People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

University of 8 May 1945-Guelma-

Faculty of Mathematics, Computer Science and Science of Matter

Department of Computer Science

Master Thesis

Specialty: Computer Science

Option: Science and technology of information and communication

Theme:

Towards an efficient Multi-Robots Search and Rescue strategy

Presented by: Doua Ghelis

Jury Members:

N°	Full Name	Quality
1	Dr. Said BRAHIMI	Chairman
2	Dr. Ouarda ZEDADRA	Supervisor
3	Dr. Mohammed Chaoui	Examiner

June 2023

Acknowledgement

First and foremost, I would like to wholeheartedly express my gratitude to Almighty Allah for giving me the strength and patience to accomplish this humble work.

I would like to express my sincerest thanks to my supervisor Dr. Ouarda ZEDADRA for her precious guidance, help, advice, and constant support during the realization of this dissertation. It was an honor for me to be supervised by such an incredible teacher.

I extend my thanks to the board of examiners for devoting their precious time and effort to read and evaluate my work.

I would like to express my special gratitude to all the teachers who taught me at the University of 8 May 1945, Guelma.

My appreciation further extends to all the people whose assistance and advice proved monumental towards the accomplishment of this dissertation.

Finally, I would like to express how deeply indebted I feel to every devoted and committed teacher that I had the privilege of learning from

Dedications

This dissertation is dedicated to:

My father*, may Allah have mercy on him, and* ***my wonderful mother****. Words are not enough to express how much I love both of them. I am extremely grateful for all the sacrifices they made for me as well as for raising me the best way they could to make of me a better version of myself with each passing day.*

My brothers and sister, *their unwavering support and love have been instrumental in my journey. I am blessed to have such a close-knit family that has always been there for me, providing guidance and encouragement,*

My supervisor*, whose expertise and guidance have been invaluable throughout this dissertation.*

My wonderful friends, *who have been my pillars of strength. In particular, I would like to express my heartfelt gratitude to my best friend****, Hasna****, for her unwavering support and companionship. Her presence in my life has been a constant source of inspiration and motivation.*

My large family*.*

Abstract

Due to the hazardous and hostile conditions that arise following a natural disaster, multi-robot systems are employed as substitutes for human beings in the task of searching and rescuing victims. The exploration of unknown environments is considered as a paramount concern in the field of mobile robotics. It consists a foundational stage for various applications such as search and rescue, cleaning tasks, and foraging.

We proposed in this work, a new swarm robotic search strategy for search and rescue operations. The proposed strategy is inspired by the hunting behavior of Penguins. It hybridizes the Penguin Search Optimization Algorithm with the Random Walk Algorithm to modulate the global and local search behaviors of robots. To evaluate the effectiveness of our strategy, we implemented it in the ARGoS multi-robot simulator and conducted a series of experiments. The results obtained from these experiments clearly demonstrate the efficiency and effectiveness of our proposed search strategy.

Keywords

swarm intelligence, swarm robotics, search and rescue problem, Penguin Search Optimization Algorithm, Random Walk Algorithm.

Table of Content

List of figures

List of tables

List of Abbreviations

ABC: Artificial Bee Colony

ABCMEP: Artificial Bee Colony with Modified Evolutionary Programming

ACO: Ant Colony Optimization

A-Eval: The across-instance evaluation

AI: Artificial intelligence

ARGoS: Autonomous Robots Go Swarming

ASPV: Ant Search Path with Visibility algorithm

BA*: Boustrophedon Motion Algorithm with A* Search

BA: Bees Algorithm

BCD: Boustrophedon Cell Decomposition Algorithm

BM: Boustrophedon Motion Algorithm

CNP: Contract Net protocol

CTDE: Centralized training approach with decentralized execution

DAI: Distributed Artificial Intelligence

DPSO: Distributed particle swarm optimization

FA: Firefly Algorithm

FOA: Fly Optimization algorithm

GA: Genetic Algorithm

MARL: multi-agent reinforcement learning

MEP: Modified Evolutionary Programming

M-Eval: Multi-run evaluation

MILP: Mixed-Integer Linear Program

MPeSOA: Modified Penguin Search Optimization Algorithm

PeSOA: Penguin Search Optimization Algorithm

PSO: Practical Swarm Optimization

RS: Random Search

RSH: Randomness Search Heuristic

UAVs: Swarms of unmanned aerial vehicles

UAVs: Unmanned aerial vehicles

VRFA: Velocity-inspired Robotic Fruit Fly Algorithm

General introduction

1. Scientific positioning

To determine whether a machine is conscious, Alan Turing proposed an experiment known as the Turing test in 1950. Based on this experiment, the term "Artificial Intelligence" was coined, which is defined as the development of computer programs that simulate human intelligent behavior to solve problems using high-level cognitive capabilities. The latter include learning, memory organization, and reasoning. An overall objective of this discipline is to make machines rational by using autonomous decision-making and perception functions in an unknown environment (**Zedadra, 2016**).

Unlike AI, which models the intelligent behavior of a single entity, Distributed Artificial Intelligence (DAI) is a distributed group of entities that collaborate to solve global problems. An entity may range from a simple processing element to a complex entity that exhibits rational behavior. Collaborative problem-solving or task performance implies that information must be shared between the group as a whole to accomplish a task (**Demazeau et al., 1990**).

Through the expansion of DAI, new research fields have emerged, such as “Multi-Agent Systems”, and “Swarm Intelligence”. This latter, aims to emulate the behavior of some living organisms such as ants, bees, termites, fireflies…etc.

2. Problematic, objectives, and contributions

The exploration of unknown environments is a fundamental and pivotal concern within the realm of mobile robotics. It serves as a primary stage for a diverse range of applications, spanning domains such as cleaning, search and rescue, and foraging, among others. This dissertation introduces a new contribution aimed at addressing the challenge of achieving efficient exploration in the context of Search and Rescue operations. This is accomplished through the utilization of a coordinated swarm of robots, inspired by the behavior of Penguins. The proposed algorithm uses two swarm intelligence-based algorithms: The Penguin Search Optimization Algorithm (PeSOA) and the Random Walk Algorithm (RWA). The proposed algorithm has been successfully implemented on the ARGoS mobile robotics platform, showcasing its effectiveness in various mobile robotics experiments.

3. Structure of the dissertation

This manuscript is organized in four different chapters. A brief description of these chapters is given in below:

The first chapter: This chapter provides an overview of the research domains of swarm intelligence and swarm robotics.

The second chapter: Within this chapter, we undertake a comprehensive synthesis of pertinent research in the field of search and rescue. Subsequently, we present a comparative analysis in tabular form, where we evaluate the various search and rescue approaches against a predetermined set of criteria that we consider valuable for assessment.

The third chapter: In this chapter, we delve into the domain of problem design, addressing the research problem, stating the objectives, and presenting our proposed solution. Furthermore, our focus is directed toward the MPeSOA algorithm, as we illustrate its flowchart, expound upon the textual description of its states, and provide the accompanying pseudocodes for effective implementation.

The fourth chapter: Within this chapter, we provide a comprehensive presentation of the outcomes derived from the implementation and rigorous testing of the proposed algorithm. Additionally, we offer an in-depth overview of the ARGoS simulation platform, along with a detailed description of the characteristics of the employed robots (footbots).

We conclude this manuscript with a general conclusion and some future works.

Chapter 1: Swarm Intelligence and Swarm Robotics: definitions and principles

1. Introduction

Interest in technology has refreshingly grown in recent decades and has resulted in revolutionary changes in all life domains. The recent technological development led plenty of new and complex problems to emerge, among which are optimization problems. In order to solve such type of problem, many solutions have been proposed, including swarm intelligence.

Swarm intelligence is considered as one of the most efficient ways to solve complex problems in fields such as computer science, engineering, business, etc. It takes inspiration from the biological systems and the behaviors of living organisms, as it ensures good communication between members (direct and indirect).

Research about the insect world has found that each member of the colony works according to its own specifications, without receiving central orders. This leads us to ask the following questions: What shapes the identity of the leader? who makes future expectations? and who is responsible for improving strategies and achieving balance?

This chapter is dedicated to answer the aforementioned questions, by diving more into the realm of swarm intelligence, shedding light on its algorithms, applications. It also aims at finding out more about swarm robotics, and its characteristics, strengths, behaviors, applications, and simulators.

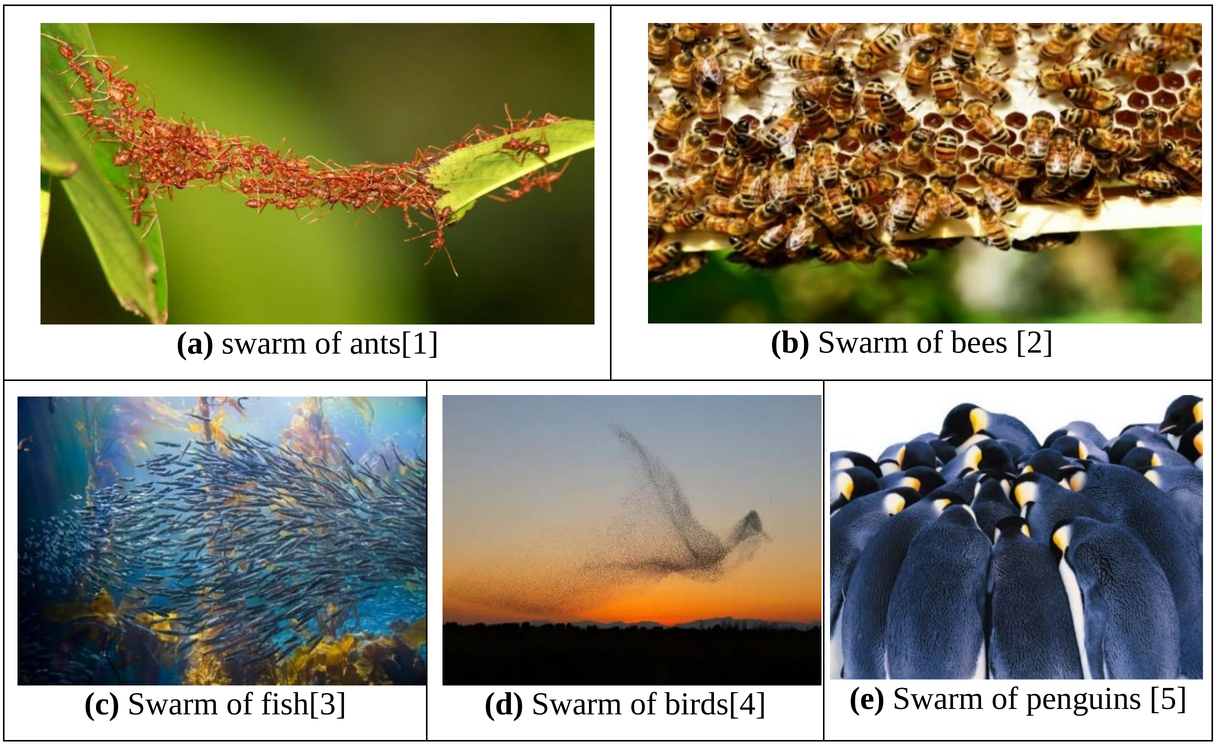
2. Swarm Intelligence

2. 1 Definitions

The term Swarm Intelligence was used for the first time by Gerardo Beni and Jing Wang in their study of Cellular Robotic Systems. They defined this discipline as a collective behavior of Autonomous, self-organized Systems **(Beni and Wang, 1989)**.

Swarm Intelligence can also be defined as an emerging field inspired by the biological nature system that ensures the cooperative work between simple entities (individuals). These latter have the ability to work collectively in decentralized, self-organized environments and to interact harmoniously without taking orders from each other (**Sadiku et al. ,2021**).

Natural Swarm Intelligence Systems include colonies of ants, bird flocks, animal herds, bees, etc. In this sense, swarms can be defined as large numbers of homogeneous (or heterogeneous) individuals (**Sadiku et al. ,2021**).



(a) swarm of ants[1]

(b) Swarm of bees [2]

(c) Swarm of fish[3]

(d) Swarm of birds[4]

(e) Swarm of penguins [5]

Figure 1.1: shows some examples of swarm intelligence systems observed in nature, where Figure 1.1(a) shows a colony of ants, Figure 1.1(b) shows swarm of bees, Figure 1.1(c) shows a group of fish, Figure 1.1(d) shows a group of birds, and Figure 1.1(d) shows a swarm of penguins.

Íñiguez introduced in **(Íñiguez, 2017**) five principles of swarm intelligence: awareness, autonomy, solidarity, expandability, and resiliency. We explain in below each of these principles:

- ***Awareness*****:** each member must be aware of the surroundings and abilities.
- ***Autonomy:*** each member must operate as an autonomous master (not as a slave); this is essential to self-coordinate allocation of labor.
- ***Solidarity:*** *c*ooperation is needed when a team achieves a task that could not be done by a single individual. Each member must cooperate in solidarity. When a task is completed, each member should autonomously look for a new task.

- ***Expandability:*** the system must permit expansion where members are dynamically aggregated.
- ***Resiliency:*** the system must be self-healing; when members are removed, the remaining members should undertake the unfinished tasks.

2. 2 Swarm Intelligence-based Algorithms

Swarm Intelligence-based algorithms are computational models inspired by the nature system. Several algorithms have been proposed based on natural phenomena (systems) which have proven their effectiveness in solving various real-life problems **(Ahmed et Glasgow ,2012)**. In below we present the most ones: Ant Colony Optimization (**ACO**), Practical Swarm Optimization (**PSO**), Bees Algorithm (**BA**), and Firefly Algorithm (**FA**).

1) **Ant Colony Optimization**: The ACO algorithm is used to solve optimization problems inspired by ant colony behavior. Each ant of the colony lays some pheromone (in varying quantities) on the way to mark the path for other ants to follow and find the resources located (considering the shortest schema) (**Colorni et al.,1991**)**.**

2) **Practical Swarm Optimization**: The PSO algorithm is a search technique introduced by Kennedy and Eberhart (1995). It is inspired by the flocking behavior of birds and schooling behavior of fish in nature (**Ahmed et Glasgow ,2012**). PSO is used for nonlinear function optimization and neural network training, in addition to several other real-life problems. PSO aims at finding the near-optimal solution, respecting particular rules and criteria.

3) **Bees Algorithm:** Food foraging is considered as one of the primary purposes behind the existence of living beings. Each animal has its own method of finding food resources. An example would be the foraging behavior of honey bees, which inspired researchers to develop a new search algorithm called Bees Algorithm. This algorithm is used to solve optimization problems by finding the optimal solution.

4) **Firefly Algorithm:** The firefly is an insect which produces flashlights called bioluminescence. The latter is used for three main purposes, namely mating, preying, and protecting. This phenomenon (the flashing behavior of fireflies and bioluminescent communication) inspired Yang to introduce a new Metaheuristic algorithm for solving several problems **(Yang, 2010).**

2. 3 Domain Applications

The basic principles of Swarm Intelligence, like any other modernistic field of study, can be applied to different life domains in order to achieve further growth and development. Some of the areas in which swarm intelligence principles are applied include:

- **Swarm robotics:** Researchers worked on implementing the principles of swarm intelligence on the robot system, which resulted in the emergence of swarm robotics as a refreshingly new field of study. The latter strived to accomplish difficult tasks that are impossible to execute by one single individual, through the simulation of certain cooperative and collective behaviors of some natural entities. Nevertheless, swarm robotics promotes principles of autonomy and self-organization within the robot swarm system **(Sadiku et al., 2021)**.
- **Engineering:** The use of SI-based algorithms can be beneficial in solving practical engineering problems, such as optimization problems in mechanical engineering and transportation engineering. As a part of the Swarm Engineering approach, a large number of simple robots, whose interactions produce a global behavior, are used **(Sadiku et al., 2021)**.
- **Business:** The evolution of Swarm intelligence in recent years has forced businesses to keep up with changing markets. Several optimization problems have been solved by using business systems that mimic swarm behavior. Examples of these include freight logistics, telecommunication network routing, and distribution warehouse planning workflows **(Sadiku et al., 2021)**.

3. Swarm Robotics

3. 1 Definitions

The collective behavior of natural swarms has inspired researchers to apply different principles of swarm life to the robotics field in order to accomplish complex tasks that are impossible for single individuals to perform. Consequently, the term "Swarm Robotics" was coined. Several definitions have been given in literature, in below we present some of them:

Definition 3.1: "*Swarm robotics is the study of how large number 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*." **(Şahin, 2005)**

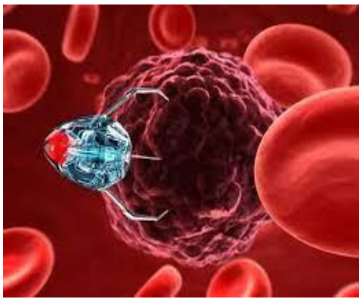
Definition 3.2: *"Swarm robotics is defined following the principles that a swarm sys-tem should have a large number of robots, tasks should be solved and improved using a swarm system, and that the robots exchange local information through limited com-munication distances"* **(Huang et al., 2019).**

Definition 3.3: *"Swarm robotics is the study of how to create groups of robots that operate without dependence on external infrastructure or any form of centralized control. In a robot swarm, the collective behavior of the robots results from the local inter-action between the robots and the environment in which they interact"***(Dorigo et Gambardella, 1997)**.

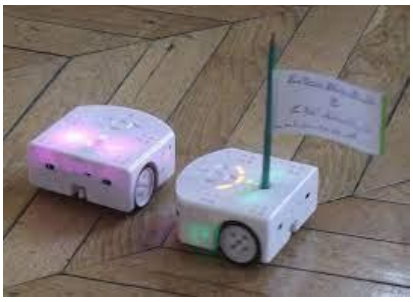
(a) Swarmanoids [6]

(b) Drones [7]

(c) Kilobots [8]



(d) Nanobots [9]



(e) Tymio [10]

Figure 1.2*: shows some examples of swarm robotic systems, where Figure 1.2(a) shows a group of Swarmanoids, Figure 1.2(b) shows a group of drones, Figure 1.2(c) shows Kilobots, Figure 1.2(d) shows Nanobots, and Figure 1.2(e) shows Tymio robots.*

3. 2 Characteristics of swarm robots

Ekelhof and Paoli **(Ekelhof and Paoli, 2020)** have presented the five main characteristics of swarm robotics:

1. **Mass**: Mass refers to the size of a particular swarm. The number of members belonging to a single swarm can range from a few to thousands of individuals. Swarms cannot be defined, however, in terms of a specific number for that the optimal size of a swarm will ultimately be dictated by the its capabilities and the mission assigned to it.

2. **Diversity**: A robotic swarm is not necessarily composed of identical robotic units, as they can be heterogeneous in their composition. In other words, a robotic swarm can mix simple robots with more complex robots or manned systems with unmanned systems. Additionally, the swarm can operate across multiple domains with robots operating both on land and at sea.
3. **Collective and Collaborative Behavior:** This characteristic is considered the most important one in the world of swarms. The reason is that we do not focus on the identification of individuals' numbers in the swarm, but we rather show interest in how the members of a swarm engage in collective and collaborative behavior to achieve common goals. This collaborative behavior can be both internal (between the members themselves) and external (between the members and the environment). For instance, when a basketball team participates in a competition, each individual develops a specific mental representation of his mission and role in the team. However, actual success cannot be achieved unless members of the team effectively execute the plan that governs their missions and roles in the game.
4. **Intra-swarm Communication:** Collaborative behavior can be achieved through different forms of communication in order to facilitate information exchange among robots. Communication between swarm members can be either direct or indirect. Usually, direct communication is based on observing the behavior of a neighbor or a fellow member, unlike indirect communication, which depends on environmental changes.
5. **Autonomy and Decentralization:** Within the swarm, each robotic unit functions as an autonomous member that reacts in accordance with internal rules and environmental conditions. It can make its own decisions without referring to a central entity to control its actions.

3. 3 The strengths of Swarm Robotics

According to Cheraghi et al. **(Cheraghi et al., 2021)** some of the strengths of swarm robotics are:

- **Flexibility**: Flexibility is the ability of individuals to adapt and change their roles quickly in response to changes in the external environment and new assignments.

- **Robustness**: Robustness refers to the ability of the group to continue working even if one or more units fail, which means that its performance is not impacted by the failure. Instead, the group attempts to find and fix all errors as quickly as possible.
- **Self-Organization:** A self-organizing robot is one that is capable of interacting with the environment and reorganizing itself as needed. In other words, a self-organizing system is a system in which members of a group aim to accomplish tasks cooperatively without relying on a central authority to coordinate their actions.

3. 4 Swarm robotic behaviors

Schranz et al. **(Schranz et al., 2020)** have classified and listed the basic swarm robotic behaviors as follows:

3.4.1. Spatial Organization

Using these behaviors, robots can move in a swarm while spatially organizing themselves or other objects in the environment:

1) **Aggregation**: Aggregation involves each robot congregating spatially in a specific region of the environment. By doing so, individuals of the swarm are able to get spatially close to one another for more efficient interaction.

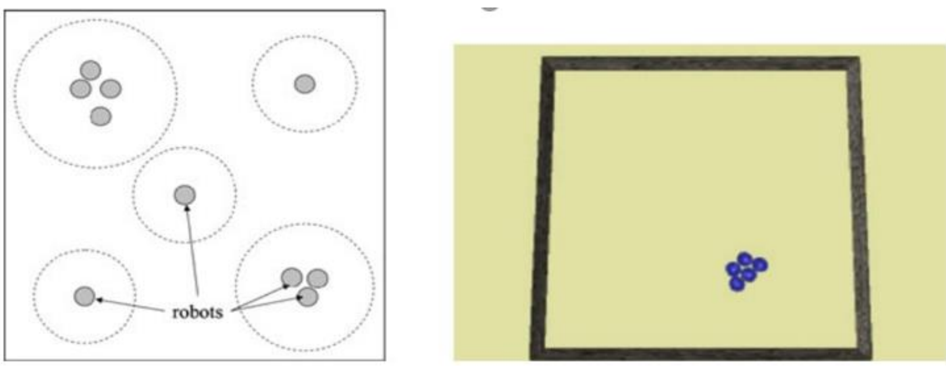


Figure 1.3**: Examples of the aggregation collective behavior* ***(Brambilla et al., 2013)

2) **Pattern formation:** The pattern formation of robots is the process by which robots in the swarm arrange themselves into a specific shape. It is possible for them to create a chain in which they form a line, typically in order to establish multi-hop communication between two points.

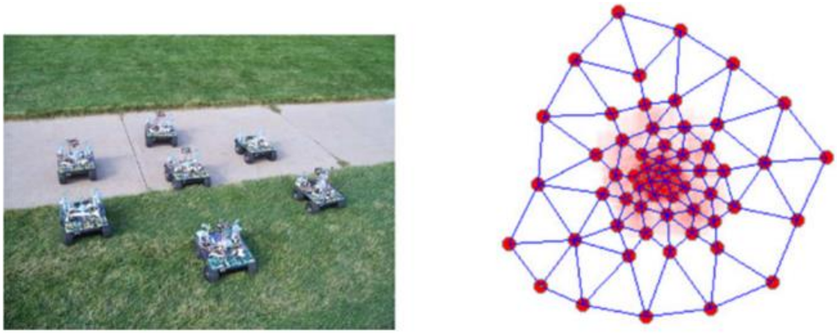


Figure 1.4**: Examples of the pattern formation collective behavior **(Brambilla et al., 2013)

3) **Self-assembly**: Self-assembly works on establishing strong connections between robots, for the purpose of forming structures. In this sense, robots can either be connected physically or virtually through communication links. One special case is morphogenesis, in which the swarm evolves into a predefined shape.

Figure 0.5:** Examples of the self-assembly collective behavior **(Brambilla et al., 2013)

4) **Object clustering and assembly:** A swarm of robots can manipulate spatially distributed objects by clustering and assembling them. In order for construction processes to be successful, objects must be properly clustered and assembled.

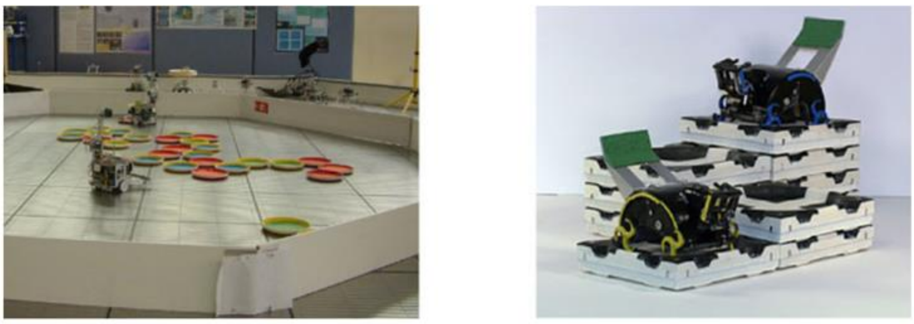


Figure 1.6**: Examples of the object clustering and assembling collective behavior **(Brambilla et al., 2013)

3.4.2. Navigation

As a result of the following behaviors, swarms of robots are able to move in harmony throughout an environment:

1) **Collective exploration:** During collective exploration, a swarm of robots explores the environment cooperatively. A situational overview, object search, environmental monitoring, or network communication can be accomplished using this behavior.

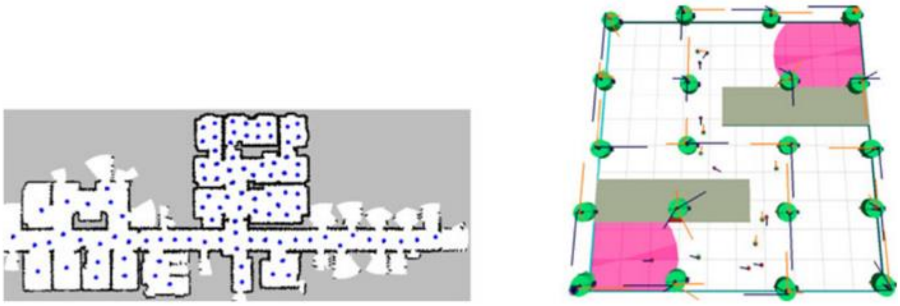


***Figure 0.7**: Examples of the collective exploration behavior* **(Brambilla et al., 2013)**

2) **Coordinated motion**: Coordinated motion is used to move a swarm of robots in a formation. Depending on the formation, the swarm can take the form of a well-defined shape, such as a line, or can be arbitrary, such as a flock.

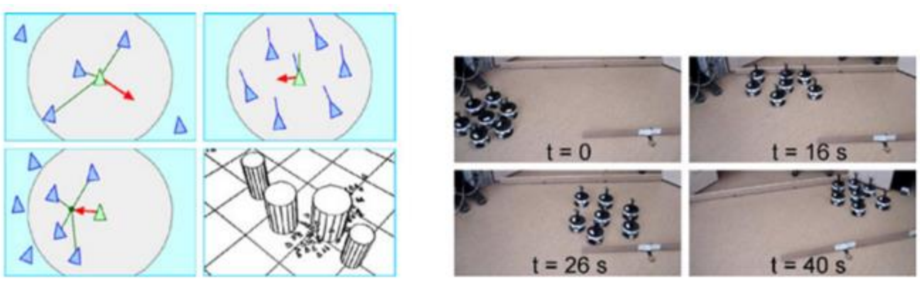


***Figure 1.8:** Examples of the coordinated motion behavior* **(Brambilla et al., 2013)**

3) **Collective transport:** The collective transportation of objects by a swarm of robots is an efficient method of moving heavier and larger objects that would not be possible for a single robot to move or bear.

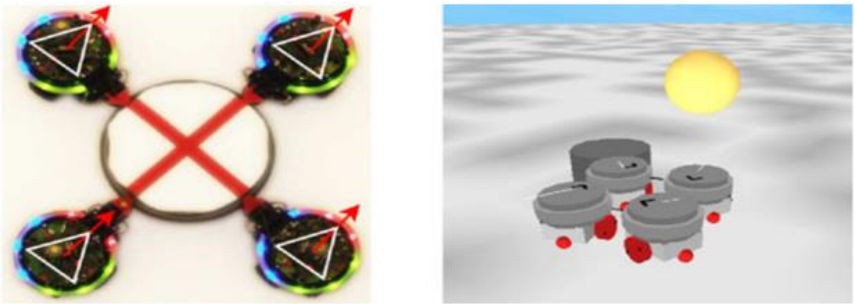


Figure 0.9: *Examples of the collective transport behavior* ***(Brambilla et al., 2013)***

4) **Collective localization:** The concept of collective localization refers to the mechanism by which a local coordinate system is established in order to allow robots in a particular swarm to locate their positions and orientations relative to each other.

3.4.3. Decision Making

These behaviors allow the robots in a swarm to take a common decision on a given issue.

1) **Consensus**: refers to the ability of swarm robots to agree on a single common choice among several alternatives.

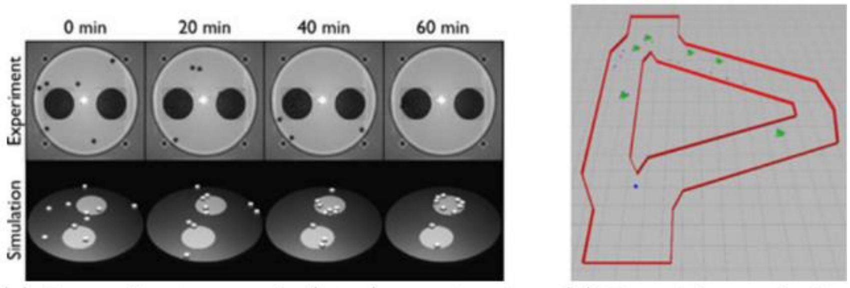


Figure 1.10: *Examples of the consensus behavior* ***(Brambilla et al., 2013)***

2) **Task allocation:** In the swarm, tasks are dynamically allocated to individual robots. This aims to maximize the performance of the entire swarm system. In the case of a heterogeneous swarm, the performance of the system can be further enhanced by distributing tasks in accordance with the robots' heterogeneity.

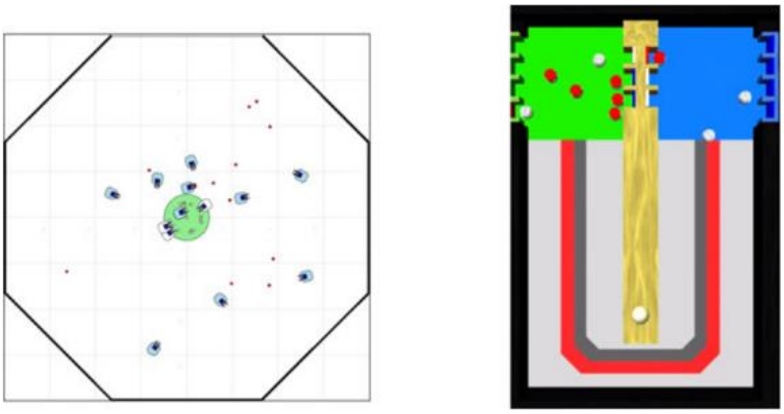


Figure 0.11: *Examples of the task-allocation behavior* ***(Brambilla et al., 2013)***

3) **Collective fault detection:** In a swarm of robots, collective fault detection determines the robots' deficiencies. In other words, collective fault detection decides which robots are deviant from the desired behavior of the swarm.
4) **Collective perception:** Through collective perception, robots in the swarm are able to combine the data they sense locally into a more comprehensive view. This process enables the swarm to make informed decisions collectively, such as the accurate classification of objects, the allocation of the appropriate fraction of robots to each task, and the determination of the ideal solution for a global problem.
5) **Synchronization:** Synchronization involves aligning the phase and frequency of the oscillators of the robots within the swarm. Thus, robots are able to perform tasks in a synchronous manner as a result of sharing a common understanding of time.
6) **Group size regulation:** this behavior is used for establishing groups of desired size. When the swarm size overrides the desired group size, the swarm will be divided into multiple groups.

3.4.4. Miscellaneous

1) **Self-healing:** this behavior refers to the swarm's ability to recover from errors caused by deficient robots in order to minimize the impact of robot failure on the rest of the swarm.

2) **Self-reproduction:** Swarms of robots can reproduce themselves by creating new robots or replicating patterns that have been created by many individuals. Self-reproduction is intended to enhance the autonomy of the swarm by eliminating the need for a human engineer to create new robots.

3) **Human-swarm interaction:** In the context of human-swarm interaction, humans can control robots in the swarm or receive information from them. Interaction can occur remotely, or proximally in a shared environment.

3. 5 Applications of swarm robotics

- **Agriculture:** In agriculture, swarm robotics plays a significant role. An example would be the SAGA platform. The latter is an experimental platform in precision farming where swarms of unmanned aerial vehicles (UAVs) monitor and weed the field as well as count and detect areas where there is a sufficient amount of weed [11].
- **Medical:** A wide range of treatments have been provided for almost all types of cancer. Although these treatments are effective, the possibility of side effects remains a serious concern. One of the major concerns is the attack on healthy cells. Swarming nanorobots are one such approach to treating cancer without harming healthy cells. Nanorobots can be designed to target only cancer cells. To identify cancer cells, they must have the ability to navigate their environment - the human body - and recognize specific objects that constitute cancer cells. Using nanorobots, cancer cells can either be found and injected with medications to kill them or drilled and broken without the use of medications. Whenever nanorobots move in swarms within a cancerous body, they can quickly eliminate cancer cells [30].
- **Hazardous Zones:** Snake robots are designed to fit through narrow passageways and tunnels inaccessible to humans or other machines. These metal-bodied robots have the ability to climb, swim, and crawl. A camera and light are also mounted on the snake robots so that the control center can monitor the entire area [11].

- **Military:** In order to perform various military tasks, Endeavor Robotics developed several robots. An example is the Cobra robots, which are designed to lift heavy objects that weigh up to 150 kilograms. Packbots are other military robots that are used to dispose of bombs. AlphaDogs, which have been developed by Boston Dynamics, are robots that resemble dogs and have the ability to transport heavy loads for soldiers. These robots are capable of walking up to 20 miles, carrying 180 kilograms. This type of robots has the intriguing property of being able to move in any direction without the need for any control. AlphaDogs follow their leader automatically, thanks to computer vision [11].

- **Household:** An automated cleaning system called "MAB" consists of miniature flying robots designed by the Colombian designer Adrian Perez Zapata. The flying robots are equipped with cleaning supplies, which can clean most of the surfaces in a house. Dust and dirt flakes can be transported and deposited on the surface. Direct communication between miniature robots is used to exchange environmental data [11].
- **Astronomy:** One of the interesting new fields for swarm robotic systems is Astronomy. Researchers are trying to find out how robots can help with the exploration of non-earth environments like Mars, the sun, and the moon. The “robotic telescope” concept was introduced in the early 1980s. As a result, the eSTAR project was developed for telescopic tasks, which consists of a multi-agent system that is considered as a collection of diverse robotic telescopes [11].

3. 6 Swarm robotics simulators

Simulators are an intermediate step between abstraction (such as mathematical models) and real validation, using physical robots. These are considered the most effective tools for analyzing and validating swarm robotic systems. In this section, we will list some simulators that help with swarm robotics validation **(Calderón-Arce et al., 2022)**:

1) **Stage:** Stage is a free C++ library that simulates a large number of mobile robots with a capacity of up to 100,000 agents. Code reuse, transparency, replication of experiments, and modification are some of the advantages of this simulator. There are several sensors and actuator models that can be simulated by Stage. Examples are infrared rangefinders, scanning laser rangefinders, color-blob tracking, fiducial tracking, bumpers, grippers, and mobile robot bases with global localization (odometric). In terms of exploration and foraging, Stage performs well, but it does not perform well when modeling trail-following or cooperative behaviors.
2) **Bio-PEPA:** As a high-level modeling language, Bio-PEPA is intended for the analysis of biochemical systems, enabling stochastic simulation, fluid flow analysis, and model checking. Moreover, its structure is capable of modeling scalable systems and space-time characteristics that are suitable for swarm robotics. Bio-PEPA, from this point, models specific behaviors of individual entities, such as robots, defining how they interact.
3) **TeamBots:** TeamBots is a 2D simulator for multi-agent mobile robotics research that is easy to use, free, and capable of running simulation codes on actual robots.

Prototyping simulations can be conducted with TeamBots using the same control systems that can be used on real mobile robots.

4) **Gazebo:** Gazebo is a 3D simulator designed for outdoor environments. The system generates realistic sensor feedback and physically consistent interactions between objects. As a result, the user has the option of selecting between multiple dynamics engines. Nevertheless, it is known for being slow when running with large populations. In a controlled environment, Gazebo has been used to compare algorithms for navigation and grasping.
5) **ARGoS:** As a modular and multi-engine simulation for heterogeneous swarm robotics, ARGoS allows for the assignment of different physics engines to different parts of the environment. A design robot may be equipped with controllers, sensors, actuators, physics engines and visualizations.
6) **USARSim:** USARSim is a free, open-source 3D simulator similar to Gazebo, Webots, and Microsoft Robotics Studio. This simulator may be used for educational and research purposes and was initially developed for the RoboCup Urban Search and Rescue Competition. USARSim can be used to model different application scenarios.

3. 7 Comparative Analysis: Swarm Robotics Versus Multi-Robot System

Within the confines of this particular subsection, we aim to explicate a comparative analysis between the utilization of swarm robotics and multi-robot system.

Swarm Robotics	Multi-Robot System
Variation in a great range of population size.	Small population size
Decentralized and autonomous control.	Usually centralized control
Homogeneous and heterogeneous entities.	Usually uses heterogeneous entities.
High flexibility.	Low flexibility.
High scalability.	Low scalability.
Unknown environment.	Unknown or known environment.
Each robot is unable to do anything meaningful.	Each robot can do some meaningful part of a task.
Complex task that is impossible for a single robot.	Simple (or complex) task that is possible for a single robot to do a part of it.

Table 1.1: *Comparison of SR and MRS (**Tan et Zheng, 2013**)*

4. Conclusion

In this chapter, we aim to provide a solid foundation for understanding the research domain and its key concepts. We provide an overview of the research domain by introducing the key concepts of swarm intelligence and swarm robotics. We begin by defining swarm intelligence and exploring some of its algorithms, along with highlighting the diverse range of applications within its domain. Subsequently, we delve into swarm robotics, starting with comprehensive definitions and an exploration of the characteristics that define this field. We also discuss the strengths of swarm robotics, highlighting its unique attributes and advantages. Additionally, we examine the fundamental behaviors exhibited by swarm robotics systems, emphasizing their collective and emergent properties. Moreover, we provide insights into the applications of swarm robotics across various domains. To facilitate experimentation and evaluation, we also discuss some commonly used simulators in the field of swarm robotic.

Chapter 2: Search and Rescue: state of the art

1. Introduction

In search and rescue operations, multi-robot systems have demonstrated promising outcomes by leveraging their ability to cover larger areas, collect more data, and enhance operational efficiency. These systems are deployed to perform tasks that are either perilous, time-consuming, or beyond the capabilities of human rescuers.

However, coordinating and communicating among the robots pose significant challenges in multi-robot search and rescue operations. Effective collaboration and information sharing are vital to ensure comprehensive coverage of the search area. Designing systems that enable seamless coordination and communication becomes crucial to maximize the effectiveness of the multi-robot team. Another critical factor is the ability of these systems to operate in environments with limited communication and navigation capabilities. Search and rescue scenarios often involve areas where communication infrastructure may be damaged or non-existent. Therefore, multi-robot systems need to be equipped with robust communication and search methods that can adapt to such challenging environments.

In this chapter, our primary focus is on defining the multi-robot search and rescue problem. We begin by providing a clear and concise definition of the problem, outlining its scope and key challenges. We proceed to summarize a selection of relevant and recent works that are closely related to the field of multi-robot search and rescue. These works represent the current state-of-the-art and highlight the various approaches and methodologies employed in tackling this problem. After that, we employ a table with predefined characteristics to compare and evaluate the reviewed works. After the comparisons, we engage in a thorough discussion of the results. We analyze and interpret the findings from the comparison, highlighting notable trends, patterns, and insights. Additionally, we explore the implications of the comparisons and identify potential areas for further research and improvement.

2. Swarm Robotics for Search and Rescue

The main purposes of using swarm robotics in search and rescue operations are the reduction search and rescue time and the safe transportation of survivors to the depots.

2. 1 Definition

Search and rescue operations play a critical role in disaster management, aiming to ensure the safety of individuals and minimize rescue time. While these operations can be challenging for human rescuers, mobile robots are utilized to navigate and explore disaster-stricken areas, locate targets, and perform various tasks that enhance the effectiveness and efficiency of rescue operations. They can access hazardous or inaccessible areas, gather information, and provide assistance to human responders.

However, we must consider the limitations and challenges of this technology, such as obstacles (robot energy, rough surface, inability to see due to rain, smoke, sandstorms, etc.)

2. 2 Related works

The use of SR in search and rescue operations can significantly enhance the efficiency and effectiveness of search and rescue efforts. In this Section, we start with summarizing some relevant and recent works related to the search and rescue problem. Then, we compare them qualitatively using predefined characteristics. Finally, we present a discussion that may help in producing new research ideas and orientations.

1) Multiple oriented robots for search and rescue operations (Firthous and Kumar, 2020)

Firthous and Kumar proposed in **(Firthous and Kumar, 2020)** a new algorithm called Artificial Bee Colony with Modified Evolutionary Programming (ABCMEP). This latter, is based on the combination of two algorithms Artificial Bee Colony (ABC) and Modified Evolutionary Programming (MEP). The authors used the ABC algorithm to avoid obstacles and the MEP algorithm to find the shortest and smoothest path (path optimization). The ABCMEP algorithm allows the robots that have been deployed in an unknown region to find targets by avoiding obstacles and reaching them efficiently.

The simulation of this algorithm was performed in MATLAB Ra2018 software, by creating different complex environments and setting different parameters (population, number of iterations) to test them for each algorithm (ABC, ABCEP, and ABCMEP). For each chosen path, the cost value is calculated based on the cost function, which uses the position of the bees in the field and takes the lowest value calculated based on the path that took the best cost. After each simulation, an optimal path is found relying on the respective algorithms (ABC, ABCEP, ABCMEP), and the best cost values are found after several iterations for each environment. According to this work, a comparison between the above three algorithms was presented and listed their best cost values for different environments, including the best cost seen in the proposed algorithm.

2) Comparative Analysis of Fruit Fly-Inspired Multi-Robot Cooperative Algorithm for Target Search and Rescue (Garg et al., 2022)

A search strategy is presented in **(Garg et al., 2022)** for faster victim search and real-time assessment and management of search and rescue operations. This strategy is named *Velocity-inspired Robotic Fruit Fly Algorithm (VRFA)*. The proposed strategy is based on: (1) the Fruit Fly Optimization algorithm (FOA) which helps the proposed system in searching and tracking multiple targets, and (2) the Particle Swarm Optimization (PSO) algorithm, which aims to update the position and velocity of fruit flies. While interacting with the systems, the independent robot performs four activities, including local computation:

1) Recording and processing data locally.
2) Sharing data between all robots.
3) Formulating a movement plan.
4) Generating timelines for executing the actual movement.

To evaluate this algorithm, other techniques such as PSO, FOA, and BSO were tested in parallel with the VRFA algorithm using two approaches (centralized and decentralized cooperation strategy) for static and dynamic targets. The authors studied the impact and effect of the number of targets, the complexity of the environment, and the number of robots on the system performance. The obtained results show that the performance is better when the robots work in decentralized cooperation with less search and rescue time and the best average is shown by the VRFA algorithm.

3) Leader-follower Behavior in Multi-agent Systems for Search and Rescue Based on PSO Approach (Gomez et al., 2022)

Authors in **(Gomez et al., 2022)** proposed a leader-follower behavior based on the PSO approach. This approach aims to guide a multi-robot group in search and rescue operations by a leader team that already knows the location of victims. This guided group loads the necessary supplies to provide them to the victims. The entire system has the ability to avoid collisions and obstacles, which considers non-navigable locations as static obstacles and other robots as dynamic obstacles. To address this multi-agent navigation problem, the authors proposed an objective function that dictates the navigation path. A cost function is calculated based on the interaction between the agents' repulsion and attraction forces.

In order to make sure that agents are kept at a safe distance from each other and to avoid collisions, the navigation cost function is described as a plane in which repulsion forces are expressed as Gaussian functions. The second part of the objective function is a negative Gaussian function, which acts as a global minimum. Lead agent distance and position are used to determine this function. In this work the PSO algorithm is used to minimize the objective function. Even though the system's behavior demonstrated the effectiveness of leader tracking, the position of the agents failed to take into consideration their size. This made their implementation impossible due to the collisions that occurred, which led the authors to resort to a repulsion algorithm.

To evaluate the proposed model, two simulations were performed. The first one is a visual representation of the mathematical model in Python, which offers the possibility to evaluate it without physical interaction and to observe its behavior. The second part of the simulation uses Pybullet as a development environment. It simulates physical robotic interaction using models of physical drone platforms. The result of this experimentation showed satisfactory performance.

4) Multi-robot cooperative rescue based on two-stage task allocation algorithm (Huang et al., 2022)

Huang et al. proposed in **(Huang et al., 2022)** a new algorithm for task allocation problems in a multi-robot cooperative rescue system based on the Ant Colony Contract Net Protocol (CNP). The authors used the ACO algorithm to determine the initial allocation results, and the CNP to adjust these results when the environment changes dynamically. The advantages of the proposed algorithm are parallel computation, time-efficient computation, and the ability to quickly adapt to changes in the environment.

Experiments were conducted on the scenario of a fire in an underground garage to rescue all the targets present in the garage and provide them with the necessary rescue equipment. The simulation was carried out using python 3.7 by setting some parameters ($size\ of\ the\ underground\ garage\ =\ 100 \times 100$, $number\ of\ robots\ =\ 3$, $number\ of\ targets\ =\ 21$, $robot\ speed\ =\ 1m/s$). To verify the performance of the proposed algorithm in a dynamic environment, the authors implemented three dynamic scenarios and tested them using the classical ACO algorithm and the proposed CNP algorithm using three performance metrics: *computation time (seconds), average task completion cost (m)* and *average task completion time (seconds)*.

According to the experimental results, the CNP algorithm gave satisfactory results compared to the ACO algorithm in terms of the used metrics, which led to revealing clear advantages in solving the multi-robot task assignment problem in a dynamic environment.

5) Research on Artificial Bee Colony Method Based Complete Coverage Path Planning Algorithm for Search and Rescue Robot (Yang et al., 2022)

In order to solve the problems of full-coverage path planning in search and rescue operations, which strive to completely cover the area of interest in a limited time and avoid obstacles autonomously in unknown areas, the authors in **(Yang et al., 2022)** proposed a new neural network algorithm through the ABC method. The neural network takes as input information about obstacles and coverage in the five directions and gives as output the speed of the left and right wheels. In the initial situation, the parameters are randomly generated, which makes the path planned by the neural network has a lower score.

According to the advantages of the ABC algorithm, such as the increased probability of finding the optimal solution and the robustness of the system, the authors used this method to optimize the parameters of the neural network and improve the training efficiency and effects of the entire system.

To analyze the performance of the algorithm, the authors define an evaluation function, which is divided into three parts: the coverage rate, the path repetition rate, and the failure rate. Based on experimental results, the ABC method, combined with a neural network path planning algorithm, can effectively control rescue robots to plan complete coverage paths effectively and can be migrated to various environments with high robustness.

6) Distributed Particle Swarm Optimization for Multi-Robot System in Search and Rescue Operations (Paez et al., 2021)

Authors in **(Paez et al., 2021)** proposed a new multi-robot exploration strategy for search and rescue operations, called distributed particle swarm optimization (DPSO). In their work, the authors focus on how a swarm of robots can navigate the search space avoiding collisions and obstacles and locating victims by modeling the problem by defining some general concepts listed below:

- Robots cannot cross static obstacles and other robots considered as dynamic obstacles.

- Victims are randomly scattered in the environment, which means they are in unknown locations for the robots.
- The robots can sense their local environment and acquire local information about obstacles, other robots, and victims.
- All robots can share their position with other robots working in the environment.

The authors proposed the DPSO algorithm to eliminate the drawbacks of the classical PSO algorithm, which are: (1) poor performance, (2) slower convergence speed, (3) trapping in a local optimum, (4) and the non-consideration of collision avoidance.

Two main functions are applied in the DPSO algorithm, the first is the artificial potential function, which attracts forces to an unknown area and victims, and the second is the repulsion forces function, which helps to avoid collisions. The repulsion function is divided into two sub-functions, namely intra and inter repulsion forces.

To perform the experiments, the authors implemented the multi-agent model and environment using Python and VRep. They used the Python environment to implement the proposed mathematical model and evaluate the performance of its features in different cases, the VRep to observe the behavior of the proposed model in 3D environments, and validate the decentralized multi-robot navigation using distributed multi-swarm systems in realistic disaster scenarios.

Four experiments were performed to evaluate the proposed algorithm. After analyzing the results of the experiments, the authors concluded that the DPSO algorithm helps the robots to escape from the local minimum and find alternative paths through disaster scenarios to reach an optimum.

7) AdverSAR: Adversarial Search and Rescue via Multi-Agent Reinforcement Learning (Rahman et al., 2022)

In **(Rahman et al., 2022)** authors describe an adversarial Multi-Agent Reinforcement Learning (MARL) approach to learning efficient coordination and collaboration strategies to achieve SAR mission objectives in the presence of adversarial communications. A Centralized Training approach with Decentralized Execution (CTDE) is used in this training approach. The authors presented some goals to be achieved in their system, such as coordinating the respective search to avoid visiting the same location multiple times and reducing the overall target exploration time by cooperative agents in the presence of adversaries.

The realization of the proposed algorithmic approach was divided into three main concepts described as follows:

- **Adversarial modeling**: aims to increase the time taken by cooperative agents to find missing assets.
- **Reward structure for optimal coverage**: aims to extend the MARL algorithm for a general optimal coverage problem and to encourage cooperative agents to explore the environment,
- **Training algorithm**: aims to train cooperative agents in the presence of interference from adversarial agents to provide a certain degree of robustness to the overall system.

To investigate how adversary interference can derail the performance of cooperative agents in a SAR mission and to what extent the proposed training algorithm can mitigate adversary actions, the authors conducted four case studies to evaluate the proposed model. According to the results of each case, the agents were able to locate the targets more successfully in case with the modified reward structure.

8) Area-Optimized UAV Swarm Network for Search and Rescue Operations (Ruetten et al., 2020)

The difficulties of search and rescue operations when the space is difficult to cover by human rescuers and ground robots, have led to the emergence of a new technology for use, namely unmanned aerial vehicles (UAVs). Authors in **(Ruetten et al., 2020)** proposed a self-organizing mesh network that is optimized for area coverage by maximizing the search area and maintaining wireless communication.

This new system lays the groundwork for the behavior of more than two unmanned aerial vehicles. Whenever a UAV is connected to more than one other UAV, it will attempt to remain within the range of all of them. Additionally, this system prevents UAVs belonging to the same swarm from colliding with one another. Rather than having a leading coordinator UAV, all of the UAVs adjust based on each other. The authors proposed a Mesh Reliance architecture, which works by assigning the swarm a connectivity matrix, with each UAV’s connections being a rep, and a swarm controller based on Genetic Algorithm (GA) and Neural Network (NN)

Based on the simulation results, the proposed methods are valid for organizing swarms. However, they are not as effective for swarm travel, and adding each drone did increase computation time.GA runs very slowly, making it a poor choice in emergencies.

9) Search and rescue with sparsely connected swarms (Dah-Achinanon et al., 2022)

A SAR algorithm based on ad-hoc networks accepting sporadic connectivity was proposed by **(Dah-Achinanon et al., 2022)**, to bridge the gap between theoretical approaches and practical applications. This work aims to provide autonomy, decentralization, and an effective research strategy for the system by ensuring communication links between the swarm members to inform others about target findings and share its location. A base station receives this information to rescue the targets. The search method is based on the belief space exploration in order to incorporate crucial priors from the authorities, such as the last known locations of the targets.

Following is a description of the principles of the above algorithm:

- Each drone executes its search for the target based on the available belief information
- The belief information, represented as a map, is updated and distributed to neighbors during the search to avoid searching the same area multiple times.
- The distribution of the belief map performed by the virtual stigmergy

To validate the presented algorithm and evaluate the necessary time required to find targets and obtain the final relay chain, the authors performed a series of tests in the ARGOS simulator platform and real-world experiments.

The results obtained during simulations and real-world experiments confirmed the feasibility of the proposed approach compared to a random walk.

10) Cooperative USV–UAV marine search and rescue with visual navigation and reinforcement learning-based control (Wang et al., 2023)

Naval operations such as marine search and rescue operations require a higher performance and efficient control to rescue survivors. The authors **(Wang et al., 2023)** presented new method for visual navigation and USV control.

The authors divided their work into two main parts summarized as follows:

- Deep learning-based visual navigation architecture for USV and floating object positioning in USV-UAV operations. This approach improves visual positioning accuracy and computational efficiency by introducing a soft-max spatial layer and a two-stage structure.

- Reinforcement learning-based USV control strategy approaches and encircles a floating object. The trained control policy can improve the performance of the control system under wave disturbances by adding observable wave features to the state space.

The proposed visual navigation architecture consists of two stages: the first consists of estimating the position of the USV and the floating object, and the second stage consists of estimating the heading angle of the USV. This two-stage architecture is built based on CNN and the network structure.

USV controller based on RL is developed for avoiding collisions with floating objects. The basic principle of RL is to maximize the accumulated reward at every step by iteratively optimizing a policy.

The authors conducted several experiments to evaluate the performance of the visual navigation model and the USV controller model by evaluating four network models (CNN-SS, CNN, FCN, YOLOv5) for the first proposed algorithm and designing three tasks to evaluate the second algorithm.

Therefore, the proposed visual navigation architecture can provide high-accuracy position, velocity, and handing angle estimations under most weather conditions. However, this latter still has some limitations in heavy foggy weather. Finally, after the simulation, the authors demonstrated the effectiveness and superiority of the proposed algorithms.

11) Ant colony optimization for path planning in search and rescue operations (Morin et al., 2022)

Authors in **(Morin et al., 2022)** developed a new algorithm called the Ant Search Path with Visibility algorithm (ASPV) for the NP-hard optimal search path problem with visibility. This algorithm draws inspiration from ACO principles, which define 96 variants based on four key components of traditional ACO: pheromone initialization, pheromone update, restart, and boosting.

To identify the best variant of the 96 proposed ACO algorithm variants, the authors conducted several experiments. In these experiments, three phases were conducted:

- Configuration phase, which determines the optimal parameter pairs for each variant.
- Multi-run evaluation (M-Eval) phase allows for determining how the performance of an ACO varies between runs on a given instance.

- The across-instance evaluation (A-Eval) phase provides a better understanding of an ACO's performance in a variety of instances.

The authors compared the performance of the best ASPV algorithm variants with that obtained through a Mixed-Integer Linear Program (MILP) and with a simple greedy heuristic. Results indicated that the proposed algorithm produced search paths with higher success probabilities in a shorter period.

12) Coverage path planning for maritime search and rescue using reinforcement learning (AI et al., 2021)

The maritime SAR environment is complex and time-varying, which led **(AI et al., 2021)** to present an autonomous coverage planning model for maritime SAR operations based on reinforcement learning. The contributions presented by the authors are summarized in below:

- A method for building a complex environment model for marine search and rescue is proposed.
- A reinforcement learning algorithm is proposed, which designs a multi-objective reward function that specifies the learning goal. This reward function consists of avoiding maritime obstacles, avoiding repeated paths, and preferentially searching for high-probability areas. It better identifies the good and bad actions of the vessel agent in the learning process, effectively solves the problem of sparse rewards and sub-goal conflicts, and reduces overlapping paths.
- An action selection policy is developed to balance exploitation and exploration to determine the global-optimal solution, which uses the sine function to improve the model's stability. In the later stage of the model calculation process, the probability of random action selection gradually decreases so that the model can converge stably.

Experiments are conducted in the simulation area to demonstrate the model's validity. A comparison of different trajectory planning algorithms is performed in several maritime SAR simulations. The proposed algorithm was tested in different scenarios and compared to the Boustrophedon Motion Algorithm (BM), the Boustrophedon Cell Decomposition Algorithm (BCD), and the Boustrophedon Motion Algorithm with A* Search (BA*).

The comparison of results indicates that the search path generated by the model can cover the entire SAR area safely, maintain a low repetition rate, and prioritize the high-probability areas for searching.

13) Multi-UAV Search and Rescue with Enhanced A∗ Algorithm Path Planning in 3D Environment

Authors in **(Du, 2023**) proposed a new extension of the A* algorithm for the 3D SAR environment. This algorithm is based on the A* and task allocation algorithms to obtain a faster and more efficient path-planning method. The objectives achieved by the author are summarized in below:

- General 2D and 3D trajectory planning of UAVs.
- Granting UAVs the ability to avoid obstacles when performing tasks.
- Identifying the shortest path for all UAVs in a minimum of time.

In this paper, the task allocation algorithm provides advantages to the system, such as a parallel search of the environment, allowing the drones to complete the entire task by distributing the assignments to each drone, helping to reduce the total route planning time and memory space of each drone. Moreover, the A* algorithm is generally used for the path planning optimization problem.

To evaluate the performance of the proposed algorithm, the author created a 2D and 3D simulation scenarios to test the functionality of the task allocation algorithm and performed experiments in a 3D environment to verify and evaluate the performance of the improved A* algorithm compared to the classical A* algorithm. According to the results obtained from the simulation it can be observed that the improved algorithm has better running time, speed, and efficiency than the classical algorithm and has higher efficiency of the coverage environment and reduces the amount of storage required.

14) Coverage path planning for multiple unmanned aerial vehicles in maritime search and rescue operations (Cho et al., 2021)

Because of the increase in the number of casualties in the maritime sector, maritime authorities and operation centers are trying to develop a quick search for survivors at sea using different technologies such as USV, UUV, and UAV. **(Cho et al., 2021)** introduced a new contribution by using a UAV swarm. This latter was divided into a two-phase method to solve the (CPP) problem. The first phase presents a methodology for transforming the search area into a graph consisting of vertices and edges using grid-based area decomposition. Hence, the proposed method takes into account the angle at which the area is decomposed to minimize the size of the coverage area.

The second phase focuses on determining the optimal path for multiple UAVs based on the results of the first phase to reduce the completion time. The authors developed a mixed-integer linear programming (MILP) model that minimizes the time required to cover the search area. As a constraint, the region (position of nodes, the angle between nodes, etc.) and the dynamics of the robot covering the nodes are considered.

The researchers in this paper observed that the proposed MILP model was time-consuming for large-scale real-world problems. To solve this problem, a new algorithm called Randomness Search Heuristic (RSH) was developed. This approach consists of three phases in each iteration:

1) Random construction: constructs the coverage path of UAVs by sequentially selecting the adjacent nodes.
2) Repair: repairs the generated roads if all nodes are uncovered to recover the feasibility.
3) Local search: improves the constructed path for each UAV to obtain high-quality solutions that are close to the optimal road.

Numerical experiments to validate the efficiency and effectiveness of the proposed approach and real-world experiments with two UAVs to verify the validity of the proposed algorithm in real-world were considered in this paper.

According to the results of the numerical experiments, the RSH algorithm can produce near-optimal solutions in a much shorter computational time than a commercial solver. In contrast to the real-world experiments, the mission execution time was longer due to the influence of wind or network communication uncertainty.

2. 3 Qualitative comparison of related work

After comprehensively reading, understanding, and summarizing the related works in the field of search and rescue operations with Multi-Robot Systems, the next step is to qualitatively compare them using several characteristics. This comparison allows us to identify the strengths and weaknesses of each work, providing valuable insights into their contributions and limitations.

Axes 1	***Axes 2***	***Axes 3***	***Axes 4***	***References***													
				(Firthous and Kumar,	(Garg et al., 2022)	(Gomez et al., 2022)	(Huang et al.,2022)	(Yang et al.,2022)	(Paez et al.,2021)	(Rahman et al.,2022)	(Ruetten et al.,2020)	(Dah-Achinanon et al.,	(Wang et al., 2023)	(Morin et al.,2022)	(AI et al., 2021)	(Du, 2023)	(Cho et al., 2021)
Environment	Search space	Complexity	Obstacles	X	X			X	X	X		X			X		
			Obstacles-free			X	X				X		X	X		X	X
	Sinks	Number	Single	-	X	X	X	-	X	-	-	X	X	-	-	X	-
			Multiple	-				-						-	-		-
		Position	Center	-				-						-	-		-
			Fixed	-	X	X	X	-	X	-	-	X	X	-	-	X	-
	Objects	Nature	Dynamic		X			-					X	X	-		-
			Static	X	X	X	X	-	X	X	-	X	X		-	X	-
		Position	Fixed	X			X	-							-		-
			Random		X			-	X	X	-	X	X	X	-	X	-
			Clustered			X		-							-		-
			uniform					-							-		-
Collective	Robot	Number	single	X				X							X		
			Multiple		X	X	X		X	X	X	X	X	X		X	X
		Initial location	Random		X	X			X	X	X	X	X	X		X	
			Fixed	X			X	X							X		X
			nest														
	composition	Architecture	Homogenous	X	X	X	X	X	X	X	X	X		X	X	X	X
			Heterogeneous										X				
Strategy	Control		Centralized		X	X											
			Decentralized	X	X		X	X	X	X	X	X	X	X	X	X	X
	Cooperative		Yes		X	X	X		X	X	X	X	X	X		X	
			No	X				X							X		X
	Communication		Direct	-	X	X	X	-	X		X	X	X	X	-	X	-

			indirect	-				-		X					-		-
	Exploration	Redundancy	Yes		X			X						X			
			No	X		X	X		X	X	X	X			X	X	X
		Type	ABCEMP	X													
			VRFA		X												
			CNP				X										
			MARL							X							
			ABC					X									
			ANN					X			N						
			belief-based search									X					
			GA								X						
			Leader-Follower			X											
			D-PSO						X								
			DL										X				
			RL										X		X		
			ASPV											X			
			Enhanced A∗													X	
			RSH														X
	Homing	Type	...														
Simulation	metrics	time															
	execution	R.W experiment										X					X
		Simulating	ARGOS									X					
			Netlogo		X												
			Python			X	X			X			X		X		X
			Pybullet			X		-	X		-						
			VRep						X								
			Matlab	X												X	
			C++											X			

Table 2.1: Qualitative comparison of the summarized related works

2. 4 Discussions of related works

Based on *Table 2.1*, we can group the research according to the scenarios used in the different SAR operations (Based-ground SAR, Maritime SAR, and Aerial SAR).

Scenarios	References
Based-ground	(Firthous and Kumar, 2020)

	(Huang et al., 2022)
	(Yang et al., 2022)
	(Paez et al., 2021)
	(Rahman et al., 2022)
	(Morin et al., 2022)
Aerial	(Garg et al., 2022)
	(Gomez et al., 2022)
	(Ruetten et al., 2020)
	(Dah-Achinanon et al., 2022)
	(Du, 2023)
Maritime	(Wang et al., 2023)
	(AI et al., 2021)
	(Cho et al., 2021)

2.4.1. Based-ground SAR

- The majority of the work used an obstacle environment.
- Some works used a single sink with a fixed position, the others did not specify the number of sinks and their positions.
- Most of the works have used static targets with a random position.
- The majority of the works used multiple robots with a random position.
- All research done in the based-ground SAR focuses on homogeneous robots and is based on decentralized control.
- Most research focuses on cooperation between robots using direct communication.
- The majority of studies take into account the lack of redundancy in space exploration and all of these studies use different search methods.
- In the SAR based-ground, all research experiments were conducted by simulation using different simulators

2.4.2. Aerial SAR

- The majority of the work used an obstacle environment.
- Most works used a single sink with a fixed position.
- Most of the works have used static targets with a random position unless in **(Garg et al., 2022)**, which use both dynamic and static targets.
- All articles used multiple robots with a random position.
- All the research done in aerial SAR focuses on homogeneous robots and is based on decentralized control, except for **(Gomez et al., 2022)**, which is based on centralized control.

- All research focuses on cooperation between robots using direct communication.
- The majority of studies take into account the lack of redundancy in space exploration and all of these studies use different search methods.
- In the airborne SAR domain, all research experiments were conducted by simulation using different simulators, except for **(Dah-Achinanon et al., 2022)**, who performed the experiments in both simulations with the ARGoS simulator and the real world.

2.4.3. Maritime SAR

- All works used an obstacle-free environment except for **(AI et al., 2021).**
- In maritime SAR, the research did not specify the number of sinks and their position except **(Wang et al., 2023)** used a single sink with a fixed position.
- **(Wang et al., 2023)** have set static and dynamic targets with a random position. The others did not specify the object's type and position.
- The majority of the works used multiple robots and setting fixed positions.
- All works done in the maritime SAR focus on homogeneous robots and are based on decentralized control except for **(Wang et al., 2023),** which used a heterogeneous system containing UAVs and USVs.
- Most research focuses on the absence of cooperation between robots.
- The majority of studies take into account the lack of redundancy in space exploration and all of these studies use different search methods.
- In the maritime SAR, all research experiments were conducted by simulation using different simulators, except for **(Cho et al., 2021),** who performed the experiments in both simulation and the real world.

As for all the works the sue of swarm intelligence based algorithms is very weak, and need to be investigated since these algorithms provide scalable, flexible and robust solutions to systems like search and rescue one.

3. Conclusion

In the aim to make a valuable contribution to the SAR problem, we conducted in this chapter a review of related works. We first started with a definition of the problem. Then, we presented a summary of the related works through an in-depth analysis of the proposed methods and contributions. After that, we qualitatively compared the reviewed related works through some important characteristics. Finally, and using the comparison table, we discussed and concluded some important points which helped us in proposing our contribution.

The next chapter, will present a new strategy of multi-robot search using a swarm intelligence-based algorithm.

Chapter 3 : Conception

1. Introduction

In this work, we are interested in modeling and solving the multi robot search and rescue problem in unknown environments. Since the SAR problem is a large scale one, the use of swarm intelligence algorithms is one of the best solutions. Based on the review and the comparisons presented in Chapter 3, we proposed a multi robot search and rescue problem. This latter, is inspired by the hunting behavior of penguins. It uses Penguin Search Optimization (PeSOA) and the Random Walk Algorithm (RWA) to search for survivors in catastrophic environments.

We start this chapter by describing the problem, objectives, and proposition. After that, we present the related algorithm (PeSOA and RWA). Then, we present the details of the proposed algorithm: the state machine, the formulations, and the pseudo codes of different states. We finish the chapter with a conclusion.

2. Problematic, Objectives, and proposition

Mobile robotics has applications in a wide range of fields, including SAR. Essentially, it involves navigating through catastrophic space to search for and transport victims to a central depot. The task consists of two basic behaviors: exploration and homing. A well-chosen exploration strategy can improve rescue performance in terms of people rescued or located, and time reduction. Swarm intelligence has been a source of inspiration for various applications of mobile robotics. For example, the exploration, return, and communication strategies of some animals have shown great effectiveness, especially in the field of mobile robot exploration.

In this work, we proposed a new SAR strategy based on SI algorithms to achieve the following objectives:

1) Increase the explored areas ;
2) Disperse robots widely in their environment to increase the number of found victims;
3) Reduce search time.

In the preceding chapter, our observations revealed that the majority of research in the SAR field predominantly relies on old SI algorithms (mature algorithms such as ACO, PSO, ABC…etc). Consequently, we aim to use in our contribution a SI-based algorithm that was not yet used in multi robot exploration and specifically in SAR problems. At the best of our knowledge, the PeSOA was used for optimizing mathematical benchmark function and not yet used in multi robot related problems. That is why, we adapted and used the PeSOA in SAR problem. In our proposition robots move collectively from an area to another following their best local position. They can change (migrate) their group based on the fluctuation of the surroundings groups. Whenever, a group of robots is in exploration of a rich region, it emits light to attract robots from other groups.

3. Analysis of Related Algorithms

In this Section, we present the related algorithms: The PeSOA, and the RWA algorithms. We first present the original version of the PeSOA. After that, we present its adaptation to the SAR problem.

3. 1. Penguin Search Optimization Algorithm

Penguins are seabirds and cannot fly due to their aquatic adaptation. Their wings are ideal for swimming and can be considered fins. The behavioral ecology of penguin foraging has many similarities to modern metaheuristics. As a result of this nature's intelligence, the PeSOA algorithm was developed for the first time by **(Gheraibia and Moussaoui, 2012)**. Subsequently, this algorithm was further elucidated by **(Gheraibia et al., 2018)**.

Figure 3.1: *Hunting behavior of penguins [13]*

The authors summarized their algorithm with simple rules described as follows:

- A penguin population consists of several groups. Each group contains a number that varies according to the food availability in the corresponding foraging region.

- Each group of penguins starts foraging in a specific depth under the water according to the information about the energy gain and the cost to obtain it.
- They feed as a team and follow the local guide who fed the most on the last dive. The penguins scan the water for food until their oxygen reserves are exhausted.
- After a number of dives, the penguins come to the surface to share the location and abundance of food sources with their local affiliates through intra-group communication.
- If food resources are insufficient for the penguins in a given group to live on, part of the group (or the entire group) migrates to another location through inter-group communication.

Penguin hunting behaviors	**Optimization heuristic principles**
The sea	Solution space
Most abundant shoals of fish	Global optimum
Penguin position	A candidate solution
Energy content of prey	Fitness of solution
Oxygen reserve	Acceleration coefficient
Penguin swimming	Solution update
Intra-group communication	Intensification search
Inter-group communication	Diversification search

Table 3.1*: Metaphors of penguin hunting behaviors for optimization heuristic principles* (**Gheraibia et al., 2018**).

Parametric formulas:

In this section, we will define the basic expressions of the PeSOA algorithm **(Gheraibia et al., 2018):**

- the movement of the Penguin is done according to the following expression:

$$x_j^i(t+1) = x_j^i(t) + O_j^i(t) \times rand() \times \left|x_{Localbest}^i - x_j^i(t)\right| \qquad (1)$$

- After each dive, the penguin's oxygen reserve is updated as follows:

$$O_j^i(t+1) = O_j^i(t) + \left(f(x_j^i(t+1)) - f(x_j^i(t))\right) \times \left\|x_j^i(t+1) - x_j^i(t)\right\| \qquad (2)$$

- The food abundance degree can be estimated by the Quantity of Eaten Fish (QEF), which is expressed as follows:

$$QEF^i(t+1) = QEF^i(t) + \sum_{i=1}^{di} (O_j^i(t+1) - O_j^i(t)) \tag{3}$$

- The penguin updates its group membership by referring to a function related to the degree of food abundance of the different groups. Thus, the penguin joins a group with a probability proportional to the QEF of the corresponding group given below:

$$\mathrm{P_i}(\mathrm{t}+1) = \frac{QEF^i(t)}{\sum_{I=1}^{K} QEF^j(t)} \tag{4}$$

Notations	**Descriptions**
N	Number of total penguins
K	Number of groups
f	Objective function of the problem
$O_j^i(t)$	The oxygen reserve of the **j^th^** penguin of the **i^th^** group
$x_j^i(t)$	The position of penguin j allocated to the **i^th^** group at **t^th^** instance
$x_{Localbest}^i$	The best solution found by the **i^th^** group
$QEF^i(t)$	Quantity of eaten fish of the **i^th^** group at the **t^th^** instance
$\mathrm{P_i}(\mathrm{t})$	Probability of existence of fish of **g^th^** group at **t^th^** instance
$rand()$	A random number drawing from (0, 1)

Table 3.2: Notation description (PeSOA) (Gheraibia et al., 2018)

3. 2. Random Walk Algorithm

The random walk algorithm is a mathematical concept used in various fields. It simulates the movement of a particle or entity in a series of random steps within a given space. At each step, the particle randomly chooses a direction and magnitude for its movement. The algorithm is based on the principle that the cumulative effect of numerous random steps can lead to interesting patterns and behavior. Random walk algorithms have been applied to diverse problems, such as modeling diffusion processes, generating random numbers, optimizing search strategies, and simulating financial market movements (**Du et al., 2003**).

4. Proposed algorithm: Modified Penguin Search Optimization Algorithm (MPeSOA)

In this Section, we present the proposed SAR algorithm named: The Modified Penguin Search Optimization Algorithm (MPeSOA). First, we present the state machine showing the behavior of our robots with a textual description of each behavior. After that, we introduce the parametric structure of MPeSOA. Finally, we present the pseudo code of the different behaviors.

4.1. Robots State machine

During its life cycle, the robot can be in one of the four behaviors: *Global Serach*, *Local Serach*, *Migrayion* and *Obstacle Avoidance*. The robots start all from the central depot and change their behaviors according to input data gathered by their sensors. The state machine representing the bahvior of the robots is given by Figure 3.1.

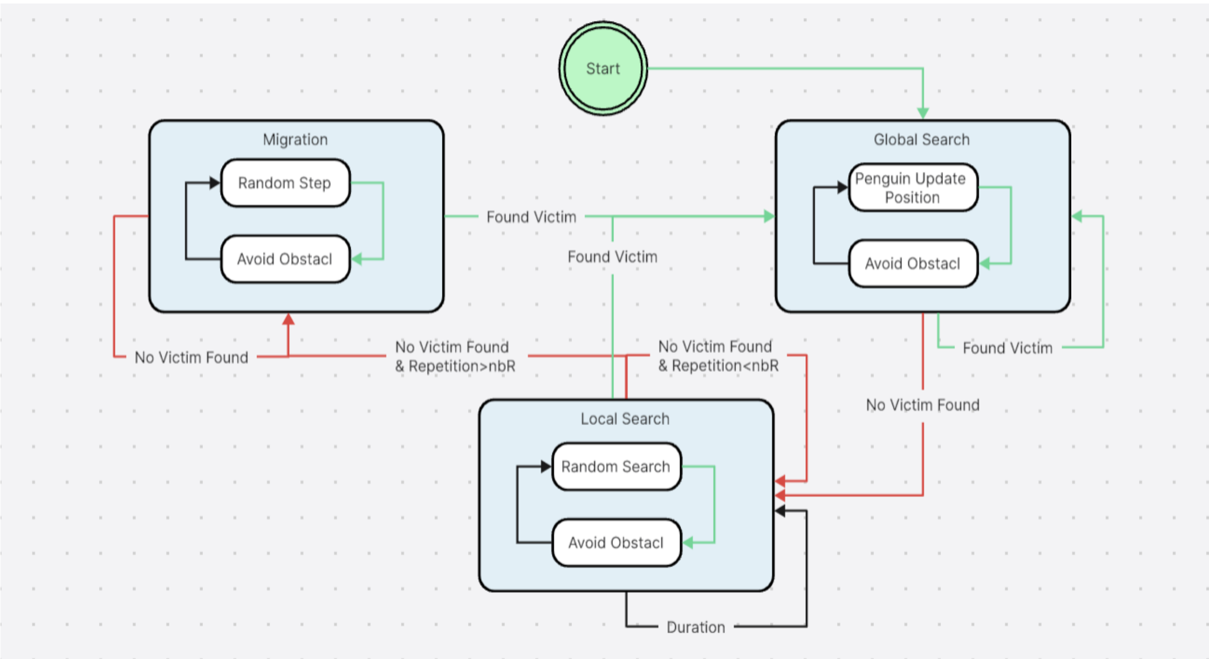


***Figure 3.2**: MPeSOA's State Machine*

In below, we explain the different behaviors of the state machine:

1) **Global search:** During this phase, each group of robots undergoes a relocation process to a new position, guided by the LocalBest solution obtained from the previous dive. This relocation is executed by employing the penguin search position equation as depicted in *Equation 5* (Section 3.1). Throughout the search process, individual robots may encounter obstacles or victims, prompting distinct decision-making. For instance,

when faced with barriers, a robot must navigate around them to avoid any obstacles. Conversely, upon discovering victims, the robot must prioritize their rescue and subsequent safety. Once the group reaches a new position, all group members must shift their state to the Local Search state.

2) **Local Search:** Before starting the local search, the robots must patiently await the arrival of all group members, ensuring the collective initiation of the search procedure employing the Random Walk Algorithm (RWA). As in the *Global search*, the robots have to get around obstacles and effectively rescue victims encountered along the way. However, the key distinction lies in the search duration. In this case, the robots apply the RWA for a predefined period and subsequently repeat this process in the absence of victim discoveries, thereby providing additional opportunities to locate victims. In instances where no victims are detected, the robots must undergo migration by reverting to the *Migration state*. Conversely, upon successful victim detection, the group updates their local best and transitions to the *Global search state*.

3) **Migration State:** When the group fails to detect or rescue victims, it becomes imperative for them to migrate and join another group that has achieved notable success in locating victims. As a result, the group will divide into two sub-groups, each following the RWA for movement. It is important to note that the sub-groups will take opposite directions, ensuring that all members within a specific sub-group share the same step length and move coherently in a consistent orientation.

4) **Obstacle avoidance:** Our explorer robot is equipped with *24 IR proximity sensors* positioned strategically around its structure. These sensors possess detect obstacles within a proximity range of *30 cm*. In accordance with the readings obtained from these sensors, if the robot identifies the presence of an obstacle, it retrieves the angle information and adjusts its movement accordingly. When encountering a negative angle, the robot executes a right turn, whereas a positive angle prompts a left turn.

4.2. Parametric structure

Throughout the subsequent rules, the oxygen reserve *O(t)* will be treated as a constant value set to 1:

1) **Global search:** The Penguin Update Position method, as expressed by *Equation (5),* is used in the MPeSOA *Global search*. This method is specifically employed when the robots change their position.

$$x_j^i(t+1) = x_j^i(t) + rand() \times \left| x_{Localbest}^i - x_j^i(t) \right| \tag{5}$$

2) **Local search:** In local search, we employ the RWA defined by the E*quation (6)*. This search methodology is implemented when the robots explore within the current subspace. The RWA is represented as

$$sl_i = rand(a, b) \tag{6}$$

3) **Group migration:** If victims remain undetected, the group will initiate migration and merge with another group. Before the migration process, it is necessary to compute the probabilities of the existence of fish for all groups using *Equation (4)*. However, if all computed probabilities are found to be less than ***0.5***, the current group will refrain from migrating and takes a random *LocalBest*. Conversely, if at least one probability exceeds or equals ***0.5***, the group will initiate migration based on the aforementioned approach outlined in the Migration State. It is important to highlight that the equation governing the update of the Quantity of Eating Fish will be as follows:

$$QEF^i(t) = \sum_{j=1}^{di} E_j^i(t) \tag{7}$$

Notations	**Descriptions**
$E_j^i(t)$	Quantity of eaten fish of the $\mathbf{j^{th}}$ robot of $\mathbf{i^{th}}$group at the $\mathbf{t^{th}}$ instance
sl_i	Step length for $\mathbf{i^{th}}$ robot
$rand(a, b)$	A random number drawing from (a, b)

Tableau 3.3: *hunting ecology of penguins*

4.3. Pseudocodes

In this subsection, we provide an overview of the pseudocodes required for the development of our proposed algorithm. We have formulated the MPeSOA algorithm, presented in Algorithm 1 pseudocode, which encompasses four key functions: *Global_search (Algorithm 5)*, *Local_search (Algorithm 2)*, *Obstacle_avoidance (Algorithm 3)*, and *Migration (Algorithm 4)*.

Algorithm 1 : MPeSOA

```
Initialize the number of groups
```

```
Initialize the total number of robots
Generate a random local best for each group
Distribute randomly the robots in groups
while stopping criterion is not reached do
for each group i do
  go to Global_Search
  Update the food abundance degree for this group by equation (7)
end for
Update the totale eating food
Update the global best solution
Update membership function values for each group by equation (4)
end while
```

Algorithm 2 : Local_search

```
while (repetition_process > 0) do
while (the duration has not expired) do
While(∄ obstacle or victim or pheromone)do
  Random walk by using equation (5)
end while
if(∃ obstacle) then go to Obstacle-avoidance
else (∃ victim) then
  Get the victim
  Update the number of victim rescued
  Update the quantity of eating fish for this goup
  Update duration for this goup
end if
end while
Update LocalBest for this goup
if(new localbest == the last localbest)
  repetition_process—
  If (repetition_process==0)
    go to Migration
else
  go to Global_Search
end while
```

Algorithm 3 : Obstacle_avoidance

```
Get readings from all proximity sensors
Accumulate all readings and get the accumulated value and the
angle
if(accumulated value > 0) then
  if(accumulated angle > 0) then turn left
```

```
  else turn right
end if
```

Algorithm 4 : Migration

```
return to the last position
calc Pi using equation (4)
if (all probabilities <0.5)
  set a random localbest
  go to Global_Search
else
  divide the group into two subgroups
  Give the sub groups an opposite directions
  While(∄ obstacle or victim or pheromone)do
    Use the same random step for all members within the same group until they reach the hunting group
  end while
  if(∃ obstacle) then go to Obstacle-avoidance
  else (∃ victim) then
    Get the victim
    Update the number of victim rescued
    Update LocalBest for this goup
    Update the quantity of eating fish for this goup
    go to Global_Search
```

Algorithm 5 : Global_Search

```
calculate new position using the equation (5)
while(the group does not reach the new position) do
  if(∃ obstacle) then go to Obstacle-avoidance
  else (∃ victim) then
    Get the victim
    Update the number of victim rescued
    Wait for few time before update LocalBest for this group
    if (the time is expired)
      Update LocalBest for this goup
    Update the quantity of eating fish for this goup
    go to Global_Search
End while
if (the group reached the new position)
  go to Local_Search
```

5. Conclusion

We presented in this chapter our contribution to the resolution of SAR problem. We proposed a multi robot search and rescue algorithm based on the hunting behavior of penguins and the random walks. Each robot passes through different states according to its surrounding events. We described in detail the different behaviors and the state machine of our rescuing robots. We also presented the equations used as adaptation of the penguin algorithm and we presented the pseudo code of the different states.

Our system meticulously delineates each step required to construct a highly efficient model. Subsequently, the next chapter will elucidate the implementation particulars of our application

Chapter 4 : Implementation and Results

1. Introduction

In order to validate the proposed algorithm (MPeSOA) discussed in the previous chapter, we implemented it using the ARGoS mobile robotics simulator. Subsequently, we conducted simulations encompassing various environmental configurations to evaluate the performance of our proposed algorithm.

This chapter begins with an introduction to the ARGoS simulator, followed by an overview of the characteristics exhibited by the robots employed in our study. Subsequently, we propose a series of simulation scenarios designed to evaluate the performance of the proposed algorithm. Furthermore, we present and analyze the outcomes attained through the application of these scenarios.

2. Development environment

ARGoS (Autonomous Robots Go Swarming) (**Pinciroli, 2012**) is a simulator created by Carlo Pinciroli as part of the Swarmanoid project. This simulator is based on new design principles, the main purpose of which is to achieve both flexibility and efficiency. The attainment of flexibility in ARGoS is facilitated through the implementation of a modular framework. Each fundamental architectural element is realized as a software module that possesses the ability to be dynamically selected during runtime, known as plug-ins. These plug-ins encompass diverse functionalities, including robot control code, sensors, actuators, physical engines, visualizations, and simulated entities such as robots and other objects. Efficiency in ARGoS is accomplished through the utilization of a parallel architecture that capitalizes on the capabilities of multi-core processors while ensuring the calculation of only the essential components. ARGoS is an openly accessible simulator, implemented using the programming languages of C++ and Lua and other frameworks.

3. ARGoS Modular feature

The ARGoS simulator has three main features. In this section, we will focus only on the modular structure. This structure is designed to enhance code reuse and facilitate customization, allowing users to choose which modules have to use for their experiments, bringing accuracy and rapidity to the experience. *Figure 4.1* shows the ARGoS architecture.

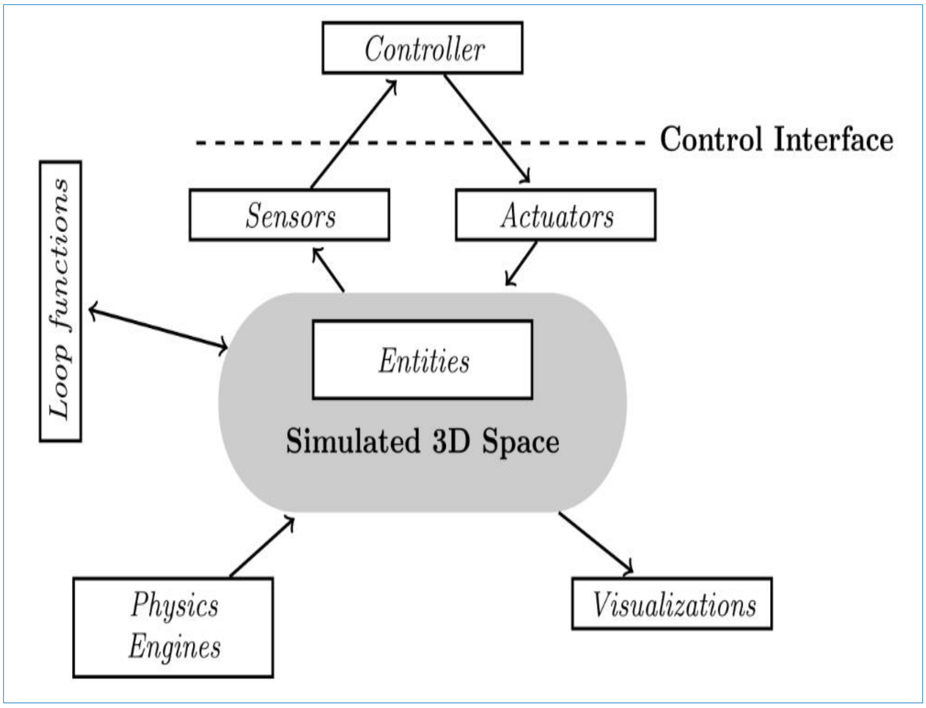


Figure 4.1: *ARGoS Architecture (**Pinciroli, 2012**)*

1) **The simulated 3D space:** the 3D simulation space consists of a collection of data structures containing the complete state of the simulation. Object position and orientation are part of the state information. In addition, the object state can be composed of specially equipped devices. All this information is stored in space. The development team, therefore, offers basic entities to organize this data, enabling users to customize them or add new ones as required.

2) **Sensors and Actuators:** Sensors and actuators are plug-ins that access the state of the simulated 3D space. Sensors are granted exclusive read-only access to the simulated 3D space, allowing them to retrieve relevant information. In contrast, actuators possess the capability to modify specific aspects of the state information contained within the simulated 3D space.

3) **Physics engines:** The physics engine is responsible for updating the embodied entities that occupy their assigned portion of space. By adopting this design approach,

ARGoS enables the simultaneous execution of multiple engines of varying types in parallel throughout an experiment.

4) **Visualizations:** Visualizations are also plug-ins that operate by accessing the state information of the simulated 3D space and generating an output that visually represents the simulation.

5) **Controllers:** The robot controller is also plugins that contain the logical control of robot behavior within an experiment.

6) **Loop Functions:** Loop functions are considered user-defined function hooks, which allow you to customize the initialization and end of an experiment and add personalized functionality executed before and (or) after each simulation step. Loop functions allow access and modify the entire simulation, including adding, moving, or deleting entities in the environment and other functionalities.

4. System components modeling

Within our system, there exists five principal components that collectively contribute to its functionality: the environment, robots, victims, obstacles, and light. In the following of this section, we elucidate the proposed modeling approach for each of these components:

- The environment is modeled by a continuous 3D space.
- The used robots are Foot-bots (a detailed explanation of its characteristics is given in below).
- The light zone of a robot is represented by a faded yellow circle.
- Victims are represented by a gray 3D cylinder *5cm X 6.5cm* in size.
- Obstacles are represented by 3D gray boxes of *150cm X 150cm*.

4.1. Foot-bot characteristics

The foot-bot represents a specific arrangement of modules built upon the marXbot robotic platform. This configuration encompasses several key elements, including a top CPU and vision module, a distance scanner, a range and bearing module, a self-assembling module, as well as a fundamental mobility and battery module. This robot has a compact size: *17 cm diameter x 29 cm, 1.8 kg* and two wheels.

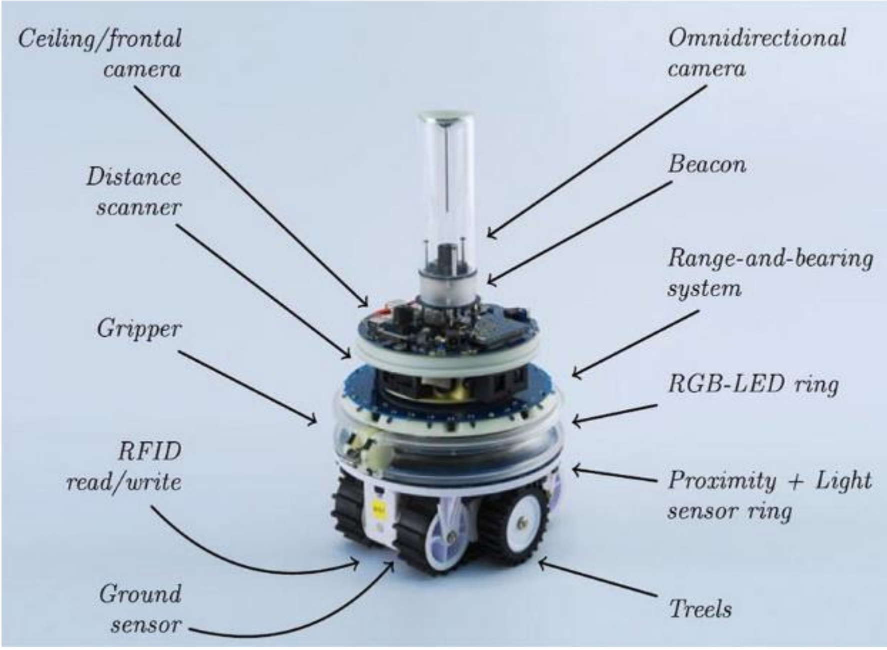


Figure 4.1: *Foot-bot components (**Pinciroli, 2012**)*

In our simulation, the foot-bot incorporates the following features:

- 24 IR Proximity sensors with detection range = 0.3m.
- 24 Light sensors with detection range = 0.3m.
- 13 Led actuators.
- A constant velocity V= 50 cm/s.

4.2. Controller analysis

1) **Navigation:** The Foot-bot maneuvers within the environment by utilizing a wheel pair for locomotion. In the control code, we employ the "setLinearVelocity(l, r)" procedure to drive the robots, where "l" and "r" represent the respective speeds of the left and right wheels. To detect obstacles and victims in the proximity of the robots, we employ proximity sensors. Each sensor possesses a range of 0.3 meters and provides a reading comprising an angle in radians and a value within the interval [0, 1]. The value is contingent upon the distance between the sensor and the detected object, where a value of 0 indicates no detection and the value exponentially increases to 1 as the object approaches closer. The angle measurement is determined between the local X-axis and the position of the sensor on the robot body. For the detection of light sources, we utilize light sensors. These sensors also yield a value ranging between 0 and 1, with 0 denoting the absence of any light within range, while 1 signifies saturation due to the perceived

light. The values between 0 and 1 are influenced by the distance from the perceived light source. Additionally, the light sensors provide an angle measurement in radians.

2) **Obstacle avoidance:** During each step of the simulation, the robot's proximity sensors gather information regarding nearby objects. Following this, the robot analyzes the sensor values and calculates the sum of all the reading values. This computation enables the robot to determine the distance between itself and the detected obstacle. When the robot detects an obstacle (value > 0), it evaluates whether it can proceed forward safely or if it needs to turn in the opposite direction of the reading angle.

5. Simulations and analysis of results

Finding the right parameters for the algorithm will certainly increase its efficiency. In our proposed algorithm, we can manipulate parameters to improve simulation results (performance). The parameters chosen are the number of robots, the number of groups, the number of clusters, and the environment size. In addition, we used as performance criterion the number of collected victims in a fixed time (*20 minutes).*

5.1. Simulation scenarios

Within this subsection, we provide an overview of four simulation scenarios, each corresponding to one of the aforementioned parameters. We test by these scenarios the influence of robots' number, group numbers, clusters number and environment size on the performances of the proposed strategy. *Table 4.1* shows the different proposed scenarios:

Scenario 1: The influence of robot numbers	**Scenario 2: The influence of group numbers**
Number of robots: 30, 40, 50, 60. Target distribution: Clusters. Number of groups: 6. Number of clusters: 12. Number of victims in environment: 910. Obstacle density: 10% of environment size environment size. Environment size: 120m X 120m.	Number of robots: 50. Target distribution: Clusters. Number of groups: 3, 5, 6, 8. Number of clusters: 12. Number of victims in environment: 910. Obstacle density: 10% of environment size environment size. Environment size: 120m X 120m.
Scenario 3: The influence of cluster numbers	**Scenario 4: The influence of environment size**

Number of robots: 50. Target distribution: Clusters. Number of groups: 6. Number of clusters: 2, 4, 8, 16. Number of victims in environment: 1000. Obstacle density: 10% of environment size Environnement size: 120m X 120m.	Number of robots: 50. Target distribution: Clusters. Number of groups: 6. Number of clusters: 12. Number of victims in environment: 910. Obstacle density: 10% of environment size Environment size: 80^2 m^2, 100^2 m^2, 150^2 m^2, 190^2 m^2.

Table 4.1: *The proposed simulation scenarios*

5.2. Performance criteria

As delineated in the preceding chapter, this research study is guided by two primary objectives:

1. To maximize the successful rescue of victims within the specified timeframe.
2. To minimize the overall duration of the rescue operations.

To assess the efficacy of our proposed solution, we will adopt specific performance criteria centered around the number of victims successfully rescued within a predetermined timeframe of 20 minutes.

5. 3. Results, and comparisons

The performance evaluation of the PeSOA algorithm has been conducted using the simulation scenarios presented in the preceding table. In the subsequent section, we introduce and analyze the results obtained by applying this algorithm across diverse scenarios. It is important to note that the reported results represent the average outcome from five simulations.

1) Scenario 1

With this scenario, we intend to study the impact of the number of robots on the performances of the MPeSOA algorithm. With increasing the number of robots the total number of found victims increases accordingly. **Table 5.4** shows the results obtained by this scenario.

	30	40	50	60
MPeSOA	360	387	434	590

Table 4.2 : *Influence of robot number on performance.*

2) Scenario 2

In this scenario, our investigation focuses on evaluating the impact of varying the number of groups, specifically ranging from 3 to 7, on the performance of our algorithm. The increase in the number of groups increases accordingly the number of found victims. **Table 5.5** shows the results obtained by this scenario.

	3	4	5	7
MPeSOA	350	433	586	659

Table 4.3: *Influence of group numbers on performance.*

3) Scenario 3

We investigate in this scenario the influence of the number of clusters, spanning from 2 to 16, on the performance of the proposed algorithm. The number of found victims decreases accordingly with the increase in the number of clusters. This is due to the density of victims in the clusters.

	2	4	8	16
MPeSOA	1000	748	603	572

Table 4.3 *: Influence of cluster numbers on performance.*

4) Scenario 4

In this scenario, we examine the impact of varying the environment size on the performance of the MPeSOA. Each simulation involves an incremental increase in the environment size, ranging from 80^2m to 200^2m. With increasing the size of the environment, the number of found victims decreases. This decrease is related to the large size of the environment and the number of obstacles. Table 5.7 shows the obtained results.

	80^2 m^2	100^2 m^2	150^2 m^2	190^2 m^2
MPeSOA	763	734	636	353

Table 4.4 : *Influence of environment size on performance.*

6. Conclusion

In this chapter, we presented the ARGoS platform as the framework used to implement our algorithm and its corresponding architecture. Additionally, we provided an overview of the

Foot-bot robots and their distinctive features. Furthermore, we conducted a comprehensive analysis of our algorithm's performance by proposing a series of scenarios featuring variations in specific parameters to ascertain their impact on the simulation performance. The results obtained from each simulated scenario consistently demonstrate the performances of the proposed algorithm.

General conclusion

The utilization of Swarm Intelligence (SI) algorithms for collaborative problem solving has gained significant popularity within the field of mobile robotics. These algorithms have found application in diverse problem domains, such as foraging, cleaning tasks, and search and rescue operations. One of the fundamental aspects of these applications is the exploration of the environment, where swarm of robots navigate through space in accordance with a predetermined deployment strategy, with the objective of locating valuable targets.

In the context of this study, our focus lies in the dispersion of robot groups achieved through an implicit division of the environment based on the positions of each group. To address this, we propose a swarm robot search and rescue algorithm, MPeSOA (Modified Penguin Search Optimization Algorithm). The algorithm has been successfully implemented within the ARGoS simulation platform, and the simulation results have demonstrated that our proposed algorithm is capable of producing satisfactory outcomes, even when exposed to the influence of the random walk strategy.

Moving forward, we envision several perspectives for further enhancing this work. Firstly, we aim to extend the robot's behavior to incorporate considerations for energy limitations. Secondly, we aspire to develop a more efficient local search strategy that surpasses the randomness of the current approach. Lastly, we aim to implement mechanisms for implicitly saving search spaces, thereby preventing redundant exploration of previously examined areas and consequently reducing the overall search time.

Bibliography

(Ahmed et Glasgow, 2012) : Ahmed, H., & Glasgow, J. (2012). Swarm intelligence: concepts, models and applications. *School Of Computing, Queens University Technical Report*.

(Ai et al., 2021) : Ai, B., Jia, M., Xu, H., Xu, J., Wen, Z., Li, B., & Zhang, D. (2021). Coverage path planning for maritime search and rescue using reinforcement learning. *Ocean Engineering*, *241*, 110098.

(Beni et Wang, 1993) : Beni, G., & Wang, J. (1993). Swarm intelligence in cellular robotic systems. In *Robots and biological systems: towards a new bionics?* (pp. 703-712). Springer, Berlin, Heidelberg.

(Brambilla et al., 2013) : Brambilla, M., Ferrante, E., Birattari, M., & Dorigo, M. (2013). Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, *7*(1), 1-41.

(Calderón-Arce et al., 2022) : Calderón-Arce, C., Brenes-Torres, J. C., & Solis-Ortega, R. (2022). Swarm Robotics: Simulators, Platforms and Applications Review. *Computation*, *10*(6), 80.

(Cheraghi et al., 2021) : Cheraghi, A. R., Shahzad, S., & Graffi, K. (2021, September). Past, present, and future of swarm robotics. In *Proceedings of SAI Intelligent Systems Conference* (pp. 190-233). Springer, Cham.

(Cho et al., 2021): Cho, S. W., Park, H. J., Lee, H., Shim, D. H., & Kim, S. Y. (2021). Coverage path planning for multiple unmanned aerial vehicles in maritime search and rescue operations. Computers & Industrial Engineering, 161, 107612.

(Colorni et al.,1991) : A. Colorni, M. Dorigo et V. Maniezzo, *Distributed Optimization by Ant Colonies*, actes de la première conférence européenne sur la vie artificielle, Paris, France, Elsevier Publishing, 134-142, 1991.

(Dah-Achinanon et al., 2023) : Dah-Achinanon, U., Marjani Bajestani, S. E., Lajoie, P. Y., & Beltrame, G. (2023). Search and rescue with sparsely connected swarms. *Autonomous Robots*, 1-15.

(Demazeau et Müller, 1990) : Demazeau, Y., & Müller, J. P. (Eds.). (1990). *Decentralized Ai*. Elsevier.

(Dorigo et Gambardella, 1997) : Dorigo, M., & Gambardella, L. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans. evolutionary computation, 1(1), 53-66.

(Du et al., 2003) : Du, J., Li, H., Xu, X., Shi, M., Wu, J., Zhou, X., & Han, R. (2003). Experimental implementation of the quantum random-walk algorithm. Physical Review A, 67(4), 042316.

(Du, 2023) : Du, Y. (2023). Multi-UAV Search and Rescue with Enhanced A Algorithm Path Planning in 3D Environment. *International Journal of Aerospace Engineering*, *2023*.

(Ekelhof et Giacomo, 2020) : Ekelhof, M., & Giacomo P. P., Swarm Robotics: Technical and Operational Overview of the Next Generation of Autonomous Systems, Geneva, Switzerland: United Nations Institute for Disarmament Research, 2020.

(Firthous et Kumar, 2020) : Firthous, M. A. A., & Kumar, R. (2020, August). Multiple oriented robots for search and rescue operations. In *IOP Conference Series: Materials Science and Engineering* (Vol. 912, No. 3, p. 032023). IOP Publishing.

(Garg et al., 2022) : Garg, V., Tiwari, R., & Shukla, A. (2022, June). Comparative Analysis of Fruit Fly-Inspired Multi-Robot Cooperative Algorithm for Target Search and Rescue. In *2022 IEEE World Conference on Applied Intelligence and Computing (AIC)* (pp. 444-450). IEEE.

(Gheraibia et al., 2018) : Gheraibia, Y., Moussaoui, A., Yin, P. Y., Papadopoulos, Y., & Maazouzi, S. (2018). PeSOA: penguins search optimisation algorithm for global optimisation problems. *arXiv preprint arXiv:1809.09895*.

(Gheraibia et Moussaoui, 2013) : Gheraibia, Y., & Moussaoui, A. (2013). Penguins search optimization algorithm (PeSOA). In Recent Trends in Applied Artificial Intelligence: 26th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2013, Amsterdam, The Netherlands, June 17-21, 2013. Proceedings 26 (pp. 222-231). Springer Berlin Heidelberg.

(Gómez et al., 2022) Gómez, N., Peña, N., Rincón, S., Amaya, S., & Calderon, J. (2022, March). Leader-follower behavior in multi-agent systems for search and rescue based on pso approach. In *SoutheastCon 2022* (pp. 413-420). IEEE.

(Huang et al., 2019) : Huang, X., et al. (2019). Exploration in Extreme Environments with Swarm Robotic System. International Conference on Mechatronics (ICM), Vol. 1, pp. 193- 198.

(Huang et al., 2022) : Huang, J., Song, Q., & Xu, Z. (2022, October). Multi robot cooperative rescue based on two-stage task allocation algorithm. In *Journal of Physics: Conference Series* (Vol. 2310, No. 1, p. 012091). IOP Publishing.

(Íñiguez, 2017) : Íñiguez, A. (2017). The Octopus as a Model for Artificial Intelligence.

(Morin et al., 2023) : Morin, M., Abi-Zeid, I., & Quimper, C. G. (2023). Ant colony optimization for path planning in search and rescue operations. *European Journal of Operational Research*, *305*(1), 53-63.

(Nogueira, 2014) : Nogueira, L. (2014). Comparative analysis between gazebo and v-rep robotic simulators. *Seminario Interno de Cognicao Artificial-SICA*, *2014*(5), 2.

(Paez et al., 2021) : Paez, D., Romero, J. P., Noriega, B., Cardona, G. A., & Calderon, J. M. (2021). Distributed particle swarm optimization for multi-robot system in search and rescue operations. *IFAC-PapersOnLine*, *54*(4), 1-6.

(Pinciroli et al., 2012) : Pinciroli, C., Trianni, V., O'Grady, R., Pini, G., Brutschy, A., Brambilla, M., ... & Dorigo, M. (2012). ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm intelligence*, *6*(4), 271-295.

(Rahman et al., 2022) : Rahman, A., Bhattacharya, A., Ramachandran, T., Mukherjee, S., Sharma, H., Fujimoto, T., & Chatterjee, S. (2022). AdverSAR: Adversarial Search and Rescue via Multi-Agent Reinforcement Learning. *arXiv preprint arXiv:2212.10064*.

(Ruetten et al., 2020) : Ruetten, L., Regis, P. A., Feil-Seifer, D., & Sengupta, S. (2020, January). Area-optimized UAV swarm network for search and rescue operations. In *2020 10th annual computing and communication workshop and conference (CCWC)* (pp. 0613-0618). IEEE.

(Sadiku et al., 2021) : Sadiku, M. N., Musa, S. M., & Ajayi-Majebi, A. (2021). *A Primer on Multiple Intelligences*. Springer.

(Şahin, 2005) : E. Şahin, "Swarm robotics: from sources of inspiration to domains of application," in Swarm Robotics Workshop: State-ofthe-Art Survey, E Şahin and W.

Spears, Eds., Lecture Notes in Computer Science, no. 3342, pp. 10–20, Berlin, Germany, 2005.

(Schranz et al., 2020) : Schranz, M., Umlauft, M., Sende, M., & Elmenreich, W. (2020). Swarm robotic behaviors and current applications. *Frontiers in Robotics and AI*, *7*, 36.

(Tan et Zheng, 2013) : Tan, Y., & Zheng, Z. Y. (2013). Research advance in swarm robotics. *Defence Technology*, *9*(1), 18-39.

(Wang et al., 2023) : Wang, Y., Liu, W., Liu, J., & Sun, C. (2023). Cooperative USV–UAV marine search and rescue with visual navigation and reinforcement learning-based control. *ISA transactions*.

(Yang et al., 2022) : Yang, L., Xing, B., Li, C., & Wang, W. (2022, April). Research on Artificial Bee Colony Method Based Complete Coverage Path Planning Algorithm for Search and Rescue Robot. In *2022 5th International Symposium on Autonomous Systems (ISAS)* (pp. 1-6). IEEE.

(Yang, 2010) : Yang, X. S. (2010). *Nature-inspired metaheuristic algorithms*. Luniver press.

(Zedadra, 2016) : Zedadra, O. (2016). Résolution des problèmes de coordination d’agents réactifs dans un environnement incertain.

Webography

[1] Pradeep Budhathoki. (07 December 2018). Swarm Intelligence. Basically, swarm is a large or dense… | by Pradeep Budhathoki | The Zerone | Medium. Medium. https://medium.com/zerone-magazine/swarm-intelligence-676e40968473.

[2] Sillon Belge. (1 September 2020). Honeybee Valley (UGent) fait avancer la recherche sur la résistance des abeilles - SillonBelge.be. SillonBelge. https://www.sillonbelge.be/6450/article/2020-09-01/honeybee-valley-ugent-fait-avancer-la-recherche-sur-la-resistance-des-abeilles.

[3] Vincent Lucchese. (20 December 2019). Usbek & Rica - Surpêche : peut-on faire de l'agroécologie dans l'océan. Usbeketrica. https://usbeketrica.com/fr/article/surpeche-agroecologie-dans-ocean-sauver-le-present.

[4] La Rédaction D HYPERLINK "https://www.desdomesetdesminarets.fr/"& HYPERLINK "https://www.desdomesetdesminarets.fr/"M . (7 January 2018). EN IMAGES. Ces clichés d'une murmuration d'étourneaux sont à couper le souffle ! - Des Dômes & Des Minarets. Des domes et des minarets. https://www.desdomesetdesminarets.fr/2018/01/07/en-images-ces-cliches-dune-murmuration-detourneaux-sont-a-couper-le-souffle/.

[5] SARAH GRIFFITHS FOR MAILONLINE. (7 December 2015). Emperor penguins get so hot in their huddles they EAT snow to cool down | Daily Mail Online. Dailymail. https://www.dailymail.co.uk/sciencetech/article-3349517/Now-s-warm-hug-Emperor-penguins-hot-huddles-EAT-fresh-snow-cool-down.html.

[6] Kimberley Mok. (12 October 2017). Swarmanoids: Modular Robots that Assemble to Form Artificial Nervous Systems: THENEWSTACK. https://thenewstack.io/swarmanoids-modular-robots-assemble-form-artificial-nervous-systems/.

[7] (4 May 2021). Initial operating capability for swarm drone technology could be achieved in less than ten years: Poll. Airforce Technology. https://www.airforce-technology.com/news/initial-operating-capability-for-swarm-drone-technology-could-be-achieved-in-less-than-ten-years-poll/.

[8] Guru Med. (17 August 2014). Constituer le plus grand essaim robotique, capable de s'auto-organiser, pour pas cher – GuruMeditation. GuruMeditation.

https://www.gurumed.org/2014/08/17/constituer-le-plus-grand-essaim-robotique-capable-de-sauto-organiser-pour-pas-cher/.

[9] Gérard LIZARD. (30 march 2018). Diapositive 1. bioperoxil.u-bourgogne. https://bioperoxil.u-bourgogne.fr/images/stories/pdf/2018_03_30_Conf_Nano_Club_Condorcet.pdf.

[10] Gwen72. (15 December 2020). Découverte du robot Thymio — Wikidebrouillard. Wikidebrouillard. https://www.wikidebrouillard.org/wiki/D%C3%A9couverte_du_robot_Thymio.

[11] Editorial. (03/12/2022). Real-life applications of swarm robots. Roboticsbiz. https://roboticsbiz.com/real-life-applications-of-swarm-robots/.

[12] Heba Soffar (21 December 2018). Online-Sciences. https://www.online-sciences.com/robotics/rescue-robots-mobile-robots-types-uses-advantages-and-disadvantages/.

[13] (14 November 2016). WWF.https://www.wwf.fr/vous-informer/effet-panda/creation-dune-aire-maritime-protegee-en-antarctique.