JOURNAL OF THE ROYAL SOCIETY Interface

Ant-inspired sorting by robots: the importance of initial clustering

Chris Melhuish, Ana B Sendova-Franks, Sam Scholes, Ian Horsfield and Fred Welsby

J. R. Soc. Interface 2006 **3**, 235-242
doi: 10.1098/rsif.2005.0081

JOURNAL OF THE ROYAL SOCIETY Interface

J. R. Soc. Interface (2006) 3, 235–242
doi:10.1098/rsif.2005.0081
Published online 16 September 2005

Ant-inspired sorting by robots: the importance of initial clustering

Chris Melhuish[1,*], Ana B. Sendova-Franks[1,2], Sam Scholes[1], Ian Horsfield[1] and Fred Welsby[1]

[1]*Intelligent Autonomous Systems Lab, and* [2]*School of Mathematical Sciences, University of the West of England, Frenchay Campus, Coldharbour Lane, Frenchay, Bristol BS16 1QY, UK*

For engineers the prospect of scalable collective robot systems is very appealing. Such systems typically adopt a decentralized approach in their control and coordination mechanism, which employs local sensing and action as well as limited communication. Under these constraints and informed by research on *Temnothorax* ants, two puck sorting algorithms were tested in a combination of simulation and with real robots. Both algorithms employed puck density as a cue. Only the overall local density, irrespective of puck type, was found to be required which offers the prospect for a more simple mechanism than had been previously considered. For one algorithm, this density cue was used both for picking up and dropping items and is, therefore, referred to as the 'double density' algorithm (DD). In the second algorithm, density was used as a cue only for picking up. Depositing an item was governed by the distance travelled which was specific to the type of item being carried. This was referred to as the 'single density' algorithm (SD). Unlike the DD it was found that, for the SD, the clustering of items is a necessary pre-condition for sorting. Results from ant experiments also showed that sorting is carried out in two phases: a primary clustering episode followed by a spacing phase. This strongly suggests that clustering may also be a precondition for spacing in ants.

Keywords: puck sorting; brood sorting; density cue; ant; robot

1. INTRODUCTION

For engineers the prospect of scaleable collective robot systems is very attractive. Typically, such systems adopt a decentralized approach in their control and coordination mechanism which employs local sensing and action as well as limited communication (e.g. Camazine *et al.* 2001). It is interesting for engineers and roboticists to see how far this approach can be taken—how minimalist can such systems be? With the caveat that social insects, viewed individually and particularly as a collective, are far from simple, they do offer inspiration as they represent an existence proof that decentralized systems can be adaptive, homogeneous and exhibit intelligent behaviour. Such systems can employ radically different mechanisms compared to those often used in conventional artificial intelligence.

Inspired by observations and experimental evidence on ants, the focus of this study is on object sorting in groups of both simulated and real robots. Experimental evidence shows that ants such as *Temnothorax* (formerly *Leptothorax)* sort their brood in two phases (Sendova-Franks *et al.* 2004). Thus, during colony migration, brood is carried to the new nest and placed in a cluster. When ants cease to bring in new brood, they start to space out the different brood items into characteristic 'annular' patterns in which brood of different types tend to belong to a particular annulus. For each brood type a characteristic spacing (density) is seen which is likely to be associated with different tending requirements (Franks & Sendova-Franks 1992). In the clustering phase the choice of whether to pick up or put down a brood item is probably based on the density of other items in the vicinity (Sendova-Franks *et al.* 2004). This is in stark contrast to the ants' behaviour during the 'spacing' phase, when the ants' movements of the brood can be described as a differential diffusion process. Brood items are picked up if they are packed too densely and dropped after being moved a consistent and brood specific, Euclidean distance from the brood pile (Sendova-Franks *et al.* 2004). Through this process of diffusion, brood items are sorted into order with the smallest in the centre and the largest on the outside of the pattern (Franks & Sendova-Franks 1992; Sendova-Franks & Franks 1995; Sendova-Franks *et al.* 2004).

Encouraged by our current understanding of the biology we carried out experiments with different algorithms for real and simulated robots in the hope that results from these experiments would feed back into further elucidation of the underlying biological

*Author for correspondence (chris.melhuish@uwe.ac.uk).

Received 21 April 2005
Accepted 12 August 2005

mechanisms employed by the ants. Two types of 'artificial' sorting have been described in the literature; namely, patch sorting and annular sorting (Melhuish *et al.* 1998; Wilson *et al.* 2002). Both types segregate objects of different type into 'cohesive' groups, while the geometrical configurations are different. Patch sorting creates 'clumps' of each object type whereas in annular sorting items are arranged into rings or parts of rings of each type of item with characteristic spacing of items within each ring.

Patch sorting has not only been accomplished in simulation and with real robots using density of the different object types as in the case of Deneubourg *et al.* (1991) but also by comparing the 'degree of similarity' between different types of object using simulation and real robots (Melhuish *et al.* 2001). Wilson *et al.* (2004) created an algorithm which was a continuation of Melhuish *et al.*'s (1998) two colour annular sorting which did not employ density but did use differential pull-back distances for different object types. By discriminating between three puck types, the robots could drop the first type of object on colliding with another puck, drop the second object type after pulling back a short distance and drop the third puck type after pulling back a further distance.

Wilson *et al.*'s (2004) algorithm produced structures that were close to the annular sorting observed in *Temnothorax* ant colonies (Franks & Sendova-Franks 1992). In contrast with the density based mechanism employed in this paper, the algorithm employed 'leaky integrators' to reset behaviours and effect behaviour transitions. It was found that the resulting structure became less annular when the robots attempted to sort more than three objects (Wilson 2003).

In the case of ants carrying out annular sorting of brood the experimental evidence suggests the use of a density cue (Sendova-Franks *et al.* 2004), and we confine ourselves to this approach in this study. Furthermore, experimental evidence supports the idea that ants use brood density to pick up a brood item but not to put it down and this is in contrast to other sorting algorithms where density for both picking up and putting down brood is employed (Deneubourg *et al.* 1991).

In this paper we explore two density based sorting algorithms: (1) a sorting algorithm which employs density for picking up and putting down an item—we refer to this as the 'double density' algorithm (DD); and (2) an algorithm which only uses density for picking up an object—we refer to this as the 'single density' algorithm (SD). For our experiments it was necessary for a simulated or real robot to be able to tell which item type it was dealing with as, depending on the algorithm, items were moved different distances according to their item type. In our previous experiments, a robot could tell which puck type it was picking up or putting down by the reflectivity of the surface of the puck (Melhuish *et al.* 1998). In this study we use colour rather than reflectivity to discriminate between puck types. The simple camera we used for this system is described later.

Three different 'colours' of objects were employed and these three types of item are later referred to as types 1–3 (figure 1). All pucks were the same size and therefore, we did not attempt to employ differences in

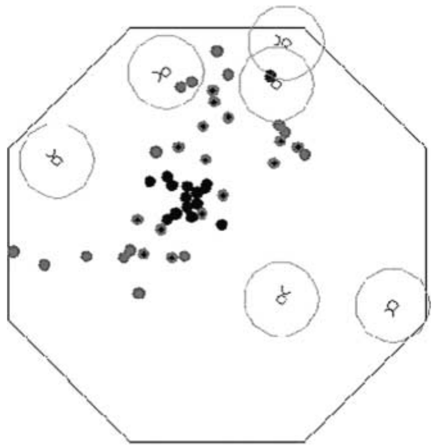

Figure 1. A frame of simulated robots sorting pucks. The type 1 pucks (black) are clustered towards the middle of the structure. Type 2 pucks (dotted grey) are further outward and type 3 (dark grey) are near the periphery of the structure. Each of the six simulated agents is shown inside its circular sensory field. The position of the scoop is also denoted by the 'horns'.

size or weight as possible discrimination cues. However, importantly, unlike Deneubourg *et al.*'s (1991) approach, our SD and DD algorithms do not rely on the density measurement of these *different* object types (for example the density of only type 2 objects) but employ a measure of the density of all objects, irrespective of type, within its sensing range—an 'omni-density' cue.

We examine the SD algorithm in simulation and with robots and the DD algorithm in simulation to explore further why the ants use a different process in each phase of sorting. After initial trials we did not use the DD algorithm in robots. The requirement to constantly monitor density while transporting objects was not a pragmatic choice, as updating density values could only be made at a relatively low rate. This issue and its possible implications to biological mechanisms and behaviour are discussed later.

Each algorithm was tested either with the pucks initially randomly distributed or with the pucks in a central cluster. In this way we were able to explore the utility of the algorithms and from our results speculate on why ants might have evolved to employ a SD-like algorithm with its concomitant constraint of requiring items to be clustered together as an initial condition.

2. METHODS

2.1. *Simulation implementation details*

The experiment with each initial condition was replicated 50 times. Forty-five pucks were distributed equally between three object types. Six simulated robots with a circular sensory area with a radius of 30 units (pixels) were used in each replicate. The density of pucks is simply the number of pucks within that sensory area. The diameter of the simulated arena was 338 units (pixels). Robot step lengths, distance to centroid of the type 1 items and dropping distances are given in these units. All experiments were run for

```
Go forward until scoop switch contacts                    (hit an obstacle)
Check forward infra red sensors for tall obstacle
If tall obstacle=True                                     (hit a wall or robot)
          move back 10 units and turn through a random angle
Else check  puck type in scoop                            (hit puck)
          If type 1 item
                    check density
                    If density is within pick up range for type 1 item
                              pick up puck in scoop
                              move until density is within drop puck range for type 1 item
                              drop puck
                    Endif
          Else If type 2 item
                    check density
                    If density is within pick up range for type 2 item
                              pick up puck in scoop
                              move until density is within drop puck range for type 2 item
                              drop puck
                    Endif
          Else If type 3 item
                    check density
                    If density is within pick up range for type 3 item
                              pick up puck in scoop
                              move until density is within drop puck range for type 3 item
                                        drop puck
                    Endif
         Endif
         back 10 units and turn through a random
Endif
```

Figure 2. The 'double density' or DD algorithm. The values of the density ranges for simulation are set out in table 1.

500 000 time-steps, the unit during which each robot moves one pixel. The simulator was based on a U-bot robot simulator (Wilson 2003; Wilson *et al.* 2004) and was further modified to reflect the behaviour observed in the ants (Scholes 2005; Scholes *et al.* 2004). The code was written using an object-oriented Java platform where all simulated robots were identical but treated as individual entities. In the simulation, agents could push coloured pucks, 'captured' in a front-mounted scoop, around a two-dimensional arena, the area of which could be specified by the experimenter (figure 1). Each simulated robot also had forward-facing proximity detection sensors mimicking the infra-red sensors of the real robots.

Simulated robots and pucks could travel anywhere inside the arena as long as the space was not already occupied by another agent or a puck. If an agent partially hit a puck, the puck would either be knocked into the simulated robot's puck scoop or knocked aside with equal probability. If pucks were pushed together, they would move in accordance with the rules of conservation and momentum associated with 'billiard balls'. This involved a high coefficient of friction as the pucks represented flat disks rather than spherical balls. If a robot collided with another simulated robot or a wall, it would turn through a random angle and move away.

Unhindered simulated robots always travelled in straight lines. After each iteration of the program, the simulated robots moved one pixel. This constituted a 'robot step'. Each simulated robot could calculate the density of pucks within a circular sensory field (figure 1) and choose to pick up (as in the case of SD and DD algorithms) or put down (for the DD algorithm only) a puck that it collided with based on this information. When the simulated robots had moved 500 000 steps, the program would terminate and calculate the distance to the centroid of the type 1 items.

2.2. Density algorithms

The first algorithm was based on the observations of ant behaviour in the clustering phase (i.e. the first phase of sorting) whereby the experimental biological results are consistent with using density for picking up and depositing brood (Sendova-Franks *et al.* 2004). Under this DD algorithm, an agent will pick up an item when it is within a specified density range and move it until it is inside a second density range before dropping it (density two), without employing any 'distance travelled' information. For type 1 items a low density is required to initiate pick up since a high density would imply that this type of item is already well 'packed'. A suitably higher density is chosen to initiate deposition since type 1 should only be added to at least partially packed structures. Type 3 items, which are required to be at the outside of the sorted structure, are picked up if they are near another object and put down when they are the only puck in the sensory range. Type 2 objects are 'intermediate' and therefore picked up when either the density is too low or too high and deposited when the density is within a range where the density is not too low or too high. The DD algorithm for spacing was

Table 1. Threshold densities for picking up and dropping pucks for the simulated robots using the DD algorithm.

spacing	type 1	type 2	type 3
pick up density	$D<7$	$D<7$ or $D>13$	$D>1$
drop density	$D>13$	$D>6$ or $D<14$	$D<1$

implemented using the algorithm shown in figure 2 (table 1).

The second algorithm was based on the observations of ant behaviour during spacing which constitutes the second phase of sorting, whereby the biological results are consistent with an algorithm which employs dropping an item after moving it a distance based on the type of item and a density cue for pick up (Sendova-Franks *et al.* 2004). This SD algorithm assumed that an agent will pick up an object if the density of the objects around it is not within a specified threshold (the only density cue) and used the same principles for picking up an item as the DD algorithm described above. Items were dropped after first moving them a pre-specified distance. The exact distance the agent moves before releasing the object is dependent on the type of object being carried. The relative magnitudes of the distances are based on those used by the ants (Sendova-Franks *et al.* 2004), but have been tuned to fit the area of the arena. The SD algorithm for spacing was implemented using the algorithm shown in figure 3.

2.3. Robot implementation details

The experiment with real robots comprised of five minimalist, autonomous U-bots, programmed using C. Each robot was fitted with a 'CMUcam' camera system for measuring puck density (figure 4). The use of the camera system highlights an important difference between the ants and the robots in that the ants appear to use antennae to sense the presence of brood items, allowing them to sort in total darkness. The real robots have no antennae to sense the pucks, although these could have been constructed but would have required more time and resources. Instead, the short range sensing capabilities are replicated by the fixed camera which can discriminate the colour of 256 squares in a small area (50 cm×50 cm) in front of the robot. However, the camera system does not use any high level image processing but simply works using colour percentages in the visual field. This retains the local sensing indicative of agents completing a task in a self-organized manner. Using this simple camera system, the density component of the ants' behavioural algorithm can be incorporated in the robot behavioural algorithm.

The method the real robots used to calculate the density of the pucks from the CMUcam image was to divide the camera's field of vision into a 16×16 grid array. Each of the 256 images was assigned a value dependent on the mean RGB value of the image and the absolute deviation of the colour found in that region. The values assigned using simple chrominance based rules were red, green, blue and floor, that is, the item type is recognized by its colour. The colour of the puck (or floor) held in the robots scoop was associated with a certain region of the 16×16 grid. If there was no obstacle but there was a puck present in the robot's scoop a grid image was obtained from the camera. The colour of the puck in the scoop of the robot determined a pull back action and the distance (table 3). Using the same principles for picking up and depositing pucks as described in simulations, densities of pucks were calculated in terms of the number of coloured (i.e. non-floor) squares. Type 1 items were left if a robot had more than 65 coloured squares in its camera array. If there were less than 65 squares, the puck would be pulled back 3 cm. Type 2 items were left if there were between 30 and 65 coloured squares in a robot's camera array but were otherwise pulled back 30 cm. Type 3 items were left if there were less than 30 coloured squares in a robot's camera array but otherwise pulled back 100 cm. On completing the whole of the distance, a robot would drop the puck. A robot would then proceed forward once again. It should be noted that if an obstacle was encountered during a pull back routine, a robot would make a random turn to avoid the obstacle and complete the set pull back distance. Each experiment was replicated three times and was recorded using a time lapse camera set to capture one frame at 5 min intervals for a total duration of 2 h.

2.4. Experiment details

In simulation we carried out two sets of experiments: one with the SD algorithm and one with the DD algorithm. Both algorithms were tested for performance in the spacing phase of sorting. We used two initial conditions: (1) pucks distributed randomly over the whole area of the arena—henceforth referred to as randomly distributed pucks and (2) pucks distributed randomly within a packed cluster in the centre of the arena—henceforth referred to as clustered pucks. The SD algorithm was also validated with real robots under both initial conditions.

After each algorithm had run for a pre-specified length of time, we logged the position of each puck in the final structure in terms of the x- and y-coordinates of its centre. Then we calculated the distance between the centre of each puck and the centroid of the structure, defined as the mean x- and y-coordinates of all type 1 items. For each experimental replicate the median distances to the centroid, for each puck type and under each starting condition, were analysed using a two-way ANOVA implemented with a General Linear Model (GLM) in the statistical package Minitab (http://www.minitab.com).

For robots, as in the case of the simulation, trials were also conducted for two categories of initial distribution of 45 pucks (15 of each type). In the first case with clustered and the second with randomly distributed pucks. The pucks consisted of three puck types each with 15 pucks and the experiment with this initial condition was replicated three times.

```
Go forward until scoop switch contacts                (hit an obstacle)
Check forward infra red sensors for tall obstacle
If tall obstacle=True,                                (hit a wall or robot)
      back 10 units and turn through a random angle
Else check colour of puck in scoop                    (hit puck)
      If type 1 item
            check density
            If density is within pick up range for type 1 item
                  pick up item in scoop
                  move 20 units and drop the puck
            Endif
      Else If type 2 item
            check density
            If density is within pick up range for type 2 item
                  pick up item in scoop
                  move 40 units and drop puck
            Endif
      Else If type 3 item
            check density
            If density is within pick up range for type 3 item
                  pick up puck in scoop
                  move 80 units and drop puck

            Endif
     Endif
      back 10 units and turn through a random angle
Endif
```

Figure 3. The 'single density' or SD algorithm. The values of the density ranges for simulation and robots are set out in tables 2 and 3, respectively.

Table 2. Density ranges for picking up items of each puck type in the SD algorithm for simulated robots. (The distances which each type of item are moved before they are dropped are shown. All distances are in 'arena units'. The diameter of the arena is 338 units (pixels). All densities refer to the number of pucks or part pucks within the sensory range.)

parameter	type 1	type 2	type 3
pick up density	$D<7$	$D<7$ or $D>13$	$D>1$
drop distances	20	40	80

Figure 4. One of the U-bots with the camera system for the measurement of puck density.

3. RESULTS

3.1. The SD algorithm

3.1.1. Simulation results. Effect of initial clustered puck distribution in simulation. The simulated robots using the SD algorithm created a well clustered group of type 1 pucks with some outliers (figure 5*a*). The type 3 items were furthest toward the periphery and although the type 2 items were spread at a similar distance from the centroid, they had a significant proportion of their number between the central cluster of type 1 items and the most peripheral type 3 items. This shows that the robots were able to sort the pucks successfully when they started as a cluster.

Effect of initial randomly distributed pucks in simulation. With the pucks randomly distributed at the start of each trial, simulated robots using the SD algorithm were unable to sort the pucks effectively within 500 000 robot movements (figure 5*b*). The type 1 items were still randomly distributed and the type 2 items were on average further from the centroid than the type 3 items.

Comparison between the two starting conditions. There is an effect of starting condition on the formation of the final structure (GLM, $F_{1,4494}=222.82$, $p<0.0001$). There is an effect of item type ($F_{2,4494}=1036.37$, $p<0.0001$) and interaction between the two factors ($F_{2,4494}=161.54$, $p<0.0001$). When the pucks start clustered, the type 1 items end up more clustered than when the pucks start randomly (Tukey test, $T=21.59$, $p<0.0001$). The type 2 items in the clustered start condition finish separated from, but closer to, the centre than the type 2 items in the random start

Table 3. Thresholds for picking up and dropping distances for the robots using the SD algorithm. (The density values refer to the number of squares containing colour in a 256 square grid, picked up by a camera.)

spacing	type 1	type 2	type 3
pick up density	$D<65$	$D<30$ or $D>65$	$D>30$
drop distances	3 cm	30 cm	100 cm

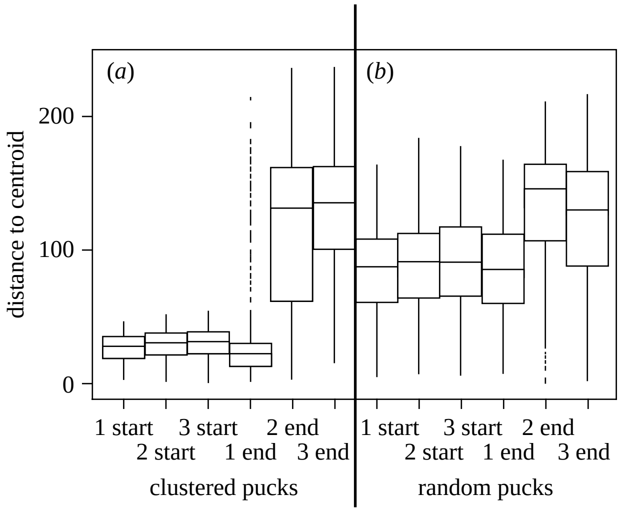

Figure 5. Comparison of distance to centroid at the start and after 500 000 time steps when the pucks start (*a*) clustered and (*b*) randomly. Each boxplot represents the data for all replicates under each condition. The distance to centroid is given in arena units where the arena is 338 units in diameter. The box in each boxplot encompasses the middle 50% of the distribution between the upper and lower quartile. The line across the box is the median. The 'whiskers' are drawn to the nearest value within 1.5 times the interquartile range. The figure is arranged showing the starting displacements (1 start, 2 start and 3 start) followed by the final displacements for each of the experiments (1 end, 2 end and 3 end).

condition (Tukey test, $T=8.08$, $p<0.0001$). The difference between the two conditions is much less pronounced than with the type 1 items. More type 3 items finish on the periphery of the structure when the pucks begin clustered than when the pucks begin randomly distributed (Tukey test, $T=-3.81$, $p=0.0019$). This suggests that the algorithm will only work if the pucks start in a cluster. This is confirmed by comparing the average final structure created using the SD algorithm when the pucks start clustered and when the pucks start randomly spread over the arena surface.

3.1.2. Robot results. Effect of initial clustered pucks distribution with robots. The robots successfully sorted the pucks. The type 1 items were well clustered with the type 2 items further from the centroid and the type 3 items around the periphery of the structure (figure 6*a*).

Effect of initial randomly distributed pucks with robots. The real robots were unable to sort the pucks effectively after 2 h of sorting. The type 1 items were not clustered and ended up pushed further to the

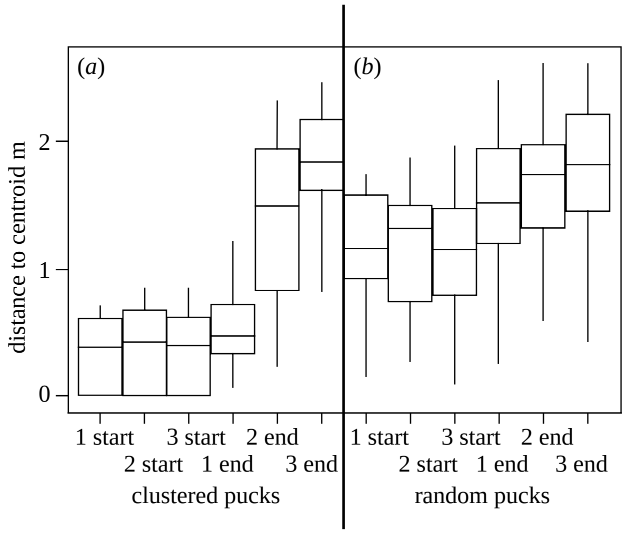

Figure 6. Comparison of the change in distance of each puck type from the centroid when the real robots were sorting using the SD algorithm when the pucks started (*a*) clustered and (*b*) randomly distributed in the arena. The robots were stopped after 2 h.

periphery of the arena than when they started (figure 6*b*). The type 2 and type 3 items were also left around the periphery of the arena and were not distributed differently to one another.

Comparison between the two starting conditions. As with the simulation, there is an effect of starting condition on the formation of the final structure (GLM, $F_{1,257}=34.11$, $p<0.0001$). There is also an effect of object type ($F_{2,257}=33.89$, $p<0.0001$) and interaction between the two factors ($F_{2,257}=15.11$, $p<0.0001$). When the pucks start clustered, the type 1 items end up more clustered than when the pucks start randomly (Tukey test, $T=7.766$, $p<0.0001$). The type 2 items do not occupy significantly different positions in the clustered start condition than in the random start condition (Tukey test, $T=2.114$, $p=0.2799$). Likewise, the type 3 items do not occupy significantly different positions in the clustered start condition than in the random start condition (Tukey test, $T=0.276$, $p=0.999$). The similarity between the type 2 and type 3 pucks in the two conditions is most likely due to the pucks being pushed against the arena wall. The robots are not sorting under the initial conditions with a random distribution of pucks. This is shown by comparing the distances of each puck type from the centroid in the condition where the pucks started randomly distributed. At the end of the experiment, the type 1 pucks are not distributed differently to the type 2 pucks (Tukey test, $T=0.9726$, $p=0.9267$) or the type 3 pucks (Tukey test, $T=2.0072$, $p=0.3381$). There is also no difference in the distribution of the type 2 and the type 3 pucks (Tukey test, $T=0.117$, $p=0.9003$).

3.2. The DD algorithm

3.2.1. Simulation results. Effect of initial clustered distribution in simulation. In simulation, robots using the DD algorithm created 'near-perfect' distances to

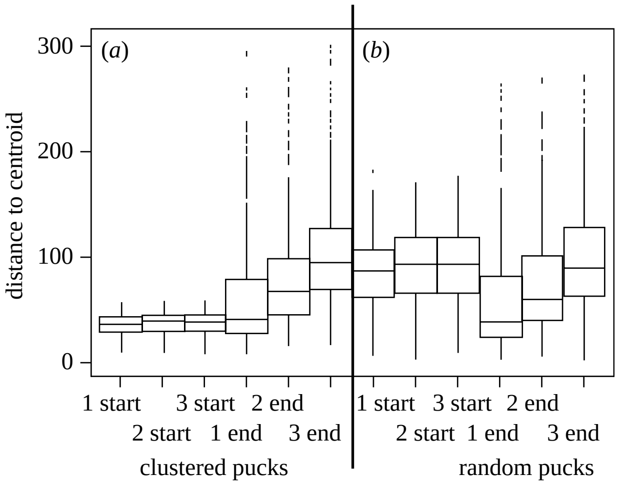

Figure 7. Comparison between the distance to centroid at the start and after 500 000 time steps (end) using the DD algorithm on (*a*) clustered pucks and (*b*) randomly distributed pucks.

the centroid of all puck types (figure 7*a*). There are outliers but the type 1 items are clustered and surrounded by a band of type 2 items, ringed by a halo of type 3 items. This is an excellent demonstration of the intended structure.

Effect of initial random distribution in simulation. At the end of the experiments the type 1 items were closer to the centre than the type 2 items and the type 2 items are closer to the centre than the type 3 items. The results show that the DD algorithm can successfully sort pucks when they start randomly in the arena (figure 7*b*).

Comparison between the two starting conditions. There was no effect of the starting conditions when using the DD algorithm. At the end of the experiments, type 1 items ended up at the same distance from the centre in both conditions (Tukey test, $T=2.793$, $p=0.0585$), type 2 items further out towards the periphery but equally so in both start conditions (Tukey test, $T=0.956$, $p=0.9315$) and type 3 items ended up furthest from the centre but at equivalent distances in each condition (Tukey test, $T=1.49$, $p=0.6715$).

3.3. Comparison between the SD and DD algorithms

After 50 000 time steps, the robots using the 'double density' algorithm had created a denser cluster of pucks than when they used the 'single density' algorithm. Pucks of all types were closer to the centroid in the replicates when the 'double density' algorithm was used.

4. CONCLUSIONS

Two algorithms which employed puck density as a cue were compared. Unlike previous work, such as reported by Deneubourg *et al.* (1991), the density of different types of object was not used and instead the density of all pucks within the local sensing range was employed. Although we have currently not employed different sizes of items we have used colour rather than size or weight to differentiate between types of item. Furthermore, as we are using the notion of total density of sensed items, we therefore speculate that this method could still be used with items of different sizes. Such ideas clearly require further experimentation. We argue that using all objects within the sensory field to generate a measurement of density is more minimalist as it requires less sophisticated sensing and processing than if the density of each item type was assessed separately. The DD algorithm employed density as a cue for a robot to pick up and also deposit a puck. In contrast, the SD algorithm used density as a cue for pick up only. It used a distance 'mechanism' to execute the deposition of a puck. We found that the SD algorithm could sort three different types of puck provided that all the pucks were clustered together to begin with. The DD algorithm, however, did not need the pucks to be clustered in the beginning and could sort effectively whether the pucks were randomly distributed or organized into a cluster at the start of each trial.

Do these findings help us further our understanding of how ants go about sorting? Ants do not appear to employ density as a cue on both pick up and deposition (as in the case of the DD algorithm), that is, their behaviour is consistent with employing something more akin to the SD algorithm in which the clustering of objects into a single cluster precedes spacing (Sendova-Franks *et al.* 2004). Our results show that the SD algorithm is more sensitive to the initial distribution of the objects to be sorted, that is, a single cluster is a necessary start condition. Furthermore, we have suggested that the dropping distances used in the SD algorithm, which relate to the duration and trajectory of the moved object, is consistent with using weight as a cue for determining brood-specific dropping distances in ants (Sendova-Franks *et al.* 2004). In other words, in terms of time, ants move heavy items in 'straight' lines and drop them quickly whereas light items are carried for much longer but are taken via a much more tortuous path i.e. a shorter Euclidean distance. In this way an ant carries a lighter item on a longer path but since the path is very sinuous she ends up dropping the item at a shorter Euclidean distance than would be the case for a heavy item which she carries on a shorter but 'straighter' path. Importantly, in terms of distance travelled, the difference in the way each type of item is carried means that they are dropped at increasing Euclidean distances from where they started as the weight of the items increases.

It is also interesting to speculate why ants would not use the DD approach which could possibly, at first sight, be less time consuming and energetically more efficient. We propose the following hypotheses: (1) constantly monitoring density while carrying brood could be difficult, since brood items could obstruct antennae movement; (2) the speed at which an individual ant could perform a density measurement could, in some or even all ants, introduce a temporal lag leading to a loss of 'synchronization'; (3) with the above two points in mind, the employment of a SD mechanism which does not require the extra sensing and processing demands to discriminate between different densities of different object types would make evolutionary sense; and (4) using the SD approach could confer adaptive flexibility with respect to the number of items sorted—it does not matter, for example, if brood items were lost during emigration to a new nest site. Furthermore, it could be argued that if ants were attempting to sort at the same time as their co-workers were bringing more brood into the nest, then this would be disruptive and hinder spacing. If the cue to switch from the clustering phase to the spacing phase is governed by stability of density, then an incomplete cluster would be (since brood is constantly being dropped nearby) sensed as having more density variation in time than a completed cluster. If the cue to switch behaviour from clustering to spacing is based on the stability of the density cue, then employing the SD algorithm would make sense since it would facilitate cleaner switching between behaviours and result in better spacing.

Alternatively if constantly monitoring density while transporting objects does not present problems in terms of processing demands, sophistication of sensing and budget, then the DD algorithm is an attractive mechanism. For this study our density measuring system in the real robots operated at approximately 0.2 Hz and thus the constant monitoring of density required by the DD algorithm was not a practical option. Of course for simulated robots the constraints may be less severe but the same problems of processing demands might reveal themselves as the simulation scales.

Although sorting, as in the case of patch sorting, does not necessarily have to be carried out using density as a cue (Wilson *et al.* 2004) researchers such as Deneubourg *et al.* (1991) have put forward a mechanism which relies on the perception of density for cues to pick up and put down the items being sorted. We have demonstrated that, for the SD algorithm, density can be used only on pick up and thus constant density measurement is unnecessary. To a large degree this fits the biological evidence, but interestingly, ants will move an object if the density is too high and do not appear to move an object when the density is low (Sendova-Franks *et al.* 2004). In order to make our algorithm work we were required to include a lower density threshold and further work is required to explore this difference.

From the engineer's perspective a device which is simple is likely to be more robust than a more complex one since there is less to go wrong. It is also likely to be cheaper to build and easier to mass produce which is naturally attractive for those considering the building of swarm systems in which decentralization is the key to scalability. Robotics research in this field also offers ideas and questions to ant researchers. Our work demonstrates that clustering is a necessary condition for successful sorting in simulated or real robots that use the SD algorithm. This suggests that ants may employ a similar mechanism. Future work will need to test this hypothesis with further experiments and examination of ant behaviour. Future simulation and robot experiments will also need to take into account such factors as morphological differences between robots and ants as well as means of locomotion.

We gratefully acknowledge the EPSRC and, in particular, the Life Science Interface Programme for supporting this research. We would also like to thank the reviewers for their insightful and useful comments.

REFERENCES

Camazine, S., Deneubourg, J. L., Franks, N. R., Sneyd, J., Theraulaz, G. & Bonabeau, E. 2001 *Self-organization in biological systems*. Princeton, NJ: Princeton University Press.

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. From animals to animats. In *Proc. First International Conf. Simulation of Adaptive Behavior* (ed. J.-A. Meyer & S. W. Wilson), pp. 356–363. Cambridge, MA: MIT Press.

Franks, N. R. & Sendova-Franks, A. B. 1992 Brood sorting by ants: distributing the workload over the work-surface. *Behav. Ecol. Sociobiol.* **30**, 109–123. (doi:10.1007/BF00173947)

Melhuish, C., Holland, O. & Hoddell, S. 1998 Collective sorting and segregation in robots with minimal sensing. In *Proc. Fifth Conf. on the Simulation of Adaptive Behaviour, Zurich* (ed. R. Pfeifer, B. Blumberg, J. A. Meyer & S. W. Wilson), pp. 465–470. Cambridge, MA: MIT Press.

Melhuish, C., Wilson, M. & Sendova-Franks, A. 2001 Multi-object clustering: patch Sorting with simulated minimalist robots. *Proc. Towards Intelligent Mobile Robots (TIMR)*. Manchester. April. Technical Report Series, Manchester University, Department of Computer Science.

Scholes S., (2005) 'Brood sorting in the ant *Temnothorax albipennis*, from biology to collective robotics' *Ph.D. Thesis*, University of the West of England, Bristol.

Scholes, S., Wilson, M., Sendova-Franks, A. B. & Melhuish, C. 2004 Comparisons in evolution and engineering: the collective intelligence of sorting. *Adapt. Behav.* **12(3-4)**, 147–159. (doi:10.1177/105971230401200302)

Sendova-Franks, A. & Franks, N. 1995 Division of labour in a crisis: task allocation during colony emigration in the ant *Leptothorax unifasciatus* (Latr.). *Behav. Ecol. Sociobiol.* **36**, 269–282. (doi:10.1007/BF00165836)

Sendova-Franks, A. B., Scholes, S. R., Franks, N. R. & Melhuish, C. 2004 Brood sorting by ants: two phases and differential diffusion. *Anim. Behav.* **68**, 1095–1106. (doi:10.1016/j.anbehav.2004.02.013)

Wilson, M. 2003 Strategies for sorting and segregation using collective minimalist robots. *Ph.D. Thesis*, University of the West of England, Bristol.

Wilson, M., Melhuish, C. & Sendova-Franks, A. 2002 Multi-object segregation: ant-like brood sorting using minimalism robots. In *Proc. Seventh International Conf. on the Simulation of Adaptive Behaviour, Edinburgh, UK*, pp. 369–370.

Wilson, M., Melhuish, C., Sendova-Franks, A. B. & Scholes, S. 2004 Algorithms for building annular structures with minimalist robots inspired by brood sorting in ant colonies. *Auton. Robots* **17**, 115–136. (doi:10.1023/B:AURO.0000033969.52486.3d)