Go ahead, make my day: Robot conflict resolution by aggressive competition

Richard T. Vaughan **Kasper Støy** **Gaurav S. Sukhatme** **Maja J. Matarić**
Robotics Research Laboratories, Department of Computer Science,
University of Southern California, Los Angeles, California 90089-0781

Abstract

We examine a simulated but realistic multi-robot transport task that suffers from spatial interference. Previously described techniques to reduce interference are not appropriate for this and related tasks.

We demonstrate the utility of an aggressive competition to reduce interference and increase efficiency in our system. A controller is described which breaks deadlocks in favour of the most 'aggressive' robot. Simulation trials are performed to evaluate a variety of aggression functions. Our results and subsequent discussion suggest that neither a linear dominance hierarchy nor a simple sensor bias method offer any advantage over a random outcome.

Finally we discuss some strategies that might favour the 'correct' outcome of competitions to increase the efficiency of the system. We are currently implementing these controllers on a real robot team.

1. Introduction

The disadvantages of centralised control for robot teams were apparent in "Star Wars Episode 1: The Phantom Menace" (Lucas, 1999). The bad robots had a single centralized point of failure; a whole army was rendered useless when their controlling computer was destroyed.

In general, decentralized and distributed systems based on local interactions between otherwise autonomous nodes show robustness to local failures and scale very well. They also readily support ad-hoc changes in population, connectivity and local constraints. (Cao et al., 1995). The real-world ARPAnet was conceived as protection against the Star Wars scenario (Heart et al., 1978). The success of its subsequent development into the Internet is evidence of the power of these ideas.

For these reasons we are developing the abilities of robot teams using only local interaction and no centralized control. In this study we are concerned with teams of autonomous robots performing an ant-like transportation task in the relatively confined space of an office building. We have previously published a biologically inspired trail-following algorithm and demonstrated its utility in loosely-constrained environments (Vaughan et al., 2000b). The motivation for the current work was clear as soon as we began to implement our trail following methods on real robots in our office environment. The problem is that only one robot will fit through a doorway at a time. They may be able to pass each other in a clear corridor, but even small obstacles may effectively reduce the corridor to one 'lane'.

Robot size cannot be reduced indefinitely since many tasks will require some minimum robot size to carry enough widgets or the correct effector. If several robots are to operate in the same space and pass through one-robot-size gaps we need to design controllers which can resolve these conflicts robustly and efficiently. This problem is a special case of interference, the familiar phenomenon of agents reducing overall effectiveness by getting in each other's way.

In nature similar conflicts over resources are often resolved by fighting. Physical combat is undesirable in our robot system as it risks damage to the participants and would be costly in time and energy. Similar costs apply to real creatures and many have evolved stylized aggressive competitions as abstractions of dangerous combat (Maynard-Smith, 1982, Krebs and Dawkins, 1984).

We are investigating the use of aggressive behaviour to improve the efficiency of robot teams; this paper presents our initial simulation experiments. Our eventual goal is to demonstrate these methods running on real robots, so we have tried to keep the simulations and controllers realistic and as easy to transfer to the real world as possible. We demonstrate that a simple stylised fighting behaviour improves the overall performance of our system by reducing interference. We then discuss the (non)usefulness of social dominance hierarchies and suggest ways to improve overall efficiency.

2. Task

A team of robots starts from a common home position, explores an unknown environment and locates a supply of resource. On reaching the supply location

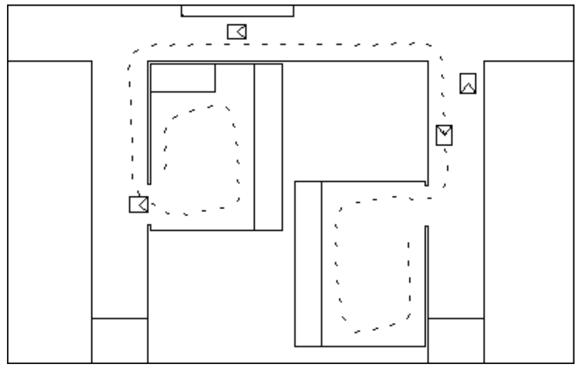

Figure 1: Experimental environment in Arena. Dashed line is the crumb trail, small rectangles are robots.

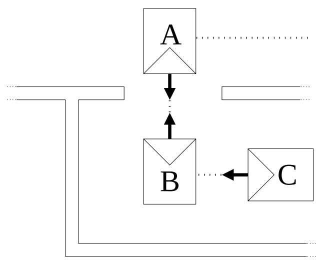

Figure 2: Co-localization conflict: robots A and B are following the trail (dotted line) in opposite directions and must get through the doorway, but only one will fit at a time.

each robot receives a unit of resource and must return home with it, then return to fetch more resource repeatedly for the length of the trial. In a previous paper (Vaughan et al., 2000b) we describe an ant-like trail laying and following algorithm that achieves the task, is decentralized and is robust to large uncertainty in robot localization. In this paper our robots follow a fixed pre-layed 'crumb trail' from home to source in order to test our navigation and interference-reducing strategies in isolation from the ant algorithm.

Figure 1 shows a screenshot of our simulation environment in which two rooms of an office building are home and source locations connected by a crumb trail.

3. Fighting interference

3.1 Too many cooks

The benefit of adding robots to a task decreases as the incidence of interference increases. The plot in Figure 3 shows a hypothetical performance P versus population N trade off. The curve will touch the $P = 0$ axis at least twice; at $N = 0$ and $N >= N_{max}$ where the interference is so great that the system completely breaks down. As an extreme example this would happen if the the floorspace became totally full of robots so that none could move. In between these points lies the working range of the system. For any given (robot, controller, environment, task) combination, there is at least one optimum population size.

The benefit gained from an interference-beating behaviour will be evident on this graph. There are three ways this can occur: (1) increasing the maximum performance M; (2) increasing the range of the population size which performs adequately Q; (3) both of these, thus increasing the area under the curve but above the minimum acceptable performance level O. Ultimately O would be determined in advance by the system user. Note that it is possible for P to be a multi-modal function of N, and that the curve may have several peaks rather than the single peak shown here.

3.2 Anti-interference strategies

In order to be successful in a constrained environment like our office building (Figure 1), robots must traverse the same path many times resulting in a high incidence of spatial interference. Several authors have described systems for reducing interference, typically by requiring robots to keep as far apart as possible and/or work in non-overlapping regions.

In any environment where there are one-robot-size gaps robots cannot simply be repelled from each other as they are in (Arkin and Balch, 1998). This will inevitably lead to deadlock. Consider Figure 2 in which three robots are shown approaching a doorway. Robot A follows the trail (dotted line) down the page; B and C follow it up the page. If robots A and B were mutually repellent, they would be stuck on either side of the doorway, blocking robot C until some other mechanism broke the symmetry.

Bucket brigade algorithms have been demonstrated to reduce interference (Fontán and Matarić, 1998) and might be useful in this task. However this would require a robust method of passing resource between robots which may be object specific and therefore expensive,

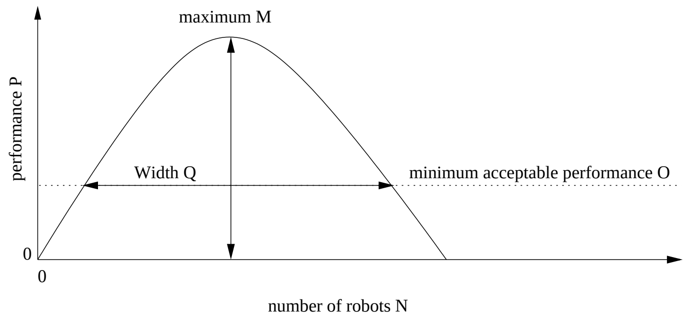

Figure 3: Population versus performance trade-off

or may be prohibitively difficult or even dangerous for some objects. A bucket brigade would not prevent interference in some similar tasks, for example if there were two unrelated non-interoperable teams of robots working in the same space.

Dominance hierarchies have also been shown to reduce interference in some scenarios. (Goldberg and Matarić, 1997) explored dominance hierarchies in foraging experiments with four mobile robots; the dominant robots got to move first while the others waited their turn. While interference was thus significantly reduced, the overall time-to-completion of the experiment increased undesirably.

We examine the utility of a dominance hierarchy for resolving aggressive conflicts in the experiments below.

3.3 Aggressive signalling

There is an extensive and active literature on aggressive behaviour and signalling in nature. This work suggests that aggressive signalling behaviour offers several advantages for the agents competing over a resource compared with all-out fighting. If rival agents are able to assess their relative chances of winning a competition and act according to the most likely outcome, then the potentially high cost of a real fight can be avoided (Enquist, 1985).

Aggressive signals tend to honestly reflect the fighting ability of the signaler. They may be costly to perform, requiring the presence of the trait they indicate to ensure reliability and prevent exploitation by bluffers (Zahavi and Zahavi, 1997, Grafen, 1990, de Bourcier and Wheeler, 1995). Even so, they are less costly than the direct competition they represent.

3.4 Channelling aggression

The experiments in this paper investigate the use of aggressive behaviour to improve performance in multi-robot systems. Specifically, we demonstrate a stylized aggressive-competition behaviour which increases the system's effectiveness compared to an adequate but non-aggressive controller. The competition breaks the symmetry between robots with identical controllers, preventing deadlocks.

Our robots have a trait we call *aggression*, a scalar function of some robot state. We evaluate three alternative aggression functions. When a conflict for space occurs a robot performs a stereotypical movement that indicates its level of aggression to its competitor. Each robot compares its own level of aggression with that displayed by the other; if its own aggression is smaller it performs a stereotypical submissive behaviour for a short time. The more aggressive robot will resume its normal behaviour immediately.

The advantages of this scheme are:

1. it requires only existing sensors and slightly modified behaviours;
2. it is totally decentralised;
3. it is independent of the navigation strategy;
4. it requires only trivial computation;
5. it is useful for heterogenous robot systems. Nothing need be known about the internal mechanisms of the competitor, so long as it displays its aggressiveness.

4. Experiments

This section describes a number of experiments performed to evaluate the ability of aggressive competition to reduce interference in the resource transportation task.

4.1 Simulation

Arena is a multiple mobile robot simulation for UNIX-like operating systems. It was developed and is used as a research and teaching tool at the University of Southern California (USC). It provides a simple kinematic and sensor model for differential-steer ground robots such as those used in many robotics research labs. The default robot size and sensor arrangement is similar to the ActivMedia Pioneer 2DX (`http://www.activmedia.com`)and

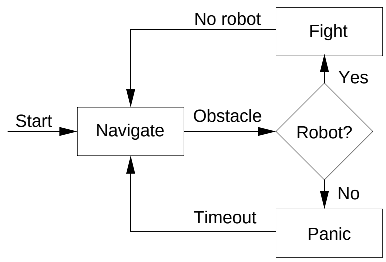

Figure 4: Overview of the control system

is used in these experiments, but this is easily customized by the user. The experiments in this paper use the simulated robot's front and rear sonar rings and the SICK laser range finder.

The robots' sensors and wheel motors are modeled with low fidelity to achieve a very high update frequency (≈200Hz with 20 robots on a modest 400MHz Pentium II). We believe this is justified by the presence of noise and wide variation in characteristics of real sensors and actuators and the presence of noise. These factors combine to reduce the utility of carefully constructed hi-fi robot models, particularly if the user intends her control algorithms to run on real robots (Jakobi, 1997, Vaughan et al., 2000a). For these experiments we do not impose artificial sensor noise on top of the low accuracy values reported by the simulator. However, Arena runs in real time and the update period per simulation cycle is determined by unpredictable external factors (system load, network traffic, etc.). This injects considerable stochastic variation and we observe that no two runs are the same.

Collisions are treated simply; if the robots bump into something they are stuck unless they turn away from the obstacle.

Arena is implemented as a TCP/IP network server. Robot controllers are independent client processes communicating with Arena via a socket. Arena sends a message to each connected controller at 10Hz indicating the corresponding robot's current speed, turn rate, position estimate and sensor readings. Controllers asynchronously send messages back to Arena indicating the latest speed and turn rate demands for their robot.

Environments are specified as occupancy grids, imported as black and white bitmaps. We can draw simple environments with boulder-like obstacles, or we can import more interesting CAD models of real buildings to provide rich, realistic environments.

4.2 Robot behaviour

The robot controller is composed of three high-level behaviours: *navigate*, *fight* and *panic*. One of these behaviours is active at a time as determined by the state transition diagram shown in Figure 4. The behaviours and transitions are outlined here; details of their implementation are given later.

- *Navigate* drives the robot around the environment following walls, going through openings (e.g. doorways) and following crumb trails. The robot does its useful work in this behaviour.

- *Panic* is designed to cope with situations that confound *navigate*. It 'unsticks' the robot from obstacles, dead ends and traffic jams by turning randomly and moving into free space.

  Navigate and panic alone are sufficient to perform the transportation task.

- *Fight* is our conflict resolution mechanism. It consists of two mutually exclusive sub-behaviours: *brave* and *afraid*. *Fight* chooses which one to use based on a 'fear threshold'.

An emergency-stop mechanism runs continuously in parallel, monitoring the laser and sonar sensors; if the robot gets too close to any obstacle it will stop and the controller will change state.

If the robot has emergency-stopped and senses a robot in front of it, it switches to the *fight* behaviour. Otherwise it switches to *panic*. If no aggression mechanism is used the control system always switches to *panic*.

4.3 Details of the behaviours

4.3.1 Navigate

The navigate behaviour is fairly sophisticated and contains several sub behaviours. It is a development of the controller described in our previous trail-following paper (Vaughan et al., 2000b), adapted for constrained environments. It works well on our real Pioneer 2DX robots equipped with a SICK laser scanner and front and rear sonars.

At each time step the behaviour detects openings to the left and right of the robot using the laser scanner. An opening is identified if there is an obstacle followed by a large discontinuity in the extreme left and rightmost laser samples (the SICK provides 360 samples over 180 degrees). If an opening is detected and the robot is not already turning the robot makes a sharp 90 degree turn into the opening.

The opening-seeking behaviour is overridden by a trail-following component if the trail points away from the opening. If the robot is heading in the opposite direction to the trail it turns 180 degrees. If there is no reason to make sharp turns the behaviour defaults to wall following.

The wall following behaviour is implemented using a sliding box algorithm. A virtual box a little bigger than the robot is placed to the left in the laser scan. The

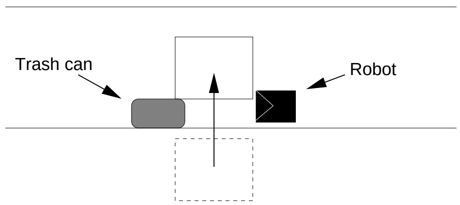

Figure 5: Principle of the sliding box algorithm. The free-space box is initially placed to the left of the robot and is slid in the direction of the arrow until there are no obstacles detected inside the box.

box is moved left to right in front of the robot until the laser scan shows no obstacles within the box. The robot moves forwards at a constant speed, turning towards the center of this box (see Figure 5). If there are no obstacles in the first box that is placed it is assumed that the robot is in a big open space and therefore moves forward.

The advantage of this approach compared with approaches that try to keep a fixed distance to the wall is that the size of the box can be tuned so that the robot keeps minimum distance to the wall but still has time to move away from the wall to avoid obstacles. If no open space is found the control system turns the robot in the opposite direction of the nearest obstacle. This mechanism is also used by the *afraid* sub-behaviour (Section 4.3.3) to drive the robot safely backwards. No laser data is available behind the robot, so *afraid* uses the rear-facing sonars instead.

In order to reduce the interference between the robots as much as possible we use a simple algorithm to filter other robots out of the laser scan. All behaviours except emergency stop are given this filtered laser scan. The laser samples that correspond to a robot are removed by interpolating the first non-robot points on either side of the robot's image. If the robot is at the edge of the scan only one point is used. The effect of this can be seen in Figure 6. It is not perfect, but prevents the robots from seeing each other as openings and walls. In the real world the parts of a laser scan that correspond to robots can be identified approximately by using a colour camera to detect the directions to the robots and then find the corresponding region in the laser scan. Our simulated laser scans return different 'reflectance' values for robot strikes than wall strikes.

This navigation controller is carefully designed to make the most of the limited space available in our environment. Navigate will keep control of the robot and follow the trail unless the emergency stop mechanism detects an obstacle critically close to the front of the robot. These behaviours make it possible for the robots to perform their task in small rooms and narrow corridors, passing within as little as 40cm of each other before interference occurs.

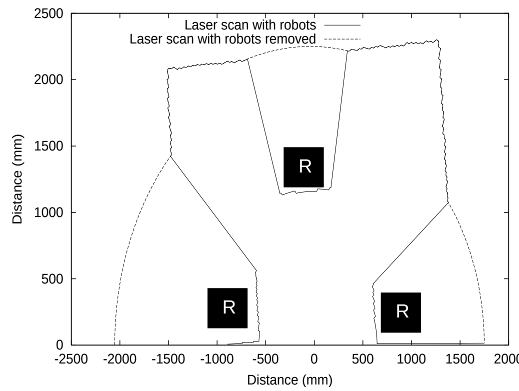

Figure 6: A scan made by laser scanner located at (0,0) heading in the direction of the y-axis. Three robots are detected in the scan; their positions are marked with black boxes. The samples corresponding to robots are removed and the resulting filtered scan is shown.

4.3.2 Panic

Panic drives the robot forward if the area in front of it is clear of obstacles. Otherwise the behaviour turns the robot in a random direction until the area in front of the robot has been clear for a short time. Then the robot moves forward again. After a few seconds the behaviour times-out and switches to the *navigate* behaviour.

4.3.3 Fight

Each robot calculates its fear threshold (the minimum distance it will tolerate to another robot) when it first detects another robot within 2m in front of it, and remembers this value until no robots are detected within 2.5m. The fear threshold is a function of the robot's aggression level, discussed later.

If a rival robot is detected within the fear threshold, the *afraid* sub-behaviour is chosen. This drives the robot backwards away from the competitor (using the rear sonars to avoid obstacles if necessary) as long as the fear threshold is invaded.

If the rival is outside the fear threshold, the *brave* behaviour uses the same mechanism as *navigate* to continue to follow the trail. If the control system has been continuously *brave* for two seconds the controller switches back to the *navigate* behaviour.

The emergency stop mechanism continues to operate during *fight*: if an obstacle (including other robots) gets too close the control system emergency stops the robot and switches to the *panic* behaviour.

The most aggressive robot will be the first to become *brave*. It will usually get past its rival by pushing it back until there is room to pass. Once the winner is past and outside the loser's fear threshold, the loser becomes *brave* and continues to follow its trail. Figure 7 shows

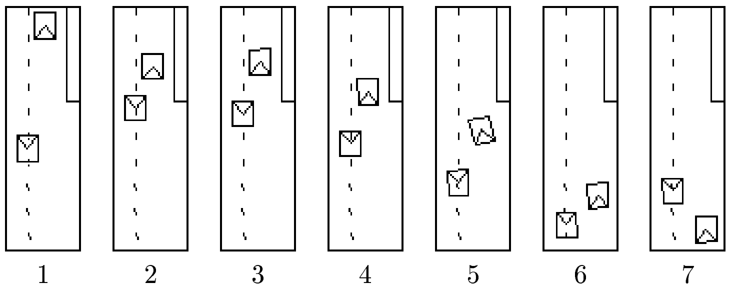

Figure 7: The aggression mechanism in action. 1) Two robots approach each other in a narrow part of a corridor. 2) The robots get too close and make an emergency stop. 3) The robots start to fight. Both are *afraid*; they back off because the other robot is within their fear thresholds. 4) The top robot reaches its fear threshold first, becomes *brave* and moves forwards. 5-6) The bottom robot is still *afraid* and is pushed down the corridor. 7) The robots have successfully passed each other.

how two robots successfully negotiate a narrow corridor by *fight*ing.

4.4 Aggression function

At the beginning of a competition each robot must determine its aggression level. The most aggressive robot will win the competition. We tested four alternative methods for setting aggression:

1. **None:** no fight is performed - a control test;
2. **Random:** aggression is determined at random for each encounter;
3. **Fixed hierarchy:** a different fixed aggression is assigned to each robot initially so that they form a linear dominance hierarchy;
4. **Personal space:** aggression is proportional to the free space behind the robot measured with the rear sonar. Thus a robot with a lot of space behind it is very aggressive. This is intended to bias the outcome of a fight towards the robots trying to get out of rooms; they will tend to have more empty space behind them compared to those in the corridor.

These aggression functions are qualitatively different; **random** will ensure a random outcome to each fight, while **fixed hierarchy** and **personal space** are intended to test alternative systematic outcomes; a hierarchy is a common symmetry-breaking technique, while the sensor-based **personal space** attempts to use some local information to determine a favourable outcome.

4.5 Procedure

Each controller was evaluated in simulation for populations of n =1,2,3 and 4 robots. n robots are placed in fixed starting positions in Arena in each trial. The simulation is started and the identical controller runs on all the robots. Each trial runs for 800 seconds. Each (controller, population) pair was run 20 times to gather enough data for statistical testing.

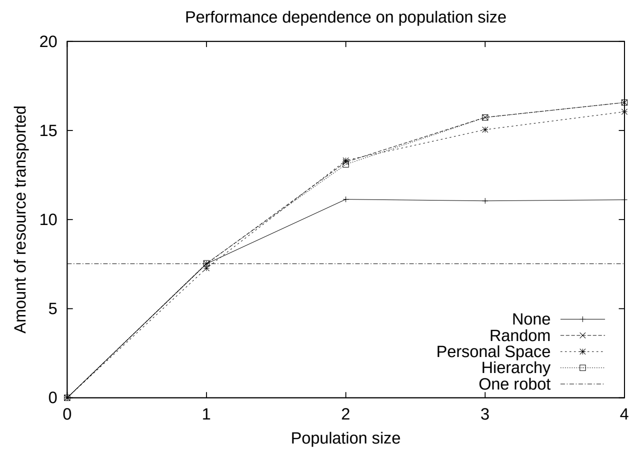

Figure 9: Experimental results: population versus performance trade-off

4.6 Measurements

The performance of the system was measured by counting the number of units of resource that are transported over the length of the trial. This is an ideal metric because it directly reflects the purpose of the system and is objective and easy to measure.

The fraction of total time spent in each behaviour was also recorded. Time spent in the *navigate* behaviour will correlate well with useful work; time spent in other behaviours is overhead due to the limitations of *navigate* or interference from other robots. The use of metrics like this to evaluate interference effects is discussed in (Goldberg and Matarić, 1997).

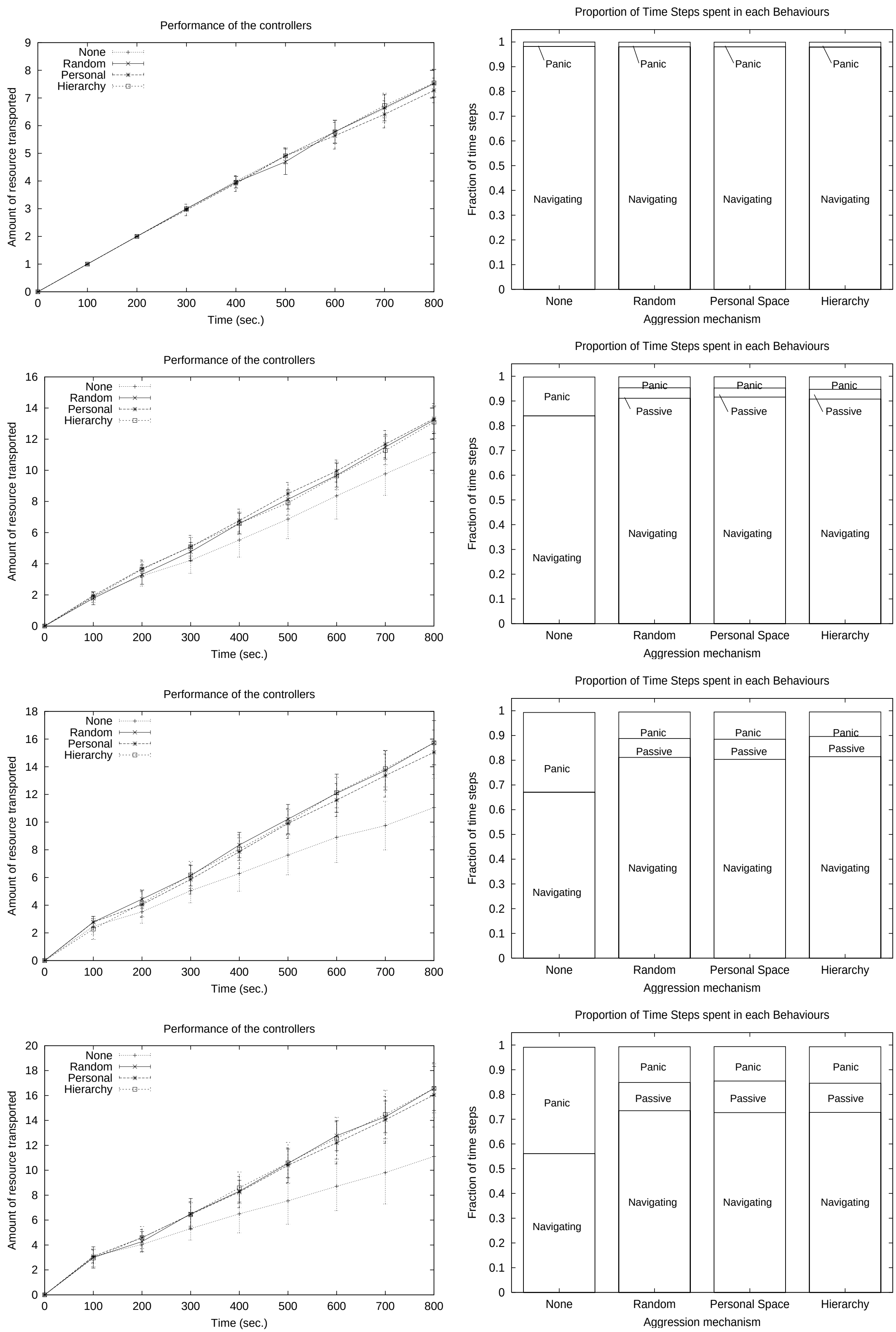

Figure 8: Experimental results: performance (left) and behaviour decomposition (right) for populations of 1 (top) to 4 (bottom).

4.7 Results

Figure 8 presents summary results of all the experiments. Two graphs are shown for each population size 1 (top) to 4 (bottom); those in the left-hand column show the resources transported over time; those in the right-hand column show a breakdown of the proportion of time spent in each behaviour. Each graph shows results for all four alternative aggression functions.

The top left graph in Figure 8 shows the resources transported over time with a population size of one. No interference can occur; the fighting behaviour is never used, and all four control systems perform similarly. The low variance of the results over 20 trials show that the system performs consistently. The top right graph in Figure 8 shows the behaviours performed during the trial. The robot spent 98% of its time in *navigate* and 2% in *panic*.

The performance graphs show that as the number of robots is increased the controllers that use an aggression mechanism show a higher number of resources transported than the non-aggressive controller. It can also be seen that regardless of which aggression mechanism is used the performance is the same. These results are confirmed by statistical analysis below.

The behaviour breakdown graphs show that as the number of robots is increased the control systems spend more and more time in the panic behaviour due to the increased interference between the robots. In the experiment with a population size of four the non-aggressive controller spends 43.9% of the time using the panic behaviour. This percentage is reduced to 14.6% when an aggression mechanism is used, but an additional 12.7% is used in the *afraid* behaviour (labelled passive in the graphs) meaning that a total of 27.3% is spent in behaviours other than *navigate*.

To compare these results, Figure 9 shows the population size versus performance trade-off for this experiment. The non-aggressive controller reaches its maximum performance at a population of 2. The performance of the aggressive controllers continues to increase with population size, but the increase is not very significant when using four instead of three robots.

Comparing Figure 9 with Figure 3 we observe that there is insufficient data to draw firm conclusions about the maximum performance M or the useful population range Q.

However, if we assume that there is no further increase in performance with population size for any controller, as seems likely, then we see a benefit of around 30% in M for aggressive over non-aggressive controllers. It is difficult to say anything about Q because it depends on how fast the performance drops when using larger population sizes. Further experiments can clarify this.

f-test	n,r	n,p	n,h	r,p	r,h	p,h
1	<0.01	0.79	0.01	0.79	0.01	0.77
2	0.07	0.27	0.57	0.92	0.62	0.65
3	0.40	0.45	0.43	0.08	0.04	0.04
4	0.03	0.85	0.10	0.20	0.74	0.47

Figure 10: F-test. Left column indicates population size. Other columns shown the result of a two-tailed f-test between datasets gathered using the different aggression mechanisms (n = none, r = random, p = personal space and h = hierarchy).

t-test	n,r	n,p	n,h	r,p	r,h	p,h
1	>0.99	0.09	0.88	0.09	0.88	0.07
2	<0.01	<0.01	<0.01	0.75	0.66	0.49
3	<0.01	<0.01	<0.01	0.16	0.98	0.18
4	<0.01	<0.01	<0.01	0.46	>0.99	0.48

Figure 11: T-test. Left column indicates the population size. Other columns shown the result of a two-tailed t-test between datasets gathered using the different aggression mechanisms (n = none, r = random, p = personal space and h = hierarchy).

4.8 Statistical analysis

For each population size, we aim to establish whether the performance data for each controller is drawn from distributions with equal mean value.

First for each population we use a two tailed F-test to see if the variances of performance for each controller are significantly different. In table 10 the probabilities that the variances are not significantly different are shown.

Second, based on the above test we use the appropriate T-test to test for significantly different mean values. In table 11 the probabilities that the data sets belong to distributions with equal mean value are given. The table shows that on a 95% confidence level the mean values for performance measured using the non-aggressive controller are significantly different from the remaining data sets. Additionally it is shown that the data sets gathered using the random, personal space and hierarchy based aggression mechanisms belong to distributions with equal means on a 95% confidence level.

Intuitively this means that the performance of the control system using an aggression mechanism is significantly better than the control system that does not. It also shows that the performance gain of the aggression mechanism is independent of our choice of aggression function.

5. Discussion

We have shown that a stylized aggression mechanism can improve performance by reducing interference in a transportation task. However, we saw that the two systematic aggression functions *dominance hierarchy* and *personal space* performed no better than a random resolution. This section discusses the reasons for this and suggests some possible ways to improve upon this result.

5.1 Failure of the dominance hierarchy

A dominance hierarchy offers no measurable advantage in these experiments. A fixed hierarchy guarantees that conflicts will be resolved, and resolved the same way between the same individuals on every meeting. This is a useful symmetry-breaking mechanism; why does it not help here?

Hierarchies in nature tend to reflect the fitness (in some sense) of the competing individuals. The means whereby the hierarchy is established usually requires that the fitness is demonstrated in some way. This ensures that the position in the hierarchy is an honest indication of the ability of the individual. For a dominance hierarchy to be effective, individuals must maintain some real advantage over one another for the duration of several conflicts. This produces a net advantage in resolving tasks without having to recalculate the hierarchy.

However our individuals are identical; none has any true advantage over the others, except in their current situation. The fitness of an individual in our task is determined by the very short-term situation that the robot finds itself in: a particular arrangement of doorways, robots and obstacles. The fixed hierarchy cannot encode any information about this state - it is of the wrong temporal resolution - so cannot determine the Right Thing to do. This puts a limit on the utility of fixed dominance hierarchies in populations of clones.

This result is consistent with (Matarić, 1993) who finds that a fixed dominance hierarchy imposed upon identical robots in a dispersal task offers no performance benefit over random arbitration.

We hypothesize that a dominance hierarchy will be effective only when there are (i) non-uniform abilities in the group and (ii) a relatively slow change in the abilities of individuals. These conditions are relatively unusual in robotics with the important exception of systems with learning, evolution or other long-term adaptation.

5.2 Doing the Right Thing

In some robot conflicts the designer will favour a 'correct' outcome; that which has the greatest benefit to the performance of the overall system. For example in Figure 2 it is probably more efficient for robot A to to back off, giving way to robot B, as robot A has more room to do so. This may be a challenging task for a decentralized controller.

Decentralized systems are seemingly at a disadvantage compared to a central executive when it comes to resolving such conflicts in small-scale systems. With global knowledge of the state of the world and control over all robots, planners and schedulers can perform well. We note however that with increasing population, task complexity and/or increasing uncertainty in the environment, conventional planning becomes less tractable. We aim to design decentralized solutions that do not rely on world models and will scale readily.

5.3 Memory and (a)social behaviour

One way of increasing the information available to the signaler is to remember past interactions. The signal can now be a function of *state history* instead of current state. If we allow that the robot has memory of past competitions, there are two simple functions that can be considered:

1. **Past wins raise current aggression, increasing probability of further wins:** This will lead to the emergence of a classic dominance hierarchy and we have seen that this has no advantage over a random outcome.
2. **Past wins lower current aggression, decreasing probability of further wins**: This will encourage turn-taking as the aggression level across the population will tend to converge on the average value. The difference in aggression level between competitors will decrease until it is imperceptible to the robots sensors. At or near this point the outcome is essentially random and can offer no advantage.

These observations are interesting because many social structures are based on the ability to remember past interactions and recognize individuals that have been encountered before (Chase, 1982). In particular this is a requirement for cooperation to be stable in a population (Axelrod, 1984). In the current task with no capacity for long-term differences between robots, the ability to recall past interactions or recognize individuals is no use at all.

In order to improve the performance of this system, we must find a way for the robots to figure out who is in the instantaneously best position to progress, i.e. which robot should win in order to maximise the system's throughput.

5.4 Integration and investment

The naive *personal space* strategy was also indistinguishable from a random fight. Perhaps it was more successful at escaping from rooms as was intended, but suffered an equal penalty elsewhere. It seems more likely that it was just not a useful piece of state to communicate. What other information could be used?

Aggression could accumulate over time rather than being calculated instantaneously. This allows the robot to signal its *investment* in a task. We hypothesize that integrating personal space measurements over time may give a better indication of the 'right' robot to win a fight. A robot that has been navigating in a cramped corridor for a long time will accumulate aggression and should prevail over a competitor who has spent less time in the corridor and has thus invested less in the manoeuvre. The notion of 'investment' in a task is an important concept in models of autonomy in animals (McFarland and Bosser, 1993).

5.5 Increasing signalling efficiency

The fighting behaviour we employ takes a little time to perform, as the robots face off and retreat in their inverse game of chicken. We may find some performance increase by designing a more efficient signalling strategy, perhaps using colour or sound to indicate aggression. However, the current scheme is attractive as it does not require any special purpose sensors; our (and many other) robots carry a laser ranger for navigation already.

5.6 Fighting intentionally

We have discussed the utility of encoding and displaying various properties of the robots' state in an abstract form; in this case aggression. The display can be made independent of the innards of the signaler, so that hetergeneoous robots could interact through displays that are indications of their intent. A receiving robot adopts the 'intentional stance' (Dennett, 1987), interpreting the display as an intent to perform some action and can react appropriately. The intentional stance can thus be thought of as an effective interface between autonomous systems, and therefore used as a design tool for building controllers for robot teams. We intend to develop these ideas further in the future.

6. Conclusion

Building on our previous work in decentralized behaviour design for robot teams, we have shown how an aggressive signalling strategy reduces interference in a realistic multi-robot resource transport task. Previously described anti-interference measures are not appropriate for this class of task.

A suitable behaviour-based control system was described and tested in simulation. It was designed to be feasible for real world robots. It is independent of the navigation strategy employed, though it complements our ant-inspired trail building strategies.

We have shown that dominance hierarchies are not useful in this class of system and have discussed ways to increase the efficiency of the system by incorporating the right kind of temporal information into an aggression function.

We are currently implementing these controllers on a real robot team.

Acknowledgements

This work is supported by DARPA grant DABT63-99-1-0015 and contract DAAE07-98-C-L028.

References

Arkin, R. C. and Balch, T. (1998). Cooperative multiagent robotic systems. In Kortenkamp, D., Bonasso, R. P., and Murphy, R., (Eds.), *Artificial Intelligence and Mobile Robots*. MIT Press.

Axelrod, R. (1984). *The Evolution of Cooperation*. Basic Books, New York, NY.

Cao, Y., Fukunaga, A. S., Kahng, A. B., and Meng, F. (1995). Cooperative mobile robotics: Antecedents and directions. In *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS'95)*.

Chase, I. D. (1982). Dynamics of hierarchy formation: the sequential development of dominance relationships. *Behaviour*, 80:218–240.

de Bourcier, P. and Wheeler, M. (1995). Aggressive signalling meets adaptive receiving: further experiments in synthetic behavioural ecology. In Moran, F., Moreno, A., Merelo, J., and Chacon, P., (Eds.), *Advances in Artificial Life: Proc. Third European Conference on Artificial Life*, pages 760–771. Springer.

Dennett, D. C. (1987). *The Intentional Stance*. MIT Press, Cambridge, MA.

Enquist, M. (1985). Communication during aggressive interactions with particular reference to variation in choice of behaviour. *Animal Behaviour*, pages 1152–1161.

Fontán, M. S. and Matarić, M. J. (1998). Territorial multi-robot task division. *IEEE Transactions on Robotics and Automation*, 14(5).

Goldberg, D. and Matarić, M. J. (1997). Interference as a tool for designing and evaluating multi-robot controllers. In *Proceedings, AAAI-97*, pages 637–642, Providence, RI. AAAI Press.

Grafen, A. (1990). Biological signals as handicaps. *Journal of Theoretical Biology*, 144:517–546.

Heart, F., McKenzie, A., McQuillan, J., and Walden, D. (1978). *ARPANET Completion Report*. DARPA and BBN, Washington, D. C.

Jakobi, N. (1997). Evolutionary robotics and the radical envelope of noise hypothesis. *Adaptive Behavior*, 6(2):325–368.

Krebs, J. R. and Dawkins, R. (1984). Animal signals: mind reading and manipulation. In Krebs, J. R. and Davies, N. B., (Eds.), *Behavioural Ecology: an evolutionary approach*, pages 380–402. Blackwell.

Lucas, G. (1999). Star wars episode 1: The phantom menace (movie). LucasFilm/Fox.

Matarić, M. J. (1993). Kin recognition, similarity, and group behavior. In *Proc. Fifteenth Annual Conf. Cognitive Science Society*, pages 705–710. Lawrence Erlbaum Associates.

Maynard-Smith, J. (1982). *Evolution and the Theory of Games*. Cambridge University Press.

McFarland, D. and Bosser, T. (1993). *Intelligent Behavior in Animals and Robots*. MIT Press.

Vaughan, R., Sumpter, N., Henderson, J., Frost, A., and Cameron, S. (to appear 2000a). Experiments in automatic flock control. *Robotics and Autonomous Systems*.

Vaughan, R. T., Stoy, K., Sukhatme, G. S., and Mataric, M. J. (to appear 2000b). Whistling in the dark: cooperative trail following in uncertain localization space. In *Proc. Fourth Int. Conf. Autonmous Agents*.

Zahavi, A. and Zahavi, A. (1997). *The Handicap Principle: a missing piece of Darwin's puzzle*. Oxford University Press.