Turk J Elec Engin, VOL.15, NO.2 2007, © TÜBİTAK

A Review of Studies in Swarm Robotics

Levent BAYINDIR and Erol ŞAHİN
Kovan Research Lab., Dept. of Computer Eng. Middle East Technical University, Ankara-TURKEY
e-mail: levent@ceng.metu.edu.tr • e-mail: erol@ceng.metu.edu.tr

Abstract

Swarm robotics is a new approach to the coordination of large numbers of relatively simple robots. The approach takes its inspiration from the system-level functioning of social insects which demonstrate three desired characteristics for multi-robot systems: robustness, flexibility and scalability.

In this paper we have presented a preliminary taxonomy for swarm robotics and classified existing studies into this taxonomy after investigating the existing surveys related to swarm robotics literature. Our parent taxonomic units are modeling, behavior design, communication, analytical studies and problems. We are classifying existing studies into these main axes. Since existing reviews do not have enough number of studies reviewed or do have less numbers of or less appropriate categories, we believe that this review will be helpful for swarm robotics researchers.

1. Introduction

Swarm robotics [15] is a new approach to the coordination of large numbers of relatively simple robots. The approach takes its inspiration from the system-level functioning of social insects which demonstrate three desired characteristics for multi-robot systems: robustness, flexibility and scalability.

Robustness can be defined as the degree to which a system can still function in the presence of partial failures or other abnormal conditions. Social insects are highly robust. Their self-organized systems can still work even after losing lots of system components or changing the environment parameters considerably.

Flexibility can be defined as the capability to adapt to new, different, or changing requirements of the environment. Flexibility and robustness have partly conflicting definitions. The difference between two occurs in problem level. When the problem changes, the system has to be flexible (not robust) enough to switch to a suitable behavior to solve the new problem. The biological systems have this level of flexibility and can easily switch their behaviors when problems change. For instance, ants are so flexible that they can solve foraging, prey retrieval and chain formation problems with the same base self-organized mechanism.

Scalability can be defined as the ability to expand a self-organized mechanism to support larger or smaller numbers of individuals without impacting performance considerably. Although there is a range in which the swarm performs in acceptable performance levels, this range is preffered to be as large as possible.

Our aim in this paper is to present a taxonomy for swarm robotics and classify existing studies into this taxonomy. In order to do this, we decided to split existing studies into different axes (parent taxonomic units) which represent the most important research directions in our view. Following section describes what these axes are and why we chose them. After discussing each axis in detail in separate sections, we will discuss about swarm robotics related fields and finish the paper with a conclusion section.

2. Research Axes

In order to decide on research axes, we investigated previous literature surveys related to swarm robotics.

Dudek et al., [16] classified the swarm robotics literature in terms of swarm size, communication range, communication topology, communication bandwidth, swarm reconfigurability and swarm unit processing ability. They prepared a taxonomy instead of a survey on swarm robotics and fit some limited number of sample publications inside this taxonomy.

We believe that swarm size criteria is not much applicable to characterization of swarm robot systems since scalability is one of the desired characteristics of swarm robotics and swarm systems should work with large numbers of system components. We also did not choose communication topology and communication bandwidth as subcategories since the communication should be kept limited as much as possible and preferably communication should be done using broadcasting instead of using robot names or addresses or complex hierarchies based on robot addresses. Although future studies will investigate the communication aspect of swarm systems more; having limited diversity in current studies, require us to have a communication axis which does not include bandwidth and topology of communication as a category in this survey.

Cao et al., [10] presented the survey of cooperative robotics in a hierarchical way. They split the publications into five main axes: group architecture, resource conflicts, origins of cooperation, learning and geometric problems. Group architecture is further divided into centralization/decentralization, differentiation (denotes the homogeneous or heterogeneous robot groups), communication structure and modeling of other agents dimensions. Modeling of other agents dimension contains studies which models the intentions, beliefs, actions, capabilities, and states of other agents to obtain more effective cooperation between robots.

Iocchi et al., [35] presented a taxonomy of multi-robot systems and address some multi-robot system studies in their taxonomy. They presented their taxonomy hierarchically using levels. First level is cooperation level which is divided into aware and unaware categories as the lower knowledge level. Aware category is divided into three more categories namely strongly-coordinated, weakly-coordinated and not-coordinated as the coordination level. Strongly-coordinated category is divided into strongly-centralized, weakly-centralized and distributed categories as the organization level. They also wrote a separate section for describing the application domains of multi-robot systems.

Gazi and Fidan [23] presented a review of multi-agent systems from the system dynamics and control perspective. The authors focused on agent dynamics models and described them in a relatively easy to follow way. Then they presented a section for some swarm coordination and control problems and a section for approaches to modeling, coordination and control of swarms. These two sections are similar to our behavior design and problems axes described below.

Our classification of swarm robotics literature has a hierarchical structure as in the work of Iocchi et al., [35]. Three main factors considered when designing this classification are: the current state of the literature (e.g. we cannot create a category which does not have any studies before), importance of the axis for swarm robotics (if the category still has remaining open problems or has important impacts on the field, it is preferred to define it) and pedagogical value.

Figure 1 shows our taxonomy of swarm robotics literature. In the main level; modeling, behavior design, communication, analytical studies and problems axes are defined.

Modeling dimension is divided into sensor-based, microscopic, macroscopic and cellular automata modeling subcategories.

Behavior design axis is divided into nonadaptive, learning and evolution axes. A reinforcement learning subsection, which is divided further into local and global reinforcement subsections, is added to learning axis.

While communication axis is divided into "Interaction via sensing", "Interaction via environment" and "Interaction via communication" categories, pattern formation, aggregation, chain formation, self-assembly, coordinated movement, hole avoidance, foraging and self-deployment problems are discussed in problems axis.

3. Modeling Axis

Modeling is a method used in many research fields to better understand the internals of the system that is investigated. But as we will discuss in the following paragraphs, modeling has some more advantages for swarm-robotics compared to other fields.

The existence of possible risks for the robots and the limited power of the robots require a human observer to follow the experiments and do some house keeping works periodically. The time spent on these experiments and possible risk of losing the robots even if a human observer exists become a bottleneck when several experiments are needed to validate the results of the studies. To eliminate these problems, it is safer and easier to model the experiments and simulate them on computers.

Another importance of modeling for swarm robotic studies appears when the scalability of the experiments are tried to be tested. Most of the time, scalability requires to test the control algorithms on more than hundreds of robots. But the cost of an individual robot prohibits testing of the experiments on more than a few tens of robots within the current state of the robot technology. Since scalability is an important aim of swarm-robot systems, it seems that the models will be needed until much more cheaper robots are manufactured.

Despite having such advantages of modeling, there is one more point need to be considered by swarm robotic researchers. Although models may be valuable for understanding the internals of the system being worked on, there will always be a difference between the simulation results and real world results. Although this difference is tried to be minimized by simulator developers, complex dynamics of interactions between the robots and unpredictable noise in the sensors and the actuators of the robots makes simulations impossible to be fully realistic.

We specified four types of modeling in this axis: sensor-based, microscopic, macroscopic and cellular automata modeling. Although adding cellular automata modeling as another type of modeling method is open to discussion and we might consider it as a special type of microscopic modeling method, we chose to add it as another type of modeling method because of the following reasons.

First, it is used as a modeling tool for several self-organized systems in biology [9] which shows that it is an established modeling method for biologists as well as computer scientists. Second, cellular automata is a simple and mature field which has lots of analytical tools [34] and is strongly connected to dynamical systems theory [34]. These properties of cellular automata make it a powerful modeling tool for swarm robotic studies.

3.1. Sensor-Based Modeling

Sensor-based modeling is a modeling method which uses the models of sensors and the actuators of the robots and objects in the environment as the main components of the modeled system. After modeling these main components, the interactions of the robots with the environment and the interactions between the robots are modeled. This modeling method is the mostly used and the oldest method for modeling robotic experiments.

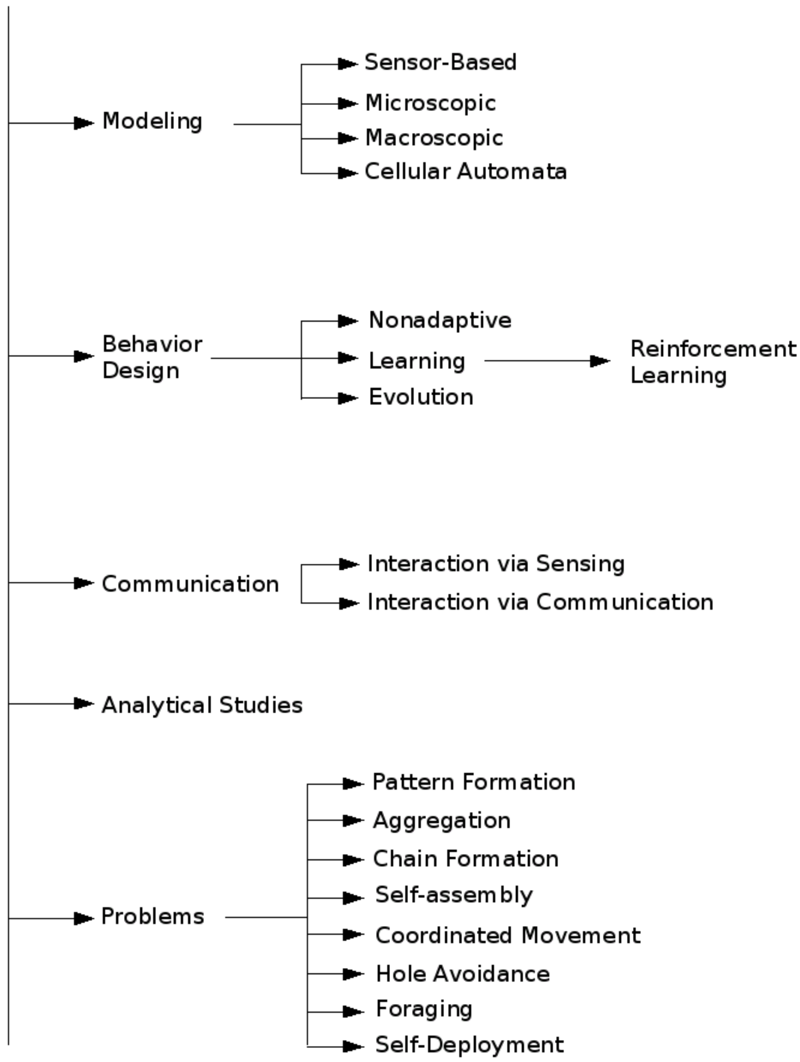

Figure 1. Taxonomy of Swarm Robotics Literature. The taxonomy is divided into five main axes namely modeling, behavior design, communication, analytical studies and problems.

The key in this modeling is to make interactions discussed above as realistic and simple as possible. These interactions should be modeled as simple as possible since the complexity of these interactions becomes very important when the scalability of the experiments are tried to be tested. They also need to be realistic to be useful for swarm robotic systems. These two aims are contradictory and presents a realism-simplicity dilemma in sensor-based modeling.

There are two main approaches for sensor-based modeling: non-physical simulations and physical simulations. In the former case, the dynamics of the robots and the objects in the environment are ignored and they are considered as objects without physical properties except adding some logic to eliminate collisions between them. Some of the studies using this approach are [27], [64], [30], [31], [6] and [63].

The latter approach models the interactions of the robots and the environment based on physical rules by assigning physical properties to the objects like the mass and the motor force to be able to move the robots. To obtain realistic results, off the shelf physics engines are integrated to simulation. This approach adds much more complexity to the model for the sake of obtaining more realistic results.

The examples of this approach can be found in [4] and [58]. The authors physically modeled the environment using an open-source physics engine and run the experiments in parallel over multiple computers connected via a network to overcome the increased complexity of the simulations. Some other examples using this approach are [66] and [65].

3.2. Microscopic Modeling

Microscopic models models robotic experiments by modeling each robot and their interactions mathematically. In this method, behaviors of robots are defined as states and the transition between these states are bound to internal events inside robot and external events in the environment.

The main difference between microscopic models in this section and macroscopic models in the following section is the granularity of the models developed. While microscopic approach models the experiments by modeling each robot, the macroscopic approach models the whole behavior of the system directly.

As a special case of microscopic and macroscopic modeling, probabilistic microscopic and probabilistic macroscopic models are used in swarm robotics. By assigning probabilities to transitions between robot actions (for microscopic models) or transitions between system states (for macroscopic models), the system behavior and the noise in the environment are easily integrated into these probabilistic models.

In probabilistic microscopic models [46], [45], [36], a time unit is defined based on a primitive event [1] to be able to advance the model at each model step. After specifying this time unit, the probability of each state transition is computed with systematic experiments performed with real robots. In other words, the probabilities of all events are computed per time unit of the model. After finding these state transition probabilities, the mathematical model is run for each robot by generating random numbers between 0 and 1 for each possible event transition of the selected robot and comparing these numbers with state transition probabilities. If some of these numbers are lower than the predefined transition probabilities of the associated events, those events are assumed to be occurred and the state of that robot is changed.

Jeanson et al., [36] studied aggregation strategies in cockroaches. They tried to prove that cockroaches perform the global aggregation from local interactions. To do this they measured the important system parameters from the experiments with cockroach larvae like probability of stopping in an aggregate or

[1] Ijspeert et al., [32] defined this primitive event as the average detection time of the smallest object in the environment when the robot is moving with its average speed.

probability of starting to move. A numerical model of behaviors of cockroaches is created from these measurements and tried to be validated by numerical simulations. Although their numerical model reveals a quantitative disagreement with real experiments, they claimed that it also offers strong evidence that aggregation can be explained in terms of local interactions between individuals.

Ijspeert et al., [32] applied microscopic modeling to stick pulling problem. They developed the microscopic model from the finite state automata (FSA) of the robot controller. The transitions between states of the FSA and variables of the simulation (e.g. robot speed or stick detection range) are approximated with systematic experiments. The results of microscopic model are compared to the implementation of the controller in a sensor-based simulation and to the implementation on real robots. It is shown that the probabilistic model predicts the collaboration dynamics successfully.

3.3. Macroscopic Modeling

Another kind of mathematical modeling method of robotic experiments is macroscopic modeling. In macroscopic modeling, the system behavior is defined with difference equations and each of the system states (variables of difference equations) represents the average number of robots in a particular state at a certain time step.

While the system need to be iterated for each robot in microscopic models [2], macroscopic models are solved only once to obtain the steady state of the model. Although this feature allows great speedups for macroscopic models when compared to microscopic models, microscopic models allow to catch the fluctuations in the experiments. In other words, while macroscopic models allows to obtain a rough global behavior of the robotic system quickly, microscopic models allow to obtain a more realistic global behavior slowly.

Similar to microscopic models, probabilistic version of macroscopic models [46], [42] are used in swarm robotic studies to handle noise in a simple way. Martinoli et al., [46] applied macroscopic modeling to stick pulling problem. The authors presented the model incrementally starting from a basic model which only contains Search and Obstacle-Avoidance states up to the most complex model which contains all states in the robot controller. For each stage, a difference equation (DE) is developed and the steady state of the DE system is analyzed to obtain average number of robots in each state at the end of the experiments. Comparisons of microscopic, macroscopic and sensor-based models are also presented and the limitations of macroscopic modeling for stick pulling problem are described.

De Wolf et al., [13] used a different way of macroscopic modeling in their experiments. Their method is based on the "equation-free" macroscopic analysis [37] which aims to use the algorithms that are designed for equation-based models even if the only available model is individual-based. The method does this by replacing evaluations with the results of individual-based simulation whenever the numerical algorithms need to evaluate the equation. Also an estimation method (e.g. Newton's method) is used to accelerate the simulation. The aim of using an estimation method was to wipe out the need to request all evaluations from the individual-based simulation. Instead, they only requested some initial evaluations and extrapolated the remaining values with the help of estimation method.

Another distinguished feature of this study is the definition and tracking of system-wide guarantees for self-organizing emergent systems. The authors developed an equation-free macroscopic model and system-wide guarantees for an automated guided vehicle warehouse transportation system. They validated the

[2] The experiments with microscopic models also need to be performed several times to obtain the average behavior of the robotic system.

results of the model by comparing the results of the accelerated equation-free macroscopic model with the non-accelerated one[3]. Although they found that some accuracy are lost which is normal for all macroscopic models, the model managed to find the steady state successfully.

Trianni et al., [63] tried to find macroscopic models of aggregation and chain formation problems. But the results of macroscopic model did not fit to the results obtained from sensor-based simulations. They thought that the possible problems are the lack of spatial information in the mathematical model, carrying out the simulation in discrete time and the lack of interaction dynamics in the model. At the end of their experiments, they decided to make their sensor-based simulations more realistic using physical sensor based modeling instead of improving their macroscopic model in their future studies.

3.4. Cellular Automata Modeling

Cellular automata (CA) is among the simplest mathematical models of complex systems [34]. The CA models contain discrete lattice of cells in one or more dimensions where each cell in the lattice has finite number of possible states. Each cell interacts only with the cells that are in its local neighborhood and the system dynamics are characterized by the local rules executed locally on the cells in discrete time steps.

Several CA models are developed for the natural phenomenas [17], [12] around us. In addition to using these models as inspiration sources of swarm robotic studies, CA can be used as a modeling tool for CA based experiments. The studies of Shen et al., [56], [57] is an example of this type of studies. The details of these studies are summarized in section 8.3.

4. Behavior Design Axis

Adaptation is any change in the structure or the function of an entity (e.g. a component of a complex system) that allows it to survive more effectively in its environment.

Adaptation in biological systems can be classified as structural, behavioral and physiological adaptation. Structural adaptations are special body parts of an organism that help it to survive in its natural habitat, like its skin color, shape, body covering and teeth. Behavioral adaptations are special ways a particular organism behaves to survive in its natural habitat. Physiological adaptation are subsystems present in an organism that allow it to perform certain biochemistry reactions like secreting slime, being able to keep a constant body temperature or producing pheromones.

An important property of adaptation is its time scale. There are two types of adaptation based on time scale: evolution and learning. Especially structural and physiological adaptations do not develop during an individual's life but over many generations with evolution. In addition to evolution, the individuals may fine-tune their behaviors in their lifetime. This kind of adaptation is performed in a relatively shorter time scale and called learning.

In swarm robotics literature, researchers mostly tried to utilize the behavioral adaptation to control large number of robots to accomplish a task collectively. Because of this and importance of adaptation, we decided to categorize existing behavior design approaches into three sections based on the behavioral adaptation capability of the robot controllers: manual, learning and evolution.

While we describe the works which uses nonadaptive robot controllers in nonadaptive section, the works which show learning capabilities are described in learning section and the ones which try to mimic natural selection for adapting the robot controllers are described in evolution section.

[3] Non-accelarated equation-free model can be considered as a kind of microscopic model.

4.1. Nonadaptive

Most of the studies utilizing nonadaptive behavior design are categorized into four subcategories: subsumption, probabilistic finite state automata, distributed potential field methods and neural networks. While these categorized studies are described in the following subsections respectively, the nonadaptive studies which do not belong to these categories are described below.

Brooks et al., [8] presented their initial studies about developing small bulldozer robots and developing coordination strategies for these robots for achieving tasks that will be useful in building a manned lunar base. After describing the benefits of using collective robotic systems, they described the robots and the initial behaviors they developed. The behaviors are described in an abstract way and the proof of their success is not presented even if in simulation.

Payton et al., [54], [55] described a new approach in swarm robotics called pheromone robotics based on the biologically inspired concept of 'virtual pheromone'. They developed robots with personal digital assistant (PDA) attached at the top which allows to do computationally expensive operations. The virtual pheromones are signaled between robots with a mechanism attached at the top of the robots which contains eight radially-oriented, directional infrared receivers and transmitters. The information is transferred between the robots as 10-bit messages which have message type, hop-count and data fields. The intensity and orientation values obtained from received messages are also used in obstacle detection and in determining distance and direction of neighboring robots.

They defined three main concepts in their studies: virtual pheromone, world embedded computation and world embedded display. Virtual pheromones are working with the help of infrared mechanism described above. With the help of virtual pheromones, the robots may solve problems like generating the map of a field or solving the shortest path problem for a field. This feature is called as world embedded computation. An external observer can also be informed about the results obtained in world embedded computation with the help of a video camera mounted on the observer's head which receives and displays coded infrared signals from each robot. This feature is called as world embedded display.

4.1.1. Subsumption

Subsumption architecture [7] is one of the distinguished and classical architectures in behavior-based robotics [2]. The architecture allows efficient coordination of behaviors by using a simple inhibition mechanism between the behaviors and incremental building of robot controllers by considering each behavior as a separate module which can inhibit other behaviors.

Mataric [47] presented design of some behaviors from simple to complex using subsumption architecture in a clear way. The behaviors designed are collision avoidance, following (inverse of collision avoidance), dispersion (used in order to balance goal-directed behavior against interference), aggregation, homing and flocking. Although the author showed some simulation screenshots as examples of the success of the behaviors, she did not show any evidence or analysis results about the implementation of the behaviors on real robots even if she claimed that the developed behaviors are tested on a herd of physical mobile robots in the abstract of the publication.

Nouyan and Dorigo [52] implemented a chain formation behavior in a sensor-based simulation. The robots had two phases: explorer and chain member. In explorer state, the robots search for other chain members or the nest. Whenever a robot finds the nest or a chain member, the robot tried to keep permanent visual contact with it using an omni directional camera. The aim of the robots was to find the end of the chain and stay there after the explorer timeout is reached. The robots can distinguish chain members and the nest based on the color of the LED ring around their body. The authors made systematic experiments by modifying the number of robots and the explorer timeout to see the changes in the speed of the chain formation process and the shape of the formed chains. It is observed that while short explorer timeout leads to the fast formation of many chains, a long explorer timeout results in the slow formation of fewer chains.

Nouyan [53] also extended this work with more detailed configurations in his thesis. The author also used same behaviors for the problem of establishing a path towards a goal location from the nest.

4.1.2. Probabilistic Finite State Automata

Probabilistic finite state automata (PFSA) is a way to represent dynamical systems with finite state spaces. In a probabilistic automata, the transitions between the states of the system are triggered with certain probabilities. The general approach is to model the robot behaviors as states and defining the state transitions with some external input and probabilities. This section will summarize the swarm-robotics studies using this approach.

Soysal and Şahin [58] performed systematic experiments using a probabilistic finite state machine based controller for performing aggregation task. There are four behaviors in the controller which are connected with subsumption architecture: obstacle avoidance, approach, repel and wait. Normally robots start in approach state and switches to the wait state when they sense another robot. The switches between repel and approach states, and wait and repel states are determined by P_{return} and P_{leave} probabilities respectively. The authors changed the size of the arena to compare different strategies obtained by modifying the P_{return} and P_{leave} parameters. They showed that the best performance is obtained when both of the parameters equal to 1. They also stated that this strategy may not be very feasible on all robotics systems since there is a risk of having large number of robots moving in close proximity and the large power consumption due to continuous movement.

Labella et al., applied a PFSA based adaptation algorithm to prey retrieval task [41], [39], [40]. The PFSA based controller of the robots has the Search, Retrieve, Deposit, Rest and Give Up states which are in fact the robot behaviors. All transitions between states are triggered by external events except the transition between Rest and Search state which is triggered probabilistically. The probability of triggering Rest-Search transition is updated depending on the number of consecutive successes or failures. They tested the algorithm on Lego Mindstorm robots and showed that task allocation occurred between the robots because of the minor mechanical differences of the robots. At the end of experiments, some of the robots become foragers and the others become loafers.

A self-organized model of the aggregation behavior of cockroaches in a bounded circular arena is developed by Jeanson et al., [36] and Garnier et al., [22]. The authors used an approach which is similar to microscopic modeling developed by Martinoli et.al [46], [45] and Jeanson et al., [36]. They first define a self-organized model for the behaviors of the cockroaches and measured the important transition probabilities between behaviors along with the average time spent on each behavior by real cockroaches. They compared the results obtained from the developed numerical model with the real experiments' results. They claimed that their model better approximates real data than most of the previous global level models which shows that the cockroaches may behave based on local interaction rules.

4.1.3. Distributed Potential Field Methods

The method of behavior design used in the studies of this category is very similar to the one (called potential field method) used by Khatip [38] and Arkin [1] for single-robot case. The method mainly represents all interactions of the robot with other objects in the environment as vectors. These vectors can be attractive (e.g. moving towards a goal) or repulsive (e.g. moving away from obstacles). The vectorial summation of these forces are computed and used as the action of the robot. The studies in this section all computes these vectors locally from the viewpoint of the robot which makes these studies fully distributed.

Spears et al., [59] defined a new distributed framework (called artificial physics) for the control of large number of robots using artificial force concept. Their work is very similar to potential field method used in single robot systems but this method does all computations at runtime different from the potential field method. The computations are also done on each robot locally. No global map is generated. Other objects in the environment are assumed to be applying virtual forces to the robot selected. The robot computes the average force computed by its observations and moves towards the direction of average force. The force exerting on a robot from an external object depends on two things: the bearing and the distance with the external object. Since both of these parameters can be computed from local observations, the framework is suitable to swarm robotics studies.

Spears et al., first used artificial physics methodology for forming hexagonal lattices both in 2D and 3D. Then they tested the methodology on obstacle avoidance, surveillance and perimeter defense tasks with real robots. They also tested the robustness of the system and applied some theoretical analysis on the parameters used in the framework.

Balch and Hybinette [6] presented a distributed algorithm which is based on potential fields method [38] to achieve a formation while navigating to a goal location. The algorithm has several behaviors represented as motor schemas [1]. The overall behavior of the robots are the summation of the vectors returned from the motor schemas. Other than avoid-static-obstacles and avoid-robots motor schemas; noise, move-to-unit-center and maintain-formation motor schemas are also used. While noise motor schema adds a random noise vector to escape from local minimas in the system, move-to-unit-center motor schema is used as an attractive force to draw all of the robots together. The result vector is pointing to the approximate center of the robots which is computed based on local information available to the robot.

Maintain-formation motor schema is executed based on the "attachment site" concept. Depending on the formation aimed to form, the number of attachment sites around a robot and the angle need to be in contact with these attachment sites are changed. It is assumed that the attachment sites are positioned around the robots uniformly. Maintain-formation motor schema generates an attractive vector towards the closest site. The algorithm is tested on simulation and shown that it is scalable.

4.1.4. Neural Networks

Neural networks [29], [28] are powerful learning mechanisms inspired from nervous system of humans. There are two general types of swarm robotics studies performed using neural networks. The first type uses genetic algorithms to evolve the weights of a neural network to obtain a desired behavior with a fitness function appropriate to the problem. This type of studies [4], [64], [66], [65] are discussed under section 4.3.

The second type of studies with neural networks considers the neural networks as a generalization mechanism and do not use its learning capabilities. The remaining part of this section summarizes this type of studies.

Grob et al., [25] investigated self-assembly problem with a group of robots. They defined the problem as controlling the robots in fully autonomous manner in such a way that they locate, approach and connect with an object that acts as a seed or connect to other robots already connected to the seed. The seed and the robots connected to the seed are discriminated based on the color of the ring around them.

The controller of the robots was a simple perceptron which connects sensory inputs to motor outputs of the robots. The controller was preprogrammed with the controller obtained from another study. The experiments are done on flat and rough terrains with real robots. The results show that robots achieves self-assembly in a scalable way.

Martinoli and Mondada [44] implemented object clustering and stick pulling experiments with two simple behaviors (handle-object and avoid-obstacle) coded as a neural network. Depending on the hard coded weights of the neural network, one of the behaviors is activated at each time step. The output of the neural network is directly connected to motor outputs. For object clustering experiments, they reported that increasing the number of robots in the experiment resulted with a decreased performance because of the interferences of the actions of the robots. Although they reported that the stick pulling experiments are successful, they did not show any quantitative results in this publication.

4.2. Learning

Montemanni and Gambardella [50] presented a distributed protocol for minimum power topology (MPT) problem in wireless networks. The aim in MPT problem is to assign transmission powers to the nodes of a mobile network in such a way that all the nodes are connected by bidirectional links and the total power consumption is minimized.

The authors used one of the previous protocols called MLD (Minimum Link Degree) and made it more distributed. The name of the new protocol is LMPT (Local Minimum Power Protocol) which uses some local information about neighbors to obtain better results.

MLD protocol works as follows: There is an ngb (link degree) parameter which is used as a minimum number of links any node should have to obtain full connectivity on the network. The nodes increase their transmission power in small amounts until they reach to ngb number of neighbors. Whenever a node hears another node in this increasing transmission power phase, it realizes that its neighbor has less than ngb neighbors and sets its transmission power as the power of its neighbors‘ transmission power if it is greater than its current transmission power. If it is lower than its current transmission power, then current transmission power is not changed. These phase goes on until each node has at least ngb neighbors. They all stop increasing their transmission powers at this point.The ngb parameter is an heuristic obtained from the global information known about the network. It does not need to be perfect information but the more it is approximated better, the lower the total transmission power at the end.

Montemanni and Gambardella's LMPT protocol uses the same logic for ngb. In the first phase, all nodes reaches to ngb neighbors using MLD protocol. While reaching this information, the nodes also gets an extra information from their neighbors: the power for the neighbor required to reach to its neighbors. After getting all these information, each node runs a local optimization over these local information for deciding the minimum power requirement of all the nodes in the neighborhood. After obtaining this information, the neighbors are informed for their new transmission powers which allows full connectivity with lowest possible power consumption on that locality.

The local optimization procedure was an instance of integer programming method run on these head nodes. The performance of MLD and LMPT are compared based on three criteria: total power, average number of neighbors and maximum number of neighbors. The results showed that LMPT is much better than LMD in terms of both total transmission power and the number of neighbors for all of the experiments. It is also observed that LMPT is much less sensitive to the ngb value because LMPT works much better than LMD when ngb is overestimated. Interestingly, it is observed that LMPT is better than LMD even when the first one is run with an overestimated value of ngb, and the latter uses the smallest possible value of ngb.

4.2.1. Reinforcement Learning

Reinforcement learning (RL) [61] systems consist of a discrete set of environment states, a discrete set of agent actions and a set of scalar reinforcement signals. In robotic studies, environment states are higher level representations of sensor readings (e.g. existence of an object in front of the robot based on the thresholded values of front sensor readings). Similarly agent actions are higher level representations of actuator commands. Generally behaviors [2] are used as the actions of the robots.

The reinforcement value is the core concept in RL which differentiates it from other types of learning methods [49] (e.g. supervised or unsupervised learning) The reinforcement value gives a numerical hint to the agent for the relative success of the executed action in achieving the goal of the agent. The aim of the agent in this setting is to learn a policy (which maps states to actions) that maximizes the cumulative reinforcement values obtained in the long-term.

One of the important properties of RL is that the RL algorithms have clean theoretical convergence properties because of their dynamic programming roots [61]. Despite advantages of RL, there are serious problems in applying RL to multi-robot studies. First, theoretical convergence properties of RL require large numbers of learning trials that are difficult to perform with physical robots.

Another problem is the size of the search space. The RL algorithms are proved to converge on toy problems which has limited search space compared to the robotic problems. Large search space (both state and action spaces) of robotic problems requires lots more epochs to be able converge to acceptable results.

Noise is another serious problem while applying RL to multi-robot studies. Besides having lots of noise in sensor readings and actuator actions, interaction between the robots make the environment noisier and more unpredictable. Having multiple robots in the environment also breaks the convergence assumption of some of the well known popular reinforcement learning algorithms (e.g. Q-learning [67]) since noise converts the environment to a dynamic one from a stationary one.

The last problem is probably the most difficult and classical problem in machine learning: the credit assignment problem. Both temporal and spatial credit assignment problems exist in multi-robot problems since the actions of the robots can be rewarded with a delay and the result may depend on the actions of multiple robots.

We divided reinforcement learning studies into two categories: the studies which use local reinforcement and the ones which use global reinforcement. In the former one, the reinforcement is only given to the robots which are close to the location where the reinforcement is generated. In the latter one, all robots are rewarded as if the last action is a result of the collective actions of all robots. In other words, even if some robots do not contribute to the goal, all of the robots are rewarded in global reinforcement scheme.

As we discussed in section 2, the communication should be kept limited as much as possible in swarm robotic systems. Because of this preference, local reinforcement scheme is more realistic for swarm robotics. But investigating the global reinforcement and comparing its results with local reinforcement may offer new insights in swarm robotics.

Yang and Gu presented a survey about multi-robot reinforcement learning studies in [68]. They first discussed preliminaries of the subject starting from Markov decision processes up to relation of multi-agent reinforcement learning with the game theory. Later, they summarized theoretic frameworks for multi-agent reinforcement learning, algorithms utilizing these frameworks and the studies performed with these algorithms. After discussing these, they summarized the works done up to that time for scaling reinforcement learning to multi-robot systems. Finally they described main challenges of multi-robot systems and future research directions in the field which are mainly obtaining team cooperation, abstracting state and action spaces, generalization and approximation of look-up tables used in reinforcement learning algorithms and extending the reinforcement learning into continuous state and action spaces.

Local Reinforcement

In local reinforcement scheme, the reinforcement value generated after achieving a subgoal is only shared by the robots which contributed achieving that subgoal. One of the studies using local reinforcement was the study of Li et al., [43]. The authors used Balch‘s social entropy metric [5] to analyze the effect of diversity and specialization on a stick-pulling experiment. Since Balch‘s social entropy metric can only be used to measure the diversity of the robot groups, Li et al., defined specialization as a new metric of the correlation between the diversity and the performance.

Authors‘ previous stick pulling experiment described in [32] had robots equipped with gripper turrets and proximity sensors. The robots were searching for sticks in a circular arena and pulling them out of the ground. In this study, Li et al., added two more experiments to their previous stick pulling experiment by including two additional types of sticks: longer and heavier sticks. Both of the sticks were requiring collaboration of robots to pull them out. While robots were needing sequential collaboration in former one, parallel collaboration was needed in the latter one.

Li et al., used the adaptive line-search algorithm used in their previous study [32]. The algorithm was based on a parameter called gripping time parameter (GTP). GTP was the maximal length of time a robot waits for the help of another robot while holding the stick. The robots‘ behavior was basically consisting of searching for sticks, gripping the stick when any of them found and waiting GTP seconds while holding the stick. If the minimum number of robots required to pull a stick reached to that stick in GTP seconds then the stick was considered to be pulled out and the robots continue to search for new sticks. If the minimum number of robots required to pull a stick could not be reached then the waiting robot fails to pull the stick and switches to searching sticks behavior.

The GTP was being updated with a kind of reinforcement learning algorithm. In this learning algorithm, both local and global reinforcement signals were tested. The local reinforcement signal rewarded a robot when it completely pulled out a stick or passes the stick to another agent. The global reinforcement signal was defined to be the general swarm performance, the number of sticks pulled out, in a predefined time period.

The learning algorithm basically starts with a random direction and GTP. When a predefined amount of time passes for a robot, an average reinforcement is computed for that time period. Then the GTP value is updated for that robot depending on both the current and the previous average reinforcements. If the current average reinforcement value is greater than the previous one then the GTP is modified in the same direction selected in the previous step. If the current average reinforcement value is lower than the previous one (It means the performance becomes worse.) then GTP is modified in the opposite direction of the previous modification of GTP.

Li et al., performed systematic experiments using local and global reinforcement signals with different group characteristics (homogeneous, heterogeneous and caste-based robot groups). Although the performance of the learning swarms achieved the same level of performance independent of the initial GTP, the performance of homogeneous swarms with a fixed GTP is decreased when the initial gripping parameters are increased. This shows that a higher level of robustness is achieved with this learning algorithm.

Tangamchit et al., [62] used Monte Carlo learning method to solve foraging problem. They modified the foraging problem definition a little bit to be able to allow cooperation between the robots. The modified problem is discussed in section 7.7.

The authors showed the validity of their beliefs about using cumulative discounted reward based learning methods unable to induce cooperation and therefore give suboptimal results on cooperative problems. They compared the result of an average reward learning method (specifically Monte Carlo algorithm) to a cumulative discounted reward based learning method (specifically Q Learning). The authors used both local and global rewards in their comparisons. It is shown that only Monte Carlo learning with a global reward scheme can achieve cooperation. They claimed that local reward scheme does not produce cooperative behavior since the robots do not want to help other robots if they cannot get any reward.

Global Reinforcement

In global reinforcement scheme, the reinforcements obtained by robots in a specified period of time are shared between robots. Mataric [48] solved foraging problem using reinforcement learning in multi-robot domain. The author defined two challenges for applying reinforcement learning to multi-robot domain. The first one is that even if for single robot experiments the domain has very complex state space; when more than one robot is used, the problem becomes more complex because of the inferences between the robots. The second challenge is the structuring and assigning reinforcement learning. The first problem is handled with the help of behaviors and conditions. The complexity if state and actions spaces are reduced considerably with the help of them. The second problem is handled with the help of shaped reinforcement which consists of heterogeneous reward functions and progress estimators.

Mataric developed a simple reinforcement algorithm called reinforcement summation algorithm which adds and normalizes the reinforcement values obtained for state action pairs over time. The author compared the results of two different variations of this algorithm with a hand coded optimal solution and pure q-learning algorithm without shaped reinforcement. This first variation of her algorithm was the reinforcement summation algorithm with only heterogeneous reward functions and the second variation was the reinforcement summation algorithm with both heterogeneous reward functions and progress estimators. The results showed that the first variation is the best when compared to others and q-learning lagorithm is better than the hand coded optimal solution.

4.3. Evolution

The swarm robotics studies mimicing evolution use genetic algortihms as the implementation method. Genetic algorithms [24] are one of the mostly used offline optimization algorithms in robotics because of their ability to escape from local optimum and previous successes when applied to similar problems.

The genetic algorithms are generally used to evolve weights of the neural networks to obtain the desired behaviors. This approach is very powerful since it combines the generalization ability of the neural networks with the ability to escape from local optimums of genetic algorithms. The remaining part of this sections is used to summarize the studies performed in swarm robotics using this approach.

Bahçeci and Şahin [4] systematically studied the performance and the scalability of evolved aggregation behaviors. They used a neural network as the controller of the robots which has 12 inputs and 3 outputs. While the first four of the input neurons encodes sound value obtained from the speaker, the remaining input neurons encodes the infrared sensors of the robot. Similarly, the first output neuron is used to control the omni-directional speaker and the remaining two are used to control the wheels.

The authors used a genetic algorithm to evolve the weights of the neural network. The fitness function used in this study was based on the neighbor and connected predicates. The robots i and j are assumed to be connected if there is a path from robot i to robot j over the predicate neighbor and two robots are neighbors if the distance between them is below 10 units. They defined the fitness of a single evaluation of a controller (chromosome) as the ratio of the number of robots forming the largest cluster to the total number of robots. The fitness of a chromosome over the whole experiment is computed by averaging all fitness values of that chromosome. Alternatively, median, minimum and maximum operators are tested instead of mean operator.

Trianni et al., [64] used genetic algorithms to evolve aggregation behavior for a group of simulated robots. The controller of the robots was a simple perceptron which connects sensory inputs to motor outputs of the robots. The weights of the perceptron were evolved using a fitness function which computes average distance of the robots from their group's center of mass.

A static and a dynamic clustering behaviors were evolved in this study. Although the static one created very compact behaviors, the clusters did not move as in the case of dynamic clustering behavior. The behavior of moving cluster allowed to join smaller cluster into bigger ones and resulted in much more scalable behavior.

Trianni et al., [66], [65] tried to achieve hole-avoidance behavior with a swarm of robots. The robots have to perform coordinated motion in an environment which has holes too large to be traversed. The robots had to be connected with their turrets while sensing the environment. When a robot detects a hole using its ground sensors, it should move in the opposite direction of the hole. The force generated with this movement is detected by the traction sensors of other robots. The challenge was to learn reacting to these forces in an appropriate way so that hole avoidance can be achieved independent from the robots' relative positions and position/size of the holes.

Trianni et.al used a genetic algorithm to obtain the controller of the robots. The controller was a simple perceptron which connects sensory inputs to the motor outputs of the robot. The weights of the perceptron were evolved using a fitness function which has three components. While the first component was measuring the performance in terms of coordinated movement, second one was measuring exploration of the arena and the third one was using fast reaction to the detection of a hole. The evolved strategies were also tested on more difficult environments by varying the size and the shape of the robot swarms. It is observed that the strategies are successful and scalable.

5. Communication Axis

We have used the same classification categories Cao et al., [10] used in their survey of cooperative robotics for classifying the swarm robotics studies based on the communication mechanisms used by the swarms.

The first category (interaction via sensing) is the simplest and the most limited type of communication between the robots. This type of communication requires the robots to distinguish between other robots and the environment objects. The details are discussed in the corresponding section below.

In the second category (interaction via environment), the robots used the environment as a communication medium. There are well known examples of this communication type in biology like communication via pheromones in ants [9].

The ants communicates with each other through chemicals called pheromones. For example, when an ant finds food, it will leave a trail along the ground on its way back to home, which in a short time other ants will follow. When they return home they will reinforce the trail, bringing other ants, until the food is exhausted. The slow dissipation property of the pheromone trials will allow the ants to find new food sources when the older ones are exhausted.

Although the communication scheme is simple in this approach, the physical implementation of it is not so easy because of the difficulty of creating special environments allowing communication between agents.

Most of the studies using this approach uses only simulation of this communication scheme with the help of a short range wireless communication mechanism (e.g. RF or Infrared) [54], [55], [56], [57]. Because of this, we decided not to create a separate section for "interaction via environment" method and described the simulation attempts of this communication method in "interaction via communication" section.

The third category (interaction via communication) involves explicit communication with other robots by broadcast messages. Although Cao et al., [10] included the communication via directed messages (using robot identification numbers) in this category, we did not prefer this since swarm robotics prefers to use the communication in a limited way.

Following two sections describe the studies using "interaction via sensing" and "interaction via communication" methods subsequently.

5.1. Interaction via sensing

The discrimination of "interaction via sensing" from "interaction via communication" can be difficult time to time. Our guideline to do this discrimination is to look at the aim of the information sender side. If the sender in the interaction aims to give information to other robots intentionally then that study is categorized as "interaction via communication" instead of "interaction via sensing". So if two robots interacting to pull a stick and sensing each other's action in a limited way, this work is considered as "interaction via sensing". And if robots broadcasts information packages or switches on/off a light around them to show their state, these studies are considered to be the type of "interaction via communication".

Interaction via sensing requires the discrimination of other robots from the environment objects, also called as the kin recognition. Kin recognition is an important feature of animals in nature. With the help of kin recognition, animals can behave different to their kins, work together to accomplish some tasks, and protect themselves from their enemies better.

We considered kin recognition as a kind of minimalist communication mechanism since just by discriminating the kin and observing their behaviors (without explicit communication), the robots can manage to solve several problems (e.g. flocking, chain formation and cooperative stick pulling) in swarm-robotics. It is also required to solve many problems (e.g. aggregation and flocking) efficiently.

Most of the swarm robotics studies (e.g. [59], [55], [54], [58], [43], [27], [64]) uses kin recognition as a communication medium since most of the problems requires (e.g. flocking, chain formation and cooperative stick pulling) discrimination of robots in the environment to obtain acceptable performance. As an example, Soysal and Şahin [58] need the robots to discriminate other robots from obstacles since it is possible for the robots to aggregate near the walls instead of each other in a rectangular arena.

Ijspeert et al., [32] studied collaborative stick pulling problem. In order to pull a stick from its place, collaboration of two robots are required in these experiments. Since the first robot can only pull the stick up to a point (because it is too long to be pulled by one robot), it should wait until a second robot helps to pull it.

Trianni et al., [66], [65] tried to solve hole-avoidance problem using genetic algorithms to evolve the weights of a simple perceptron based controller. The robots are connected to each other with joints and they have to perform coordinated motion in an environment which has holes too large to be traversed. The aim of the study is to learn the correct dynamic to move away from the holes as a group when the robot(s) on an edge of the formation sense the hole with its (their) ground sensor(s). The robots can sense their neighbors' relative movements with the help of their traction sensors. The communication with the help of traction sensors can be considered as an example of interaction via sensing since there is no intention to send information to other robots in this communication scheme.

Some of the experiments which do not use kin recognition are [39], [40], [48], [4].

5.2. Interaction via communication

A more advanced version of communication requires direct communication of robots by broadcasting or one-to-one communication. As mentioned before, one-to-one communication using identification of robots is not preferred in swarm robotics studies since this may reduce the scalability and flexibility of the system.

Nouyan and Dorigo [52] implemented a chain formation behavior. Initially the robots search for other chain members or the nest. Once a robot finds a chain member or the nest, it becomes a chain member depending on two predefined timeouts. The robots distinguish chain members and the nest based on the color of the LED ring around their body. A chain member can have three different colors: blue, green and red. It activates the color blue, if it connects to the nest or to a red chain member. It activates the color green, if it connects to a blue chain member and color red otherwise. This coloring mechanism allows robots to find the direction of the chain. Since having a long chain instead of a chain with several branches is preferred, the robot follows the color to reach to the end of the chain to connect. Nouyan [53] also extended this work with more detailed configurations in his thesis.

Grob et al., [25] studied the self-assembly problem. The aim of the work is to locate, approach and connect with an object that acts as a seed or connect to other robots already connected to the seed. Similar to the Nouyan and Dorigo's previous work described above, a robot discriminates the robots connected to the seed with the help of the LED ring around robot's body. The initial color of the robots are set to blue. Once a robot connects to the seed or to a robot already connected to the seed, it activates the color red permanently.

It is also worth to mention the studies performed by Payton et al., [54], [55] Shen et al., [56], [57] in this section since they used broadcasting to simulate "the interaction via environment" type of communication. The details of their works related to communication are already described in section 5.1.

Although the works of Payton et al., [54], [55] and Shen et al., [56], [57] can be seen as simulation attempts of "interaction via environment" method, we decided to describe these studies in this section.

Payton et al., [54], [55] simulated the communication mechanism used by ants. The ants communicates with each other through chemicals called pheromones. When an ant finds food, it will leave a trail along the ground on its way back to home, which in a short time other ants will follow. When they return home they will reinforce the trail, bringing other ants, until the food is exhausted. The slow dissipation property of the pheromone trials will allow the ants to find new food sources when the older ones are exhausted.

Payton et al., simulated pheromone based communication by attaching a platform at the top of the robots containing eight radially-oriented directional infrared receivers and transmitters. These infrared receivers and transmitters allow to transmit/receive 10-bit messages between the robots. The messages contain a parity check field to detect errors in the transmission and intensity field representing the intensity of the virtual pheromone, type field used to discriminate between different kind of virtual pheromones.and a hop-count field is used to detect the newest pheromone message when more than one copy of the same type of pheromone is received. On each step, the robots receive the virtual pheromone messages from the environment and send their own. If the robot decides to propagate the pheromone message it gets, it decrements the hop count and intensity of that pheromone message and sends it to the opposite direction of the direction of the original message received. The direction information is automatically obtained since the sensors are positioned radially around the robot. The same infrared sensors are also used to detect the obstacles in the environment.

Shen et al., [56], [57] used a similar approach to be able to simulate the diffusion of hormones in the environment. Although they did not test their ideas on real hardware, they claimed that the diffusion of hormones can be implemented using a short range wireless communication (either using RF or Infrared).

In their experiments, the robots broadcast packets containing the hormone type information. To implement the diffusion of the hormones, each receiver robot determines the direction (e.g. via a directional antenna) of the message and the distance of the signal source (e.g. by measuring the strength of the signal). The robot then applies diffusion function defined in the paper to compute the concentration of that particular hormone at the current and nearby cells. After collecting all hormonal signals coming from neighbor cells for some period of time, the robots computes the reaction of collected hormones and broadcast this information to simulate the diffusion of hormones.

6. Analytical Studies Axis

Analytical studies axis is defined to include studies which contributes to the theoretical understanding of swarm systems. These studies are the methods usable for different kind of problems or the studies which contributed some valuable mathematical tools to the swarm-robotics literature which allows us to understand deeper details of the swarm-robot systems.

We already discussed microscopic, macroscopic and cellular automata modeling in sections 3.2, 3.3 and 3.4 respectively. We suggest the interested readers to read those sections for further details of these studies.

Balch described two quantitative metrics for measuring diversity of robot groups in [5]. The author claimed that although existing multi robot studies contain homogeneous robot groups, the differences of robots in terms of hardware or behavior are totally ignored in the analysis. He claimed that by the introduction of these metrics the researchers can investigate the effect of group diversity with other metrics (e.g. performance).

The first metric called simple social entropy was based on Shannon's information entropy measure. Although it can be used in robotics experiments with small state/behavior space, it was not enough to capture the differences between the behaviors in fine granularities. The main weakness of simple social entropy is its lack of sensitivity to the distribution of the robots in the space. So, if we have two different configuration of robot groups in which P_is are equal, even if the distances of the robots from each other are different in these configurations, the metric returns the same value for both of the configurations.

To solve the problems of simple social entropy, Balch decides to use dendrogram concept from numerical taxonomy whose aim is to order organisms hierarchically. The dendrograms are taxonomic trees that allow to visualize relations of groups including the spatial distribution of elements in the system. The idea of hierarchical social entropy is to obtain a combination of simple social entropies in different levels. Balch used the hierarchical entropy for the clustering problem to show how metric works. He used C_u or u-diametric clustering method which allows overlaps up to the diameter of $u * h$ for the level h. He also tested the metrics on two different problems namely multi-foraging and simulated-soccer tasks.

Li et al., [43] used Balch‘s hierarchical social entropy metric to analyze the effect of diversity and specialization on a stick-pulling experiment in [4]. The details of [43] are discussed in “Local Reinforcement” sub-section under section 4.2.1.

7. Problems Axis

Problems trying to be solved in a research field has some pedagogical and practical importance in development of that field since they both help to understand the practical value of that field and help to divide the real problems into manageable sized sub-problems.

In this section, we identified the general problems tried to be solved in swarm robotics. Then the studies trying to solve these problems are described in corresponding sections from problem perspective. This section is especially valuable for the researchers who decide to solve a specific problem. With the help of this section, the researchers can easily locate some of the studies already have done and start their research easily.

After investigating the existing works in swarm-robotics, we decided to divide the problem axis into eight different problems: pattern formation, aggregation, chain formation, self-assembly, coordinated movement, hole avoidance, foraging and self-deployment.

7.1. Pattern Formation

Pattern formation can be defined as the emergence of global patterns from local agents and interactions. Since the definition of pattern goes much deeper topics like complexity, chaos and order, we would like to skip this discussion in this paper and just tell that pattern formation is an important phenomenon in nature and it is worth to research about it. The pattern formation is visible in all kinds of natural and social sciences. Interested readers may follow the literature about chaos and complexity to understand the relation of pattern with other important concepts in artificial life.

The pattern formation is important in swarm robotics too since coordinated behavior of a group of robots forms a pattern when viewed globally. And our main problem can be seen as creating these patterns with local processors and interactions.

Bahçeci and Şahin [3] presented a review of pattern formation and adaptation strategies in multi-robot systems. They divided pattern formation studies into centralized and decentralized pattern formation categories. They claimed that having a central unit assumption in centralized pattern formation makes the approach more costly, less robust to failures and less scalable. Bahçeci and Şahin also divided previous studies which used adaptation strategies into two categories: individual level and group level adaptation. Group level adaptation can only be obtained using a centralized control or allowing communication between agents to be able to share the information they obtained separately.

Fredslund and Mataric [18], [19] developed an algorithm for robot formation using local sensing and minimal communication. The algorithm works for a particular class of formations specifically the ones that can be folded from an open bicycle chain, keeping either the middle or the end of the chain in front.

The algorithm requires that each robot has a unique ID (IDs are only used for numerical comparison purpose. For example, the robot turns only to the robot with the lowest ID when more than one robot exist around it) and a friend sensor which is used to track the friend robot defined in the algorithm. The friend sensor needs to measure the relative direction of the friend robot and the distance of it. The friend robot can be the robot which has the nearest lower or the nearest higher ID depending on the target formation. Then the aim of the robots becomes tracking of the neighbors in a predefined angle which depends on the formation type and the number of robots in the experiments.

Although there are some deficiencies of this method (e.g. having a conductor robot and the requirement to know the total number of robots used in the experiment beforehand) the approach is still distributed and the algorithm still works with relaxed versions of the above requirements (e.g. the conductor robot can be changed online and an approximate total number of robots can be used). Having the freedom of applying the algorithm to different formations and the moving ability of the formation with the help of the conductor robot are other powerful features of this study. A formal evaluation criteria for robot formations is also defined in this study.

Trianni et al., [63] presented an architecture for pattern formation problem. In this architecture, they used a higher level of abstraction of sensor readings called context and the behavior of the robots are basically defined as the probability of applying an action in a given context. On each step, the context of the robot is found using the sensor readings. Then the firing of the actions in that context is being decided by with the well known roulette wheel selection. These probabilities are specified by hand before the experiments for both aggregation and chain formation tasks.

Both of the tasks are accomplished successfully with the help of some simplification assumptions like placing the robots in a hexagonal grid to simplify the implementation of connecting and disconnecting of the robots in the simulation.

The authors also tried to fit a macroscopic model to the experiments but the results showed that the mathematical model does not fit to the simulation experiments. The authors think that the possible problems are the lack of spatial information in the mathematical model, carrying out the simulation in discrete time and the lack of interaction dynamics in the model.

7.2. Aggregation

Aggregation problem requires aggregation of a randomly placed robots in an environment. The problem is easy when a centralized control approach is used but the problem is not trivial when the distributed control is used. The robots should behave autonomously and should use local information to aggregate. Selecting an approach like moving to the closest robot is not an acceptable solution since in that case several small number of groups occur. Aggregation has an important role for many biological systems because it is at the basis of the emergence of various forms of many collective tasks. The examples of aggregation in biological systems can be found in [9].

Trianni et al., tried to solve aggregation problem in [63] using a probabilistic controller. They used a higher level of abstraction than the sensor readings and actuator commands whose main elements are called as contexts (abstraction of sensor data) and behaviors (abstraction of actuator commands).They defined the probability of switching between behaviors in all contexts with a probability matrix and observed that the aggregation is possible with a predefined matrix in a simple sensor-based simulation environment.

Trianni et al., [64] also used genetic algorithms to evolve aggregation behavior by simply evolving the weights of a perceptron. The fitness function of the evolution is defined as the average distance of the robot group from its center of mass for each epoch. They observed two types of controllers in the final population:

the one which creates a very compact aggregate and the one which is looser than the previous one but moves as a group. It is observed that the latter one is more scalable when the number of robots are increased in the experiments.

Soysal and Şahin [58] performed systematic experiments using a probabilistic controller. There are four behaviors in the controller which are connected with subsumption architecture: obstacle avoidance, approach, repel and wait. The approach and repel behaviors are using a sound sensor to approach or repel from the loudest sound. The transitions between repel-approach and wait-repel states are defined with two different probabilities. A transition is achieved when a random number selected between zero and one is greater these probabilities. Soysal and Şahin investigated the behavior differences by testing different values of these probabilities. Interestingly they showed that the best performance is obtained when both of the parameters equal to 1 which means that the robot always tries to approach to the possibly biggest aggregate. But their point is that this approach will have the high risk of collision and will have large energy consumption because of the lack of the usage of wait state.

Bahçeci and Şahin [4] tried to achieve the aggregation behavior by evolving the weights of a neural network with 12 inputs and 3 outputs. While the first four of the input neurons encodes sound value obtained from the speaker, the remaining input neurons encodes the infrared sensors of the robot. Similarly, the first output neuron used to control the omni-directional speaker and the remaining two are used to control the wheels. They used the same sound sensor and emitter used in [58] to estimate the direction of the largest cluster. The fitness of a single evaluation is defined as the ratio of the number of robots forming the largest cluster to the total number of robots in the experiment. The fitness of a chromosome is computed in four different ways (average, median, minimum and maximum fitness of all runs) for comparison purposes.

Jeanson et al., [36] studied aggregation strategies in cockroaches. They tried to prove that cockroaches perform the global aggregation from local interactions. To do this they measured the important system parameters from the experiments with cockroach larvae like probability of stopping in an aggregate or probability of starting to move. They created a numerical model of behaviors of cockroaches from these measurements and tried to validate that their model in numerical simulations. Although their numerical model reveals a quantitative disagreement with experiments, they claimed that it also offers strong evidence that aggregation can be explained in terms of interactions between individuals which use only local information.

Mataric presented design of aggregation in [47]. Although the author showed some simulation screenshots as examples of the success of the behaviors and claimed that the experiments are tested on real robots, without giving any further details about the experiments. To do this robots tries to be within a determined distance from each others. The robots mainly determines the center of the robot groups based on local information and move forward to that position and if it finds itself within the determined distance then it just seems to stop in the given algorithm. The details of the experiments are not given to understand the global behaviors of the robots and needed interactions between other behaviors are hidden.

7.3. Chain Formation

In chain formation problem, the aim is to move the robots so that they form a chain pattern. This is useful for many applications like passing a corridor in a coordinated way and prohibiting other objects passing to some important place that needs to be guarded.

Nouyan and Dorigo [52] implemented a chain formation behavior in a sensor-based simulation. The robots have two phases: explorer and chain member. In explorer state, the robots search for other chain members or the nest. Whenever a robot finds the nest or chain member, the robot tries to keep permanent visual contact with it using an omni directional camera. The aim of the robots is to find the end of the chain and stay there when the explorer timeout is reached. The authors made systematic experiments by modifying the number of robots and the explorer timeout to see the changes in the speed of the chain formation process and the shape of the formed chains. It is observed that while short explorer timeout leads to the fast formation of many chains, a long explorer timeout results in the slow formation of fewer chains.

Nouyan [53] extended this work with more detailed configurations in his thesis. The author also used same behaviors for the problem of establishing a path towards a goal location from the nest.

7.4. Self-assembly

Self-assembly can be defined as creating more complex structures from large numbers of relatively simple units only with local interactions. In swarm-robotics, the relatively simple structures denotes the robots and the complex structures can be any global pattern or behavior obtained by the robots.

Grob et al., [25] defined the self-assembly problem as controlling the robots in fully autonomous manner in such a way that they locate, approach and connect with an object that acts as a seed or connect to other robots already connected to the seed. The seed and the robots connected to the seed are discriminated based on the color of the ring around them.

7.5. Coordinated Movement

Coordinated movement problem requires to keep a global pattern between robots while they are moving. It is a well studied behavior in biology, being observed in many different animal species [9]. As an example, the movement of flocks of birds or schools of fish can be considered as coordinated movement. The aim in the coordinated movement is keeping some global pattern while moving.

Hayes and Dormiani-Tabatabaei describe a leaderless distributed flocking algorithm in [27]. They used two behaviors to obtain flocking behavior: collision avoidance and velocity matching flock centering. Collision avoidance is activated when an agent's collision sensors detects an obstacle (it can be an environmental obstacle or another agent), and it mediates a turn away from the obstacle. Flock centering is active whenever the collision avoidance is not active. It generates a center of mass vector (CoM), CoM difference vector and a mapping result from those vectors to wheel speed commands. Although normally CoM is enough to implement flock behavior, they tracked the change of the CoM vector (CoM difference vector) to obtain an alignment term which may allow to get better performance.

Hayes and Dormiani-Tabatabaei used an off-line optimization method to optimize the unknown parameters in the model. After optimization is performed, they validated their results on real robots. Since the robot's sensors for obtaining relative range and bearing data are not available at the time of publishing this paper, they used an overhead camera to obtain this information.

7.6. Hole Avoidance

In hole avoidance, the aim of the robots is to move over or escape from the holes bigger than them with help of coordinated movement.

Trianni et al., [66], [65] tried to achieve hole-avoidance behavior with a swarm of robots. The robots have to perform coordinated motion in an environment which has holes too large to be traversed. The robots had to be connected with their turrets while sensing the environment. When a robot detects a hole using its ground sensors, it should move in the opposite direction of the hole. The force generated with this movement is detected by the traction sensors of other robots. The challenge is to learn reacting to these forces in an appropriate way so that hole-avoidance can be achieved independent from the robots' relative positions and position of the hole.

Trianni et.al used a genetic algorithm to obtain the controller of the robots. The controller was a simple perceptron which connects sensory inputs to the motor outputs of the robot. The weights of the perceptron are evolved using a fitness function which has three components. While the first component is measuring performance in terms of coordinated movement, second one is measuring exploration of the arena and the third one is using fast reaction to the detection of a hole. The evolved strategies are also tested on more difficult environments by varying the size and the shape of the robot swarms. It is observed that the strategies are successful and scalable.

7.7. Foraging

Foraging is one of the mostly used test applications in multi-robot systems. The aim of robots in a foraging task is to find the preys and bring them to the nest. This task is also known as prey retrieval task.

Steels [60] presented a distributed, self-organization based solution to multi-robot object aggregation problem. He described the solution in a clear way, starting from the possible logic based approach and its possible problems up to step-by-step construction of the self-organization based solution.

The solution was defined in terms of behaviors connected to each other with subsumption architecture. The solution is defined incrementally and each step introduced some performance improvement to the previous step. First random-movement, object-handling and obstacle-avoidance behaviors are introduced. Although these behaviors are sufficient to get a result in finite number of steps, it may take lots of time. In the second step, a gradient field is added around the home field so that the robots would be easily return the objects they picked up. Although this is an improvement to the first step, it is possible to improve the performance more by allowing communication between the robots. For this aim, a mechanism similar to the one used by ants is applied to the problem: crumb handling behavior. In this behavior, the robots drops 2 crumbs if they carry a sample. In addition to this, if the robots do not carry an object and crumbs are detected, the robots pick up one crumb. With this mechanism, a positive and a negative feedback mechanism is being added to the system and the communication between the robots are being handled using the environment itself without a need to have complex mechanisms to handle communication. The results are tested in simulation and compared.

Tangamchit et al., [62] used Monte Carlo learning method to solve foraging problem. They modified the foraging problem definition a little bit to allow cooperation between the robots. They defined a Home region where the collected pucks will be dropped at the center of it. Depositing a puck in the home region is made time consuming for the first robot but made it is easy for the second robot. The first robot can also navigate around whole environment while the second robot is restricted to move only in the home region. Therefore the best way to get best reward from the environment for both of the robots is designed to be the way in which the first robot finds pucks and transfers it to the second robot in the home region so that the second one can easily transfer it to the center of the home region. The details about the learning algorithm they used and their results are discussed in "Learning Axis" section previously.

7.8. Self-Deployment

The aim of robots in the self-deployment problem is to deploy themselves to the environment by covering the environment as much as possible. Since the robots are distributed to an unknown environment randomly and they have limited perception, the problem is non-trivial.

It is also worth to mention that solving the self-deployment problem implicitly solves the map building problem since the above requirement allows to obtain the map of the environment in a distributed way.

Howard et al., [30] described a distributed self-deployment algorithm which aims to maximize the area covered while simultaneously ensuring that each node can be seen by at least one other node. The developed algorithm is estimated to have a polynomial computation time complexity of order n^2 in the number of deployed nodes.

The noticeable feature of this work is that no global information or communication is used to develop the environment map. The algorithm is divided into four phases: initialization, selection, assignment and execution. The downside of the algorithm is that the selection and the assignment phases are computationally complex. This requires a live connection to a base station which is powerful enough to specify new target positions to deploy nodes (selection phase) and assign the most appropriate node to the target positions (assignment).

Four different selection policies are tested using sequential execution scheme. Since parallel execution of the nodes may fail because of the interferences, this scheme is left as a future work. The selection policies are tested based on two metrics: total coverage of the area by the nodes and total deployment time. The best policies are found to be between 70% and 85% of the value obtained for a greedy algorithm which has the map of the algorithm.

Howard et al., [31] also applied a distributed version of potential fields method [38], [1] to sensor network deployment problem. The nodes are treated as virtual particles subject to virtual forces. The virtual particles repels from each other and from obstacles. A virtual friction force is also added to the system to be able to reach static equilibrium since only dissipative systems whose total energy decreases can reach to static equilibrium. The performance of the method is analyzed with two metrics: total coverage of the area by the nodes and total deployment time.

The similarity of this approach to virtual physics method [59] is worth to notice since both of the approaches models the interactions in the environment as virtual forces.

8. Related Fields

There are some fields which may give the swarm robotics researchers some new insights. Of course, every scientific area is related to another one since the main aim of science can be seen as to better understand the whole phenomenon around us. But in the following section, we tried to present some of the fields which we think more related to swarm-robotics. Our presentation is of course limited, but our aim is to show the relation and some directions for the swarm-robotics researchers interested with those fields.

8.1. Distributed Artificial Intelligence

One of the main differences between the swarm-robotics and distributed artificial intelligence (DAI) is that DAI also supports deliberative or hybrid controllers for the robots rather than reactive ones. Swarm-robotics strictly discards deliberative and hybrid controllers because of its bias on simplicity. [15].

In spite of this difference, because of the remaining common features, DAI can be valuable for swarm robotics by giving some new ideas or formulations.

8.2. Self Organization

Many natural phenomenas around us can be considered to be complex systems. A complex system is a system whose properties are not fully explained by an understanding of its components. Complex systems have large number of interacting components which makes the analysis of these systems harder. The global behavior of these systems are said to be emerged from the interactions of their components.

Emergence [4] is a key property in complex systems which means the behavior of the complex system cannot be understood by examining only the components of the system. Although the components of the complex system can be simple, the resultant system may be complex because of the interactions of the system components.

Although lots of efforts spent for understanding how emergence occurs, there is no satisfactory theory explaining what characterizes emergence or what are the conditions for its existence. A promising approach for understanding the emergence of complex systems is self-organization [9]. Because of some key characteristics of self-organized systems (e.g., flexibility, scalability and robustness) self-organization became one of the main inspiration sources of swarm robotics idea.

Several self-organized models are developed for describing complex behaviors in physics, chemistry, biology and sociology. Unfortunately, to our best knowledge, reviews and books are only available on self-organization in biological systems [9] [17]. For other fields, the researchers need to locate the models one by one.

Since swarm robotics gets its inspiration mainly from self-organization, many of the studies in swarm robotics has some kind of inspiration from self-organized natural systems. The remaining part of this section will summarize those aspects of the mentioned studies.

Brooks [7] pointed some features of self-organized systems when he was defining the principles of behavior-based robotics. He claimed that complex behavior need not necessarily be a product of an extremely complex control system. Rather complex behavior may simply be the reflection of a complex environment. (It may be an observer who ascribe complexity to an organism.) Self-organized systems extends this assumption by pointing out that the complexity can be the result of interactions of simple entities in the environment. Brooks also pointed out that the control rules of the robots should be simple and robust.

Colorni et al., [11] take their inspiration from ant colonies for developing ant colony optimization method. The authors observe that some animals (e.g. bacterias, ants, caterpillars) exhibits complex collective behaviors even if they have poor individual capabilities. The authors mainly develop models for describing the behaviors of ant colonies. By assuming each ant as an individual processor and defining the problem as the habitat of the ants, the developed models are used to solve several optimization problems.

As described in the next section and the modeling section, cellular automata (CA) is a way of modeling complex systems including the self-organized ones [17], [12]. Shen et al., [56], [57] used CA modeling to model swarm behaviors. Their model gets its inspiration from self-organized formation of feathers. In chickens, the feathers are developed from homogeneous skin cells by first aggregating to form approximately same sized feather buds. Then these buds grow into different types of feathers depending on the region of the skin. The researchers find that the process of forming feather buds can be described with a self-organized model which mainly characterized by hormone diffusions between the homogeneous skin cells. They found that the size

[4] Synergy is used in some contexts instead of emergence.

of the feather buds remains approximately the same regardless of different population densities, but the size of the feather buds mainly depends on the profiles of the activator and inhibitor hormones secreted from the skin cells.

Shen et al., developed a model with pre-specified hormone models using these observations. They first validated their model with simulations in [56]. In [57], they extended these studies by solving different robotics problems. The details of their studies are discussed in section 8.3 in more detail.

Jeanson et al., [36] and Garnier et al., [22] develop a self-organized model of aggregation behavior of cockroaches in a bounded circular arena. They first performed experiments with real cockroaches to be able to obtain the values of parameters in their probabilistic model. Some example parameters approximated with this method are mean speed of the cockroaches and the probability of stopping in an aggregate.

The approach was similar to the microscopic modeling described in “Modeling Axis” section. At the end of this parameter estimation process, the authors tested the validity of their models by comparing the results of their model with the results of real experiments.

Labella et al., [41], [39], [40] tried to improve the performance of the foraging task with the help of task allocation. The authors inspired from a previously developed self-organized model of task allocation by Deneubourg and his colleagues. The model consists of updating the probability of leaving the nest depending on the previous success or failure of the agents. Labella et al., improved previous results of Deneubourg’s numerical model by validating it using real robots.

Nouyan and Dorigo [52], [53] inspired from the pheromone based communication of ants. Instead of developing a separate mechanism for simulating pheromone based communication, they used the robots as trail markers, in place of pheromone trails. They implemented a chain formation behavior to be able to connect the home of the robots to a prey. The robots were able to connect each other or explore the area using a timeout mechanism. The direction of the chain (the direction from home to prey) could be detected by the robots using color of the leds around the robots. There were three possible colors for the leds: red, green and blue. The robots connecting to the nest or a red chain member activates the color blue, the robots connecting to a blue chain member activates the color green and the ones connecting to a green robot activates the color red. With the help of this coloring scheme a robot observing just two of the chain members can easily determine the direction of the chain.

Payton et al., [54], [55] developed a more realistic model of the pheromone based communication of ants with the help of eight radially-oriented, directional infrared receivers and transmitters attached at the top of the robots. The pheromones are assumed to be transfered between the robots as 10-bit messages via infrared receivers and transmitters. Each robot retransmit a message it gets to the opposite direction by decrementing the hop count and intensity of that pheromone message. The method is mainly used to generate the map of an unknown area by a swarm of robots.

8.3. Cellular Automata

Cellular automata (CA) is among the simplest mathematical models of complex systems [34]. CA was initiated by J. Von Neumann in 1951 [51]. His aim was to model the biological evolution of organization. He designed a relatively complex dynamics to achieve self-reproduction on a two dimensional lattice. Until the publication of John Conway’s well known Life game (or ‘Game of Life‘) by Gardner [21], the scientists did not give much attention to the subject. The Life game changed this situation because of its ability to obtain complex dynamics even though having simple local rules. Later, the scientists from several different research fields (e.g. physics, chemistry, biology and sociology) started to use CA as a modeling tool for the phenomenas in their research area.

Ilachinski [34] specified five generic characteristics for CA models: discrete lattice of cells, homogeneity, discrete states, local interactions and discrete dynamics. The CA models contain discrete lattice of cells in one or more dimensions where each cell in the lattice has finite number of possible states. Each cell interacts only with the cells that are in its local neighborhood and the system dynamics is characterized by the local rules executed locally on the cells in discrete time steps.

When the above characteristics are observed, it is obvious that all these characteristics except the first one is shared by self-organized systems. So we can consider CA models as the mathematical modeling effort of self-organized systems. Discretization may allow us to simplify the analysis. Since discretization is highly accepted in robotics studies like discretizing time, sensor and actuator values, this is a preferable characteristics for swarm robotics. The only problem is implementing this in real robotics experiments. As we will see shortly in this section, Shen et al., [57] already solved this problem using a short range wireless communication mechanism.

Ilachinski [34] also discussed some extensions of the CA models characterized by the above characteristics. These are asynchronous CA which allows asynchronous updates by the lattice cells, coupled-map lattices in which the cell values can have arbitrary real values instead of a few discrete values, probabilistic CA which can have probabilistic state transitions instead of deterministic ones, non-homogeneous CA in which different cells can have different state transition rules, mobile CA in which some sites or cells are free to move on the lattice and structurally dynamic CA in which the lattice also influence the dynamics of the model by changing the parameters of the model (e.g. the connection between the sites).

Since swarm robotics studies cannot escape from its undeterministic nature because of the noise in the sensors/actuators and the interaction between the robots, it is obvious that CA based swarm robotics models should have characteristics of both mobile CA and probabilistic CA models.

Because of its underlying simplicity, CA is one of the preferred ways for developing analytical models of phenomenas (e.g. pattern formation) in natural sciences. Ilachinski [34] presents mathematical description of general CA models, examples of purely analytical tools useful for describing CA with relation to dynamical systems theory. Gutowitz [26] also presented some pioneering studies for mathematical analysis of CA inside a section in his book.

To the best of our knowledge, the only work connecting CA with swarm robotics is presented by Shen et al., [56], [57]. Shen et al., [56] presented a computational model for self-organization. The model called Digital Hormone Model (DHM) is a combination of stochastic cellular automata models and reaction-diffusion models. DHM is defined on a grid based world where living cells occupy one cell at a time and have only two actions: secretion and migration. Secretion produces activator and inhibitor hormones based on Gaussian distributions on every time step. Migration is done to a neighbor cell stochastically based on the hormone distribution on the neighbor cells. This probability is proportional to the concentration of activator hormone and inversely proportional to the concentration of inhibitor hormone.

The authors manage to show that the DHM enables the cells to form patterns which matches the observations made in the biological experiments of feather bud formation among uniform skin cells. Furthermore, they obtain different shaped patterns by changing the hormone diffusion profiles of the living cells.

Shen et al., [57] also successfully showed their method may also work on real robots with the help of a short range wireless communication (either RF or Infrared) as the implementation of the diffusion and reaction of hormones. They implemented DHM solutions for attacking target, area coveraging, self-repairing and barrier avoiding problems. The solutions are tested in simulation.

The modeling efforts using CA in biological modeling can be found in Ermentrout and Edelstein-Keshet's review [17]. Deutsch and Dormann also wrote a book about CA modeling of biological pattern formation [12]. For the usage of CA in other domains, the review prepared by Ganguly et al., [20] can be a nice starting point. The books written by Ilachinski [34] and Gutowits [26] can be used as starting points for CA research.

9. Conclusions

In this paper we have presented a preliminary taxonomy for swarm robotics and classified existing studies into this taxonomy. Before developing our own taxonomy, we investigated the existing surveys related to swarm robotics literature.

After investigating related literature surveys, we selected the main and sub dimensions of our taxonomy mainly based on the publications we identified inside swarm robotics field. The main dimensions were modeling, behavior design, communication, analytical studies and problems. In this list; problems can be considered as auxiliary dimensions because it was not critical/relevant as much as other dimensions for/with the performance of swarm systems. The problems were presented as supplementary views to the previous works and for their pedagogical value.

In contrast to problems dimension; modeling, behavior design, communication and analytical studies were the core dimensions since they are important factors while designing a swarm system and different selections of these factors can effect performance considerably.

As described in the paper, modeling is a necessity in the current state of robotic technology and each modeling method has its own assumptions which shows a different deviation path from reality. Behavior design is another important factor while designing a swarm system since the swarms with adaptation capabilities may perform much better in their environments.

Communication was another core dimension for swarm robotics because communication is one of the core elements that makes a multi robot system preferable to single robot systems. With the help of communication, a group of robots can accomplish their tasks better than single robots or they can accomplish the missions which cannot be performed by single robots.

Although analytical studies dimension can be considered either as a core or auxiliary dimension, we selected to consider it as a core dimension here. What we call analytical study and the value of these studies are described in the corresponding section.

Although we know that this is not a complete review, we believe this is a nice step towards more complete swarm robotics literature surveys.

Acknowledgements

This work was partially funded by the "KARİYER: Kontrol Edilebilir Robot Oğulları" Career Project (Project no: 104E066) awarded to Erol Şahin by TÜBİTAK (Turkish Scientific and Technical Council).

References

[1] R. Arkin, "Motor Schema-Based Mobile Robot Navigation", The International Journal of Robotics Research 8 92–112, 1989.

[2] R. Arkin, Behavior-Based Robotics, The MIT Press, 0262011654, 1998.

[3] E. Bahçeci, O. Soysal, E. Şahin, "A Review: Pattern Formation and Adaptation in Multi-Robot Systems", Technical Report CMU-RI-TR-03-43. Carnegie Mellon Univ, Pittsburgh, PA, USA, October 2003.

[4] E. Bahçeci, E. Şahin, "Evolving Aggregation Behaviors for Swarm Robotic Systems: A Systematic Case Study", Proc. of the IEEE Swarm Intelligence Symposium, Pasadena, California, 2005.

[5] T. Balch, "Hierarchic Social Entropy: An Information Theoretic Measure of Robot Group Diversity", Autonomous Robots, vol. 8, pp. 209-238, 2000.

[6] T. Balch, M. Hybinette, "Social Potentials for Scalable Multi-Robot Formations", IEEE International Conference on Robotics and Automation (ICRA-2000), San Francisco, 2000.

[7] R. Brooks, "A robust layered control system for a mobile robot", IEEE Journal of Robotics and Automation, 2(1):14– 23, 1986.

[8] R. Brooks, P. Maes, M. Mataric, G. More, "Lunar base construction robots", In Proc. IEEE Work- shop on Intelligent Robots and Systems, Tsuchiura, Japan, 1990.

[9] S. Camazine, J. Deneubourg, N. Franks, J. Sneyd, G. Theraulaz, E. Bonabeau, Self-Organization in Biological Systems, Princeton University Press, 0691012113, 2001.

[10] Y. Cao, A. Fukunaga, A. Kahng, "Cooperative Mobile Robotics: Antecedents and Directions", Autonomous Robots, vol. 4, pp. 7-23, 1997.

[11] A. Colorni, M. Dorigo, V. Maniezzo, "Distributed optimization by ant colonies", In F. Varela and P. Bourgine, editors, Proceedings of the European Conference on Artificial Life, pages 134–142, Amsterdam, ECAL, Paris, France, Elsevier, 1991.

[12] A. Deutsch, S. Dormann, Cellular Automaton Modeling of Biological Pattern Formation: Characterization, Applications, and Analysis, A Birkhauser book, 0817642811, 2005.

[13] T. De Wolf, G. Samaey, T. Hovoet, Analysis and Synthesis of a Bio-inspired Swarm Robotic System", In E. Şahin, W. Spears and A. Winfield, editors, Proceedings of the Second International Workshop on Swarm Robotics at SAB 2006, volume 4433 of Lecture Notes in Computer Science, pages 56-70. Springer Verlag, Berlin, Germany, 2006.

[14] Heiko Hamann and Heinz Worn, "An Analytical and Spatial Model of Foraging in a Swarm of Robots", In E. Şahin, W. Spears and A. Winfield, editors, Proceedings of the Second International Workshop on Swarm Robotics at SAB 2006, volume 4433 of Lecture Notes in Computer Science, pages 43-55. Springer Verlag, Berlin, Germany, 2006.D. Roose, "Decentralised Autonomic Computing: Analysing Self-Organising Emergent Behaviour using Advanced Numerical Methods", Proceedings of the 2nd International Conference on Autonomic Computing (ICAC'05), IEEE Computer Society Press, pp 52-63 June 2005.

[15] M. Dorigo, E. Şahin, "Swarm Robotics - Special Issue", Autonomous Robots, vol. 17, pp. 111-113, 2004.

[16] G. Dudek, E. Jenkin, D. Wilkes, "A taxonomy for swarm robots", In Proc. 1993 IEEE International Conference on Intelligent Robots and Systems, pp 441–447, 1993.

[17] B. Ermentrout, L. Edelstein-Keshet, "Cellular automata approaches to biological modeling", Journal of Theoretical Biology, 160:97-133, January 1993.

[18] J. Fredslund, M. Mataric, “Robots in Formation Using Only Local Sensing and Control”, The 7th International Conference on Intelligent Autonomous Systems (IAS-7), Marina del Rey, California, USA, March 25-27, 2002.

[19] J. Fredslund, M. Mataric, “A General Algorithm for Robot Formations Using Local Sensing and Minimal Communication”, IEEE Transactions on Robotics and Automation, 18(5):837-846, 2002.

[20] N. Ganguly, B. Sikdar, A. Deutsch, G. Canright, P. Chaudhuri, “A survey on cellular automata”, Technical report, Centre for High Performance Computing, Dresden University of Technology, December 2003.

[21] M. Gardner, “Mathematical Games: The Fantastic Combinations of John Conway’s New Solitarire Game Life”, Scientific American, Volume (4)223, 120-123, Oct. 1970.

[22] S. Garnier, C. Jost, R. Jeanson, J. Gautrais, M. Asadpour, G. Caprari, G. Theraulaz, “Collective decision-making by a group of cockroach-like robots”, In Proceedings of the 2nd IEEE Swarm Intelligence Symposium, Pasadena, California, USA, 8-10, BEST PAPER AWARD, June 2005.

[23] V. Gazi, B. Fidan, “Coordination and Control of Multi-agent Dynamic Systems: Models and Approaches”, In E. Şahin, W. Spears and A. Winfield, editors, Proceedings of the Second International Workshop on Swarm Robotics at SAB 2006, volume 4433 of Lecture Notes in Computer Science, pages 71-102. Springer Verlag, Berlin, Germany, 2006.

[24] D. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Professional, 0201157675, 1989.

[25] R. Grob, M. Bonani, F. Mondada, M. Dorigo, “Autonomous Self-assembly in a Swarmbot”, In K. Murase, K. Sekiyama, N. Kubota, T. Naniwa, and J. Sitte, editors, Proceedings of the Third International Symposium on Autonomous Minirobots for Research and Edutainment, pages 314–322. Springer Verlag, Berlin, 2006.

[26] H. Gutowitz, Cellular Automata: Theory and Experiment, The MIT Press; 1st Mit Pr edition, 0262570866, 1991.

[27] A. Hayes, P. Dormiani-Tabatabaei, “Self-Organized Flocking with Agent Failure: Off-Line Optimization and Demonstration with Real Robots”, Proc. of the 2002 IEEE Int. Conf. on Robotics and Automation, Washington DC, USA, pp. 3900-3905, May 2002.

[28] S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice Hall, 0132733501, 1998.

[29] J. Hertz, A. Krogh, R. Palmer, Introduction to the Theory of Neural Computation, Perseus Books Group, 0201515601, 1991.

[30] A. Howard, M. Mataric, G. Sukhatme, “An Incremental Self-Deployment Algorithm for Mobile Sensor Networks”, Autonomous Robots, Special Issue on Intelligent Embedded Systems, 2002.

[31] A. Howard, M. Mataric, G. Sukhatme, “Mobile Sensor Network Deployment using Potential Fields: A Distributed, Scalable Solution to the Area Coverage Problem”, DARS 02, Fukuoka, Japan, June 2002.

[32] A. Ijspeert, A. Martinoli, A. Billard, L. Gambardella, “Collaboration through the exploitation of local interactions in autonomous collective robotics: The stick pulling experiment”, Autonomous Robots, vol. 11, no. 2, pp. 149–171, Kluwer Academic Publishers,Analysis and Synthesis of a Bio-inspired Swarm Robotic System”, In E. Şahin, W. Spears and A. Winfield, editors, Proceedings of the Second International Workshop on Swarm Robotics at SAB 2006, volume 4433 of Lecture Notes in Computer Science, pages 56-70. Springer Verlag, Berlin, Germany, 2006.

[33] Heiko Hamann and Heinz Worn, "An Analytical and Spatial Model of Foraging in a Swarm of Robots", In E. Şahin, W. Spears and A. Winfield, editors, Proceedings of the Second International Workshop on Swarm Robotics at SAB 2006, volume 4433 of Lecture Notes in Computer Science, pages 43-55. Springer Verlag, Berlin, Germany, 2006. 2001.

[34] A. Ilachinski, Cellular Automata: A Discrete Universe, World Scientific Publishing Company, 9810246234, 2001.

[35] L. Iocchi, D. Nardi, M. Salerno, "Reactivity and Deliberation: A Survey on Multi-Robot Systems", Balancing Reactivity and Social Deliberation in Multi-Agent Systems, From RoboCup to Real-World Applications (selected papers from the ECAI 2000 Workshop and additional contributions), 3-540-42327-3, 9–34, Springer-Verlag, London, UK, 2001.

[36] R. Jeanson, C. Rivault, J. Deneubourg, S. Blancos, R. Fourniers, C. Jost, G. Theraulaz, "Self-Organized aggregation in cockroaches", Animal Behaviour, 69, 169-180, 2005.

[37] I. Kevrekidis, C. Gear, J. Hyman, P. Kevrekidis, O. Runborg, C. Theodoropoulos, "Equation-free, coarse-grained multiscale computation: enabling microscopic simulators to perform system-level tasks", Communications in Mathematical Sciences, 1:715-762, 2003.

[38] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots", The International Journal of Robotics Research 5(1):90–98, 1986.

[39] T. Labella, M. Dorigo, J. Deneubourg, "Self-Organised Task Allocation in a Group of Robots", In R. Alami, editor, Proceedings of the 7th International Symposium on Distributed Autonomous Robotic Systems (DARS04). Toulouse, France, June 23-25, 2004.

[40] T. Labella, M. Dorigo, J. Deneubourg, "Efficiency and Task Allocation in Prey Retrieval", In A.J. Ijspeert, D. Mange, M. Murata, and S. Nishio, editors, Proceedings of the First International Workshop on Biologically Inspired Approaches to Advanced Information Technology (Bio-ADIT2004), Lecture Notes in Computer Science, pages 32-47. Springer Verlag, Heidelberg, Germany, 2004.

[41] T. Labella, "Prey Retrieval by a Swarm of Robots", Technical Report TR/IRIDIA/2003-16, IRIDIA, Universit Libre de Bruxelles, Brussels, Belgium, DEA thesis, May 2003.

[42] K. Lerman, A. Martinoli, A. Galstyan, "A Review of Probabilistic Macroscopic Models for Swarm Robotic Systems", Proc. of the Swarm Robotics Workshop at the Eight Int. Conference on the Simulation of Adaptive Behavior SAB-04, E. Şahin and W. Spears, editors, July 2004, Los Angeles, CA. Lecture Notes in Computer Science, 2004.

[43] L. Li, A. Martinoli, Y. Abu-Mostafa, "Learning and Measuring Specialization in Collaborative Swarm Systems", In: Adaptive Behavior, vol. 12, num. 3–4 (2004), p. 199–212, 2004.

[44] A. Martinoli, F. Mondada, "Collective and cooperative group behaviours: Biologically inspired experiments in robotics", In O. Khatib and J. K. Salisbury, editors, Proceedings of the Fourth International Symposium on Experimental Robotics ISER-95, pages 3–10, Stanford, U.S.A., Springer Verlag, June 1995.

[45] A. Martinoli, K. Easton, "Modeling Swarm Robotic Systems", Proc. of the Eight Int. Syrup. on Experimental Robotics ISER-02, Sant'Angelo d'lschia, Italy, July, 2002. Springer Tracts in Advanced Robotics, pp. 285-294, 2003.

[46] A. Martinoli, K. Easton, W. Agassounon, “Modeling Swarm Robotic Systems: A Case Study in Collaborative Distributed Manipulation”, Special Issue on Experimental Robotics, B. Siciliano, editor, Int. Journal of Robotics Research, Vol. 23, No. 4, pp. 415-436, Invited paper, 2004.

[47] M. Mataric, “Designing Emergent Behaviors: From Local Interactions to Collective Intelligence”, Proceedings, From Animals to Animats 2, Second International Conference on Simulation of Adaptive Behavior (SAB-92), J-A. Meyer, H. Roitblat and S. Wilson, eds., MIT Press, 432-441, 1992.

[48] M. Mataric, “Reinforcement Learning in the Multi-Robot Domain”, Autonomous Robots 4, 73-83, 1997.

[49] T. Mitchell, Machine Learning, McGraw-Hill Science/Engineering/Math, 0070428077, 1997.

[50] R. Montemanni, L. Gambardella, “Swarm approach for a connectivity problem in wireless networks”, Proceedings of the IEEE Swarm Intelligence Symposium (SIS 2005), 265-272, Pasadena, U.S.A., June 2005.

[51] J. Neumann, “The general and logical theory of automata”, L.A. Jeffress, ed., Cerebral Mechanisms in Behavior - The Hixon Symposium, John Wiley & Sons, New York, pp. 1-31, 1951.

[52] S. Nouyan, M. Dorigo, “Chain Formation in a Swarm of Robots”, Technical Report TR/IRIDIA/2004-18, IRIDIA - University Libre de Bruxelles, Belgium, March 2004.

[53] S. Nouyan, “Path Formation and Goal Search in Swarm Robotics”, Technical Report TR/IRIDIA/2004-14, IRIDIA - University Libre de Bruxelles, Belgium, DEA Thesis, September 2004.

[54] D. Payton, M. Dally, R. Estkowski, M. Howard, C. Lee, “Pheromone robotics”, Autonomous Robots, 11(3), 2001.

[55] D. Payton, R. Estkowski, M. Howard, “Pheromone Robotics and the Logic of Virtual Pheromones”, SAB 2004: 45-57.

[56] W. Shen, C. Chuong, P. Will, “Simulating Self-Organization for Multi-Robot Systems”, International Conference on Intelligent and Robotic Systems, Switzerland, 2002.

[57] W. Shen, P. Will, A. Galstyan and C. Chuong, “Hormone-Inspired Self-Organization and Distributed Control of Robotic Swarms”, Autonomous Robots, vol. 17, pp. 93-105, 2004.

[58] O. Soysal, E. Şahin, “Probabilistic Aggregation Strategies in Swarm Robotic Systems”, Proc. of the IEEE Swarm Intelligence Symposium, Pasadena, California, 2005.

[59] W. Spears, D. Spears, J. Hamann, R. Heil, “Distributed, Physics-Based Control of Swarms of Vehicles”, Autonomous Robots, Volume 17(2-3), August 2004.

[60] L. Steels, “Cooperation between distributed agents through self-organisation”, In Proceedings of the First European Workshop on Modelling Autonomous Agents in a Multi-Agent World, Elsevier Science Publishers Holland, 175–196, 1990.

[61] R. Sutton, A. Barto, Reinforcement Learning: An Introduction, The MIT Press, 0262193981, 1998.

[62] P. Tangamchit, J. Dolan, P. Kosla, “The necessity of average rewards in cooperative multirobot learning”, In IEEE International Conference on Robotics and Automation. ICRA ’02., pages (2)1296– 1301, 2002.

[63] V. Trianni, T. Labella, R. Grob, E. Şahin, M. Dorigo, J. Deneubourg, “Modeling Pattern Formation in a Swarm of Self-Assembling Robots”, Technical Report TR/IRIDIA/2002-12, IRIDIA, Universit Libre de Bruxelles, Bruxelles, Belgium, May 2002.

[64] V. Trianni, R. Grob, T. Labella, E. Şahin, M. Dorigo, "Evolving Aggregation Behaviors in a Swarm of Robots", In W. Banzhaf, T. Christaller, P. Dittrich, J. T. Kim and J. Ziegler, editors, Advances in Artificial Life - Proceedings of the 7th European Conference on Artificial Life (ECAL), Lecture Notes in Artificial Intelligence 2801, pages 865-874, Springer Verlag, Heidelberg, Germany, 2003.

[65] V. Trianni, M. Dorigo, "Emergent Collective Decisions in a Swarm of Robots", In Proceedings of the 2005 IEEE Swarm Intelligence Symposium (SIS 2005), pages:241-248 June 8-10, 2005

[66] V. Trianni, S. Nolfi, M. Dorigo, "Cooperative Hole Avoidance in a Swarm-bot", Robotics and Autonomous Systems, Volume 54, number 2, pp. 97-103, 2005.

[67] C. Watkins, "Learning from Delayed Rewards", Thesis, University of Cambidge, England, 1989.

[68] E. Yang, D. Gu, "Multiagent Reinforcement Learning for Multi-Robot Systems: A Survey", CSM-404, Technical Reports of the Department of Computer Science, University of Essex, 2004.