A New Model for Solving the Simultaneous Object Collecting and Shepherding Problem in Flocking Robots

Ellips Masehian and Mitra Royan
Industrial Engineering Department, Tarbiat Modares University, Tehran, Iran

Keywords: Robot Flocking Systems, Shepherding, Object Collecting, Fuzzy Expert System, Obstacle Avoidance.

Abstract: Shepherding behavior is a class of collective behaviors in flocking systems which requires that a swarm of mobile robots enter an area populated with known or unknown obstacles, collect a flock of static or dynamic particles (objects), and guide them safely to a predefined goal position. Applications of this behavior are in sheep or duck shepherding and fishing. In this paper, a new algorithmic model is developed for online formation control, decision making, behavior selection, and motion planning of a team of homogeneous and anonymous (no leader and follower) flocking robots which simultaneously perform object collecting and shepherding tasks. The model's architecture is enriched with various complex flocking actions such as flock deformation, flock split and merge, flock expansion, and flock obstacle avoidance. Contributions of this paper include (*i*) defining a new class of problems for flocking robots called Simultaneous Object Collecting and Shepherding (SOCS) problem, (*ii*) incorporating online obstacle sensing and avoidance methods in the flocking behavior, and (*iii*) developing a fuzzy expert system for determining the strategy of environment exploration. The fuzzy inference engine provides an effective way to minimize the time spent on collecting objects while maximizing the gain obtained by object collection, in a way that the flock's formation and integrity is maintained. The proposed model was implemented on a number of simulations and produced rational and satisfactory results.

1 INTRODUCTION

Swarm robotics is an interesting branch of artificial intelligence, which is inspired from natural behaviors of bees, ants, fish, birds, etc. Flocking, as a basic collective behavior in swarm robotic systems, has being studied for a decade. In general, flocking is a natural phenomenon where a group of animals move together as a single entity. The motion of flocking robots is a result of integrated actions of all members in the group, such that each member acts based on a local perception of its surrounding.

Reynolds (1987) proposed the following three fundamental rules for simulating flocking and herding behaviors:

Separation: when flock members get very close to each other (closer than a 'repulsion range'), they must move away each other via a repulsive force. As a result, sufficient free space around each member is guaranteed.

Alignment: each member should be moving along the general direction of its neighboring members.

Cohesion: members should move toward the center of its local neighbors. As a result, they stay close to the group, until they sense repulsive forces.

The logic behind these rules is that while each individual follows relative simple rules, when taken as a whole, they move as an organized group. Brett (2009) presented many applications for flocking behaviors, like mobile sensor network, surveillance, control and covering problems, or transporting large objects. The whole group tries to adjust its velocity and align with other agents in the flock, while maintaining the predetermined pattern and avoiding obstacle collisions, and move toward the goal while trying to minimize collisions between the members of the flock.

There are varieties of problems in the literature that require and utilize flocking as a behavior of swarm robots. Many problems are demonstrated in different environments which may be totally unknown or partially known to the group. Some of them consider leader–follower models, where the flock leader's velocity may or may not change during the task. The way the robots communicate with each other is important for the flock's successful

Masehian E. and Royan M..
A New Model for Solving the Simultaneous Object Collecting and Shepherding Problem in Flocking Robots.
DOI: 10.5220/0004161600960105
In *Proceedings of the 4th International Joint Conference on Computational Intelligence* (ECTA-2012), pages 96-105
ISBN: 978-989-8565-33-4

task execution. Generally, they have a local communication and should enter the environment, obtain information about the surrounding, and update and share their acquired information.

In the following we categorize the main approaches of solving flocking problems in free space or in presence of multiple obstacles:

Leader–follower Methods: In Leader-Follower approaches, one robot assumes the leader role and the rest of the flock follows it. The leaders use a tracking strategy to lead the flock toward the destination. In general, one agent acts as a group leader and the others just follow the separation, alignment, and cohesion rules, resulting in leader following (e.g., Xiong et al. (2008)).

Roadmap–based Methods: Searching and moving toward the goal in this type of flocking problems is accomplished based on the global information and the roadmap of the environment imposed on the system. Bayazit et al. (2002) proposed three distinct group behaviors: homing, exploring and shepherding, that exploit global knowledge of the environment with the use of medial axis probabilistic roadmap.

Control Theory–based Methods: In this approach, each robot has to follow a certain control theory law to converge to a stable state. These control laws can be used to coordinate the motion of each flock member that is capable of local sensing and communication, and can be related to both kinematics and dynamics of robots (e.g., Sharma et al. (2009) and Navarro et al. (2008)).

Fault Tolerant Methods: These types of methods assume that the flock there is a possibility of a faulty robot to fail during a task execution such that the crash can be either permanent, or temporary and recoverable in future. Also, there is a model in leader-follower flocks when the leader crashes and the group choose another leader to guide the flock.

1.1 Shepherding

Shepherding is an interesting flocking behavior: it is a cooperative task of controlling a group of agents by one or more groups of agents via employing repulsive forces. In the literature there are single and multiple shepherd variations for the shepherding behavior, of which the multi robot type can be viewed as a kind of task manipulation that has applications more than just herding a group of animals.

Brett (2009) proposed different cooperative applications for shepherding behaviors like collecting oil spilt from oil tankers, keeping animals off of airport runways, and keeping people from dangerous areas such as unsafe waters, construction zones or other restricted areas. In spite of this, shepherding has received little attention up to now, and there are many open problems to be worked in future.

In the literature, shepherding has been used merely for controlling and directing a number of known objects toward a goal, sometimes in presence of obstacles. By considering the influence of the shepherd's (robots) motion on the flock (objects), the flock can be prevented from scattering and can be controlled easier. Christopher et al. (2010) showed in the robot sheepdog project how a robotic system that gathers a flock of ducks in a circular arena based on the potential field algorithm is used to generate movements for each duck and maneuver them safely to a predetermined goal position. Garrell et al. (2009) proposed a new approach for guiding people in open areas of urban settings by using multiple robots acting in a cooperative way.

2 THE SOCS PROBLEM

In all of the shepherding-related researches it is assumed that the collectible objects (particles), as well as workspace obstacles, are fully known. However, in some real-world applications like fishing there is no information about the number and distribution of collectible objects (e.g. fish). Information about obstacles is also missing when operating in unknown environments. Therefore, the flock must identify and collect objects, while simultaneously shepherding them toward a goal region.

In this paper we propose a new class of problems called "Simultaneous Object Collecting and Shepherding (SOCS)" for flocking robots. The SOCS problem has some real-world applications, such as collecting distributed mines in an unsafe area, collecting oil spills or trashes off the sea, casting a fish net and directing the hunted fish toward the ship (an instance of 3D space problem).

In offline mode, when there exists a full knowledge about the workspace (including objects and obstacles) before the robots start their task execution, the SOCS problem is analogous to the Traveling Salesman Problem (TSP), in which a salesman starts his trip from a city, visits each and every city he plans to visit only once, and return to his starting city. Mathematically, the TSP is about finding a Hamiltonian tour on a given graph, which is an NP-hard problem, meaning that the time to optimally solve the problem grows exponentially as the number of cities increases. In fact, we can draw parallels between cities in the TSP and objects (or clusters of objects) in the SOCS, and between the salesman in the TSP and the flock in the SOCS. The only difference is that the flock should not necessarily return to its starting position, and that the flock is not limited to visit a certain location only once (this relaxation still does not reduce the NP-hardness of the problem).

In online mode, however, the robots must acquire environmental knowledge through their sensors, both about collectible objects and obstacles, and so the SOCS problem interweaves the shepherding task with sensor-based motion planning and obstacle avoidance. In the SOCS problem we assume that collecting each object by the flock has a gain or reward, and the flock has a limited time to execute its task. The ideal situation would be to collect all objects and direct them to the goal point in minimum time. Put differently:

The SOCS problem is to maximize the gain of collecting objects by a flock while minimizing the total time.

This problem, however, is NP-hard in both offline and online modes, and so finding the optimal solution is not practical for large number of objects. Instead, we have proposed a heuristic method to overcome the complexity and produce a collective behavior for gathering scattered objects and shepherding them toward the goal region in online mode.

The main contributions of this paper include:

(i) Defining a new class of problems for flocking robots called the Simultaneous Object Collecting and Shepherding (SOCS) problem,

(ii) Incorporating online obstacle sensing and avoidance methods in the flocking behavior, and

(iii) Developing a fuzzy expert system for determining the strategy of environment exploration. The fuzzy inference engine provides an effective way to minimize the time spent on collecting objects while maximizing the gain obtained by object collection, in a way that the flock's formation and integrity is maintained.

The proposed model was implemented in a number of simulations and produced rational and satisfactory results.

3 OUTLINE OF THE PROPOSED MODEL

Our proposed model for solving the online SOCS problem is composed of two main 'Exploration' and 'Exploitation' behaviors, and two auxiliary 'Fuzzy Expert System' and 'Motion Planning' modules.

The Exploration behavior is adopted when the flock intends to explore the environment for collecting objects. Here the main emphasis is on covering the environment as much as possible and moving toward regions with dense population of objects, as temporary goals. On the other hand, the Exploitation behavior is triggered when the flock has collected sufficient number of objects, or the available time is nearly over. In this case, the flock heads toward the final goal and collects all objects on its way.

The Fuzzy Expert System Module is utilized for deciding about where the flock should move to collect more objects (hence more gain), and when to stop collecting and move toward the final goal, such that the task is finished within a time limit.

The Motion Planning Module implements the Potential Fields method for helping the flock to avoid obstacles locally, and move toward either the final goal region or a temporary goal near a cluster of collectible objects. The module also decides about executing some complex actions like stretching, shrinking, splitting and merging. In this way, the flock becomes a deformable and coherent group, which during its navigation in the environment, can shrink or elongate to pass through narrow passages, or split and merge when encountered with obstacles or corridors (while retaining its connectivity and not losing any collected object), and shepherd the objects toward the goal region.

The model's assumptions are as follows:

1. The workspace is planar, bordered, and initially unknown to the robots. It contains static polygonal obstacles which should be avoided.

2. The robots are homogeneous, circular, and can move in the workspace without kinodynamic constraints. They are equipped with range sensors for identifying both obstacles within the range R_{obs} (Figure 1) and particles within the range $R_{part} < R_{obs}$. We also assume that there are no localization and sensing errors.

3. The robots form a flock by taking on the shape of a circular arc, with its open segment facing forward. The flock's integrity is maintained by regulating and equalizing the robots' velocities with each other. The flock must finish its task within a time limit T_{max} and collect at least Q_{min} particles.

4. The particles are small circular objects scattered over the workspace, which may be fixed or moving. Collecting a particle has a gain for the flock.

5. The goal region is known to the robots and once the flock's center lies inside that region the search is terminated.

Table 1 introduces some of the more important variables and parameters of the model.

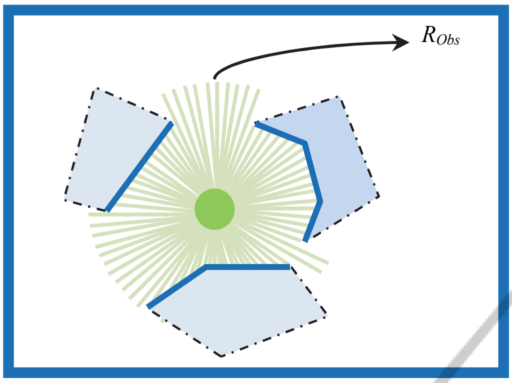

Figure 1: Identifying the surrounding obstacles through range-finder sensors.

Table 1: Variables and parameters of the model.

Symbol	Description
$\mathbf{X}_R(t)$	Position vector of robots at time t
$\mathbf{X}_P(t)$	Position vector of particles at time t
$\mathbf{V}_R(t)$	Velocity vector of robots at time t
$\mathbf{V}_P(t)$	Vector of particles velocities at time t
$Q(t)$	Number of collected particles inside the flock at time t
$D(t)$	Distance between the flock's center and the final goal at time t
$C(n)$	Capacity of the flock with n robots; $n = 1, \ldots, N$
R_{obs}	Robots' sensing range for detecting obstacles
R_{part}	Robots' sensing range for detecting particles
D_{Rmax}	Maximum distance between two neighboring robots for maintaining connectivity
D_{Rmin}	Minimum distance between two neighboring robots for avoiding collision
R_F	Radius of the flock's circular shape
S_p	Safety radius for particle p
G_p	Gain of collecting particle p
T_{max}	Upper bound of the allowable time interval
T_{min}	Lower bound of the allowable time interval
Q_{min}	Minimum required number of collected particles

In the beginning, N robots reside in a Depot, and an initial number of them (calculated based on the parameters D_{Rmax} and D_{Rmin}) are selected to form the flock by adjusting their positions on the circumference of a circular arc with radius R_F (Figure 2). The arc's angular span is between 180 and 270 degrees, with its open segment facing toward the moving direction. When the flock collects as much particles as it can accommodate (i.e., $C(n)$), it checks the possibility (regarding time and cost) of an expansion by incorporating one or two robots settled in the depot. The flock explores the workspace by being attracted to areas with higher number of objects until either there is no object left, or the available time is over. The overall architecture of the model is shown in Figure 3.

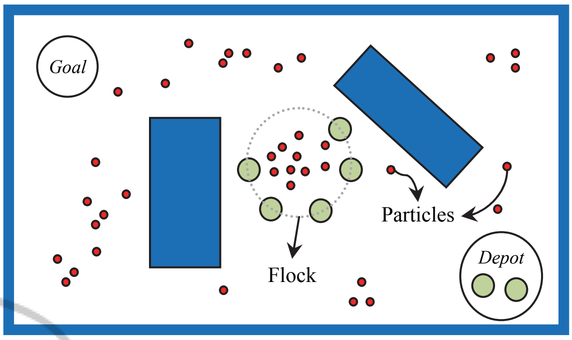

Figure 2: Simultaneous object collecting and Shepherding: The robots collect objects by trapping them inside their arc-shaped flock and direct them toward the goal.

4 FLOCKING BEHAVIORS AND ACTIONS

Our proposed flocking system has two basic behaviors: Exploration (covering the environment to find as much particles as possible) and Exploitation (moving toward the final goal). These techniques are applied to the entire flock as an integrated shape. Besides, other actions like traversing through narrow passages, splitting, merging and deformation can occur during the Exploration and Exploitation.

4.1 Exploration Behavior

In the Exploration behavior, each robot senses its surrounding within the range R_{part} and finds a number of particles around it. Then, all robots communicate their obtained knowledge of environment, and by integrating the whole knowledge, create a map of the distribution of nearby particles. The sensed objects are then clustered into a few groups, and the group with the most particles (and hence, the highest gain) is marked for exploration. The center of this cluster is fixed as a temporary goal and the flock starts moving towards it. The flock's motion is guided and obstacles are avoided using the Potential Fields method (discussed in section 4.3). An area is considered explored when the flock passes over it.

This collaborative effort of exploring the environment is repeated from a temporary goal to another until either there are no sensed but uncollected particles left, or the flock cannot accommodate more objects due to fullness of its capacity. The capacity $C(n)$ of a flock with n robots is determined based on the safety radius of particles (S_p), and the maximum

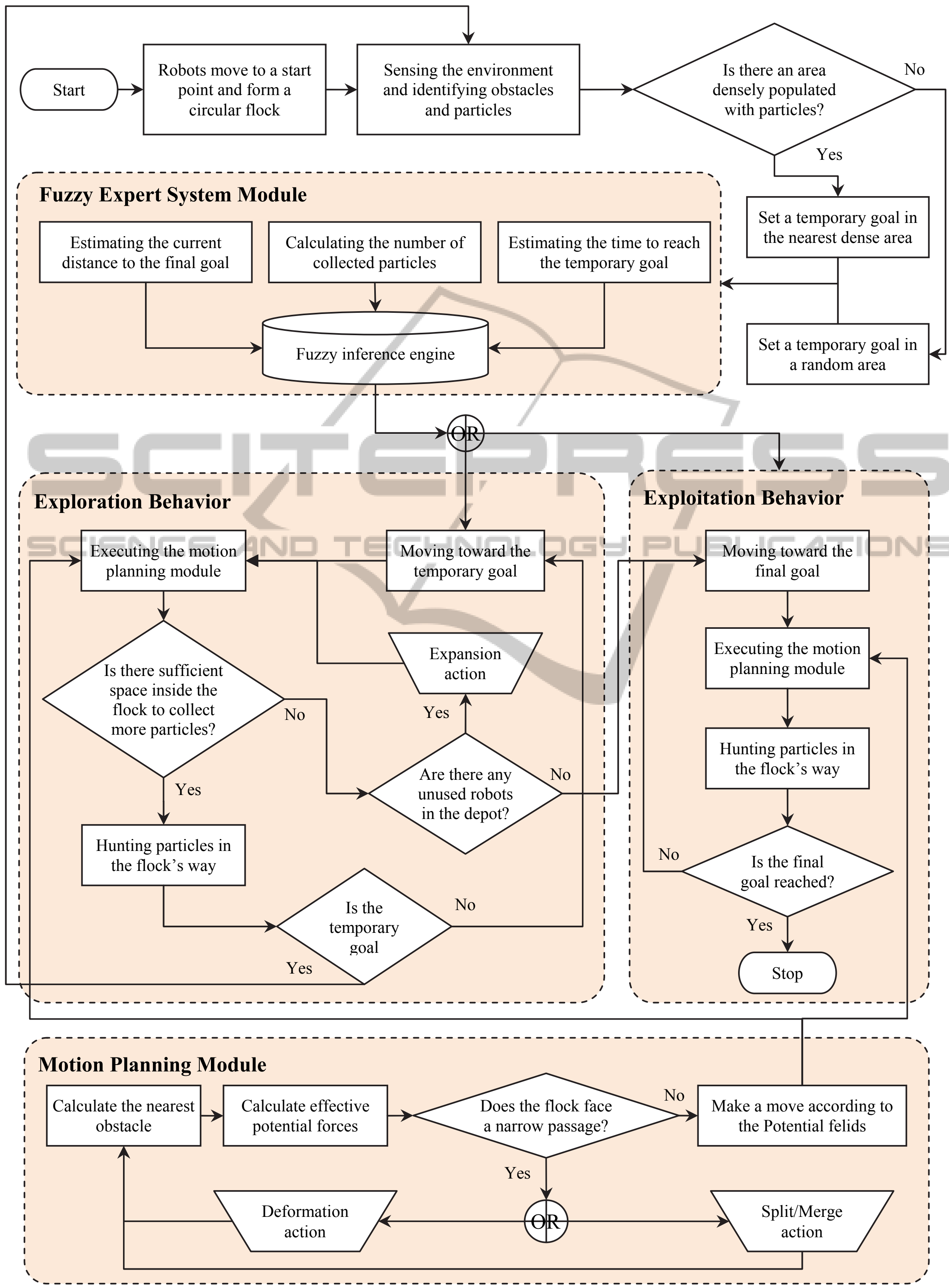

Figure 3: The proposed architecture for solving the SOCS problem.

and minimum allowable distance between the robots (D_{Rmax} and D_{Rmin}, respectively). If no objects are marked for collection, a temporary goal is randomly set in an unexplored area and the flock moves there, while caging and shepherding all collected particles.

In case that the flock is too full to hunt another particle, it invokes the Expansion action.

4.1.1 Expansion Action

As the flock gets larger, for preserving its connectivity and preventing the inner particles from escaping from it, the robots should remain in a proper distance from their neighbors. If this is not possible due to the outward pressure exerted by the inside particles, the flock needs to call for extra robots to join the flock. Adding a robot to the flock, however, takes time and cost which should be compared and balanced with the gain which will possibly be obtained by hunting more particles. Figure 4 shows a schematic view of how new robots are joining the flock after the flock's robots move outwards and form an expanded flock along a larger arc, making room for the newcomers.

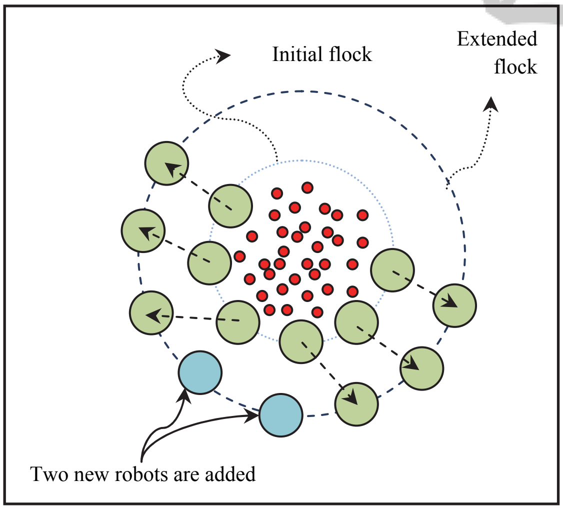

Figure 4: Expansion of the flock makes room for adding more robots, and hence accommodating more particles.

4.2 Exploitation Behavior

Unlike the Exploration mode in which the flock does not have any final destination and navigates through the workspace to collect more and more objects, in the Exploitation behavior the flock is attracted toward the one and only final goal, which might be a cage for ducks or a pier in fishing. Exploitation can be viewed from two perspectives: (1) moving straight to the goal after collecting a sufficient number of objects and approaching the time limit, and (2) intensifying the search around 'good' areas, that is, those with higher probability of having dense particles. In such a case, the flock selects the closest dense area and sets it as a temporary goal.

As it will be explained in section 5, the Fuzzy Expert System module decides the proper time for switching from the Exploration mode to the Exploitation mode based on elapsed and available times and the flock's current distance to final goal region. When the flock is in the Exploration mode but has no space for hunting more particles (and there are no robots left in the Depot for the Expansion action), then Exploitation mode must start.

For both the Exploration and Exploitation behaviors of flock is guided toward its temporary or final goal using the famous Potential Field method (Khatib, 1986), as described below.

4.3 Motion Planning Module

The Motion Planning module is responsible for guiding the flock from a point toward another point such that no collision is occurred between any robot and obstacle, and the traversed path is short, smooth, and safe. This module is activated in both Exploration and Exploitation behaviors, and is based on the well-known Artificial Potential Fields method, proposed by Khatib (1986). In this method, the robot is directed toward the goal as if it is a particle moving in a gradient vector field. Gradients can be intuitively viewed as forces acting on a positively charged point-robot which is attracted to the negatively charged goal. Obstacles also have a positive charge, which forms repulsive forces to repel the robot away from them.

Specifically, in our model, the sum of the following three forces draws a robot in the flock toward the goal while keeping it off from obstacles:

- Repulsions from the other robots,
- Repulsion from the closest detected obstacle,
- Repulsion from the particles inside the flock,
- Attraction toward the temporary or final goal.

The combination of repulsive and attractive forces will hopefully direct the robot from the start location to the goal location while avoiding obstacles. Various applications of the Potential Fields approach in coordinating a multi robot system are present in the literature for many different tasks. For example, in the collision avoidance problem, the group cohesion property can be maintained using an artificial potential field that is dependent entirely on the relative distances between the agents (Tanner et al., 2003).

As mentioned earlier, a circular arc pattern is applied for shepherding the collected particles: this works well in workspaces with relatively large free spaces. However, in cluttered environments with narrow or maze-like passages, the flock might not navigate easily, while keeping its full round shape. As a matter of fact, a number of challenges have been identified by researchers in recent works on pattern formation: Varghese and McKee (2010) showed that transformation of patterns is necessary when a robotic swarm needs to react to obstacles in the way of its motion, and presented a mathematical model for swarm pattern formation based on the foundations of the Complex Plane.

In order to properly react against the encountered obstacles and passageways, the flock can launch two effective actions: Deformation, and Split and Merge.

4.3.1 Deformation Action

Encountering narrow passages is a big challenge for flocks. Although different group formations may be used in relatively open areas, there are few shapes suitable for passing through narrow regions, which are generally shrunk along one axis and elongated along the other axis (Figure 5). Also, in during Expansion action, the flock may encounter obstacles as it expands, and so it has to deform. A reconfiguration can be achieved by repositioning all or a few agents in the pattern, which can lead to the deformation of the pattern.

Care should be taken to maintain the maximum and minimum distances between any two neighboring robots so that the flock is not disintegrated.

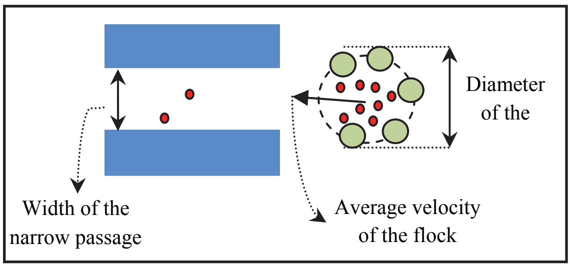

Figure 5: An example of a narrow passage: The flock's diameter is larger than the width of the passage and so cannot enter it without deformation.

4.3.2 Split and Merge Action

According to the workspace and obstacles conditions near and on the way of the flock, it may prefer to split into two or more smaller flocks to be able to detour an obstacles or pass through a narrow passage, and merge together afterwards, while trying not to lose any collected particle (Figure 6).

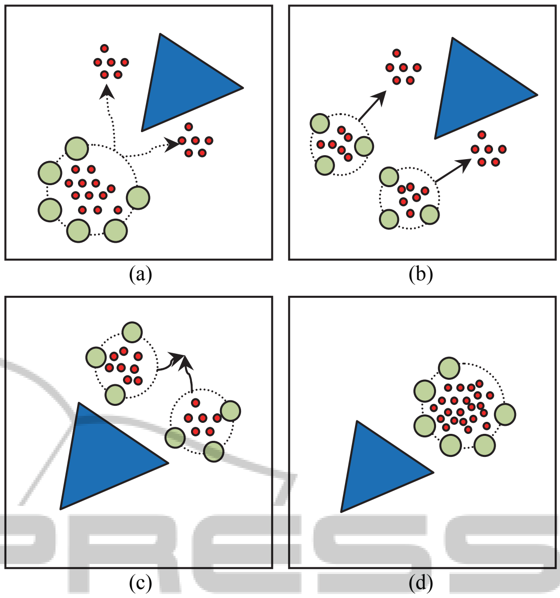

Figure 6: The flock faces two separate groups of dense particles and decides to split: (a) The flock is splitting, (b) the flock moves toward the particles in two small flocks, (c) The flock is merging, (d) The flock is reunited.

5 FUZZY EXPERT SYSTEM

The overall objective of the proposed model is to solve the SOCS problem in the online mode: that is, maximizing the total gain (i.e., covering the whole unknown workspace) while minimizing the total completion time.

In order to successfully solve this problem, the model must be able to make right decisions at the global search level, that is, when to explore, and when to exploit. This is done by implementing a Fuzzy Expert System module. On the other hand, local strategies are planned by the Motion Planning module, by deciding how to avoid an obstacle and when to undergo a deformation or a split and merge.

As it is obvious from the definition of the SOCS problem, it has two independent conflicting objectives: minimizing execution time and maximizing object collecting gain (as shown in (1)), in which T_f is the time of finishing the whole task and G_p is the gain of the particle p:

$$Z = \min\left(T_f\right) + \max\left(\sum_{\forall p} G_p\right) \quad (1)$$

In our proposed method, we assume a time interval $[T_{min}, T_{max}]$ during which the flock is allowed to execute and accomplish the collecting and shepherding tasks, and also a required minimum number of particles Q_{min} to be collected by the flock. As a result, the flock must do its best to collect as much particle as possible and reach the goal region before spending a time more than the defined upper limit. Naturally, the flock should choose areas with highest number of objects (i.e. densest area).

For deciding when to abandon the Exploration behavior the flock needs to estimate the time to reach the goal region from its current position, which can be done by calculating the distance $D(t)$ between the flock's current average position $\overline{\mathbf{X}}_{\mathrm{R}}(t)$ and the goal position $\mathbf{X}_{\mathrm{Goal}}$ via a simple straight line heuristic, as in (2):

$$D(t) = \left\| \mathbf{X}_{\mathrm{Goal}} - \overline{\mathbf{X}}_{\mathrm{R}}(t) \right\| \tag{2}$$

Given the velocity of the flock $\mathbf{V}_{\mathrm{R}}(t)$ and the remaining time ($T_{max} - t$), the flock can find out if it has enough time to further explore the workspace by visiting another temporary goal or it is time to move directly toward the final goal. Actually, the critical distance $D_C(t)$ is a distance that the robot can traverse within the remaining time:

$$D_C(t) = \overline{\mathbf{V}}_{\mathrm{R}}(t) \cdot \left(T_{\max} - t \right) \tag{3}$$

Similarly, the flock must terminate the Exploration behavior whenever it cannot collect more objects, even after utilizing all its N robots in the Depot. That is, when (3) holds, in which $C(N)$ is maximum possible capacity.

$$Q(t) \geq C(N), \tag{4}$$

Since a robot in a formation must handle additional problems such as avoiding collision with other members of the flock and relying on usually-incomplete sensory data to detect the obstacles' locations, time and distance calculations in (2) and (3) are not always exact and real. On the other hand, a flock formation should be able to successfully operate in a real-time world with lots of noisy data and must deal with the uncertainties found in such an environment. Consequently, in order to cope with these problems and possible localization and sensing errors, a fuzzy-based approach is adopted to make decisions about the flock's next behavior. This will make the model more robust and responsive toward unexpected variations in sensing or motion.

We define fuzzy membership functions for three variables: (1) time, t; (2) number of collected objects at time t, $Q(t)$; (3) direct distance to the final goal, $D(t)$; respectively as μ_t, μ_D, and μ_Q, illustrated in Figure 7. As can be seen, right parts of all these functions tend to zero; this means that for example when the time exceeds its upper limit, it is high time to exploit the search toward the goal region, or when the number of collected objects exceeds the maximum possible capacity, Exploration must end.

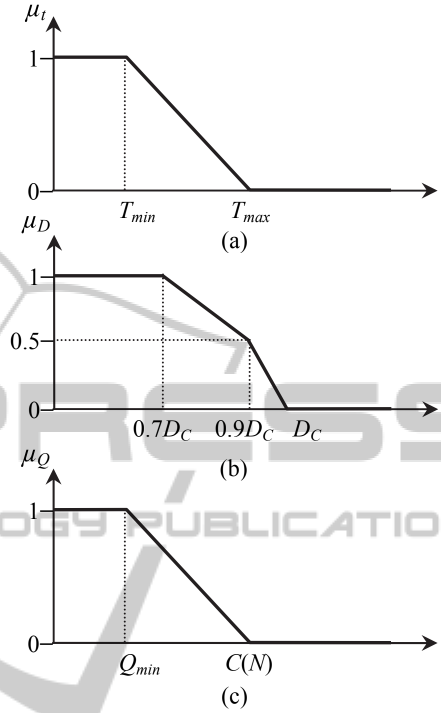

Figure 7: (a) Fuzzy membership functions for (a) Elapsed time, (b) Flock's distance to goal, (c) Quantity of particles.

Introducing fuzziness in decision making reduces the risk of making wrong decisions in the presence of incomplete perception or improperly-set parameters and thresholds. A number of fuzzy rules can be defined for integrating the above membership functions and decision variables. A typical fuzzy rule contains commonly used linguistic modifiers (like low, medium, high) and has the following structure:

RULE R_i
IF Elapsed time is Low, **AND**
Collected quantity is Low, **AND**
Distance to the final goal is Large, **AND**
Distance to the nearest temporary goal is Low
THEN *Behavior* = Exploration

We can also blend the above fuzzy membership functions into a single Fuzzy Decision criterion:

$$\begin{aligned} FD &= \mu_t(t_i) \otimes \mu_D(D_i) \otimes \mu_Q(Q_i) \\ &= \min\left\{ \mu_t(t_i), \mu_D(D_i), \mu_Q(Q_i) \right\} \end{aligned} \tag{5}$$

after which the behavior is determined by comparing the criterion's value with a threshold α, as:

$$Behavior(t) = \begin{cases} \text{Exploitation} & \text{if } FD < \alpha \\ \text{Exploration} & \text{if } FD \geq \alpha \end{cases} \quad (6)$$

6 SIMULATION

In order to assess the efficiency of the proposed model in simultaneously collecting and shepherding workspace objects we programmed it in Matlab® and implemented on a number of simulations. The performance measures were time, number of collected particles, and the total gain of particles.

Figure 8 shows a typical input to the SOCS problem. There is a Depot with 11 robots at the lower right corner, three polygonal obstacles and 64 particles scattered over the workspace. The obstacles and particles are unknown to the robots, and the final goal is located at the top center. Collecting a particle has a gain of 3 points, and each second of runtime exceeding the upper time limit has a 0.5 point penalty. The T_{max} was set to 400 seconds.

We used the PSO algorithm for simulating and coordinating the movements of particles inside the flock. The particles are dynamic and change their position and speed over time. As the robots move, they push the particles forward while preventing them from leaving the flock. At each iteration the particles try to adjust their velocities with the 'best' velocity among themselves so far, with movements and positions of their neighbors, and with the average velocity of robots (Kennedy and Eberhart, 1995). The best direction is the one that has the lowest deviation between the flock's average direction of and direction of each particle.

Figure 9 shows the experimental result: the flock moved from the Depot with 6 robots, sensed the obstacles and detected and collected 37 particles, with a gain of 111 points. The figure also reveals that the flock selected 5 temporary goals before exploiting toward the final goal region, and did not use additional robots available in the Depot. The total runtime was 412 seconds, about 3% longer than the upper time limit, and for the same reason the flock lost $(412-400) \times 0.5 = 6$ points, making the total gain equal to $111 - 6 = 105$ points.

We could not find any model in the literature to compare with our proposed model in online mode. So we considered the TSP problem as a benchmark to compare with our model in offline mode, i.e., assuming that the flock has complete information about the obstacles and objects. On the other hand, for the TSP formulation, the particles were clustered into different groups and the center of mass of each group was taken as a city (site). Also, in order to implicitly consider the presence of obstacles, distances between cities were calculated based on their geodesic distance, and the start and goal points were added to the set of cities.

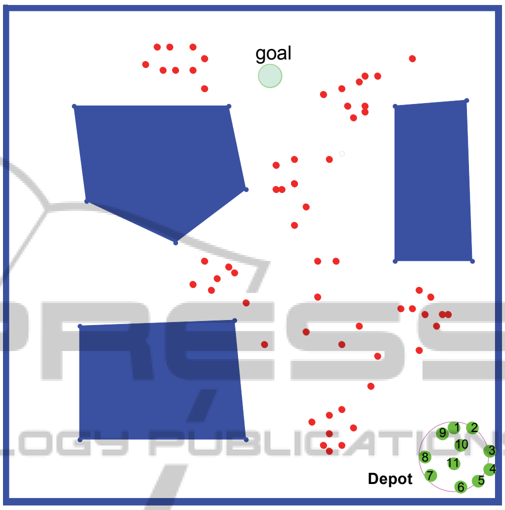

Figure 8: A sample workspace used for testing the model.

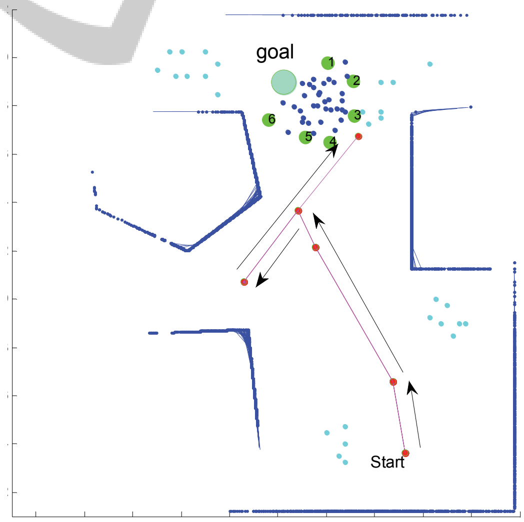

Figure 9: The traversed path and collected objects. Note the partial perception of the obstacles through range-finder sensors.

We solved a number of problems with different workspaces and various numbers of clusters by both the TSP and our model. As the results show in Table 2, the proposed model performs quite comparable to the optimal solutions obtained by solving the mTSP models.

Table 2: Comparison of the TSP and proposed models.

Model	No. of sites	Criteria		
		Path length	No. of collected particles	No. of visited sites
Proposed	11	30.69	28 out of 63	5
	10	31.89	31 out of 60	7
	7	21.54	26 out of 61	6
TSP	11	35.37	36 out of 63	7
	10	28.32	20 out of 60	5
	7	40.13	61 out of 61	7

7 CONCLUSIONS

In this paper, we have proposed a new class of problems called Simultaneous Object Collection and Shepherding (SOCS), in which a flock of robots must collect some objects and guide them to a goal region. The problem is analogous to the Traveling Salesman Problem which is NP-hard. We also incorporated online obstacle sensing and avoidance methods in the flocking behavior, and proposed a fuzzy expert system for determining the strategy of environment exploration. The model is enriched with a number of complex group actions like deformation, expansion, split and merge. A potential advantage of the proposed model is its ability in adapting its behavior to a previously-unknown environment and simultaneously performing collecting and shepherding tasks.

Future works will focus on extension of the model to dynamic environments where the obstacles or even the goal are not static and their movements are unpredictable over the time. Also we can consider the situation in which the flock has the opportunity for discharging its contents in a depot and continue collecting more objects. Also, adding the physical properties of the environment like steepness, roughness, etc. which can affect the robots' paths and velocity adjustments can be interesting.

REFERENCES

Bayazit, O., Lien, J. and M. Amato, N., 2002. Simulating Flocking Behaviors in Complex Environments. *Proc.of the Pacific Conf on Computer Graphics and Applications*.

Brett, J., 2009. *Applied Flock Theory, in Scrum Alliance*, Available at Http://Www.Agileacademy.Com.Au/ Agile/ Sites/Default/Files/Flock Theory Applied.Pdf.

Christopher, V., Joseph, F. H. and Jyh-Ming, L., 2010. Scalable and Robust Shepherding via Deformable Shapes. in *3rd Int. Conf. on Motion in Games*, Nov. 2010, Utrecht, Netherlands.

Garrell, a., Sanfeliu, a. and Moreno-Noguer, F., 2009. Discrete Time Motion Model for Guiding People in Urban Areas using Multiple Robots. *IEEE Int. Conf. on Intelligent Robots and Systems*, 486-491.

Khatib, O., 1986.Real-Time Obstacle Avoidance for Manipulators and Mobile Manipulators. International *Journal of Robotics Research* 5(1):90-98.

Kennedy, J., Eberhart, R., 1995. Particle Swarm Optimization. in *IEEE Int. Conf. on Neural Networks*. IV. 1942-1948.

Lien, J. M., Rodriguez, S., Malric, J. and Amato, N. M., 2005. Shepherding Behaviors with Multiple Shepherds. in *IEEE Int. Conf. on Robotics and Automation*.

Manh La, H., Lim, R. and Sheng, W., 2010. Hybrid System of Reinforcement Learning and Flocking Control in Multi-Robot Domain. in *2nd Annual Conference on Theoretical and Applied Computer Science,* Stillwater.

Navarro, I., Gutiérrez, a., Matía, F. and Monasterio-Huelin, F., 2008. an Approach to Flocking of Robots using Minimal Local Sensing and Common Orientation. *Proceedings of 3rd International Workshop on Hybrid Artificial Intelligent Systems*, 5271, 616-624.

Reynolds, C. W., 1987. Flocks, Herds, and Schools: a Distributed Behavioral Model. in *Computer Graphics*, 25-34.

Renzaglia, A. and Martinelli, A., 2010. Potential Field based Approach for Coordinate Exploration with a Multi-Robot Team. in *IEEE International Workshop on Safety, Security and Rescue Robotics*, 1-6.

Sharma, B., Vanualailai, J. and Chand, U., 2009. Flocking of Multi-Agents in Constrained Environments. *European J. of Pure and Applied Math.*, 2, 401-425.

Tanner, H. G., Jadbabaie, A., and Pappas, G. J., 2003. Stable Flocking of Mobile Agents, Part I: Fixed Topology. *Decision and Control*, 2010-2015.

Varghese, B., and Mckee, G. T., 2010. a Mathematical Model, Implementation and Study of a Swarm System. *Robotics and Autonomous Systems*, 58(3): 287-294.

Xiong, N., Li, Y., Park, J. H., Yang, L. T., Yang, Y. and Tao, S., 2008. Fast and Efficient Formation Flocking for a Group of Autonomous Mobile Robots. in *IEEE International Symposium on Parallel and Distributed Processing*, April 2008, Miami, FL, 1-8.