[1]Multi-object Clustering: Patch Sorting with Simulated Minimalist Robots

Chris Melhuish[1], Matt Wilson[1], Ana Sendova-Franks[2]

[1]Intelligent Autonomous Systems (Engineering) Laboratory, Faculty of Engineering,
[2] Intelligent Computer Systems Centre, Faculty of Computer Studies and Mathematics
University of the West of England, Coldharbour Lane, Frenchay, Bristol BS16 1QY

Chris.Melhuish@uwe.ac.uk Matthew.Wilson@uwe.ac.uk Ana.Sendova-Franks@uwe.ac.uk

Abstract. This study shows that a task as complicated as patch sorting can be accomplished with a minimalist solution of four simple rules. The solution is an extension of the object clustering research of Beckers *et al.* [1994] and the object sorting research of Melhuish *et al.* [1998a]. Beckers *et al.* [1994] used a very simple mechanism and achieved puck clustering in an arena with simple robots. Melhuish *et al.* [1998a] extended this technique to sort two objects, again using a simple mechanism. This paper reports on a new mechanism, which explores the sorting of any number of different objects into separate clusters. The method works by comparing the object with which the robot has collided with the object it is carrying using a special antenna. The results in this paper are a demonstration of the success of the n-colour mechanism in simulation.

1. Introduction

Deneubourg *et al.* [1991] began the research into the idea of sorting objects using minimal rules. In their paper, *"The dynamics of collective sorting: ant-like robots and robot-like ants"*, they present a simulation which demonstrates a simple mechanism that is sufficient to generate separate clusters of two different objects. The mechanism modulates the probability of dropping objects, as a function of the local density of objects near the robot. Although it was not discussed in the paper, the possibility exists to scale this method to sort any different number of objects into separate clusters. The mechanism used, while successful in simulation has a major drawback in that the agents need to be able to sense the local densities of the different types of objects. While this information is easily made available to simulated robots, it is difficult to transfer to real robots operating with minimum sensing capability.

A simpler mechanism was used by Beckers *et al.* [1994] and was successfully implemented on a group of real robots. This mechanism was later modified and tested by Melhuish *et al.* [1998a]. It involves robots moving in straight lines in an arena. They reverse and turn though a random angle whenever their scoop is depressed. While moving in the arena the robots are able to push objects (frisbees) with which they collided in the direction of their motion. If another frisbee is collided with, and the robot is currently pushing a frisbee, the frisbee currently being pushed is deposited and the robot reverses and turns through a random angle.

After testing the above mechanism on their robots, Melhuish *et al.* [1998a] extended the mechanism by the addition of an extra rule allowing two-object segregation. The physical implementation used red and yellow frisbees and an infrared sensor able to detect which of these two colours the robots are carrying. The extra rule differentiates the action to take place when the robot is carrying a frisbee and it collides with another frisbee: red frisbees are dropped, while yellow frisbees are first pulled back a distance before being dropped. This pullback mechanism allows the yellow frisbees to be pulled away from any potential red cluster.

This paper describes research into a further extension to the mechanism above to enable the sorting of any number of different types of objects. An antenna has been added to each of the robots, which is used to sense the colour of a frisbee. If a robot is 'carrying' a frisbee and a collision takes place with another frisbee, the antenna moves and senses the colour of the frisbee in front. A comparison is then made with the frisbee being carried so that one of two possible actions can result: if the sensor comparison judges the two frisbees to be of a similar colour, then the frisbee being carried is dropped otherwise the frisbee is taken away so that it can possible be deposited elsewhere.

Proc. TIMR – Towards Intelligent Mobile Robots, Machester 2001.
Technical Report Series, Department of Computer Science, Manchester University, ISSN 1361 – 6161.
Report number UMCS-01-4-1. http://www.cs.man.ac.uk/cstechrep/titles01.html

All of the mechanisms described above could be implemented using a single robot. However, groups of robots are used to increase fault-tolerance and to decrease the time taken to complete the task. Groups of robots can also allow better comparisons to be made with social insects, allowing for the study of robot-robot interactions.

Object sorting work has been inspired by the behaviour of social insects. Patch sorting behaviour (defined below) is clearly visible in the nests of certain ant species, where *"eggs are arranged in a pile next to a pile of larvae and a further pile of cocoons or else all three categories are placed in entirely different parts of the nest"* [Deneubourg *et al.* (1991)].

The antenna plays an important role both in the robot mechanism described in this paper and in social insect behaviour. Ants have two antennae, which they use to sense differences in pheromone strength [Holldobler, Wilson (1990)]. This difference can then be used to follow a pheromone trail: a behaviour, which has also been implemented in real robots [Webb (1998)]. *Apis mellifera* honey bees can use their antennae to estimating the difference between two sources and to measure the thickness of nest walls during building [Lindauer, Martin (1963)].West-Eberhard [1969] claims that wasps can use their antennae to perceive their local environment. Downing *et al.* [1990] provide the first experimental evidence of this in the species *Polistes fuscatus*. When half a wasp's antenna is amputated, the shape and structure of the nest constructed by the wasp is significantly different from a wasp with an intact antenna. The observations of *Polistes dominulus* wasps by Karsai and Penzes [1993] show that wasps antenna the bottom of cells to distinguish between small cells and large cells. The robots described in this paper behave in a similar way to social insects. They antennae objects directly in front of them and perceive the difference between these object and the objects being carried.

As with previous research [Beckers *et al.* (1984), Melhuish *et al.* (1998a)], this work uses robots with minimalist abilities in order to conduct object sorting. The use of minimal agents aims to *"shed light on the use and usefulness of robots that have been severely constrained in terms of sensory, motor, computational, and communication abilities."* [Melhuish *et al.* (1998b)].

There are two ways in which the robots' abilities can be minimized:

- The robot hardware itself can be made with the minimum of components implying minimal sensing and communication ability. This allows the production of cheaper homogeneous robot units, which are less prone to malfunction [Johnson (1994)]. It also enables the development of rules for future use in micro or nano robots. These robots may need to *"operate in very large groups or swarms to affect the macroworld"* [Melhuish *et al.* (1998a)]; where for ease of production, only basic hardware can be used and non-communication will be important to scale to the number of robots required [Kube, Zhang (1994)].

- The robots can be labelled minimalist because they are only able to run a minimal set of rules. It is well known in biology that simple behavioural rules used by animals are more robust than complicated rules [Axelrod, Hamilton (1981)] and this is presumably also the case with multi-robot behavioural rules.

This study employs minimalist robots in the widest sense, where both the physical capabilities and the behavioural rules aim to be the minimum required to complete the task.

This paper is divided into 5 sections. Section 2 contains a description of the physical robots and environment which have been simulated to provide the results in this study. Section 3 describes the method and includes a description of the simulation and performance measures. Section 4 contains the results obtained from the simulation and Section 5 contains a discussion about the implications of, and possible extensions to, this study.

2. The Robots and the Environment

This study presents the results of a simulation, which models a real multi-robot system. The photographs and full descriptions of the real environment, robots and frisbees can be found in Melhuish *et al.* [1998a]. In brief, the robots operate in an octagonal arena containing coloured frisbees. They are 23 cm in diameter and are easily portable. In preparation to validate the results in this paper (obtained by simulation), the robots have been modified, by fitting an antenna fitted to each robot. At the end of this antenna is a cup containing an infrared sensor. During forward movement of the robot, these antennae are kept in an upright position. They come into use when a robot is carrying a frisbee and it bumps into another frisbee in front of it. Under these circumstances the antenna slowly lowers and the infrared sensors compare the colour of the frisbee in front with the colour of the frisbee being carried. If the sensor readings give similar values, then the frisbee being carried is dropped.

3. Method

3.1 The Simulation

Physical robot implementations are extremely time-consuming and usually plagued by technical constraints due to the unreliability of robot hardware [Coa *et al.* (1995)]. It was therefore decided that before validation took place on real robots, the patch sort mechanism should first be tested in simulation to determine the likelihood of success [Kube, Zhang (1992)]. The paper presents the results from these tests. Before producing the simulation, two different methods were considered: a probabilistic model and a direct simulation of the robot and the environment.

Probabilistic modelling as described by Martinoli *et al.* [1999], involves the representation of robot clustering as a sequence of probabilistic events. Cluster sizes are modified based on simple geometric considerations and robot control parameters. There are two advantages, given by Martinoli *et al.* [1999], in using this method. The first advantage is that it enables the researcher to investigate and determine which of the characteristics of the experiment are the most influential to the clustering process. The second advantage is that probabilistic models run faster than direct simulations. Despite these advantages, the probabilistic model was rejected. While it is true that sometimes the main characteristics of the simulation may be determined with this method, it is also possible that characteristics which are not responsible for the behaviour in the real robots but also lead to the same desired outcome could mistakenly be chosen as being relevant.

Instead of choosing the probabilistic model, a direct simulation was chosen for implementation. We argue that, in this case, a direct simulation allows a better feel to be gained of the behaviour of the robots. We hope that in the future it will be possible to use the simulation to develop new rules for the robots. This will be made possible by direct visual comparisons between the simulated and real systems.

The problems associated with simulated work are well known, and involve difficulties in reproducing all the properties of the robot and the environment accurately. The real world presents many opportunities for a situation to occur that would not occur in results from a simulation[2] [Brooks (1992)]. In fact, according to Krieger and Billeter [2000], the robot-robot and robot-environment interactions are almost always more complex and unpredictable in the real world than in simulation; real environments are subject to their own dynamics with a set of constraints difficult to list and therefore difficult to model. False assumptions are easily made about the properties of the system and these can lead to severely misguided models [Lambrinos *et al.* (2000)]. One of the most difficult simulation problems involves the modelling of sensors. Physical sensors deliver uncertain values, whereas in simulations sensors often deliver perfect values. Physical sensors all

[2] An extreme example of an occurrence that could not be anticipated through running a simulation is reported in Beckers *et al.* [1994]. In order to test what happened when seeds were introduced, a large plate and a saucer were placed in the arena. The experiment was stopped after three hundred minutes, because the plate was on top of a cluster containing most of the pucks, and the saucer was being moved gradually towards the plate.

perform slightly differently, because of their mechanics or electronics or their differing positions within the robot. [Nolfi *et al.* (1994)]

Although simulations often miss some of the problems associated with real robots, they can also introduce new problems that are not encountered in real robots. Actually a simulation may be more difficult to work with because in the real world noise and stochastic processes occur which are useful to help “smooth things out” [Brooks (1992)].

Webb [2000] argues that the problems involved in making comparisons between simulation and naturally occurring phenomenon are greater than the problems involved in making comparison between real robots and nature. An advantage of real robot implementation is that the researcher is forced to confront assumptions about the nature of the stimulus and the possible actions given real characteristics of the environment. Unlike a simulation a robot cannot choose an arbitrary form of input to avoid sensing problems, or have an interpreted output that skips actuator problems. Despite the problems associated with simulations, they can be useful especially if work with physical robots follows which can reinforce their results.

Written in Java and running on an 800MHz PC, the simulation provides an octagonal arena in which artificial robots can push around artificial frisbees using the same rules as the ones used by the robots in the real arena. When producing simulations, care needs to be taken to simulate only what can actually be built in real life [Kube, Zhang (1992)]. For this work a robot was built and tested first, before the simulation was written, allowing the behaviour in simulation to be compared with the behaviour of the real robot.

Time within the simulation is measured in iterations. For each iteration the expected number of moves per robot is one, with one move being the movement the equivalent to one pixel horizontally or vertically on the virtual arena. In addition, all collision with walls and frisbees are checked each iteration and frisbees are moved as appropriate. By measuring the time it takes for a robot to cross the real arena, a comparison can be made between simulation time and real time. The conversion rate works out to be approximately 10000 iterations in simulation to half an hour of robot time.

One of the most important design considerations is the introduction of noise (in the form of randomness) into the simulation. Jakobi [1998] and Jakobi *et al.* [1995] used simulations in developing controllers for robots using evolutionary effects. They found that even minimal simulations were sufficient to provide solutions capable of crossing the reality gap provided a large enough amount of random variation is included in this simulation in the right way. Besides the random turn motion included in the patch sorting mechanism, the possibility of a random ‘error’ being made has also been incorporated into the simulation. In real robots the sensors sometimes provide values that allow mistakes to be made about the colour of frisbees. This possibility has been included as a simple probability within the simulation. If a ‘mistake’ is made the robot may drops a frisbee next to a frisbee of the wrong colour.

Robot motion has also been simulated to include random variations: at each iteration robots are chosen at random for movement. The expected number of moves per iteration for each robot is one. However every robot has n (= number of robots) chances to move, each chance being a 1/n probability. This method of movement is both more realistically (real robots do not run smoothly) and it eliminates the possibility of step lock with all the robots being stuck in a fixed pattern of behaviour. As well as including this random choice of robots, a probability of a small random movement in any direction is included per iteration, again to model the unpredictable nature of real robot movements.

Figures 1a and 1b below provide an illustration of the simulation. Here six robots are participating in the sorting of six different types of objects. The colours of the frisbees have been represented by different shapes for clarity within this black and white report. Figure 1a shows the arena close to the start of the experiment with a random distribution of the frisbees. Figure 1b shows the arena after 100,000 iterations when 80% patch sorting (for definition see below) has been achieved.

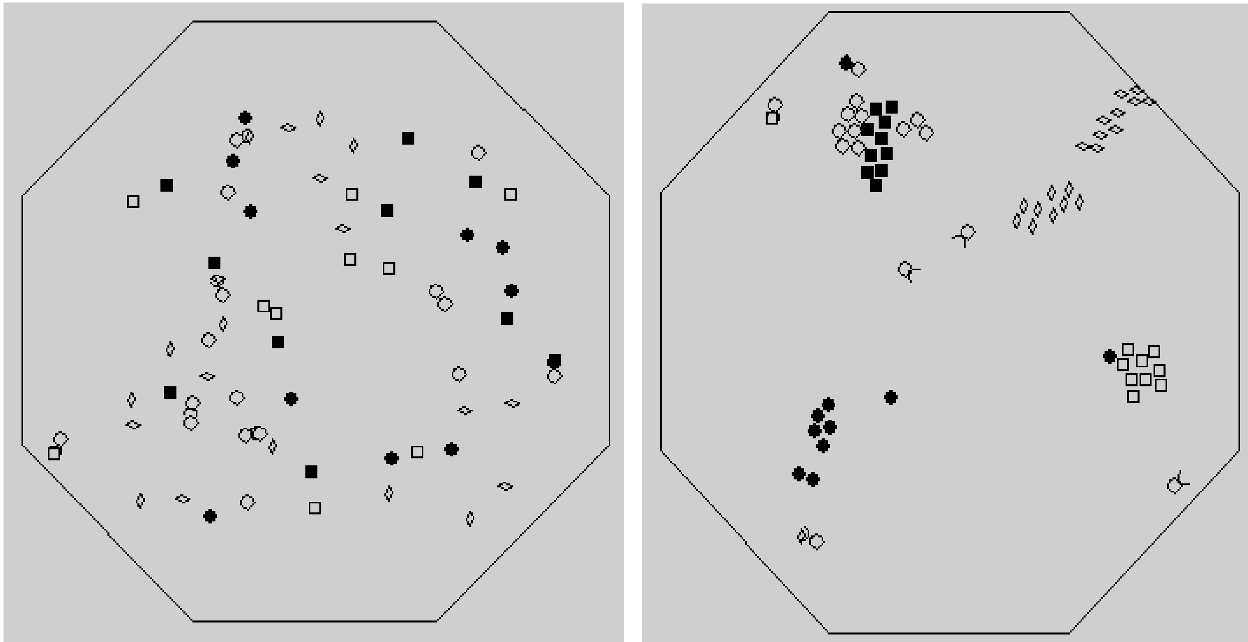

Figure 1.
a) The arena at the start of a simulated trial. *b) The arena after 100000 time steps.*

The results of the simulation were first tested using two algorithms that have already been shown to be successful on real robots: the clustering of Beckers *et al.* [1994] and the segregation of Melhuish *et al.* [1998]. For both algorithms the number of robots (six) and the number of frisbees (22 for clustering and 44 for segregating) were set to be the same as those used by Melhuish. The same criterion was used as a measure of success (i.e. 90% clustering – see below). The simulation showed successful and comparable results in the case of each algorithm. For the clustering algorithm, the mean number of iterations taken to achieve the 90% clustering was 97038, with a standard error of 12208 and for the segregation algorithm the mean time taken to reach 90% clustering for the red frisbees was 149382 with a standard error of 15469.

3.2 Definitions and Performance Measures

In order to judge the success of the patch sorting algorithm, a clear definition of patch sorting is required. Melhuish *et al.* [1998a] define different types of sorting:*"clustering"* involved *"grouping a class of objects within a continuous area that is a small fraction of the area of the available environment"*; *"segregation"* is *the "grouping of two or more classes of objects so that each occupies a continuous area of the environment which is not occupied by members of any other group"* and *"patch sorting"* is *the "grouping two or more classes of objects so that each is both clustered and segregated, and that each lies outside the boundary of the other"*. This definition of patch sorting allows two clusters to be close together, but does not allow them to overlap.

We employed two metrics in order to examine the success of the patch sorting process and allow comparisons to be made under differing conditions. During each iteration, the largest cluster size was calculated for each colour of frisbee. This cluster count was based on the definition of a cluster as given by Melhuish *et al.* [1998a]: *"a group of frisbees in which any member was within a frisbee radius of at least one other member"*. We estimate the overall progress of the clustering process by summing the largest cluster size counts for each colour of frisbee separately and giving this as a percentage of the overall total number of fribees in the arena. This progress value allows a terminating criterion for the experiment to be defined as the percentage value at which the frisbees are considered to be sorted.

The rules in Table 1 below define the patch sort mechanism used in the present paper.

*Rule 1: **If** (gripper pressed & Object ahead) **then***
Make a random turn away from the object.
*Rule 2: **If** (gripper is pressed & no Object ahead & colour in front different from colour carried) **then***
Reverse for pull-back distance
Make a random turn left or right
*Rule 3: **If** (gripper is pressed & no Object ahead & colour in front same as colour carried) **then***
Drop object and reverse small distance
Make a random turn left or right.
Rule 4: Go forward

Table 1: The rules of the patch sorting mechanism.

4. Results

The experiments conducted for this study aim to explore the success of the patch sorting algorithm. The algorithm's performance is investigated as the number of different types of objects to be sorted is increased. Six robots were used throughout the experiments. The value six was chosen so that the number of robots used was feasible for physical implementation[3]. The number of frisbees used was kept at 60, with the colours being distributed as evenly as possible- for three colours 20 frisbees of each type were used and for ten colours only six frisbees of each type were used. The chance of dropping a frisbee in 'error' was set at 1%.

4.1 Experiment 1: Time to Achieve 92% completion.

The first set of experiments explore the time the robots take to sort one to seven colours. 20 trials were conducted for each of the seven colours (140 trials in total). A value of 92% was chosen, because it allows four misplaced frisbees, the same number used by Melhuish *et al.* [1998a] for their segregation experiments. Each trial was deemed to be successful and was therefore terminated at this 92% completion rate. For 60 frisbees, a 92% completion rate corresponds to 56 out of the 60 frisbees being within the largest cluster of their particular colour. The results of these experiments are shown in Figure 2.

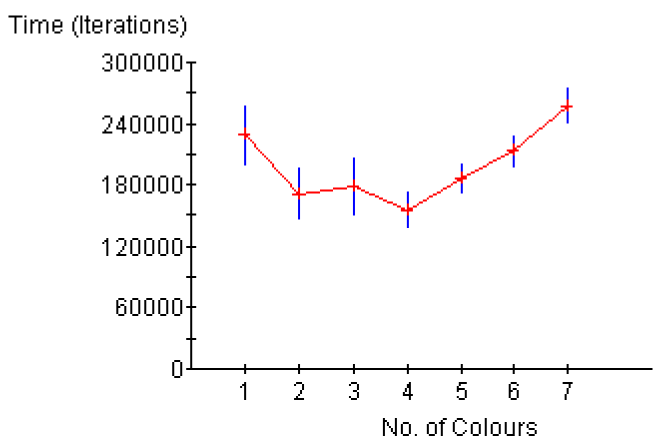

Figure 2: Mean time (+/- Standard Error) to complete 92% sorting for 1 to 7 colours.

It can be seen from the graph that the clustering of one colour is more time consuming than sorting two and three colours. This is due to two large clusters forming which are stable for a period. During this time, both clusters have frisbees removed from their perimeter by the robots. The smaller cluster is more vulnerable, because it is more likely to be disrupted by robot collisions. These frisbees can then be added to the other cluster or deposited back in the cluster from which they were removed. The probability of these frisbees being added to the larger cluster is slightly greater than the probability of them being added to the smaller cluster, due to the larger cluster's greater perimeter. This mechanism means that gradually the smaller cluster reduces in size with its frisbees being added to the larger cluster – a time consuming process. This effect explains the

[3] Currently we have ten working real robots able to perform patch sorting.

shorter completion time when sorting more than one colour, where the aim is to produce two or more separate clusters. However, as the number of colours increases beyond four, the graph indicates that the frisbees become more difficult to sort. This could be because of any of the reasons below.

As the number of frisbees increases:

1. The chance that a robot that is carrying a frisbee collides with a frisbee of the same colour is reduced because there are less frisbees of each colour within the arena.
2. Isolated individual or groups of frisbees can become encapsulated within clusters of other colours.
3. Small clusters tend to be unstable and so clusters of different colours form together. There is a tendency for these clusters to merge due to the disruption caused by collisions of the robots with the clusters.

Points 2 and 3 are illustrated in Figure 3, which shows the state of the arena after 500,000 iterations. An encapsulated frisbee and a merging cluster have been labelled.

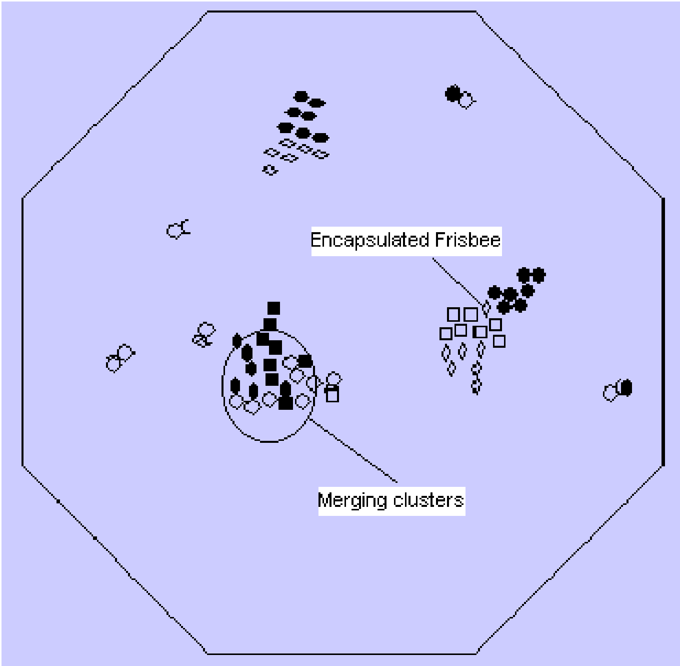

Figure 3: Eight colours being sorted by 6 robots. An encapsulated frisbee and a merging cluster are clearly visible.

Experiment 2: Average Percentage at Plateau.

Due to the mechanisms proposed in experiment 1, which inhibit the sorting as the number of colours increases, it was not possible to obtain results beyond seven colours using the 92% completion criteria. Instead a new experiment was devised. This experiment involves waiting for the clustering performance to plateau and then taking the mean percentage completion value for a period of 100,000 iterations. This is the mean percentage value around which the performance of the sorting process was able to stabilize. A total of two hundred trials took place; twenty trials for each of one to ten different colours.

The results of the last experiment were used to get a value at which we could be confident that the trial had reached the plateau level. For each colour a value two standard deviations from the mean was calculated. This gave a confidence of approximately 98% that any particular trial of an experiment had reached its plateau by this time and this was the value from which percentage performance readings started to be taken. Where results were not available from the previous experiment (i.e. 8,9 and 10 colours), readings began at 1,000,000 iterations, to err on the side of caution. Readings were taken for every iteration from this point in the trial onwards and for a further 100000 iterations. At the end of a trial, the mean of the readings found was calculated, then at the end of the twenty trials for each colour the mean and the standard deviation of these means was determined. A graph of these results can be seen in Figure 4, which clearly shows how the performance of the patch sorting degrades as the number of colours being sorted increases. The standard deviation of these values also increases as the number of colours being sorted increases. The reason for this may lie in the increased number of patterns, that can be created from an increased number of colours: each pattern having a different level of complication and a sorting difficulty. This will be investigated further in a future paper.

The number of colours used was limited to ten. Beyond ten colours, the chance of each robot carrying a frisbee of the same colour is significantly increased. If this situation occurs, it leaves no other frisbee of the same colour in the arena for the frisbees being carried to be dropped against and the success of the sorting process would then be dependant on the random dropping 'error'.

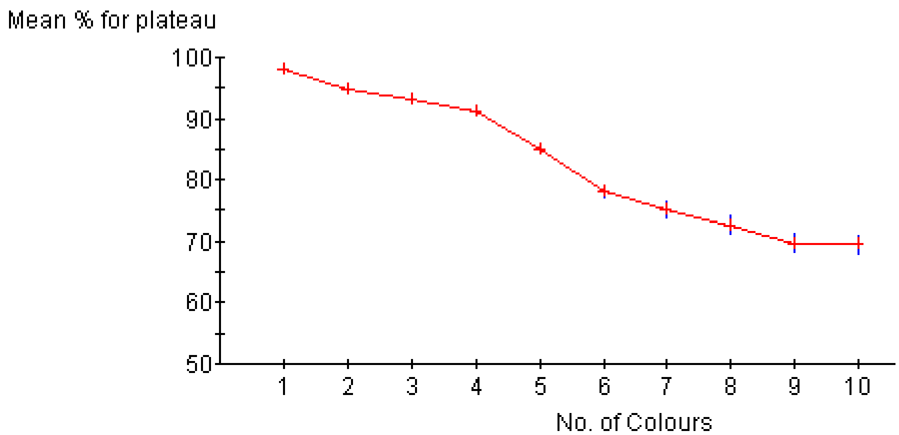

Figure 4: Mean (+/- Standard Error) of 20 trials of the mean sorting percentage (+/- Standard Error) for 100, 000 iterations after trials have reached plateau.

5. Discussion

This study extended the object clustering research of Beckers *et al.* [1994] and the object sorting research of Melhuish *et al.* [1998a]. It has shown that patch sorting can be achieved by a systems of simulated robots, with limited sensing ability and limited set of rules. To obtain the results for this study, the experiments were conducted by running a simulation. The simulation modelled an actual physical arena, which contains real robots, and different coloured frisbees to represent the different types of objects. All the experimental trials were conducted using six robots and 60 frisbees. The performance of the patch sorting mechanism was tested using 1 to 20 different colours of frisbee in the virtual arena at one time. It was found that for 1 to 7 different coloured frisbees the robots could achieve a high level of clustering for every test. However, as the number of colours was increased beyond 10, this performance fell and although the mechanism was shown to conduct some sorting behaviour when using twenty different colours, the frisbees could not be sorted with the same level of success.

There were three possible reasons given for this fall in performance. The first possible explanation is related to the probability per iteraton of a robot carrying a frisbee colliding with a frisbee of the same colour decreasing as the number of colours increases. This is because there are less frisbees of each colour within the arena. The second possible explanation is that isolated individual or groups of frisbees can become encapsulated within clusters of other colours. The final reason is that small clusters tend to be unstable and so clusters of different colours form together. There is a tendency for these clusters to merge due to the disruption caused by collisions of the robots with the clusters.

It is possible that both the problem of encapsulation and the problem of merging clusters could possibly be reduced or eliminated by altering the default settings. One idea is that the arena size affects the probability of clusters of different colours forming together. Simply increasing the size of the arena increases the space into which the frisbees can spread out. This may place different clusters at a distance far enough away from each other to prevent the clusters being nudged together by the collisions the collisions experience from the robots.

A further improvement in performance could be seen by increasing the total number of frisbees as the total number of colours increases. This may eliminate the need for clusters to form together to become stable, because each colour would have sufficient frisbees to form a stable cluster on its own. It could be that the patch sorting mechanism scales up to a large number of colours well, without a drop in performance and it is simply the circumstances of the experiments that have caused the reduction in performance demonstrated in these results. In to draw conclusions on this, experiments, which involve increasing the arena size and experiment, which involve a larger number of frisbees need to be conducted. We aim to investigate this phenomenon further in a future paper.

In conclusion, a task as complicated as patch sorting has been shown to be achievable using the simple mechanism described in this paper for up to twenty different colours. However, the quality of this sorting achieved here was higher for a smaller number of colours than for a larger number of colours. With future research we aim to discover whether it is the patch sorting mechanism or simply the arena size and the number of frisbee have caused this reduced performance.

References:

Axelrod R., Hamilton W.D. (1981), "The evolution of coorperation", Science 211, 1390-1396

Beckers R., Holland O., Deneubourg J-L. (1994), "From Local Actions to Global Tasks: Stigmergy & Collective Robots", Proceedings of the 4th International Conference on Artificial Life

Brooks R. (1992), "Artificial Life and Real Robots", Proc. 1st Eur. Conf. On Artificial Life

Cao Y.U., Fukunaga A.S., Kahng A.B., Meng F. (1995), "Cooperative Mobile Robotics: Antecedents and Directions", IEEE Int. Conf. On Intelligent Robots and Systems

Deneubourg J.L., Goss S., Franks N., Sendova-Franks A., Detrain C., Chretien L. (1991), "The Dynamics of Collective Sorting: Robot-Like Ants and Ant-Like Robots", In; Simulation of Adaptive Behavior: from Anmals to Animats, Meyer J.-A., and Wilson S. (eds), MIT Press, 356-365

Downing H. A., Jeanne R.L. (1990), "The regulation of complex behavior in the paper wasp *Polistes fuscatus* (Insecta, Hymenoptera Vespidae)", Animal Behavior 39, 105-124

Holldobler B., Wilson E.O. (1990), "The Ants", Cambridge, Massachusetts: Harvard University Press

Jakobi N. (1998), "Half baked, Ad-hoc and Noisy: Minimal Simulaitons for Evolutionary Robotics.", 4th Conference on Artificial Life.

Jakobi N., Husbands P., Harvey I. (1995), "Noise and the reality gap: The use of simulation in evolutionary robotics.", In F. Moran, A. Moreno, Merelo J.J, Chacon P., editors, Advances in Artificial Life: Proc. 3rd European Conference of Artificial Life. Springer-Verlag

Johnson P. (1994), "Cooperative Control of Autonomous Mobile Robot Collectives in payload Transportation", MSc Thesis Virginia Polytechnic USA

Karsai I., Penzes Z. (1993), Comb building in Social Wasps: Self-organization and stigmergic Script., J. Theor. Biol. 161, 505-525

Krieger M., Billeter J.-B. (2000), "The call of duty: Self-organised task allocation in a population of up to twelve mobile robots", Robotics and Autonomous Systems 30 65-84

Kube R.C., Zhang H., (1992), "Collective Robotic Intelligence", 2nd Int. Conf. On the Simulation of Adaptive Behavior

Kube C., Zhang H. (1994), Collective Robotics: From Social Insects to robots, Adaptive Behaviour 2 189-218

Lambrinos D., Moller R., Labhart T., Pfeifer R., Weigner R. (2000), "A mobile robot employing insect strategies for navigation", Robotics and Autonomous Systems, 30, 39-64

Lindauer M. & Martin H. (1963), "Uber die orientierung der biene im duftfeld", Naturwissenschaften 50(15), 509-514

Martinoli A., Ijspeert A. J., Gambardella L. M., "A Probabilistic Model for Understanding and Comparing Collective Aggregation Mechanisms", ECAL 99

Melhuish C., Holland O., Hoddell S. (1998a), "Collective Sorting and Segregation in robots with Minimal Sensing", 5th International Conference on the Simulation of Adaptive Behaviour. Zurich. From Animals to Animats. MIT Press

Melhuish C., Holland O., Hoddell S. (1998b), "Using Chorusing for the Formation of Travelling Groups of Minimal Agents", Int. Conf. On Intelligent Autonous systems, Sapporo Japan

Nolfi S., Floreano D., Miglino O., Mondada F. (1994), "How to evolve autonomous robots: Different approaches in evolutionary robotics.", In R. Brooks and P. Maes, editors, Artificial Life IV, 190-197, MIT Press/ Bradford Books

Webb B. (1998), "Robots, crickets and ants: models of neural control of chemotaxis and phonotaxis", Neural Networks II 1479-1496

Webb B. (2000), "What does Robotics offer animal behaviour?", Animal Behaviour 2000, 60, 545-558

West-Eberhard M.J. (1969), The social biology of polistine wasps, Misc. Publ. Mus. Zool. Univ. Mich. 140, 1-101