

Review

Swarm Robotic Interactions in an Open and Cluttered Environment: A Survey

Maxime Vaidis [†]  and Martin J.-D. Otis ^{*,†} 

LAR.i Lab, University of Quebec at Chicoutimi, Saguenay, QC G7H 2B1, Canada; vaidis.maxime@gmail.com

* Correspondence: martin_otis@uqac.ca

† These authors contributed equally to this work.

Abstract: Recent population migrations have led to numerous accidents and deaths. Little research has been done to help migrants in their journey. For this reason, a literature review of the latest research conducted in previous years is required to identify new research trends in human-swarm interaction. This article presents a review of techniques that can be used in a robots swarm to find, locate, protect and help migrants in hazardous environment such as militarized zone. The paper presents a swarm interaction taxonomy including a detailed study on the control of swarm with and without interaction. As the interaction mainly occurs in cluttered or crowded environment (with obstacles) the paper discussed the algorithms related to navigation that can be included with an interaction strategy. It focused on comparing algorithms and their advantages and disadvantages.

Keywords: swarm mobile robots; human-swarm interaction; autonomous rovers



Citation: Vaidis, M.; Otis, M.J.-D. Swarm Robotic Interactions in an Open and Cluttered Environment: A Survey. *Designs* **2021**, *5*, 37. <https://doi.org/10.3390/designs5020037>

Academic Editors: Yunsick Sung and Wei Song

Received: 14 May 2021
Accepted: 4 June 2021
Published: 10 June 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The study of mobile robots swarm has reached a high level of maturity including human-swarm interaction (HSI) [1]. Unlike most existing robotic systems, swarm robotics bear a very large number of robots and promote scaling, which implies that the swarm must work regardless of its size (from a certain minimum size). Their number varies from fifty to a hundred robots. Each robotic unit making up the swarm is easier to reproduce and replace if there is a problem (a hardware failure, a bog, energy storage and management failure, etc.). Favored forms of communication are the use of local communications, infrared or wireless. Robots communicate with each other both for decision making and for sharing information about their perceived environment. The redundancy of perceived information promotes the stability and robustness of the system. This implies the capacity of the swarm to continue to function despite the failures of certain individuals composing it and/or the changes that may occur in the environment. The swarm is able to adapt in a better way to its environment compared to an external disturbance. This flexibility implies a capacity to propose solutions suitable for the tasks to be carried out. However, some issues arise as the number of sensors and data limit the capacity to analyze and find an analytical solution. Moreover, due to a lack of standardization in methodologies, software and hardware, swarm has less real-world applications [2]. Moreover, each robot composing the swarm has a simple individual performance. The robots are almost identical to each other, and this is common in most of swarms. They are controlled in decentralized mode. For swarm systems in decentralized mode, the individual performance of each robot is asynchronous. This means that the sequence of their perception-decision-action loop (sensing, processing, until servomotor actions) is performed independently of other robots. They do not have a global knowledge of the system in which they cooperate. A swarm improves the execution of complex task when decentralized sensing is required compared to a single robot. Examples are in applications such as field exploration, searching for a target, surveillance or rescue. This is possible because of their number as well as their group intelligence which allows distributing tasks between robots in the swarm. These various characteristics of the swarms

of robots allow them to have certain properties compared to simpler and less complex robotic systems.

Despite all of these proprieties, most modern mobile swarms are controlled by one or more operators. The operators must follow the evolution of robots, and influence their performance: if necessary, they should assign them a different goal to achieve. There are many problems seen in the implementation of more automated robots swarm. One of them, and not the last, is to find an optimal balance between the individual command of a robot and the overall performance of the swarm. The robot must have enough liberty to perform its actions, and also comply with the aims and goals of the swarm. Another important problem is the planning of the trajectory. The swarm must ensure that each robot which composes it is moving to the right direction and avoids obstacles present on the road. There is massive literature on this subject of simple robotics systems. There are many types of planning suggested: local planning and overall planning. The local one works on the assumption that the robot does not have all the information about its position and that of its target. Therefore, it must progress towards its target with the information it is detecting and it is looking for as it progresses. On the contrary, the overall planning is only possible if the robot knows the entire environment between its position and the targeted one. The first planning is often preferred because the environment in which robots are progressing is variable.

A large number of algorithms for simple robotic systems exist for this purpose; most of them are inspired by the animal or physical world such as genetic algorithms or potential fields. There is currently no literature review of algorithms used for moving swarms of mobile robots interacting with human, such as an operator or a person needing help. Moreover, less research works are interested in using swarm to help migrants in hazardous environment. To better understand design methodologies, taxonomies related to design, analysis methods and collective behavior were previously addressed in [3]. This review will therefore aim to fill the information gap on interaction and trajectory planning concepts for robots swarm by identifying key issues and future work. First, we will introduce our article selection methodology for our review in Section 2, and secondly, we will present in detail in Section 3 the concept of robots swarm, specifically the targets, goal or objectives that they are asked to achieve. Section 4 is the main contribution of the paper and it will focus on the interaction media between a human and a swarm of robots. This section will help to better understand the behavioral outlook of a swarm dedicated to helping people on the move. We will propose taxonomy of these algorithms as well as a discussion detailing our conclusions in Section 5. Finally, we will conclude our discussions on the remaining problems and issues which have to be resolved and future research to be carried out in Section 6.

2. Methodology

In this study, an in-depth two-step search was carried out on swarms of mobile robots, both on the means of interaction between these and the operator, as well as the various algorithms that can exist to make them navigate in an open and cluttered environment. First, some research was done based on the Scopus database for articles related to the domains of the swarms of robots. Keywords were used such as ‘swarm interaction human’, ‘human swarm mobile robot interaction’, ‘swarm robot interaction human’, ‘mobile swarm intelligence’, ‘swarm motion planning’, ‘swarm outdoor’. An attempt was made to find papers related to rover helping migrants journey, but the result is limited to one paper related to rovers swarm. However, four papers related to UAV along borders for surveillance and policing were found. Secondly, the articles related to the subject of this study were kept manually.

The study selection criteria for scientific articles are based on the definition of a swarm of robots. Indeed, the authors only selected swarms of robots completely or in mobile parts on the ground. Some of the selected references are related on multi-robots system in order to compare their algorithms. Drones (or UAV) swarms are not considered because

many of their characteristics are different compared to mobile homogeneous robots in a swarm. For instance, due to less power autonomy and weight load of sensors, they need different strategies to pursue their goals. The authors have read the selected articles and those dealing with either interaction between a human and a swarm, or algorithms that make them to navigate in an open and cluttered environment.

After applying these criteria, 12 articles concerning the human-swarm interaction and 60 articles concerning the use of algorithms which can further allow a swarm to navigate in an open environment with obstacles were found. These articles will be analyzed and discussed in this survey. The study will provide answer to the following research questions:

1. Which media are currently used to control a swarm of robots?
2. What are the constraints to the use of each of the supports?
3. How does interaction support influence the relationship between the robots swarm and humans?
4. How does this support influence the level of autonomy of the swarm?

The taxonomy of these interaction supports is presented in Section 4 as well as the answer to the questions above. Section 5 will focus on the different algorithms used by mobile robots swarm in an open and cluttered environment. The study will provide answer to the following questions:

1. What are the existing algorithms?
2. In what ways does the algorithm used influence the performance of swarm?
3. In which contexts can each algorithm be used?
4. What level of autonomy does the algorithm offer to swarm?
5. Which constraints of use does the algorithm impose on swarm?

3. Swarm of Robots

The design and manufacturing of a robots swarm must, before anything else, be made as a function of its use. The swarm must be adapted to the task it does; otherwise the aim may not be achieved. Through the reading of these articles, we have arranged the activities of the swarm into three categories: (1) navigation and trajectory, (2) task to do and (3) maintaining the structure of the swarm aimed for the conception of these swarms.

3.1. Navigation and Trajectory

This category is the one that the majority of swarms of mobile robots must accomplish. It is divided into two subcategories: exploration/avoidance of collision, and reaching a targeted position given by an operator or by the swarm itself. The existing algorithms for achieving this objective are detailed in Section 5.

3.2. Swarm Robotics Tasks

One of the advantages of robots swarms is that they can do many tasks faster by dividing the work. Seven tasks done by swarm are presented in this paragraph and papers which are doing these tasks are listed:

Localization of the target :

Husnawati et al. [4] developed a robots swarm to identify a gas leak. Aniketh et al. [5] set up a swarm to find people who needed help. The literature review by Senanayake et al. [6] and Saeedi et al. [7] describes most of the algorithms which can locate a target. Garzn et al. [8] created a swarm capable of detecting a chemical source or radiation source, particularly for mines. Fricke et al. [9] built immune system T cells to develop a target search algorithm that can be applied to robots swarm. Zhang et al. [10] developed a swarm capable of assisting a hunter in locating a target for hunting.

Surveillance of a region:

Hacohen et al. [11] created a swarm capable of intercepting targets which are not desirable in a surveyed zone. In [10], the robots swarm also allows the survey of the zone with the aim of finding prey to hunt.

Rescue:

In [5], the swarm can locate a person in order to warn the emergency services so as to step in. The possibility of location offered by [6] and [7] also helps to warn emergency services if a person in danger is found. Gutierrez et al. [12] proposed a humanitarian swarm platform of multifunctional robots (land, sea, air) that helps to rescue people in danger during natural disasters.

Follow-up of a target:

The literature review by [6] describes the existing algorithms for the follow-up of a target by a robots swarm.

Prevention and detection of a forest fire:

The literature review by [7] proposes a robots swarm which is capable of detecting and warning emergency services in the case of forest fire.

Maintenance of installation:

The literature review by [7] also proposes a robots swarm which can ensure the maintenance of an installation.

Transport of material/cooperation:

Contreras-Cruz et al. [13] created a swarm of mobile robots that can transport objects in warehouses. Ardakani et al. [14] offer a swarm of robots capable of transporting plastic plates. Sun et al. [15] also developed a swarm of robots that can carry objects in a warehouse.

3.3. Maintaining the Structure of the Swarm

The structure of the swarm considers its geometric formation in the space under some constraints such as energy storage and management, geometry of the environment while exploring different zones, signal strength to share wireless data, etc. Then, these constraints were detected to maintain the structure of the swarm:

Adapting the size of the swarm:

Zelenka et al. [16] propose an algorithm capable of adapting the size of a robot swarm during the exploration of a zone. When there are too many robots in the swarm located in a same zone of proximity, they can decide to explore another zone.

Data sharing:

Dang et al. [17] chose a strategy to share all the data concerning the environment of their robot swarm to make some exploration of ground.

Coordination of the swarm:

In [13], the use of an algorithm of colony of artificial bees allows the maintenance of the cohesion of the swarm. Bandyopadhyay et al. [18] created an algorithm using the properties of the chains of Markov to ensure the stability of their swarms. Araki et al. [19] leaned on an algorithm that optimizes the movement of a swarm of robot taking into account the environment of the mobile and flying robots, battery voltage and state of charge (remaining battery energy) as well as the objective to achieve. Hattori et al. [20] present an algorithm of estimation of position for mobile robots to maintain their formation during their movement. Luo et al. [21] use an algorithm of movement of a swarm of robots in which a robot finds its way in relation to the others robots and moves forward randomly. Das et al. [22] proposed an improvement of the algorithm Particle swarm optimization to maintain the coordination of swarm. Bandyopadhyay et al. [23] used a probabilistic approach to lead a swarm of mobile robots. Liu et al. [24] presented a swarm of mobile robots capable of adapting to their environment by ensuring that they agree with each other thanks to the data collected on their environment. Poundmaker et al. [25] used an algorithm that keeps the formation of the swarm of robots due to the position of the leader and that of the robots relative to each other. Wallar et al. [26] used the combination of potential fields and probabilistic methods to maintain this coordination. Kim et al. [27] created a Firefly algorithm to satisfy this objective. Chang et al. [28] developed an algorithm

capable of maintaining the formation of a swarm of mobile robots subjected to strong disturbances due to wind.

Energy optimization:

Jabbarpour et al. [29] used an improved ant colony algorithm to optimize the energy consumption of a mobile robots swarm. As noted earlier, Araki et al. [19] used an energy optimization algorithm for its swarm.

3.4. Conclusions

As we have seen, swarms of robots can have many purposes depending on their ability to achieve a task. All of these tasks and actions can be done if a swarm is able to move itself into the environment of its mission. To do these, the swarm needs algorithms to plan its path and motion. The next sections will present many algorithms developed to achieve these goals, according to the type of swarm. We will do a taxonomy to sort them and compare them between each other.

4. Interaction Models for Human Being-Swarm

The interaction between a human and a swarm can pose many problems and issues. Indeed, there are many obstacles that can prevent the swarm from achieving the human objective:

The human objective:

This must be attainable by the swarm according to its capabilities. If the target is too complex for the swarm functionalities, it will not be achieved.

The means of communication:

To communicate their objectives, the operator must use an appropriate means of bidirectional communication that enable both the operator and swarm to be understood.

The travelling environment:

Depending on the environment, there are different problems involved in moving a swarm. In outdoor sites, weather conditions and fields of deployment are the main challenges to overpass. In indoor areas or building, communication between the swarm and operator can be very difficult due to loss of communication signals. The difficulty also increases if the operator does not have a line of sight on the swarm, and if he controls it through a graphical interface that gives him the essential information.

The level of autonomy of the swarm:

If a swarm is very dependent on the operator's decision, the operator must constantly observe the evolution of the swarm and guides it swarm in its task. If the swarm has a high level of autonomy, this would not be the case. An optimal operational shared autonomy between a swarm and an operator depends on the complexity of the mission and environment. An operator should only submit commands at a strategic level. Of course, a complex mission could need submitting commands at a tactical level. The strategy chosen will influence the number of robots deployed.

The number of robots composing the swarm:

With more robots composing the swarm, it becomes more difficult for the operator to control the swarm behavior considering all constraints such as battery voltage or state of charge, the current state of the mission and what has been accomplished in the mission.

4.1. Swarm Interaction Taxonomy

This section will present the studies that have been conducted for this purpose. Figure 1 shows a possible taxonomy for these different means of interaction depending on the support used. In this figure, hybrid method is possible such as using Augmented Reality to see the swarm, Haptic to control the structure of the swarm and electrocardiogram to control, as an example, the velocity and orientation of the swarm.

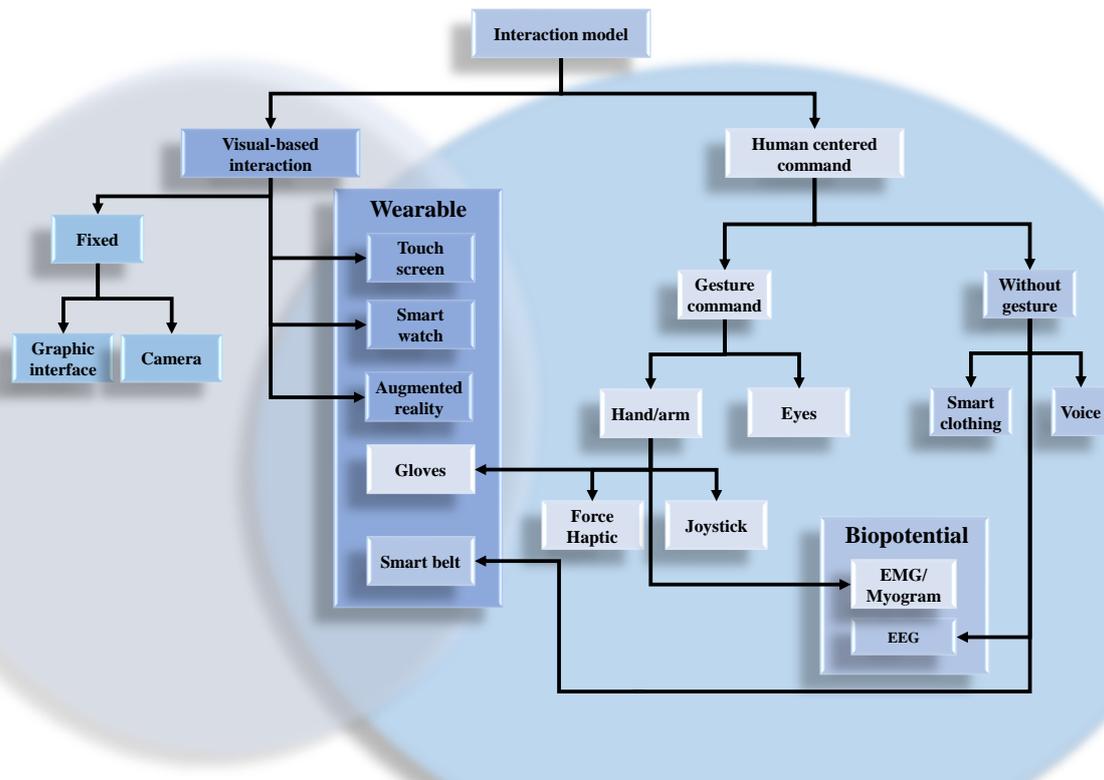


Figure 1. Taxonomy of interaction support for mobile robots swarm.

In their article, Bowley et al. [30] proposed to control a swarm of robots from a phone or tablet with their touch screen. It has several functions that can be used due to finger movements (touching or removing fingers, scanning the screen, enlarging or reducing with two fingers, etc.). With this interface, the operator uses an algorithm to influence the behavior of the swarm through several attractive or repulsive beacons:

The attractive beacon:

It attracts the robots swarm towards its position.

The obstacle beacon:

It emits repulsive force so that the robots avoid going to its zone and thus avoid collision with the obstacle.

Recall Beacon:

Similar to attractive beacon, it is used in an emergency or at the end of a test exercise.

The management beacon:

It is supposed to lead the swarm towards this target.

The beacon circle:

It is a mix between the attractive, obstacle and the management beacon. It is used for zone control.

Dividing or multiplying beacon:

It is used to change the perception of the environment of robots in an area in order to change their behavior accordingly.

Each of the beacons located on the screen has a modifiable influence radius. Simulations were carried out to validate the operation of this concept, which allows the behavior of a swarm of robots to be intrinsically modified.

Crandall et al. [31] developed an interface that allows an operator to interact directly with a swarm of modeled mobile robots following a bee colony. This is done in order to share decision making process and offer fault-tolerant capabilities. Thus, the goal of the swarm is to find quality sites to collect resources. Each robot behaves like a bee. It can enter different states: exploration, observation, pause, evaluation and dancing as a message.

Each bee will initially explore an area at random. If she encounters a potential site, she will evaluate it and go back to the colony to dance more or less according to the quality of the site. Then she rests before starting the cycle again. Observers watch bees dance to visit potentially interesting sites. If many bees have detected a good site, the colony will exploit it. Initially, the project performed computer simulations of a bee colony. Subsequently, they wanted to improve the safety and speed of bee exploration. To do so, they allowed an operator to place beacons to guide bees in their tasks, and then they evaluated the impact of this interaction on the robots swarm. From this experience, they were able to define several categories of control on the swarm:

Parametric control:

It can be achieved by exciting or inhibiting the behavior of bees in their exploration whether by specifying a direction of research or altering their speed.

Association control:

The operator can directly control one robot of the swarm, which will then influence the overall swarm.

Environmental monitoring:

This is done by placing attractive or repulsive beacons in the bee environment.

Strategic control:

It is to ensure that the swarm changes the allocation of its own goal in order to select the best strategy to adopt. In this case, it would be to reassess the quality of a site after a certain operating time.

In conclusion, Crandall et al. [31] admit that these methods of influence work well if the operator knows exactly how to give the tasks to be carried out by the swarm and accepts sharing its control with others.

Kim et al. [32] developed a swarm of mobile robots capable of tracking people's movement. The system consists of three steps: (1) sequence of operation, (2) receiving/sending messages and (3) approximate location of robots. This interaction takes place through a connected watch and a connected belt. The swarm is composed of a leader who receives orders from the watch via a Bluetooth Low Energy (BLE) communication. The belt is used to assess the distance between the person and the swarm through infrared communication. The leader then sends instructions to the other robots by radio and infrared communication. The authors created the communication protocol for this swarm in order to keep it in formation. This system works for a small number of robots. Indeed, the system is tested with real mobile robots and realized that communication becomes noisy if the number of robots is high. The user can choose the formation of the swarm when moving according to several prefixed patterns.

To interact in various ways with a swarm of robots, Ferrer [33] made an enumeration of various physical supports existing for this purpose. First of all, he takes a gesture taxonomy from the existing hand to be able to apply it to a swarm of mobile robots. This gesture recognition is done via a camera that associates the gesture with a command to be made for the swarm. Of course, hand gesture could also be executed with an electromyography (EMG) such as with an eight-channel armband [34]. In their paper, Mendes et al. described how they can obtain better results by selecting the best feature reduction process of EMG signals data before the classification of gestures. Then another method of communication with the swarm is presented. Several studies have been carried out on the interaction between a swarm and a human via the haptic, especially with the aim of obtaining feedback from others instead of using visual information to help the operator in his control. The operator uses some haptic sensors which send some feedback to him. It does not make a human being an external operator of the swarm, but rather a special member of the swarm. Both methods are hard enough to put in work and cannot allow interacting with a large swarm. Subsequently, various means of interaction by augmented reality are presented. Finally, Ferrer concludes on portable tools on a human that can act as a support for interaction between a swarm and an operator. First, a gesture recognition can be done by an armband that can recognize the gestures of the fingers, hand and wrist

thanks to the muscles of the forearm. The armband used was a Myo armband by Thalmic Labs. With each of these gestures, we can associate a command with the swarm. Then, usually for gesture recognition, it is possible to use the Leap Motion [35] to detect the movement of the fingers via infrared light. It identifies the gestures of the fingers, their movements and their spatial coordinates if necessary. It is a precise tool that can provide a wide range of control for an operator. The last physical support presented is a vest for video game players acting as a connected garment. It is equipped with haptic devices that allow the user to feel immersed in a chosen environment. Ferrer concludes by comparing the advantages and disadvantages of different media of interaction.

In their work, Mc Donald et al. [36] developed a method of interaction with a swarm of mobile robots based on haptic. The purpose of the robots swarm is to carry out patrols and encircle buildings at the request of an operator. When robots encircle a building, they are represented by virtual force fields which then allow the formation of the swarm to be represented by a flexible virtual ring. The operator can perform three types of handling when the robots are in encirclement mode:

Shape exploration mode:

The haptic tool allows the operator to feel the shape of the swarm without changing it. This is possible because of the virtual force field created by mobile robots.

Shape manipulation mode:

This mode allows the operator to modify the formation of the swarm by means of the haptic remote control which changes the shape of the virtual ring.

Spacing mode:

In normal mode, the spacing between each robot is identical. This mode allows the operator to change these values. The operator also has actions to perform during the patrol of mobile robots.

Near travel mode:

This mode activates if the swarm has selected its target position to be reached and it is not in encirclement mode. Its purpose is to allow the operator to reach the target position faster.

Shape exploration mode:

During the work of the swarm, the operator may choose to feel the formation chosen by it without modifying it.

Mc Donald et al. were able to simulate their systems in order to validate them and test the effects of this physical medium on the performance of the operator's controls on the swarm of mobile robots.

Kapellman et al. [37] suggested using Google Glass as physical support. These allow an operator to guide a swarm of robots for the transportation of an object. One of the robots is appointed as the leader of the swarm, the one that the operator can influence. It acts as an intermediate target which the other robots are going to recognize and follow. The operator has the possibility of choosing the leader among the robots of the swarm. He can also check the state of each robot by selecting them and communicating orders via Bluetooth:

Starting the task of the robot:

It is the basic behavior of the robot that is activated.

Becoming the leader:

Movement of the robot can be directly controlled by the operator (go ahead, back, turn right/left, stop).

Overdrive mode:

The robot must ignore all commands from a remote control other than glasses.

Disconnection:

Via connection.

These instructions can be given by the voice command or by touching the glasses. This support could be tested with a real swarm of mobile robots. This medium allows

the operator to have free hands to perform other actions. It was also demonstrated that interaction allows for dynamic selection of the target to reach.

In their work, Mondada et al. [38] decided to process Control operator's electroencephalography (EEG) signal so that it can select a swarm's robot to control it. It is based on the stationary state of the potential evoked by vision (Steady-State visually evoked Potential: SSVEP). This detection is done by flashing light on each robot, allowing knowing whether the selected robot is the one the operator wants. For this, an EEG acquisition helmet is placed on the operator's head. Three parameters are important to extract the SSVEP signal from the EEG: the flashing frequency of the lights, the color of the lights and the distance to the stimulus. Mondada et al. [38] used existing literature to select the ranges of parameters to be tested. The blinking frequencies were chosen according to [39] study. The distance between the target and the operator was chosen according to [40] study. For the color of the LED, the scientific community is not able to give the best one (there is some debate between white, red, green and blue). Several tests were conducted with individuals. The results indicate that the success rate varies greatly from person to person (on average 75% success with a standard deviation of around 15% success depending on the frequencies used). More trained operators are in this process, the better the results will be. This method also delays for several seconds in the recognition of the signal, as does gesture recognition by image or voice. The main disadvantages are the uncontrollable factors for a real application such as the personal attitude of the different operators, the distance from the robots, the brightness, etc.

In their article, Setter et al. [41] used the haptic to obtain feedback about the swarm of mobile robots. The swarm used is made up of a leading robot and other followers robots that maintain a given formation. The operator can control the speed of the leader, which can influence the behavior of the swarm. This is done through a haptic device. The feedback given by the force of the haptic device indicates to the operator whether his control is good or bad for the swarm, that is to say whether the speed of the following robots is more or less different from that of the leading robot. This information allows the operator to adjust the leader's speed. The system was successfully experimented with a real swarm of mobile robots.

Podevijn et al. [42] developed a gesture recognition interface capable of ordering a swarm of mobile robots. A Microsoft Kinect RGB-D sensor is used for body tracking and to identify the gestures of the user. This interface allows the operator to dedicate himself fully to the management of his swarm. The contribution is to have a simple command interpreted by the swarm of decentralized robots and also to allow it to give some feedback. Since a swarm is too difficult to command directly, the swarm could be subdivided it into several sub-swarms. The following commands are used by the operator:

Direct:

The operator can guide a sub-swarm to a target position.

Stop:

The sub-swarm stops.

Division:

Creation of new sub-swarms.

Merger:

Gathering of two sub-swarms.

Selection:

The operator chooses the sub-swarm with which he wants to interact.

Each of these controls is associated with a gesture of the operator's arms. Eighteen participants were able to test this interface with a real swarm of mobile robots.

Kolling et al. [43] provided a 2D graphical interface, which is optimized to display only important information for the operator, to simulate interaction with a swarm of mobile robots. The robots move following Voronoï graphs based on [44], in the environment to be explored. For each new information retrieved, they must return to a departure station that will update the swarm movement card. The operator can visualize these movements from

his interface and interact with a mouse on the swarm via a few commands: stop, go to a zone, appointment point, deployment, random movement, update data, leave a zone. It can also use other means of control, such as a robot selection rectangle, which defines a sub-swarm that is obedient to different commands of the swarm in general, and also places a beacon that attracts robots to its area.

Diana et al. [45] used a joystick made of modeling paste as a physical medium for interaction. This allows the operator to control the formation of the robots swarm with the geometry of the modeling paste. It uses modeling paste to define the desired formation for its swarm. A camera is used to take the form (scan the geometry) and compare it to a library for the identification and classification of the geometry. Once this is done, the information is sent to the swarm that performs the desired formation using a method that minimizes the energy of the system during its displacement. Simulations were carried out with a real swarm of mobile robots.

Alessandro et al. [46] developed a human-swarm interaction based on the recognition of hand gestures. For this, 13 gestures was used and 70,000 images of these gestures was collected by cameras representing the position of all the fingers of the hand. These data were used to train a vector support machine that will perform the classification of the 13 gestures by affecting a probability of belonging to a category of the gesture to be recognized. Every swarm robot has a camera on it. The robot move around the operator to improve their point of view and facilitate gesture recognition. The robots then share the information obtained by their classification and the swarm makes a decision afterwards.

4.2. Discussion

Table 1 shows a summary of the various interaction media. Through these various articles, we have been able to observe the diversity of the interaction between human and swarms. These have several advantages and disadvantages depending on their nature. One of the advantages is the ability to control the formation of the swarm in order to adapt it to its changing environment. Despite this control, the operator must always be able to explicitly give a target to the swarm. There is no interaction support that can do this implicitly. This has an impact on the autonomy of the swarm, which certainly remains at a fairly high level but cannot be completely autonomous in its decision-making. Its autonomy is limited to planning its displacement and mastering its deployment training. The following section will be devoted to algorithms that can perform these actions.

Table 1. Summary of the various supports of interaction.

| Papers | Way of Interaction | Type of Interaction | Interaction Context | Swarm Autonomy | Advantages | Usage Constraints |
|------------------------|--|--|---|--|---|---|
| Qin et al. [30] | Touch screen on the phone or tablet | Beacon to influence the swarm | Change behavior of the swarm to easily explore areas | The swarm needs only a target to work | Change global behavior of the swarm without complex commands | Not allow selecting robots separately |
| Crandall et al. [31] | Graphic interface | Change parameters of hub-based colonies | Change behavior of the swarm to easily explore areas | The swarm needs only a target to work | Allow us to have a deep control on the swarm behavior | Need knowledge about the algorithm to use it correctly. Not allow selecting robots separately |
| Kim et al. [32] | Smart watch/belt | Command send to the leader | Control the form of swarm during his motion | The swarm control his motion and the form ordered | The operator controls the swarm's form | Not possible to control the motion of the swarm and to select one robot separately |
| Ferrer [33] | Hand gestures by camera/haptic/Myo band/connected vest | Command to control the swarm form Feel the feedback of the swarm | Control the form of swarm during his motion | The swarm control his motion and the form ordered | The operator controls the swarm's form and have some feedback | The operator should see the swarm and each of his gesture could be interpreted as a command |
| Mc Donald et al. [36] | Haptic | Control the form of the swarm and change it if needed | Control the form of swarm during his motion | The swarm needs only a target to work | Many people can control the state of the swarm at the same time | The operator cannot see the swarm. He can only feel feedback provide by the swarm |
| Kapellman et al. [37] | Google glass | Command send to the leader | Allow us to guide the swarm during the transportation of objects | The swarm needs a regular monitoring to achieve his target | The operator can select any robots and can send many orders to the leader | The operator should follow the swarm during his motion. He also should see it |
| Mondada et al. [38] | EEG signal | Select one robot by thought and vision | Allow us to select a robot in order to perform a task | The selection depends of the operator | The operator doesn't need to do gesture to interact with the swarm | This method is difficult to apply and needs learning (depend of the operator) |
| Setter et al. [41] | Haptic | Command send to the leader | Allow us to control the behavior of the swarm through the leader | The swarm needs a regular monitoring to achieve his target | The operator can change behavior of the swarm through one robot | The operator should follow the swarm during his motion. He also should see it |
| Podevijn et al. [42] | Gestures recognition | Control the swarm form | The operator can give order by selecting one or several robots | The swarm follows the choice of the operator | The operator can guide the swarm as he wants | The operator should check the behavior of the swarm constantly |
| Kolling et al. [43] | Graphic interface | Give order to the swarm (shape and target) | Change shape of the swarm during his motion to easily explore areas | The swarm needs only a target to work | The operator can select any robot and give him several orders | The operator should follow the swarm during his motion. He also should see it |
| Diana et al. [45] | Joystick and camera | Control the form of the swarm | Allow us to select the form of the swarm | The swarm follows the choice of the operator | The operator can select any form for the swarm | Quite some time is required before a command is executed by the swarm |
| Alessandro et al. [46] | Gestures recognition | Decision taken by the swarm | Give some orders to robots by gestures | The swarm follows the choice of the operator | The operator can select any form for the swarm | The operator should see the swarm and make an exact gesture to give an order |

5. Algorithms to Motion a Swarm in an Open Environment with Obstacles

There are many challenges in moving swarms of robots, especially if their environment is crowded. Because of this uncertain environment, uncertainties may arise when operating mobile robots. These may be due to vagueness of sensor measurements, lack of environmental knowledge and lack of control of external disturbances on robots. It all depends on the set up of the swarm as well as the type of environment in which they operate.

One of the big challenges today is to allow robots to operate in an environment without having to adapt them to the environment, i.e., robots are self-sufficient to carry out their mission. In these circumstances, ensuring the performance of a task under the conditions of safety and efficiency requires consideration of the environment as it can be perceived by embedded sensors. In addition, the swarm must be equipped with algorithms that enable it to move and perform the tasks it must perform. This section will be devoted to the presentation of existing algorithms for this purpose. We will describe them and discuss their effectiveness using a high level taxonomy of these algorithms as presented in Figure 2.

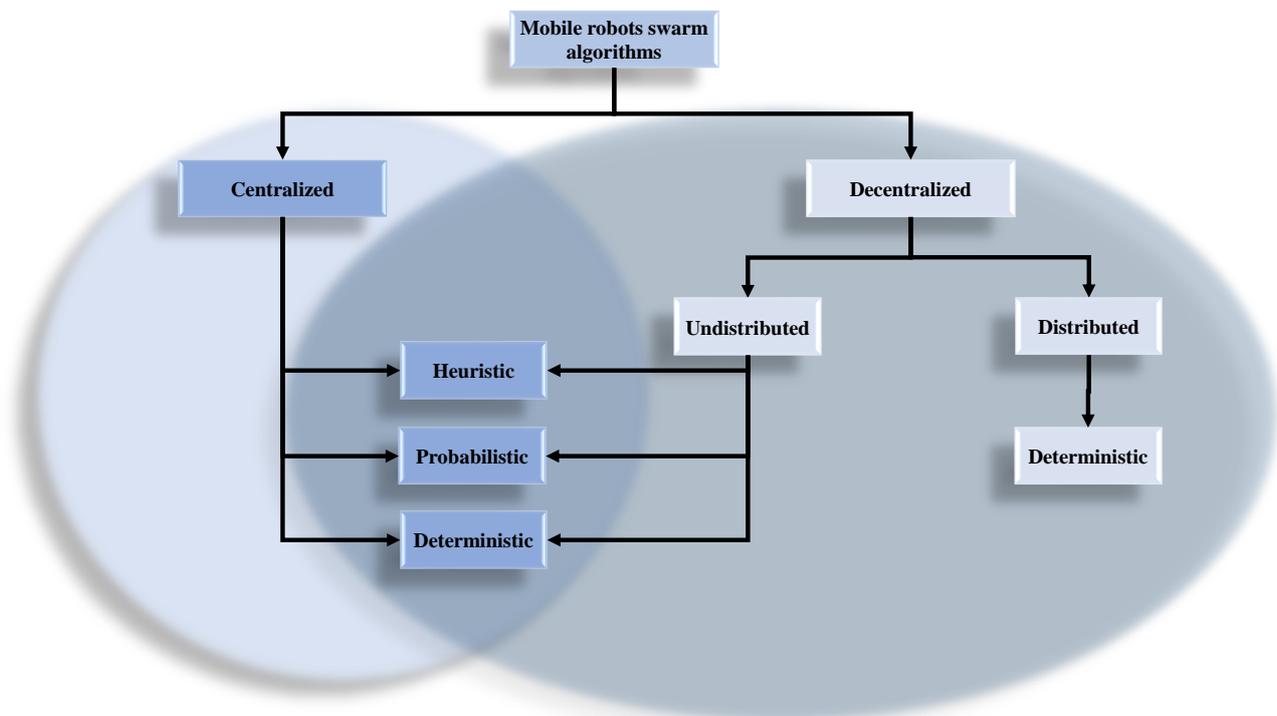


Figure 2. Taxonomy of algorithms for mobile robots swarm.

5.1. Centralized Swarm

A centralized swarm is a swarm controlled by a leader, which can be a robot of the swarm or a distant server which sends command to the robots. The leader can also be a human operator sending the commands to the swarm. In this section, we will present all the algorithms developed for this kind of swarm.

5.1.1. Deterministic Algorithm

Vaidis and Otis [47] created a swarm which is capable of adapting its shape according to the displacement of a group of migrants. The main purpose of this swarm is to protect these people from an attack when they are moving. The swarm is commanded by a leader which analyzes the situation and sends some commands to all robots. The algorithm used

to control the position of each robot is divided into three steps. The first step is to find the position around the people each robot will have to reach. The position of people is processed and allows the swarm to create a convex hull around them. Each robot has a position to reach on this convex hull, where these positions are uniformly distributed according to the number of robots. Then, a path planning algorithm is used to compute the path of each robot in order to reach their targeted position. The path planning used a Vector Field Histogram (VFH) method [48] to detect obstacles and bypass them. The last step is an algorithm which takes the result of the VFH algorithm, and converts it into a motor command for each robot. This last algorithm used a fuzzy logic to find the good command according to the target position and the obstacle to be avoided. With all these three parts, the leader is able to control all the robots and move them around the group of migrants. Vaidis and Otis also used a state detection algorithm in order to detect some issues with robots. This algorithm used a Convolutional Neural Network (CNN) to process the data coming from an Inertial Measurement Unit (IMU). The data of the IMU are converted into a picture, then these pictures are analyzed by the CNN to find the state of the robots. Four states were studied: normal state, fallen state, skid state and collision state. The result shows a good performance of the detection compared to other methods used. The goal of this detection is to find an issue with one robot, and then replaces it by another one of the swarm to do the task it cannot do anymore. The swarm was tested in an indoor environment with real robots.

Qin et al. [49] developed an algorithm in three (3) stages which can make this mission for a marine swarm of robots: assignment of the objectives, the planning of the trajectory and order of engines. An operator is necessary to oversee the swarm. This one can send simple orders to robots, for example the target, the goal or the objective to achieve. The first stage tries to position a robot in relation to the others. A central point is located and their position is defined by the variation of their distance face to face from this point. Then, the algorithm tries to define the best orientation and the speed to be given for robots. To avoid collisions between robots or with obstacles, a method of the fields of potential is applied. It gives the desired orientation value and speed for the movement of each robot. Robots are controlled by a Lyapunov function [50]. Simulations were conducted to validate the algorithm in different situations. They are able to deal with different kinds of barriers and do optimization, computation and analysis in real time. The formation of the swarm is not maintained but this does not prevent it from achieving its objectives.

Araki et al. [19] offer a system capable of directing robots that can fly and move on the ground called Crazyflie. This flying car is composed of two wheels, a ball caster, a motor for the wheels and four motors for the rotors used as a quadcopter. The weight of the platform is around 41 g. The swarm takes into account the energy consumption of each of the robots to carry out their displacement. Two algorithms share this task: one performs the path planning for the swarm, while the other optimizes the solutions found by the first one. Trajectory planning is based on a graph of the robot environment. A travel energy cost function for each robot is defined and will need to be minimized. The cost of travel varies based on whether the robot is on the ground or flying in the air. Algorithm A* based on [51] is used to find a solution to the displacement problem. Several paths are considered and the optimization of the problem is then carried out according to the energy consumed by the robots as well as the non-collision constraints. This path planning is computed according to a cost function calculated for each edge of the map, based on the work due to the displacement of the flying car. The cost function $c(e_i)$ of one edge e_i is presented in Equation (1).

$$c(e_i) = \mu \frac{W}{W_{max}} + (1 - \mu) \frac{t}{t_{max}}, \quad 0 \leq \mu \leq 1 \quad (1)$$

W_{max} and t_{max} are the maximum possible energy and time of any edge in the graph. W is the work due to the displacement of the flying car calculated depending on if the car is flying or driving with the distance between the edges, its power consumption and

velocity in both cases. Power consumption is calculated in real time and a threshold is used to indicate if the power is low and limits the displacement of the robot. The parameter μ is used to tune the planner according to the weight energy and time in the cost function. Simulations and experiments have been carried out and have shown that robots consume much less energy by driving rather than by flying; but the flying mode is quicker than the driving one. Because of this, flying can serve as a high-cost and high-speed transport option, while driving serves as a low-cost and low-speed option. The robots were also able to travel without collisions.

Wei et al. [52] use the principle of the graphs of Voronoi [44] to move their swarm of mobile robots. These have to reach a platform where they will have to make their tasks. Their environment is cut in cells of polygonal shape in which the center of these is placed in their centroid (Centroidal Voronoi Tessellation [53]). The algorithm acts in several steps:

1. The target of robots is defined.
2. The system initializes its parameters with the aim of computation.
3. The diagram of Voronoi is generated and cells are computed.
4. The error of position of every robot is evaluated.
5. If this one is bearable, the algorithm pursues its execution. Otherwise, it begins again from the beginning by updating the position of the robot.
6. The robot performs the given trajectory. If the target is reached, the robot performs its task. Otherwise the next iteration is done to plan its next move.

Each robot is represented with a rectangular prism in order to simplify the recognition of collisions. Several simulations were performed by varying several parameters such as the number of robots used or the error tolerance threshold. They show that as the number of robots increases, the time the algorithm iterates increases.

Vatamaniuk et al. [54] offer an algorithm capable of representing the swarm of mobile robots with a convex envelope. Each robot is represented by a small circle of a fixed radius. The algorithm consists of six steps:

1. Analysis of the shape of the desired convex envelope and assignment of the coordinates to be attained on it.
2. Placing possible passage points on the contour of the convex envelope to allow robots to cross it without collisions.
3. Adding two normal equidistant points to the convex envelope in relation to each final coordinate point or in relation to each point at the crossing points.
4. Assigning final coordinates to each robot on the convex envelope.
5. Tracking planning for robots: they must successively reach the nearest normal points in order to rationalize their final objective.
6. Setting a deadline to avoid collisions between robots. It depends on the distance between the moving robot and the one closest to it, as well as its speed. Once all the delay problems have been resolved, the order is sent to each of the robots.

This algorithm is interesting for several reasons. First of all, the computation time is very low, which allows the swarm to move in real time. In addition, the trajectories are all segments which simplify the movement of robots. They change directions up to three times during their trip, saving the battery life. Simulations show that algorithm performance is acceptable in up to 100 robots in the swarm.

Garzon et al. [8] developed an algorithm that can help a swarm of mobile robots explore an area. Exploration takes place in different spiral forms of robots' movement. Their goal is to find a signal from a beacon, which is used to simulate mines or detect chemical source. Each robot has an area around it where it can detect obstacles or listen to the transmission of information. The algorithm optimizes the movement of robots to cover as much ground as possible within this area. The spirals made will move the robot from the center of the area to be explored to its periphery in a square or rectangular shape. The robot sends a signal every 100 ms to detect the beacon. If it obtains a response, it measures the strength of the signal in order to evaluate the transmitting distance. Experimentations

were conducted with three robots each covering a specific area. Several beacons were placed in them for the robots to detect. Comparison between the different strategies used was successful.

Liu et al. [24] developed a mobile robots swarm control system that can be operated by an operator. It sends orders to the group leader. The leader communicates and executes tasks to the entire swarm. Path planning is done by minimizing a defined cost function for each robot. It takes into account the distance between the robot and an obstacle and the distance between the robot and the rest of the swarm. The stability of the formation of the swarm is controlled through a function of Lyapunov-Krasovskii [55]. Simulations were conducted to validate the operation of the system in obstacle configurations and by changing several parameters. They have shown that the swarm is well able to move without collisions and can maintain training through redundancy of information.

Radu-Emil Precup et al. [56] created a trajectory planning system for mobile robots that can adapt to load levels of robots. A finite number of rovers is composing the swarm. At the beginning of the algorithm, their initial position is known. At each iteration, they move a certain distance in a straight line to their target. The goal of the algorithm is to minimize the distance traveled for each robot as well as avoid collisions. To do this, four optimization variables are introduced into the computation:

1. One which minimizes the Euclidean distance between the position of each robot specific to the same population at each iteration.
2. Another which maximizes the distance between robots of the same population and the nearest robot of another population in order to avoid collisions.
3. The third and fourth variables are used to maximize the distance between the trajectories of each of the robots in X and Y to avoid collision.
4. A fifth penalty variable can be added in certain situations that need to be avoided.

The algorithm works in five steps: first it initializes the optimization parameters, the robot population and the maximum number of iterations. Then, it performs the unconstrained solution search on the robots during the maximum 20% of iterations. The third step is to add the stresses on the robots to the calculation for an additional 40% of the computation. The next step refines the result obtained under a threshold set by the user. The last step is verified by simulation that the results obtained are correct and then validate them.

Sun et al. [15] developed an autonomous team of robots capable of coordinating and delivering boxes of goods on fixed stations in a warehouse. The robot is of a size of 50 by 50 cm possessing a weight of 60 kg as well as an holonomic command. It is equipped with LiDAR, odometry and inertial measurement unit sensors. The position of every robot is found by the law of Monte Carlo via the previous sensors. Robots synchronize together via local wireless communication. This swarm possesses eight types of behaviors:

Follow-up points of reference : the robot reunites them one after the other until it reaches its target position. If it is the case, another target will be allocated to it and it will begin again this action.

Avoiding: the robot bypasses the obstacle in its path and will continue to follow its landmarks.

Exchange: if there is a frontal collision, the two robots will bypass each other and then continue to track the marker afterwards.

Passing through: if a side collision occurs, the robot continues its way while the other waits for it to pass in front of it. Subsequently, it conducts the benchmark tracking.

Docking: the robot reaches its target and is placed in its intended location.

Waiting for a safe distance: the robot expects another robot and keeps a safe distance from it. When the other robot leaves the area, it resumes its normal activities.

Waiting to get through: following a side collision, the robot is waiting for the time the other robot passes in front of it. Then it continues its activities.

Waiting for docking: the robot must wait for another robot to finish mooring at the same dock.

All these behaviors allow the swarm to organize and carry out their tasks. The advantage of this algorithm is that it does not require a computational time to do trajectory planning such as Roads maps. It can work specifically in confined environments with obstacles.

5.1.2. Discussion

Table 2 shows a comparison of the previous algorithms. Deterministic algorithms are not widely used to move mobile robots swarm to the outside environment. This is because they have several inherent disadvantages to their design. Algorithms can meet different uses for the swarm of robots as long as the objective is clear. Their level of centralized swarm autonomy is less than the decentralized swarms of robots. This is due to the fact that the leader of the centralized swarm has to give commands to each of the robots in the swarm. Without these commands, the robots will not be able to achieve the task of the swarm. In a decentralized swarm of robots, robots communicate with each other and then distribute their tasks between themselves. This prevents some issues due to miscommunication between the leader and the swarm, and also allows the swarm to do difficult tasks. Nevertheless, centralized swarms can perform simple very well tasks because of their ease of implementation.

5.1.3. Probabilistic Algorithms

Husnawati et al. [4] use a combination of three algorithms to set up a swarm of mobile robots capable of detecting gas leaks. They propose to use as an algorithm the followings:

Fuzzy logic to control robots:

Each robot has three infrared sensors (front, left and right). The values of these are leveraged into the system to allow the robot to control its speed when an obstacle is present.

Swarm Optimization (PSO) particle algorithm:

This optimizes the trajectory planning of robots. If a gas leak is detected by a robot, the algorithm will lead the robot to its source. Otherwise the robots move freely in the area to be explored.

Algorithm support vector machines (SVM):

This is used to detect a gas leak with MQ3 (Alcohol Vapor) and MQ5 (LPG, Natural Gas, Town Gas) sensors.

The combination of these algorithms boosts the performance of robots in locating a gas leak.

Hacohen et al. [11] developed a probabilistic navigation algorithm for mobile robots. The positions of all objects are considered to be random variables. The purpose of the algorithm is to focus on the probability of localization of different objects (robots, obstacles, targets). Objects can have a different geometry of a point (circle/disc of a fixed radius), which changes their probability of location. In addition, priority values can also be attached to targets, which further changes their localization probability. To move robots, the algorithm performs several iterations. At each iteration, a probability map of the location of objects is updated. A gradient descent of the map probabilities is carried out to direct the robots towards their targets. Simulations have shown that this solution can be applied to real-time problems.

Table 2. Comparison of the different deterministic algorithms for centralized swarm.

| Skills | Vaidis and Otis [47] | Qin et al. [49] | Araki et al. [19] | Wei et al. [52] | Vatamaniuk et al. [54] | Garzon et al. [8] | Liu et al. [24] | Radu-Emil Precup et al. [56] | Sun et al. [15] |
|--------------------------------|----------------------|-----------------|-------------------|-----------------|------------------------|-------------------|-----------------|------------------------------|-----------------|
| Swarm with leader | ✓ | | | | | | ✓ | | |
| Local intercommunication | | | | | | | ✓ | | |
| Motion in outdoor environment | | ✓ | ✓ | | ✓ | ✓ | | | |
| Static obstacles avoidance | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Dynamic obstacle avoidance | ✓ | ✓ | | | | | | | |
| Control of the swarm form | ✓ | | | | ✓ | | ✓ | | ✓ |
| Map of the environment | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Storing the different motion | ✓ | | | | | ✓ | | | |
| Different types of robots used | | | ✓ | | | | | | |
| Simulated | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| Real-life experience | ✓ | | ✓ | | | ✓ | | | ✓ |

Bandyopadhyay et al. [23] proposed a new way to plan the movement of a very large swarm of mobile robots by keeping a precise formation (Probabilistic Swarm Guidance using inhomogeneous Markov Chains). A heterogeneous matrix of Markov with a desired stationary distribution is implemented using feedback based on Hellinger's distance. This matrix satisfies the travel constraints, minimizes the cost of transitions at each moment and distributes the number of robots where it is lacking. Simulations were conducted to compare algorithm performance with others. It turns out that it reduces the transition costs by 16 compared to a homogeneous Markov chain algorithm (HMC). Experimentations were also conducted with three to five quadrotors. In their other work, Bandyopadhyay et al. [18] improved the robot control part by adding an algorithm based on the Voronoï graph algorithm. It was successfully tested.

In their work, Nurmaini et al. [57] developed a fuzzy logic algorithm that allows a swarm to move. The robots are equipped with three infrared sensors used for obstacle detection. A CCD camera is used for experimenting and allows seeing the position of the robots and their orientation. Each robot can be identified by its color (in the tests: red, green, blue). All this information is given at the input of the fuzzy logic block which sends out the engine speed (in translation and rotation) for each robot. This allows them to reach the target position they have received.

Finally, Chang et al. [28] developed a trajectory planning algorithm for swarms of robots subject to disturbance flows. Their objective is to find the source of the flow and lead the swarm. First, they look at the mathematical representation of a chemical plume and these characteristics. Then the problem of going back to the source is posed. The swarm is made up of a finite number of mobile robots. A marker is defined and the speed of each robot can be found in it. Once this is done, the trajectory planning takes place in three steps:

1. Measuring the turbulence of the flow over a small period of time.
2. Estimating based on probability of distance to source: the speed of the different robots is then defined for the trajectory planning.
3. Moving robots for a short period of time.

Simulations confirmed the validity of this algorithm based on blue crabs. The waiting time between each decision-making has a great importance on the behavior of robots. The bigger it is, the more robots will go directly in the right direction to find the source.

5.1.4. Discussion

Table 3 shows a comparison of the previous explained algorithms. Probabilistic algorithms of centralized swarms rely little on the use of maps to locate themselves. They mainly use distance sensor data to learn about their environment and can plan their route. They are not very good at avoiding dynamic obstacles or controlling swarm formation.

5.1.5. Heuristic Algorithms

Sharma et al. [58] use a new Lyapunov function acting as a field of artificial potential to control a swarm of mobile robots. Their contributions relate to:

1. Avoidance of a swarm of moving obstacles.
2. Design of a heterogeneous robotic system in a closed environment with obstacles.
3. Control laws for the non-linear heterogeneous robotic system and invariant according to its accelerations.

The swarm of mobile robots should therefore be able to avoid the other swarm of obstacles. The artificial potential field represents the energy of the system and the forces generated by it or on it. The goal is to minimize this function. The result is a translation and rotational control for the swarm robots. Simulations were made to validate the functioning of the algorithm.

Roy et al. [59] compare two algorithms so that their swarm of mobile robots can move around avoiding obstacles: bacterial foraging and particle Swarm Optimization.

Functions designating the purpose to be achieved and the obstacles to be avoided are defined. Another function defining time errors is then set from the previous two. The purpose of both algorithms is to minimize this function. To do this, the swarm must first move in a coordinated way, i.e., each robot must have about the same average speed as well as the same average direction. The control of the swarm must then be defined autonomously. Simulations show that the first algorithm is more concerned with maintaining the formation of the swarm, while the second optimizes its movement.

Table 3. Comparison of different probabilistic algorithm for centralised swarm.

| Skills | Husnawati et al. [4] | Hacohen et al. [11] | Bandyopadhyay et al. [18,23] | Nurmaini et al. [57] | Chang et al. [28] |
|------------------------------------|----------------------|---------------------|------------------------------|----------------------|-------------------|
| Swarm with leader | | | | | |
| Local communication between robots | | | | | |
| Motion in outdoor environment | ✓ | ✓ | ✓ | | ✓ |
| Static obstacle avoidance | ✓ | ✓ | ✓ | ✓ | ✓ |
| Dynamic obstacle avoidance | ✓ | | | | |
| Control of the swarm form | | | ✓ | ✓ | |
| Map of the environment | ✓ | ✓ | | | |
| Storing the different motion | | | | | |
| Different types of robots used | | | | | |
| Simulated | ✓ | ✓ | ✓ | | ✓ |
| Real-life experience | ✓ | | ✓ | ✓ | |

In their work, Jann et al. [60] use the D*lite algorithm [61] to obtain a mobile robots swarm through an obstacle field. Several checkpoints are defined in the obstacle zone and the robots must go through one of them. Once it has passed, it goes into closed mode and no robots are allowed to return to it. The algorithm already possesses information on the map and then updates itself when moving the robots. A cost function is defined based on the cost of moving the robot between two nodes of the map, as well as the heuristic cost of travel. The purpose of the algorithm is to minimize this function. Several simulations were carried out with different changing parameters: the number of vehicles, static or dynamic obstacles. In all cases, the robots were able to reach their objective without hindrance. Trajectory planning is highly dependent on the disposition of obstacles as well as the grid used.

Devi et al. [62] used gorilla behavior to create an algorithm for moving a swarm of mobile robots. In this algorithm, three behaviors are possible:

Action of climbing/moving: the gorilla will move to an elevation position that will allow it to have an overview of its environment.

Observation of an easier path: once the gorilla has reached a peak, it observes the surroundings in order to find a higher point to reach it.

Jumping: the gorilla changes position by rotating forward or backward to the new higher point of view.

In the algorithm, the highest point to be reached is assimilated to the target position that the robots will have to reach. The robots will perform each iteration of the algorithm (three steps). However the path obtained will not be optimal. This is why the algorithm is linked to the open vehicle routing problem (OVRP). Simulations validated the operation of this algorithm.

Zhang et al. [10] developed in their work an algorithm based on the model of a simplified virtual force for moving a swarm of mobile robots to help with hunting. This model prevents obstacles and robots from colliding with each other. The purpose of this algorithm is to evenly distribute robots on a circle around a target. The robots follow the contour of the circle and stand one by one at the coordinates assigned to them. Several simulations were carried out in environments with or without obstacles to verify the proper functioning of the algorithm. The advantage of this method is that it avoids local minimum problems.

Caska et al. [63] use an algorithm whose purpose is to compute the number of drones and mobile robots composing a swarm in order to cover all the landmarks of a surveillance

zone, and also to plan their trajectory optimally. As a first step, the algorithm defines coordinated points to be reached for vehicles on the ground and for drones. Then it calculates the greatest distance to travel between the previous points, taking into account the climb or descent of a slope. A computation of the energy consumption is then carried out to determine whether the vehicle and the drone can carry out the distance without any problems. If so, a drone and vehicle will suffice. Otherwise the algorithm proposes to increase the number of vehicles and drones until the energy consumption is sufficient to carry out the journeys. It is estimated that each robot and drone can travel three kilometers at full load and starting with a full charge. A genetic algorithm was also used to compute the optimal solution to this problem.

Wallar et al. [26] proposed to combine two types of algorithms in order to move a swarm of mobile robots in a congested and dynamic environment: Roadmaps Probabilistic and potential fields. The roadmaps are used to carry out an overall planning of the path of the swarm to its target position. The global trajectory search is chosen by the potential field algorithm that allows mobile robots to avoid collisions with obstacles or with other robots. Simulations have demonstrated the validity of this combination of algorithms. It can work for a hundred robots and at least fifty dynamic obstacles.

Agrawal et al. [64] developed an algorithm based on ant colonies so that the mobile robots swarm can move without collisions. This algorithm makes it possible to find the shortest path between the swarm and the desired target. It is based on the deposit of pheromones and the probability that one robot will choose one path over another. The algorithm will browse the map ahead for robots following several trajectories. The shorter a trajectory is, the more important will be the deposition of pheromone. This will increase the probability that this path will be chosen. In the end, this path will be chosen to lead the robot. Each path found for these will be added as you go on the obstacle map. Simulations were performed to validate the functioning of the algorithm.

Vicmudo et al. [65] used genetic algorithms to direct their swarm of underwater robots. They initialize the algorithm with random positions as the starting population. Chromosomes were used to contain all the robots movement coordinates. When the initial population changes, the chromosomes will be sorted according to the sum of the distances they will contain to get to the target. If this distance is too great, the chromosome will be removed. If two robots were to have the same position during the algorithm, a penalty is given to the chromosomes. Three different simulations were conducted with several starting populations (150, 250 and 500). The conclusions are that the larger the initial population, the more the algorithm will converge towards the optimal solution. This method is able to plan the trajectory of robots moving in swarms.

Hedjar et al. [66] used a collision avoidance algorithm for mobile robots swarm. It creates a safety ring around the robot that prevents it from moving towards the obstacle if the ring is in it. The ring is capable of adapting to several types of robot shapes. In addition to this, trajectory planning is achieved using convex optimization of a nonlinear equation system. A cost function is defined for each route of the robots. This must be minimized to plan their route. Each robot considers the other robots as dynamic obstacles. Simulations and experiments were conducted to validate this model. Using convex optimization prevents local minimum problems. In addition, this algorithm is capable of being integrated into centralized and decentralized robots swarm systems. Additionally, the position of the obstacles must be known in advance. Otherwise, you have to add to the system a means of detecting them.

Dang et al. [17] developed a control algorithm for a swarm of mobile robots using artificial potential fields combined with a rotary vector field. This allows each robot of the swarm to move towards a target position while retaining its formation. Repellent potential is defined for obstacles and attractive potential is given to the objective to be achieved. The rotary vector field is used to avoid oscillation problems. An attractive force is defined so that robots can maintain their formation. Simulations were performed to validate the functioning of the algorithm.

5.1.6. Discussion

A comparison of the previous algorithms is given in Table 4. The advantage of heuristic algorithms is that they allow the swarms of centralized robots to move in difficult outdoor environments. Indeed, most of them are combinations of different algorithms that allow them to eliminate the disadvantages of each of them. All are based on a map to complete the trajectory planning. They also do not need robots to communicate with each other. Some other research works present algorithm for three-dimensional path planning using cuckoo search algorithm with Levy flight in a random way [67]. However, this algorithm needs to be improved for a swarm of rovers.

Table 4. Comparison of different heuristic algorithms for centralized swarm.

| Skills | Sharma et al. [58] | Roy et al. [59] | Jann et al. [60] | Devi et al. [62] | Zhang et al. [10] | Caska et al. [63] | Wallar et al. [26] | Agrawal et al. [64] | Vicmudo et al. [65] | Hedjar et al. [66] | Dang et al. [17] |
|--------------------------------|--------------------|-----------------|------------------|------------------|-------------------|-------------------|--------------------|---------------------|---------------------|--------------------|------------------|
| Swarm with leader | | | | | | | | | | | |
| Local communication | | | | | | | | | | | |
| Motion in outdoor environment | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Static obstacle avoidance | ✓ | | ✓ | | | | ✓ | | | ✓ | |
| Dynamic obstacle avoidance | | ✓ | | | ✓ | | | | | | |
| Control of the swarm form | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Map of the environment | | | ✓ | | | | | ✓ | | | |
| Storing the different motion | | | | | | ✓ | | | | | |
| Different types of robots used | | | | | | | | | | | |
| Simulated | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Real-life experience | | | | | | | | | | ✓ | |

5.2. Undistributed Decentralized Swarm

A decentralized swarm does not have one leader. Instead, it uses its multiple robots as leader, each of which usually stores a copy of data of the other robots to take a decision. A decentralized system can be just as vulnerable to issues as a centralized one. However, by design, there are more tolerant and robust due to the fact that robots have their own information to take decision, and share them with others. A distributed system is similar to a decentralized swarm. The difference is the way robots share information between each other. In an undistributed and decentralized swarm, the information is not uniformly distributed. Some robots will have more information than others. This section is dedicated to this type of swarm.

5.2.1. Deterministic Algorithms

Aniketh et al. [5] developed an algorithm based on weights according to different situations to move a swarm of mobile robots in an environment with obstacles. The weights are fixed on the surrounding boxes of the robots. The travel direction chosen will be the one with the highest value. The value of the weights is 0 if there is an obstacle or a robot; 1 if the box has been explored; 4 if it is the target and 5 if the box has not been explored. The map is updated after every robot moves. Tests were performed with real robots. The algorithm runs quickly and allows you to quickly explore the entire map. The robots behave independently and can thus move on various types of terrain.

5.2.2. Probabilistic Algorithms

Mendonça et al. [68] developed an algorithm using dynamic Fuzzy cognitive maps [69]. Robots have several capabilities: mobility, autonomy, responsiveness, adaptability, collaboration and caring. Several basic rules are built around these capabilities. They allow robots to move according to the situations encountered. Each robot can then enter a particular state and does the actions associated with it: exploration, avoidance of obstacles, reaching object or target and reversing when there is an obstacle. Points are set between the transitions of the different states and the actions to be carried out. The learning of these rules is given to the robot using a method similar to Q-learning in order to find the weights of the

system. Simulations in a virtual environment were conducted to observe the results. The algorithm has yielded good results and allows the swarm of mobile robots to learn from situations encountered, to adapt and cooperate.

A. Belkadi et al. [70] used the Swarm Optimization particle algorithm [71] to direct their drone swarm. It acts like a decentralized swarm: drones have their own behavior and are independent. The goal is to minimize a cost function that will be used to optimize the drone's trajectory. The law of control is based on their quaternions. The algorithm can very well be implanted for mobile robots swarm. Tests with real drones were performed in different situations (without/with obstacles, number of drones).

Ayari et al. [72] used the Swarm Optimization particle algorithm to guide a swarm of mobile robots to its target. This algorithm has several key principles:

1. Defining a position in a space.
2. Assessing this position.
3. Associating one speed to this position to have the following.
4. Memorizing possible movements with this speed to find the best next position.
5. Selecting the following position.

Starting populations are initialized at random. The speed of the particles will be dependent on the previous best positions as well as on randomly selected variables. The algorithm stops when the maximum number of iterations is reached. This algorithm is combined with two other parameters to avoid maximum local problems for the best overall position and stop the algorithm when it converges. Collision management is performed by computing the distance between each obstacle and each robot. Simulations were conducted with static obstacles. These show that the algorithm is capable of properly directing the swarm of mobile robots in its environment.

Alam et al. [73] also proposed a Swarm Optimization particle algorithm so that the swarm can avoid sources of danger. In their work, the algorithm first calculates the distance between the starting distance of the robots and that of their lens, and then draws a line between these two points. The map is then cut into a finite number of sections. If there are no obstructions in the sections, a reference point is attached to the intersection of the right to the target and the right to the section. Otherwise the Swarm Optimization particle algorithm looks for the smallest distance that will allow the robots to bypass the obstacle. The algorithm will successively perform this method for each of the swarm robots. Simulations in different environments have demonstrated the validity of the algorithm. It could only be tested for static obstacles.

Das et al. [22] chose to improve the Swarm Optimization particle algorithm for the trajectory planning of a mobile robots swarm. They developed a method to adapt the weights and accelerations of the coefficients of the algorithm to increase its rate of convergence. It works according to the following steps:

1. The robot knows its current position and that of its target.
2. They look towards their target to see if there are obstacles or not: if they do, they make the decision to shoot.
3. If there are no obstacles, it goes to the target.

The planned path is determined by the improved algorithm. Simulations and experiments have shown that it allows several robots to move in an environment with static obstacles. It could not be used for dynamic obstacles.

Sharma et al. [74] proposed a new algorithm capable of directing a swarm of mobile robots to carry out area exploration. It starts by dividing the environment into several partitions. Each will be assigned to a robot to explore. The path planning of each robot is done by the Swarm Optimization particle algorithm. The method of moving them can be in two ways: either it is random or zig-zag. The aim is, of course, to travel as quickly as possible through the area to be explored. Several parameters are taken into account and are computed: the distance of movement at each iteration, the energy consumed, the coverage performed and the time to perform this coverage. Simulations were conducted to validate

the functioning of the algorithm. Its performance depends on the number of robots used as well as the type of direction to be taken.

Luo et al. [21] developed a swarm of mobile robots capable of moving to a target. They used the Golden Shiner Fish movement [75] to design their system. The displacement of robots is therefore influenced by several factors of their environment that change their speed and direction of travel. These factors are the brightness and presence of robots in their vicinity. These are detected by measuring the force of their transmission signal by three antennas located on the robot. They show that robots are able to reach a darker area that is their target.

5.2.3. Heuristic Algorithms

In their works, Zelenka et al. [16] present a method to create a swarm of mobile robots that is decentralized and can adapt its form with the aim of exploring a zone. The algorithm bases itself on the use of artificial pheromones. Robots travel into their environment and store the information perceived on a map which will then be transmitted in all the swarm. The zone to be explored is divided into cells. As soon as a robot explores one of them, it leaves a pheromone to indicate its passage and sends on the information to the other robots. The motion of every robot is dictated by several rules: the robot moves towards a cell possessing the least possible pheromones. If several cells possess the same quantity, the robot chooses it randomly. This method makes it possible to add several robots during the operation in order to cover the area to be explored more easily. It also anticipates the optimal number of robots and removes some if they are too many. Simulations were conducted to test its validity.

Del Ser et al. [76] used bats to design a trajectory planning algorithm for mobile robots. This is based on the echolocation of obstacles by robots. In their case, each robot moves randomly at a certain speed. Sound wave emission is done at a fixed frequency, varying wavelengths and intensity. At each iteration of the algorithm, the values of the robots' speed, the wavelength and the intensity of the sound wave used are modified randomly according to a uniform distribution. Trajectory planning is also done at random while taking into account the obstacles detected by the robot. Simulations and experiments were carried out with small mobile robots. The algorithm allows them to move well within the area to be explored. Despite this, robots may find themselves trapped in particular wall shapes (U or V wall).

Contreras-Cruz et al. [13] applied an algorithm based on the honey-bee colonies [77] to manage their swarm of mobile robots. The difficulty is to determine when there will be a possible collision between robots. For that purpose, the algorithm decomposes into two parts: one, planning of paths and two, coordination. The first part takes care to generate paths by associating their levels of priority according to their time of motion. The second part manages the speed of robots according to the obstacles and to the level of priority of trajectories. It is implemented by the algorithm of the honey-bee colonies. It works as follows: each robot predicts the future position of the other robots from the information of the previous iteration. If a collision is detected, the robot is put on hold while the danger passes. It establishes another trajectory planning and sends the information to other robots with a low probability of collisions. At the end of an iteration, all robots communicated their future route plan in order to synchronize their movement. On the next iteration, it begins again. Simulations were carried out to validate its operation.

Ardakani et al. [14] developed an artificial potential field method for controlling mobile robots capable of moving plates in an environment with obstacles. The robots have to coordinate to move the plate together. The forces on the robots and this one were modeled to predict the optimal control to be carried out. A potential field algorithm is then used to plan the path of the swarm robots. It allows for the avoidance of obstacles and to reach the targets of the robots. Tests were carried out by real mobile robots. The algorithm is capable of adjusting to different forms of plates, in particular by modifying the formation of the swarm and the speed of the robots.

Jabbarpour et al. [29] developed a swarm algorithm (ant-based method) of mobile robots that seeks to minimize their energy consumption when moving. This method is based on that of ant colonies using pheromones. An energy consumption model was developed according to the control parameters. The entire algorithm consists of four steps:

1. A phase of exploration in which robots collect and memorize information about their environment.
2. The second phase consists of computing the energy of the trips to be made for each trajectory planning.
3. The third concerns the exploration phase of the map defined in the first stage.
4. The last step determines the path to be taken for the robot. The decision is based on the path with the most pheromone.

Simulations were performed and the results were compared with the PSO and ant colony algorithms. The performance is better than these two algorithms based on the distance of the journey and the time of execution of the algorithm.

Fricke et al. [9] based their algorithm on a method called Lévy [78] to allow a swarm of mobile robots to explore an area. The aim of this method is to optimize the target search by playing on the intensity of the searches and the distance traveled by the robots. This involves cutting each robot's journey into several stages defined by a small-time interval. Each robot randomly selects a direction according to a uniform distribution and travels to it during the time interval. At the end of this one, the robot restarts the process. If it encounters an obstacle, it changes its direction as done previously. The algorithm is inspired by the movement of T cells in a human being.

Shi et al. [79] applied a combination of pheromone algorithms and Q-learning to optimize the movement of a mobile robots swarm. A comparison with the Swarm Optimization particle algorithm is performed. The Q-learning is based on Markov's decision chain algorithm [80]. At each iteration, the robot observes its environment, then chooses an action according to its possibilities. It then proceeds to the next iteration, learning whether it was good or not. The study focuses on learning an optimal strategy of all the actions carried out. The contribution of this article concerns the contribution of pheromones during the learning of actions. This allows the algorithm to explore more terrain and share more information between different robots. It has been tested on several labyrinth maps and compared to the PSO algorithm, indicating that it is more efficient.

5.2.4. Discussion

A comparison of the previous algorithms is given in Table 5. Most of the algorithms presented for the swarms of non-distributed decentralized mobile robots can work in an outdoor environment. Few are able to avoid dynamic barriers, which can be problematic in such environments. The vast majority use a map to move it. It has the advantage of representing obstacles and thus allows swarms to avoid them. In some cases it is also used to memorize the movement of robots for it not to happen again. The task performed by robots of the same swarm is always the same for all, most of the time exploring an area. The swarms following this provision have a very high level of autonomy. All they need is a goal to achieve.

5.3. Distributed Decentralized Swarm

This last section is dedicated to distributed and decentralized swarms. Few swarms work according to this type of communication. This is due to the difficulties to share uniformly information between all the robots. Indeed, the means of communication are usually huge constraints to share information, especially in difficult environment. This section will present the two papers on this type of swarm.

Table 5. Comparison of different algorithms for decentralized and undistributed swarm.

| Skills | [5] | [68] | [70] | [72] | [73] | [22] | [74] | [21] | [16] | [76] | [13] | [14] | [29] | [9] | [79] |
|------------------------------------|-----|------|------|------|------|------|------|------|------|------|------|------|------|-----|------|
| Swarm with leader | | | | | | | | | | | | | | | |
| Local communication between robots | | | | | | | | ✓ | ✓ | | ✓ | ✓ | | ✓ | ✓ |
| Motion in outdoor environment | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ |
| Static obstacle avoidance | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Dynamic obstacle avoidance | | ✓ | | | | | | ✓ | | ✓ | | | | | |
| Control of the swarm form | | | | | | | | | | | | ✓ | | | |
| Map of the environment | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ |
| Storing the different motion | ✓ | | | | | | | | ✓ | | | | | | |
| Different types of robots used | | | | | | | | | | | | | | | |
| Simulated | | ✓ | | ✓ | ✓ | ✓ | ✓ | | ✓ | | ✓ | | ✓ | ✓ | ✓ |
| Real-life experience | ✓ | | ✓ | | | ✓ | | ✓ | | ✓ | | ✓ | | ✓ | |

5.3.1. Deterministic Algorithms

In their work, Hattori et al. [20] developed a generalized measuring-worm (GMW) algorithm for map creation and position estimation that is decentralized and allows robots to do separate tasks. This is an upgrade to the SLAM algorithm [81]. It proposes to estimate the position of a robot with fewer resources and calculate its displacement. The robots are divided into two classes: one is designated as the parent and the other as the son. The robots are both equipped with a camera and markers. The father robot receives the coordinates to be reached and travels to them. The son robot then tries to follow the father robot by estimating the position of the robot thanks to the camera of his own marker. Robots regularly communicate their data to each other to synchronize.

Seng et al. [25] offer an algorithm that can move a swarm of mobile robots while retaining their formation. It is divided into two stages: the first allows the swarm to maintain the formation without the robots exchanging information with each other, and the second involves the planning of the trajectories of the different robots. Each of them can perform collision avoidance by their own means, but an algorithm has been added to keep the formation of the swarm. One robot is considered the leader, while the others are to follow it and maintain the formation. Experimentations were conducted to validate the method. This gives a good result and a very high robot placement accuracy.

5.3.2. Discussion

A comparison of the previous algorithms is given in Table 6. There are a few algorithms for decentralized and distributed mobile robots swarm. This is due to the fact that most robots perform the same task within the swarm. The two algorithms presented differ from this case since the robots have two different behaviors: leaders (father/mother) and followers (son/daughter). This leads to a few contexts for use in real life especially because of the difficulty in implementing the system, including disturbances from the environment. The robots are autonomous in their movement as long as the target is indicated for the swarm.

Table 6. Comparison of the different algorithms for decentralized and distributed swarm.

| Skills | Hattori et al. [20] | Seng et al. [25] |
|------------------------------------|---------------------|------------------|
| Swarm with leader | ✓ | ✓ |
| Local communication between robots | ✓ | ✓ |
| Motion in outdoor environment | | |
| Static obstacle avoidance | ✓ | ✓ |
| Dynamic obstacles avoidance | | |
| Control of the swarm form | | ✓ |
| Map of the environment | | ✓ |
| Storing the different motion | | |
| Different types of robots used | | |
| Simulated | | |
| Real-life experience | ✓ | ✓ |

6. Conclusions and Future Works

Through this survey, we were able to present the different types of physical support for interacting with a swarm of robots and detail the operation of existing algorithms for moving them into an open and crowded space.

First of all, we saw the different advantages and disadvantages of human-swarm interaction media. The choice of an interacting medium depends above all on the intended use of the swarm in order to facilitate the operator's control of the swarm. It also revealed that the autonomy of the swarm was more or less affected; it could not obtain a complete autonomy because the operator must always give an objective to be attained. The various types of algorithms existing for the path planning and navigation are presented in this literature review. The realized taxonomy allows seeing certain peculiarities of the functioning of the algorithms. Additionally, it is necessary to choose the algorithms according to the action that the swarm wants to make. It is noticed that there is lack of distributed and decentralized swarm. It could be due to the fact that it is still difficult to design algorithms for this application.

Future work may have several lines of research. First, the operator should be allowed to send implicit orders to the swarm via a chosen interaction medium. The operator would do his job and the swarm would all understand the action. Then it would be fully self-sustaining. Second, research can be carried out on the swarms of decentralized and distributed mobile robots. As we have seen, little research has been done in this area, and there is limited research on possible applications. The main interest of this research would be to design a swarm capable of performing and distributing tasks to its robots in autonomous ways, while controlling its formation and trajectory planning.

Author Contributions: Conceptualization, M.J.-D.O.; methodology, M.V.; formal analysis, M.J.-D.O. and M.V.; investigation, M.V.; resources, M.J.-D.O.; data curation, M.V.; writing—original draft preparation, M.V. and M.J.-D.O.; writing—review and editing, M.V. and M.J.-D.O.; supervision, M.J.-D.O.; project administration, M.J.-D.O.; funding acquisition, M.J.-D.O. Both the authors have read and agreed to the published version of the manuscript.

Funding: While performing this project, Maxime Vaidis received a scholarship from REPARTI Strategic Network supported by Fonds québécois de la recherche sur la nature et les technologies (FRQ-NT). This research was funded by Natural Sciences and Engineering Research Council of Canada (NSERC), Discovery grant, under Grant Number RGPIN-2018-06329.

Acknowledgments: We would like to thank the Department of Applied Sciences, UQAC, Canada for allowing access to the rovers to the LAR.i Laboratory. Francis Deschênes and Danny Ouellet gave us precious advices related to the technical design and maintenance of the rovers.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kolling, A.; Walker, P.; Chakraborty, N.; Sycara, K.; Lewis, M. Human Interaction With Robot Swarms: A Survey. *IEEE Trans. Hum. Mach. Syst.* **2016**, *46*, 9–26. [[CrossRef](#)]
2. Nedjah, N.; Luneque, S., Jr. Review of methodologies and tasks in swarm robotics towards standardization. *Swarm Evol. Comput.* **2019**, *50*. [[CrossRef](#)]
3. Brambilla, M.; Ferrante, E.; Birattari, M. Swarm robotics: A review from the swarm engineering perspective. *Swarm Intell.* **2013**, *7*, 1–41. [[CrossRef](#)]
4. Husnawati, G.; Fitriana, F.; Nurmaini, S. The development of hybrid methods in simple swarm robots for gas leak localization. In Proceedings of the International Conference on Signals and Systems, Bali, Indonesia, 16–18 May 2017; pp. 197–202.
5. Aniketh, R.; Manohar, E.B.; Yazwa, G.R.S.P.R.; Nithya, M.; Rashmi, M.R. A decentralized fault-tolerant weights based algorithm for coordination of swarm robots for a disaster scenario. In Proceedings of the 2016 IEEE Annual India Conference, Bangalore, India, 16–18 December 2016.
6. Senanayake, M.; Senthoran, I.; Barca, J.C.; Chung, H.; Kamruzzaman, J.; Murshed, M. Search and tracking algorithms for swarms of robots: A survey. *Robot. Auton. Syst.* **2016**, *75*, 422–434. [[CrossRef](#)]
7. Saeedi, S.; Trentini, M.; Seto, M.; Li, H. Multiple-Robot Simultaneous Localization and Mapping: A Review. *J. Field Robot.* **2016**, *33*, 3–46. [[CrossRef](#)]

8. Garzón, M.; Valente, J.; Roldán, J.J.; Cancar, L.; Barrientos, A.; Del Cerro, J. A Multirobot System for Distributed Area Coverage and Signal Searching in Large Outdoor Scenarios. *J. Field Robot.* **2016**, *33*, 1087–1106. [[CrossRef](#)]
9. Fricke, G.M.; Hecker, J.P.; Cannon, J.L.; Moses, M.E. Immune-inspired search strategies for robot swarms. *Robotica* **2016**, *34*, 1791–1810. [[CrossRef](#)]
10. Zhang, H.; Zhang, J.; Zhou, S.; Ouyang, P.; Wu, L. Hunting in Unknown Environments with Dynamic Deforming Obstacles by Swarm Robots. *Int. J. Control. Autom.* **2015**, *8*, 385–406. [[CrossRef](#)]
11. Hacoheh, S.; Shoval, S.; Shvalb, N. Multi agents' multi targets mission under uncertainty using probability navigation function. In Proceedings of the IEEE International Conference on Control and Automation, Ohrid, Macedonia, 3–6 July 2017; pp. 845–850.
12. Gutierrez, M.A.; Nair, S.; Banchs, R.E.; Enriquez, L.F.D.; Niculescu, A.I.; Vijayalingam, A. Multi-robot collaborative platforms for humanitarian relief actions. In Proceedings of the IEEE Region 10 Humanitarian Technology Conference, R10-HTC, Cebu, Philippines, 9–12 December 2015.
13. Contreras-Cruz, M.A.; Lopez-Perez, J.J.; Ayala-Ramirez, V. Distributed path planning for multi-robot teams based on Artificial Bee Colony. In Proceedings of the 2017 IEEE Congress on Evolutionary Computation, Donostia, Spain, 5–8 June 2017; pp. 541–548.
14. Ardakani, E.S.; Ebel, H.; Eberhard, P. Transporting an elastic plate using a group of swarm mobile robots. In Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Munich, Germany, 3–7 July 2017; pp. 1393–1398.
15. Sun, D.; Kleiner, A.; Nebel, B. Behavior-based multi-robot collision avoidance. In Proceedings of the IEEE International Conference on Robotics and Automation, Hong Kong, China, 31 May–7 June 2014; pp. 1668–1673.
16. Zelenka, J.; Kasanický, T.; Budinská, I. A self-adapting method for 3D environment exploration inspired by swarm behaviour. *Mech. Mach. Sci.* **2018**, *49*, 493–502.
17. Dang, A.D.; Horn, J. Path planning for a formation of autonomous robots in an unknown environment using artificial force fields. In Proceedings of the 2014 18th International Conference on System Theory, Control and Computing, Sinaia, Romania, 17–19 October 2014; pp. 773–778.
18. Bandyopadhyay, S.; Chung, S.J.; Hadaegh, F.Y. Probabilistic and Distributed Control of a Large-Scale Swarm of Autonomous Agents. *IEEE Trans. Robot.* **2017**, *33*, 1103–1123. [[CrossRef](#)]
19. Araki, B.; Strang, J.; Pohorecky, S.; Qiu, C.; Naegeli, T.; Rus, D. Multi-robot path planning for a swarm of robots that can both fly and drive. In Proceedings of the IEEE International Conference on Robotics and Automation, Singapore, 29 May–3 June 2017; pp. 5575–5582.
20. Hattori, K.; Homma, E.; Kagawa, T.; Otani, M.; Tatebe, N.; Owada, Y.; Shan, L.; Temma, K.; Hamaguchi, K. Generalized measuring-worm algorithm: High-accuracy mapping and movement via cooperating swarm robots. *Artif. Life Robot.* **2016**, *21*, 451–459. [[CrossRef](#)]
21. Luo, E.; Fang, X.H.; Ng, Y.; Gao, G.X. Shinerbot: Bio-inspired collective robot swarm navigation platform. In Proceedings of the 29th International Technical Meeting of the Satellite Division of the Institute of Navigation, Portland, OR, USA, 12–16 September 2016; Volume 2, pp. 1091–1095.
22. Das, P.K.; Sahoo, B.M.; Behera, H.S.; Vashisht, S. An improved particle swarm optimization for multi-robot path planning. In Proceedings of the 2016 1st International Conference on Innovation and Challenges in Cyber Security, Greater Noida, India, 3–5 February 2016; pp. 97–106.
23. Bandyopadhyay, S.; Chung, S.J.; Hadaegh, F.Y. A probabilistic Eulerian approach for motion planning of a large-scale swarm of robots. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Daejeon, Korea, 9–14 October 2016; pp. 3822–3829.
24. Liu, Y.-C. Task-space coordination control of bilateral human-swarm systems. *J. Frankl. Inst.* **2015**, *352*, 311–331. [[CrossRef](#)]
25. Seng, W.L.; Barca, J.C.; Sekercioglu, Y.A. Distributed formation control of networked mobile robots in environments with obstacles. *Robotica* **2014**, *34*, 1403–1415. [[CrossRef](#)]
26. Wallar, A.; Plaku, E. Path planning for swarms by combining probabilistic roadmaps and potential fields. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2014; Volume 8069, pp. 417–428.
27. Kim, H.C.; Kim, J.S.; Ji, Y.K.; Park, J.H. Path planning of swarm mobile robots using firefly algorithm. *J. Inst. Control. Robot. Syst.* **2013**, *19*, 435–441. [[CrossRef](#)]
28. Chang, D.; Wu, W.; Webster, D.R.; Weissburg, M.J.; Zhang, F. A bio-inspired plume tracking algorithm for mobile sensing swarms in turbulent flow. In Proceedings of the IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; pp. 921–926.
29. Jabbarpour, M.R.; Zarrabi, H.; Jung, J.J.; Kim, P. A Green Ant-Based method for Path Planning of Unmanned Ground Vehicles. *IEEE Access* **2017**, *5*, 1820–1832. [[CrossRef](#)]
30. Bowley, S.J.; Merrick, K. A 'Breadcrumbs' Model for Controlling an Intrinsically Motivated Swarm Using a Handheld Device. In Proceedings of the AI 2017: Advances in Artificial Intelligence, Melbourne, Australia, 19–20 August 2017; pp. 157–168.
31. Crandall, J.W.; Anderson, N.; Ashcraft, C.; Grosh, J.; Henderson, J.; McClellan, J.; Neupane, A.; Goodrich, M.A. Human-swarm interaction as shared control: Achieving flexible fault-tolerant systems. In Proceedings of the EPCE: International Conference on Engineering Psychology and Cognitive Ergonomics, Vancouver, BC, Canada, 9–14 July 2017; pp. 266–284.

32. Kim, M.S.; Kim, S.H.; Kang, S.J. Middleware design for swarm-driving robots accompanying humans. *Sensors* **2017**, *17*, 392. [CrossRef]
33. Ferrer, E.C. A wearable general-purpose solution for Human-Swarm Interaction. *arXiv* **2017**, arXiv:1704.08393.
34. Mendes, J.J.A., Jr.; Freitas, M.L.B.; Siqueira, H.V.; Lazzaretti, A.E.; Pichorim, S.F.; Stevan, S.L. Feature selection and dimensionality reduction: An extensive comparison in hand gesture classification by sEMG in eight channels armband approach. *Biomed. Signal Process. Control.* **2020**, *59*, 101920. [CrossRef]
35. LeapMotion Developer. Available online: <https://developer.leapmotion.com/> (accessed on 8 June 2021).
36. McDonald, S.J.; Colton, M.B.; Alder, C.K.; Goodrich, M.A. Haptic Shape-Based Management of Robot Teams in Cordon and Patrol. In Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction—HRI'17, Vienna, Austria, 6–9 March 2017.
37. Kapellmann-Zafra, G.; Chen, J.; Groß, R. Using google glass in Human-Robot swarm interaction. In Proceedings of the Towards Autonomous Robotic Systems. Towards Autonomous Robotic Systems (TAROS 2016), Sheffield, UK, 26 June–1 July 2016; pp. 196–201.
38. Mondada, L.; Karim, M.E.; Mondada, F. Electroencephalography as implicit communication channel for proximal interaction between humans and robot swarms. *Swarm Intell.* **2016**, *10*, 247–265. [CrossRef]
39. Akhtar, A.; Norton, J.J.; Kasraie, M.; Bretl, T. Playing checkers with your mind: An interactive multiplayer hardware game platform for brain-computer interfaces. In Proceedings of the 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Chicago, IL, USA, 26–30 August 2014; pp. 1650–1653.
40. Wu, C. H.; Lakany, H. The effect of the viewing distance of stimulus on SSVEP response for use in brain—Computer interfaces. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC), Manchester, UK, 13–16 October 2013; pp. 1840–1845.
41. Setter, T.; Kawashima, H.; Egerstedt, M. Team-level properties for haptic human-swarm interactions. In Proceedings of the American Control Conference, Chicago, IL, USA, 1–3 July 2015; pp. 453–458.
42. Podevijn, G.; O'Grady, R.; Nashed, Y.S.G.; Dorigo, M. Gesturing at subswarms: Towards direct human control of robot swarms. In Proceedings of the Towards Autonomous Robotic Systems, Oxford, UK, 28–30 August 2013; pp. 390–403.
43. Kolling, A.; Sycara, K.; Nunnally, S.; Lewis, M. Human Swarm Interaction: An Experimental Study of Two Types of Interaction with Foraging Swarms. *J. Hum. Robot. Interact.* **2013**, *2*, 103–129. [CrossRef]
44. Bullo, F.; Cortés, J.; Martinez, S. *Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms*; Princeton University Press: Princeton, NJ, USA, 2009.
45. Diana, M.; De La Croix, J.P.; Egerstedt, M. Deformable-medium affordances for interacting with multi-robot systems. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 5252–5257.
46. Alessandro Giusti, J.N.; Gambardella, L.M.; Bonardi, S.; Di Caro, G.A.; Human-Swarm Interaction through Distributed Cooperative Gesture Recognition. In Proceedings of the 2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI), Boston, MA, USA, 5–8 March 2012.
47. Vaidis, M.; Otis, M.J.-D. Toward a robot swarm protecting a group of migrants. *Intel. Serv Robot.* **2020**, *13*, 299–314. [CrossRef]
48. Siddaiyan, S.; Arokiasamy, R.W. DVFH—VFH*: Reliable Obstacle Avoidance for Mobile Robot Navigation Coupled with A* Algorithm Through Fuzzy Logic and Knowledge Based Systems. In Proceedings of the International Conference on Computer Technology and Science (ICCTS), New Delhi, India, 18–19 August 2012.
49. Qin, Z.; Lin, Z.; Yang, D.; Li, P. A task-based hierarchical control strategy for autonomous motion of an unmanned surface vehicle swarm. *Appl. Ocean. Res.* **2017**, *65*, 251–261. [CrossRef]
50. Panagou, D. Motion planning and collision avoidance using navigation vector fields. In Proceedings of the IEEE International Conference on Robotics and Automation, Hong Kong, China, 31 May–7 June 2014; pp. 2513–2518.
51. Yu, J.; LaValle, S.M. Optimal Multirobot Path Planning on Graphs: Complete Algorithms and Effective Heuristics. *IEEE Trans. Robot.* **2016**, *32*, 1163–1177. [CrossRef]
52. Wei, H.-X.; Mao, Q.; Guan, Y.; Li, Y.-D. A centroidal Voronoi tessellation based intelligent control algorithm for the self-assembly path planning of swarm robots. *Expert Syst. Appl.* **2017**, *85*, 261–269. [CrossRef]
53. Du, Q.; Faber, V.; Gunzburger, M. Centroidal Voronoi tessellations: Applications and algorithms. *Siam Rev.* **1999**, *41*, 637–676. [CrossRef]
54. Vatamaniuk, I.; Panina, G.; Saveliev, A.; Ronzhin, A. Convex Shape Generation by Robotic Swarm. In Proceedings of the 2016 International Conference on Autonomous Robot Systems and Competitions, Bragança, Portugal, 4–6 May 2016; pp. 300–304.
55. Kharitonov, V. L.; Zhabko, A.P. Lyapunov-Krasovskii approach to the robust stability analysis of time-delay systems. *Automatica* **2003**, *39*, 15–20. [CrossRef]
56. Precup, R.-E.; Petriu, E.M.; Radac, M.-B.; Voisan, E.-I.; Dragan, F. Adaptive Charged System Search Approach to Path Planning for Multiple Mobile Robots. *IFAC* **2015**, *48*, 294–299. [CrossRef]
57. Nurmaini, S.; Tutuko, B. Motion coordination for swarm robots. In Proceedings of the 2014 International Conference on ICT for Smart Society: “Smart System Platform Development for City and Society, GoeSmart 2014”, Bandung, Indonesia, 24–25 September 2014; pp. 312–315.
58. Sharma, B.N.; Raj, J.; Vanualailai, J. Navigation of carlike robots in an extended dynamic environment with swarm avoidance. *Int. J. Robust Nonlinear Control.* **2018**, *28*, 678–698. [CrossRef]

59. Roy, D.; Maitra, M.; Bhattacharya, S. Study of formation control and obstacle avoidance of swarm robots using evolutionary algorithms. In Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, Hungary, 9–12 October 2017; pp. 3154–3159.
60. Jann, M.; Anavatti, S.; Biswas, S. Path planning for multi-vehicle autonomous swarms in dynamic environment. In Proceedings of the 9th International Conference on Advanced Computational Intelligence, Doha, Qatar, 4–6 February 2017; pp. 48–53.
61. Koenig, S.; Likhachev, M. *D* Lite*; American Association for Artificial Intelligence: Menlo Park, CA, USA, 2002.
62. Devi, R.V.; Sathya, S.S.; Kumar, N. Monkey algorithm for robot path planning and vehicle routing problems. In Proceedings of the 2017 International Conference on Information Communication and Embedded Systems, Chennai, India, 23–24 February 2017.
63. Caska, S.; Gayretli, A. An algorithm for collaborative patrolling systems with unmanned air vehicles and unmanned ground vehicles. In Proceedings of the 7th International Conference on Recent Advances in Space Technologies, Istanbul, Turkey, 16–19 June 2015; pp. 659–663.
64. Agrawal, A.; Sudheer, A.P.; Ashok, S. Ant colony based path planning for swarm robots. In Proceedings of the 2015 Conference on Advances In Robotics—AIR'15, Goa, India, 2 July–4 July 2015.
65. Vicmudo, M.P.; Dadios, E.P.; Vicerra, R.R.P. Path planning of underwater swarm robots using genetic algorithm. In Proceedings of the 2014 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), Palawan, Philippines, 12–16 November 2014.
66. Hedjar, R.; Bounkhel, M. Real-Time Obstacle Avoidance for a Swarm of Autonomous Mobile Robots. *Int. J. Adv. Robot. Syst.* **2014**, *11*, 67. [[CrossRef](#)]
67. Song, P.C.; Pan, J.-S.; Chu, S.-C. A parallel compact cuckoo search algorithm for three-dimensional path planning. *Appl. Soft Comput.* **2020**, *94*. [[CrossRef](#)]
68. Mendonça, M.; Chrun, I.R.; Neves, F.; Arruda, L.V. R. A cooperative architecture for swarm robotic based on dynamic fuzzy cognitive maps. *Eng. Appl. Artif. Intell.* **2017**, *59*, 122–132. [[CrossRef](#)]
69. Brooks, R.A. A robust layered control system for a mobile robot. *IEEE J. Robot. Autom.* **1986**, *2*, 14–23. [[CrossRef](#)]
70. Belkadi, H.A.A.; Ciarletta, L.; Castillo, P.; Theilliol, D. Distributed Path Planning for Controlling a Fleet of UAVs: Application to a Team of Quadrotors. *IFAC* **2017**, *50*, 15983–15989.
71. Kennedy, J.; Kennedy, J.F.; Eber-Hart, R.C. *Swarm Intelligence*; Morgan Kaufmann: Burlington, MA, USA, 2001.
72. Ayari, A.; Bouamama, S. Collision-free optimal paths for multiple robot systems using a new dynamic distributed particle swarm optimization algorithm. In Proceedings of the 2017 18th International Conference on Advanced Robotics, Hong Kong, China, 10–12 July 2017; pp. 493–497.
73. Alam, M.S.; Rafique, M.U.; Kausar, Z.; Saleem, M. Swarm intelligence based multi-objective path planning in environments cluttered with danger sources. In Proceedings of the 2016 17th International Conference on Mechatronics-Mechatronika (ME), Prague, Czech Republic, 7–9 December 2016.
74. Sharma, S.; Sur, C.; Shukla, A.; Tiwari, R. Multi-robot Area Exploration Using Particle Swarm Optimization with the Help of CBDF-based Robot Scattering. In *Computational Vision and Robotics*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 113–123.
75. Liang, H.; Gao, G. Navigating robot swarms using collective intelligence learned from golden shiner fish. In Proceedings of the Collective Intelligence Conference (CI-2014), Seoul, Korea, 24–26 September 2014.
76. Del Ser, J. Harmony Search Algorithm. In Proceedings of the 2nd International Conference on Harmony Search Algorithm, Seoul, Korea, 19–21 August 2015.
77. Karaboga, D.; Gorkemli, B.; Ozturk, C.; Karaboga, N. A comprehensive survey: Artificial bee colony (ABC) algorithm and applications. *Artif. Intell. Rev.* **2014**, *42*, 21–57. [[CrossRef](#)]
78. Katada, Y.; Nishiguchi, A.; Moriwaki, K.; Watakabe, R. Swarm robotic network using Lévy flight in target detection problem. *Artif. Life Robot.* **2016**, *21*, 295–301. [[CrossRef](#)]
79. Shi, Z.; Tu, J.; Zhang, Q.; Zhang, X.; Wei, J. The improved Q-Learning algorithm based on pheromone mechanism for swarm robot system. In Proceedings of the Chinese Control Conference, CCC, Xi'an, China, 26–28 July 2013; pp. 6033–6038.
80. Bellman, R.E. A Markov decision process. *J. Math. Mech.* **1957**, *6*, 679–684.
81. White, H.D.; Bailey, T. Simultaneous localization and mapping: Part I. *IEEE Robot Autom. Mag.* **2006**, *13*, 99–108.