needs both temporal and spatial organization of individuals.
- Cooperation, tasks cannot be fulfilled by single individual.
- Deliberation, collective decision making process when multiple choices are faced toward every individual.
- Collaboration, individuals simultaneously conduct different activities but make contribution to the same task from colony level of view.

Environment factors modulate self-organized behaviors can be classified as "out colony" factors and "inner colony" factors. Corpse clustering example illustrates how "out colony" factor modulates behaviors and on the contrary how these behaviors modulate the "out colony" factor [7]. As for "inner colony" factors, division of labor in wasps are allocated according to colony size and individual experience by the method of colony stimulus level and individual task respond threshold [8].

2.1.2 Swarm Engineer

Looking into swarm robotics system in a engineer perspective, especially when researchers try to duplicate what happens in colony animals like the bird flocking, problems first come are how can we design individual behaviors and communications between individuals. The most intuitive method is the bottom-up individually designed method, which focused on designing individual level behavior at first and fulfill the colony task through a trial-error process. In more detail, two different kinds of model are used to describe mechanisms behind individual behaviors: PFSMs [9](probabilistic finite state machine) models the process of robots taking sensor input and make state transitions accordingly. Fluctuation is preserved through introducing probability when state transition is decided to keep features of bifurcation and multi-stable. Virtual physics-based method treat individuals in swarm robotic as physical entities and model interactions as forces between them. For example, the Lennard-Jones potential are commonly used to formulate attractive and repel forces between robots according to their Euclidean distance [10]. Apart from these, automatic method are also available without developer interference: RL(reinforcement learning) [11] method need to be adapted when used in swarm robotics because positive feedback from the colony level should be decomposed locally while the training process. On the other hand, EA(evolution algorithm) is used for finding the optimized parameters of an artificial neural networks controller.

The categorization of collective behaviors presented in swarm robotic literature is as follows. These behaviors are trying to deploy the design method we mentioned before like PFSMs, to mimic colony animal behaviors with robots. To solve a search task, we focus the study of collective behaviours on those that deal with the spatial distribution of robots. Aggregation achieved by state transition between obstacle avoidance, approach, repel, and wait [9] in PFSMs depict how to keep robots in a focused region. Novelty behaviors like pattern formation performed by kilobot [12] provide possible solutions for timely environment surveillance in certain shapes as parts of the dynamic environment mapping. As for a networked exploration, chain formation behavior [13] shows its success in search and rescue tasks. Apart from these, there are other behaviours for the spatial organization of robots, such as the self-assembly behaviour [14] and the object clustering behaviour [15]. They are not initially intended for spatial coverage, but can be effective in certain circumstances. In the areas of collective robotic motion control, we can find collective behaviors useful for our navigation task. Swarm coverage inspired by the pheromone leaved by ants on trails [16] make contribution to an efficient path planning. Similar tasks are also solved by method from wireless sensor networks

(WSN) community [17]. Flocking [18] represent the coordinated motion control we can conduct on a groups of robots. At last, collective behaviors study are about collective decision making, which is composed of consensus achievement, like path choosing when bifurcation exists [19], and task allocation most obviously reflected in frontier competence [20] during exploration tasks involved multiple robots.

In conclusion, trying to solve navigation and mapping problems with swarm robotics comes with advantages of scalability, robustness and flexibility. Scalability means stable system performance dealing with different individual size as we can recruit auxiliary robots. Robustness is the alias for fault tolerant, when individual collapse other robots could make up lost caused by the broken one and keep system functioning. Finally flexibility in context of robotic mapping and navigation would make sense in circumstances varies from light conditions, in-door or out-door arena, etc.

2.2 Exploration Tasks with Swarm Robots

Exploration tasks have importance in many scenarios like post-accident victim search and rescue, hazardous chemical ingredient leaking localization. Because of the dangerous operating environment, avoiding participation of human-beings promote research conducted with robotic system. In detail, literature surround this topic is about space coverage, area surveillance and dynamic monitoring. Methods of robot coordinated exploration and consensus about optimized path construction should be designed. The essence problem of swarm robotic exploration would be swiping certain times over certain area and the finite goal is building maintainable links between entrance and targets. Research recent years also extend applications to mining detection and space clearance. Exploration unavoidably consist SLAM(simultaneously localization and mapping) as kind of path modernization method. But we will focus on solving from swarm perspectives in this section, for the reason that SLAM will be reviewed in detail in the section 2.3.

The Simplest way of coverage tasks is the randomized search method when robots are controlled with strategy only avoiding obstacles and other robots. Experiments proves this method, although very straightforward, will finally lead to the goal of entirely coverage. More importantly, this method is usually treated as reference of efficiency for comparison with other designed methods.

Biological inspired method like combining flocking and pheromone mechanism [21] . Flocking with the basic rules of separate, alignment, cohesion will lead robotics running in an consensus direction while avoiding collisions. And for pheromone mechanism, if the pheromone does not evaporate and the perimeter of the zone of pheromone detection is not fixed, the behavior of the robots is aimed strictly towards space exploration, but if the pheromone does evaporate and the perimeter is fixed, the behavior of the swarm resembles environment surveillance [22]. The strategy of pheromone also varies, as in simulations it can be laid on either vertices or edges when the area is represented by graph and the information of pheromone also can either be accumulation of its current position or the accumulation of next square which brings the prediction one-step ahead in learning real-time A* (LRTA*) method [23]. But when it comes to real robot, pheromone is hard to be implemented, so virtual pheromone [24] is introduced in the way pheromone is carried by robot and finally forming a dynamic chain between two points.

In the paper about frontier search[20], the novelty of introducing laser-limited sonar is emphasized because of its combination of distance and accuracy. The floor of arena is symbolized as squares which can be categorized as occupied, free and unexplored, and the consistent unexplored squares above certain length will be marked as frontiers, where robots will head for searching. This method was initially introduced for single robot exploration, but adaptations were made to fit the requirement of swarm robotics, the market strategy [25]. Multiple robots will bid for the same frontier offering their cost of reaching that frontier and the estimation of information gain after reaching. What's more, discount is also introduced for the sake of keeping the robot, which win the bid explore continuously. Auction can be conducted distributively. But this method still have its drawbacks like sensor occultations may cause multiple frontier in the same place, to avoid this, map be divided into different regions and robot be assigned to them instead of specific position [26]. And voronoi graph [27] needs to be generated in order to divide the map. Also Hungarian method [28] can be used to assign robot to region simultaneously compared with auction(market) strategy. Stable marriage algorithm [29] can also be adopted to assign frontiers to robots, as this is an algorithm designed for allocating one-one pair problem, we will create fake frontiers if we have more robots than frontiers and vice versa.

Space filling curves like the Hilbert Curve [30] can be generated if the area is known ahead, and for the sake of multiple robotics, robots can be distributed on the curve with separate distance to travel. This mechanism [31] also has its redundancy and scalability because robot will take on the work of their neighbors if breakdown happens. The spanning tree [32] method is developed from off-line method to on-line method and ant-like method to implement searching in an unknown environment and act more efficiently. Back-tracking is necessary if robots are distributed in the area randomly, where distance to travel for one robot may be longer than half of the Hamilton Circle(the circle surround the spanning tree). Spanning forest [33] method is another branch of spanning tree method, splitting spanning trees to sub trees, but not depending on robots initial positions. Strategy of cellular decomposition [34] is essential if obstacles are placed inside the searching area or the searching area is non-convex, then decisions could be made to search in a depth first manner or breadth first manner.

Chain based method [13] is going to build links between nest and the target positions in a way robots becoming landmarks if they explore far enough from the existing landmark. The chain become dynamic when the tail of the chain moves and the rest vertices of the chain align their position consequently. If we use probability when one robot switches from explorer to chain member as landmark and the last one of a chain has probable to leave the chain, the quantity and length of chains varies. And the different distance between nest and target position call for different quantity and length of chains, like the longer distance needs longer chain to reach it. One important area of research is how to maintain already formed chains between nest and target and if breakdown happens what is the quickest method to diagnose where it is and the strategy to restore.

Algorithms also concerns task allocation in collectives is proposed in [35]. In tasks detecting dangerous chemicals with swarm robotics, collective deploy alpha-beta coordination method allocate robots to complementary roles of alpha and beta. Alpha robots is coded to be active and be responsible for exploring new areas, on the other hand beta robots tend to be conservative and aggregate around average center o f neighbors like what happens in flocking. Robots switch their roles dynamically according to own individual state and entire colony state. Balancing between exploration and exploitation is achieved through this alpha-beta role partition.

2.3 Visual SLAM

2.3.1 Anatomy of SLAM

In order to record transferable trace of robot linking arena entrance and searching target, information regarding precise localization and distinguishable landmarks are necessary for replicating the trajectory. So accurate localization and certain method of environment modeling and recording are necessary component of proposed robotic system. Early researcher take localization and mapping separately, however this lead to an chicken-egg problem in consideration of high accuracy. For the reason that localization and landmark recording take outcome of each other as reference, later researchers came to the conclusion that we cannot solely solve one task without another and this lead to the concept of SLAM(Simultaneously Localization and Mapping). SLAM system is commonly separated into front-end process and back-end process [36]. The front-end process deals with sensor input and organize data into certain map style.On the other hand, back-end works on constructed map optimization by eliminating accumulated error. Apparently hardware is more hardware dependable. Vast option of sensors are available for SLAM, like sonar, radar, lidar and camera. We propose camera as perspective component of our system not only because camera could provide precise range information sufficient for robot SLAM tasks, but also camera provide robot perceptions of textures, colors and shape information for further semantic meanings abstraction as we will discuss in Section 2.5. But reviews in this section will still cover other sensor based algorithms because they may still inspire our works after. In addition, visual SLAM has its sub-categorizations of monocular SLAM, stereo SLAM and RGBD SLAM according to different kinds of cameras it applies respectively. Difference lies in process of initialization and calculating object point depth between them, which will be covered lately and it is relatively trivial in studying algorithms of whole SLAM system.

Initial solutions comes with sensor input from odometry and laser/ultrasonic scanner. Because of uncertainty introduced by noise in those perception unit input probabilistic distribution models are deployed to represent trust on output of robot pose and environment representation. Controversial sensory noise eliminating approach relates to Gaussian distribution which models the sensor input mixed with noise. EKF(extended Kalman filter) algorithm [37] consecutively take inputs from odometry and laser scanner then estimate the joint state of robot pose and landmark pose. Apparently drawbacks sourced from linearization of sensor and motion model. In addition, computation complexity quadratic increase with quantity of landmarks makes EKF-SLAM unsuitable for persistent navigation in long-term scenarios. For the reason that EKF-SLAM only cater for linearized system, other literatural work on convergence and consistency of SLAM by other filter based algorithm. As an example, FastSLAM [38] is the first proposed algorithm try to directly model SLAM process in an non-linear manner and estimate the non-Gaussian pose distribution. FastSLAM applies particle filter recursively decreasing errors accumulated during robot navigation. But directly deployment of particle filter is unavailable as state space dimension drastically increasing during the navigation process. Rao-Blackwellization according to Bayesian Inference separate the trajectory estimation and mapping estimation. SIS(sequential important sampling) and re-sampling recursively estimate the best distribution of robot trajectory from particles and for each particle EKF is performed to update observed landmark pose based on known robot pose. Researchers found Bayesian filtering methods also work well on visual based SLAM as computer vision algorithms abstract features in image to landmarks, and some of them even perform as good as modern algorithms list below [39].

Modern SLAM algorithms interpret state distribution problems as MAP(maximum a posteriori) estimation and find resemblance with the BA(bundle adjustment) used in Structure from Motion, a system that can be described as SLAM under the assumption that the working scenario is static. Also the key factor behind modern SLAM solutions is sparsity of the map as landmarks are visible locally to certain nodes. Transforming MAP estimation into nonlinear least squares problem current SLAM libraries(e.g., G2O [40], Ceres [41]) use graph-based method, which represent robot pose and landmarks with nodes and represent motion equations and monitor equations with vertices in graph theory as edges of constraints. PTAM(Parallel Tracking and Mapping) set the milestone for monocular visual SLAM [42]. It creatively use two thread parallelizing visual odometry and map optimization. Lots of later works take this parallel framework and key-frame Bundle Adjustment map management.

2.3.2 State-Of-Art Visual SLAM Solutions

After reviewing past research on kinds of SLAM algorithms, we decide to focus on several state-of-art visual SLAM methods, dig into details of them, compare and choose components for our proposed robotic mapping and navigation system. ORB-SLAM [43](Oriented FAST and Rotated BRIEF) and LSD-SLAM [44](Large Scale Direct) are the two hottest branches in modern SLAM research. From its literal meaning, LSD-SLAM not like other feature based method, deals with pixel data directly. Because of that, higher computation resource is required and usually a GPU is installed for achieving real-time execution. Even though LSD-SLAM generate dense maps easier for human read and understand, we prefer its counterparts ORB-SLAM as the algorithm we will learn in more detail as for consideration of keep simple in hardware requirement.

ORB-SLAM is composed of modules taking state-of-art method from feature extraction to loop detecting, also organized as a high consistency structure in the software engineer perspective. Since then we spent times not only studying publications about it but also read its source code in order to get an more grounded knowledge about a modern SLAM system. Being part of feature based visual SLAM, ORB-SLAM takes ORB features which are FAST corners abstracted from multiple scale pyramid of raw image maintaining scale information and adapted rotation invariant BRIEF descriptors. Choosing ORB features is due to considerations from real-time performance(less than 33 ms), as comparison other features takes time an order of magnitude slower, like popular SIFT features ($\simeq$ 300 ms) , SURF($\simeq$ 300 ms), or A-KAZE($\simeq$ 100 ms). The lack of direct depth information calls for initialization to triangulate beginning set of landmark points and decide depth scale. RANSAC method is used for ticking out outliers after initial feature pairing after motion. According to the planar geography, Homography and fundamental matrix are calculated in parallel for initial motion models of planar scene and nonplanar respectively. A heuristic score function will select a better model from them. Like works done in PTAM.

ORB-SLAM use three parallel threads executing tracking, local mapping and loop closing. There are several important tools used in ORB-SLAM needs explanation. BA(Bundle Adjustment) [45] optimize the joint state of robot pose and landmarks pose by method of non-linear optimization techniques and trying to minimum projection errors between images and key frames. Bags of Words [46] is introduced for image similarity comparison. Words are local features in images, trained by off-line dataset. Each word is combined with a score describing inverse document frequency, which means a high scored feature appears rarely in the training dataset and it is very distinguishable. So images will be transferred to bags of word, a numeric vector contains information about every word's frequency. When trying to compare similarities between two images, a calculation of distance between two vectors is conducted which drastically decrease time taken for computation. In tracking thread, firstly estimate motion from last frame through a constant velocity motion model and check matched feature points. If matches less than certain threshold, convert current frame to Bags of Words and try to compare similarities with the nearest key frame. If they are not similar too, global relocalization is performed by compare similarities with key frames globally. After getting the rough prediction of robot pose, optimization with local map is conducted by projecting feature points back into current frame connected key frames and their co-visible key frames. By co-visible, it means number of sharing observable feature points greater than 15. At last, this frame will be inserted as key frame when a determined time period past after last insertion of key frame. Local mapping thread update edges connection new inserted frame to other key frame, and also the co-visibility graph. A local BA is conducted within co-visible frames. Key frames that could observe the co-visible key frame are also included in the local BA, but their pose retain fixed. Local mapping create new landmark point through accurate triangulations from wide baselines between key frame. To avoid redundancy of key frame, local mapping thread will tick out key frame share its 90% feature points or more with more than three other key frames. Loop closing functions in every modern visual SLAM system, by ways of recognizing retreat to positions traversed before. In ORB-SLAM, Bags of Words is used for recognizing of retreating. Loop correction fuse duplicated landmark points and insert new edges in the co-visibility graph. Finally, global accumulated error is optimized and dispersed among essential graph, which is a spanning tree generated from co-visibility graph in approach of selecting edges share most landmark points. Experiments prove the accuracy of ORB-SLAM outperform other visual method and a timely re-localization always can be achieved if tracking lost happens. One most significant drawback is the time needed loading visual vocabularies of the Bags of Words while system booting. Later works also extend ORB-SLAM to perform semi-dense reconstructions over key frames [47].Map of semi-dense is generated which contains abundant information for possible object reorganization and high-level message abstraction. In addition, researchers based on ORB-SLAM build a MOARSLAM(multiple operator augmented relative SLAM) system [48] which is a scalable client-server multiple robots SLAM system could possible share augment reality in it. And more detail about multiple robots SLAM will be covered in 2.4.

2.3.3 Biological Inspired Visual SLAM

Researchers are always trying to solve robotic problems from imitating behaviors of biological entities, from insects to mammal, even human beings, and so does the problem of SLAM. Desert ants are found to build reliable habitual routes between food source and nest not relying on trail pheromones but visual landmarks [49]. Animal neuroscientist [50] study the navigation and mapping behaviors of rodents, like food source and nest position recognition and region boundary labeling. Output of their research shows that the structure called hippocampus in rodent brain contains nerve cells responsible for spatial response. While the rodent moves around, some nerve cells are found to be activated by certain locations relates to visual cues captured by them in the arena and some are activated by absolute heading direction of the rodent. Apparently joint state of an Cartesian x-y position and angle θ is sufficient for representing a rigid body displacement in 2-dimensional planar space. So the first edition of RatSLAM [51] takes input from visual landmark cues with boarded camera and motion cues with wheel encoders. And as counterpart to the rodent hippocampus model, robotic researchers apply the continuous attractor network [52] to implement firing mechanisms of pose and angle at first. In continuous attractor network, every cell has excitatory connections to its adjacent cells and inhibitory connections to every other cells. And unlike other neural networks, continuous attractor network operate by varying activation of neurons instead of tweaking weights connecting them.A wrapping from one side of boundary to the opposite side of boundary is also connected to avoid problems of crossing border when stimulation remains on the same direction. As result, when plot from the over-head view, a tessellation like activation of position cells is achieved because of the wrapping up of continuous attractor network. But the first version has two separate networks for position and heading direction and this causes ambiguity. So the next version combines them into a 3-dimensional attractor network and got the conjunction cells called pose cells. This refines the performance in progress but makes no difference to output formation, which still is the repeating reflection of real physical world into a fixed size rectangle. Then the third and almost the final version of SLAM give birth to the experience map more meaningful for navigation. Experience in this map is connected with activation pose cell and activation of local view cell(visual template created consecutively for recording traversing distinguishable places), it will have its own position with Cartesian coordination meanings, also a time stamp is registered accompanying creation of experience. Just like loop closure in those graph based SLAM or smoothing SLAM method, the connections between local view cells and pose cells is strengthened. When a re-localization happens by recognizing traversed local view cell, the relaxation of experience map is always needed as mismatch between positions of them is caused by accumulated odometry error. In consideration of memory usage, the method of experience map maintenance is overlaying grid over it and merge duplicate experiences in same grid. Robot conserves two separate maps: global experience map and local obstacle map. The local obstacle map is in the robot centered coordination and constructed in real-time scanning obstacles around the robot. Respectively navigation tasks also be decomposed in global level and local level. If navigation goal is experience exits in the global experience map, then global path generator calculate the quickest path connecting navigation target and robot current position based on the time stamp recorded with experience. Local path generator takes a short window time of sampling from range and bearing obstacle detector and outputs the optimized local trajectory applying tree search method. When it comes to non-target exploration circumstances, local path generator chooses the most greedy path to achieve efficient space covering. Experiments are undertaken in different conditions ranging from indoor in-door artificial build arena and out-door suburb area in St Lucia. As a novel solution to robotic navigation and mapping tasks, the real-time output output accuracy of RatSLAM can not match those state-of-art algorithms like ORB-SLAM we discussed in detail 2.3.2. But without complicated visual processing algorithms like feature abstraction and descriptor calculation, not to mention explicitly loop closure performing re-localization recognition and global accumulated error relaxation, RatSLAM has its strength in simple hardware requirement. This strength of RatSLAM cater to the design policy of swarm robotic individuals, so we will test it as first candidate SLAM solution in our future real robot work.Then works of later researchers augment communication abilities to iRat(the robot used for RatSLAM experiment) and call them Lingodroids [53]. Related literature shows even heterogeneous robots with different environment perceptual abilities and hold different map models could share their cognition in spatial knowledge and temporal knowledge using communications of words from their self-centered lexicon abstraction at meeting place. The Lingodroids offers incentive thought to us with respect to design of communication mechanism in the robotic navigation and mapping system and literature in this area will be reviewed in Section 2.5. And thesis [54] does the similar work in aspect of robotic semantic communication, swarm robots build the consistence lexicon descriptions of food source state in foraging tasks with evolution approach.

2.3.4 Frontier Area of Visual SLAM and Its Future

Just like the literature being reviewed in subsections before, visual SLAM method, which has been concluded going through periods of classical age(1998 - 2004), algorithmic-analysis age(2004 - 2015) and now experiencing the robust-perception age [36], has reached an satisfactory maturity that could handle loads of scenarios and fulfill tasks of localization and mapping precisely in real-time. But researchers have never taken SLAM as the solved problem because the versatile solution fit for all hardware conditions is still unavailable and visual SLAM still is prone to collapse when facing violent dynamics either from views of robot motion or environment. So works have being done toward directions of robustness. Failure sourced from data association(negative loop detection as the main reason), un-modeled dynamic in the mapping period(e.g., illumination condition changes), hardware failure, etc. As solution, future SLAM system should have failure awareness, failure recovery strategies and auto tunning parameters for loop closure recognition. Researchers also suggest SLAM system with scalability. Efficient map store method, partition computation consumption to multiple cores of processor(e.g., submapping algorithm), discarding and recalling mechanism all make contribution to possible solutions. For mapping tasks of long-term operation and broad area coverage, distributed SLAM with multiple robot as we proposed is necessary. Different levels and different models of map representation ranging from feature based sparse landmark map to direct dense constructions and high-level object oriented topological representation are the options with which we can build our system, not merely a single layer system but can be multilayer with different models of map overlapped. Further on, adaptive model choosing mechanism can be designed regarding to different tasks. Research targeted on cognition mapping is also called active SLAM, with control strategies eliminating uncertainty in localization and mapping [55]. Methods used in swarm robotic exploration and mapping also suit well for solving active SLAM. Frontier allocation [20], potential field(e.g., entropy)[56] and method computing utility of information gain [57] are possible solutions. Visual SLAM method always partially depend on its hardware condition, camera the most. So innovations about the algorithm may also come from new types of it. ToF(Time of Flight) camera like Kinect Version2 is the kind of camera could sample depth information directly [58]. Plenoptic camera provide directions of light rays besides pixel intensity, which will make loop closure recognition more reliable [59]. Event based camera updates inputs depending on changes happen in every pixel instead of updating in certain time interval [60]. And this update rule might change key frame creation and insertion mechanisms. Unavoidably deep learning method is considered, visual odometry is implemented in a neural network manner from frame-to-frame pair [61]. Neural network also works as the module for object recognition in semantic SLAM. In conclusion, future visual SLAM should make progress in robustness, scalability, task driven and cognition.

2.4 Multiple Robot SLAM

Researchers in the robotics community have long tried to extend methods that work on a single robot to multiple robots. Advantages of this is obvious, same as benefits from the swarm robotic system. A multiple robot SLAM system will improve efficiency of navigation process and accuracy of mapping result. In addition, when implement system in fully distributed construction strong robustness could deal with individual failure as we discussed before. But multiple robot SLAM is much more challenging in areas like map fusion and coordinated navigation.The most demanding issue in a multiple agent system is always how to design the communication schema, which include formation of data interchanged, communication network structure, communication protocol and so on. Hardware conditions relates to communication bandwidth and available communication range are always the bottleneck of system design, which will need real robot experiment feasibility confirmation. The information that is exchanged between the robots can be divided into two categories: the raw sensor input and the processed map. Communication of raw sensor input is more informative as reserving all data but it takes much more bandwidth and computation resource [62]. Map fusion is the more controversial way of data interacting. According to [63], the architecture of multi-robot SLAM can be divided into the following types,

- Centralized architectures, individuals in system transfers raw sensor input to central processor which hold global communication with every individual. Real SLAM algorithms executes on the central processor, releasing computation and memory burden of individual.

- Decoupled centralized architectures, individuals perform visual odometry and maintain self-center coordination based on this odometry. But loop closure job is only done in the central processor and optimized pose estimation is fed back to individuals. So this system could still work if individuals lose contact with central processor in a short term.

- Distributed architectures, every individual runs minimal SLAM system on board, so totally off-line individual still works normally just like single robot SLAM. But there is still a central processor for global map fusion.

- Decentralized architectures, no central processor is included, so every individual functions equal and maintains incomplete copy of global map. Entire system operate in strong robustness, but a sub-optimal solution may usually be executed because of the local information individuals acquire.

While extending SLAM to multiple robots, new problems arise beyond the single SLAM method. Reviewing literature leads to problems as follows [64]. Robots are always initialized from unknown relative poses, relative transition and rotation from other robots are necessary for transforming maps into self coordination when doing map fusion. And noise distribution estimation is also required, a noise elimination should be performed when direct observation happens. Updating global map and global loop closure should take data from both self perceived data and communicated data. It is highly possible that there are duplicates of the same landmark due to the uncertainty of the position of the landmark. Computation power restriction and communication availability need to be considered in high priority, so we always pursue a hardware friendly and high level distribution system. System performance evaluation method is also hard to design, inherent challenging problem with SLAM to compare trajectory and environment modeling.

We can expect, the fusion of 3D maps will certainly take much more storage and computation power than the 2D case, and as trade-off between abundant environment representation and restricted hardware feasibility, we will narrow down to 2D map fusion that represent the planar metric. Map fusion problems can always be decomposed into relative estimation and alignment of received maps. According to [65], literature about map fusion can be categorized to four types,

- Known initial configuration, this is the most primitive method require full knowledge of every individual initial configuration. Maps are fused based on known initial poses and later odometry estimation. But the initial configuration and always available communication of odometry contradict to the distributed essence we pursuing.
- Rendezvous, this is the method explicitly arrange robots meet at certain points. Relative pose is estimated through line-of-sight observations. But this method will introduce new problems of coordinated motion required by rendezvous.
- Relative localization, instead of a explicitly meeting with each other, robot transfers coordination of other robots into its own map. Without designing rendezvous, relative localization is more flexible. But it still need appearance of other robots in map of every individual and this method is prone to fail when false positive hypothesis about relative localization happens.
- Map overlays, the spatial repetition of map process is used for estimating alignment of relative pose. Without requiring either coordinated motion and consistent appearance, matches between maps is really a challenging work and it requests high accuracy of individual mapping result.

Like the literature we reviewed before, multiple robot SLAM are always developed as extension of classical SLAM problem. Reflecting single SLAM algorithms, we will start again from filter based SLAM to graph optimization SLAM, learning how are they extended to multiple robots manner. Because the EKF-SLAM method modeling SLAM process in equations of state space transition, it is quite easy to introduce more robot by linearly adding equations. For solving the duplicated landmark in merged maps, the Sequential Nearest Neighbor Test is performed testing the Mahalanobis distance of landmarks [66]. For particle filtering SLAM, Rao–Blackwellization of multiple robot poses are executed at their first meeting. Line-of-sight observation later is ignored and trajectory later is also considered as decoupled. The observation model is optimized by ray tracing [67]. Graph optimization based SLAM method will add edges between maps when direct encounter of robots or indirect encounter(to the same landmark) happens. A decentralized data fusion method have been proposed to address the distributed graph based SLAM problems using constrained factor graphs [68]. Scalability and robustness are demonstrated in simulation experiment.Submapping techniques are invented to reduce the cost of computation in SLAM. By looking into structure of SLAM system, they divide down the global map into submaps. Compared with the multiple robots SLAM system it is a serial process building submaps while multiple robots SLAM is a parallel one.

Coordinated motion method CPS(the cooperative positioning system) [69] subsequently moves two group of robots. The moving group called parent group takes the other group of robots(called child group, remain static) as landmarks so they get precise localization while using perceptual units mapping the environment. Manifold representation models the 2D environment with a spiral in the 3D space. Map inconsistency happens in traditional planar representation method when robot retrieve to places traversed before and this is caused by errors accumulating during the navigation. [70] propose to model multiple robots maps with manifold representation achieving lazy loop closure. And that means part of the map will be merged as the same trajectory after pretty strict checks about the matching accuracy. This not only save computation resources taken for point-to-point loop closure but also eliminate lots of false positive about loop closure detection. Topological representations like probabilistic generalized Voronoi diagram in [71] pre-process the map and abstract it in logical level. Experiment result confirm that if the abstraction reserves enough information for designed tasks, topological method could help fast and accurate map sharing. The most intuitive method of evaluating output from SLAM system is to compare generated trajectories or maps with the real traversed trajectory or concrete restoration of real environment. But this is always hard to implement, because either a full coverage overhead camera or precise cartographer is needed. And these two usually is hard to implement in scenarios demand for SLAM. [72] propose to measure identities of generated maps with ground-truth map by abstracting both of them to topological Voronoi maps. This method ignores unnecessary details about the map and boosts speed of map comparison. Considering the difficulty in even single robot SLAM result evaluation, we could realize more works need to be done when try to extend the evaluation to multiple robot SLAM. In our proposed system, apparently maps should be constructed with certain level of accuracy and efficiency of map area coverage is also considered. Thinking from how the robots require from SLAM, tasks are usually bounded to the mapping process. Similar to what is done in evaluating swarm robotic algorithms, we could assign tasks like foraging to the group of robot. The quantity judgment of evaluating mapping efficiency then should be time taken or distance passing to find the food source. As for evaluating mapping accuracy, the efficiency of foraging after one robot successfully build connecting path between nest and food source could reflect how precise the map is constructed and transferred among robots.

Therefore, when multiple robots are used to map the environment, mechanisms are needed for the robots to cooperate and exchange information between themselves. For the system hierarchies, we would like to choose the decentralized organization method. Because we want to implement the multiple robots SLAM system with scalability and robustness. And from the view of exchanged information, we would like to choose more high-level abstracted representations like the topological maps or even semantic maps reviewed next in 2.5 as they are compatible to the decentralized system structure, still offers sufficient information and demand for least communication bandwidth.

2.5 Semantic SLAM

Firstly think about the mapping problem from the perspective of our human behaviors, for most of the conditions we do not need a detailed constructed map from those high accuracy measuring instruments. When a child try to remember the routine from home to school or when native try to guide road for tourists, semantic descriptions are often used based on their common sense knowledge, like following current path and turn left/right and also abstractions about time will be taken are provided sometimes. Back to the point of view from robots, like what happens with the RatSLAM [53], an extracted global topological map or a spatial semantic hierarchy could behave like semantical descriptions shared between human beings. The communication between robots also do not necessarily cover every feature point they extracted. Instead, task oriented and robot common sense based semantic maps will help ignore redundant information and promote map sharing efficiency.

Starting from the simple robot without camera like we did before, robots still could extract key points which used for constructing spatial semantic hierarchy during their navigation. In [73] the spatial semantic hierarchy is composed of distinctive places and distinctive travel paths. The robot NX in simulator has sixteen sonar distance sensors installed around the body covering 360 degrees evenly. Distinctive measure from these sensors like distance to objects in vicinity, symmetry based on the self-centered coordinated of robots and temporal recording about obstacles could let the robot decide it is near to a distinctive point. And from there a hill-climbing search method will lead robot to the distinctive point. Local control strategies, follow middle of two walls, move along the left/right hand side obstacles or keep walking straight control the robot finishing the distinctive path and leading to next distinctive point. The experiment is targeted at exploration, robot would record the direction it leaves from every distinctive point and keep other directions in travel agenda for further exploration when return to this point next time. Local features may repeat in different positions while the entire exploration process, so in order to avoid confusions about local distinctive points, a rehearsal procedure will compare points and paths meeting in the next few steps with those points and paths after the distinctive point recorded in memory to confirm the robot returns to the same distinctive point traversed before. The experiment in simulation successfully coverage entire mapping arena and the introduce of random error to sonar sensors prove the spatial semantic hierarchy promote system performance on robustness.

A more state-of-art solution to the semantic problem is about how to label maps build with exist SLAM methods. In [74], the author implement semantic mapping by combining a real-time mapping(such as gmapping) with the place categorization to build an overlapped map. ConvNets(Convolutional neural network, Places205 [75] network in this paper) is applied for classifying every image captured. And considering the drawbacks of ConvNets, that only could recognize already trained classes, a computationally cheap one-vs-all classifier help to classify a new door class in the experiment. In addition, different from classification work done in computer vision community, the process of robot navigation is temporal continuous. So this paper also introduces Bayesian Filtering for testing temporal coherence of categorization. Case study is performed on two two applications of the semantic information. Object recognition speed and accuracy is improved providing the semantic mapping information. For example, the robot will have a high belief of detecting objects of rectangle shape in an office as books. The other application is about cognition path planning. In the experiment, robot is required to navigate across places like corridors and offices. With the semantic context, robot could generate different paths according to corridors should be crowded or empty in different times of a day. And this is implemented using A^* search algorithm with cost values from different semantic classes [76].

Apart from these methods of extracting semantic information from visual features like shapes, colors and textures, [77] proposes a method of studying the relationship of human-object interactions. Through acquiring data from human wearable motion tracking sensors, the robot make inference about human activities based on models trained off-line in advance. And then the probabilistic distribution of the furniture type is estimated. According to [78], methods of semantic SLAM can be classified as,

- Semantic information inferred from single cue, sole source of perception is used for annotate the areas robot is mapping. And depending on the system architecture, it is further separated into Scene Annotation(semantic characters overlay on the metric maps) and Pixel-Wise Point Cloud Annotation(semantic partition of 3D point clouds).

- Semantic information inferred from multiple cues, usually a combination of visual features, temporal cohesion, object recognition etc.

In [79] the topological and spatial structure of environment is expressed in the form of symbolic semantic phrases, like "B is down A" or "C is beyond B".The semantic descriptions about the environment layout is interpreted by a novel spring-damper system. Through reasoning the symbolic semantic description and observing grounded information, the door label, robot could navigate to unknown places with small bias against the optimal path.

The semantic SLAM has advantages in areas more than robot map sharing, the semantic abstraction makes the map more human comprehensible and on the other hand the raw command with high level abstractions sourced from human thoughts could easily taken by robots. Both aspects make contribution to refining the interface of robotic mapping and navigation system to human beings. And thinking about organizing semantic factors lead to the study of knowledge model in section2.6.

2.6 Knowledge Model

Semantic mapping do the work of extracting high level information from robot navigation process, but we still need a kind of processing mechanism to record and organize the information we stored. In addition, from the views of linguists, the research of semantic is studying meanings of words and the meanings are often interpreted by relationships between the words. So we may propose the knowledge model, metaphor from human perception and reasoning behaviors as solutions for the problem. And the ontology [80] from the philosophy study is applied as the frame of shared hierarchical concept. Researchers plan the generic knowledge model from the view of whole robotics community and propose the ontology for knowledge model as [81] In that model, all robotic semantic concepts would fall into five categorization of Actions, Contexts, Resources, Events and Actors. DL(Description Logic) is applied to describe the logical relationships among concepts, specifically the OWL(Web Ontology Language) which stores DL equations in an XML-based record organize [82]. Similarly, in [83] the KnowRob knowledge processing system used by RoboEarth project [84], also store the concepts with the OWL files. But developing from the need of describing robotic tasks, it classify knowledges into uppermost branches of temporal things, actions, spatial things and mathematical concepts. Prolog is chose as the intermediate language doing new knowledge recording, existing knowledge recalling and probable connections inferring. Prolog, in specific the SWI-Prolog is a kind of logical programming language for recording first order logic. And it outperforms other reasoning engine in aspects of speed, external dataset incorporation and generalization. In KnowRob, bootstrapping knowledge is pre-installed and new knowledge learning could be accomplished through robot-to-robot communication, observation of human behaviors, import of natural language task instructions or even sources on the Internet. The system performance is evaluated through scalability(object instance) and response time assessment. And this knowledge processing system is open-sourced as a ROS package for usage and learning [85].

The knowledge model focusing on the robotic navigation task is described in [86]. For domestic scenarios, physical rooms are used to label regions in the topological map and physical objects are used to describe entities that the robot could recognize and augment semantic meanings. Both of the physical rooms and objects are connected to real elements recorded in the metric map. Conceptual rooms and objects are correspondingly abstract concepts, like living room and washing machine,etc. Like the logical relationship recorded with Prolog, this system maintains tables storing links between these physical elements and conceptual elements. Semantic relational links are also stored in tables, they are define in four classes,

- Interaction, logical relationships and spatial relationships consist of belonging, used with or inside of.
- Utility, description defined by human behaviors, like drink, wash, play, etc.
- Meaning, descriptions from human emotions, like comfortable or relaxing.
- Characteristic, attributes of objects make them different, like cold or warm characteristic of water.

Semantic navigation targets set for the system could be known or unknown object and room, the robot has to make queries about the data-set and make inferences about potential positions of targets if it is unknown. Priory knowledge is fulfilled by the programmer, but new relationships could be constructed by interaction with human or acquiring external knowledge databases.

2.7 The Searching Problems

2.7.1 Searching Problems in Different Disciplines

The problem of searching can be interpreted differently by researchers from different disciplines beyond robotics, e.g. biology. Although the motivation of researchers and application context may drastically vary in robotics community, research outcomes including models and algorithms contribute to the design of robotic systems.

According to [87], the animal foraging behaviours is analogous to robotic searching tasks. Results of most reviewed works conclude that robot should subdivide the searching arena into small patches. Due to the existence of patch boundary, detailed search can be conducted by animals (robots) inside each patch. However, Jesus and Robin [87] also argue there is fundamentally different motivation between animal foraging and robotic search tasks. Exhaustive search should be performed by robots while animals try to maximize the energy level of their nests to stay alive.

The study of [88] is based on the searching behaviour of T cell in the immune system. Specifically, T cells search for dendritic cells in lymph nodes for the purpose of initiating the adaptive immune response. The requirement of efficiency, scalability, robustness against errors and flexibility can be shared with robotic searching tasks. The experiment result with a robot swarm suggests efficiency of Lévy-type random walks which models the T cell searching behaviour depends strongly on the distribution of targets.

Early application of robots in search and rescue scenario appeared in the 2001 World Trade Center (WTC) collapse [89] for rescuing victims after the tragedy. The deployed robots are equipped with abundant sensors so as to develop a versatile mobile robot platform in extreme operational environments. According to [90], the control of current rescue robots mostly relies on tele-operation by human operators. The study toward autonomous control scheme has been extended hierarchically. Low-level behaviours include robots traversing uneven terrain and SLAM to build maps of the search and rescue scene. On the high-level, collaboration and task allocation both between human-robot and robot-robot are the directions worth exploring.

Review paper [91] surveys robots used for localizing source of pollution in environmental monitoring applications. Chemotaxis (gradient-based) and anemotaxis (wind detection) methods show weak resistance to propagation turbulence. So infotaxis methods are devised based on information principle with the propagation model of pollutant in turbulent medium. While the works reviewed are designed for single robot source localization, applying them into multi-agent systems also has been envisaged as the future direction. [92] also categorizes works on robotic odor localization by the environmental conditions, which determines how the odor is dispersed. Reynolds number of the flow is used to quantify different fluid flow situations. In low Reynolds number flow, the fluid motion is dominated by diffusion and the dispersal of the chemical concentration can be modelled by a Gaussian distribution. Searching robots may come across this situation when they work underground. But in high Reynolds number flow, turbulence dominates, which is common in aerial and aquatic conditions. This may cause problems of sub-optimum.

2.7.2 Robotic Searching Task Variants

The problem of target searching can be configured with different set up in robotics research community. Parameters and assumptions vary in different works, e.g. number of targets or the mobility of targets. We try to classify searching tasks applying robots by referring the work of [93] and [94]. This will help us narrow down the design of search problem and concentrate on certain scenario.

- Number of targets. When searching for single target with MRS (multiple-robots system), most research are focused on improving the accuracy of estimated target position. However, in multiple targets scenario the main attention is paid to the assignment of robots to a target by certain mechanism, e.g. utility. The problem also diverges when the ratio between the number of target and robots changes.

- Mobility of targets and robots. The complexity of searching problem will be increased if target could move. And the moving pattern (range, mode and models) of target may be diverse. Relatively, various robotic platforms have emerged for the purpose of searching with different modes of mobility, e.g. wheeled robot moving on ground, biomimetic robot swimming underwater or quad-rotor flying in the air.

- Complexity of environment. Environment acts as key role in robotic searching tasks as most of the algorithms rely on sensory perceptions from it , e.g. chemotaxis method in robotic source localization application. The arena of searching can be defined as open space or with boundaries. This makes difference at the time choosing path-planning method. In addition, the existence of obstacles also will collide with robots' trajectory.

- Type of cooperation and coordination among robots. The cooperation of multiple robots targeted at either improving the accuracy of target estimation or optimally allocating robots to targets. The approaches coordinating robots can be categorized into explicit or implicit coordination. Though implicit coordination has its drawback of unclearance, the brevity of information saves the communication bandwidth and guarantees the scalability when the number of robot increases.

- Evaluation criteria. The evaluation criteria depends heavily on the specific subtasks of searching problem. In situations where the target is perceived by robots with uncertainty, the tracking accuracy is often used as evaluation metric. On the other hand, if robots can clearly recognize the target, previous works tend to measure consumed resources before achieving certain criteria. [95] defines first passage time, which is the average time every robot takes to pass by the target. Besides, the efficiency of information sharing is also assessed through the defined convergence time. It is the time taken for all robots getting direct or indirect access to the target. In addition, by drawing lesson from works on swarm robotics aggregation, the time taken for leading the robots to converge near the target might be optional criteria, and the convergence can be judged by average robots' distance to the target. Apart from time, the total travel distance of all robots can be recorded as a measurement of energy consumption for comparing searching algorithms.

2.7.3 Algorithms for Swarm Robotic Systems

The advantages of multiple robot systems (MRSs) over single-robot counterparts in searching applications is shown by the parallelism in space and time. Swarm robotic systems (SRSs), being part of MRSs specify the study of collective behaviour emerging from local interaction among robots and between the robots and the environment [96]. [96] also distinguishes SRSs from other MRSs by sets of criteria, e.g. the study should targeted at scalability, relative incapability of individual robot and the robots' sensing (communicating) should have limited range. Because SRSs do not rely on centralized control architecture, it scales well with the quantity of robots. When individual robot fails, other robots will try to make up the loss by proceeding the work of failed one [97]. In addition, the stochasticity maintained by individuals makes contribution to self-adaptiveness to changes in environment. These merits of SRSs fit perfectly to searching tasks, especially when the searching efficiency is highly demanded and robots operating conditions are usually severe.

Referring [93], algorithms for searching tasks using SRSs is classified as swarm intelligence (SI) based algorithms and other algorithms.

2.7.3.1 Searching Algorithms Based on Swarm Intelligence

The analogy between robotic searching problems and the problem of optimization is found by [93, 98, 99]. To be more exact, how the particles/agents behave in SI algorithms for solving optimization problems resemble the way individual robots working in SRSs. As a result, lesson could be drawn from SI algorithms for designing SRSs.

- Particle Swarm Optimization. The original algorithm of particle swarm optimization (PSO) was firstly proposed by Kennedy and Eberhart in 1995 [100]. The algorithm is designed to solve optimization problems with a set of particles. Each particle holds a candidate solution. The updating rule of particles is known as mimicking the social behaviour in bird flocks or fish schools. A particle synthesizes its inertial velocity and two vectors, one pointing to its best position and the other pointing to neighbours' best position. There are two constants acting as weights balancing the influence from best solution from memory and best solution from other particles. And in [101], weight is also added to inertia velocity, in order to trade-off global search and local search.

  The early work of [102] implements PSO algorithm on swarm robotic platform. However, only the robots' communication range is limited while the robots still need to acquire global coordination. In their later work, the global coordination is replaced by robots' short memory of last position.

In order to lead the robots to aggregate at the target position, [102] also proposes the robot sensing very strong signal of the target should stay as a static beacon for the subsequent robots. With regard to the weight augmented to inertia velocity, they suggest it can be designed to be adaptive, which decreases when robots approaching the target.

Huge amount of literature can be found on variants of PSO inspired SRSs searching algorithms. We will not list all of them, but see what the state-of-art variants doing by a review paper [98]. The work in [103] proposes Extended Particle Swarm Optimization for dealing with real world constraints. Robots use the Braitenberg obstacle avoidance algorithm as the position updating approach if they meet obstacles. The Physically-embedded Particle Swarm Optimization in work [104] considers the angular acceleration of robots can not be infinite. This means that the robot can only turn a limited angle at a time. The Robotic Darwin Particle Swarm Optimization (RDPSO) algorithm firstly developed by Couceiro, etc. extends the original algorithm to support multiple dynamic swarms [105]. In RDPSO, if solution of the sub-swarm has not been updated for a determined time, the worst performing robot in this sub-swarm will be excluded.

- Bees Algorithm. As being described before, searching behaviour happens when animal forage. Bees algorithm (BA) was firstly designed by Pham et al. mimicking bees searching for food [106]. The scouts are randomly initialized in the searching area, and flower patches are defined in the vicinity. Foraging bees are recruited to flower patches according to their quality and execute local search. If no improvement has been found by foraging bees in a flower patch, the size of the flower patch shrinks. And if this happens consecutively for certain iterations, this flower patch will be abandoned and new scout is spawned. Work in [107] introduces the Distributed Bees Algorithm (DBA). The original global recruiting process is replaced by a probabilistic based distributed counterpart. The probability recruiting a foraging robot to a flower patch is based on utility of the patch, which is positively relevant to the quality estimated by the scout and negatively relevant to the distance between the scout and foraging robot.

- Ant Colony Optimization. In [108], the Ant Colony Optimization (ACO) algorithm is proposed. The ACO algorithm shows how a consensus on shortest path leading from ant nest to a food source is reached. Ants leave pheromone on their trajectories and the pheromone evaporates constantly. Ants will be more likely to choose the direction has higher density of pheromone to move when they plan their path.

The application of ACO onto robotic platforms faces difficulty in implementing pheromone in the real world. Although physical marks, e.g., alcohol, heat, odor or RFID tags could be substitutions for pheromone, work in [109] proposes a neater way through inter-robot communication. Some of the swarm will be assigned to be static beacons for depositing and transmitting artificial pheromone. However, in the work of [110] pheromone is carried on every robot. Robots' searching is splitted into local traversal search and global search for promoting efficiency. The global search has modes of random search and probabilistic search. If the best solution in communication range is still lower than determined threshold or the worst solution reaches a really low value, random search is activated. On the contrary, the probability of moving toward another agent depends on the quality of position and the distance.

- Bacterial Foraging Optimization and Biased Random Walk. The chemotaxis behaviour of bacteria inspires the Bacterial Foraging Optimization (BFO) algorithm [111]. The locomotion of bacteria is modelled by combination of swimming (moving in fixed direction) and tumbling (spinning a random angle). In the original algorithm, the proportion of bacteria with bad solution will be eliminated and reproduced. This strategy is infeasible in robotic implementation. And the searching efficiency hardly can match other algorithms, because no messages between agents are transmitted. However, [112] adopts BFO for mapping the distribution of chemical in an unstructured scene.

  The Biased Random Walk (BRW) algorithm is also inspired by the chemotaxis behaviour of bacterium like BFO [113], including the movement model. The difference is that in BRW if the improvement of quality has been detected (which requires a very short memory), the swimming length is increased and the probability of tumbling is decreased. The work in [95] presents the analysis of different random walk modes. In the common scenario where boundaries exist, correlated random walks (CRWs) outperforms Levy walks. And this can be explained by the influence from collisions with walls.

- Glowworm Swarm Optimization. Agents in Glowworm Swarm Optimization (GSO) algorithm hold a luminescence quantity known as luciferin [114]. Like pheromone in the ACO algorithm, the quantity of luciferin decays by time, but is enhanced by the fitness value of agents' current location. Agents continually perceive their neighbours' brightness inside limited range. During the motion planning stage, the position of brighter neighbourhood will have a higher probability being targeted by other agents. This decision making stage also happens in a delimited range named as local decision range, which is inside the brightness perception range. The original local decision range updating methods in [114, 115] both extend the range when number of neighbour decreases and vice versa. But the later one is described to be smoother by defining an explicit threshold. The work in [116] also explores the maximum relative speed between the mobile target and the pursuing agents for accomplishing successful tracking with GSO algorithm.

- Firefly Algorithm. The Firefly Algorithm (FA) deploys the same medium for inter-robot communication as GSO, the light emitted by agents [117]. However, the attractiveness of neighbour is modelled by combining distance besides the brightness. In addition, the movement of agents is updated by the average of the delimited neighbours' attractiveness and randomness instead of probably following one neighbour.

  The coefficients for brightness absorption and randomness are designed to be adaptive in work [118]. Random motion is expected to dominate the path planning at the start of searching for spreading agents to all the possible position. But at the later period of searching, the demanding of a quick convergence calls for neighbourhood attraction to be dominating. The work in [119] proposes the modified FA, in which randomly picked fireflies are replaced by new generated counterparts around those nearer to target.

- Swarm Environment Based Aggregation Methods. Aggregation is a very common task in swarm robotics research, requiring agents to gather in the working area. In [120], a subcategory of aggregation which lead agents to some preferential regions is reviewed. Insects, e.g. bees and cockroaches could aggregate to place more suitable for their living habit in terms of temperature, brightness or humidity.

  The BEECLUST algorithm in [121] is inspired by the aggregation of young honeybees at warm zones in their nests. A honeybee executes totally random walk until it meets another one. Then this honeybee stops for the duration of time decided by local temperature of the collisional site, warmer site causes longer stop time and vice versa. In the experiment setups of robotic research, the temperature distribution is replace by the luminance of a light source. Later work [122] also adapts the moving velocity of robots in accordance with the local luminance to improve the aggregation efficiency.

The aggregation behaviour of cockroaches inspires the work in [123] analogous to BEECLUST. However, the duration of stop also depends on the quantity of neighbours. In this way, ideal number of cockroaches are gathered in certain spots considering the size of that area. This could be an indication for designing task allocation method in multiple targets scenario. The work in [124] shares some similarity with the variant of ACO in [110]. Each agent carries a scalar in correspondence with quality of its current position. Only when the scalar is better than a designed threshold, agent tries to approach the local best neighbour.

2.7.3.2 Searching Algorithms Based on Other Methods

The work in [125] proposes a distributed Kalman filter (DKF) method for updating the estimated position of a single moving target. Robots combine the tracking of target and an artificial potential function to implement a flocking controller. The potential field based methods are also presented in work [126, 127]. The interactions between robots are modelled by repulsive forces and the navigation toward target is driven by the attractive forces. Another work in [128] takes morphology related approach to track multiple moving targets. The robots can not directly sense targets are recruited by those robots who are following targets to maintain equilateral triangles.

In conclusion, most of the reviewed algorithms rely on the timely quantified evaluation of robots' positions. However, this is only available in those odour localization scenarios, where the robots are equipped with precise sensors to distinguish even trivial variations of gas density. For a generalized application, swarm robotic systems needs a concise mechanism for recording the searching environment and for communication.

2.8 Statistics Analysis

This section reviews the methods for statistically analyzing the results generated by the experiments in this work.

2.8.1 Significant Difference Test

The experimental results of different set-ups should be compared as statistically significantly different. The methods are reviewed as follows,

- Mann-Whitney U-Test

  The Mann-Whitney U test is a nonparametric test of the null hypothesis that the distribution underlying two sets of data are the same [129]. The test can be used with directions, to compare the distribution underlying one sample is stochastically less than another. In the outputs, the p-value indicates a reject of the null hypothesis or not.

- Vargha-Delaney A-Test

  The Vargha-Delaney A-Test computes the Vargha and Delaney A effect size measure. It quantifies the difference between two groups of data. The score of A-Test is compared with certain threshold values to describe a qualitative assessment of the magnitude of effect size [130].

- The independent T-Test

  The independent T-test is an inferential statistical test, which exams the statistically significant difference between the means in two groups. It is calculated with the null hypothesis that the means from the two groups are equal [131]. The outcome includes a p-value for rejecting the null hypothesis and the difference between the means.

2.8.2 Correlation Test

The correlation between knowledge learning and the efficiency of searching will be explored in this thesis. As we presume it to be linear, the method to determine such correlation is found as the Pearson correlation coefficient (r). The p-value generated by the test will be used for rejecting the null hypothesis that the distributions underlying the samples are uncorrelated and normally distributed [132]. It also returns a number between -1 and 1, measuring the strength and direction of the relationship between two groups of data.

2.9 Summary

This chapter has reviewed the topic of swarm robotics and the techniques required for searching tasks, which are the strategies for exploration, the mapping methods for recording the searching environment, and the models for managing semantic knowledge. The searching problems are discussed in detail. Finally, the approaches for statistical analysis are reviewed.

Chapter 3

Searching Task and The Fuzzy Cognitive Map Inspired Robot Controller

As reviewed in the chapter 2, the target searching problem falls into different domains of robot navigation, localization, and mapping, multiple robots cooperation, etc. This work can not cover the whole range of sub-problems. Instead, we focus on building a robot controller, which could manage the semantic knowledge for the task of searching.

This chapter discusses the definition of the searching task. By listing the limitations and requirements, a detailed searching task is shown in a simulator. The relevant hardware components of E-puck mobile robot are provided to present their function in the semantic knowledge based searching. The robot system architecture is constructed by analyzing the search problem from the top level and the robot behaviours are implemented from the bottom. At last, the mechanism for semantic knowledge management is presented.

3.1 The Searching Task Deploying Semantic Knowledge

The working arena of robot searching for a target should just be like places we reside every day, for example an offices or an apartments. The whole arena is divided into separate rooms with areas which could be considered as gateways connecting the neighbouring rooms. In each room, the relevant objects will be found according to the type of the room defined by human. The target in this work should be in an area where the robot receives a unique sensor perception.

The decentralized robot searching algorithms like the particle swarm optimization [133] and the bees algorithm [134] rely on acquiring robot's global X-Y-Z coordinates to evaluate the quality of their current position in real-time. However, we expect the designed system to work in more generalized applications where global coordinates might not be feasible. Therefore, those prevailing path finding algorithms like A* search algorithm [135] or Dijkstra's algorithm [136] are not available.

The target was initially proposed to be a source of gas. The human operator deploys robots to search for the source of gas by providing semantic knowledge about objects and rooms. The operator also indicate a possible room by his experience on the gas. For example, the gas source which has a smell of rotten food might be found in the room of kitchen. As we are keen on how the semantic knowledge about the environment promoting the searching efficiency of robot, the deployment of gas sensors which returns high resolution gas density will become non-essential. Therefore, this work will not expand on those chemotaxis (gradient-based) and anemo-taxis (wind detection) methods, which also show weaknesses on resisting against propagation turbulence [137].

The work on object recognition is also simplified in this thesis. There are lots of packages like yolov5 [138] which are open-source and effective. As stated earlier, the low-cost robot we will be using has out-dated on-board sensors and limited computational resources because the swarm system will deploy a large number of robots. And the object recognition only serves as one of the sensor inputs and does not involve into the architecture of robot controller. Therefore, we replace the object recognition module by the distinguishment of the colours.

The experiments will be carried out in simulation. This will save enormous time on constructing the large arena for searching and dealing with the out of date robot hardware.

3.1.1 The Searching Task in The ARGoS Simulator

The ARGoS simulator [139] called Autonomous Robots Go Swarming, is a prevailing multi-physics robot simulator which is dedicated for running simulations of swarm robots. The feature of parallelism offers a big advantage when running experiments with a large swarm size. It is open-source and actively maintained. This work uses the simulator in version 3.0.0-beta56.

In the simulator, we build up the searching scenario as follows,

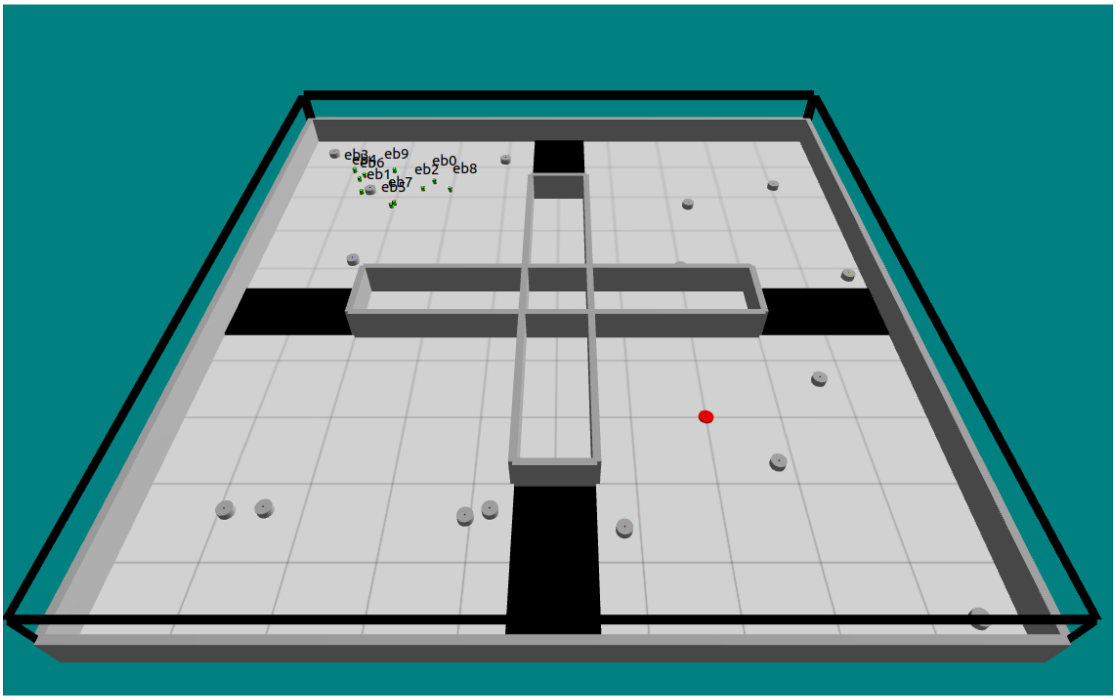


FIGURE 3.1: The Arena Built in The ARGoS Simulator For The Searching Task.

- Rooms are built by boxes as walls, each in size 5 meters * 5 meters.
- Gateways are the gap between walls which connect the neighbouring rooms. The colour of floor is shown as black in the figure 3.1.
- Objects are presented as cylinders with coloured blob on the top. The blob colour is unique and meant for object recognition. The radius of each cylinder is 0.1 meter. The placement of objects are randomized. But the number of objects in each room is kept as 4.
- Target is the place where floor colour is painted red, with radius of 0.1 meter.

3.1.2 The E-puck Robot Model

The robot in use should have the abilities of free moving in the arena, objects recognition, target recognition and communication.

Gives the requirements of robots, we find fulfillment on the E-puck mobile robot [140] which is commonly used in the swarm robotic research community. As required,

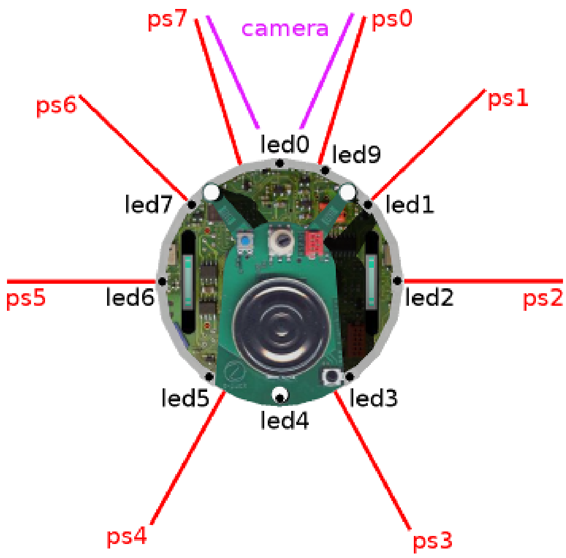


FIGURE 3.2: Proximity Sensors on The E-puck Robot, taken from [1] .

- Moving ability is implemented by two wheels driven by step motors, 1000 steps of resolution when the wheel runs a round.
- Obstacle detection requires the robot to predict possible collisions in advance. We use the four proximity sensors on the front side to detect obstacles, denoted as ps0, ps1, ps6, ps7 in the figure 3.2 [1] .
- Objects recognition is simplified by using an coloured blob omni-directional camera. This camera is not a component on the standard E-puck robot hardware platform. However, we can modify the raw entity files of E-puck robot in the ARGoS simulator to mount the camera on top of it. This camera could detects coloured blobs around the robot in 360 degrees and returns a list of blobs, each defined by its colour. The perception range, not indicated in the official document of ARGoS simulator, is tested to be less than 0.5 meter.

- Target recognition is implemented by the ground colour sensors.
- Communication is implemented by the range and bearing sensor and actuator. The communication range is set to 0.5 meter.

3.2 The Fuzzy Cognitive Maps Inspired Robot Controller

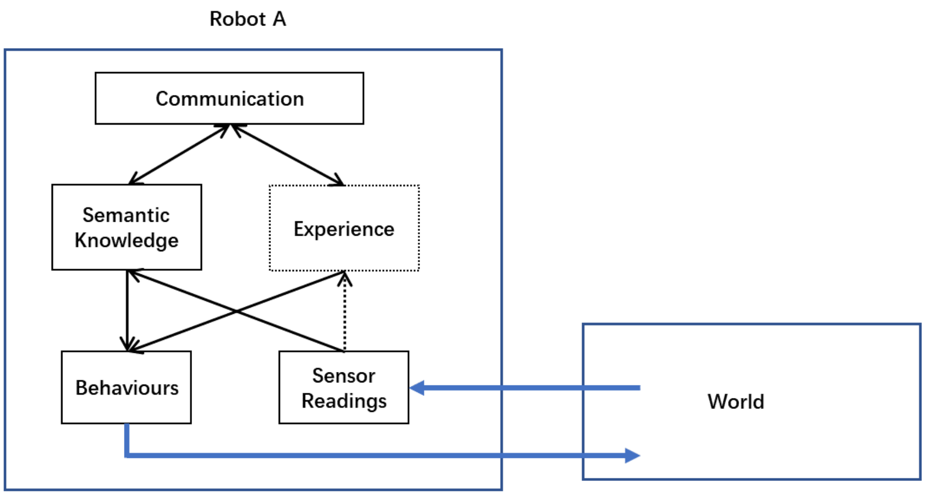


FIGURE 3.3: The Proposed System Architecture of Robot Controller for Searching Tasks in This Work

In this section, we propose a robot controller which incorporates semantic knowledge about the environment from humans when performing searching tasks. The behaviours of robot are designed toward solving the searching task by expert experience from humans. While the searching process, a robot should navigate through the arena from where it is distributed to find the target. As an autonomous agent, a robot takes sensor readings and switches behaviours through combining the semantic knowledge and the experience, as shown in the figure 3.3. The combination of semantic knowledge and experience works like the cause-effect relationships in the Cognitive Maps [141].

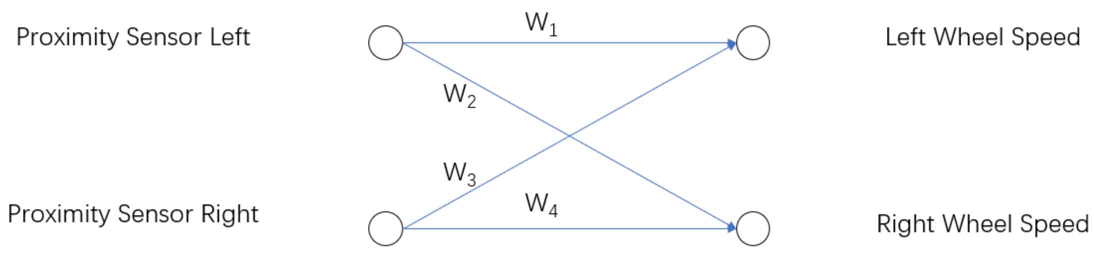


FIGURE 3.4: The Fuzzy Cognitive Map for Obstacle Avoidance.

3.2.1 The Fuzzy Cognitive Map

The system architecture of swarm robot controller based on Dynamic Fuzzy Cognitive Maps by Márcio Mendonça proposed an solution for connecting the node of sensor input to the node of actuator output for robot navigation [142]. In a fuzzy cognitive map, concepts denoted as nodes represents the factors like sensor inputs, actuator outputs and intermediate control mechanisms, for example the designed robot behaviours. Relationships denoted as directional connections are made between concepts. The connections are given weights to describe the power of relationship between nodes. The node being pointed to will update its value in accordance with the product of source node and the weight of connection between them.

As is shown in the figure 3.4, the Fuzzy Cognitive Map for a robot which could avoid obstacles is built by connecting the nodes of proximity sensors from both sides of the robot to the speed of both wheels. If an obstacle on the left is detected, the speed of right wheel decreases. In that way, the robot makes a turn to the right side. The connection is shown as the following equation,

$$V_{Right} = P_{Left}W_2 + C$$

where V_{Right} is the speed of the right wheel, P_{Left} is the reading from proximity sensor on the left side, W_2 is set as a negative value and C is a common value to keep the right wheel rolling.

Similar obstacle avoidance behaviour could be achieved using fuzzy logic with fuzzy rules in the work of Mendonça [143]. The fuzzy rules are designed to optimize the weights of connections, so that the robot achieves precise movement. However, we take the fuzzy cognitive maps with straightforward connections, as long as they can achieve the expected behaviours.

In this work, the architecture based on Fuzzy Cognitive Maps will provide an interface for injecting the semantic knowledge and the experience on searching tasks. Because the utilization of semantic knowledge will work also as the cause-effect relationship in the Fuzzy Cognitive Map. The human operator provides the semantic knowledge which are the relationships between objects and rooms and the inferred possible room by his experience on the smell of the gas, to the robot. The robot navigates in the searching arena uses the semantic knowledge and switches behaviours following the experience.

3.2.2 Robot Behaviours

As being discussed, we abandon the ideas of using the global X-Y-Z coordinates for localization and mapping. The available source of data force the design of behaviours to only includes the detailed search in the current room and go-to-next-room. With prior semantic knowledge about the objects, rooms and a possible target room inferred by humans, the robot should decide if the current room is the correct room to search or not. After that, the robot will make decision on choosing the detailed searching or the go-to-next-room behaviour as shown in the figure 3.5. The connection from Object to Room Confidence implemented by an Hidden Markov Model will be described in the next section.

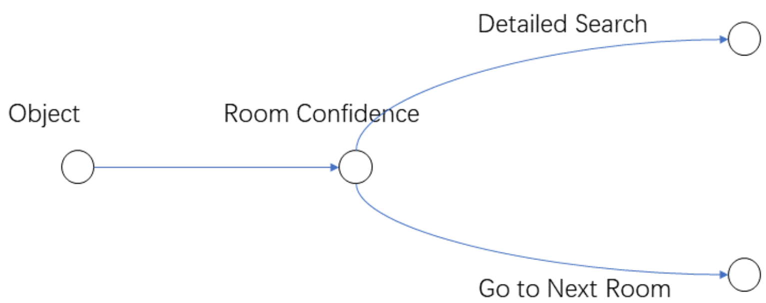


FIGURE 3.5: The Decision Making Process in The Searching Task.

The work of Márcio Mendonça in the Fuzzy Cognitive Map based swarm robotic system architecture also classify the behaviours of robots into reactive behaviours, deliberative behaviours and intelligent behaviours [142]. The reactive behaviours directly connect sensor readings to the actuator outputs, like a knee jerk reflex. The deliberative behaviours are designed with deliberative purpose, for example to navigate from Point A to Point B in the work of Márcio Mendonça. The intelligent behaviours are the ones using navigation experience from other robots or evolving the connections in the Fuzzy Cognitive Map to dynamically adapt to the environment changes.

We will make the the detailed search in the current room and go-to-next-room behaviours as deliberative (or intelligent in the after part) behaviours. Apart from that, we use a different categorization of behaviours which classifies the behaviours of robots into top-layer behaviours and bottom-layer behaviours. The bottom-layer behaviours are the ones same as the reactive behaviours, which are reactive connections from sensor inputs to actuator outputs. But the top-layer behaviours are the behaviours which are more complex than the bottom-layer ones and could be composed by the bottom-layer behaviours. The detailed search in the current room and go-to-next-room behaviours will be designed by combing behaviours from the bottom-layer.

In this work, for the purpose of searching a target, the reactive behaviours or the bottom-layer ones are obstacle avoidance, straight-line walk and random direction turning. As an experience, a robot which keeps straight-line walking is more likely to leave the room than a robot which periodically changes its walking direction. On the other hand, if the robot periodically changes its walking direction, it is more likely to run as a detailed search. So we build the go-to-next-room behaviour by using the combination of obstacle avoidance and straight-line walk. And the detailed searching behaviour is designed by running straight-line walk and random direction turning consecutively if no obstacle is detected.

The top-layer behaviours which are composed by bottom-layer ones need to manage the running of bottom-layer behaviours. In the subsumption architecture [144] proposed by Brooks, all behaviours of robots execute in parallel. The behaviours are classified by their priorities. The high prioritized behaviour can inhibit the execution of behaviours which have low priority. And in the Fuzzy Cognitive Map, there exists another type of connection which denotes the selection relationship, for implementing such mechanism. Therefore, the detailed searching behaviour is built as shown in the figure 3.6. The line with solid triangle denotes the selection relationships between nodes.

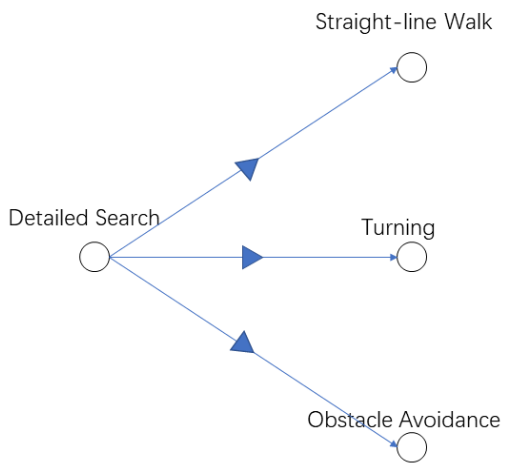


FIGURE 3.6: The Detailed Searching Behaviour.

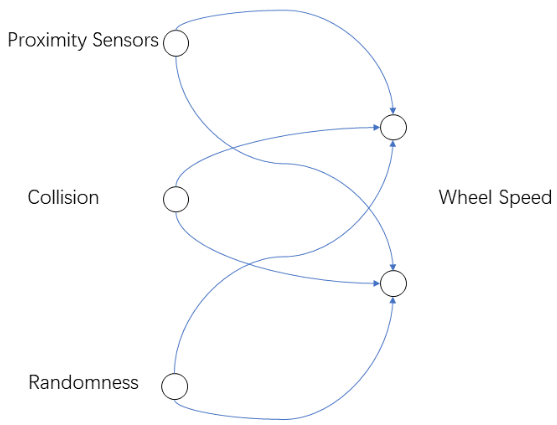


FIGURE 3.7: The Fuzzy Cognitive Maps of The Bottom-Layer Behaviours.

The bottom-layer behaviours are implemented as shown in the figure 3.7. The implementation details of them are presented in the following,

- The obstacle avoidance behaviour: The nodes of proximity sensors are connected to the nodes of wheel speeds. The proximity sensors return a value between 0 and 1, by which 1 means the robot is touching the obstacle and 0 means no obstacle detected. This value increases when the robot is getting closer to the obstacle. The weights of the relationships which connects the proximity sensor readings to the speeds of wheels on the other side in set to -0.5 as initial value. This will make the robot turn to the other side of the obstacle.

- The collision avoidance behaviour: While trying the robot in the arena with the Fuzzy Cognitive Map for obstacle avoidance, the occurrence of collision is observed. The collision happens in corners of a room, where proximity sensors on both sides detect obstacle. The obstacle avoidance behaviour can not lead the robot out of the corner by turning to the side without obstacle. Therefore, the collision avoidance behaviour is devised. The node of collision is activated when the robot detects obstacle on both sides. When activated, the collision node sets left wheel speed to 0 and right wheel speed to the full speed of 0.2 meter per second. This makes the robot perform an reverse turning and escape from the corner.
- The straight-line walking behaviour: It sets full speed of 0.2 meter per second to both wheels.
- The random direction turning behaviour: It is implemented by selecting the randomness node. The node of randomness generates randomized speed values to each wheel.

3.3 The Hidden Markov Model for Semantic Knowledge

When the robot detects an object, it will make recognition of it. After that, the Room Confidence node in the Fuzzy Cognitive Map (shown in the figure 3.5) updates the estimation of current room and calculates the confidence on estimation, by utilizing the semantic knowledge. The estimated current room is compared with the possible target room, which is provided by human operator in advance. The decision on switching behaviours is made using the outputs of the Room Confidence node.

In this section, we will discuss the definition of semantic knowledge on the view of robot in this work. And the mechanism of updating the Room Confidence node by using semantic knowledge is presented after.

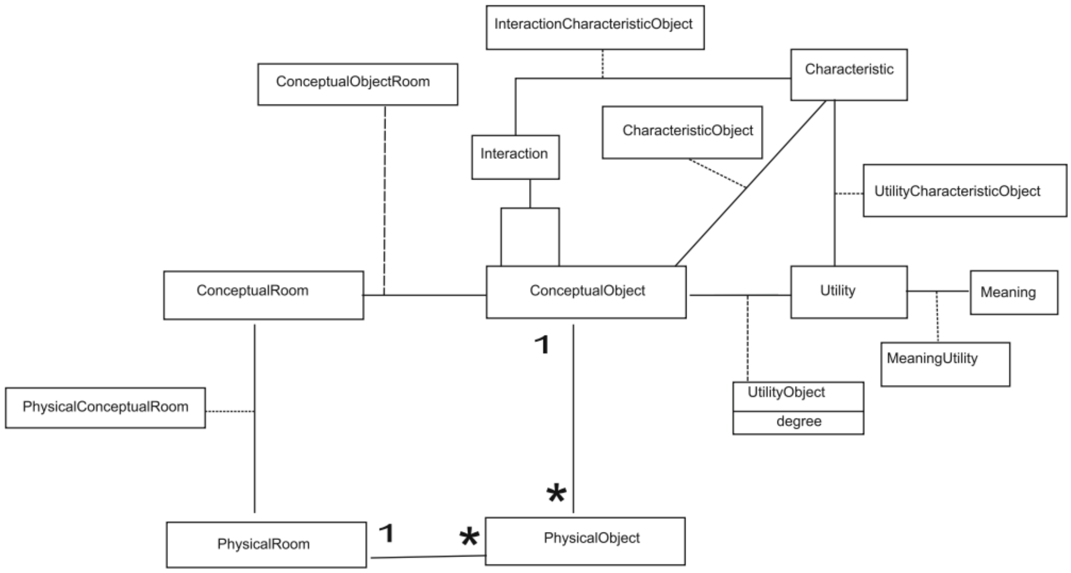


FIGURE 3.8: The Relational Model Between Objects and Rooms in The Work of J. Crespo (The * denotes a many-to-many relation), taken from [2]

.

3.3.1 The Semantic Knowledge

The semantic knowledge useful for a searching task includes the relationships between room types and objects. In the work of J. Crespo, a semantic relation model is proposed to help the service robot record knowledge about the environment and utilize the knowledge for path planning [2], as shown in the figure 3.8. The conceptual room and object are the ones which could be understood as concepts, for example kitchen and refrigerator. And the physical room and object are the ones exist as entities in the physical world. The work of J. Crespo provides inspiration for this thesis in defining the relationships between the object and the room.

The relationships between rooms and objects are also formalized as ontology in the thesis of Nicolas Maillot on object recognition [145]. The ontology is a term from the philosophical study, which is applied as the frame of shared hierarchical concepts. The relations between concepts could vary in their transitivity, refexivity and symmetry.

In this thesis, the relationships between object and room are defined by the probability of existence. In other words, the probabilities of the objects existing in each room are recorded as the relationships. The spatial distribution of rooms is also considered as useful knowledge for the searching task, also shown in the figure

3.8 by the work of J. Crespo [2]. We will discuss the definition of it in the next sub-section.

For details of the environment in the ARGoS simulator, we build four rooms in the arena, as shown in the figure 3.1. In each room, four objects denoted by cylinders with coloured-blob on top are randomly distributed. Each room connects to its two neighbouring rooms.

3.3.2 The Hidden Markov Model

The place recognition method which takes the result of object recognition can be found in the work of Antonio Torralba, applying the Hidden Markov Model [146]. To solve a similar problem in the searching task, the Hidden Markov Model is used to store the semantic knowledge and update the robot's estimated current room.

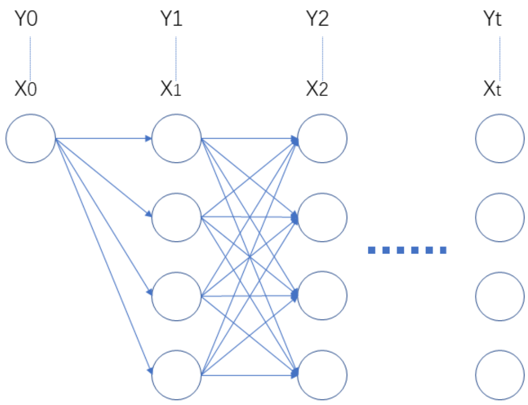


FIGURE 3.9: The Hidden Markov Model

The HMM models an Markov process with unobservable state X, which cannot be observed directly. The unobservable state X influences another process Y, which can be observed, as shown in the figure 3.9. With parameters of transition probability and emission probability, the state of X can be estimated. The transition probability gives the probability of the unobservable state changing from one state to another. And the emission probability gives the probability of observing Y when X is in each state.

Back to this thesis, the current room is an unobservable state by the robot. But the recognized object can be observed. Meanwhile, the probabilities of the object existing in each room is provided as the prior semantic knowledge. The spatial

	Room 1	Room 2	Room 3	Room 4
Room 1	0.7	0.15	0.01	0.15
Room 2	0.15	0.7	0.15	0.01
Room 3	0.01	0.15	0.7	0.15
Room 4	0.15	0.01	0.15	0.7

Table 3.1: An Example of the Transition Matrix A Used in This Thesis

	Object 1	Object 2	Object 3	Object 4
Room 1	0.7	0.1	0.1	0.1
Room 2	0.1	0.7	0.1	0.1
Room 3	0.1	0.1	0.7	0.1
Room 4	0.1	0.1	0.1	0.7

Table 3.2: An Example of the Emission Matrix B Used in This Thesis

distribution of rooms can be recorded as the state transition probabilities. The values in the transition probabilities is given by the probability of the robot travels from one room to another between the update of the estimated current room. Therefore, the value between rooms which are connected should be greater than the one which are not. And the probability of staying in the same room should be the largest.

In this thesis, we define the rooms shown in the figure 3.1 as Room 1, Room 2, Room 3 and Room 4, starting from the top-left ones and following the clockwise order. The objects are defined as Object 1, Object 2, etc.

An example of the semantic knowledge is shown as transition matrix A and emission matrix B in the table 3.1 and table 3.2. The values in both tables are chosen arbitrarily. In the table 3.1, the values of 0.7 show that the robot has a high probability of staying in the current room. The values of 0.15 are for rooms with a direct connection, and 0.01 for those without in the figure 3.1. In the table 3.2, the values of 0.7 mean that an object is more likely to be in one room than in the other rooms, e.g. the bed in the bedroom.

As is shown in the subsection 3.1.2, the robot only detects object in a limited range. Therefore, only at the time when an object is detected and recognized the robot updates its estimation of current room and the confidence on estimation by calculating $P\left(X_t = x\right)$ showing as follows[146]:

$$\begin{aligned} P\left(X_t = x \mid v^O_{1:t}\right) &\propto p\left(v^O_t \mid X_t = x\right) P\left(X_t = x \mid v^O_{1:t-1}\right) \\ &= p\left(v^O_t \mid X_t = x\right) \sum_{x'} A\left(x', x\right) P\left(X_{t-1} = x' \mid v^O_{1:t-1}\right) \end{aligned}$$

in which, $v_{1:t}^{O}$ are the the objects have been recognized by time t, $A(x', x)$ is the transition matrix and $p\left(v_t^O \mid X_t = x\right)$ is the emission probability. $P(X_0 = x)$ is provided in advance as the initial state.

Every time the Hidden Markov Model receives an recognized object, $P(X_t = x)$ in each room is updated. $\max P(X_t = x)$ is calculated as the confidence on estimation, and x is taken as the estimated current room. With the results, x is compared with the inferred target room.

3.4 Summary

This chapter narrows down the research on searching tasks by eliminating works on gas sensing, chemotaxis algorithms, and object recognition. The controller based on Fuzzy Cognitive Map is built, which is composed of robot behaviours designed after the human experience. The semantic knowledge is utilized by the Hidden Markov Model to estimate the current room. Comparing the estimated current room with the inferred target room will let the room make the decision on switching behaviours.

Chapter 4

Prior Semantic Knowledge Utilized by A Single Robot

In the chapter 3, we build a robot controller which incorporates semantic knowledge for the target searching problem. The semantic knowledge is provided in advance by the human operator for promoting searching efficiency. The semantic knowledge includes object-to-room relationships, room spatial distribution, and a possible room inferred by human experience. The effectiveness of this system needs to be tested from different dimensions, for example, the way of distributing objects and the variation of semantic knowledge.

In this chapter, we will run experiments on a single robot with different sets of parameters to testify that prior semantic knowledge can help promote searching efficiency on a single robot.

4.1 The Effectiveness of Semantic Knowledge on A Single Robot

To testify the effectiveness of semantic knowledge, we need to compare a single robot performing the searching task with and without the semantic knowledge provided in advance.

	Object 1	Object 2	Object 3	Object 4
Room 1	4	0	0	0
Room 2	0	4	0	0
Room 3	0	0	4	0
Room 4	0	0	0	4

Table 4.1: Amount of Objects Distributed in Each Room in The Experiments Comparing Robot With and Without Semantic Knowledge.

4.1.1 Experiments Set-up

For the robot with prior semantic knowledge, the transition matrix and the emission matrix are shown in the table 3.1 and the table 3.2. And the inferred target room is also supplied as Room 3.

For the robot without prior semantic knowledge, it runs with no need to recognize objects, to infer the current room and to compare with the inferred target room. So the connection from Object to Room Confidence in the figure 3.5 is deactivated. Without the ability of verifying the current room is the target room to search, the robot is designed to keep running straight-line walking and obstacle avoidance in order to traverse the whole arena.

The searching experiment arena is set up as the figure 3.1 but with one robot. In each room, the four objects are distributed by random. The amount of objects placed in each room is shown in the table 4.1.

The experiment ends when the robot finds the target and the experiment ticks is recorded as the time consumed for searching. In ARGoS simulator, the experiment runs 10 ticks in a second. So we can convert the searching time into seconds by dividing 10.

4.1.2 Results and Discussions

Because of the randomness introduced in the designing of behaviours, we plan to run large numbers of experiments to compare the overall performance. The number of experiment is set to 200 to be consistent with previous work (shown in the Appendix A). While the exact number of runs is arbitrary, it is important to choose a number large enough to ensure that the results are statistically significant - which is the case here.

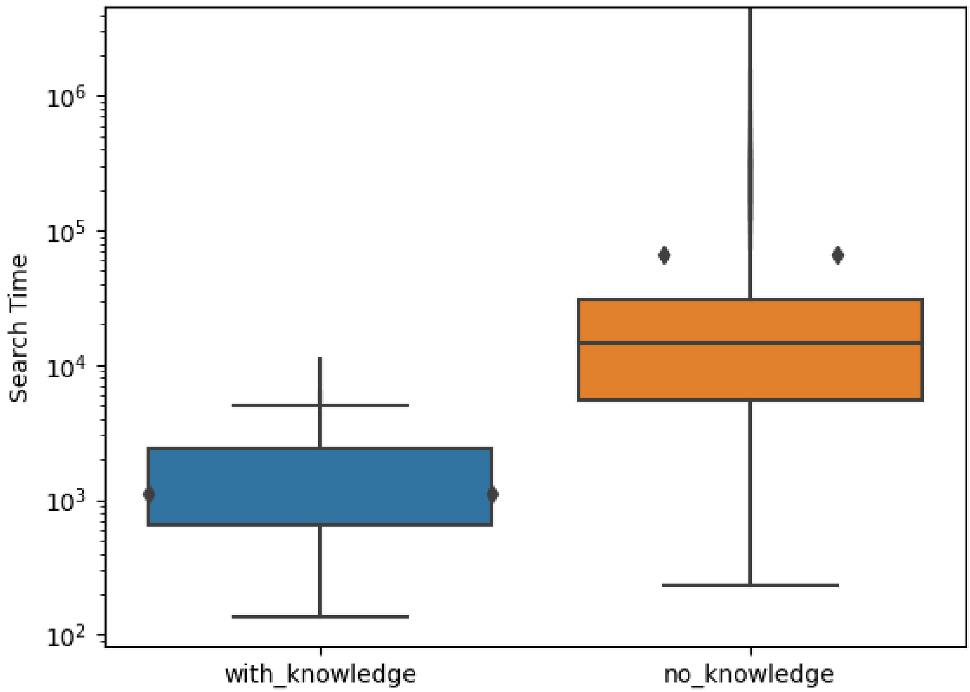


FIGURE 4.1: The Box-plot of Searching Time Used by Single Robot With and Without Semantic Knowledge. The Search Time in presented in seconds.

As shown in the figure 4.1, the robot with prior semantic knowledge use less time for searching than the robot without prior semantic knowledge[1]. The p value of Mann-Whitney U-Test scored less than 0.001, with the null hypothesis set as searching time consumed by robot with prior semantic knowledge is greater than the robot without prior semantic knowledge. This interprets as the robot with prior semantic knowledge uses significantly less searching time than the ones without. By deploying the Vargha-Delaney A-Test, we get the result score of 0.066, which falls into the scope of large effect size interval. Which means, there is a large effect between the robot with prior semantic knowledge and without prior semantic knowledge.

But a controversy may arise for the way of distributing objects. In these experiments, the objects and rooms are in the relation of strictly one-to-one arrangement, like Object 1 is only in Room 1. Suspicion could be given to the actual function of semantic knowledge based system. In other words, why not just let the robot

[1]The distribution of search times is skewed towards long search times due to the randomness. For this reason, a log-scale has been used here to facilitate comparison. Box-plots later in this thesis are drawn in a similar way for the same reason.

change behaviour to detailed searching every time it recognizes the Object. Therefore, we need to enrich the type of the objects and change the distribution of them.

4.2 Enriching Relations between Objects and Rooms

Let us go back to the practical scene, the object finding in a room gives us indication of the room type (number in this thesis) as we proposed. But the confidence of making such estimation varies. There will be objects which could show in different rooms. For example, a mark up could be in the kitchen or in the office. Meanwhile, there will be objects which owns only by one type of room. For example, a cooker hood should normally be found in the kitchen.

Therefore, we classify the objects into Specific Objects and General Objects. The Specific Object is like the cooking hood, has a strong connection with the kitchen. On the other hand, the General Object is like the mug, could be shared by the kitchen room or office. In a room, the robot might find Specific Objects, General Objects, or both of them.

In more detail, we classify the General Objects into,

- General Object shared by 4 rooms.
- General Object shared by 3 rooms.
- General Object shared by 2 rooms.

In the section 4.1, only Specific Objects are deployed. In the next step, we will place both Specific Object and General Object in the searching arena. The number of each object in each room will be presented. And the emission matrix B for describing the relevant semantic knowledge is provided relatively.

4.2.1 Combining Specific Object and General Object Shared by 4 Rooms

The emission matrix B in the Hidden Markov Model is shown in the table 4.2. Object 1 to Object 4 are the Specific Objects, so the emission probabilities are set

	Object 1	Object 2	Object 3	Object 4	Object 5
Room 1	0.7	0.1	0.1	0.1	0.25
Room 2	0.1	0.7	0.1	0.1	0.25
Room 3	0.1	0.1	0.7	0.1	0.25
Room 4	0.1	0.1	0.1	0.7	0.25

TABLE 4.2: The Emission Matrix B of 4 Specific Objects (1-4) and 1 General Object (5) Shared by 4 Rooms.

	Object 1	Object 2	Object 3	Object 4	Object 5
Room 1	3	0	0	0	1
Room 2	0	3	0	0	1
Room 3	0	0	3	0	1
Room 4	0	0	0	3	1

TABLE 4.3: The Amount of Each Object in Each Room, with 4 Specific Objects (1-4) and 1 General Object (5) Shared by 4 Rooms.

as the values before. Object 5 is the General Object shared by the four rooms, the emission probability is arbitrarily set to 0.25 in every room.

The amount of each object distributed in every room is shown in the table 4.3.

When the robot recognizes Object 5, the state updating in the Hidden Markov Model outputs its last estimated state because Object 5 gives the same emission probability for every room. In other words, the robot will not change its behaviour when it recognizes Object 5. And by eye observation, the robot runs as expected.

4.2.2 Combining Specific Object and General Object Shared by 3 Rooms

The emission matrix B in the Hidden Markov Model is shown in the table 4.4. Object 1 to Object 4 are the Specific Objects, so the emission probabilities are set as the values before. Object 5 to Object 8 are the General Objects shared by 3 rooms, the emission probabilities are arbitrarily set to 0.3 in the rooms sharing the objects.

The amount of each object distributed in every room is shown in the table 4.5.

The robot is initially distributed into Room 1. The target area is in Room 3 as always. The initial value of current room in the Hidden Markov Model is given as Room 1. In the process of searching, if the robot recognizes Object 3, the Hidden

	Object 1	Object 2	Object 3	Object 4	Object 5	Object 6	Object 7	Object 8
Room 1	0.7	0.1	0.1	0.1	0.3	0.3	0.3	0.1
Room 2	0.1	0.7	0.1	0.1	0.3	0.3	0.1	0.3
Room 3	0.1	0.1	0.7	0.1	0.3	0.1	0.3	0.3
Room 4	0.1	0.1	0.1	0.7	0.1	0.3	0.3	0.3

TABLE 4.4: The Emission Matrix B of 4 Specific Objects (1-4) and 4 General Objects (5-8) Shared by 3 Rooms.

	Object 1	Object 2	Object 3	Object 4	Object 5	Object 6	Object 7	Object 8
Room 1	3	0	0	0	1	0	0	0
Room 2	0	3	0	0	0	1	0	0
Room 3	0	0	3	0	0	0	1	0
Room 4	0	0	0	3	0	0	0	1

TABLE 4.5: The Amount of Each Object in Each Room, with 4 Specific Objects (1-4) and 4 General Object (5-8) Shared by 3 Rooms.

Markov Model will output Room 3 as estimated room. But if Object 5 or Object 7 or Object 8 is recognized, the outcome of current estimated room depends on the last estimated room. If the robot lastly estimated in Room 3 and recognizes Object 6, its estimated current room will be Room 1, or Room 2 or Room 4 by random. And its behaviour will change from detailed searching to go-to-next-room. This condition occurs in a low frequency and it is testified by eye observation. None of the General Object can change its behaviour from go-to-next-room to detailed searching.

4.2.3 Combining Specific Object and General Object Shared by 2 Rooms

The emission matrix B in the Hidden Markov Model is shown in the table 4.6. Object 1 to Object 4 are the specific objects, so the emission probabilities are set as the values before. Object 6 and Object 7 are shared by 2 rooms, the emission probabilities is arbitrarily set to 0.4 in the rooms sharing the objects.

The amount of each object distributed in every room is shown in the table 4.7.

In experiments with 2 General Objects shared by two rooms, the robot could change behaviours from detailed searching to go-to-next-room by recognizing Object 6 when its last estimation is in Room 3. These conditions happen more often

	Object 1	Object 2	Object 3	Object 4	Object 5	Object 6
Room 1	0.7	0.1	0.1	0.1	0.4	0.1
Room 2	0.1	0.7	0.1	0.1	0.1	0.4
Room 3	0.1	0.1	0.7	0.1	0.4	0.1
Room 4	0.1	0.1	0.1	0.7	0.1	0.4

Table 4.6: The Emission Matrix B of 4 Specific Objects (1-4) and 2 General Objects (5-6) Shared by 2 Rooms.

	Object 1	Object 2	Object 3	Object 4	Object 5	Object 6
Room 1	3	0	0	0	1	0
Room 2	0	3	0	0	0	1
Room 3	0	0	3	0	1	0
Room 4	0	0	0	3	0	1

Table 4.7: The Amount of Robots in Each Room, with 4 Specific Objects (1-4) and 2 General Object (5-6) Shared by 2 Rooms.

than the experiments using General Objects shared by 3 rooms and they are testified by eye observation. But none of the General Object can change its behaviour from go-to-next-room to detailed searching as well.

4.2.4 Randomized Semantic Knowledge

Combining the Specific Objects and General Objects in the experiments makes the environment for searching more complex. But the conditions listed above are still limited. We want to further explore performance of the system when different prior semantic knowledge is provided. In the worst case, the semantic knowledge about object to room relationships are described randomly. The randomized semantic knowledge might denote the chaotic knowledge from someone who holds the ambiguous connections between objects and rooms. It differs from no-knowledge in the aspect of switching the robot's behaviour, the latter one does nothing. The values in the emission matrix B are given random values between 0 and 1.

The objects are also randomly distributed in the arena. But the configuration of four objects in each room is still maintained.

	General 4	General 3	General 2	Random
Specific	0.032	0.138	0.126	<0.001

TABLE 4.8: U-test Scores for Comparison Between The Experiments With Specific Objects, General Objects, Randomized Semantic Knowledge

4.2.5 Results and Discussions

Experiments are carried out in simulation, with different emission matrix B and the distribution of objects as discussed above. Two hundred times of trials are running to collect the data of time used for searching.

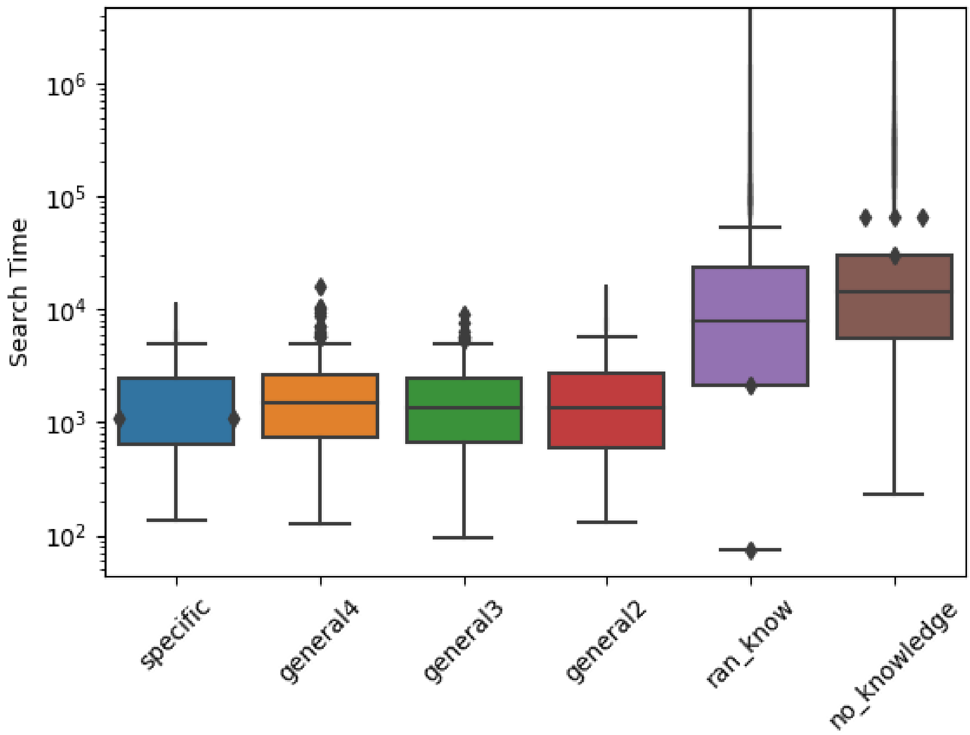


FIGURE 4.2: The Box-plot of Searching Time Used by A Single Robot When The Objects Varies by Its Classification. (The Search Time in presented in seconds. specific is for 4 Specific Objects in each room; general4 is for Specific Objects combined with General Objects shared by 4 rooms; general3 is for Specific Objects combined with General Objects shared by 3 rooms; general2 is for Specific Objects combined with General Objects shared by 2 rooms; ran_know is for randomized semantic knowledge; no_knowledge is for the robot without prior semantic knowledge.)

As shown in the figure 4.2, the searching time increases when the General Objects shared by different rooms are placed in every room together with the Specific Objects.

	General 4	General 3	General 2	Random
Specific	0.446	0.468	0.466	0.165

Table 4.9: A-test Score for Comparison Between The Experiments With Specific Objects, General Objects and Randomized Semantic Knowledge

The U-test is applied to the searching time used by experiments only using Specific Objects, and using combination of various Specific Objects and General Objects. General objects vary between sharing by 4 rooms, 3 rooms and 2 rooms. The p value is calculated for rejecting the null hypothesises that the searching time in experiments with only Specific Objects are longer than the searching time in experiments with each variation of General Objects, shown in the table 4.8. The p values of experiments using General Objects shared by 3 and 2 rooms are greater than 5%, the null hypothesis cannot be rejected. Therefore, the searching time of experiments with only Specific Objects are not significantly shorter than the experiments with General Objects shared by 3 and 2 rooms. But the p value of experiments with General Objects shared by 4 rooms is less than 5%. So the searching time of experiments with only Specific Objects are significantly shorter than the experiments with General Objects shared by 4 rooms.

The A-Test is also calculated between the searching time in experiments only using Specific Objects, and using combination of various Specific Objects and General Objects. Scores are shown in the table 4.9. The scores of experiments with General Objects shared by 4, 3 and 2 rooms are greater than 0.44 and less than 0.56, which is the scope of negligible effect size. But the score of experiments with General Objects shared by 4 rooms is very close to the boundary of 0.44 as small effect size. In general, adding in General Objects shared by different rooms shows a negligible effect on the searching time. In particular, the General Objects shared by 4 rooms is closely to make small effect on the searching time.

This phenomenon can be explained by the behaviour changing conditions discussed above. The General Objects shared by 4 rooms is useless for updating robot's current room estimation.

For the experiments using General Objects shared by 3 rooms, the General Object could trigger the behaviour switching from detailed searching to go-to-next-room in a low frequency. And for the experiments using General Objects shared by 2 rooms, the General Object also triggers behaviour switching from detailed searching to go-to-next-room, but in a higher frequency. By analyzing the p value and effect size score, the difference in this frequency is trivial. In conclusion, the General Object shared by different rooms in the entire searching arena is useful for promoting the searching efficiency. Whereas, the General Objects shared by all rooms are close to slow down the searching. In another point of view, these experiments testify that the system could utilize semantic knowledge about those General Object.

For the experiments using randomized semantic knowledge and distributing objects randomly. The p value generated by the U-test is less than 5% for the null hypothesis that the searching time in experiments with only Specific Objects are longer than the searching time in experiments with randomized prior semantic knowledge, listed in the table 4.8. The p value rejects the null hypothesis, which means the experiments with correct prior semantic knowledge could finish the searching task more efficiently than the experiments with randomized prior semantic knowledge.

The A-test comparing experiments with only Specific Objects and randomized prior semantic knowledge produces a score of 0.165 in the table 4.9, which is in the area of large effect size, below 0.29. The variation from correct prior semantic knowledge to randomized one has a large effect on the searching time.

The comparison between experiments with randomized prior semantic knowledge and without prior semantic knowledge is also made. The U-test gives p value less than 0.001, for the null hypothesis that experiments with randomized prior semantic knowledge use longer time than experiments without semantic knowledge. The p value rejects the null hypothesis, so that time used by the experiments with randomized prior semantic knowledge is less than the experiments without prior semantic knowledge. The relevant A-test scored 0.596, in the scope of small effect size. That is to say, the variation of experiments from randomized prior semantic knowledge to without knowledge is a small effect change. In conclusion, the randomized prior semantic knowledge makes no difference with no knowledge on the searching time of a single robot.

4.3 Reliance on The Specific Object

In the section 4.2, the experiments which combine General Objects and Specific Object are all using the set-ups of 3 Specific Objects and 1 General Object in each

Set-ups		Object 1	Object 2	Object 3	Object 4	Object 5	Object 6
Set 1	Room 1	3	0	0	0	1	0
	Room 2	0	3	0	0	0	1
	Room 3	0	0	3	0	1	0
	Room 4	0	0	0	3	0	1
Set 2	Room 1	2	0	0	0	2	0
	Room 2	0	2	0	0	0	2
	Room 3	0	0	2	0	2	0
	Room 4	0	0	0	2	0	2
Set 3	Room 1	1	0	0	0	3	0
	Room 2	0	1	0	0	0	3
	Room 3	0	0	1	0	3	0
	Room 4	0	0	0	1	0	3
Set 4	Room 1	0	0	0	0	4	0
	Room 2	0	0	0	0	0	4
	Room 3	0	0	0	0	4	0
	Room 4	0	0	0	0	0	4

TABLE 4.10: The Amount of Robots in Each Room For The Experiments Varying in Numbers of Specific Objects.

room. We want to further explore the system's performance when this set-up is varied. By this, the reliance on Specific Objects can be revealed.

The General Objects shared by 2 rooms are picked for replacing the Specific Objects gradually in this section. Therefore, the emission matrix B adopts the settings in the table 4.6.

For the number of each object distributed in each room, it is shown in the table 4.10. From Set 1 to Set 4, the amount of Specific Objects decreases from 3 to 0. Experiments on each set run 200 times to collect the searching time.

The box-plot in the figure 4.3 shows the searching time used by a single robot when number of Specific Objects in each room varies.

From 3 Specific Objects in each room to 2 Specific Objects in each room, the U-test gives value p as 0.047 to reject the null hypothesis than experiments with 3 Specific Objects in each room finish the searching task quicker than the ones with 2 Specific Objects. In other words, the searching time used by a single robot in the experiments with 3 Specific Objects in each room is statistically less than the one in experiments with 2 Specific Objects in each room. The relevant A-test produces score of 0.451. As it is in the negligible scope, the effect of changing from 3 Specific Objects in each room to 2 is trivial. Combining the results from U-test

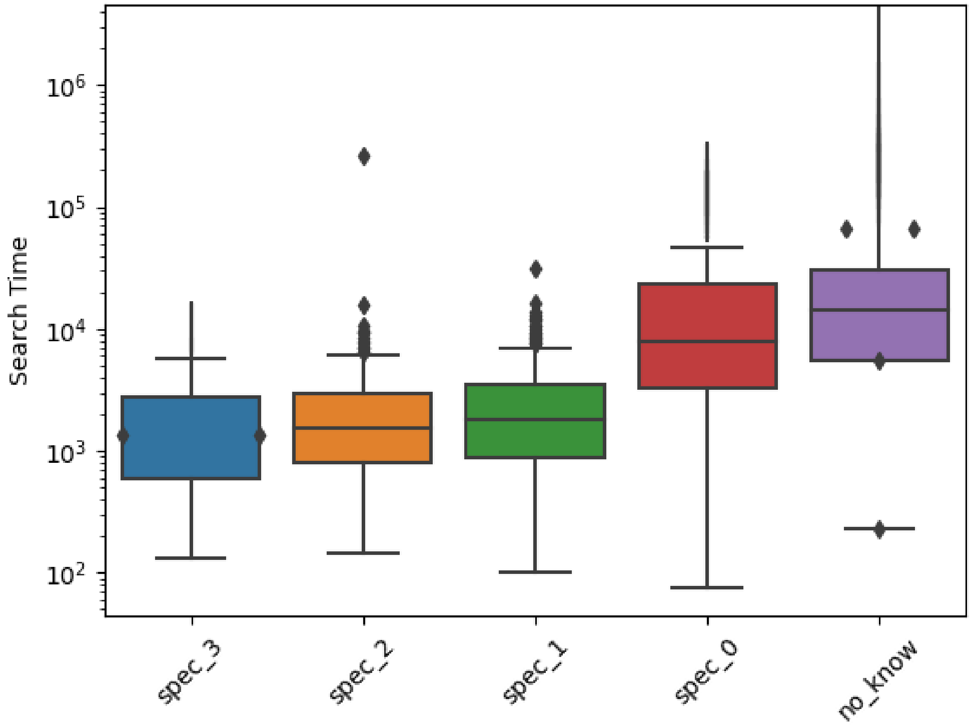


FIGURE 4.3: The Box-plot of Searching Time Used by A Single Robot When The Number of Specific Objects Varies. (The Search Time in presented in seconds. spec_3 is for 3 Specific Objects in each room; spec_2 is for 2 Specific Objects in each room; spec_1 is for 1 Specific Objects in each room; spec_0 for no Specific Object in each room; no_know for the experiments without prior semantic knowledge.)

and A-test, we could say the variation in number of Special Objects in each room from 3 to 2 makes no significant difference to the searching time.

Comparing experiments with 2 Specific Objects in each room to 1 in each room, the U-test generates value p of 0.053. The null hypothesis in the same way, the searching time of experiments with 2 Specific Objects in each room is longer than the experiments with 1 in each room. The null hypothesis can not be rejected. The relevant A-test scored 0.546 as negligible effect size. Therefore, changing from 2 Specific Objects to 1 in each room still makes no significant difference.

The U-test between results of 1 Specific Object and no Specific Object generates value p less than 0.001. This rejects the null hypothesis of with 1 Specific Objects in each room the robot needs a longer time for searching. And the relevant A-test scored 0.805 as a large effect size. That is to say, to totally replace the Specific

Objects with the General Objects makes the robot search significant less efficiently, compared with saving 1 Specific Object in each room.

The searching time in experiments with no Specific Object are also compared with the experiments without prior semantic knowledge. The U-test gives value p less than 0.001, which rejects the null hypothesis experiments with no Specific Object has a longer searching time. The relevant A-test scored 0.406. The score indicates a small effect is made. Therefore, the robot in experiments with only General Objects perform the searching task only slightly, but significantly more efficiently than the one without prior semantic knowledge.

In conclusion, the system relies on the semantic knowledge on the Specific Objects to perform the searching task efficiently. But the knowledge of the General Objects still makes small contribution to promote efficiency of the searching, compared with the condition without prior semantic knowledge.

4.4 Summary

This chapter testifies to the effectiveness of prior semantic knowledge for promoting the searching efficiency on a single robot. A comparison is made between the robot with prior semantic knowledge and the robot without. The result of experiments shows a significant promotion in searching efficiency by statistic analysis. But the question is raised about the strictly one-to-one relation between objects and rooms.

To deal with the question, we propose a classification of objects by the relationships between objects and rooms. By varying different types of objects in the proposed classification, their relevant effectiveness is presented. The experiment results show that the robot's searching time in the arenas containing Specific Objects and different General Objects make no difference from the one in the arena only containing Specific Objects. However, if the semantic knowledge is provided at random, the searching time shows a small difference from the experiments without prior semantic knowledge.

The reliance on the Specific Objects has been explored. The existence of Specific Objects in each room is a great help in promoting searching efficiency. The experiment results show that the General Objects shared by different rooms make a small contribution to promoting the searching efficiency.

Chapter 5

Robots Learning The Semantic Knowledge

The effectiveness of our proposed system on a single robot is testified in the chapter 4. Semantic knowledge helps promote searching efficiency when it is provided in advance. However, we want to generalized the utility of this system by extending it to scenarios where no prior semantic knowledge is provided.

In the absence of sensitive sensors and global X-Y-Z coordinates, the robot cannot carry out those complex navigation strategies. And it relies on randomness to run into the target by accident. To increase the number of robots deployed for searching, the target could be found faster. Therefore, the system will be extended to work on a swarm of robots. The first robot finding the target will try to record useful information about the environment into semantic knowledge. Then, it passes learned semantic knowledge to the other robot in the communication range. The robot receives the knowledge and should make use of it to help perform the searching task.

In this chapter, we will discuss the design of robot learning semantic knowledge, communicating the knowledge, and utilizing it for the searching tasks. As the system is proposed to work on a swarm of robots, experiments will be run on the swarm sized 10, 20, and 40 [1].

[1] The search area is 10 metres by 10 metres. And the e-puck robot has a diameter of less than 0.1 metre. A large number of robots are therefore required to cover the large space. Swarm sizes are arbitrarily set at 10, 20 and 40, in order to investigate system performance as the swarm size increases.

5.1 Semantic Knowledge Learning, Communication and Utilization

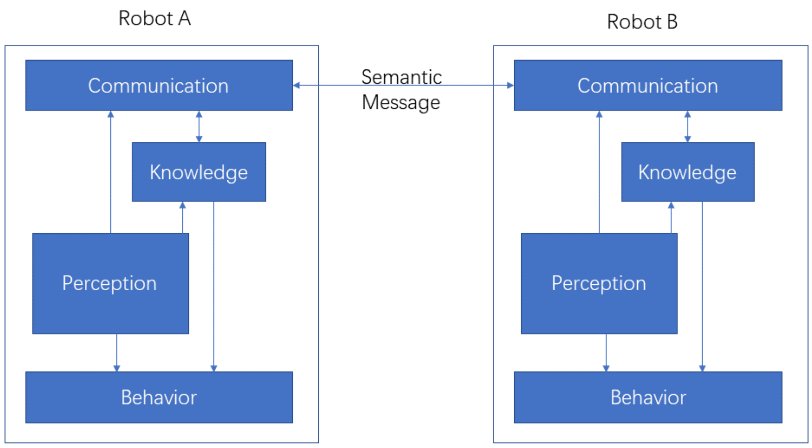


FIGURE 5.1: The Knowledge Learning and Communication in A Robot

For the robot in the swarm without prior semantic knowledge, the information recording and sharing mechanisms are required so that the information on the environment can be propagated crossing the swarm, as shown in the figure 5.1. In the research during this PhD study, we try to employ semantic map labelling for improving swarm robotics searching tasks. This work has resulted in an unpublished paper, shown in the Appendix A. The robot makes semantic abstraction of the information on the environment under the circumstances that global X-Y-Z coordinates cannot be acquired. The method is proven to work efficiently even in the presence of sensor noise. As the semantic communication between robots proven to be effective, we devise the information recording and sharing mechanisms in this thesis into semantic knowledge learning, communication, and utilization.

5.1.1 Semantic Knowledge Learning

5.1.1.1 Available Source of Information

In the unknown environment, without the prior semantic knowledge, the robot which finds the target will try to learn the environment in the arena and store it as its semantic knowledge. Following the principles of swarm robotics, the semantic knowledge should remain local to the robot. Which means, the global coordinates in not considered just like in the Chapter 4. Adopting the searching arena in the figure 3.1, the available information includes,

- Objects
- Target Area
- Gateways

As before, the semantic knowledge being learnt should be the same as relationships between objects and rooms. And this information has the feature of local.

Without the prior semantic knowledge, the robot will not have concepts of room number, for example the Room 1 in the Chapter 4. Instead, it can only label the room as Target Room after finding the target, or vaguely label the room as Non-Target Room if it hasn't found the target in that room. By vaguely, we mean there are conditions, in which the robot is in the Target Room but still hasn't found the target. A tricky problem is how to get the robot to recognise the state of leaving the Target Room.

As solution, the robot has to rely on recognizing gateways for timely stop labelling current room as Target Room when it exits the Target Room. This can still be considered as working in practical searching tasks, as the gateway can be considered as a special object. And the packages for object recognition could easily recognize doors to make this work.

For the objects, relationships will be built to connect to the Target Room and the Non-Target Rooms. The objects are classified as Specific Objects and General Objects in the Chapter 4. Both type of objects help increase the searching efficiency, but the Specific Objects work better. Therefore, we will focus our work by placing only the Specific Objects in the searching arena, i.e., without the General Objects.

5.1.1.2 Learning Process

The semantic knowledge learned by the robot is stored as the emission matrix B, which describes the probabilities of finding each object in each room. The

	Object 1	Object 2	Object 3	Object 4
Target Room	0.1	0.1	0.1	0.1
Non-Target Room	0.1	0.1	0.1	0.1

Table 5.1: The Initial Emission Matrix B for Robots Learning Semantic Knowledge

	Target Room	Non-Target Room
Target Room	0.7	0.3
Non-Target Room	0.3	0.7

Table 5.2: The Transition Matrix A for Robots Learning Semantic Knowledge.

values in it are initialized as shown in the table 5.1. Relatively, the transition matrix A is initialized as shown in the table 5.2. For the reason that, the available source of information is not abundant enough for the robot to update the spatial distribution of rooms, the transition matrix A will not change in the searching process.

After the robot finding the target and labelling the Target Room, it will carry on the detailed searching behaviour for searching objects. The object being found in the Target Room will be connected to the Target Room by setting the emission probability of that object in the Target Room as 0.7. The value of 0.7 is set arbitrarily, it will work so long as it is greater than this object's emission probability in the Non-Target Room. And the robot gives tacit consent to the object is specific for the Target Room. As can be predicted, if this object is a General Object shared by other rooms, the searching efficiency will reduce. But the effectiveness of General Objects has been shown in the Chapter 4, so this should not be a problem.

Because of the vagueness in labelling the Non-Target Room, the emission probability of objects in the Non-Target Room cannot be updated. So those values stay as 0.1.

5.1.2 Semantic Knowledge Communication

The robot which has learned the semantic knowledge will transmit the knowledge to other robots in its communication range. The content of communication should be designed with accordance to the format of semantic knowledge storage. In the system, the semantic knowledge learned by the robot is stored as the emission matrix B, in which only the emission probability of objects in the Target Room is changing. Therefore, the robot sends index of objects in the Target Room to its neighbouring robots through the range-and-bear actuator on E-puck robot.

The robot which receives semantic knowledge by the range-and-bear sensor will update its emission matrix B. Although this robot may still have not find the target, this robot still sends the semantic knowledge to its neighbouring robots.

5.1.3 Semantic Knowledge Utilization

The semantic knowledge utilization is designed for the robot which receives knowledge from other robots and still searching for the target. Like the robot in the Chapter 4 which receives semantic knowledge from the human operator, the Hidden Markov Model works to updates the Room Confidence node to switch behaviours, when the robot recognizes an object.

The Hidden Markov Model is modified so that the unobservable state will fall into Target Room or Non-Target Room. The observable state is still the object, which is influenced by the state of current room. The transition matrix and emission matrix have been described above.

When the Hidden Markov Model outputs Target Room, the robot will switch to detailed searching. And if Non-Target Room is estimated, the robot switch to go-to-next-room behaviour.

5.2 System Performance

The experiments are running 200 times in simulation with different swarm sizes ranging in 10, 20 and 40. Two types of data are collected which are the time each robot start having the semantic knowledge as learning time and the time each robot finds the target as searching time.

To testify the hypothesis that in the absence of providing prior semantic knowledge deploying a swarm of robots can promote the searching efficiency compared with a single robot, the time used for searching in experiments with swarm size 10 are compared with experiments, which deploy a single robot. In addition, we gives another proposal which is the searching efficiency can be further promoted by increasing the swarm size. Apart from searching time, we also care about the semantic knowledge learning time. Therefore, comparisons both on semantic knowledge learning time and on searching time are made between experiments with different swarm sizes.

5.2.1 Comparison Between Swarm of Robots and A Single Robot

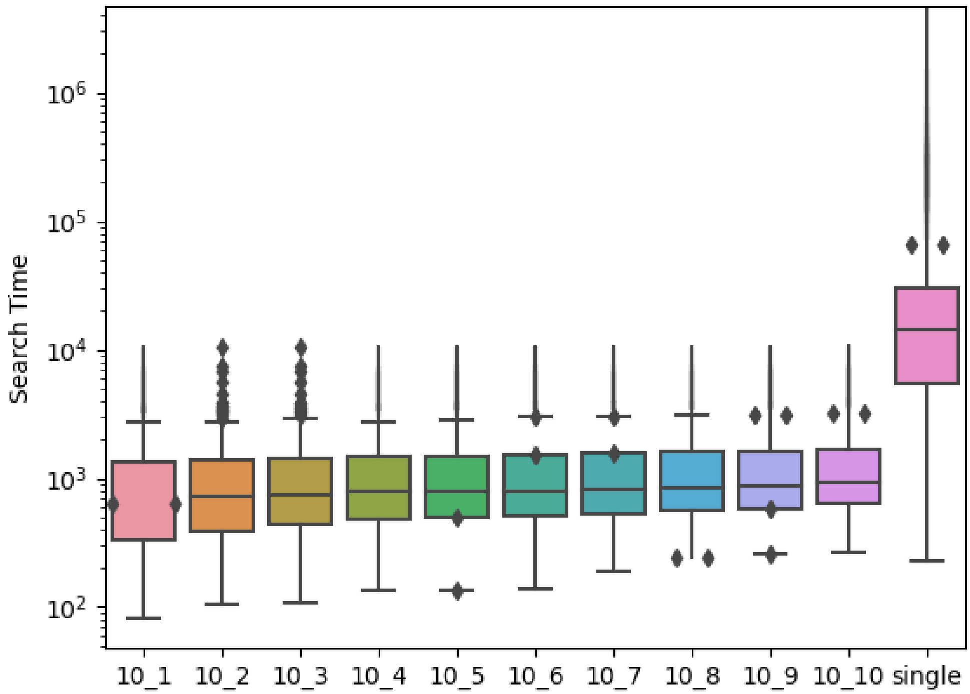


FIGURE 5.2: The Box-plot of Searching Time Used by A Swarm of 10 Robots and A Single Robot. (The Search Time is presented in seconds. 10_1 is for the first robot finding the target in the swarm; the after ones with 10_x are denoted likewise; single is for the Single Robot.)

The box-plot shown in the figure 5.2 presents the searching time consumed by each robot in the Swarm of 10 Robots and by a single robot. Statistical analysis is also made between the last robot finding the target in the swarm and the single robot. The U-test produces the p value less than 0.001, which rejects the null hypothesis of search time used by the last robot finding the target in the swarm of 10 robots is longer than the time used by a single robot. Which means, the last robot in the swarm of 10 robots finds the target significantly quicker than the single robot. The A-test scored 0.234, which belongs to the large effect size. Therefore, by deploying swarm of robots the searching efficiency is promoted compared with a single robot when the semantic knowledge is not provided in advance.

5.2.2 Comparisons Between Different Sizes of Swarm

As described before, we are interested in the semantic knowledge learning time and searching time. The robot could either learn the semantic knowledge on its own or receive the semantic knowledge from other robots. Both conditions works the same for defining the semantic knowledge learning time.

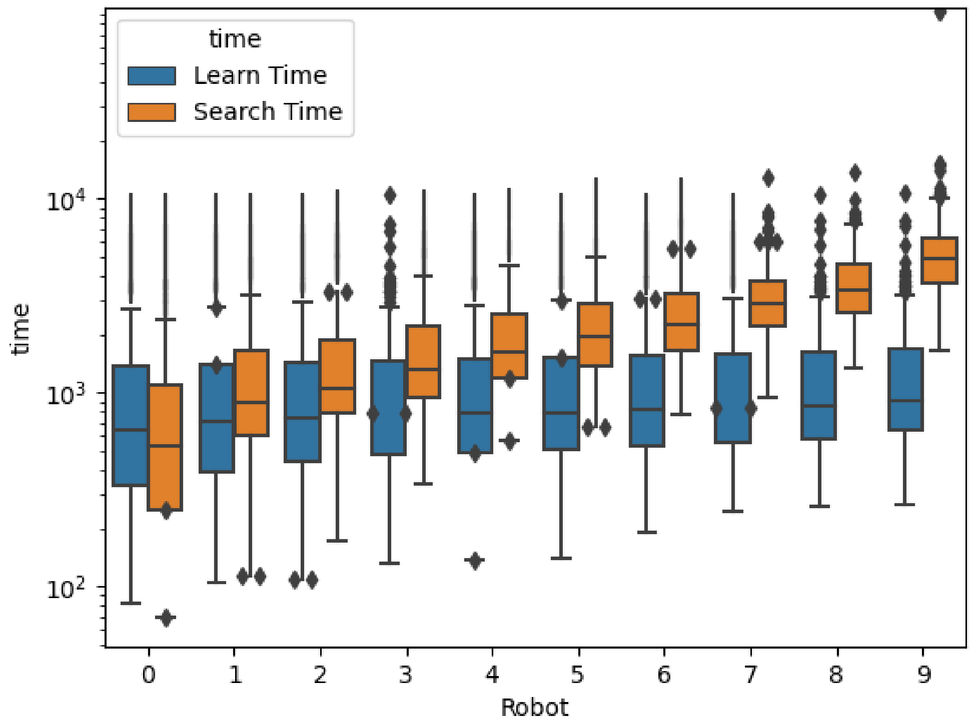


FIGURE 5.3: The Box-plot of Semantic Knowledge Learning Time And Searching Time in The Swarm Sized 10. (time is presented in seconds. 0 to 9 on the X-Axis denote the robots ordered by the time learning the semantic knowledge or finding the target respectively)

As shown in the figure 5.3, the figure 5.4 and the figure 5.5, the semantic knowledge learning time is plotted in combination with the searching time for the swarm sized 10, 20 and 40.

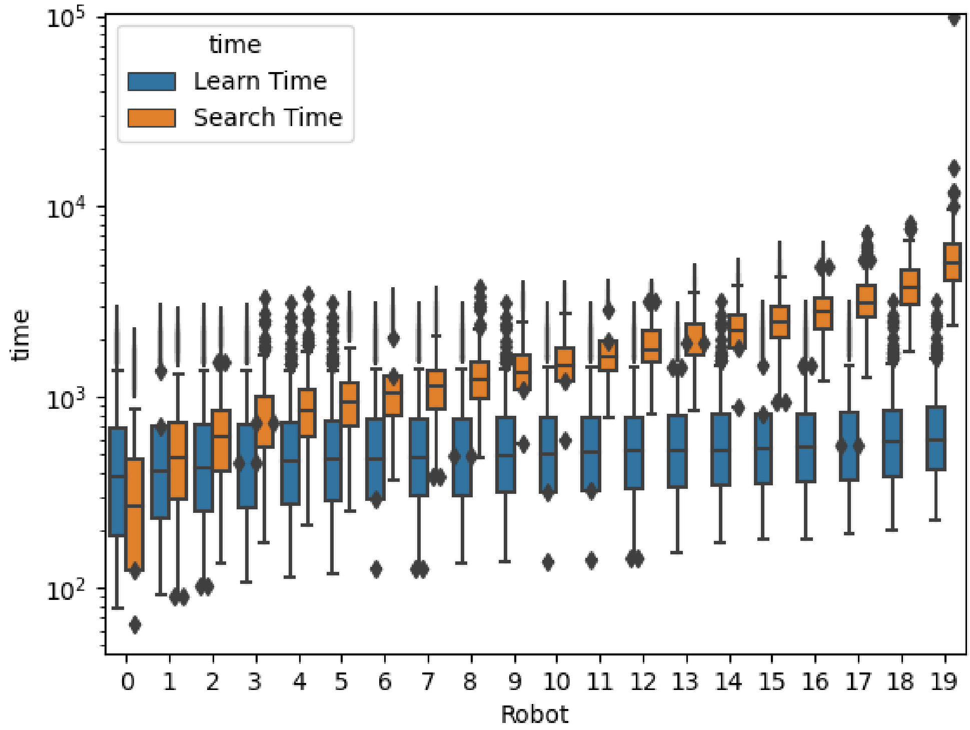


FIGURE 5.4: The Box-plot of Semantic Knowledge Learning Time And Searching Time in The Swarm Sized 20. (time is presented in seconds. 0 to 19 on the X-Axis denote the robots ordered by the time learning the semantic knowledge or finding the target respectively)

To compare the performance of the system on different swarm sizes, we pick the semantic knowledge learning time and searching time of 3 special robots in the swarm. The reason for selecting them will be discussed with the results.

- The first robot learns the semantic knowledge and the first robot finds the target.
- The robot ranks in the half of the swarm size (which are the 5_{th}, 10_{th} and 20_{th} robot for swarm size 10, 20 and 40) has the semantic knowledge and the one finds the target. We will call it as the Mid robot for convenience in this section.
- The last robot has the semantic knowledge and the last robot finds the target.

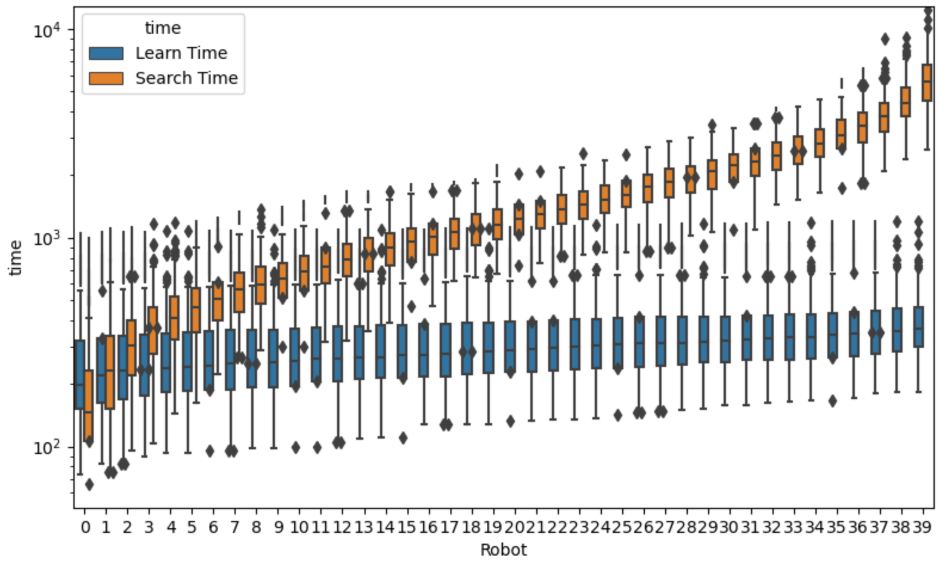


FIGURE 5.5: The Box-plot of Semantic Knowledge Learning Time And Searching Time in The Swarm Sized 40. (time is presented in seconds. 0 to 39 on the X-Axis denote the robots ordered by the time learning the semantic knowledge or finding the target respectively)

In the figure 5.6 shows the box-plot of the semantic knowledge learning time of the 3 selected robots. And in the figure 5.7 shows the box-plot of the searching time of the 3 selected robots.

5.2.2.1 The First Robot

The first robot finding the target is driven by randomness, as we discussed earlier. By increasing the swarm size, one of the robots in the swarm will be more probable in find the target. Therefore, the first robot finding the target in a larger swarm size should use less time. And the robot first learning the semantic knowledge is closely related with the first robot finding the target, but not necessarily be that one. Because the first robot finding the target may escapes from the Target Room before recognizing an object.

The U-test is calculated between the swarm sized 20 and 10 for the searching time of the first robot finding the target. The p value is less than 0.001, which rejects the null hypothesis that the first robot finding the target in swarm sized 20 takes a longer time than the first robot in swarm sized 10. The relevant A-test scored 0.317. It is in the range of medium effect size. Combining the results of U-test

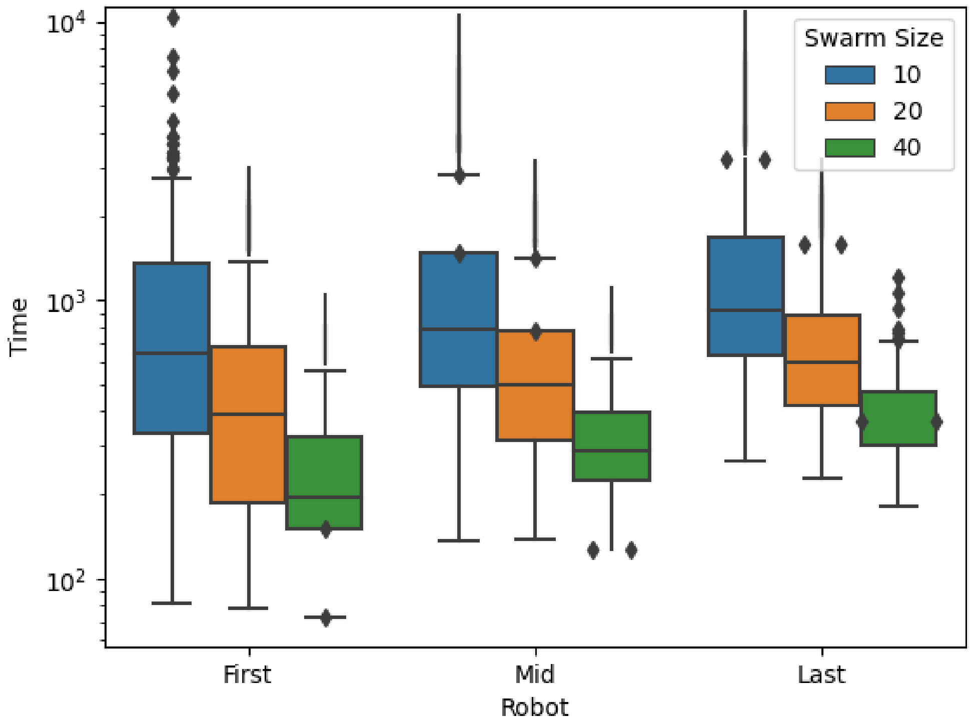


FIGURE 5.6: The Box-plot of Semantic Knowledge Learning Time. Ordered by the First, Mid, and Last Robot in Different Sizes of Swarm. (The Time is presented in seconds.)

and A-test, conclusion can be drawn that the first robot finding the target in the swarm sized 20 takes a shorter time than the first robot in the swarm sized 10. In the same way, the U-test comparing the first robot in the swarm sized 40 and 20 produces p value less than 0.001 and the A-test scored 0.320, in the scope of medium effect size. So, the first robot finding the target in the swarm sized 40 takes a shorter time than the first robot in the swarm sized 20 as well. To sum it up, by increasing the swarm size, one robot in the swarm will find the target faster as the first robot finding it.

Then, about the semantic knowledge learning time, the U-test outputs the p value less than 0.001. It rejects the null hypothesis that the first robot in the swarm sized 20 learns the semantic knowledge later than the first robot in the swarm sized 10. The relevant A-test scored 0.332, in the range of medium effect size. The p value generated by the U-test between the swarm sized 40 and 20 is less than 0.001 with the null hypothesis likewise. And the relevant A-test scored 0.285, which also indicates a medium effect. In conclusion, by increasing the swarm size,

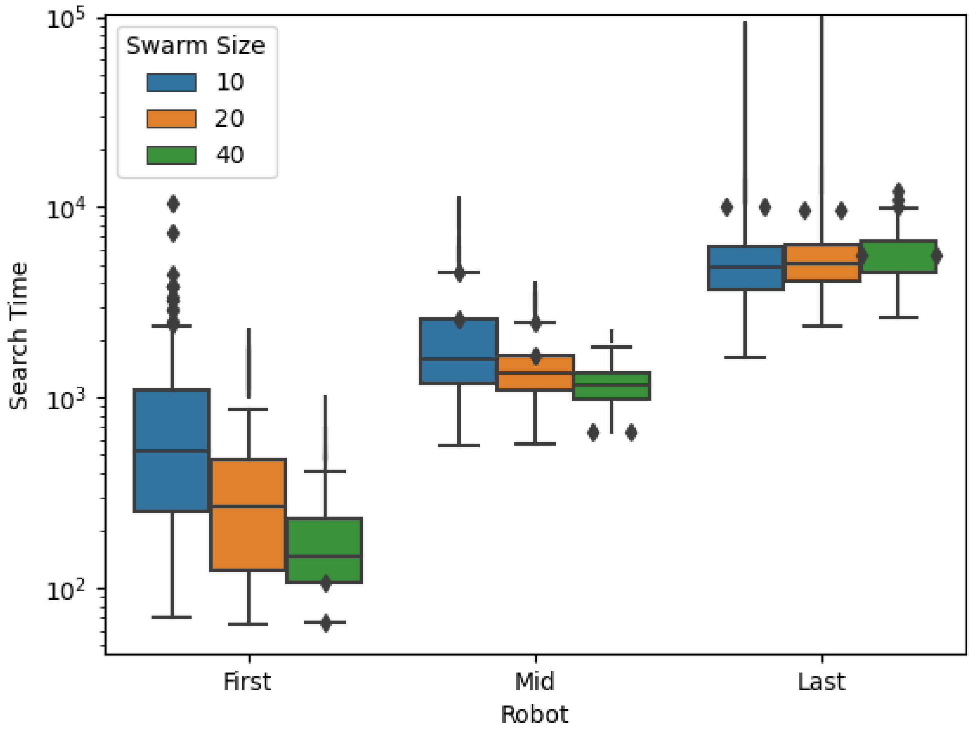


FIGURE 5.7: The Box-plot of Searching Time. Ordered by The First, Mid, and Last Robot in Different Sizes of Swarm. (The Search Time is presented in seconds.)

one of the robots in the swarm will learn the semantic knowledge faster as the first robot learning the knowledge.

5.2.2.2 The Mid Robot

By picking the robot ranks in the half of the swarm size finding the target, we want to take an insight into the dynamic performance of the system on different swarm sizes. Besides, if half number of robots in a swarm have found the target, the searching task could be judged as finished in some searching problems.

The U-test comparing the Mid robot's searching time in the swarm sized 20 and 10 gives the p value less than 0.001. It rejects the null hypothesis that the Mid robot finding the target in swarm sized 20 takes a longer time than the Mid robot in swarm sized 10. And the relevant A-test scored 0.374, in the range of small effect size. The U-test between the Mid robot's searching time in the swarm sized 40 and 20 outputs the p value less than 0.001, rejecting the null hypothesis likewise.

The relevant A-Test scored 0.345, also in the range of small effect size. In brief, half number of the robots in a swarm will use less time in finding the target if the swarm size is increased.

About the semantic knowledge learning time, the U-test outputs value p less than 0.001 for comparing the Mid robot in swarm sized 20 and 10. The null hypothesis that the Mid robot in the swarm sized 20 learns the semantic knowledge slower than the one in the swarm sized 10, is rejected. The A-test scored 0.306, in the medium effect size. The U-test between the Mid robot's semantic knowledge learning time in the swarm sized 40 and 20 produces value p less than 0.001 and rejects the null hypothesis likewise. The relevant A-test scored 0.249, in the large effect size. That is to say, half number of the robots in a swarms will learn the semantic knowledge quicker if the swarm size is increased.

5.2.2.3 The Last Robot

As for the last robot learning the semantic knowledge, comparing its knowledge learning time could track the speed of semantic knowledge propagation to cover the entire swarm. The searching time of the last robot finding the target is the key indicator for observing the performance of the whole swarm.

As is shown in the figure 5.7, when the swarm size increases, the searching time of the last robot finding the target shows an different pattern. Therefore, the U-test between the last robot in swarm sized 20 and 10 is provided an reversing null hypothesis, which is the searching time of the last robot in the swarm sized 10 is greater than the one in swarm sized 20. The generated p value is 0.052. So the null hypothesis can not be rejected. And the relevant A-test scored 0.546, as negligible effect. Therefore, the increment of the swarm size from 10 to 20 does not significantly raise the searching time of the last robot in swarm to find the target. The U-test is also calculated for the last robot between swarm sized 40 and 20, the null hypothesis is set as the searching time of the last robot in the swarm sized 20 is greater than the one in swarm sized 40. The p value in outputs is 0.005, which rejects the null hypothesis. But the score of relevant A-test is 0.573, still in negligible effect range. In combination, the increment of the swarm size from 20 to 40 still does not significantly rise the searching time for the last robot to find the target. In this case, the comparisons between swarm sized 10 and 40 should be made. The U-test, with the null hypothesis which is the searching time of the last robot in the swarm sized 10 is greater than the one in swarm sized 40 produces value p less than 0.001. It rejects the null hypothesis. The relevant A-test scored 0.611, in the range of small effect size. In brief, the increment of swarm size from 10 to 40 makes the entire swarm take a longer time for searching.

For the semantic knowledge learning time, the U-test comparing the last robot in the swarm sized 20 and the one in the swarm sized 10 returns value p less than 0.001. The null hypothesis is that the knowledge learning time of the last robot in the swarm sized 20 is longer than the one in the swarm sized 10. The p value rejects the null hypothesis. The relevant A-test scored 0.714 to indicate a medium size effect. The U-test between the last robot in the swarm sized 40 and the one in the swarm sized 20 gives the p *value* less than 0.001, with the null hypothesis likewise. The null hypothesis is rejected. The relevant A-test scored 0.777, in the range of large effect size. Therefore, by increasing the swarm size, the knowledge learning time of the entire swarm declines.

5.3 Correlations between The Searching Time and The Semantic Knowledge Learning Time

As being proposed, the searching efficiency of a robot is promoted after receiving the semantic knowledge from the other ones. Also, the semantic knowledge learning time and searching time of each robot are connected. Therefore, the next step is to explore the correlations between the semantic knowledge learning time and the searching time of each robot in different size of swarm. We plot the 5_{th} robot's searching time and knowledge learning time in scattered points by each set as an insight. The scatter plots are shown in the figure 5.8, figure 5.9 and figure 5.10. By observation, the correlations between the searching time and knowledge learning time of the 5th robot can be described as linear. To further explore the correlation, statistical analysis is required for every robot in the swarm.

It should be noted that, the searching times are collected by the order of the robot finding the target and the learning times are collected by the order of semantic knowledge learning. So, the robot with the same rank in the searching time and in the learning time, is not necessarily the same one.

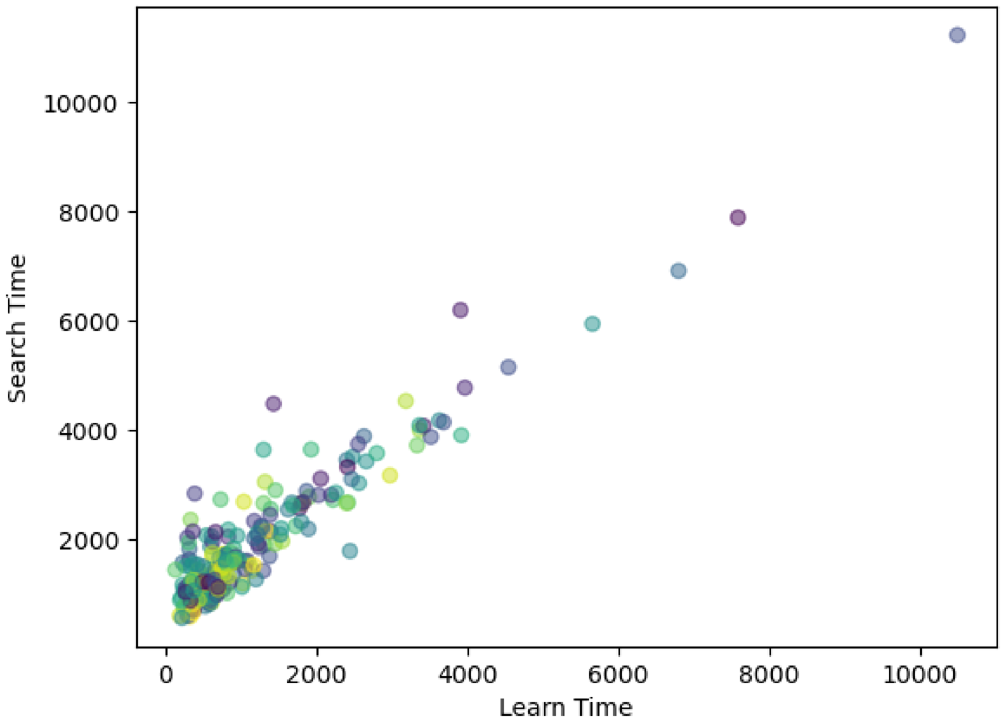


FIGURE 5.8: The Scatter Plot of The 5th Robot's Semantic Knowledge Learning Time And Searching Time in Swarm Sized 10.(The time is presented in seconds.)

To analyse statistically, the Pearson correlation coefficients and the values of p are calculated on the searching time and the semantic knowledge learning time in different size of swarm. The value p is for testing the null hypothesis that the distributions underlying the searching time and the learning time are uncorrelated and normally distributed [132]. The Pearson Correlation Coefficient and p-value between searching time and learning time in the swarm with different swarm sizes are shown in the table 5.3, table 5.4 and table 5.5.

Through analyzing the p-value, we find that the null hypothesis is rejected for every robot in the swarm of different sizes. Which means, the correlation between the searching time and the semantic knowledge learning time on every robot in different size of swarm is believed to be linear. But the coefficient varies from each robot in each size of swarm.

As is shown in the figure 5.11, the Pearson Correlation Coefficients of different sized swarm is shown in line-plots. For the reason that the robot in X-Axis is ordered by the time, this plot tells the changing of the correlation over time.

Robot	Pearson Correlation Coefficient	p-value
1	0.8890863094349426	<0.001
2	0.9483742302240594	<0.001
3	0.9688449627212664	<0.001
4	0.9572334436054362	<0.001
5	0.9373029237808638	<0.001
6	0.9128032326282319	<0.001
7	0.8651216298391436	<0.001
8	0.8261338539420441	<0.001
9	0.7389093055286127	<0.001
10	0.43214671544572825	<0.001

Table 5.3: The Pearson Correlation Coefficient and p-value Between Searching Time and Learning Time in The Swarm Sized 10.

Robot	Pearson Correlation Coefficient	p-value
1	0.7360944675497961	<0.001
2	0.9281100481643464	<0.001
3	0.9452757600848163	<0.001
4	0.9431280763140224	<0.001
5	0.937477697721824	<0.001
6	0.9203726845526657	<0.001
7	0.9106033647881997	<0.001
8	0.8952970743836418	<0.001
9	0.8553029689927922	<0.001
10	0.8421836169234055	<0.001
11	0.8127371494118334	<0.001
12	0.7953668293681524	<0.001
13	0.7439638927400143	<0.001
14	0.7170629179401834	<0.001
15	0.674611082808779	<0.001
16	0.6422783991116642	<0.001
17	0.5868246981582612	<0.001
18	0.4899114392855585	<0.001
19	0.3621760201275029	<0.001
20	0.20890812760789573	0.002

Table 5.4: The Pearson Correlation Coefficient and p-value Between Searching Time and Learning Time in The Swarm Sized 20.

Robot	Pearson Correlation Coefficient	p-value
1	0.8136053052286466	<0.001
2	0.8499476343333952	<0.001
3	0.8608336745482821	<0.001
4	0.8661147701717267	<0.001
5	0.8423591189706294	<0.001
6	0.8131797149220698	<0.001
7	0.7974136745021009	<0.001
8	0.7954637461600583	<0.001
9	0.7857163874199938	<0.001
10	0.7792870594667162	<0.001
11	0.7554362606788619	<0.001
12	0.7407843584235259	<0.001
13	0.7220936388203815	<0.001
14	0.7153091091040306	<0.001
15	0.6978354971776388	<0.001
16	0.6686629928489072	<0.001
17	0.6488210130925555	<0.001
18	0.6229696639653884	<0.001
19	0.609162916892536	<0.001
20	0.5673554257105893	<0.001
21	0.5757603156685251	<0.001
22	0.5589319064810745	<0.001
23	0.5327241477126083	<0.001
24	0.5227654653913445	<0.001
25	0.5007711657388801	<0.001
26	0.47261768560034034	<0.001
27	0.42678514346290863	<0.001
28	0.4198491785513344	<0.001
29	0.39949031353206566	<0.001
30	0.3738204271631049	<0.001
31	0.3367196920605246	<0.001
32	0.3221372461426433	<0.001
33	0.3044682251655573	<0.001
34	0.2689208606073905	<0.001
35	0.25880908201582037	<0.001
36	0.2367128494145458	<0.001
37	0.21032974200882976	0.002
38	0.16463501811438888	0.019
39	0.21687638042303622	0.002
40	0.1908009241669781	0.006

Table 5.5: The Pearson Correlation Coefficient and p-value Between Searching Time and Learning Time in The Swarm Sized 40.

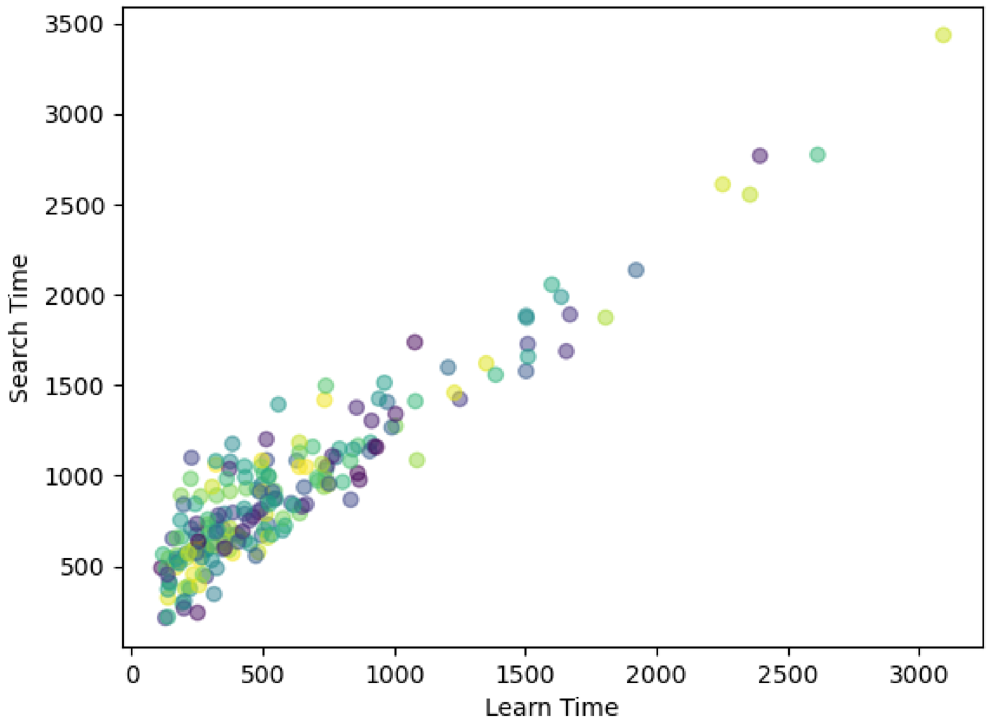


FIGURE 5.9: The Scatter Plot of The 5th Robot's Semantic Knowledge Learning Time And Searching Time in Swarm Sized 20.(The time is presented in seconds.)

The important thing we should keep in mind is that if the knowledge is learned by the robot itself, the learning time of this robot will have a strong correlation with the searching time. Because the learning happens mostly right after the robot finding the target.

The first few robots learning the semantic knowledge is closely related to the first few robots finding the target. Because the first few robots have a higher probability of learning the knowledge on their own, other than from other robots. By increasing the swarm size, the semantic knowledge spreads faster, being testified in the sub-subsection 5.2.2.2 and the sub-subsection 5.2.2.3. Which means, there will be less robots in the first few ones learn the knowledge by their own. So a larger swarm size will lead to small correlation in the early stage.

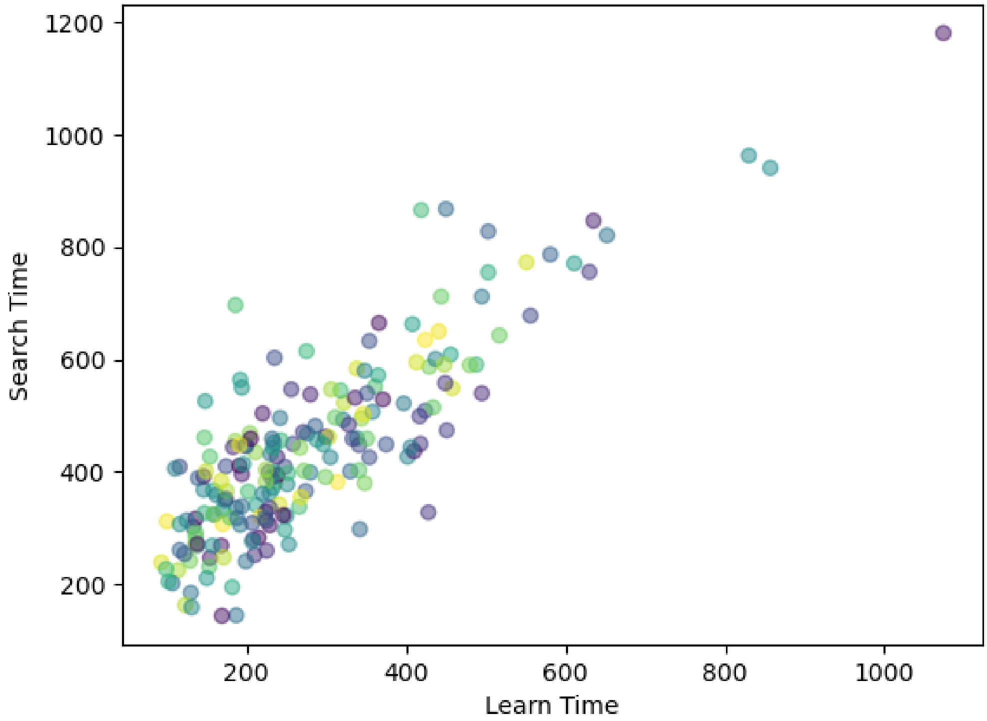


Figure 5.10: The Scatter Plot of The 5th Robot's Semantic Knowledge Learning Time And Searching Time in Swarm Sized 40.(The time is presented in seconds.)

5.4 Summary

In this chapter, we show that when semantic knowledge about the environment is not provided in advance, the efficiency of searching can be promoted by deploying a swarm of robots, compared with a single robot. The box-plot and statistical analysis illustrate the searching time of robots in the swarm sized 10 are less than a single robot when prior semantic knowledge is not provided.

Experiments are running to further explore the performance of the system when the size of swarm increases. By comparing the semantic knowledge learning time and searching time of the three representative robots (The First, Mid and Last Robot), the performance of the system on different sizes of swarm are illustrated. We find that the semantic knowledge learning time of all robots cuts down when the swarm size is increased. In terms of the searching time, half number of the robots in a swarm find the target faster as the swarm size increases. However,

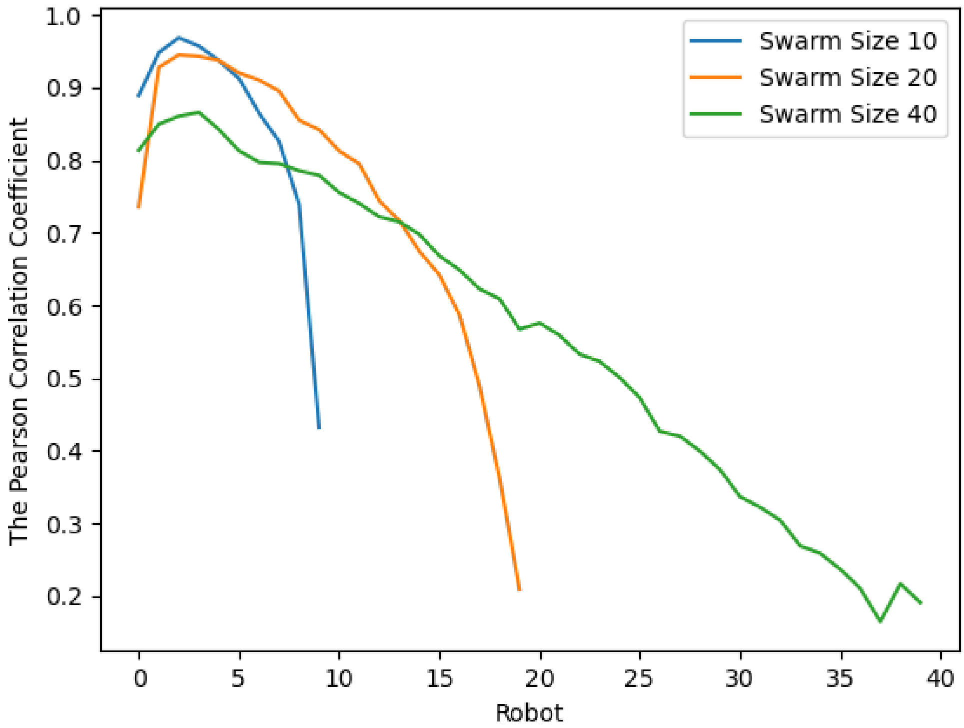


FIGURE 5.11: The Line Plot of The Pearson Correlation Coefficients between Searching Time and Learning Time in Different Sizes of Swarm.

when comparing the searching time of all robots that find the target, increasing the swarm size from 10 to 40 results in an increase.

In the last part, the correlations between the searching time and the semantic knowledge learning time are studied. The correlation is demonstrated as linear by the Pearson Correlation test. The Pearson correlation coefficient changes in the searching process of the entire swarm, as discussed.

Chapter 6

The Advantage of Prior Semantic Knowledge on Swarms

The Fuzzy Cognitive Map based robot controller is proven to be effective on a single robot when the semantic knowledge is provided in advance in the chapter 4. In the condition where the prior semantic knowledge can not be provided, the solution is given in the Chapter 5. The robot which has found the target will learn the semantic knowledge by itself and transmit the knowledge to its neighbouring robots, in order to help other robots in promoting their searching efficiency.

The proposed system is proven to be functioning both with and without prior semantic knowledge. The robot in the swarm without prior semantic knowledge could communicate with its neighbouring robots. So we are interested in what will happen if the robot in the swarm with prior knowledge also makes communication. The estimated current room and the confidence in this estimation are proposed as information being exchanged. Comparisons on the searching time used by each robot in the swarm with and without communicating such information are made.

After that, the different performances on searching time will be tested to illustrate the advantage of providing the prior semantic knowledge for the searching task. Meanwhile, the hypothesis that the prior semantic knowledge of the environment promotes the searching efficiency of a swarm of robots in an unknown environment is tested. At last, the differences in the advantage across different swarm sizes are explored.

Experiments will be run 200 times as what is done before.

6.1 Communication in The Swarm with Prior Semantic Knowledge

6.1.1 Introduction

Communication takes place in the experiments which are deploying a swarm of robots searching for a target without the prior semantic knowledge. The robots learn the semantic knowledge and try to transmit the knowledge to their neighbouring robots. In order to compare the performance of searching time between swarm of robots with and without the prior semantic knowledge, the question that whether the communication in the swarm of robots with prior knowledge should be activated, is raised.

As the prior semantic knowledge provided to every robot is complete for describing the entire searching area, there will be no need for communicating the knowledge between robots. And the information being exchanged still should keep local to each robot as proposed. The information is designed to be semantic, as shown in the figure 5.1. One of the useful information will be the current room estimated by the robot and the relevant confidence in the estimation.

Therefore, the Fuzzy Cognitive Map inspired robot controller in the Chapter 4 is extended to deal with the information of neighbouring robots' estimation on current room and their confidence in it. The robot holds the estimated current room number and the relevant confidence by itself. When two robots or more, running into the range of communicating, information about that are exchanged. For these robots, the current room may be estimated differently. Therefore, the mechanism to achieve an consensus is required. The problem of discrete consensus achievement or the Best-of-n problem is one of the most popular topics in swarm robotic research community. Solutions can be found as using the models of democratic voting in the work of Galam [147] . The model usually works for the conditions where a group of robots (more than 2) trying to make an collective decision on a controversy. In the way of voting, the decision is made on choosing the opinion of the most. If this model is used here, the value of confidence in estimation will be neglected. And most of the other solutions rely on the globalized estimation of benefits and costs of the decision being made. Therefore, in order to keep the proposed system working by only local information, the decision will be made on choosing the estimated current room with the bigger confidence value, or the biggest if 3 or more robots are involved.

For the robot which receives the estimated current room with the confidence which is greater its own confidence, it replaces its estimated current room and confidence with the received ones. In implementation, the probability of the unobservable state in the Hidden Markov Model representing room number is updated.

There is still one last thing to be fixed, which is the robot needs to be alerted when it is exiting a room. If not being solved, the robot will still hold strong confidence of being in the last room and compete with the robot holding the correct estimation on the current room. As the consequence , chaos is caused in the swarm. The solution is the same as how the robot knows exiting the Target Room in the Chapter 5. If the gateway (the floor in black colour in the figure 3.1) is detected, the confidence of being in each room is set to an equal and low value.

6.1.2 Results and Discussions

By introducing the timely communication of the estimated current room, we make an hypothesis that the timely communication of the estimated current room and the confidence in estimation between robots in a swarm promotes the searching efficiency, when the semantic knowledge of environment is provided in advance. The hypothesis is tested on a swarm of 10 robots, comparing the searching time by each robot in the order of finding the target. As usual, 200 trials are run on both sets.

The collected searching times are presented as box-plots, shown in the figure 6.1. As only small difference in the searching time can be found, statistical analysis of the data is required. The U-test comparing each robot ordered by its searching time is given the null hypothesis that the robot which exchanges the estimated current room with its neighbouring robots uses a longer time for searching than the robot does not. The values of p are shown in the table 6.1. The p-values of the 6_{th}, 7_{th}, 8_{th} and 9_{th} robot are less than 0.05, rejecting the null hypothesis. The relevant A-test scores and the effect sizes are presented in the table 6.1 as well. The effects are all calculated as negligible on 10 robots. In conclusion, the searching time of each robot in the swarm sized 10 with communication of the estimated current

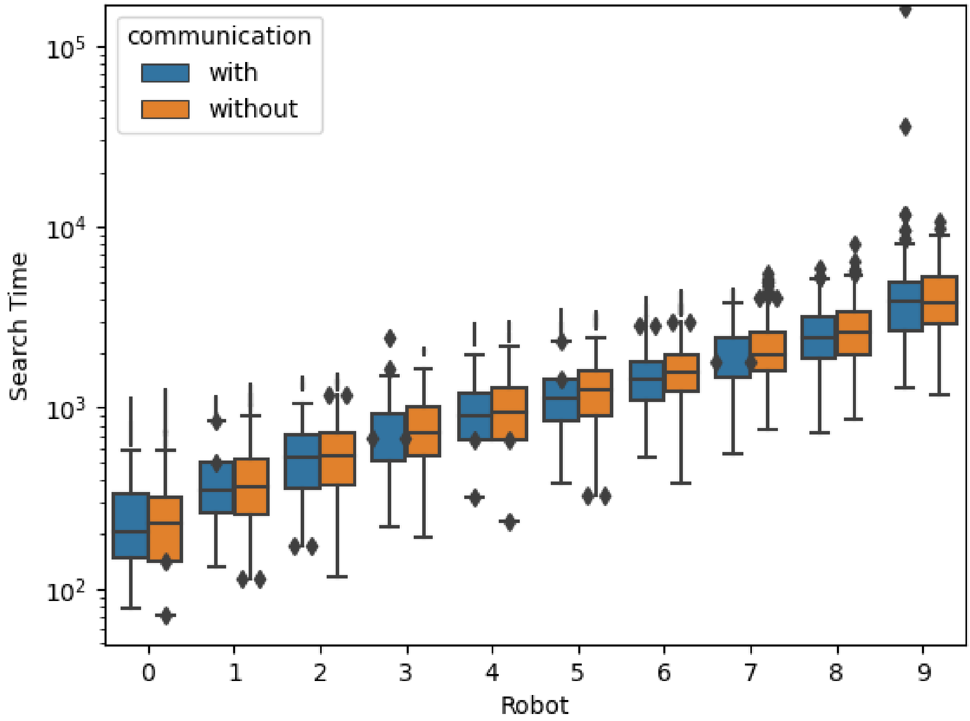


FIGURE 6.1: The Box-plot of Search Time Used by Each Time in Swarm Sized 10 with and without Communicating The Estimated Current Room and The Confidence in Estimation. (Search Time is presented in seconds. 0 to 9 on the X-Axis denote the robots ordered by the time finding the target. The communication denotes the information exchanging of robots' estimated current room and the relevant confidence value; with or without follows the literal meaning.)

room and the confidence of estimation does not show statistically difference with the robot without communication.

Therefore, our former hypothesis about the communication can not stand. This conclusion goes contrary to our first thought, because that a map always works the best when you can timely localize your current position on it. The reason why the robot with communication does not give a better performance could be blamed on the simple searching behaviours. As the detailed searching in this work relies on randomness to find the target, the information of estimated current room is not well utilized. Therefore, there will be no needs to further explore the differences on the swarm sized 20 and 40.

For the reason that the communication of the estimated current room can not work effectively for promoting the searching efficiency, the comparison between swarm of robots with and without the prior semantic knowledge is made in the condition

Robot	value p of the U-test	A-test score and effect size
1	0.510	0.500 (negligible)
2	0.419	0.494 (negligible)
3	0.224	0.478 (negligible)
4	0.078	0.459 (negligible)
5	0.067	0.456 (negligible)
6	0.021	0.441 (negligible)
7	0.008	0.430 (negligible)
8	0.019	0.440 (negligible)
9	0.032	0.446 (negligible)
10	0.174	0.472 (negligible)

TABLE 6.1: The p-value of U-test And A-test Score Comparing Searching Time of Each Robot in The Swarm Sized 10 With and Without Communicating The Estimated Current Room and The Confidence of Estimation.

that the robot in the swarm with prior knowledge makes no communication with its neighbouring robots.

6.2 The Advantage Shown on Different Sizes of Swarm

As being illustrated in the section 6.1, the communication in the swarm of robots with prior semantic knowledge can not promote searching efficiency at current stage. So the comparisons are made between swarm of robots with prior semantic knowledge but without communication and swarm of robots without prior knowledge but with communication of the semantic knowledge which is learned by robots.

The swarm size changes from 10 to 20, and to 40.

6.2.1 The Advantage on The Swarm Sized 10

As is shown in the figure 6.2, the difference in searching time can be found on every robot from the swarm with prior semantic knowledge to the swarm without prior knowledge.

To make statistical analysis, the U-test and the A-test are applied as before. Results are shown in the table 6.2. The U-tests are all given the null hypothesis

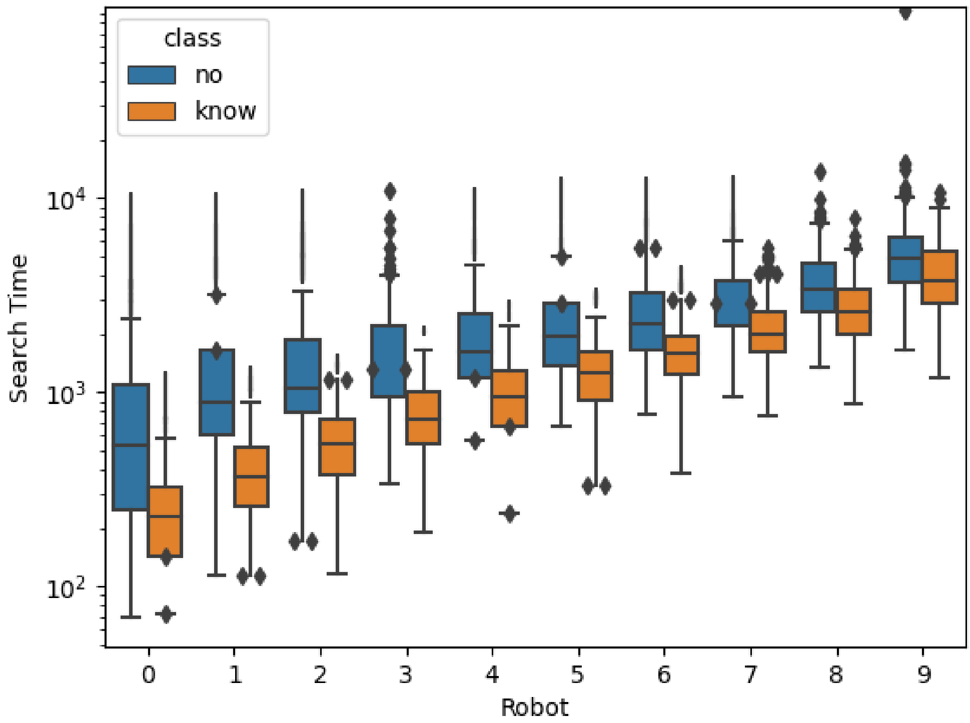


FIGURE 6.2: The Box-plot of Search Time Used by The Robots in Swarm Sized 10 With and Without The Prior Semantic Knowledge. (Search Time is presented in seconds. 0 to 9 on the X-Axis denote the robots ordered by the time finding the target. For the class: no stands for the robot without prior semantic knowledge; know stands for the robot with prior knowledge.)

that the robot in the swarm with the prior semantic knowledge takes a longer time for finding the target than the robot in the swarm without the prior knowledge. As is shown, the null hypotheses are all rejected by the p-values. The scores of A-tests present large effect size on the robots which find the target earlier. On the last two robots of finding the target, the effect size is shown as medium and small.

In conclusion, each robot in the swarm sized 10 with prior semantic knowledge takes a shorter time for searching than the robot in the swarm without prior knowledge. Thus, the advantage of prior semantic knowledge is well illustrated on the swarm of 10 robots.

6.2.2 The Advantage on The Swarm Sized 20

For the swarm of 20 robots, the searching time of each robot in the swarm with and without prior semantic knowledge are box-plotted in the figure 6.3.

In the same way, the searching time of each robot in both sets are compared by the U-test and A-test. The results are shown in the table 6.3. The null hypotheses of the U-test are the robot in the swarm with the prior semantic knowledge take a longer time for finding the target than the robot in the swarm without the prior knowledge as well. The values of p all reject the null hypotheses. The score of U-test on first robot finding the target shows a small effect size. Afterwards, the U-test produces scores which are descending from large effect size to medium, then to small and to negligible at the last robot finding the target.

To summarize, in the swarm of 20 robots, the prior semantic knowledge promotes the searching efficiency of most robots (as 19 robots in total number 20). As the U-test score of the last robot finding the target shows as negligible, the prior knowledge makes no difference for leading all robots to find the target. Therefore, the advantage of the prior semantic knowledge on the swarm of 20 robots exists on the robots which find the target early. But it becomes trivial, if all robots are required to find the target.

Robot	value p of the U-test	A-test score and effect size
1	<0.001	0.747(large)
2	<0.001	0.845 (large)
3	<0.001	0.840 (large)
4	<0.001	0.820 (large)
5	<0.001	0.800 (large)
6	<0.001	0.777 (large)
7	<0.001	0.743 (large)
8	<0.001	0.743 (large)
9	<0.001	0.682 (medium)
10	<0.001	0.647 (small)

TABLE 6.2: The p-value of U-test And A-test Score Comparing Searching Time of Each Robot in The Swarm Sized 10 With and Without Prior Semantic Knowledge.

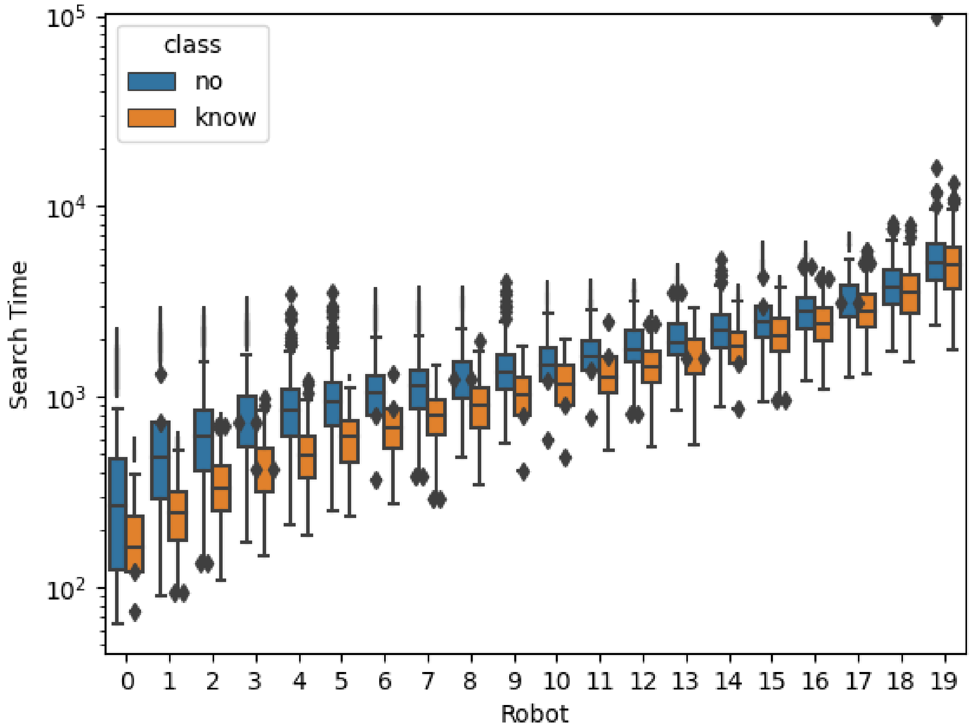


FIGURE 6.3: The Box-plot of Search Time Used by The Robots in Swarm Sized 20 With and Without The Prior Semantic Knowledge. (Search Time is presented in seconds. 0 to 19 on the X-Axis denote the robots ordered by the time finding the target. For the class: no stands for the robot without prior semantic knowledge; know stands for the robot with prior knowledge.)

6.2.3 The Advantage on The Swarm Sized 40

The box-plots shown in the figure 6.4 reveal the different searching time of each robot in swarm sized 40 with and without the prior semantic knowledge.

The values of p calculated by the U-tests and the scores of A-tests comparing each robot in the swarm with and without the prior semantic knowledge are shown in the table 6.4. The U-tests are given the null hypotheses that the robot in the swarm with the prior semantic knowledge takes a longer time for finding the target than the robot in the swarm without the prior knowledge. The p-values of the 1_{st} to the 33_{rd} robot reject the null hypothesis, while the others cannot. The scores of U-tests show as negligible effect sizes in the 1_{st} robot, the 25_{th} robot and the 29_{th} to 40_{th} robots. Large effects are only shown on the 4_{th} to 8_{th} robots.

To sum up, the descending in the searching time of each robot in the swarm sized 40 making by the prior semantic knowledge, could be found on robots which find the

Robot	value p of the U-test	A-test score and effect size
1	<0.001	0.646 (small)
2	<0.001	0.781 (large)
3	<0.001	0.799 (large)
4	<0.001	0.828 (large)
5	<0.001	0.827 (large)
6	<0.001	0.809 (large)
7	<0.001	0.800 (large)
8	<0.001	0.777 (large)
9	<0.001	0.761 (large)
10	<0.001	0.742 (large)
11	<0.001	0.729 (medium)
12	<0.001	0.725 (medium)
13	<0.001	0.728 (medium)
14	<0.001	0.701 (medium)
15	<0.001	0.687 (medium)
16	<0.001	0.663 (small)
17	<0.001	0.633 (small)
18	<0.001	0.613 (small)
19	0.005	0.573 (small)
20	0.023	0.557 (negligible)

TABLE 6.3: The p-value of U-test And A-test Score Comparing Searching Time of Each Robot in The Swarm Sized 20 With and Without Prior Semantic Knowledge.

target early, except the first robot. And in the swarm sized 40 the ratio of robots, which show the prior semantic knowledge could promote searching efficiency, to the entire swarm is less than the ratios in the swarm sized 10 and 20. In particular, the differences in the searching time are presented to be negligible on the last 12 robots finding the target. In conclusion, the advantage of the prior semantic knowledge also can be found in the swarm of 40 robots. But the advantage does not take place on the first robot and the last 12 robot finding the target. And from the view of the entire swarm, the advantage in leading all the robots to the target is shown to be non-existent.

6.2.4 Differences Across Swarm Sized 10, 20 and 40

The one last thing which interests us is the comparison of the advantage between different sizes of swarm. The T-tests [131] are made to calculate the differences in the mean values of the each robot's searching times in 200 trials of experiments.

Robot	value p of the U-test	A-test score and effect size
1	<0.001	0.566 (negligible)
2	<0.001	0.652 (small)
3	<0.001	0.721 (medium)
4	<0.001	0.750 (large)
5	<0.001	0.768 (large)
6	<0.001	0.762 (large)
7	<0.001	0.755 (large)
8	<0.001	0.746 (large)
9	<0.001	0.726 (medium)
10	<0.001	0.706 (medium)
11	<0.001	0.699 (medium)
12	<0.001	0.691 (medium)
13	<0.001	0.681 (medium)
14	<0.001	0.678 (medium)
15	<0.001	0.671 (medium)
16	<0.001	0.661 (small)
17	<0.001	0.647 (small)
18	<0.001	0.635 (small)
19	<0.001	0.634 (small)
20	<0.001	0.626 (small)
21	<0.001	0.615 (small)
22	<0.001	0.604 (small)
23	<0.001	0.591 (small)
24	0.001	0.587 (small)
25	0.007	0.570 (negligible)
26	0.003	0.578 (small)
27	0.001	0.587 (small)
28	0.003	0.579 (small)
29	0.011	0.565 (negligible)
30	0.019	0.559 (negligible)
31	0.012	0.564 (negligible)
32	0.026	0.556 (negligible)
33	0.040	0.550 (negligible)
34	0.124	0.533 (negligible)
35	0.175	0.526 (negligible)
36	0.190	0.525 (negligible)
37	0.508	0.499 (negligible)
38	0.579	0.494 (negligible)
39	0.549	0.496 (negligible)
40	0.863	0.468 (negligible)

TABLE 6.4: The p-value of U-test And A-test Score Comparing Searching Time of Each Robot in The Swarm Sized 40 With and Without Prior Semantic Knowledge.

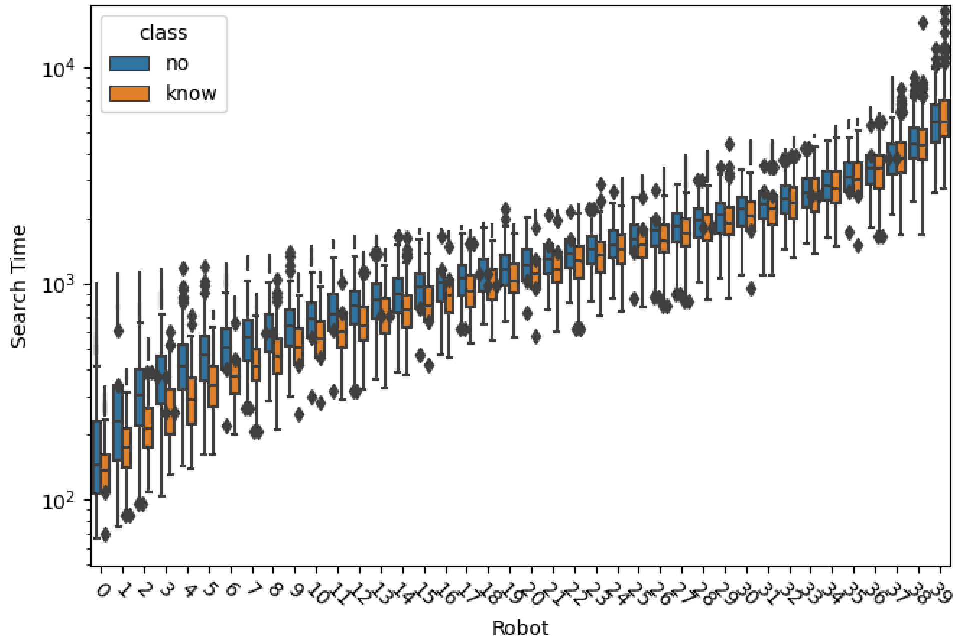


FIGURE 6.4: The Box-plot of Search Time Used by The Robots in Swarm Sized 40 With and Without The Prior Semantic Knowledge. (Search Time is presented in seconds. 0 to 39 on the X-Axis denote the robots ordered by the time finding the target. For the class: no stands for the robot without prior semantic knowledge; know stands for the robot with prior knowledge.)

The null hypotheses of the T-tests are given as the differences between the means of the each robot's searching time (200 trials) in the swarm with prior semantic knowledge and in the swarm without prior knowledge is greater than 0.

The results getting from T-tests are shown in the table 6.5. As is shown, the p-values of the 30_{th} to 40_{th} robot in the swarm sized 40 cannot reject the null hypothesis. The differences in the mean value are used for measuring the advantage of the prior semantic knowledge on each robot. Combining the p-values of the U-tests and scores of A-tests in the sub-section 6.2.1, sub-section 6.2.2 and sub-section 6.2.3. The value of difference is set as 0 for the robot with p-value from U-test greater than 0.05, or with A-test score in negligible effect size, or with p-value from T-test greater than 0.05.

The difference on the advantage of the prior semantic knowledge in different size of swarms is shown in the figure 6.5. As is shown, the line of the swarm with a larger size is drawn lower. Therefore, the advantage of the prior semantic knowledge goes down with the increase of swarm size. In another point of view the disadvantage

Robot	Swarm Sized 10		Swarm Sized 20		Swarm Sized 40	
	Difference	p-value	Difference	p-value	Difference	p-value
1	623.6	0	186.755	0	43.26	0
2	879.83	0	332.89	0	76.715	0
3	945.495	0	363.665	0	103.56	0
4	951.005	0	408.255	0	121.48	0
5	976.93	0	421.76	0	134.575	0
6	986.685	0	419.995	0	136.935	0
7	981.315	0	422.09	0	139.69	0
8	977.46	0	408.71	0	140.29	0
9	991.07	0	411.14	0	134.43	0
10	1954.12	0.0014	408.525	0	126.83	0
11			403.585	0	130.925	0
12			413.565	0	131.05	0
13			445.41	0	125.91	0
14			426.61	0	125.465	0
15			424.595	0	123.235	0
16			405.74	0	124.935	0
17			368.67	0	115.315	0
18			363.825	0.0001	109.4	0
19			331.83	0.0028	112.33	0
20			873.645	0.0428	113.285	0
21					109.985	0
22					104.02	0.0002
23					90.07	0.0013
24					85.065	0.0037
25					70.96	0.017
26					87.655	0.0076
27					101.88	0.0042
28					88.255	0.0165
29					75.035	0.0414
30					71.07	0.0627
31					87.49	0.0389
32					79.565	0.0669
33					70.915	0.1095
34					41.865	0.2522
35					20.605	0.3806
36					36.39	0.3169
37					-40.995	0.6862
38					-52.695	0.7041
39					-82.72	0.7381
40					-414.815	0.978

Table 6.5: The Differences in Means and p-values of The T-test Comparing Comparing Searching Time of Each Robot With and Without Prior Semantic Knowledge in The Swarm Sized 10, 20 and 40.

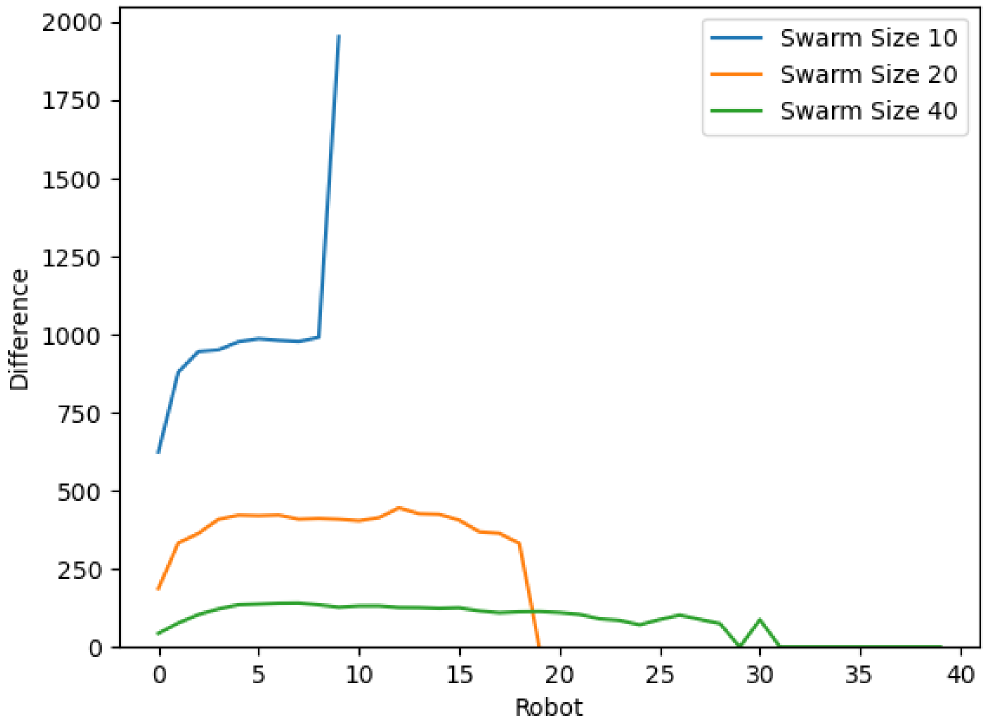


FIGURE 6.5: The Line Plot of The Difference of Each Robot's Searching Time between Swarms With and Without Prior Semantic Knowledge. (The lines are for the different swarm sizes, as annotated.)

of the swarm of robots without prior semantic knowledge could be compensated by increasing the swarm size.

6.3 Summary

In this chapter, the hypothesis that the searching efficiency of robots in a swarm can be promoted by providing semantic knowledge about the environment in advance is tested. At first, we try to make the robot in the swarm with prior semantic knowledge communicate with its neighbouring robots. The information being communicated is designed to be the estimated current room and the confidence in estimation. Experiments are executed to compare the searching time of each robot in a swarm of 10 robots with and without communication. The results show a non-significant difference between the swarm with and without communication. And we explain this by the simple behaviours designed for searching. For that reason, we decide to test the performance of the swarm with prior knowledge in the set-up of making no communication.

After that, the searching time of each robot in the swarm with and without prior semantic knowledge is compared. The swarm size increases from 10 to 20, and to 40. In the swarm sized 10, the prior semantic knowledge promotes the searching efficiency of every robot. However, in the swarm sized 20 and 40, the prior semantic knowledge makes no difference for help all the robots find the target. And in the swarm sized 20 and 40, the advantage of providing prior semantic knowledge about the environment only shows in the robots which find the target in the early stage. Therefore, the advantage will depend on how a searching task is defined for the swarm. As a further exploration, we measure the power of the advantage crossing different swarm sizes, which shows a descending trend when the swarm size is increased. So we propose that when a swarm of robots perform the searching task without providing semantic knowledge of the environment in advance, a large swarm size could compensate for the weakness in lacking prior semantic knowledge.

Chapter 7

Evaluation and Conclusions

This chapter draws conclusion on the thesis by summarising the contribution made in each chapter. Then we discuss the limitations of this work and give direction to the possible future work. At last, to evaluate this thesis against success, the main hypothesis is revised.

7.1 Summary and Contributions

This section summarizes each chapter in the thesis and gives the relevant contributions.

- Chapter 2 Background and Related Works

  This chapter reviewed the topic of swarm robotics and the techniques required by searching tasks, which are the strategies for exploration, the mapping methods for recording searching environment and the models for managing semantic knowledge. The searching problems are discussed in details. At last, the approaches for statistics analysis are reviewed.

  Contribution: A review of swarm robotics, searching tasks related techniques and a detailed discussion on searching problem. The approaches are criticized in according with the properties of swarm robots.

- Chapter 3 Searching Task and The Fuzzy Cognitive Map Inspired Robot Controller

This chapter gives the concrete definition of the searching task in a simulator. The robot controller is devised for accepting semantic knowledge as inputs. The system is built by designing task-oriented behaviours on the top-level and dividing those behaviours into low-level reactive behaviours. The model for utilizing the semantic knowledge is found.

Contribution: We create the robot controller by the novel application of fuzzy cognitive map for the searching task in swarm robots. And the incorporation of semantic knowledge to the fuzzy cognitive map in this work presents a possible way of constructing the robot controller with semantic knowledge as input. The Hidden Markov Model is shown as fitting well for recording and using the search task related semantic knowledge on the swarm robots.

- Chapter 4 Prior Semantic Knowledge Utilized by A Single Robot

  This chapter starts by testing the effectiveness of prior semantic knowledge on a single robot. The challenge is raised about the necessity of using the semantic knowledge. The challenge is answered by enriching the relationships between object and room. Experiments are run with different placements of the different objects. The searching time of a robot which is given the randomized semantic knowledge is also tested. At last, the reliance on the Specific Object is explored.

  Contribution: The enrichment of the relationships between object and room generalize the scene where the system can be applied. And the variation of object which holds different relation with the relevant rooms shows as a guidance for the semantic knowledge provided to robot for the purpose of promoting searching efficiency.

- Chapter 5 Robots Learning The Semantic Knowledge

  This chapter presents the mechanism of the semantic knowledge learning, communication and utilization for the robot in a swarm without providing semantic knowledge in advance. The experimental results reveal that the semantic knowledge learned by the robot can help promote the searching efficiency of other robots in the swarm. The comparisons of the searching time and the semantic knowledge learning time are made between robots in the swarm with different size. The correlations between the searching time and the semantic knowledge learning time are explored.

Contribution: The mechanism of the semantic knowledge learning, communication and utilization comes as a novel method for knowledge construction and sharing in the swarm robotic. In the process of knowledge learning, the robot makes task-oriented information recording. It provides a solution for mapping the environment by the swarm robot with out-dated hardware. The information being transmitted between robots is the relationships between object and room. It is shown as novel in the discipline of swarm robotic. And the experimental results point the way for using swarm of robots performing searching tasks in practical scenes.

- Chapter 6 The Advantage of Prior Semantic Knowledge on Swarms

  This chapter shows the enrichment of communication in the swarm of robots with prior semantic knowledge. The advantage of the semantic knowledge about environment provided in advance is explored in different sizes of swarm. The advantage is compared crossing the swarm of different sizes.

  Contribution: The communication of the estimated current room and the confidence in estimation is an attempt on further exploitation of the semantic knowledge in a swarm. It could be proposed as a solution for other swarm robotic tasks. The results of the experiments confirm the advantage of providing semantic knowledge in advance.

7.2 Limitations

- The searching behaviours although work for the searching task we proposed in this work, shows their limitation when the robot in the swarm with prior semantic knowledge communicates more semantic information with its neighbouring robots. In order to fully exploit the advantage taken by the semantic knowledge, more complex behaviours need to be designed in providing more source of sensor inputs.

- The searching scenario is still not complex enough compared with the practical scenes, in which massive number of objects will be recognized. Therefore, if the semantic knowledge is given as a huge database describing the relationships between objects and rooms, the robustness of this system will be challenged.

- The robot which are provided semantic knowledge about the environment in advance, is working under the condition that the semantic knowledge gives a complete description about the entire searching arena. In practice, this is not feasible.

- The system relies on the correctness of the semantic knowledge and the inferred target room. If the human operator provides the wrong relationship between object and room the system will not help promote the searching efficiency, as shown by the randomized semantic knowledge experiment. In the same way, if the inferred target room is provided by wrong, the promotion of searching efficiency will not exist.

- The system also relies on the correct recognition of the object. Faulty object recognition will let the robot with prior semantic knowledge estimate a wrong room and make the wrong decision on switching behaviour. For the robot which learns the semantic knowledge, fault object recognition could cause a wrong relationship between object and room being learned. And the wrong relationship will be propagated across the swarm, which lead to controversies over semantic knowledge in the swarm.

- The robot is working in a 2-D space. If the searching task happens in the 3-D space, the navigation strategy needs to be more complex.

- The searching problem in this work was initially found on searching the source of gas, which is static. However, if the target is moving between rooms, the robot in this work still can not updated the target room for searching. And the system in this work is designed for a single target. In some cases, there will exist multiple targets with different urgency for searching.

7.3 Future Work

This section suggests possible areas for working in the future.

- The research in this thesis has focused on showing in principle that the semantic knowledge about environment can promote the searching efficiency of robot in the unknown environment. Therefore, the work presented in this thesis is not the optimal solution for searching tasks. As being limited by the source of available sensor inputs, only basic searching behaviours are designed. And the discussion in the chapter 6 shows that the full potential of deploying semantic knowledge in searching tasks, is yet to be reached. In the next step, more complex searching behaviours will be devised by leveraging new source of sensor inputs. Then the benefit of timely semantic communication between robots can be shown. And the system needs to be tested on real robots.

- To deal with the situation that, the semantic knowledge about the environment is partially provided. In this condition, the robots are required to complete the relationships between objects and rooms. This could be solved by following the design in the chapter 5, and to enrich the knowledge learning and communication.

- To deal with the situation that, the prior semantic knowledge contains some wrong relationships from the objects to the rooms. This calls for an on-line judgement strategy, so as to make the swarm of robots estimate the correctness of the prior semantic knowledge. The judgement strategy could be implemented by combining the searching time of other robots. Besides, the robot should have a value which measures the belief over its knowledge, like the confidence in estimation shown in the chapter 6. After that, an on-line semantic knowledge correcting system will be implemented.

- To deal with the situation that, the human operator infers a wrong target room. This can be solved in the same way as the implementation of on-line semantic knowledge correcting system.

- To deal with the situation that, faulty object recognition happens on a robot in the swarm. This may cause two type of problems. If the robot has already hold complete and correct semantic knowledge, the problem will fall into the field of fault-tolerant research. The robots without the fault, should help the fault robot by changing the fault robot's experience (as shown in the figure 3.3) on using the knowledge.

  But if the robot still needs to learn the semantic knowledge when the fault on object recognition exists. The fault robot will learn wrong semantic knowledge with object being erroneously recognized. This causes the fault robot running in the swarm as an outlier. The mechanism of detecting the outliers in a swarm should then be devised.

- To deal with the situations that, the target for searching moves between rooms or there exists multiple targets for searching in the working area. Both conditions will make the robots perform detailed searching in multiple rooms. If it is a single target moving between rooms during the searching process, the robots should timely record the existence of target in a room and share with other robots. But if it is the situation of multiple targets, the robots need to schedule the priorities of the targets.

7.4 Conclusion

Chapter 1 defined the following general research hypothesis that guided the research presented in this thesis:

The application of semantic knowledge based mechanism and decentralised communication in a swarm robotic system increases its speed, efficiency when performing search tasks. Also, an expert semantic knowledge about the searching environment could further increase the efficiency if provided in advance.

It was shown that a single robot that utilizes the prior semantic knowledge about the environment by the robot controller devised in this thesis, consumes less time for finding the target, compared with the one without the prior knowledge in the chapter 4. We designed the semantic knowledge learning, communicating, and utilizing mechanism for the robot without prior knowledge in the chapter 5. With the semantic knowledge based mechanism, the robots in a swarm find the target faster than a single robot, in the absence of prior semantic knowledge. The chapter 6 was demonstrated to compare the searching time of robots in the swarm with and without prior semantic knowledge. The advantage of the provision of prior semantic knowledge can be seen in the different sizes of the swarms.

The initial goals of this thesis have been reached, and the research presented has been shown as a valuable new mechanism for storing and exchanging semantic knowledge about the environment in deploying swarm robots. The mechanism follows the local principle of swarm robotics and takes advantage of the parallelism feature. The proposed system architecture shed light on incorporating human expert knowledge into swarm robotic systems. The research presented in this thesis therefore shows a contribution to novel exploitation of semantic knowledge in swarm robotic systems for target searching.

Appendix A

Employing Semantic Map Labelling to Improve Swarm Robotic Searching Tasks

This is an unpublished paper during this PhD study.

Employing Semantic Map Labelling to Improve Swarm Robotic Searching Tasks

Guangsha Xu[1], Wei Li[1], Jon Timmis[1], and Martin Trefzer[1]

Department of Electronic Engineering, University of York, York, UK
gx535@york.ac.uk

Abstract. Swarm robotics is often discussed in the context of search and rescue applications due to the potential advantages of efficiency, scalability and redundancy when compared to a single robot approach. However, information sharing mechanisms that are required to coordinate are often limited, thus reducing the potential of the swarm. In this paper, we propose a novel knowledge construction and sharing approach through the use of semantic abstraction of raw sensor readings. Experimental results, based on searching time, demonstrates in principle that the proposed approach is able to efficiently work even in the presence of sensor noise.

Keywords: Search and rescue, swarm robotics, semantic mapping, knowledge

1 Introduction

Robotic Search and Rescue(SaR) problems have generated significant interest in the robotic research community during the last decade [33]. The applications range from the rescuing of victims after a catastrophe, to hazardous gas leaking localization.

In such applications, efficiency is a key requirement. Intuitively, efficiency can be achieved through the introduction of more robots. However, with an increase in the number of robots, challenges such as co-ordination, dispersal, effective information sharing and exploration arise. In addition, robots will face a variety of other issues including:

- The size of the searching arena is typically beyond the range of individual robot's sensor.
- Messages only can be shared between robots in a limited range, i.e. often there is no global communication.
- Global localization of robot, e.g. GPS device, is not reliable or even not available.
- Robot sensor input is often affected by noise.

Swarm robotics is the study of designing groups of robots that operate without relying on any external infrastructure or on any form of centralized control [14].

Swarm robots are often simple with respect to hardware implementation, which makes deploying large number of robots feasible in budget. According to Brambilla [8], scalability and self-redundancy can be implemented in swarm robotic systems. For SaR tasks, the fault tolerant feature of swarm robotic system also guarantees resistance against individual robot failure, e.g. control software hang or physical embodiment damage. However, work by [37] demonstrated that under certain conditions, swarms are not as reliable as first thought. Since direct control mechanism being infeasible, the success of swarm robotic systems usually emerge from collective functioning of individual robots. This causes the designing and modelling of swarm robotic systems to be challenging.

Recent reviews [49, 11] provide a good overview of state-of-the-art decentralized searching algorithms for swarm robots. Many algorithms are based on swarm intelligence optimization approaches, such as the particle swarm optimization (PSO) [58] or bees algorithm (BA) [18]. Modifications are often made to eliminate global evaluation or there are modifications made to the mechanisms to update behaviour. However, solutions that are shared between robots are still dependent on global X-Y coordinates. In addition, typically these algorithms assume a global target coordinate can be transmitted as the final outcome of the searching process. In the context of SaR, new mechanisms for sharing targets with either other robots or human operators need to be devised.

Simultaneous localization and mapping (SLAM) has been extensively studied for robotic navigation [9]. A robot models the environment while localizing itself in a defined coordinate frame. Hardware requirements have hindered its adoption to existing swarm robotic platforms. However, Milford et al. [35] propose the RatSLAM algorithm inspired by a hippocampus model of rodent, which would be suitable for a low-computational platform, often found in swarm robotic applications. Heath et al. [19] apply and extend the algorithm to work with multiple robots. In this work, robots learn and share lexicons which describe different regions to build their maps.

In other work, a semantic spatial hierarchy is used by Kuipers et al. to model the experiment arena as a map in [30]. Distinctive points can be perceived by robots using simple sensors, e.g. sonar distance sensor. Robots can then exchange a description on the points, and the paths combining them. The RoboEarth project extend the semantic model to a general robotic knowledge semantic web [55]. These works provide inspiration for the control strategy of robot described in Section 3. However these works focus on building a detailed map, whilst this paper solves the SaR problem by task-oriented abstraction of sensor readings.

Sharing knowledge and experience between robots in a swarm tasked with SaR in a decentralized way, is a challenging task. In this work, what we mean by knowledge and experience are defined as follows.

- Knowledge is typically composed of relevant descriptions about robots in a vicinity, shared landmarks and required targets.
- Experience records how robots should act when they possess the relevant knowledge.

For example, an overlapped mapping area in two robots can be used for relative position estimation and map fusion [47]. Conventional methods rely on global X-Y coordinate and a global compass to describe the relative position and orientation. However, as discussed before, global localization is often not available, which reduces the application of these methods. In addition, existing methods are often computational resource intensive and time consuming, thus can be challenging to be implemented on limited hardware platforms.

This paper proposes a novel swarm robotic system for SaR by modifying the way of information recording and exchange. In the proposed system, the mapping result of a single robot is processed by task-oriented semantic abstraction, eliminating the reliance on precise sensor readings or coordinates which are prone to variations and noises. This abstraction of mapping results is stored as robot knowledge and used to communicate with other robots. Robots switch searching strategy based on this knowledge. The advantages of this work are that the use of cheap sensors in swarm robots for SaR tasks. As a result, it could be potentially scalable to a large size of swarm.

This paper is organized as follows. Section 2 reviews searching related works and tries to get the taxonomy. Section 3 gives a proposed solution using a novel semantic mapping approach. Section 4 provides details of the implementation using a simulation based approach. Experiment setup and results evaluating efficiency of the proposed system are shown in Sections 5 and 6. Finally, conclusion and future work are presented in Section 7.

2 Related Work

2.1 Searching Problems in Different Disciplines

The problem of searching can be interpreted differently by researchers from different disciplines beyond robotics, e.g. biology. Although the motivation of researchers and application context may drastically vary in robotic community, research outcomes including models and algorithms contribute to the design of robotic systems.

According to [52], the animal foraging behaviours is analogous to robotic searching tasks. Results of most reviewed works conclude that robot should subdivide the searching arena into small patches. Due to the existence of patch boundary, detailed search can be conducted by animals (robots) inside each patch. However, Jesus and Robin [52] also argue there is fundamentally different motivation between animal foraging and robotic search tasks. Exhaustive search should be performed by robots while animals try to maximize the energy level of their nets to stay alive.

The study of [16] is based on the searching behaviour of T cell in the immune system. Specifically, T cells search for dendritic cells in lymph nodes for the purpose of initiating the adaptive immune response. The requirement of efficiency, scalability, robustness against errors and flexibility can be shared with robotic searching tasks. The experiment result with a robot swarm suggests efficiency of levy search which models the T cell searching behaviour depends strongly on the distribution of targets.

Early application of robots in search and rescue scenario appeared in the 2001 World Trade Center (WTC) collapse [50] for rescuing victims after the tragedy. The deployed robots are equipped with abundant sensors so as to develop a versatile mobile robot platform in extreme operational environments. According to [33], the control of current rescue robots mostly relies on tele-operation by human operators. The study toward autonomous control scheme has been extended hierarchically. Low-level behaviours include robots traversing uneven terrain and SLAM to build maps of the search and rescue scene. On the high-level, collaboration and task allocation both between human-robot and robot-robot are the directions worth exploring.

Review paper [5] surveys robots used for localizing source of pollution in environmental monitoring applications. Chemotaxis (gradient-based) and anemotaxis (wind detection) methods show weak resistance against propagation turbulence. So infotaxis methods are devised based on information principle with the propagation model of pollutant in turbulent medium. While the works reviewed are designed for single robot source localization, applying them into multi-agent systems also has been envisaged as the future direction. [27] also categorizes works on robotic odor localization by the environmental conditions, which determines how the odor is dispersed. Reynolds number of the flow is used to quantify different fluid flow situations. In low Reynolds number flow, the fluid motion is dominated by diffusion and the dispersal of the chemical concentration can be modelled by a Gaussian distribution. Searching robots may come across this situation when they work underground. But in high Reynolds number flow, turbulence dominates, which is common in aerial and aquatic conditions. This may cause problems of sub-optimum.

2.2 Robotic Searching Task Variants

The problem of target searching can be configured with different set up in robotics research community. Parameters and assumptions vary in different works, e.g. number of targets or the mobility of targets. We try to classify searching tasks applying robots by referring the work of [49] and [23]. This will help us narrow down the design of search problem and concentrate on certain scenario.

- Number of targets. When searching for single target with MRS (multiple-robots system), most research are focused on improving the accuracy of estimated target position. However, in multiple targets scenario the main attention is paid to the assignment of robots to a target by certain mechanism, e.g. utility. The problem also diverges when the ratio between the number of target and robots changes.
- Mobility of targets and robots. The complexity of searching problem will be increased if target could move. And the moving pattern (range, mode and models) of target may be diverse. Relatively, various robotic platforms have emerged for the purpose of searching with different modes of mobility, e.g.

wheeled robot moving on ground, biomimetic robot swimming underwater or quad-rotor flying in the air.
- Complexity of environment. Environment acts as key role in robotic searching tasks as most of the algorithms rely on sensory perceptions from it , e.g. chemotaxis method in robotic source localization application. The arena of searching can be defined as open space or with boundaries. This makes difference at the time choosing path-planning method. In addition, the existence of obstacles also will collide with robots trajectory.
- Type of cooperation and coordination among robots. The cooperation of multiple robots targeted at either improving the accuracy of target estimation or optimally allocating robots to targets. The approaches coordinating robots can be categorized into explicit or implicit coordination. Though implicit coordination has its drawback of unclearance, the brevity of information saves the communication bandwidth and guarantees the scalability when the number of robot increases.
- Evaluation criteria. The evaluation criteria depends heavily on the specific subtasks of searching problem. In situations where the target is perceived by robots with uncertainty, the tracking accuracy is often used as evaluation metric. On the other hand, if robots can clearly recognize the target, previous works tend to measure consumed resources before achieving certain criteria. [13] defines first passage time, which is the average time every robot takes to pass by the target. Besides, the efficiency of information sharing is also assessed through the defined convergence time. It is the time taken for all robots getting direct or indirect access to the target. In addition, by drawing lesson from works on swarm robotics aggregation, the time taken for leading the robots to converge near the target might be optional criteria, and the convergence can be judged by average robots distance to the target. Apart from time, the total travel distance of all robots can be recorded as a measurement of energy consumption for comparing searching algorithms.

2.3 Algorithms for Swarm Robotic Systems

The advantages of multiple robot systems (MRSs) over single-robot counterparts in searching applications is shown by the parallelism in space and time. Swarm robotic systems (SRSs), being part of MRSs specify the study of collective behaviour emerging from local interaction among robots and between the robots and the environment [62]. [62] also distinguishes SRSs from other MRSs by sets of criteria, e.g. the study should targeted at scalability, relative incapability of individual robot and the robots sensing (communicating) should have limited range. Because SRSs do not rely on centralized control architecture, it scales well with the quantity of robots. When individual robot fails, other robots will try to make up the loss by proceeding the work of failed one [17]. In addition, the stochasticity maintained by individuals makes contribution to self-adaptiveness to changes in environment. These merits of SRSs fit perfectly to searching tasks, especially when the searching efficiency is highly demanded and robots operating conditions are usually severe.

Referring [49], algorithms for searching tasks using SRSs is classified as swarm intelligence (SI) based algorithms and other algorithms.

Searching Algorithms Based on Swarm Intelligence The analogy between robotic searching problems and the problem of optimization is found by [49, 11, 41]. To be more exact, how the particles/agents behave in SI algorithms for solving optimization problems resemble the way individual robots working in SRSs. As a result, lesson could be drawn from SI algorithms for designing SRSs.

- Particle Swarm Optimization. The original algorithm of particle swarm optimization (PSO) was firstly proposed by Kennedy and Eberhart in 1995 [25]. The algorithm is designed for solving optimization problems with a bunch of particles, which solidarity holds a candidate solution. The updating rule of particles is known as mimicking the social behaviour in bird flocks or fish schools. A particle synthesizes its inertial velocity and two vectors, one pointing to its best position and the other pointing to neighbours best position. There are two constants acting as weights balancing the influence from best solution from memory and best solution from other particles. And in [51], weight is also added to inertia velocity, in order to trade-off global search and local search.
  The early work of [45] implements PSO algorithm on swarm robotic platform. However, only the robots communication range is limited while the robots still need to acquire global coordination. In their later work, the global coordination is replaced by robots short memory of last position. In order to lead the robots to aggregate at the target position, [45] also proposes the robot sensing very strong signal of the target should stay as a static beacon for the subsequent robots. With regard to the weight augmented to inertia velocity, they suggest it can be designed to be adaptive, which decreases when robots approaching the target.
  Huge amount of literatures can be found on variants of PSO inspired SRSs searching algorithms. We will not list all of them, but see what the state-of-art variants doing by a review paper [11]. The work in [44] proposes Extended Particle Swarm Optimization for dealing with real world constraints. Robots use the Braitenberg obstacle avoidance algorithm as the position updating approach if they meet obstacles. The Physically-embedded Particle Swarm Optimization in work [20] considers the angular acceleration of robots can not be infinite. So robots only can turn a instantaneous angular. The Robotic Darwin Particle Swarm Optimization (RDPSO) algorithm firstly developed by Couceiro, etc. extends the original algorithm to support multiple dynamic swarms [10]. In RDPSO, if solution of the sub-swarm has not been updated for a determined time, the worst performing robot in this sub-swarm will be excluded.
- Bees Algorithm. As being described before, searching behaviour happens when animal forage. Bees algorithm (BA) was firstly designed by Pham et al. mimicking bees searching for food [18]. The scouts are randomly initialized in the searching area, and flower patches are defined in the vicinity. Foraging

bees are recruited to flower patches according to their quality and execute local search. If no improvement has been found by foraging bees in a flower patch, the size of the flower patch shrinks. And if this happens consecutively for certain iterations, this flower patch will be abandoned and new scout is spawned. Work in [22] introduces the Distributed Bees Algorithm (DBA). The original global recruiting process is replaced by a probabilistic based distributed counterpart. The probability recruiting a foraging robot to a flower patch is based on utility of the patch, which is positively relevant to the quality estimated by the scout and negatively relevant to the distance between the scout and foraging robot.

- Ant Colony Optimization. In [15], the Ant Colony Optimization (ACO) algorithm is proposed. The ACO algorithm shows how a consensus on shortest path leading from ant nest to a food source is reached. Ants leave pheromone on their trajectories and the pheromone evaporates constantly. Ants will be more likely to choose the direction has higher density of pheromone to move when they plan their path.
  The application of ACO onto robotic platforms faces difficulty in implementing pheromone in the real world. Although physical marks, e.g., alcohol, heat, odor or RFID tags could be substitutions for pheromone, work in [21] proposes a neater way through inter-robot communication. Some of the swarm will be assigned to be static beacons for depositing and transmitting artificial pheromone. However, in the work of [61] pheromone is carried on every robot. Robots searching is splitted into local traversal search and global search for promoting efficiency. The global search has modes of random search and probabilistic search. If the best solution in communication range is still lower than determined threshold or the worst solution reaches a really low value, random search is activated. On the contrary, the probability of moving toward another agent depends on the quality of position and the distance.
- Bacterial Foraging Optimization and Biased Random Walk. The chemotaxis behaviour of bacteria inspires the Bacterial Foraging Optimization (BFO) algorithm [42]. The locomotion of bacteria is modelled by combination of swimming (moving in fixed direction) and tumbling (spinning a random angle). In the original algorithm, the proportion of bacteria with bad solution will be eliminated and reproduced. This strategy is infeasible in robotic implementation. And the searching efficiency hardly can match other algorithms, because no messages between agents are transmitted. However, [53] adopts BFO for mapping the distribution of chemical in an unstructured scene.
  The Biased Random Walk (BRW) algorithm is also inspired by the chemotaxis behaviour of bacterium like BFO [12], including the movement model. The difference is that in BRW if the improvement of quality has been detected (which requires a very short memory), the swimming length is increased and the probability of tumbling is decreased. The work in [13] presents the analysis of different random walk modes. In the common scenario where boundaries exist, correlated random walks (CRWs) outperforms

Levy walks. And this can be explained by the influence from collisions with walls.

- Glowworm Swarm Optimization. Agents in Glowworm Swarm Optimization (GSO) algorithm hold a luminescence quantity known as luciferin [29]. Like pheromone in the ACO algorithm, the quantity of luciferin decays by time, but is enhanced by the fitness value of agents current location. Agents continually perceive their neighbours brightness inside limited range. During the motion planning stage, the position of brighter neighbourhood will have a higher probability being targeted by other agents. This decision making stage also happens in a delimited range named as local decision range, which is inside the brightness perception range. The original local decision range updating methods in [29, 28] both extend the range when number of neighbour decreases and vice versa. But the later one is described to be smoother by defining an explicit threshold. The work in [24] also explores the maximum relative speed between the mobile target and the pursuing agents for accomplishing successful tracking with GSO algorithm.
- Firefly Algorithm. The Firefly Algorithm (FA) deploys the same medium for inter-robot communication as GSO, the light emitted by agents [60]. However, the attractiveness of neighbour is modelled by combining distance besides the brightness. In addition, the movement of agents is updated by the average of the delimited neighbours attractiveness and randomness instead of probably following one neighbour.
  The coefficients for brightness absorption and randomness are designed to be adaptive in work [32]. Random motion is expected to dominate the path planning at the start of searching for spreading agents to all the possible position. But at the later period of searching, the demanding of a quick convergence calls for neighbourhood attraction to be dominating. The work in [57] proposes the modified FA, in which randomly picked fireflies are replaced by new generated counterparts around those nearer to target.
- Swarm Environment Based Aggregation Methods. Aggregation is a very common task in swarm robotics research, requiring agents to gather in the working area. In [6], a subcategory of aggregation which lead agents to some preferential regions is reviewed. Insects, e.g. bees and cockroaches could aggregate to place more suitable for their living habit in terms of temperature, brightness or humidity.
  The BEECLUST algorithm in [26] is inspired by the aggregation of young honeybees at warm zones in their nests. A honeybee executes totally random walk until it meets another one. Then this honeybee stops for the duration of time decided by local temperature of the collisional site, warmer site causes longer stop time and vice versa. In the experiment setups of robotic research, the temperature distribution is replace by the luminance of a light source. Later work [3] also adapts the moving velocity of robots in accordance with the local luminance to improve the aggregation efficiency.
  The aggregation behaviour of cockroaches inspires the work in [2] analogous to BEECLUST. However, the duration of stop also depends on the quantity of neighbours. In this way, ideal number of cockroaches are gathered in

certain spots considering the size of that area. This could be an indication for designing task allocation method in multiple targets scenario. The work in [48] shares some similarity with the variant of ACO in [61]. Each agent carries a scalar in correspondence with quality of its current position. Only when the scalar is better than a designed threshold, agent tries to approach the local best neighbour.

Searching Algorithms Based on Other Methods The work in [56] proposes a distributed Kalman filter (DKF) method for updating the estimated position of a single moving target. Robots combine the tracking of target and an artificial potential function to implement a flocking controller. The potential field based methods are also presented in work [40, 39]. The interactions between robots are modelled by repulsive forces and the navigation toward target is driven by the attractive forces. Another work in [31] takes morphology related approach to track multiple moving targets. The robots can not directly sense targets are recruited by those robots who are following targets to maintain equilateral triangles.

In conclusion, most of the reviewed algorithms rely on the timely quality evaluation of robots' position. However, this is only available in those odour localization scenarios, sometime even require precise sensors to effectively distinguish trivial variation. For the more general application, swarm robotic systems needs a concise mechanism for recording individual robot's history searching result and for communication. Therefore, the main purpose of this paper is to form a novel data recording method, and to testify its performance in a searching task.

3 Semantic Mapping

This section details a proposed semantic mapping method. An abstraction is made of the numerical readings via the perception layer, to knowledge (see Fig. 1). The X-Y coordinates and angles can be abstracted into words, e.g. near/far to describe the relative distance in the view of robots. Unique labels are provided to landmarks rather than storing exact position. These labels are abstracted semantics, such as, *far* from target or *near* to target. This abstraction does not rely on exact odometry, and therefore is potentially less prone to errors. In our model, we anticipate the use and sharing of experience. This represents how the selection and evaluation of behaviours are undertaken, depending on the current knowledge of the individual robot. A reflection process, is provided, i.e. how a robot learns experience through perceiving the world, represented as a dotted line in Fig. 1. Currently the behaviour choosing strategy is fixed.

4 Implementation

This section details the design of the proposed system, composing of a basic controller switching searching behaviours, the rules for robots updating semantic descriptions in their knowledge and parameters regulating the aggregation/dispersion. In addition, the PSO inspired method is presented detailing

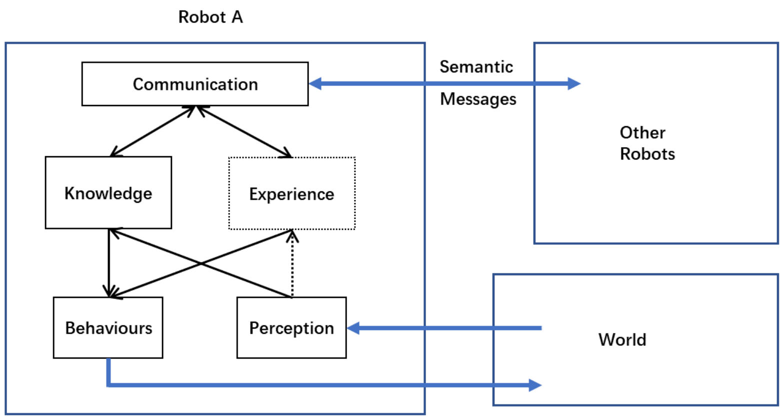


Fig. 1. The designed internal hierarchy of robot message flow. Sensor readings from perception layer are abstracted as knowledge to communicate with other robots in the swarm. Robot uses knowledge and experience to choose behaviours.

how robots generate heading direction. Finally, we present a belief mechanism to allow outdated knowledge to be abandoned.

4.1 Basic Robot Controller

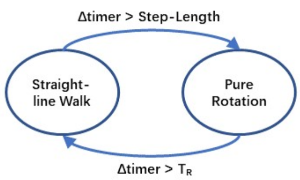


Fig. 2. The finite state machine represents robot searching behaviour. T_R is the fixed rotation time. Step-Length varies according to robot's knowledge. See texts for more details.

The default behaviour of the controller is based on a finite state machine implementing a random walk (see Fig 2). The controller switches from straight-line walk to pure rotation under the guideline of a timer, called as step-length.

When the robot is in straight-line walk state, obstacle avoidance is achieved using proximity sensors. The semantic description regarding the robot's current distance to target (far/near) is used to determine the step-length of straight-line walk. When the robot is *far* from target, a longer step-length (8 seconds [1]) is selected to get a quick escape from current region. However, *near* will lead to shorter step-length (2 seconds [1]) for the purpose of detailed search. When the robot is unclear about the distance description (robot is at the starting period of searching or robot "forgets" the former distance description, which is discussed in Section 4.5), a normal step-length (4 seconds [1]) is set. In this way, the balancing between exploitation and exploration is achieved through the variation of step-length.

The time durations for pure rotation are set to be equal for all robots. If a robot gets no guidance from neighbour robots and landmarks, e.g. each robot in the experiment starting period, a random angle ranging from -180°to 180°is generated and used for turning. The more complicated situation, considering valid neighbour robots or landmarks in range, is discussed in Sections 4.3 and 4.4.

4.2 Semantic Description Updating Rules

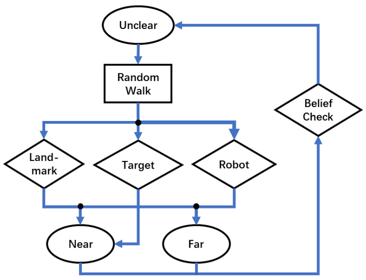


Fig. 3. The updating strategy of the semantic description on robot's distance to target. Direct contact with the target indicates *near*. Otherwise, relative description is updated by messages from neighbours or landmarks. This semantic description will be abandoned by the belief mechanism, described in Section 4.5.

At the start of searching, the semantic description on landmarks and robot distance to target are initialized to be unclear. This means the step-length of each robot is set to normal length. While the search is being undertaken, each robot will encounter different landmarks on their trajectory. Landmark positions, based on the local coordinate frame of each robot, are recorded in the perception layer. The coordinate of landmarks is calculated in accordance with odometry measures, implemented by a dead-reckoning method discussed in [34]. The robot

[1] These values are chosen empirically and have not been optimized for this study.

executes a state machine controller, whose default behaviour is a random walk (shown in Fig. 2).

The semantic description arises at the first time a robot finds the target. This robot takes target position in its local coordinate frame and reviews all the landmarks in its perception layer. An update to the semantic labels on landmarks is made by comparing the relative distance between landmarks and the target with a pre-defined threshold.

Semantic descriptions regarding a robot's distance to target and landmarks are updated by inter-robot communication. Descriptions of landmarks being shared directly are used if one of the robots does not hold the relevant landmark information. Descriptions of a robot's distance to target is updated based on a hop-count based mechanism, see Section 4.3.

The final scenario where semantic description of robot's distance to target being updated is when a robot detects landmarks that exist in their knowledge layer. As discussed previously, landmarks are either injected by self-reviewing process or transferred from other robots through local communication. When robot meets landmark with the label of *near*, it switches the description regarding its distance to *near*, otherwise switches to *far*.

4.3 The Hop-Count Based Aggregation/Dispersion

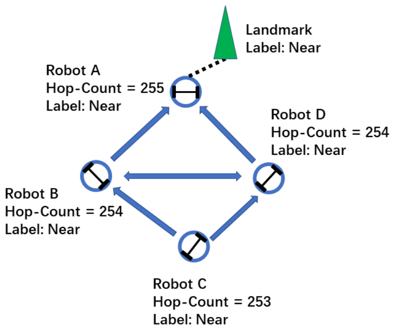
Robots update the semantic description according to target, landmarks or neighbour robots. Messages should be augmented with an order number to symbolize their significance as they may come from different sources. The order number called hop-count, inspired by the work of [48] is passed with semantic description of robot's distance to target.

There are three circumstances when a robot would get the description with the highest hop-count: 1) the robot just leaves the target, 2) the robot detects a known landmark in their knowledge base, 3) the robot gets a direct call for aggregation from a robot inside target area. Alternatively, when a semantic description on a robot's distance is passed from one robot to another, the hop-count is decreased and transferred to the receiver-robot (see Fig 4). It is worth noting that the robot inside target area holds the higher priority call for aggregation than any other circumstances, and neglects all messages from neighbour robots and landmarks.

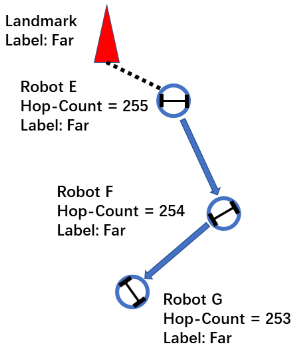
The significance of the message is reflected by aggregating or dispersing, taking into account the hop-count. Robots only take their neighbours with higher (or equal) hop-count as efficient aggregating or dispersing targets and the message flow only sources from high hop-count robots to the low ones.

4.4 PSO Inspired Searching Behaviour

A searching task requires robot behaviour that combines both searching and aggregation (or dispersion). By dividing these behaviours, an individual robot



(a)



(b)

Fig. 4. The hop-count is passed with semantic description to form the aggregation (a) or dispersion (b) following the arrow direction. The robots with lower hop-count are attracted (a) or repulsed (b) by robots with higher (or equal) hop-count.

controller combines self-inertial heading direction and the guiding directions in the pure rotation state.

$$\theta_{t+1} = \theta_t + Norm[f(t) \times Ran_1 \times \theta_{landmark} + g(t) \times (Ran_2 + 1) \times \theta_{neighbour}] \quad (1)$$

$$f(t) = \begin{cases} 1 \text{ for } Near\, landmark\, in\, range. \\ -1 \text{ for } Far\, landmark\, in\, range. \\ 0 \text{ for } No\, landmark. \end{cases} \quad (2)$$

$$g(t) = \begin{cases} 1 \text{ for } Neighbour\, labelled\, Near\, in\, range. \\ -1 \text{ for } Neighbour\, labelled\, Far\, in\, range. \\ 0 \text{ for } No\, neighbour. \end{cases} \quad (3)$$

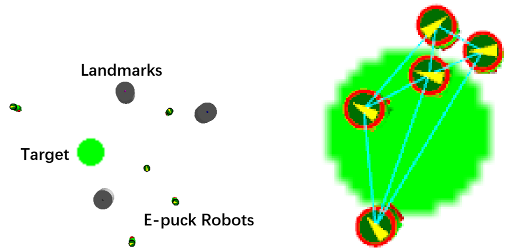
Equation 1 is the heading direction updating rule inspired by the PSO algorithm [58]. Equations 2 and 3 are calculated by whether valid neighbour landmarks and valid robots are sensed by the robot. The random coefficients Ran_1 leads robots to the direction of landmarks labelled *near* and to the reverse direction of landmarks labelled *far*. Ran_2 works in a similar manner, driving robots towards the centre of valid neighbour robots labelled *near* and away from the centre of robots labelled *far*. The final coefficient $(Ran_2 + 1)$ for neighbour robots are designed to be greater than the coefficient (Ran_1) applied to landmarks to avoid robots being stuck with the landmarks.

4.5 Belief Mechanism

The time when a robot assumes a semantic description on its distance to target is outlined in Section 4.2. However, this description will become invalid as the robot moving around. Odometry is used to record the distance between a robot's current position and the position when it picks up the relevant semantic description. The distance is compared with a threshold as belief determining whether the robot should abandon the semantic description of its distance to target. When the description is abandoned, the relevant hop-count is cleared.

5 Experimental Methods

In this paper, ARGoS [43] is used for simulation purposes. It is a physics-based simulator suitable for simulating a large number of swarm robots. The e-puck robot [36] deployed in this work is a differential wheeled mobile robotic platform with various on-board sensors. Fig. 5(a) shows the experiment scenario of the SaR task in ARGoS. A single static target for searching is modelled by a circle of green on the floor, while the other part of the arena floor is white. The target can be found by employing the e-puck's ground sensor reading, which returns a value between 0 and 1 reflecting the grey-scale value of the floor.



(a) (b)

Fig. 5. (a) In the experiment scenario, the target is static. E-puck robots and landmarks are randomly distributed. (b) The designed terminal condition of the experiment.

Landmarks are designed as unique and universal components in the arena, as indicated in Section 3. Unique, in this context, means different landmarks can be distinguished by the robot sensor and universality means the same landmark transfers identical meaning to all the robots.

In ARGoS, multi-colour LEDs can be attached to entities like cylinders. In our case, cylinders attached with colour LED are designed to work as landmarks. The colour LEDs can be perceived by the coloured blob omni-directional camera. For the purposes of this work, the source code of ARGoS describing e-puck [36] entity is modified to support the module of coloured blob omni-directional camera. The camera obtains the detected LED's colour and relative position, when it is in range.

As stated before, a single robot finding the target might not be sufficient for recruiting other robots in the swarm to the target as for lacking global communication. However, if enough numbers of robots aggregate in the region of target, mechanisms such as the chain based path formation [38] can be devised to recruit other robots to the target efficiently. Therefore, we re-define the terminal condition of our searching task as at least 5 robots forming an aggregation with direct connection to the target (See Fig. 5(b)). The video [59] presents how one successful search is accomplished in ARGoS.

For a baseline comparison for the efficiency of the proposed semantic mapping method, we designed a searching behaviour that still takes all the mechanism except landmark recording and recognition. As robots mostly rely on the random-walk mechanism to search for the target, we call the method as random-walk method in Section 6.

Robot odometry is implemented by accumulating the reading from the differential steering sensor every time step. To investigate the robustness against odometry error, we designed a comparison experiment that the systematic odometry error is injected into robots. The systematic odometry error is modelled by a method taken from [7]. Uniform random coefficients in a determined range are generated to modify the diameter of the robot's both wheels. The wheel-base length of robot is also adapted by the same method.

In order to obtain robust analysis, we use statistical analysis Spartan [1], to determine the number of trials required for an experiment to remove aleatory uncertainty. Analysis shows that 200 repeats of simulation are sufficient, therefore the figures showing in Section 6 are all based on 200 trials of the relevant parameter set.

6 Results and Discussion

To evaluate the searching efficiency, we measure the searching time consumed by robots between the initial deployment of the swarm and the terminal of searching detailed in Section 5.

The rest of this section presents results from comparison experiments between the proposed semantic method and rand-walk, the Latin-hypercube analysis, testifying scalability, exploring key factors about landmarks and comparison experiments after injecting systematic odometry error.

6.1 Comparison of Searching Efficiency

The searching efficiency of different swarm sizes are compared with the relevant random-walk method described in Section 5 in a fixed size arena. We expect that the proposed method would be more efficient than the random-walk method, thus we propose a null hypothesis:

Hypothesis 1 (Null Hypothesis): *No significant difference exists between the searching time of proposed method and the basel-line random-walk method.*

Fig. 6 shows box-plots of the results from semantic landmark label method and random-walk method deploying different swarm sizes. The Wilcoxon-Mann-Whitney rank sum test [4] is executed to determine whether the search time of the proposed method and random-walk based method has the same distribution. The p-value is shown in Table 1. When the swarm sizes are 10, 20 and 30, the p-value allows us to reject Hypothesis 1. However, for a swarm size of 40, the p-value indicates that the hypothesis be accepted - there is no difference. These results mean that when in the limited size arena and when the swarm size is below certain value, the searching efficiency of proposed semantic mapping method is significant better than random-walk. However, this is not the case when the swarm size 40. This is most likely due to the fact that the experiment arena is highly populated with large number of robots being deployed. The high density of robots leads to a rapid accomplishment of the searching task after the beginning of experiments.

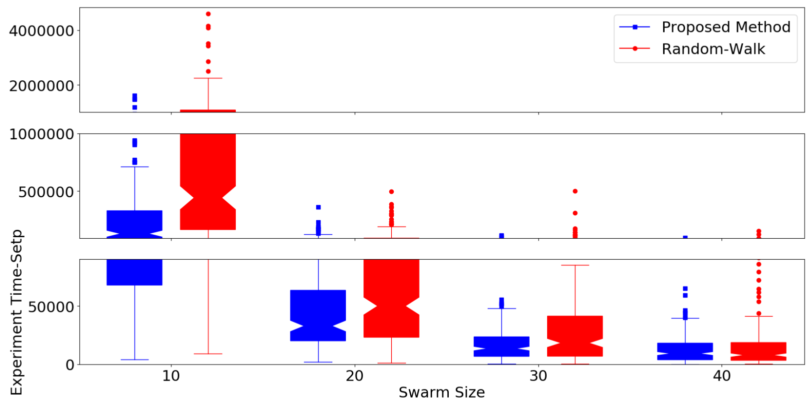
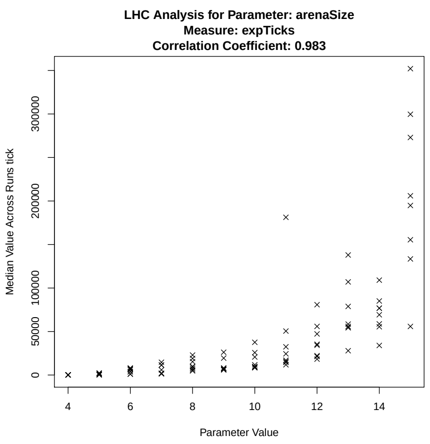


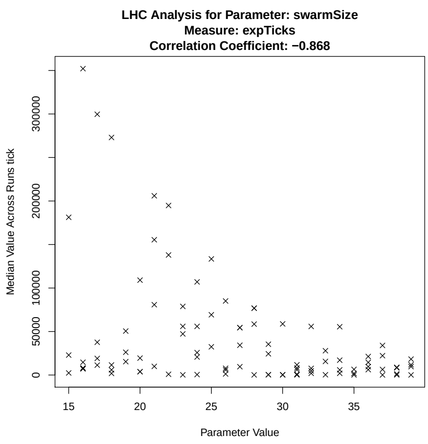
Fig. 6. The box-plot of search time comparing proposed method and random-walk deploying different swarm sizes. 200 runs of experiment is shown in each box.

Table 1. The Wilcoxon-Mann-Whitney rank sum test p-value between proposed semantic method and random-walk method.

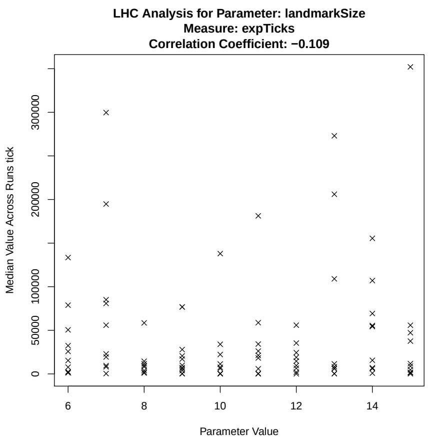
	10 Robots	20 Robots	30 Robots	40 Robots
Proposed method v.s. Random-walk	1.521e-15	0.001659	0.000692	0.4095



(a)



(b)



(c)

Fig. 7. The median value of experiment ticks (time-step) when the search task is finished. This figure shows the correlation between the search time and 3 relevant parameters: a) the arena size, b) the size of the swarm, c) the quantity of landmarks. The x-axis represents the value of each parameter. The y-axis represents searching time.

6.2 Latin-hypercube Analysis of Key Parameters

The arena size, swarm size and number of landmarks in the arena influencing the searching time are used to evaluate efficiency. However, the time measured for one parameter may be highly dependent on the value of another, in other words correlations exist between parameters. To address this issue, we employ the latin-hypercube analysis technique in Spartan [1] which is able to perform a global sensitivity analysis of simulation response on single parameter perturbation. Latin-hypercube analysis is the method used to generate the sample set in which all parameters are perturbed simultaneously. The median of response from 200 repeat of experiments following each parameter combination in the sample set is recorded as a point in Fig. 7. The generated correlation coefficient ranges from -1 to 1, with the quantity indicating how strong the searching time is related with the parameter and sign indicating which direction the time is influenced. As shown in Fig. 7, a strong positive co-relationship exists between searching time and the arena size, and a strong negative co-relationship exists between searching time and the swarm size. Weak co-relationship between searching time and quantity of landmarks is shown.

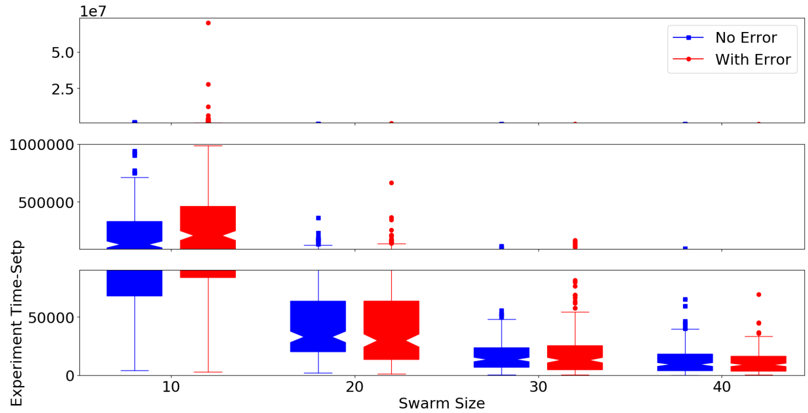


Fig. 8. The box-plot of searching time comparing before and after injecting systematic odometry error.

Table 2. The Wilcoxon-Mann-Whitney rank sum test p-value between experiments without odometry error and with odometry error

	10 Robots	20 Robots	30 Robots	40 Robots
No error v.s. Odometry error	0.006413	0.2179	0.9033	0.2449

6.3 Testifying System Scalability

As shown in Section 6.1 and Section 6.2, the searching time is perturbed when either changing the arena size or changing the swarm size. However, scalability is still a property we expect the system to have. In the context of this paper, the scalability is defined by varying the swarm size when keep the density of robots fixed. In other words, the robustness of searching time against the varying swarm size with a proportional arena size should be assessed.

To verify the scalability, we employ the parameter robustness analysis technique in Spartan [1]. The robustness is analysed by changing each parameter individually, using a one at a time approach[46]. The various simulation responses of altering a particular parameter are compared with the baseline condition, employing the Vargha-Delaney A-Test[54], a non-parametric effect magnitude test which provides a statistical measure of the difference between two distributions.

Table 3. Setup for testifying system scalability

Swarm Size	10	20	40	80
Arena Size (m^2)	7×7	10×10	14×14	20×20

The swarm size and experiment arena are configured according to the parameters generated by Spartan robustness analysis technique (see Table 3). The baseline of swarm size is chosen as 20 robots. As shown in Fig. 9, non-significant difference about the searching time is presented when changing the swarm size by 10, 20, 40 and 80 with a fixed density. Thus, the consistence of searching time is shown regardless of the number of robots.

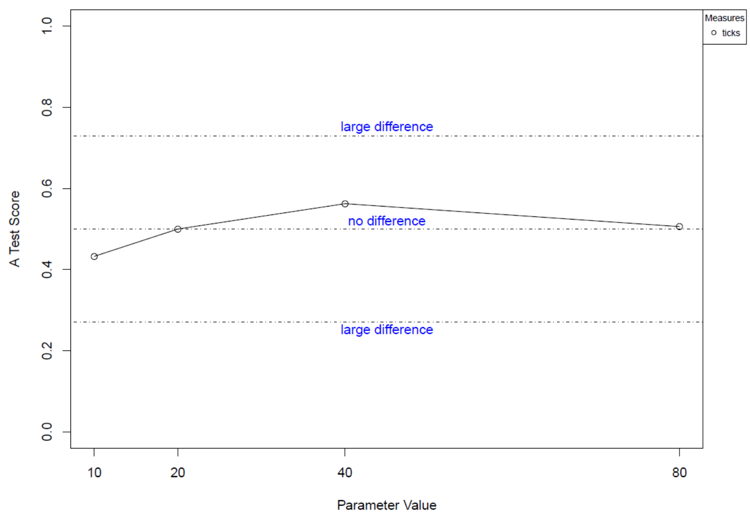


Fig. 9. The A-Test scores generated when the parameter swarm size is perturbed, keeping the robot density fixed.

6.4 Verifying The Factors of Landmarks

In order to optimize the searching efficiency, we will explore those factors influencing the functioning of landmarks in this section. The factors, in terms of robots can be classified into inner-factors and out-factors in this semantic mapping system. The inner-factors are parameters threshold the the Euclidean distance between target and landmarks to categorizes them into *near* or *far* landmarks. And the out-factors are consist of how the landmarks are distributed in the arena.

The Threshold Parameters In the proposed semantic mapping method, there exists two threshold parameter working with the semantic label. One of the threshold is for labelling landmarks as described before, the other is for belief-based abandoning the semantic description about the distance to target, see Fig. 3. Before optimizing the threshold parameter, we want firstly testify the parameter robustness against various thresholds. So the parameter robustness analysis technique employed in Section 6.3 is also used here.

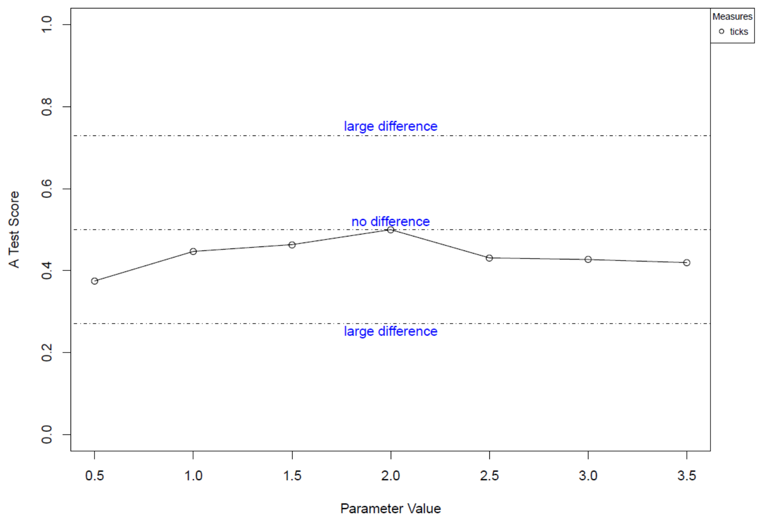


Fig. 10. The A-Test scores generated when only the parameter landmark threshold is perturbed.

The value of landmark threshold and belief threshold are altered separately in two sets of experiments. As shown in Fig. 10, when the landmark threshold changes from 0.5 to 3.5 metre, non-significant difference is shown on the searching time comparing with baseline condition of 2.0 metre. This could be an indication that the landmarks threshold value can be selected empirically. And in Fig. 11, when the belief threshold value is altered from 0.5 to 3.5 metre, only threshold

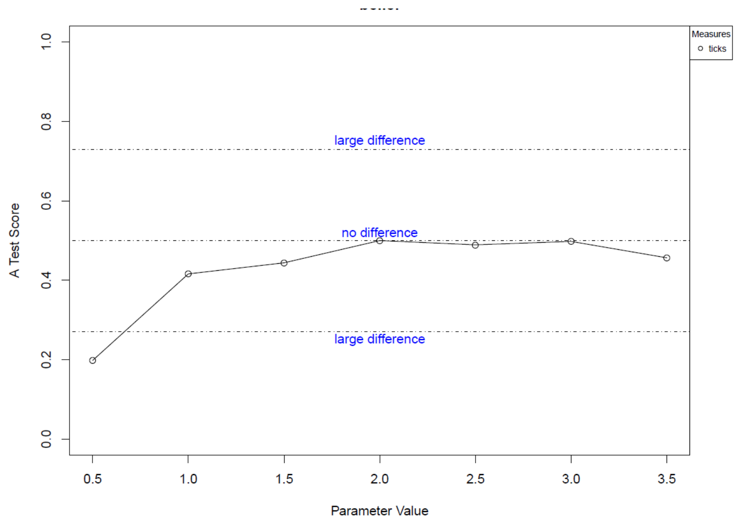


Fig. 11. The A-Test scores generated when only the parameter belief threshold is perturbed.

0.5 causes significant different searching time comparing with baseline condition of 2.0 metre. This can be explained as that 0.5 metre is a too limited distance for robots timely aggregate with their neighbours. The value for belief threshold can be inferred by a proportion to the arena size.

The Distribution of Landmarks As the out-factor for robots, the distribution pattern of landmarks might influence the searching time. So here we propose another null hypothesis:

Hypothesis 2 (Null Hypothesis): *There is no difference in searching time when the landmarks are laid with different distribution patterns.*

To testify this hypothesis, the searching task is executed in four extreme conditions with 20 robots, see Fig. 12. The p-value of Wilcoxon-Mann-Whitney rank sum test [4] on searching time between condition (b) and condition (d) is 0.0213322867489. By this, null hypothesis 2 can be rejected. Therefore, the distribution of landmarks influences searching efficiency.

6.5 Testifying Robustness against Errors

As the proposed method does not rely on precise odometry, it should show robustness against odometry error. We propose the further null hypothesis:

Hypothesis 3 (Null Hypothesis): *There is no difference in searching time between when limited errors are introduced to the odometry reading, when compared to odometry readings without errors.*

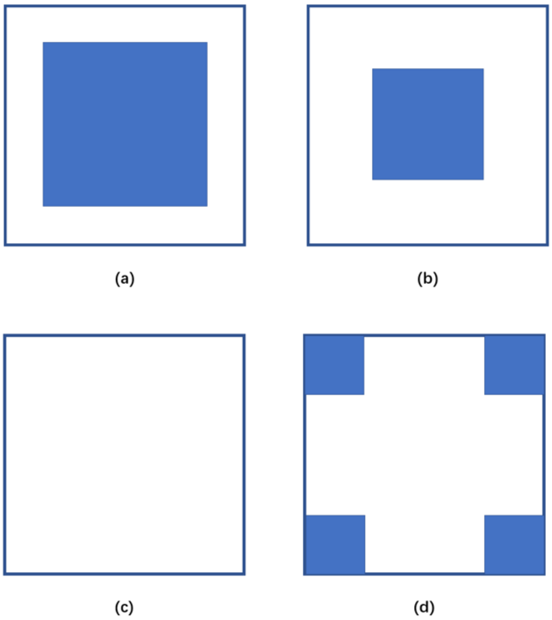


Fig. 12. Four extreme conditions of landmarks distribution, landmarks are uniformly distributed in the solid squares: a) the square centred by the arena centre and its sizes $4 \times 4m$; b) the square centred by the arena centre and it sizes $2 \times 2m$; c) no places designed to lay landmarks; d) those squares locate at corner of the arena and the size is $1 \times 1m$.

The result comparing before and after injecting systematic odometry error to robots in the searching tasks are shown in Fig. 8. The p-value of Wilcoxon-Mann-Whitney rank sum test [4] on searching time before and after injecting systematic odometry error is shown in Table 2. We can see that when the swarm sizes are 20, 30 and 40 Hypothesis 3 is accepted. However, the p-value of swarm size 10 rejects the null hypothesis. This would indicates that a small swarm size is not robust enough against odometry error.

6.6 Tentative toward Experience Sharing

As presented in 3, the interpretation of experience might just be robots' current semantic description on their distance to target. And we find a condition, when the *NEAR* robots meet the *FAR* robots. The decision about whom are correct should be made.

In the original design, the *NEAR* robots always dominate this decision making process. As the comparison experiment, we flip it to make *FAR* robots dominating. And the experiments are executed with different swarm sizes. The null hypothesis is proposed as:

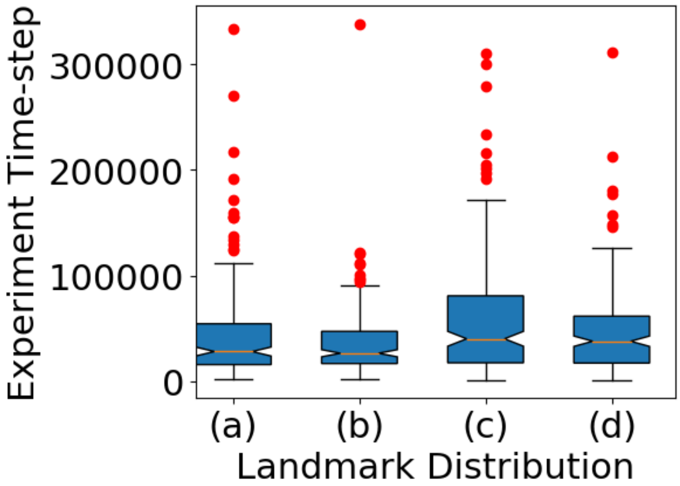


Fig. 13. The boxplot of searching time in experiments of different landmark distributions.

Hypothesis 4 (Null Hypothesis): *There is no difference in searching time between when NEAR robots dominate and when FAR robots dominate.*

Table 4. The Wilcoxon-Mann-Whitney rank sum test p-value between experiments when *NEAR* robots dominate and when *FAR* robots dominate.

	10 Robots	20 Robots	30 Robots	40 Robots
Near v.s. Far	0.000573	0.014390	0.139125	0.654435

The searching time with different swarm sizes is shown in Fig. 14. And the p-values of Wilcoxon-Mann-Whitney rank sum test [4] on searching time between when *NEAR* robots dominate and when *FAR* robots dominate are presented in Table 4. According to the p-value and the boxplot, the original *NEAR* robots dominating method is significantly better than the flipper *FAR* robots dominating method when the swarm sizes are 10 and 20. When the swarm sizes are 30 and 40, they are non-significantly different. The similarity in experiments with swarm sizes 30 and 40 can also be explained by the high density of robots as in Section 6.1.

To explain the difference shown when the swarm sizes are 10 and 20, the distance between *FAR* robots and target is recorded when *FAR* robots meet *NEAR* robots,. The landmark threshold value is set as 2.0 metre. As shown in Table 5, the situation causing this kind of decision making mostly happens in

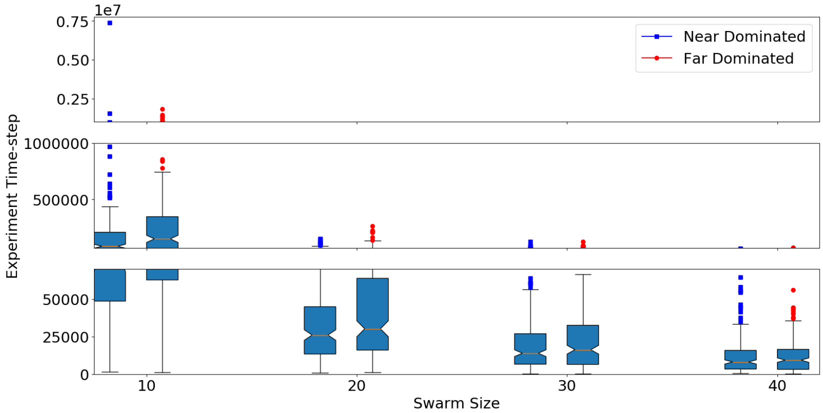


Fig. 14. The boxplot of searching time comparing between when *NEAR* robots dominate and when *FAR* robots dominate.

Table 5. The *FAR* robots' distance to target when meet with *NEAR* robots.

id	4	8	12	13	0	14	19	17	10	6	19	7	1	16	18
distance	2.21	2.16	2.41	1.55	2.48	2.61	2.49	2.18	3.02	2.57	2.51	2.51	1.82	1.48	1.95

vicinity of the landmark threshold. So, the *NEAR* robots have are more likely to hold the correct experience in this decision making process.

7 Conclusion

In this work, the hypothesis that swarm robotic systems can efficiently accomplish searching task through interaction of abstracted semantic knowledge between individual robots has been explored. High searching efficiency can be achieved by using the semantic mapping method. The influence of varying several key parameters are explored. We have also shown that the method is robust against odometry error caused by sensor noises.

For future work, we plan to implement the system in physical robots. In addition, the target used for searching will be extended to a dynamic target. We expect that the semantic knowledge can also be passed between robots with different physical modules. Therefore we plan to apply our approach in a heterogeneous swarm robotic system. Finally, for work in this paper, knowledge is built under supervised off-line rules. However, new knowledge can be learned by robots themselves through certain mechanisms, e.g. a feedback reward process in robot foraging task and richer information can be introduced into the semantic map. This will form an interesting avenue to explore.

From the perspective of practical application, human operators are usually engaged in manipulating the robots. It is possible that the human experiences that are useful for the searching tasks can be transferred to the robotic system through the semantic way. On the other hand, the searching result can be presented to operators in human-comprehensible format. Lastly, the battery carried by individual robot stores limited energy which is constantly consumed during the searching process. Hence, factors related with energy-level should be designed in robot knowledge model to determine robot searching behaviours.

References

1. Alden, K., Read, M., Andrews, P.S., Timmis, J., Coles, M.: Applying spartan to Understand Parameter Uncertainty in Simulations. The R Journal 6(2), 63–80 (2014)
2. Ame, J.m., Ame, J.m., Rivault, C., Deneubourg, J.l., De Bruxelles, U.L.: Cockroach aggregation based on strain odour recognition, Animal Behaviour. ANIMAL BEHAVIOUR 68, 2004 (2004),
3. Arvin, F., Samsudin, K., Ramli, A.R., Bekravi, M.: Imitation of Honeybee Aggregation with Collective Behavior of Swarm Robots. International Journal of Computational Intelligence Systems 4(4), 739–748 (jun 2011),
4. Bauer, D.F.: Constructing Confidence Rank Sets Using Statistics. Journal of the American Statistical Association 67(339), 687–690 (sep 1972)
5. Bayat, B., Crasta, N., Crespi, A., Pascoal, A.M., Ijspeert, A.: Environmental monitoring using autonomous vehicles: a survey of recent searching techniques. Current Opinion in Biotechnology 45, 76–84 (jun 2017),
6. Bay\ind\ir, L.: A Review of Swarm Robotics Tasks. Neurocomput. 172(C), 292–321 (2016),
7. Borenstein, J., Liqiang Feng: Measurement and Correction of Systematic Odometry Errors in Mobile Robots. IEEE Transactions on Robotics and Automation 12(6), 869–880 (1996)
8. Brambilla, M., Ferrante, E., Birattari, M., Dorigo, M.: Swarm robotics: A review from the swarm engineering perspective. Swarm Intelligence 7(1), 1–41 (mar 2013)
9. Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., Leonard, J.J.: Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. IEEE Transactions on Robotics 32(6), 1309–1332 (dec 2016)
10. Couceiro, M.S., Rocha, R.P., Ferreira, N.M.F.: A novel multi-robot exploration approach based on Particle Swarm Optimization algorithms. In: 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics. pp. 327–332. IEEE (nov 2011),
11. Couceiro, M.S., Vargas, P.A., Rocha, R.P., Ferreira, N.M.: Benchmark of swarm robotics distributed techniques in a search task. Robotics and Autonomous Systems 62(2), 200–213 (feb 2014)
12. Dhariwal, A., Sukhatme, G., Requicha, A.: Bacterium-inspired robots for environmental monitoring. In: IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004. pp. 1436–1443 Vol.2. IEEE (2004),
13. Dimidov, C., Oriolo, G., Trianni, V.: Random Walks in Swarm Robotics: An Experiment with Kilobots. pp. 185–196. Springer, Cham (2016),
14. Dorigo, M., Birattari, M., Brambilla, M.: Swarm robotics (2014),
15. Dorigo, M., Di Caro, G., Gambardella, L.M.: Ant Algorithms for Discrete Optimization. Artif. Life 5(2), 137–172 (1999),
16. Fricke, G.M., Hecker, J.P., Cannon, J.L., Moses, M.E.: Immune-inspired search strategies for robot swarms. Robotica 34(08), 1791–1810 (aug 2016),
17. Gabriely, Y., Rimon, E.: Spanning-tree based coverage of continuous areas by a mobile robot. In: Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164). vol. 2, pp. 1927–1933. IEEE,
18. Ghanbarzadeh, A., Koç, E., Otri, S., Rahim, S., Zaidi, M.: The Bees Algorithm A Novel Tool for Complex Optimisation Problems. Intelligent Production Machines and Systems pp. 454–459 (jan 2006),

19. Heath, S., Hensby, K., Boden, M., Taufatofua, J., Weigel, J., Wiles, J.: Lingodroids: Investigating grounded color relations using a social robot for children. In: ACM/IEEE International Conference on Human-Robot Interaction. vol. 2016-April, pp. 435–436. IEEE (mar 2016)
20. Hereford, J., Siebold, M.: Multi-robot search using a physically-embedded Particle Swarm Optimization. International Journal of Computational Intelligence Research 4(2) (2008),
21. Hoff, N.R., Sagoff, A., Wood, R.J., Nagpal, R.: Two foraging algorithms for robot swarms using only local communication. In: 2010 IEEE International Conference on Robotics and Biomimetics. pp. 123–130. IEEE (dec 2010),
22. Jevtic, A., Gutierrez, Á., Andina, D., Jamshidi, M.: Distributed Bees Algorithm for Task Allocation in Swarm of Robots. IEEE Systems Journal 6(2), 296–304 (jun 2012),
23. Jung, B.: Cooperative Target Tracking Using Mobile Robots. Ph.D. thesis, Los Angeles, CA, USA (2005)
24. Kaipa, K., Ghose, D.: Chasing Multiple Mobile Signal Sources: A Glowworm Swarm Optimization Approach. (2007)
25. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN'95 - International Conference on Neural Networks. vol. 4, pp. 1942–1948. IEEE,
26. Kernbach, S., Thenius, R., Kernbach, O., Schmickl, T.: Re-embodiment of Honeybee Aggregation Behavior in an Artificial Micro-Robotic System (2009),
27. Kowadlo, G., Russell, R.A.: Robot Odor Localization: A Taxonomy and Survey. The International Journal of Robotics Research 27(8), 869–894 (aug 2008),
28. Krishnanand, K., Amruth, P., Guruprasad, M., Bidargaddi, S., Ghose, D.: Glowworm-inspired robot swarm for simultaneous taxis towards multiple radiation sources. In: Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006. pp. 958–963. IEEE,
29. Krishnanand, K., Ghose, D.: Detection of multiple source locations using a glowworm metaphor with applications to collective robotics. In: Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005. pp. 84–91. IEEE,
30. Kuipers, B., Byun, Y.T.: A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. Robotics and Autonomous Systems 8(1-2), 47–63 (nov 1991)
31. Lee, G., Chong, N.Y., Christensen, H.: Tracking multiple moving targets with swarms of mobile robots. Intelligent Service Robotics 3(2), 61–72 (apr 2010),
32. Liu, C., Gao, Z., Zhao, W.: A New Path Planning Method Based on Firefly Algorithm. In: 2012 Fifth International Joint Conference on Computational Sciences and Optimization. pp. 775–778. IEEE (jun 2012),
33. Liu, Y., Nejat, G.: Robotic urban search and rescue: A survey from the control perspective. Journal of Intelligent and Robotic Systems: Theory and Applications 72(2), 147–165 (nov 2013)
34. Lucas, G.W.: A Tutorial and Elementary Trajectory Model for the Differential Steering System of Robot Wheel Actuators (2014),
35. Milford, M., Wyeth, G., Prasser, D.: RatSLAM: a hippocampal model for simultaneous localization and mapping. In: IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004. pp. 403–408 Vol.1. IEEE (2004)
36. Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klaptocz, A., Zufferey, J.c., Floreano, D., Martinoli, A.: The e-puck , a Robot Designed for Education in Engineering. Robotics 1(1), 59–65 (2006)

37. Nembrini, J., Winfield, A.F.T.: Safety in Numbers: Fault Tolerance in Robot Swarms. Journal of Adaptive Behaviour 1(1), 0–0 (2010)
38. Nouyan, S., Dorigo, M.: Chain Based Path Formation in Swarms of Robots pp. 120–131 (2006)
39. Ota, J., Arai, T., Yoshimura, Y., Miyata, N., Yoshida, E., Kurabayashi, D., Sasaki, J.: Motion planning of multiple mobile robots by a combination of learned visibility graphs and virtual impedance. Advanced Robotics 10(6), 605–620 (jan 1995),
40. Parker, L.E.: Distributed Algorithms for Multi-Robot Observation of Multiple Moving Targets. Autonomous Robots 12(3), 231–255 (2002),
41. Parker, L.E.: Decision Making as Optimization in Multi-robot Teams. pp. 35–49. Springer, Berlin, Heidelberg (2012),
42. Passino, K.: Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Systems Magazine 22(3), 52–67 (jun 2002),
43. Pinciroli, C., Trianni, V., O'Grady, R., Pini, G., Brutschy, A., Brambilla, M., Mathews, N., Ferrante, E., Di Caro, G., Ducatelle, F., Birattari, M., Gambardella, L.M., Dorigo, M.: ARGoS: A modular, parallel, multi-engine simulator for multi-robot systems. Swarm Intelligence 6(4), 271–295 (2012)
44. Pugh, J., Martinoli, A.: Multi-robot Learning with Particle Swarm Optimization. In: Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems. pp. 441–448. AAMAS '06, ACM, New York, NY, USA (2006),
45. Pugh, J., Martinoli, A.: Inspiring and Modeling Multi-Robot Search with Particle Swarm Optimization. In: 2007 IEEE Swarm Intelligence Symposium. pp. 332–339. IEEE (apr 2007),
46. Read, M., Andrews, P.S., Timmis, J., Kumar, V.: Techniques for grounding agent-based simulations in the real domain: a case study in experimental autoimmune encephalomyelitis. Mathematical and Computer Modelling of Dynamical Systems 18(1), 67–86 (feb 2012),
47. Rone, W., Ben-Tzvi, P.: Mapping, localization and motion planning in mobile multi-robotic systems. Robotica 31(1), 1–23 (jan 2013)
48. Schmickl, T., Möslinger, C., Crailsheim, K.: Collective perception in a robot swarm. In: Swarm robotics, pp. 144–157. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
49. Senanayake, M., Senthooran, I., Barca, J.C., Chung, H., Kamruzzaman, J., Murshed, M.: Search and tracking algorithms for swarms of robots: A survey. Robotics and Autonomous Systems 75, 422–434 (jan 2016),
50. Shah, B., Choset, H.: Survey on Urban Search and Rescue Robots. Journal of the Robotics Society of Japan 22(5), 582–586 (jul 2004),
51. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360). pp. 69–73. IEEE,
52. Suarez, J., Murphy, R.: A survey of animal foraging for directed, persistent search by rescue robotics. In: 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics. pp. 314–320. IEEE (nov 2011),
53. Turduev, M., Kirtay, M., Sousa, P., Gazi, V., Marques, L.: Chemical concentration map building through bacterial foraging optimization based search algorithm by mobile robots. In: 2010 IEEE International Conference on Systems, Man and Cybernetics. pp. 3242–3249. IEEE (oct 2010),
54. Vargha, A., Delaney, H.D.: A Critique and Improvement of the CL Common Language Effect Size Statistics of McGraw and Wong. Journal of Educational and Behavioral Statistics Summer 25(2), 101–132 (2000),

Bibliography

[1] Cyberbotics. Gctronic' e-puck. URL https://www.cyberbotics.com/doc/guide/epuck?version=cyberbotics%3AR2019a.

[2] J. Crespo, R. Barber, and O. M. Mozos. Relational model for robotic semantic navigation in indoor environments. *Journal of Intelligent & Robotic Systems*, 86(3):617–639, Jun 2017. doi: 10.1007/s10846-017-0469-x. URL https://doi.org/10.1007/s10846-017-0469-x.

[3] Erol Sahin. Swarm Robotics: From Sources of Inspiration to Domains of Application. pages 10–20. Springer, Berlin, Heidelberg, 2005. doi: 10.1007/978-3-540-30552-1_2.

[4] Zachary Mason. Programming with Stigmergy: Using Swarms for Construction. *Artificial Life VIII*, pages 371–374, 2002.

[5] J.-L Deneubourg, S Aron, S Goss, and J M Pasteels. The Self-Organizing Exploratory Pattern of the Argentine Ant. *Journal of lnsect Behavior*, (2), 1990.

[6] Simon Garnier, Jacques Gautrais, and Guy Theraulaz. The biological principles of swarm intelligence. *Swarm Intell*, 1:3–31, 2007. doi: 10.1007/s11721-007-0004-y.

[7] Guy Theraulaz, Jacques Gautrais, Scott Camazine, and Jean-Louis Deneubourg. The formation of spatial patterns in social insects: from simple behaviours to complex structures. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 361 (1807), 2003.

[8] Guy Theraulaz, Eric Bonabeau, and Jean-Louis Deneubourg. Response threshold reinforcement and division of labour in insect societies.

[9] Wenguo Liu and Alan Ft Winfield. Modelling and Optimisation of Adaptive Foraging in Swarm Robotic Systems. 29(14):1743–1760, 2010. doi: 10.1177/0278364910375139.

[10] Eliseo Ferrante, Ali Emre Turgut, Cristián Huepe, Alessandro Stranieri, Carlo Pinciroli, and Marco Dorigo. Self-organized flocking with a mobile robot swarm: a novel motion control method. doi: 10.1177/1059712312462248.

[11] Liviu Panait and Sean Luke. Cooperative Multi-Agent Learning: The State of the Art. *Autonomous Agents and Multi-Agent Systems*, 11(3):387–434, nov 2005. ISSN 1387-2532. doi: 10.1007/s10458-005-2631-2.

[12] Michael Rubenstein, Christian Ahler, and Radhika Nagpal. Kilobot: A low cost scalable robot system for collective behaviors. In *2012 IEEE International Conference on Robotics and Automation*, pages 3293–3298. IEEE, may 2012. ISBN 978-1-4673-1405-3. doi: 10.1109/ICRA.2012.6224638.

[13] Shervin Nouyan and Marco Dorigo. Chain Based Path Formation in Swarms of Robots.

[14] Roderich Gro, Michael Bonani, Francesco Mondada, and Marco Dorigo. Autonomous Self-Assembly in Swarm-Bots. *IEEE Transactions on Robotics*, 22 (6):1115–1130, dec 2006. ISSN 1552-3098. doi: 10.1109/TRO.2006.882919.

[15] Wu bin and Shi Zhongzhi. A clustering algorithm based on swarm intelligence. In *2001 International Conferences on Info-Tech and Info-Net. Proceedings (Cat. No.01EX479)*, volume 3, pages 58–66. IEEE. ISBN 0-7803-7010-4. doi: 10.1109/ICII.2001.983036.

[16] Marco Dorigo, Mauro Birattari, and Thomas Stutzle. Ant colony optimization. *IEEE Computational Intelligence Magazine*, 1(4):28–39, nov 2006. ISSN 1556-603X. doi: 10.1109/MCI.2006.329691.

[17] Vidyasagar Potdar, Atif Sharif, and Elizabeth Chang. Wireless Sensor Networks: A Survey. In *2009 International Conference on Advanced Information Networking and Applications Workshops*, pages 636–641. IEEE, may 2009. ISBN 978-1-4244-3999-7. doi: 10.1109/WAINA.2009.192.

[18] R. Olfati-Saber. Flocking for Multi-Agent Dynamic Systems: Algorithms and Theory. *IEEE Transactions on Automatic Control*, 51(3):401–420, mar 2006. ISSN 0018-9286. doi: 10.1109/TAC.2005.864190.

[19] Scott. Camazine. *Self-organization in biological systems*. Princeton University Press, 2003. ISBN 0691116245.

[20] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun. Collaborative multi-robot exploration. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 1, pages 476–481. IEEE. ISBN 0-7803-5886-4. doi: 10.1109/ROBOT.2000.844100.

[21] Wei Li. Persistent surveillance for a swarm of micro aerial vehicles by flocking algorithm. doi: 10.1177/0954410014529100.

[22] M. Masár. A biologically inspired swarm robot coordination algorithm for exploration and surveillance. In *2013 IEEE 17th International Conference on Intelligent Engineering Systems (INES)*, pages 271–275, June 2013. doi: 10.1109/INES.2013.6632825.

[23] Jason K. Howlett, Timothy W. McLain, and Michael A. Goodrich. Learning Real-Time A* Path Planner for Unmanned Air Vehicle Target Sensing. *Journal of Aerospace Computing, Information, and Communication*, 3(3): 108–122, mar 2006. ISSN 1542-9423. doi: 10.2514/1.16623.

[24] David Payton, Regina Estkowski, and Mike Howard. *Pheromone Robotics and the Logic of Virtual Pheromones*, pages 45–57. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. ISBN 978-3-540-30552-1. doi: 10.1007/978-3-540-30552-1_5.

[25] R. Zlot, A. Stentz, M. B. Dias, and S. Thayer. Multi-robot exploration controlled by a market economy. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, volume 3, pages 3016–3023, 2002. doi: 10.1109/ROBOT.2002.1013690.

[26] Brian Yamauchi. Frontier-Based Exploration Using Multiple Robots.

[27] Howie Choset and Joel Burdick. Sensor-based exploration: The hierarchical generalized voronoi graph. *The International Journal of Robotics Research*, 19(2):96–125, 2000. doi: 10.1177/02783640022066770.

[28] Harold W Kuhn. The Hungarian Method for the Assignment Problem Introduction by. doi: 10.1007/978-3-540-68279-0_2.

[29] Dan Gusfield and Robert W. Irving. *The Stable Marriage Problem: Structure and Algorithms*. MIT Press, Cambridge, MA, USA, 1989. ISBN 0-262-07118-5.

[30] Jacques M. Bahi, Abdallah Makhoul, and Ahmed Mostefaoui. Hilbert mobile beacon for localisation and coverage in sensor networks. *International Journal of Systems Science*, 39(11):1081–1094, nov 2008. ISSN 0020-7721. doi: 10.1080/00207720802085302.

[31] Shannon V. Spires and Steven Y. Goldsmith. *Exhaustive geographic search with mobile robots along space-filling curves*, pages 1–12. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998. ISBN 978-3-540-68723-8. doi: 10.1007/BFb0033369.

[32] Y. Gabriely and E. Rimon. Spanning-tree based coverage of continuous areas by a mobile robot. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, volume 2, pages 1927–1933 vol.2, 2001. doi: 10.1109/ROBOT.2001.932890.

[33] Xiaoming Zheng, Sonal Jain, S. Koenig, and D. Kempe. Multi-robot forest coverage. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3852–3857, Aug 2005. doi: 10.1109/IROS.2005.1545323.

[34] Howie Choset. Coverage of known spaces: The boustrophedon cellular decomposition. *Autonomous Robots*, 9(3):247–253, Dec 2000.

[35] Steven Y. Goldsmith and Rush Robinett. *Collective search by mobile robots using alpha-beta coordination*, pages 135–146. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998. ISBN 978-3-540-68723-8. doi: 10.1007/BFb0033380.

[36] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, Dec 2016. ISSN 1552-3098. doi: 10.1109/TRO.2016.2624754.

[37] Hugh Durrant-Whyte and Tim Bailey. Simultaneous Localization and Mapping: Part I History of the SLAM Problem. 2006.

[38] Michael Montemerlo, Sebastian Thrun, Daphne Koller, and Ben Wegbreit. FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem.

[39] A. I. Mourikis and S. I. Roumeliotis. A multi-state constraint kalman filter for vision-aided inertial navigation. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 3565–3572, April 2007. doi: 10.1109/ROBOT.2007.364024.

[40] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g 2 o: A General Framework for Graph Optimization.

[41] Sameer Agarwal, Keir Mierle, and Others. Ceres solver. `http://ceres-solver.org`.

[42] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 225–234, Nov 2007. doi: 10.1109/ISMAR.2007.4538852.

[43] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós. Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31 (5):1147–1163, Oct 2015. ISSN 1552-3098. doi: 10.1109/TRO.2015.2463671.

[44] Jakob Engel, Thomas Schöps, and Daniel Cremers. *LSD-SLAM: Large-Scale Direct Monocular SLAM*, pages 834–849. Springer International Publishing, Cham, 2014. ISBN 978-3-319-10605-2. doi: 10.1007/978-3-319-10605-2_54.

[45] Bill Triggs, Philip F Mclauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle Adjustment — A Modern Synthesis.

[46] D. Galvez-López and J. D. Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, Oct 2012. ISSN 1552-3098. doi: 10.1109/TRO.2012.2197158.

[47] Raúl Mur-Artal and Juan D Tardós. Probabilistic Semi-Dense Mapping from Highly Accurate Feature-Based Monocular SLAM. URL `http://www.roboticsproceedings.org/rss11/p41.pdf`.

[48] J. G. Morrison, D. Gavez-Lopez, and G. Sibley. Scalable multirobot localization and mapping with relative maps: Introducing moarslam. *IEEE Control Systems*, 36(2):75–85, April 2016. ISSN 1066-033X. doi: 10.1109/MCS.2015.2512032.

[49] Matthew Collett. How desert ants use a visual landmark for guidance along a habitual route. *Proceedings of the National Academy of Sciences of the United States of America*, 107(25):11638–43, jun 2010. ISSN 1091-6490. doi: 10.1073/pnas.1001401107.

[50] J. O'Keefe and D. H. Conway. Hippocampal place units in the freely moving rat: Why they fire where they fire. *Experimental Brain Research*, 31(4): 573–590, Apr 1978. ISSN 1432-1106. doi: 10.1007/BF00239813.

[51] Michael Milford and Gordon Wyeth. Persistent Navigation and Mapping using a Biologically Inspired SLAM System. *The International Journal of Robotics Research*, 29(9):1131–1153, 2010. doi: 10.1177/0278364909340592.

[52] S.M. Stringer, T.P. Trappenberg, E.T. Rolls, and I.E.T.d. Araujo. Self-organizing continuous attractor networks and path integration: one-dimensional models of head direction cells. *Network: Computation in Neural Systems*, 13(2):217–242, jan 2002. ISSN 0954-898X. doi: 10.1080/net.13.2.217.242.

[53] S. Heath, D. Ball, R. Schulz, and J. Wiles. Communication between lingodroids with different cognitive capabilities. In *2013 IEEE International Conference on Robotics and Automation*, pages 490–495, May 2013. doi: 10.1109/ICRA.2013.6630619.

[54] Jennifer Owen. An Investigation of Loose Coupling in Evolutionary Swarm Robotics. 2013. URL `http://etheses.whiterose.ac.uk/5018/1/thesis.pdf`.

[55] C. Leung, S. Huang, and G. Dissanayake. Active slam using model predictive control and attractor based exploration. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5031, Oct 2006. doi: 10.1109/IROS.2006.282530.

[56] Joan Vallvé and Juan Andrade-Cetto. Potential information fields for mobile robot exploration.

[57] Frédéric Bourgault, Alexei A Makarenko, Stefan B Williams, Ben Grocholsky, and Hugh F Durrant-Whyte. Information Based Adaptive Robotic Exploration.

[58] J. W. Weingarten, G. Gruener, and R. Siegwart. A state-of-the-art 3d sensor for robot navigation. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 3, pages 2155–2160 vol.3, Sept 2004. doi: 10.1109/IROS.2004.1389728.

[59] Plenoptic cameras in real-time robotics. *The International Journal of Robotics Research*, 32(2):206–217, feb 2013. ISSN 0278-3649. doi: 10.1177/0278364912469420.

[60] E. Mueggler, B. Huber, and D. Scaramuzza. Event-based, 6-dof pose tracking for high-speed maneuvers. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2761–2768, Sept 2014. doi: 10.1109/IROS.2014.6942940.

[61] G. Costante, M. Mancini, P. Valigi, and T. A. Ciarfuglia. Exploring representation learning with cnns for frame-to-frame ego-motion estimation. *IEEE Robotics and Automation Letters*, 1(1):18–25, Jan 2016. ISSN 2377-3766. doi: 10.1109/LRA.2015.2505717.

[62] Andrew Howard. Multi-robot Simultaneous Localization and Mapping using Particle Filters.

[63] Robert George Reid. Large-scale simultaneous localization and mapping for teams of mobile robots, 2016.

[64] Sajad Saeedi, Michael Trentini, Mae Seto, and Howard Li. Multiple-Robot Simultaneous Localization and Mapping: A Review. *Journal of Field Robotics*, 33(1):3–46, jan 2016. ISSN 15564959. doi: 10.1002/rob.21620.

[65] William Rone and Pinhas Ben-Tzvi. Mapping, localization and motion planning in mobile multi-robotic systems. *Robotica*, 31(1):1–23, 2013. doi: 10.1017/S0263574712000021.

[66] X. S. Zhou and S. I. Roumeliotis. Multi-robot slam with unknown initial correspondence: The robot rendezvous case. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1785–1792, Oct 2006. doi: 10.1109/IROS.2006.282219.

[67] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005. ISBN 0262201623.

[68] A. Cunningham, M. Paluri, and F. Dellaert. Ddf-sam: Fully distributed slam using constrained factor graphs. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3025–3030, Oct 2010. doi: 10.1109/IROS.2010.5652875.

[69] Ryo Kurazume and Shigeo Hirose. An experimental study of a cooperative positioning system. *Autonomous Robots*, 8(1):43–52, 2000. ISSN 1573-7527. doi: 10.1023/A:1008988801987.

[70] A. Howard. Multi-robot mapping using manifold representations. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 4, pages 4198–4203 Vol.4, April 2004. doi: 10.1109/ROBOT.2004.1308933.

[71] S. Saeedi, L. Paull, M. Trentini, M. Seto, and H. Li. Group mapping: A topological approach to map merging for multiple robots. *IEEE Robotics Automation Magazine*, 21(2):60–72, June 2014. ISSN 1070-9932. doi: 10.1109/MRA.2014.2304091.

[72] S. Schwertfeger and A. Birk. Evaluation of map quality by matching and scoring high-level, topological map structures. In *2013 IEEE International Conference on Robotics and Automation*, pages 2221–2226, May 2013. doi: 10.1109/ICRA.2013.6630876.

[73] Benjamin Kuipers and Yung-Tai Byun. Toward learning robots. chapter A Robot Exploration and Mapping Strategy Based on a Semantic Hierarchy of Spatial Representations, pages 47–63. MIT Press, Cambridge, MA, USA, 1993. ISBN 0-262-72017-5. URL `http://dl.acm.org/citation.cfm?id=164605.164614`.

[74] Niko Sünderhauf, Feras Dayoub, Sean McMahon, Ben Talbot, Ruth Schulz, Peter Corke, Gordon Wyeth, Ben Upcroft, and Michael Milford. Place Categorization and Semantic Mapping on a Mobile Robot. jul 2015.

[75] Limin Wang, Sheng Guo, Weilin Huang, and Yu Qiao. Places205-vggnet models for scene recognition. *CoRR*, abs/1508.01667, 2015.

[76] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, July 1968. ISSN 0536-1567. doi: 10.1109/TSSC.1968.300136.

[77] G. Li, C. Zhu, Jianhao Du, Qi Cheng, W. Sheng, and Heping Chen. Robot semantic mapping through wearable sensor-based human activity recognition. In *2012 IEEE International Conference on Robotics and Automation*, pages 5228–5233, May 2012. doi: 10.1109/ICRA.2012.6225305.

[78] Ioannis Kostavelis and Antonios Gasteratos. Semantic mapping for mobile robotics tasks: A survey. *Robotics and Autonomous Systems*, 66:86 – 103, 2015. ISSN 0921-8890. doi: http://dx.doi.org/10.1016/j.robot.2014.12.006.

[79] Ben Talbot, Obadiah Lam, Ruth Schulz, Feras Dayoub, Ben Upcroft, and Gordon Wyeth. Find my office: Navigating real space from semantic descriptions. In *IEEE International Conference on Robotics and Automation (ICRA 2016)*, Stockholm, Sweden, May 2016.

[80] L. Paull, G. Severac, G. V. Raffo, J. M. Angel, H. Boley, P. J. Durst, W. Gray, M. Habib, B. Nguyen, S. V. Ragavan, S. Saeedi G., R. Sanz, M. Seto, A. Stefanovski, M. Trentini, and H. Li. Towards an ontology for autonomous robots. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1359–1364, Oct 2012. doi: 10.1109/IROS.2012.6386119.

[81] Michal Laclavik, Zoltan Balogh, Emil Gatial, and Ladislav Hluchy. Agent Architecture based on Semantic Knowledge Model *.

[82] Boris Motik, Conrad Bock, Achille Fokoue, Uli Sattler, Michael Smith, and Clark & Parsia. OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition) Contributors: (in alphabetical order). 2012.

[83] Moritz Tenorth and Michael Beetz. Knowrob: A knowledge processing infrastructure for cognition-enabled robots. *The International Journal of Robotics Research*, 32(5):566–590, 2013. doi: 10.1177/0278364913481635.

[84] M. Waibel, M. Beetz, J. Civera, R. D'Andrea, J. Elfring, D. Gálvez-López, K. Häussermann, R. Janssen, J. M. M. Montiel, A. Perzylo, B. Schießle,

M. Tenorth, O. Zweigle, and R. V. De Molengraft. Roboearth. *IEEE Robotics Automation Magazine*, 18(2):69–82, June 2011. ISSN 1070-9932. doi: 10.1109/MRA.2011.941632.

[85] Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, and Andrew Ng. ROS: an open-source Robot Operating System.

[86] J. Crespo, R. Barber, and O. M. Mozos. Relational model for robotic semantic navigation in indoor environments. *Journal of Intelligent & Robotic Systems*, 86(3):617–639, Jun 2017. ISSN 1573-0409. doi: 10.1007/s10846-017-0469-x.

[87] Jesus Suarez and Robin Murphy. A survey of animal foraging for directed, persistent search by rescue robotics. In *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*, pages 314–320. IEEE, nov 2011. ISBN 978-1-61284-770-2. doi: 10.1109/SSRR.2011.6106744. URL `http://ieeexplore.ieee.org/document/6106744/`.

[88] G. M. Fricke, J. P. Hecker, J. L. Cannon, and M. E. Moses. Immune-inspired search strategies for robot swarms. *Robotica*, 34(08):1791–1810, aug 2016. ISSN 0263-5747. doi: 10.1017/S0263574716000382. URL `http://www.journals.cambridge.org/abstract{_}S0263574716000382`.

[89] Binoy Shah and Howie Choset. Survey on Urban Search and Rescue Robots. *Journal of the Robotics Society of Japan*, 22(5):582–586, jul 2004. ISSN 0289-1824. doi: 10.7210/jrsj.22.582. URL `http://joi.jlc.jst.go.jp/JST.Journalarchive/jrsj1983/22.582?from=CrossRef`.

[90] Yugang Liu and Goldie Nejat. Robotic urban search and rescue: A survey from the control perspective. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 72(2):147–165, nov 2013. ISSN 09210296. doi: 10.1007/s10846-013-9822-x.

[91] Behzad Bayat, Naveena Crasta, Alessandro Crespi, António M. Pascoal, and Auke Ijspeert. Environmental monitoring using autonomous vehicles: a survey of recent searching techniques. *Current Opinion in Biotechnology*, 45:76–84, jun 2017. ISSN 0958-1669. doi: 10.1016/J.COPBIO.2017.01.009. URL `https://www.sciencedirect.com/science/article/pii/S0958166916302312`.

[92] Gideon Kowadlo and R. Andrew Russell. Robot Odor Localization: A Taxonomy and Survey. *The International Journal of Robotics Research*, 27(8): 869–894, aug 2008. ISSN 0278-3649. doi: 10.1177/0278364908095118. URL http://journals.sagepub.com/doi/10.1177/0278364908095118.

[93] Madhubhashi Senanayake, Ilankaikone Senthooran, Jan Carlo Barca, Hoam Chung, Joarder Kamruzzaman, and Manzur Murshed. Search and tracking algorithms for swarms of robots: A survey. *Robotics and Autonomous Systems*, 75:422–434, jan 2016. ISSN 0921-8890. doi: 10.1016/J.ROBOT.2015.08.010. URL https://www.sciencedirect.com/science/article/pii/S0921889015001876.

[94] Boyoon Jung. *Cooperative Target Tracking Using Mobile Robots*. PhD thesis, Los Angeles, CA, USA, 2005.

[95] Cristina Dimidov, Giuseppe Oriolo, and Vito Trianni. Random Walks in Swarm Robotics: An Experiment with Kilobots. pages 185–196. Springer, Cham, 2016. doi: 10.1007/978-3-319-44427-7_16. URL http://link.springer.com/10.1007/978-3-319-44427-7{_}16.

[96] Erol Şahin. Swarm Robotics: From Sources of Inspiration to Domains of Application. pages 10–20. Springer, Berlin, Heidelberg, 2005. doi: 10.1007/978-3-540-30552-1_2. URL http://link.springer.com/10.1007/978-3-540-30552-1{_}2.

[97] Y. Gabriely and E. Rimon. Spanning-tree based coverage of continuous areas by a mobile robot. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, volume 2, pages 1927–1933. IEEE. ISBN 0-7803-6576-3. doi: 10.1109/ROBOT.2001.932890. URL http://ieeexplore.ieee.org/document/932890/.

[98] Micael S. Couceiro, Patricia A. Vargas, Rui P. Rocha, and Nuno M.F. Ferreira. Benchmark of swarm robotics distributed techniques in a search task. *Robotics and Autonomous Systems*, 62(2):200–213, feb 2014. ISSN 09218890. doi: 10.1016/j.robot.2013.10.004.

[99] Lynne E. Parker. Decision Making as Optimization in Multi-robot Teams. pages 35–49. Springer, Berlin, Heidelberg, 2012. doi: 10.1007/978-3-642-28073-3_4. URL http://link.springer.com/10.1007/978-3-642-28073-3{_}4.

[100] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948. IEEE. ISBN 0-7803-2768-3. doi: 10.1109/ICNN.1995.488968. URL http://ieeexplore.ieee.org/document/488968/.

[101] Y. Shi and R. Eberhart. A modified particle swarm optimizer. In *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, pages 69–73. IEEE. ISBN 0-7803-4869-9. doi: 10.1109/ICEC.1998.699146. URL http://ieeexplore.ieee.org/document/699146/.

[102] Jim Pugh and Alcherio Martinoli. Inspiring and Modeling Multi-Robot Search with Particle Swarm Optimization. In *2007 IEEE Swarm Intelligence Symposium*, pages 332–339. IEEE, apr 2007. ISBN 1-4244-0708-7. doi: 10.1109/SIS.2007.367956. URL http://ieeexplore.ieee.org/document/4223193/.

[103] Jim Pugh and Alcherio Martinoli. Multi-robot Learning with Particle Swarm Optimization. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '06, pages 441–448, New York, NY, USA, 2006. ACM. ISBN 1-59593-303-4. doi: 10.1145/1160633.1160715. URL http://doi.acm.org/10.1145/1160633.1160715.

[104] James Hereford and Michael Siebold. Multi-robot search using a physically-embedded Particle Swarm Optimization. *International Journal of Computational Intelligence Research*, 4(2), 2008. ISSN 09741259. doi: 10.5019/j.ijcir.2008.138. URL http://www.ijcir.com/publishedPapers.php?showDetails=Y{&}idArticle=138{&}volume=1{&}number=1{&}volume{_}id=2{&}number{_}id=4.

[105] Micael S. Couceiro, Rui P. Rocha, and Nuno M. F. Ferreira. A novel multi-robot exploration approach based on Particle Swarm Optimization algorithms. In *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*, pages 327–332. IEEE, nov 2011. ISBN 978-1-61284-769-6. doi: 10.1109/SSRR.2011.6106751. URL http://ieeexplore.ieee.org/document/6106751/.

[106] A. Ghanbarzadeh, E. Koç, S. Otri, S. Rahim, and M. Zaidi. The Bees Algorithm — A Novel Tool for Complex Optimisation Problems. *Intelligent Production Machines and Systems*, pages 454–459, jan 2006. doi:

10.1016/B978-008045157-2/50081-X. URL https://www.sciencedirect.com/science/article/pii/B978008045157250081X.

[107] Aleksandar Jevtic, Álvaro Gutierrez, Diego Andina, and Mo Jamshidi. Distributed Bees Algorithm for Task Allocation in Swarm of Robots. *IEEE Systems Journal*, 6(2):296–304, jun 2012. ISSN 1932-8184. doi: 10.1109/JSYST.2011.2167820. URL http://ieeexplore.ieee.org/document/6047561/.

[108] Marco Dorigo, Gianni Di Caro, and Luca M Gambardella. Ant Algorithms for Discrete Optimization. *Artif. Life*, 5(2):137–172, 1999. ISSN 1064-5462. doi: 10.1162/106454699568728. URL http://dx.doi.org/10.1162/106454699568728.

[109] Nicholas R. Hoff, Amelia Sagoff, Robert J. Wood, and Radhika Nagpal. Two foraging algorithms for robot swarms using only local communication. In *2010 IEEE International Conference on Robotics and Biomimetics*, pages 123–130. IEEE, dec 2010. ISBN 978-1-4244-9319-7. doi: 10.1109/ROBIO.2010.5723314. URL http://ieeexplore.ieee.org/document/5723314/.

[110] Yuhua Zou, Dehan Luo, and Weihai Chen. Swarm robotic odor source localization using ant colony algorithm. In *2009 IEEE International Conference on Control and Automation*, pages 792–796. IEEE, dec 2009. ISBN 978-1-4244-4706-0. doi: 10.1109/ICCA.2009.5410516. URL http://ieeexplore.ieee.org/document/5410516/.

[111] K.M. Passino. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine*, 22(3):52–67, jun 2002. ISSN 0272-1708. doi: 10.1109/MCS.2002.1004010. URL http://ieeexplore.ieee.org/document/1004010/.

[112] Mirbek Turduev, Murat Kirtay, Pedro Sousa, Veysel Gazi, and Lino Marques. Chemical concentration map building through bacterial foraging optimization based search algorithm by mobile robots. In *2010 IEEE International Conference on Systems, Man and Cybernetics*, pages 3242–3249. IEEE, oct 2010. ISBN 978-1-4244-6586-6. doi: 10.1109/ICSMC.2010.5642297. URL http://ieeexplore.ieee.org/document/5642297/.

[113] A. Dhariwal, G.S. Sukhatme, and A.A.G. Requicha. Bacterium-inspired robots for environmental monitoring. In *IEEE International Conference on*

Robotics and Automation, 2004. Proceedings. ICRA '04. 2004, pages 1436–1443 Vol.2. IEEE, 2004. ISBN 0-7803-8232-3. doi: 10.1109/ROBOT.2004.1308026. URL http://ieeexplore.ieee.org/document/1308026/.

[114] K.N. Krishnanand and D. Ghose. Detection of multiple source locations using a glowworm metaphor with applications to collective robotics. In *Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005.*, pages 84–91. IEEE. ISBN 0-7803-8916-6. doi: 10.1109/SIS.2005.1501606. URL http://ieeexplore.ieee.org/document/1501606/.

[115] K.N. Krishnanand, P. Amruth, M.H. Guruprasad, S.V. Bidargaddi, and D. Ghose. Glowworm-inspired robot swarm for simultaneous taxis towards multiple radiation sources. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 958–963. IEEE. ISBN 0-7803-9505-0. doi: 10.1109/ROBOT.2006.1641833. URL http://ieeexplore.ieee.org/document/1641833/.

[116] Krishnanand Kaipa and Debasish Ghose. Chasing Multiple Mobile Signal Sources: A Glowworm Swarm Optimization Approach., 2007.

[117] Xin-She Yang. *Nature-Inspired Metaheuristic Algorithms.* Luniver Press, 2008. ISBN 1905986106, 9781905986101.

[118] Chang Liu, Zhongqiang Gao, and Weihua Zhao. A New Path Planning Method Based on Firefly Algorithm. In *2012 Fifth International Joint Conference on Computational Sciences and Optimization*, pages 775–778. IEEE, jun 2012. ISBN 978-1-4673-1365-0. doi: 10.1109/CSO.2012.174. URL http://ieeexplore.ieee.org/document/6274838/.

[119] Zongyao Wang and Dongbing Gu. Cooperative Target Tracking Control of Multiple Robots. *IEEE Transactions on Industrial Electronics*, 59(8):3232–3240, aug 2012. ISSN 0278-0046. doi: 10.1109/TIE.2011.2146211. URL http://ieeexplore.ieee.org/document/5754570/.

[120] Levent Bay\ind\ir. A Review of Swarm Robotics Tasks. *Neurocomput.*, 172(C):292–321, 2016. ISSN 0925-2312. doi: 10.1016/j.neucom.2015.05.116. URL http://dx.doi.org/10.1016/j.neucom.2015.05.116.

[121] Serge Kernbach, Ronald Thenius, Olga Kernbach, and Thomas Schmickl. Re-embodiment of Honeybee Aggregation Behavior in an Artificial Micro-Robotic System. 2009. doi: 10.1177/1059712309104966. URL http://journals.sagepub.com/doi/pdf/10.1177/1059712309104966.

[122] Farshad Arvin, Khairulmizam Samsudin, Abdul Rahman Ramli, and Masoud Bekravi. Imitation of Honeybee Aggregation with Collective Behavior of Swarm Robots. *International Journal of Computational Intelligence Systems*, 4(4):739–748, jun 2011. ISSN 1875-6891. doi: 10.1080/18756891.2011.9727825. URL http://www.tandfonline.com/doi/abs/10.1080/18756891.2011.9727825.

[123] Jean-marc Ame, Jean-marc Ame, Colette Rivault, Jean-louis Deneubourg, and Université Libre De Bruxelles. Cockroach aggregation based on strain odour recognition", Animal Behaviour. *ANIMAL BEHAVIOUR*, 68:2004, 2004. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.118.7302.

[124] Thomas Schmickl, Christoph Möslinger, and Karl Crailsheim. Collective perception in a robot swarm. In *Swarm robotics*, pages 144–157. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. ISBN 978-3-540-71540-5, 978-3-540-71541-2. doi: 10.1007/978-3-540-71541-2_10.

[125] Gaige Wang, Lihong Guo, Hong Duan, Luo Liu, and Heqi Wang. A Modified Firefly Algorithm for UCAV Path Planning. *International Journal of Hybrid Information Technology*, 5(3), 2012.

[126] Lynne E. Parker. Distributed Algorithms for Multi-Robot Observation of Multiple Moving Targets. *Autonomous Robots*, 12(3):231–255, 2002. ISSN 09295593. doi: 10.1023/A:1015256330750. URL http://link.springer.com/10.1023/A:1015256330750.

[127] Jun Ota, Tamio Arai, Yuji Yoshimura, Natsuki Miyata, Echi Yoshida, Daisuke Kurabayashi, and Jun Sasaki. Motion planning of multiple mobile robots by a combination of learned visibility graphs and virtual impedance. *Advanced Robotics*, 10(6):605–620, jan 1995. ISSN 0169-1864. doi: 10.1163/156855396X00246. URL https://www.tandfonline.com/doi/full/10.1163/156855396X00246.

[128] Geunho Lee, Nak Young Chong, and Henrik Christensen. Tracking multiple moving targets with swarms of mobile robots. *Intelligent Service Robotics*, 3 (2):61–72, apr 2010. ISSN 1861-2776. doi: 10.1007/s11370-010-0059-2. URL http://link.springer.com/10.1007/s11370-010-0059-2.

[129] H. B. Mann and D. R. Whitney. On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. *The Annals of Mathematical Statistics*, 18(1):50 – 60, 1947. doi: 10.1214/aoms/1177730491. URL https://doi.org/10.1214/aoms/1177730491.

[130] Andrfis Vargha and Harold D Delaney. A Critique and Improvement of the CL Common Language Effect Size Statistics of McGraw and Wong. *Journal of Educational and Behavioral Statistics Summer*, 25(2):101–132, 2000. URL http://journals.sagepub.com/doi/pdf/10.3102/10769986025002101.

[131] Daniël Lakens. Calculating and reporting effect sizes to facilitate cumulative science: a practical primer for t-tests and anovas. *Frontiers in Psychology*, 4, 2013. ISSN 1664-1078. doi: 10.3389/fpsyg.2013.00863. URL https://www.frontiersin.org/articles/10.3389/fpsyg.2013.00863.

[132] Charles J. Kowalski. On the Effects of Non-Normality on the Distribution of the Sample Product-Moment Correlation Coefficient. *Journal of the Royal Statistical Society Series C*, 21(1):1–12, March 1972. doi: 10.2307/2346598. URL https://ideas.repec.org/a/bla/jorssc/v21y1972i1p1-12.html.

[133] Jinghuan Wen, Huimin Ma, and Xiaoqin Zhang. Optimization of the occlusion strategy in visual tracking. In *Tsinghua Science and Technology*, volume 21, pages 221–230. IEEE, 2016. ISBN 0-7803-2768-3. doi: 10.1109/ICNN.1995.488968.

[134] Duc Truong Pham, A Ghanbarzadeh, E Koç, S. Otri, S Rahim, and M. Zaidi. The Bees Algorithm - A Novel Tool for Complex Optimization Problems. *Intelligent Production Machines and Systems*, pages pp. 454—-460, 2006.

[135] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968. doi: 10.1109/TSSC.1968.300136.

[136] T.H. Cormen, Charles Leiserson, R.L. Rivest, and C. Stein. Section 24.3: Dijkstra's algorithm. *Introduction to Algorithms*, pages 595–601, 01 2001.

[137] Behzad Bayat, Naveena Crasta, Alessandro Crespi, António M. Pascoal, and Auke Ijspeert. Environmental monitoring using autonomous vehicles: a survey of recent searching techniques. *Current Opinion in Biotechnology*, 45:76–84, jun 2017. ISSN 0958-1669. doi: 10.1016/J.COPBIO.2017.01.009. URL https://www.sciencedirect.com/science/article/pii/S0958166916302312.

[138] Glenn Jocher, Alex Stoken, Jirka Borovec, NanoCode012, ChristopherSTAN, Liu Changyu, Laughing, tkianai, Adam Hogan, lorenzomammana, yxNONG, AlexWang1900, Laurentiu Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, Francisco Ingham, Frederik, Guilhen, Hatovix, Jake Poznanski, Jiacong Fang, Lijun Yu, changyu98, Mingyu Wang, Naman Gupta, Osama Akhtar, PetrDvoracek, and Prashant Rai. ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements, October 2020. URL https://doi.org/10.5281/zenodo.4154370.

[139] Carlo Pinciroli, Vito Trianni, Rehan O'Grady, Giovanni Pini, Arne Brutschy, Manuele Brambilla, Nithin Mathews, Eliseo Ferrante, Gianni Di Caro, Frederick Ducatelle, Mauro Birattari, Luca Maria Gambardella, and Marco Dorigo. ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence*, 6(4):271–295, 2012.

[140] Paulo Gonçalves, Paulo Torres, Carlos Alves, Francesco Mondada, Michael Bonani, Xavier Raemy, James Pugh, Christopher Cianci, Adam Klaptocz, Stephane Magnenat, Jean-Christophe Zufferey, Dario Floreano, and A. Martinoli. The e-puck, a robot designed for education in engineering. *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, 1, 01 2009.

[141] *Structure of Decision: The Cognitive Maps of Political Elites*. Princeton University Press, 1976. URL http://www.jstor.org/stable/j.ctt13x0vw3.

[142] Márcio Mendonça, Ivan R. Chrun, Flávio Neves, and L.V.R. Arruda. A cooperative architecture for swarm robotic based on dynamic fuzzy cognitive maps. *Engineering Applications of Artificial Intelligence*, 59:122–132, 2017. ISSN 0952-1976. doi: https://doi.org/10.1016/j.engappai.2016.12.017. URL https://www.sciencedirect.com/science/article/pii/S0952197616302524.

[143] Márcio Mendonça, Lúcia Arruda, and Flávio Jr. Autonomous navigation system using event driven-fuzzy cognitive maps. *Applied Intelligence*, 37: 1–14, 09 2011. doi: 10.1007/s10489-011-0320-1.

[144] R. Brooks. A robust layered control system for a mobile robot. *IEEE Journal on Robotics and Automation*, 2(1):14–23, 1986. doi: 10.1109/JRA.1986.1087032.

[145] Nicolas Maillot. *Ontology Based Object Learning and Recognition.* Theses, Université Nice Sophia Antipolis, December 2005. URL `https://tel.archives-ouvertes.fr/tel-00327542`.

[146] Torralba, Murphy, Freeman, and Rubin. Context-based vision system for place and object recognition. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 273–280 vol.1, 2003. doi: 10.1109/ICCV.2003.1238354.

[147] Serge Galam. Sociophysics: A review of galam models. *International Journal of Modern Physics C*, 19(03):409–440, 2008.