Rational Aggressive Behaviour Reduces Interference In A Mobile Robot Team

Sarah Brown*, Mauricio Zuluaga†, Yinan Zhang‡ and Richard Vaughan§
School of Computing Science
Simon Fraser University, Burnaby, BC, Canada
* Email: sbrown1@sfu.ca
† Email: mzuluaga@sfu.ca
‡ Email: yinanz@sfu.ca
§ Email: vaughan@sfu.ca

***Abstract*— Spatial interference can reduce the effectiveness of teams of mobile robots. We examine a team of robots with no centralized control performing a transportation task, in which robots frequently interfere with each other. The robots must work in the same space, so territorial methods are not appropriate. Previously we have shown that a stereotyped competition, inspired by aggressive displays in various animal species, can reduce interference and improve overall system performance. However, none of the methods previously devised for selecting a robot's 'aggression level' performed better than selecting aggression at random. This paper describes a new, principled approach to selecting an aggression level, based on robot's investment in a task. Simulation experiments with teams of six robots in an office–type environment show that, under certain conditions, this method can significantly improve system performance compared to a random competition and a non–competitive control experiment. Finally, we discuss the benefits and limitations of such a scheme with respect to the specific environment.**

I. INTRODUCTION

One problem frequently encountered in multi–robot systems, especially those without centralized control, is interference between agents. Interference in general can be characterized as competition for resources, for example, access to a charging station or use of a shared tool or sensor. Most commonly, robots simply get in each other's way during normal navigation about the environment. An acute version of this problem is getting two Pioneer–sized (0.5m) robots through a standard (0.8m) doorway from opposite directions: some symmetry–breaking mechanism is required to decide who goes first. This is a real–world problem for robot applications such as mail delivery, factory and warehouse AGVs, and assisted–operator wheelchairs.

One possible solution seen in nature is to have the agents fight to determine a winner of the conflict. However, actual robot combat is potentially costly for the individual agents involved and to the overall system as robots may become damaged. The same costs apply in nature and many species have developed aggressive display competitions as abstractions of physical combat [7], [6], [2] to solve this problem.

In a simulated agent–based competition for food resources, [10] has shown that using a display of aggression results in better performance for agents in their abstract version of fighting, a stopping game. Their agents were able to perceive the aggression levels of opponent agents and use this information to calculate the cost of fighting for a resource.

Previously, Vaughan, et al. in [12] have demonstrated a stylized aggressive display in teams of robots performing a transportation task, to resolve space conflicts in doorways and narrow corridors in a simulated world similar to that shown in Figure 1. The robots must work in the same space, so territorial methods [4], [1] are not appropriate. A key advantage of this system was that communication of 'aggression level' between agents was performed using only the existing navigation sensors and actuators, i.e. there were no special–purpose sensors, no wireless communication and no need for unique identifiers for each robot. Thus the method is perfectly scalable and can be used in heterogeneous systems and even in human–robot interaction: humans can easily understand and manipulate the behaviour of the aggressive robots.

The symmetry–breaking provided by the aggressive competition was shown to produce better overall system performance, in terms of the number of transportation trips completed, compared to an otherwise identical system that lacked the aggression mechanism. Changing the behaviour of the robots in this way does not eliminate interference: in a typical one–on–one competition, the 'winning' robot certainly interferes with the immediate progress of the losing robot. Yet as the overall system performance is increased, we may say that the overall negative interference is reduced.

Several strategies for determining a robot's aggression level during a 'fight' were evaluated: random aggression; a linear dominance hierarchy; and 'personal–space' method, where aggression was determined by the amount of free space visible to the robot. All methods were shown to have statistically similar performance: neither the dominance hierarchy nor the personal space method offered any improvement over a random outcome.

This paper describes a new, principled approach to selecting an aggression level, based on robot's investment in a task. The term 'investment' towards achieving a goal is fundamental in models of autonomy in animals [8]. Simulation experiments with teams of six robots in an office–type environment show that, under certain conditions, this method can significantly improve system performance compared to a random compe-

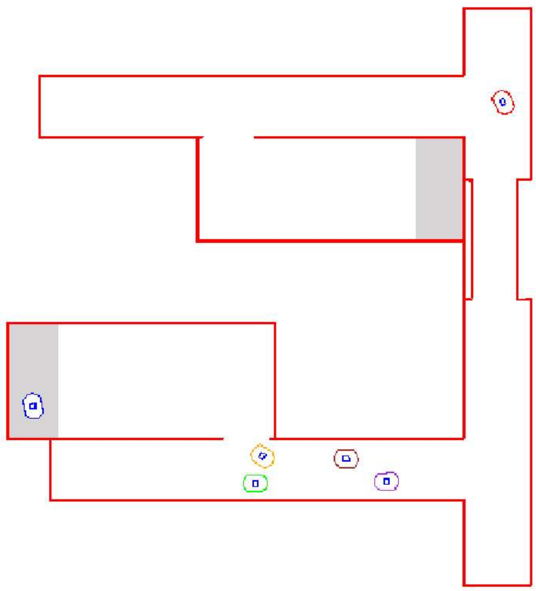

Fig. 1. The Stage world containing six robots used in Experiment 1. It closely resembles the environment used in [12].

tition and a non–competitive control experiment. Finally, we discuss the benefits and limitations of such a scheme with respect to the specific environment.

II. Rational Aggression

To improve performance compared to a random outcome, the outcome of an aggressive interaction must reflect some relevant state of the world. As argued in [12], a hierarchy of robots with fixed aggression levels can not encode any information relevant to the outcome of a particular competition; when two robots meet at a doorway, their status in the hierarchy does not matter, so long as one of them gives way. Adding memory of past robot/robot interactions does not help: similar arguments apply to dynamic hierarchies.

In general to control some parameter of a system, we must measure it or estimate it from its correlates. To maximize the amount of work done by our robot system, we need an estimate of how much work a robot is doing as an input into our control system. This principle leads to the following economic approach to this problem.

Consider a system of two robots, Black and White, working in a narrow corridor as shown in Figure 2. They have the same task; transporting widgets from A to B at either end of the corridor. Assume that it is not practical for robots to transfer widgets between themselves. Black starts at A, White at B. At some moment, shown in the top row of the figure, Black and White block each other's progress. Assume the robots have an internal aggression level, and can perform a stereotypical behaviour sequence called a *fight*, in which each robot displays its aggression to the other. If a robot perceives that its rival has a higher aggression, it goes into a passive mode and can be pushed backwards. By performing a *fight*, Black and White can can resolve their conflict; the more aggressive robot will push

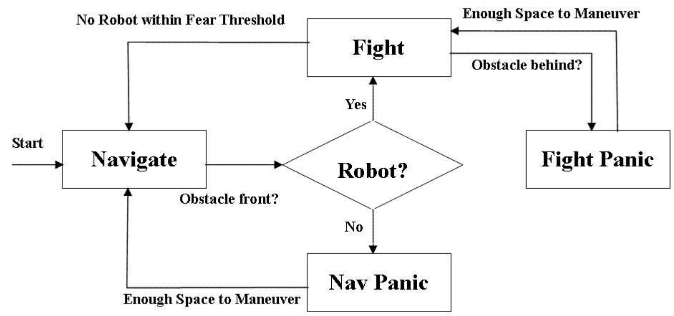

Fig. 3. Schematic of the robot control program.

the other backwards and out of the way. From the point of view of system efficiency, which robot should be more aggressive?

The arrows beneath the robots indicate how far the robots have travelled towards their goals. This travel inevitably has real cost in terms of time, energy and computation. These are *sunk costs*: they can not be recovered. In the left column, Black wins the fight and pushes White along the corridor until Black reaches its goal (middle row). Then Black switches to goal A, and proceeds down the corridor followed by White. At some point (bottom row), White is now back where it started to fight, after travelling the distance indicated by the arrow. The cost of the fight is the sum of White's sunk cost plus the cost to get back to its start position.

The right column shows the outcome if White wins the fight. The steps are the same, but the total cost of Black losing the fight (total length of arrows) is much smaller. In this thought experiment, the robot with the higher sunk costs should be more aggressive as it has more to lose. With this scheme, the system will achieve more trips from A to B in unit time than with randomly chosen aggression. The method is economically rational: it makes decisions based on the expectation of a favourable outcome. We will refer to this as the 'investment' method.

The investment method can be implemented very simply by adding a minimal memory to the robot; a counter. Each control loop cycle, the counter is incremented. On reaching a goal the counter is reset to zero. The value in the counter reflects the amount of time the robot has spent on reaching its current goal. The aggression level is set proportional to this value.

III. Transportation Task

We have implemented an investment based scheme for determining the aggression level of the robots in a similar office environment to that used by Vaughan, et al. in [12]. Our environment has the same conditions for spatial interference: only a single robot fits through a doorway, and there is one narrow corridor that permits only a single robot to pass at a time. Figure 1 shows six robots in the simulation environment. Robots must repeatedly traverse the environment between the shaded areas.

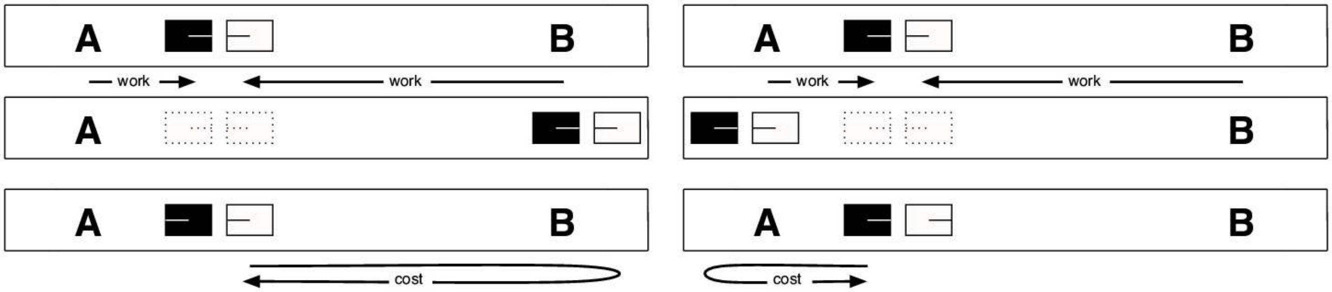

Fig. 2. Motivation for the *effort* strategy. The robot that has invested the most work in a task, and therefore has more to lose, should win a resource conflict.

A. Control Architecture

Each robot runs the same control program, shown schematically in Figure 3. Each mode is described below.

1) Navigate: *Navigate* is the default mode for the robot and executes until an emergency stop is triggered. The implementation of navigate is an adaptation of the controller in the original paper. Instead of using a crumb trail to navigate, our controller utilizes a virtual map which, for any point in the world, provides the correct general direction for the robot. The direction vectors change depending on which goal area the robot is currently seeking.

Robots perform left wall–following using a sliding box algorithm where a virtual box slightly larger than the robot is moved left to right until no obstacles from the laser scan are detected within it. This box is long enough to prevent the robot from accidentally drifting into doorways and touches the robot slightly at all times to prevent very close obstacles from going undetected. Robots adjust their heading to aim for the centre of the box providing smooth wall following and obstacle avoidance. Other robots in the system are not treated as obstacles, i.e. robots should not try to navigate around each other. To prevent this, robots are removed from the laser scan and the data to replace them is interpolated from the remaining laser scans.

The wide corridors in the simulated world have enough space for two pioneer robots to pass comfortably. When turning corners, however, robots that swing too wide, or not wide enough, can interfere with each other. To solve this problem, robots which are turning left make very sharp left turns and robots turning right use the far right corner of the wall, as detected in the laser scan, to make the widest turn possible. If a robot turns too sharply (or not sharply enough) to make a turn correctly, this may trigger emergency stop which would normally lead to the robot entering the *panic* mode. This is inefficient so a small corrective turn was added to the navigate behaviour for this situation. Robots in navigate will first attempt to make a very small rotation towards the direction in which the most space is perceived from the laser scan then resume navigation. If this fails navigate will exit abnormally and the *panic* mode will become active.

2) Panic: The purpose of *panic* is to move robots away from situations in which they are stuck, either because of obstacles or other robots interfering. Initially, the robots that are panicking simply sleep for a random number of cycles. If the obstacles which caused the robot to panic in the first place are still present, the robot will first attempt to rotate for a random length of time and if unable to rotate, the robot will try to move backwards slightly. Moving backwards provides space for robots which are too close to walls or obstacles to turn without hitting any objects. Robots stay in the panic cycle and switch between sleeping, rotating, and moving forward until they are able to navigate forward successfully for a random period of time.

3) Fight: The *fight* procedure is triggered when a robot emergency–stops and detects another robot in front. Usually this is because the other robot is blocking the way. When entering the *fight* mode, the robot calculates its *fear threshold*: the minimum distance it will tolerate to another robot. The robot starts fighting by moving backwards; it continuously backs away as long as it detects another robot within its fear threshold. If the rival is outside its fear threshold, the robot switches back to the *navigate* mode. If the robot is too close to an obstacle while it is moving backwards, the emergency stop mechanism is invoked; then, the robot stops and switches to the *panic* procedure. During a fight the robot with the smaller fear threshold will be the first to start driving forward again - this robot is the *winner*. Consequently, it will push its rival (the *loser*) backwards until there is enough room to pass. Once the winner moves outside the loser's fear threshold, for example by passing, the loser starts navigating again and the fight is over.

B. Aggression Function

The fear threshold is determined by a robot's *aggression* α, where $0 < \alpha < \alpha_{max}$, selected at the start of the *fight* procedure. *Fight* is designed so that the more aggressive robot is likely to be the winner. The aggression is determined by the Aggression Function F.

The fear threshold is the minimum distance one robot can tolerate to another robot and is inversely proportional to a robot's aggression, plus some offset distance.

$$\phi = K_1 + \frac{K_2}{\alpha} \tag{1}$$

Constants K_1 and K_2 were chosen to give a fear threshold between 450mm and 2450mm in our experiments. A tiebreaker mechanism is employed in order to reliably resolve the fight between two robots with very similar fear thresholds. It adds a small random distance to a robot's fear threshold, ensuring that two robots with the same aggression have different fear threshold values. It breaks the symmetry between two robots by preventing them from switching from *fight* to *navigate* at the same time.

1) None: No fight is performed. This is a control test, and a robot *panic*s if it is blocked by another robot.

2) Random: Aggression α is chosen at random in the range $0 < \alpha < 10$.

3) Investment: Aggression value is proportional to the time a robot has spent approaching the goal in each trip. A robot's aggression increases with the time it has spent on the *navigate* behaviour within the current trip. Specifically, we calculate the aggression A using the formula:

$$\alpha = min(K_3 \frac{T}{T_{normal}}, \alpha_{max}) \quad (2)$$

where T is the time spent approaching the current goal, T_{normal} is a normalization constant reflecting the expected time to reach the goal, and K scales the aggression to the desired range. α_{max} sets the upper bound of α, so that $0 < \alpha < \alpha_{max}$.

IV. Experimental Design

This section describes the experiments carried out to evaluate the advantage of investment based aggression over random based aggression and no aggression at all.

A. Task

Our robots have the mission of transporting resources back and forth between two goal locations (shaded areas in the Rooms, Figure 1 and Figure 6).

The Robots initially start from a pre–assigned, randomly selected location in the world and proceed to the first goal area (bottom room) to collect their first unit of resource. Then, they transport it to the second goal area (top room) where it is released and another is collected to be moved to the first goal area. Each robot transports one unit of resource in each trip between the goal areas.

B. Procedure

We have n robots living in a world W. The world has rooms and corridors where the robots can move, the doors and some sections of the corridor are narrow and only allow one robot to pass by. All other sections of the corridors are wide enough to allow two robots to pass across when going in opposite directions.

For each aggression function of 'none', 'random' and 'investment', where 'none' means that fighting is disabled, we run a number of trials n_{trials} that last for a number of seconds $trialLength$. Every time we start a new trial, the location of the robots is reset to the same initial position (all trials have the same starting conditions).

These parameters allow us to control the degree of spatial interference between the robots in the world, in a way that we can still evaluate the performance of each of the different aggression functions. In this paper we present experiments in two different worlds $W1$ and $W2$. We found that setting the number of robots $n = 6$ produced a good degree of robot interference without saturating the world, which means that the robots can still achieve their goals.

C. Performance Metric

As soon as a trial starts, all robots begin to log information regarding the type of aggression function used, the trial number, the robot number, the number of trips completed, and the total time spent in *navigate*, *fight* and *panic* behaviours.

To measure the success of a trial we count the total number of trips performed by all the robots

$$Trips_{team} = \sum_{i=1}^{n} Trips_{robot_i} \quad (3)$$

This value is easy to obtain and represents an object measurement of the performance of the system as a whole. In our resource transportation task, each trip completed by a robot is equivalent to one unit of resource transported. We are not trying to improve the number of trips that a single robot does but the number of trips that the team of robots complete.

Given the total number of trips done by a team of robots, we can compare the results of different trials and different aggression functions with one another.

We also compare the success of a group of robots against a possible best. To compute the maximum possible number of trips that a team of robots could execute, we do the following:

1) Find $Trips_{one}$, the average of the total number of trips that a single robot does during a $trialLength$ for n_{trials} trials. In this experiment, the robot uses the same controller of the multirobot experiments.
2) Multiply $Trips_{one}$ times the number of robots (n) and obtain the upper–bound on the number of trips a team of robots could do $Trips_{max}$ (See equation 4). In reality, this would only be possible if none of the robots would interfere with each other during the trial. Note that ($Trips_{max}$) is also the upper–bound on the number of resources which could be transported in the system.

$$Trips_{max} = Trips_{one} * n \quad (4)$$

$$0 \leq Trips_{team} \leq Trips_{max} \quad (5)$$

There is a direct relation between the number of trips performed in a trial and the total time spent in navigation, fighting and panic. The greater the number of trips, the greater the navigation time and the lower the fighting and panic times. Because our goal is to reduce interference between robots as much as possible, most of the time spent by a robot during a trial should be navigation time. Fight and Panic times are wasted time; they do not directly contribute to towards achieving a goal. Therefore a good interference reduction method should increase navigation time by reducing one or both of the panic and fight times.

D. Simulation

Our experiments are done in simulation using the Player/Stage robot development and simulation system [5]. Player is a commonly–used server and abstraction protocol [11] that connects a user–defined control program to the sensors and the actuators on a mobile robot. Stage is a robot simulator that provides multiple virtual robot devices to Player. Our Stage models approximate ActiveMedia Pioneer–3DX robots. Their dimensions are 44cm long and 33cm wide, and they are equipped with front and rear sonar rings, and a SICK laser range finder. Player and Stage are freely available under the GPL from `http://playerstage.sourceforge.net`.

E. Stats Tests

The primary statistical test used to evaluate the performace of the three different aggression functions was a T–test with $\alpha < 0.05$. The T-test shows whether the means for the number of resources transported by each of the aggression schemes is significantly different. Some of the trials were considered to be *outliers*, that is, their total resources transported values differed greatly from the rest of the data collected. We used 1.5 times the interquartile range (IQR) as fences and those trials which fell outside of the fences were considered outliers and removed from the data set. While the removal of the outliers did affect the standard deviations of each of the tests, it did not greatly affect the significance of the T-tests.

V. Experiment 1

In this experiment, we used the following parameters:

- $n = 6$, the total number of robots.
- W like Figure 1, the world where the robots live.
- $trialLength = 1800$, the length of each of the trials in seconds.
- $n_{trials} = 24$, the total number of trials executed for each aggression function.
- $Trips_{one} = 21$, the average number of trips performed by a single robot in the world W.
- $Trips_{max} = n * Trips_{one} = 6 * 21 = 126$, the upper bound on the number of trips for a team of robots.

We set the aggression function such that $0 < \alpha < 10$, with the aggression reaching 80% of maximum in the normal time taken to reach a goal. This time was determined empirically by measuring the mean time taken by a single robot to drive between goals, in the absence of interference, to be 84 seconds. Substituting these parameters, we obtain an aggression function of T alone:

$$\alpha = min(8\frac{T}{84}, 10) \tag{6}$$

$$\alpha = F_{exp1}(T) \tag{7}$$

A. Results

The results of the trials in this environment show no improvement in performance using the investment based aggression scheme over the random scheme. Though, both random and investment schemes perform better than having no fighting mechanism.

Based on the thought experiment described previously, we believe the reason that the investment and random aggression schemes performed no differently was due to features of the environment. First, the narrow portion of the corridor is nearly halfway between the two goal areas. Robots which fight in this region have approximately the same aggression level, so a random outcome to a fight there would be equivalent to the investment outcome.

Second, the narrow region is relatively short in length, causing robots which lose to back up for at most a few seconds of time. Over the span of a short trial, the lost time does not significantly impact the performance of the system. The penalty for the ‘wrong’ robot losing the fight is very small, so the investment method offers no significant advantage over the random method. This insight into the nature of this environment may help to explain the results in [12], where a ‘personal space’ strategy was also shown to be equivalent to random.

VI. Experiment 2

Because the first experiment failed to show a significant difference between the random and investment aggression schemes, a second experiment was performed; identical but for a different environment. The new world (Figure 6) was designed to show the features of the investment method seen in the thought experiment. To increase the penalty for the ‘wrong’ robot losing a fight, the corridors were lengthened to accommodate a longer narrow region which could also be further offset from the halfway point between the two goal areas. A second narrow area of the corridor was also added to increase the opportunities for robots to fight in the corridors. This environment has the same basic structure that the old environment had but has increased areas of spatial interference. It is also fairly realistic office environment.

The new world with its lengthened corridors is approximately twice the size of the old world. Tests with a single robot in both worlds confirmed that the average time from one goal area to the other is very close to twice the time from the old environment to the new. For this reason, the length of the trials was doubled.

The following is a list of the parameters used in this experiment:

- $n = 6$, the total number of robots.
- W like Figure 6, the world where the robots live.

TrialType	Mean Resources	σ	N	Outliers
NoFight	28.0	11.4	24	0
Random	82.4	19.2	22	2
Investment	79.5	8.5	23	1

TABLE I

EXP1 RESULTS: SUMMARY OF PERFORMANCE SCORES FROM THREE DIFFERENT CONTROLLERS.

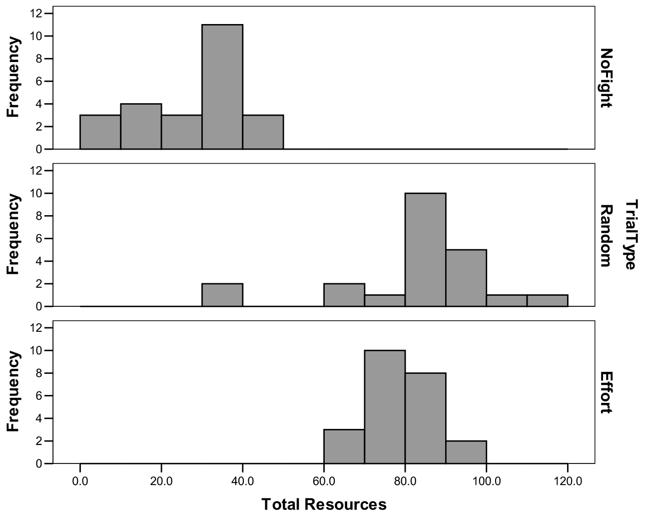

Fig. 4. Exp1 results: Histograms showing distribution of performance scores for three different controllers: no fight (top), random aggression (middle), investment aggression (bottom)

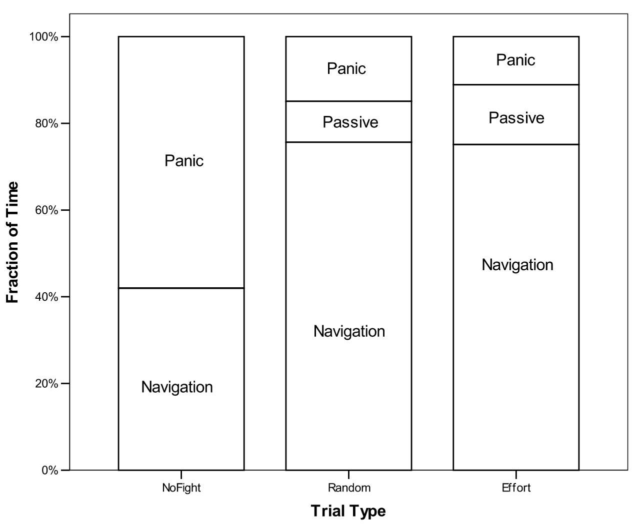

Fig. 5. Exp1 results: Proportion of time spent in each activity.

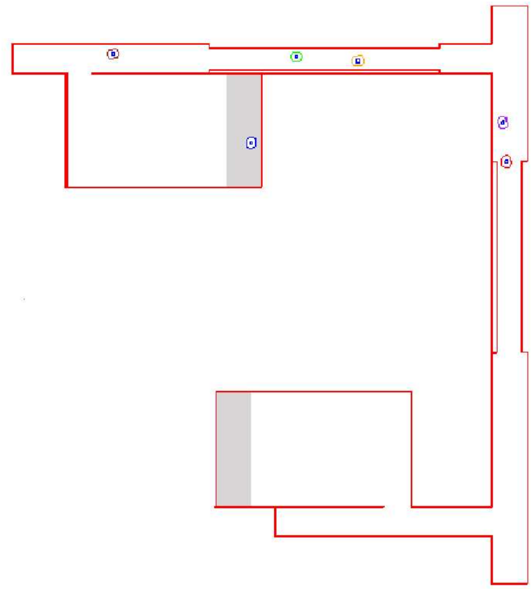

Fig. 6. The world used in the second set of experiments

- $trialLength = 3600$, the length of each of the trials in seconds.
- $n_{trials} = 24$, the total number of trials executed for each aggression function.
- $Trips_{one} = 22$, the average number of trips performed by a single robot in the world W.
- $Trips_{max} = n * Trips_{one} = 6 * 22 = 132$, the upper bound on the number of trips for a team of robots.

The Investment aggression function was set similarly to experiment 1, except that T_{normal} was larger due to the larger environment, so:

$$\alpha = min(8\frac{T}{152}, 10) \quad (8)$$

$$\alpha = F_{exp2}(T) \quad (9)$$

The configuration of this experiment allows some comparison between both experiments as the total number of trips possible for a robot is also approximately the same.

A. Results

The T–tests revealed that the investment based aggression function performed significantly better than the random scheme with reference to the average number of resources transported per trial. As well, both random and investment methods out–performed the no–fight method. The difference between the random and investment schemes can be clearly seen in the histogram (Figure 7) and in the bar chart (Figure 8).

VII. DISCUSSION

A. Large standard deviation in the Results

One issue with the results is the large standard deviation found in all three fighting schemes. We believe the major

TABLE II

RESULTS

TrialType	Mean Resources	σ	N	Outliers
NoFight	25.1	13.6	23	1
Random	67.8	17.8	23	1
Investment	112.9	13.4	22	2

TABLE III

EXP2 RESULTS: SUMMARY OF PERFORMANCE SCORES FROM THREE DIFFERENT CONTROLLERS.

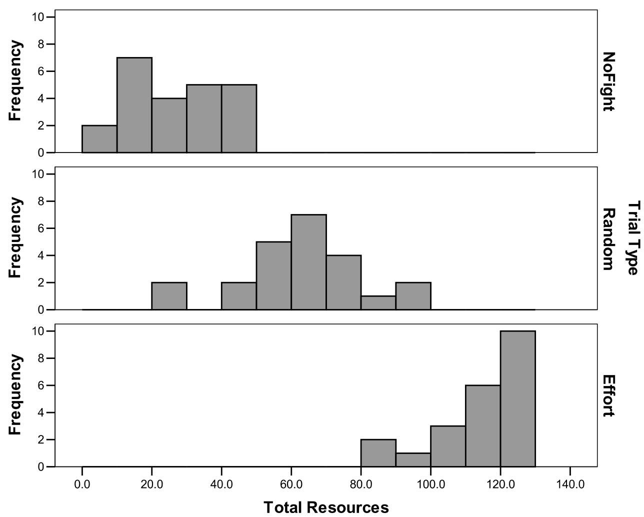

Fig. 7. Exp2 results: Histograms showing distribution of performance scores for three different controllers: no fight (top), random aggression (middle), investment aggression (bottom)

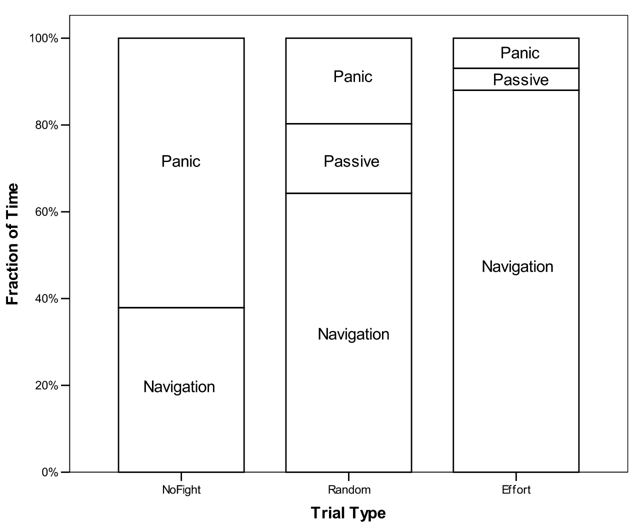

Fig. 8. Exp2 results: Proportion of time spent in each activity.

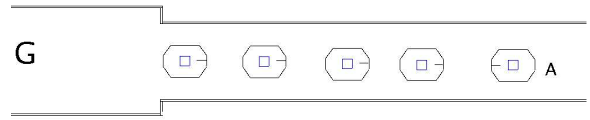

Fig. 9. The right–most robot is travelling to the left while other four robots are travelling to the right.

factors contributing to this problem are the nature of the controller, the world, and the simulation environment. First, because the environment is complex and the trials are lengthy, there may be some random situations that our controller cannot handle. Second, the simulation environment, Stage, is asynchronous and there may be a lag between the time when robots request new data from their sensors and the time when they receive it. When these factors are coupled with multiple robots running in this environment, the trials can produce wildly different outcomes. Perhaps running larger number of trials could reduce the large standard deviation to values which appear more reasonable. Nevertheless the results we have obtained are still valid and significant.

B. Minimizing the cost of a fight

The cost of a fight is proportional to the distance for which each robot has to back away before the winner moves forward. Because there is a lag between the time a robot sends a change of speed command to Stage and the time it receives the up–to–date sensory data, if two robots have very close fear threshold values, they may find each other outside their fear thresholds and start navigating again at the same time. Thus, in our controller, we deliberately make the fear threshold values of two robots with similar aggressions more distinguishable. It is more time–efficient and desirable if at least one of the participants of a fight is highly–aggressive, so neither of the robots has to back away very long before the winner stands out. In the contrary, if both robots are very submissive, they have to move away for a much longer distance. In the worst case, if both robots have an aggression of 10, in total they have to back away 4900mm before one robot wins. Also, resolving a fight in a very short amount of time can help prevent a third robot from participating in the same fight. If one robot moves backwards, and another robot comes from the opposite direction and does not stop on time, the robot that moves backwards will panic; then there will be two robots that are fighting against a panicking robot in the middle. If such a scenario takes place in a corridor, as a chain effect other robots that also tries to pass this corridor are dragged into this fight, and this usually causes a jam which may take a very long time to resolve.

C. Fight between two teams of robots

In some special cases, the Investment strategy does not do the right thing. In Figure 9, the robot A (travelling to the left) collides with a team of four robots moving in the opposite direction. It will be more reasonable if A loses the fight in favour of the overall group performance. However, under the

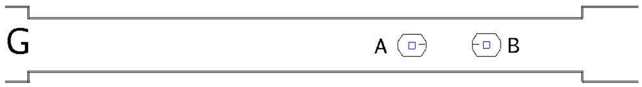

Fig. 10. The robot to the left has spent more time in the corridor than the robot to the right, so ought to win the fight.

Investment strategy, the robot A is closer to its goal than the other four robots. A wins the fight and pushes the team of robots backwards, incurring around four times the normal cost.

D. The importance of the environment

The very different results of experiment 1 and experiment 2 show that the environment plays a vital role in the performance of the aggression functions. In fact, the nature of the environment may dictate which aggression function should be selected. For instance, if the rooms in these experiments had been considerably smaller, it is conceivable that random could perform significantly better than the investment based scheme. Since investment biases fights towards robots entering rooms where the goal areas are located, this aggression function could produce situations in which rooms are filled up with robots, effectively saturating the space. In this case, random might be a better choice as an aggression function as it has no bias towards any robot in any situation.

E. Local interference aggression

In Figure 10, it is preferable for robot A (travelling to the right) to beat robot B because the robot B would soon back out of the narrow corridor, allowing A to pass. However, in its current form the investment strategy can not take this into account.

A simple modification could tackle this: the aggression of a robot should increase with the time it has spent in a narrow place such as a corridor, so that a robot that has spent a long time in a narrow space should be more aggressive than a robot that has invested a shorter time in the same corridor.

F. Aggression vs. Niceness

We have described our conflict–resolution procedure as an 'aggressive display' or 'fight'. The method was inspired by aggressive displays in animals, and the robots' behaviour tends to be described this way by naive observers. However we are not suggesting that aggression or violence is necessary to resolve conflicts in robot teams. The 'fight' procedure is logically equivalent to a stereotyped politeness routine, for example whereby two civil people meet at a doorway, each suggesting that the other goes first, until some symmetry–breaking mechanism is applied. Our 'aggression' maps onto 'politeness' or 'niceness' by simple inversion; the reader can choose their preferred metaphor.

VIII. Future Work

We are investigating other aggression functions that may avoid some of the limitations of the investment method. In particular, the local investment strategy looks promising, though it will not solve the team–fight problem. We also aim to demonstrate adaptive selection and modification of the aggression function at run–time, optimizing the system dynamically in changing environments.

IX. Conclusion

We have extended previous work on reducing interference in robot teams by devising a novel aggression function, based on the idea of investment in a task. A simple implementation was described, and the method was shown to increase the performance of a simulated robot team at a classical transportation task in the case where there was significant cost in resolving conflicts at random. Some weaknesses and limitations of the method were described, along with possible improvements. A explanation for the results described in [12] was suggested. The investment method should be widely applicable as an interference reduction technique in mobile robot teams.

Acknowledgements

The authors would like to thank Jens Wawerla for his suggestions in the creation of the tie breaker mechanism.

References

[1] R. C. Arkin and T. Balch. Cooperative multiagent robotic systems. In D. Kortenkamp, R. P. Bonasso, and R. Murphy, editors, *Artificial Intelligence and Mobile Robots*. MIT/AAAI Press, Cambridge, MA, 1998.

[2] Robert Axelrod. *The Evolution of Cooperation*. Basic Books, New York, NY, 1984.

[3] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2:14–23, 1986.

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

[5] Brian P. Gerkey, Richard T. Vaughan, Kasper Stoy, Andrew Howard, Gaurav S. Sukhatme, and Maja J. Matarić. Most valuable player: A robot device server for distributed control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1226 – 1231, 2001. (Also appears in Proceedings of the Second International Workshop on Infrastructure for Agents, MAS, and Scalable MAS at Autonomous Agents 2001, Montreal, Canada, May 29, 2001).

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

[7] John Maynard-Smith. *Evolution and the Theory of Games*. Cambridge University Press, 1982.

[8] D. McFarland and T. Bosser. *Intelligent Behaviour in Animals and Robots*. MIT Press, 1993.

[9] David McFarland. *Animal Behaviour*. Longman Scientific & Technical, 1985.

[10] Matthias Scheutz and Paul Schermerhorn. The more radical, the better: Investigating the utility of aggression in the competition among different agent kinds. In *Proc. of SAB 2004, Los Angeles, CA*. MIT Press, 2004.

[11] Richard T. Vaughan, Brian P. Gerkey, and Andrew Howard. On device abstractions for portable, reusable robot code. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2121–2427, Las Vegas, Nevada, U.S.A, Oct 2003. (Also Technical Report CRES-03-009).

[12] Richard T. Vaughan, Kasper Stoy, Gaurav S. Sukhatme, and Maja J Mataric. Go ahead, make my day: robot conflict resolution by aggressive competition. In *Proc. Int. Conf. Simulation of Adaptive Behaviour, Paris, France*, 2000.