Middle East Technical University
Department of Computer Engineering

Probabilistic Aggregation Strategies in Swarm Robotic Systems

Onur Soysal and Erol Şahin

METU-CENG-TR-2005-02

April 2005

Department of Computer Engineering
Middle East Technical University
İnönü Bulvarı, 06531, Ankara
TURKEY

Technical Report

This page contains a Turkish translation of the title and the abstract of the report. The report continues on the next page.

Oğul Robotbilim Sistemlerinde, Raslantısal Toplanma Stratejileri

Onur Soysal ve Erol Şahin

Bilgisayar Mühendisliği Bölümü
Ortadoğu Teknik Üniversitesi
İnönü Bulvarı, 06531, Ankara
TÜRKİYE

Öz

Bu çalışmada, oğul robotbilim sistemlerinde, raslantısal toplanma stratejilerinin sistematik bir analizi sunulmuştur. *Nesnelerden kaçma, yaklaşma, kaçma* ve *bekleme* olarak tanımlanan dört basit davranışın bileşiminden oluşan, geniş kapsamlı bir davranış önerilmiştir. Son üç basit davranış, iki rastlantısal geçişi olan bir üç durumlu bir sonlu durum makinesi aracılığıyla birleştirilmiştir. Değişik stratejileri birleştirmek için, iki farklı ölçüt kullanılmıştır. Sistematik deneyler kullanılarak, 1) geçiş olasılıklarıyla, 2) benzetimin adım sayısıyla, 3) arenanın büyüklüğüyle, toplanma davranışının performansının, bu iki ölçüte göre, nasıl değiştiğini incelenmiştir.

Abstract

In this study, a systematic analysis of probabilistic aggregation strategies in swarm robotic systems is presented. A generic aggregation behavior is proposed as a combination of four basic behaviors: *obstacle avoidance*, *approach*, *repel*, and *wait*. The latter three basic behaviors are combined using a three-state finite state machine with two probabilistic transitions among them. Two different metrics were used to compare performance of strategies. Through systematic experiments, how the aggregation performance, as measured by these two metrics, change 1) with transition probabilities, 2) with number of simulation steps, and 3) with arena size, is studied.

1 Introduction

Aggregation is one of the fundamental behaviors of swarms in nature and is observed in organisms ranging from unicellular organisms to social insects and mammals[1]. Aggregation helps organisms in many ways such as avoiding predators, resisting hostile environmental conditions. Some of the aggregation behaviors are known to be facilitated by environmental clues; flies use light and temperature, and sow bugs use humidity for aggregation. However, other aggregations are self-organized. The aggregation of cockroaches, young penguins and fish schools don't use such clues but are rather result of emergent cooperative decision.

This study focuses on the self-organized aggregation behaviors for swarm robotic systems[2, 3]. Aggregation behavior is essential for these systems, since for most swarm robotic behaviors robots must be in some proximity of each other in many cases.

The aggregation problem, which may seem rather trivial at first look, is challenging since in most swarm robotic systems, individuals have to rely on a rather myopic and crude perception of their world. Therefore, it is difficult to obtain behaviors to form large aggregations that are beyond the sensing range of the individuals. Thus, locally optimal aggregations of robots should be transformed into a globally optimal aggregation only by local and simple rules.

2 Related Work

Entomologists study dynamics of aggregation for understanding of behavior of social insects. A study by Deneubourg *et al.*[4] investigates dynamics of aggregation with respect to changes in individual behavior. Effect of resting behavior is examined in aggregation of cockroaches and the weaver ant. This study points out a simple individual behavior such as resting can lead to collective decision making. Jeanson *et al.* [5] present a model of aggregation behavior of *Batella Germanica* larvae in homogenous conditions. Effect of individual behavior is investigated in both individual and collective level. A model of this behavior is used in robotic experiments to obtain similar behavior in a group of Alice robots (K-Team, Switzerland).

One similar robotics applications of aggregation behavior is done by Melhuish *et al.*[6]. In this study, the size of aggregates is modulated using sound signals. This study uses infrared beacons to seed clustering so the formation of clusters is not entirely self organizing.

Another approach to generate aggregation behavior, is use of evolutionary robotics methodology[7]. In [8, 9] genetic algorithms are used to evolve neural network controllers for a team of robots in aggregation task. Performance of evolved controllers are investigated for their scalability. These controllers were observed to be able to form static and dynamic clusters .

In all of these studies, the aggregation behaviors were analyzed under a rather narrow range of parameter choices. In this paper, we propose a generic aggregation behavior obtained through a combination of some simple behaviors. Simple behaviors were combined with subsumption architecture and a finite state machine. By systematically varying the parameters of the generic aggregation behavior and the environment we analyzed the aggregation performance of different aggregation strategies using two different metrics.

In the rest of the paper, we first describe the experimental framework of our study. Then we describe the generic aggregation behavior and performance metrics that have been used. Finally we present the results of the experiments.

3 Experimental Framework

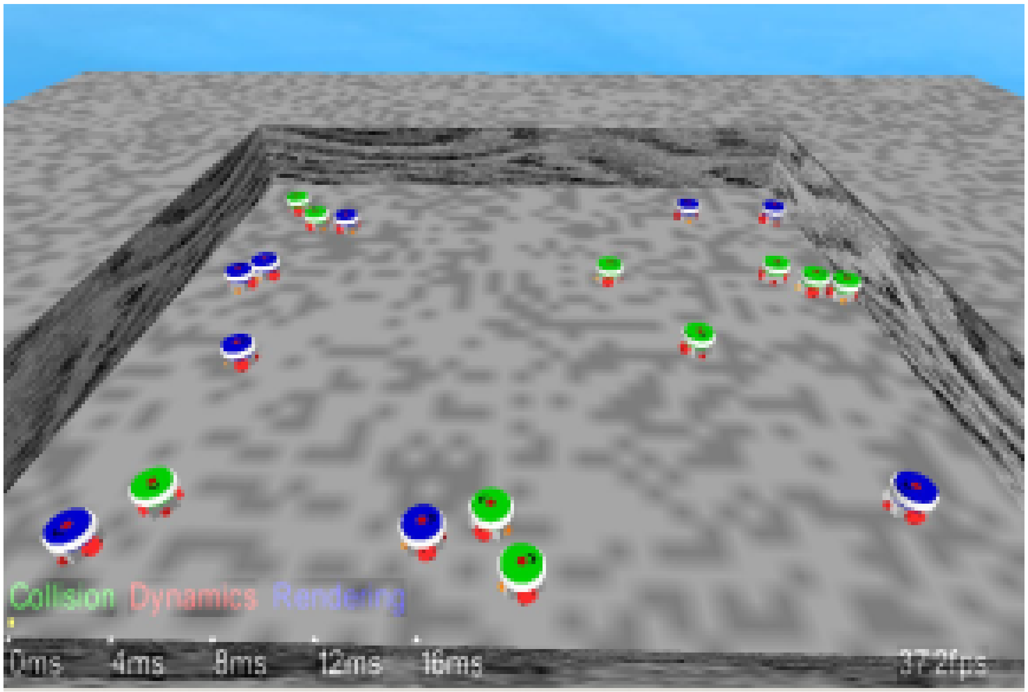


Figure 1: A screenshot of the simulator.

A port[1] of the Swarmbot3D simulator [10], a physics-based simulator developed within the Swarm-bots project, is used for the simulations. Simulation environment extends minimal s-bot model, which was used in evolution of aggregation behavior by Dorigo *et al.*. Although the Swarmbot3D environment has been verified with real robots at different levels, in this study, no such attempt is made for the extensions implemented.

The robot used in our experiments, is composed of two separately controlled wheels mounted on a circular chassis. The robot contains 4 directional sound sensors and 15 IR sensors as shown in Figure 2. The IR sensors are modeled using a sampling model described in [11]. White noise with magnitude of 5% of the range of original signal is added to this sensor's readings. This sensor is assumed to be capable of differentiating between walls and robots.

The robots have omni-directional sound emitters which are constantly kept active during this study. Sound sensors are not capable of distinguishing different sound sources. In case of multiple sound sources, the result is a sum of the effect sources. The echoes and interference are not modeled. A static white noise with magnitude of 5% of the maximum value possible for the sensor is added to sensor readings. Range of sound sensor is 100 units. Figure 3 shows the range of the sound sensor with respect to arena size.

It should be noted, however, that our simulator was neither verified against the original Swarmbot3D simulator, nor against the physical robots. Therefore, we make no claims about the portability of the controllers onto the physical robots. Yet, for the purpose of this study, we believe that the sensor and signaling models which were taken from the Swarmbot3D simulator are sufficient since our study aims to analyze aggregation behavior in swarm robotic systems in general.

[1] We ported the Swarmbot3D simulator to the Open Dynamics Engine, a free physics-based simulation library.

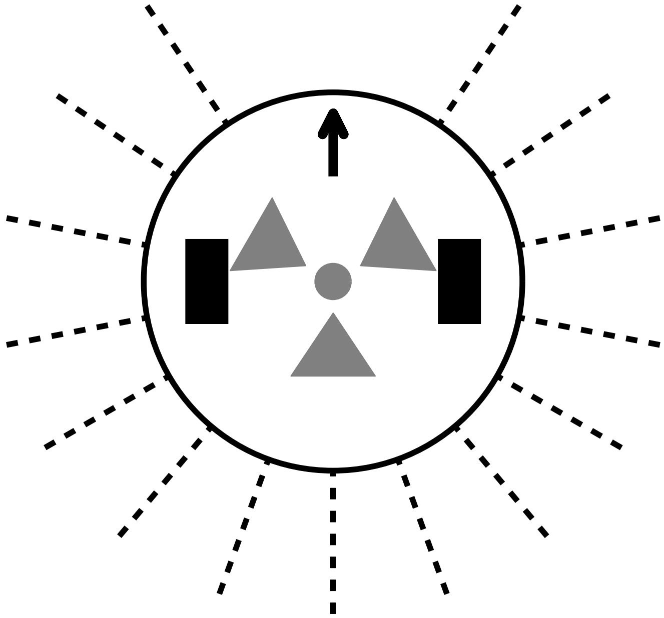


Figure 2: A schematic drawing of the robot model. The large circle (radius: 2.9 units) represents the body of the robot oriented upwards and the black rectangles denote the wheels. Gray triangles represent the directional sound sensors, (Although in [8] three microphones were shown in the model, these microphones were later extrapolated into four microphones; personal communication with V. Trianni) and the dotted lines represent the location and the approximate range of the infrared sensors. The small gray circle at the center of the body indicates the omnidirectional sound emitter. Infrared are modeled using sampling data obtained from the real robot with the addition of white noise as described in [10].

4 Aggregation Behavior

We implemented the aggregation behavior as a combination of four basic behaviors, which are arranged in a two-layer subsumption architecture as shown in Figure 4. In the lower level, an *obstacle avoidance* behavior is implemented which becomes activate when the values of infrared sensors become larger than a fixed threshold. This behavior is essential to prevent aggregations that are triggered by the existence of walls and collisions among the robots.

In the higher level, three behaviors exist: *approach*, *repel* and *wait*. The *approach* behavior uses sound sensors to estimate the relative direction of the loudest sound (a rough indication of the closest robot cluster) and drives the robot towards there. The *repel* behavior is the opposite of the *approach behavior*. It drives the robot in the opposite direction of the loudest sound. For both behaviors, when the robot senses no sound source it moves on a straight line. During the *wait* behavior, the robot stays in place.

The higher-level behaviors are arbitrated using a finite state machine (FSM) with probabilistic transitions as shown in Figure 7(a) to implement a class of aggregation behaviors. At each state, the corresponding behavior becomes active. The robot initially starts in the *approach* state. In this state, the robot approaches the largest robot cluster in its view and switches to the *wait* state when it satisfies the "robot close" condition. The "robot close" condition is signaled when a robot can be perceived using infrared sensors. During the *wait* state, the robot picks a random number uniformly within the range $[0-1]$ at each time step. If this number is larger that P_{leave} then the robot switches to the *repel* state. Otherwise, the robot stays in the *wait* state. Similarly, when the robot is in the *repel* state, with probability P_{return} the robot switches back to the *approach* state.

We believe that the FSM, shown in Figure 4, represents a generic model of reactive[2] aggregation

[2]The FSM representation is used for conceptual presentation of the behavior and to show that the

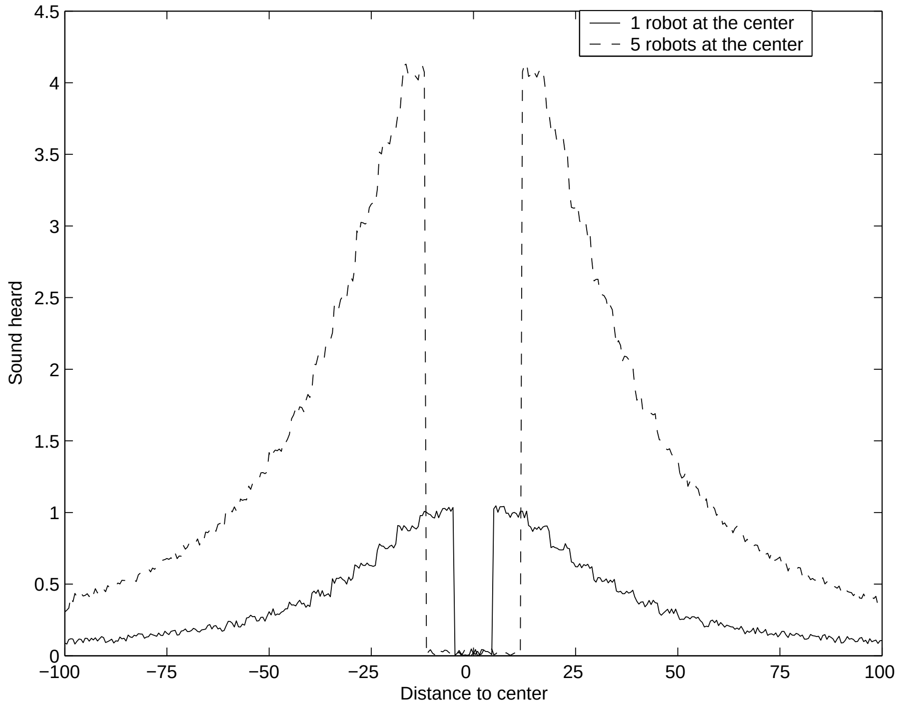


Figure 3: Sound perceived by a robot. A robot emitting sound is assumed to be placed at the center.

behaviors that are of interest to swarm robotic systems. The resulting behavior can be summarized as: *Approach the sound source. Wait within that cluster for a random time. Run away from sound sources. After random amount of time approach back to the loudest sound source.*

Note that the robot can not distinguish sound sources, causing robot to move to the middle of sound sources when two sound sources with equal intesty are at the same distance. Similarly, from the view point of the robot, a single robot that is close by will sound as loud as a large robot cluster that is further.

5 Aggregation Performance Metrics

In this study we chose two different metrics to measure performance of aggregation. First metric, *Expected Cluster Size* (ECS) uses a threshold $T_{RobotClose}$ to determine robots in the same cluster. The robots which are closer than the $T_{RobotClose}$ are considered neighbours. Let $dist(R_i, R_j)$ denote distance of i^{th} to j^{th} robot. Then neighbourhood relationship is defined as follows:

$$Neigh(R_i, R_j) = \begin{cases} 1 & ;\text{if } dist(R_i, R_j) < T_{RobotClose} \\ 0 & ;\text{o.w.} \end{cases}$$

Using this neighbourhood information, robots in the same cluster are determined using a connected component determination algorithm. Each connected component is labeled as a separate cluster.

Cluster size for a robot, $Size(R_i)$, is the number of robots in the cluster that robot belongs to. This metric calculates the average of cluster sizes for each robot in the swarm.

$$ECS = \frac{1}{n}\sum_{i=1}^{n} Size^2(R_i)$$

behaviors are implemented in a purely reactive way.

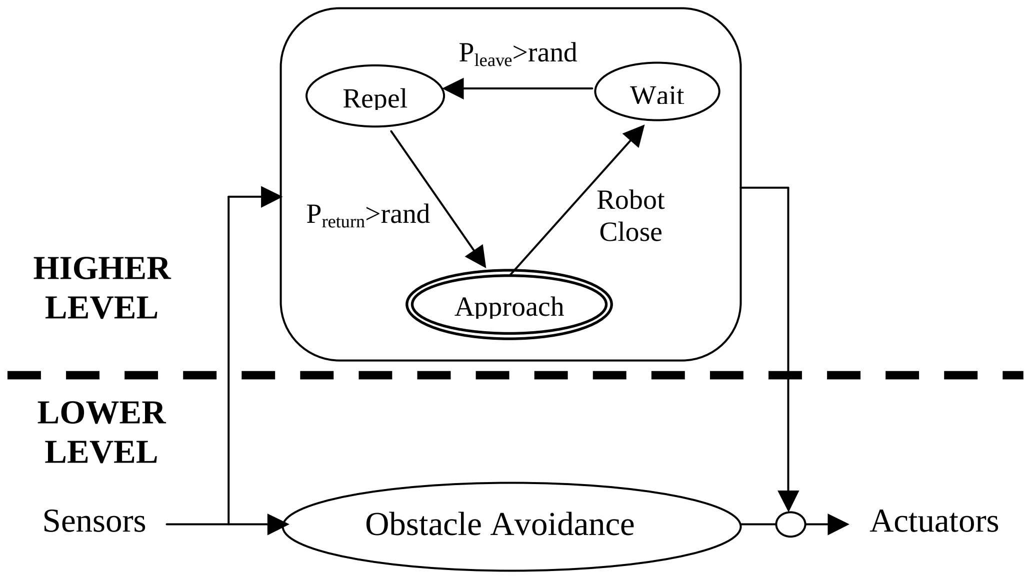


Figure 4: Controller used in experiments.

This metric ignores spatial distribution of clusters, but gives a measure for size of cluster each robot belongs to. This approach is useful for applications where robots must maintain local links with other robots in a cluster. Jeanson *et al.* [5] uses a similar metric for measuring the aggregation performance. While our metric uses average size of clusters, their work only uses the largest cluster.

The second metric, called *Total Distance* (TD), measures the total of distances between each robot pair. This metric gives more information about the spatial distribution of the swarm and clusters. This metric uses negative of distance to emphasize high metric value for better clustering. This metric is defined as follows:

$$TD = -\sum_{i=1}^{n} \sum_{j=i+1}^{n} dist(R_i, R_j)$$

This metric is useful when density of robots in an area is significant but when local communications are required this metric doesn't always correspond to locally connected groups of robots. This metric is essentially equivalent to the metric used by Trianni *et al.*[9], but our metric further penalizes outlier robots and doesn't include normalization.

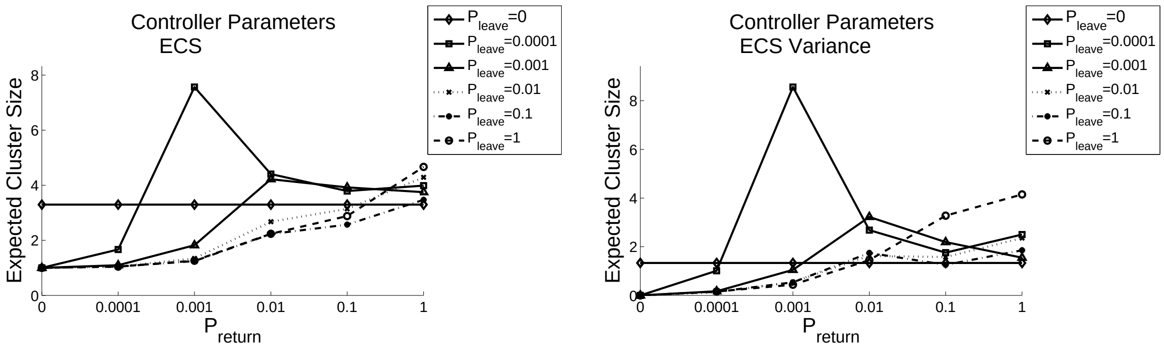
6 Experiments

Experiments are conducted in the simulator environment with 20 robots. All experiments use a square arena similar to the one shown in Figure 1. Fifty runs are made for each data point in the figures shown with different initial placements. Perfomance metrics are evaluated for the positions of robots at the end of each run. Unless otherwise stated, all runs are executed for 80000 simulation steps. For the ECS metric, $T_{RobotClose}$ is set to 10 *units* which is approximately the distance from which robots can be reliably detected.

6.1 Effect of Controller Parameters

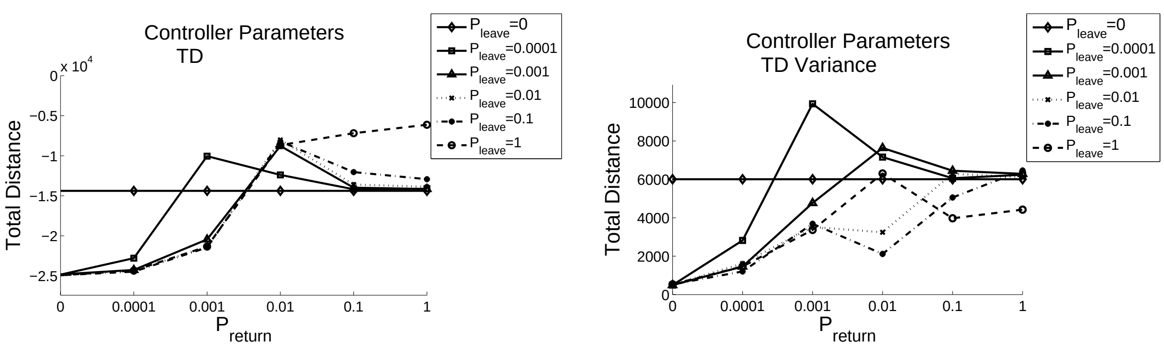
This part of the experiments aims to understand the effect of controller parameters $P_{leave}$and $P_{return}$to clustering performance. Figure 5 and 6 plot the ECS and TD metrics for different P_{leave} and P_{return} probabilities in a 200×200 arena.

Figures 5(a) and 6(a) show significant performance differences for parameters. However Figures 5(b) and 6(b) show high variance for controllers with high performance. Our studies has shown that, increasing the number of runs for each controller (up to 200) or simulation steps (up



(a) **(b)**

Figure 5: Change of metric values for varying P_{leave} and P_{return} for 200×200 arena for 50 runs. Different P_{leave} values are on different series and P_{return} changes along x axis. (a) shows mean of metric values for ECS metric, (b) shows variance of metric values for ECS metric.



(a) **(b)**

Figure 6: Change of metric values for varying P_{leave} and P_{return} for 200×200 arena for 50 runs. Different P_{leave} values are on different series and P_{return} changes along x axis. (a) shows mean of metric values for TD metric, (b) shows variance of metric values for TD metric.

to 160000 simulation steps), did not produce any significant change in variance of performance metrics.

High variance of the results can be explained by the fact that the aggregation algorithm is being observed at different stages. Although 80000 is sufficient for some of the initial conditions to form large clusters, it is not sufficient for all initial conditions. When performance grows larger, the range of different stages of aggregation also grows larger. Even though there exist asymptotic performance values for different parameters, running simulation until asymptotic performance values is infeasible in our simulation. Thus, for the rest of experiments, only mean values are reported, which gives an estimate of asymptotic performance.

By choosing different values for controller parameters, we constructed 4 strategies with different characteristics:

[**strategy 1**] When $P_{leave} \neq 0, P_{return} \neq 0$, the behavior has two competing dynamics determined by the transition probabilities. Increasing P_{leave} reduces the size of clusters and at the same time increases number of robots searching for clusters. Increasing P_{return} on the other hand reduces the distance traveled by robot while changing clusters and increases the size of clusters. This kind of controller performs best with the ECS metric and performs reasonably well with the TD metric. Figure 7(a) shows the FSM of this controller.

[**strategy 2**] When $P_{leave} = 0$, then the FSM of the resulting behavior is reduced to Figure 7(b). The behavior can be summarized as: *Move toward sound sources and when close to a robot, stay there forever.* In this strategy, when robot gets near another robot, the robot changes into *wait* state. Since the probability of switching to *repel* state is zero, the robot stays within the *wait* state forever. This behavior leads to small "frozen" aggregations. The expected cluster size is independent of P_{return} and is considered as our baseline performance.

[**strategy 3**] When $P_{leave} \neq 0, P_{return} = 0$, the FSM of the resulting behavior is reduced to Figure 7(c). The behavior can be summarized as: *Move toward sound sources initially but then run away from sound sources forever.* In this strategy, all robots eventually fall into the *repel* state and this creates no aggregations leading to an expected cluster size of 1.

[**strategy 4**] When $P_{leave} = P_{return} = 1$, the FSM of the resulting behavior is reduced to Figure 7(d). The behavior can be summarized as: *Move toward sound sources but don't stop; avoid robots like walls.* In this strategy controller never stays in states other than *approach*, since the probabilities of transition are one. This is equivalent to a fully reactive control with only *approach* and *obstacle avoidance*. This behavior creates dynamic robot aggregations. Although there is only a minor difference between this controller and the controller with $P_{leave} = 0.1$ and $P_{return} = 1$; the behavior is quite different. While in a cluster latter controller stays in wait state similar to strategy 2, only occasionally using obstacle avoidance. Whereas strategy 4 never stays in *wait* state.

Behavior of these four strategies can be seen from screenshots of the arena for varying simulation steps in Figure 8. We would like to note that for strategy 1, we used the parameters $P_{leave} = 0.0001, P_{return} = 0.001$, for which we had obtained the best aggregation performance. These parameters are kept same for the rest of experiments.

6.2 Effect of Time

Figure 9 and 10 plot the time dynamics of the four strategies. It can be seen that the ECS performance of strategy 1 is even lower than strategy 2 and 4 during the first 20000 steps. But after 40000 steps, performance of strategy 1 increases rapidly. Another interesting point is the initial increase of performance for strategy 3. All robots start with approach behavior thus they form initial clusters corresponding to increasing performance until 2500 simulation steps. After a random amount of time controlled with P_{leave} robots start moving away from sound sources. In case of strategy 3, robots never change into the *approach* behavior. Which causes the perfomance of this strategy to fall afterwards.

According to the results of the experiments with the TD metric, 4th strategy is superior. This effect can be explained by the definition of aggregation for the ECS metric. The range used for detection of neighbouring robots was 10 *units*. This threshold is found to be high enough to detect clusters in strategy 1, but is not sufficient to label clusters in strategy 4 where clusters are more sparse internally. Figure 11 shows average cluster sizes for a larger threshold (maximum distance possible for IR detection, 15.8 *units*).

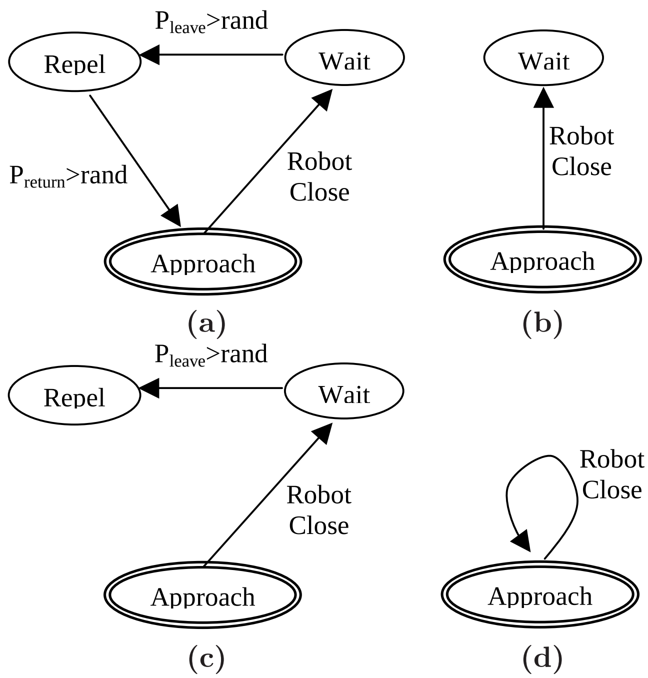


Figure 7: Finite State Machine representations of the aggregation behaviors studied are shown. (a) strategy 1, (b) strategy 2, (c) strategy 3, (d) strategy 4. See text for detailed description.

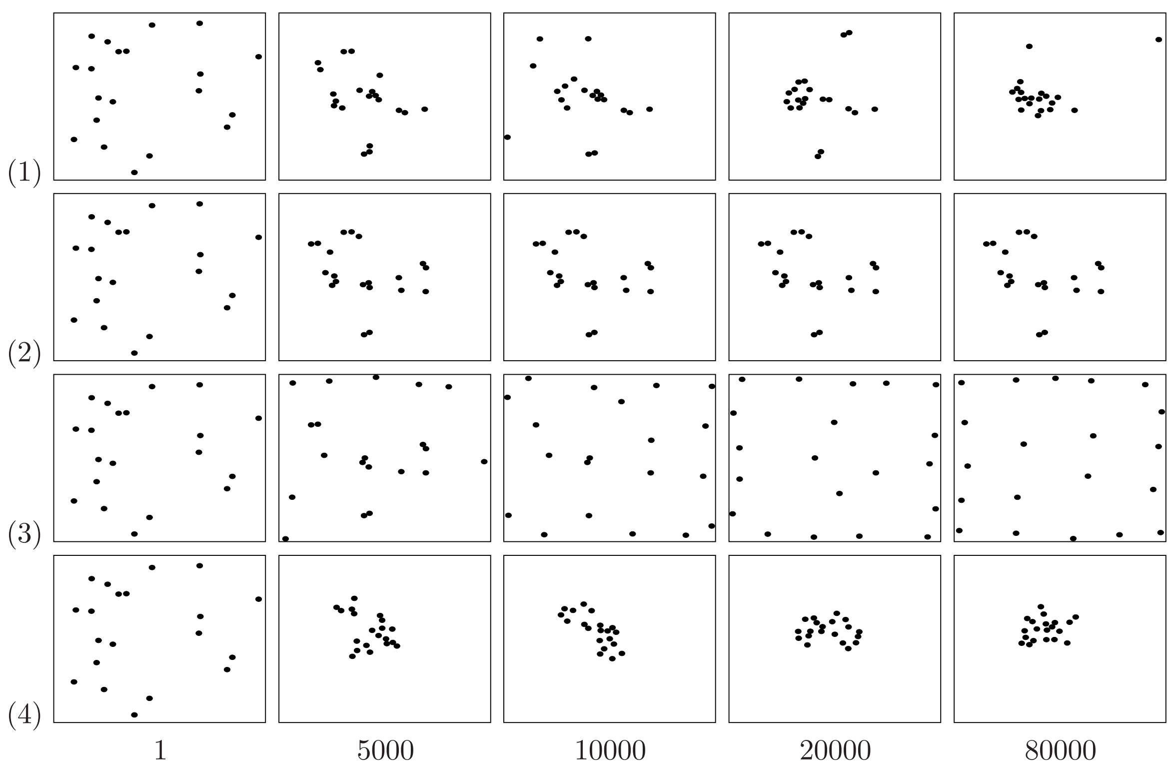


Figure 8: Positions of the robots in the arena for different time steps for four strategies.

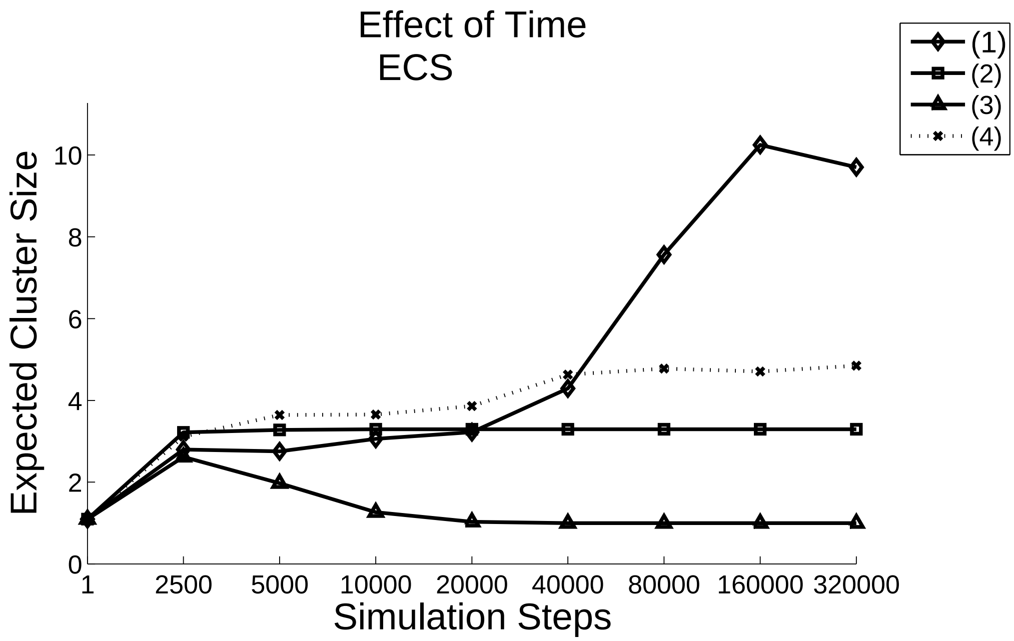


Figure 9: The effect of time on ECS metric for the four strategies. Mean values of performance of 50 runs with respect to simulation steps in a square arena of size 200×200 are shown.

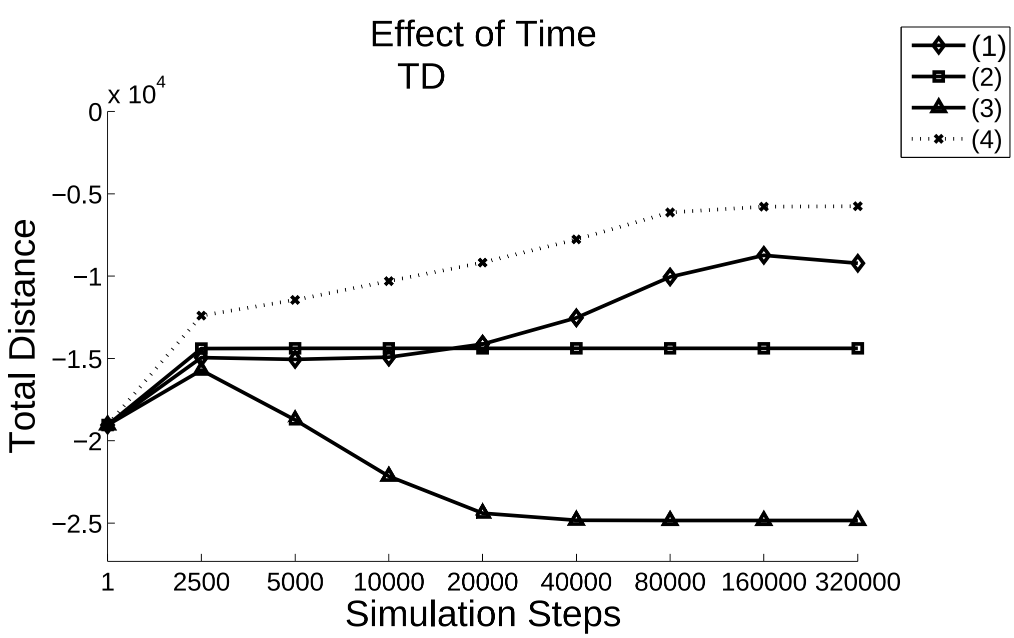


Figure 10: The effect of time on TD metric for the four strategies. Mean values of performance of 50 runs with respect to simulation steps in a square arena of size 200×200 are shown.

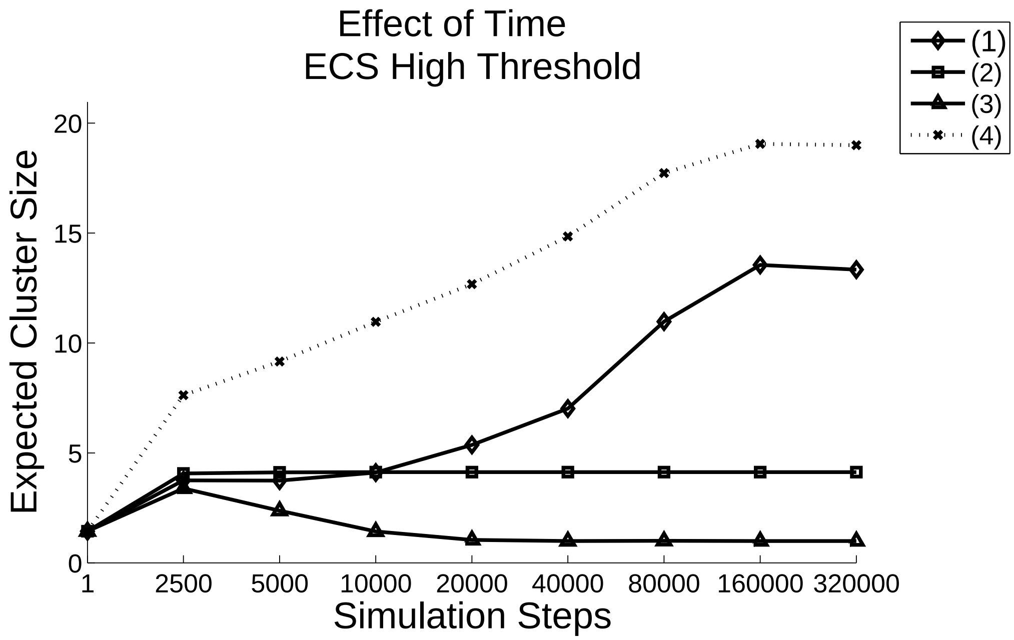


Figure 11: The effect of time on ECS for the four strategies with $T_{RobotClose}$ is equal to 15.8 $units$. Mean values of performance of 50 runs in a square arena of size 200×200 are shown.

Although the performance of strategy 4 is high, it is not very feasible for robotic systems. Apart from the risk of having large number of robots moving in close proximity, large energy consumption due to motion is problematic. In this strategy, robots never cease to move, therefore they use more energy. Figure 12 plots total distance traveled by all robots in the swarm for different strategies. In strategy 2, robots move only a little before coming to full stop and in strategy 3, robots move the largest distances among all strategies. There is also a significant difference between distances traveled by robots using strategy 1 and strategy 4.

6.3 Effect of Arena Size

Figure 13 and 14 display the performance of the four different strategies in different arena sizes. The results show that even when the optimum transition probabilities are chosen for the arena with size 200×200, the behavior can still outperform the other strategies in the smaller arena and perform reasonably well for the larger arena.

7 Conclusions

In this paper we presented a generic aggregation behavior composed of simple reactive behaviors. We analyzed parameters of this behavior and some parameters of the environment using two performance metrics.

Strategies obtained through variation of parameters provide segragation like behavior (strategy 3), static clustering (strategy 2), and dynamic clustering (strategies 1 and 4). Clustering metrics favor different dynamic clustering methods, indicating importance of an appropriate metric for the required task. Furthermore, threshold choice for the ECS metric is important since it can dramatically change the obtained results.

Although both strategies 1 and 4 form and break clusters during runs, their energy characteristics are different. Strategy 1 is more similar to cockroach behavior, where clustering is modulated by resting. This strategy allows robots to stay in closer proximity. On the other hand, strategy 4 looks more similar to schooling/flocking observed in fish and flies where agents are constantly in

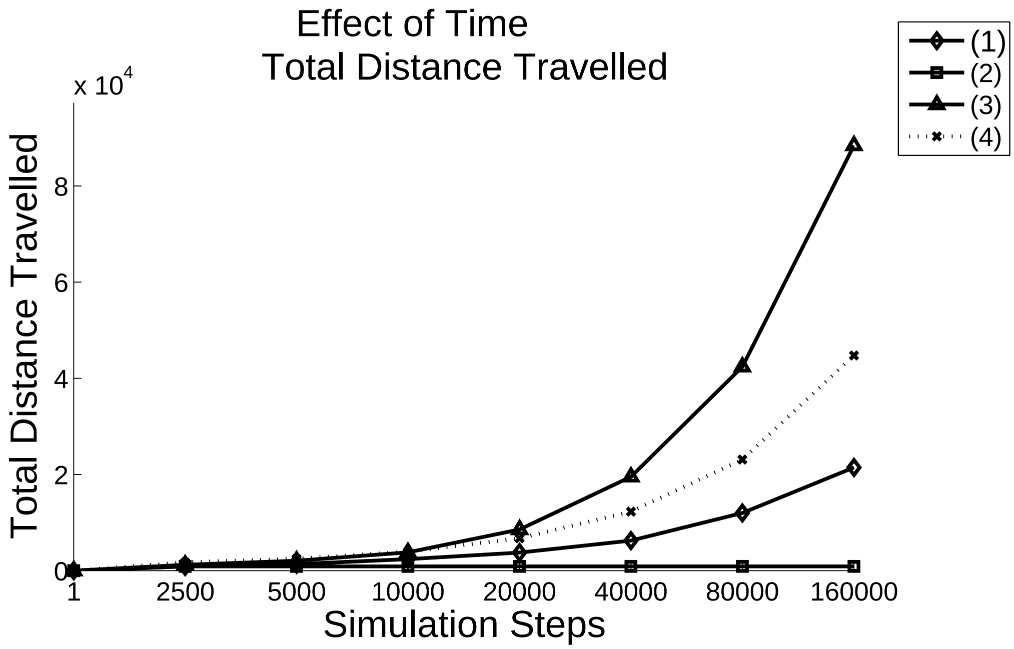


Figure 12: Total distance traveled by robots with respect to time for the four strategies. Mean values of performance for 50 runs in a square arena of size 200×200 are shown.

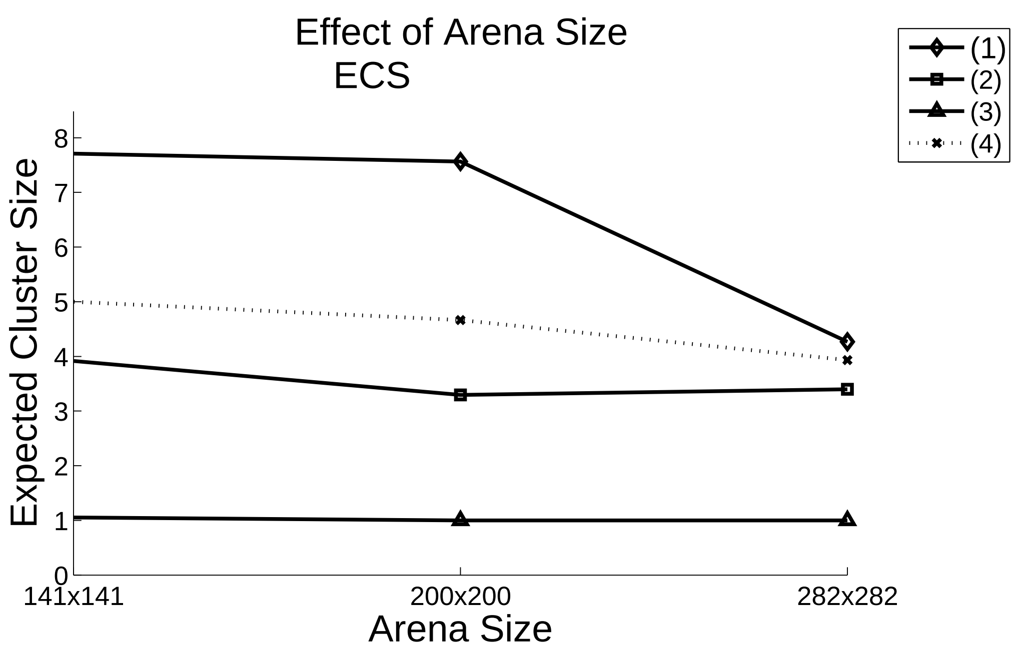


Figure 13: Effect of arena size on ECS for the four strategies. Mean values of performance of 50 runs for 80000 simulation steps are shown.

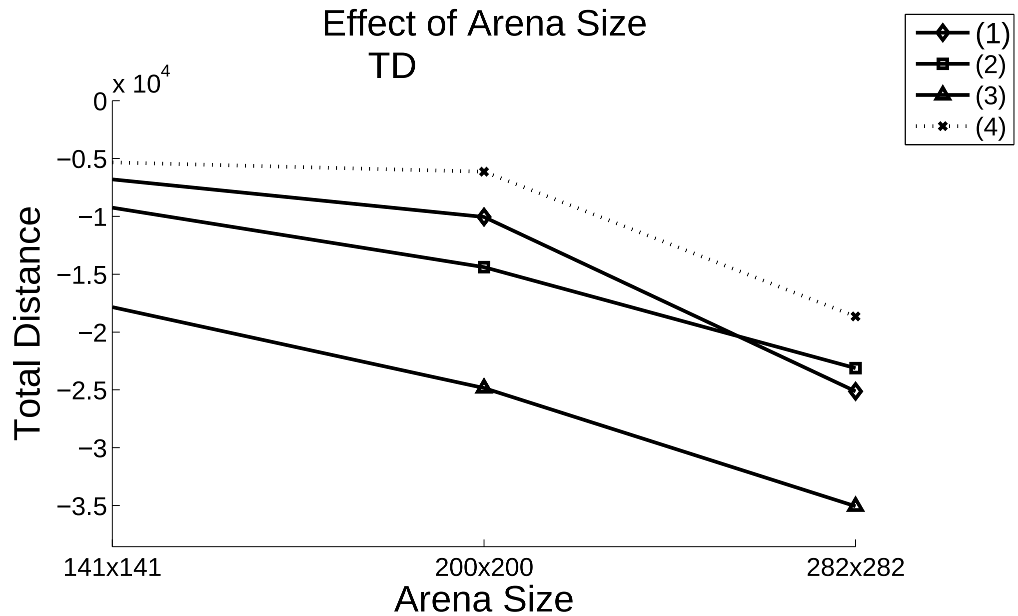


Figure 14: Effect of arena size on TD for the four strategies. Mean values of performance of 50 runs for 80000 simulation steps are shown.

motion. In this strategy average distance between robots in the same cluster is relatively larger due to obstacle avoidance.

We would also like to note that behaviors described in this study are not dependent on the specifics of the platform, thus are quite portable. Sound sensors can be replaced with any approximation of clusters around the robot and close range detection of other robots can be implemented with different methods like leds, bump sensors, etc.

Acknowledgements

This work was partially funded by the SWARM-BOTS project, a Future and Emerging Technologies project (IST-FET) of the European Community, under grant IST-2000-31010, and by the "Kontrol Edilebilir Robot Oğulları" Career Project awarded to Erol Şahin (104E066) by TÜBİTAK (Turkish Scientific and Technical Council).

The computational resources used in this work are provided by the TÜBİTAK ULAKBİM High Performance Computing Center.

References

[1] Camazine S., Deneubourg J.-L., Franks N., Sneyd J., Theraulaz G., and Bonabeau E. *Self-Organization in Biological Systems*. Princeton University Press, Princeton, NJ, 2001.

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

[3] E. Şahin and W. M. Spears, *Swarm Robotics Workshop: State-of-the-art Survey*. Erol Şahin and William Spears, editors, Lecture Notes in Computer Science 3342, Springer-Verlag, 2005, ISBN 3-540-24296-1.

[4] Deneubourg, J. L., Lioni, A., and Detrain, C.*Dynamics of aggregation and emergence of cooperation*. Biol. Bull. 202:262-267, 2002.

[5] Jeanson, R., Rivault, C., Deneubourg, J.-L., Blanco, S., Fournier, R., Jost, C. and Theraulaz, G.*Self-organised aggregation in cockroaches*. Animal Behaviour. 69, 169-180, 2005.

[6] Melhuish, C., Holland, O., and Hoddell, S.. *Convoying: using chorusing to form travelling groups of minimal agents*. Robotics and Autonomous Systems, 28(2-3):207-216, 1999.

[7] Dorigo M., Trianni V., Şahin E., Gro R., Labella T.H., Baldassarre G., Nolfi S., Deneubourg J.-L., Mondada F., Floreano D., Gambardella L.M. *Evolving Self-Organizing Behaviors for a Swarm-bot*,Autonomous Robots, 17(2-3):223-245, 2004

[8] Dorigo M., Trianni V., Şahin E., Gross R., Labella T. H., Baldassare G., Nolfi S., Mondada F., Deneubourg J-L., Mondada F., Floreano D., Gambardella L., *Evolving Self-Organization Behaviors for a Swarm-bot*, Autonomous Robots 17, 223-245, 2004.

[9] Trianni V., Gross R., Labella T. H., Şahin E., Dorigo M. *Evolving Aggregation 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, Springer Verlag, Heidelberg, Germany, pages 865-874, 2003.

[10] Mondada, F., Pettinaro, G.C., Guignard, A., Kwee, I.W., Floreano, D., Deneubourg, J.-L., Nolfi, S., Gambardella, L.M., and Dorigo, M. *Swarm-bot: A New Distributed Robotics Concept* Autonomous Robots 17, 193–221, 2004.

[11] Nolfi S., Floreano D. *Evolutionary Robotics, The Biology, Intelligence, and Technology of Self-Organizing Machines*. MIT Press: Cambridge, MA,2000.