

Altruistic Task Allocation despite Unbalanced Relationships within Multi-Robot Communities

Ryan D. Morton, George A. Bekey, and Christopher M. Clark

Abstract—Typical Multi-Robot Systems consist of robots cooperating to maximize global fitness functions. However, in some scenarios, the set of interacting robots may not share common goals and thus the concept of a global fitness function becomes invalid. This work examines Multi-Robot Communities (MRC), in which individual robots have independent goals. Within the MRC context, we present a task allocation architecture that optimizes individual robot fitness functions over long time horizons using reciprocal altruism.

Previous work has shown that reciprocating altruistic relationships can evolve between two willing robots, using market-based task auctions, while still protecting against selfish robots aiming to exploit altruism. As these relationships grow, robots are increasingly likely to perform tasks for one another without any reward or promise of payback. This work furthers this notion by considering cases where an imbalance exists in the altruistic relationship. The imbalance occurs when one robot can perform another robot’s task, thereby exhibiting altruism, but the other robot cannot reciprocate since it is physically unable (e.g. lack of adequate sensors or actuators). A new altruistic controller to deal with such imbalances is presented. The controller permits a robot to build altruistic relationships with the community as a whole (one-to-many), instead of just with single robots (one-to-one). The controller is proven stable and guarantees altruistic relationships will grow, if robots are willing, while still minimizing the effects of selfish robots. Results indicate that the one-to-many controller performs comparable to the one-to-one on most problems, but excels in the case of an unbalanced altruistic relationship.

I. INTRODUCTION

As the population of mobile robots increases, more robots will be forced to share common workspaces. Unlike typical Multi-Robot Systems (MRS), in which robots are designed, owned, or operated by a *single* individual or organization with the purpose of achieving a common goal, there will be an increase in the frequency of situations in which robots are deployed by different individuals with different goals.

We consider a Multi-Robot Community (MRC) to be a collection of robots 1) sharing a common workspace, and 2) that are designed, owned, or operated by *several* individuals or organizations that may have different goals. This greatly affects the dynamics of the community because individual robots may be designed to accomplish their own

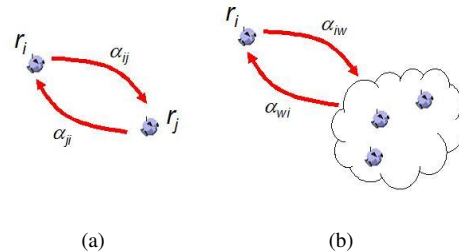


Fig. 1. Two views of altruism in a MRC. (a) Where an edge from robot r_i to r_j represents the willingness of r_i to incur a cost for doing one of r_j ’s tasks (one-to-one). (b) Alternatively, the edges can represent the willingness of r_i to incur a cost for any robot in the MRC (one-to-many). The edges in opposite direction represent the reciprocal altruism.

goals without concern for the goals of other robots. That is, robots in a MRC may be selfish.

Autonomous robots acting in such a MRC must learn how to interact with the other robots in the community. The ideal situation occurs when all relationships become mutually beneficial; in which case the community would behave as a MRS. In a MRC, however, individual robots must offer varying levels of “help” to other robots. This help, termed an altruistic action, decreases the fitness function of the helping robot without any guarantee of reward or repayment. However, this altruistic action may later be reciprocated, leading to a net increase in fitness functions for both robots over the long term.

This work presents two altruistic controllers that build altruistic relationships between willing robots, while offering protection against selfish robots (see Fig. 1). Both controllers model altruism based on Robert Trivers’ theory of reciprocity [15]. The first altruistic controller, termed *one-to-one*, allows for tit-for-tat type increases in the altruism one robot has towards another robot. The second altruistic controller, termed *one-to-many*, more closely resembles the altruism typically modeled in the social sciences [17], in that altruism increases between one robot and the whole community.

In [6], it was demonstrated that the one-to-one controller successfully improves the individual cost functions of robots in a MRC (see Fig. 2). Unlike the work from [6], this work considers *unbalanced* altruistic relationships. An unbalanced relationship exists when robot A cannot, or will not, assist with any of robot B’s tasks, but robot B can assist robot A.

Consider two ground robots and an Unmanned Aerial Vehicle (UAV) equipped with cameras for surveillance tasks in a nearby area. One of the ground robots (call it robot A) is tasked to obtain images of high-priority targets using a

R. D. Morton is a Computer Engineering student at California Polytechnic State University, San Luis Obispo, CA, USA spacemort@gmail.com

George A. Bekey is with the College of Engineering at California Polytechnic State University, San Luis Obispo, CA, USA gbekey@calpoly.edu

C. M. Clark is with the Department of Computer Science at California Polytechnic State University, San Luis Obispo, CA, USA cmclark@calpoly.edu

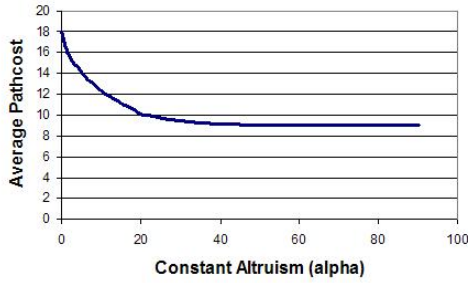


Fig. 2. Average path cost versus altruism level (α). As the level altruism increases, robots tend to help each other more and the average paths to complete all tasks decreases.

high-resolution camera. The UAV, given the task of obtaining images of local air vehicles, has a camera capable of taking images for itself and robot A. The other ground robot (B) has a low-resolution camera enabling it to only obtain images of the quality necessary for its targets and the UAV's. In addition, robot A can acquire images for robot B, but cannot point its camera upward and thus cannot assist the UAV in taking pictures of nearby air vehicles.

An altruistic triangle is formed in which A can only aid B who can only aid the UAV who can only aid A. From robot A's point of view, it can take two courses of action with respect to altruism:

- 1) Not help robot B - since there's no possible reciprocity directly from B
- 2) Help robot B, hoping it receives equal or greater help from the UAV

This work looks at whether robot A should act altruistically, by helping robot B, and will demonstrate the advantages for do so.

II. BACKGROUND

Multi-Robot Systems have been an active area of robotics research due to several potential advantages over single robot systems. Namely, they offer the possibility for greater spatio-temporal sampling, force multiplication, and robustness to failure. Hence, advancements in the areas of MRS mission planning, MRS motion planning, MRS localization, MRS mapping, and the most relevant subject of MRS task allocation have occurred over the last two decades.

Task Allocation: In a MRS task allocation problem, the MRS must accomplish a set of tasks often characterized by their geographic location. The problem seeks to determine the optimal assignment of task points to robots, and the optimal sequence for robots to visit these task points while minimizing the time to visit all task points. This is a variation of the Multi-Traveling Salesperson Problem (MTSP), a problem with many applications, which has no polynomial time solution (e.g. [14]). Regardless, many good heuristic driven methods have been developed that yield sub-optimal solutions.

One popular method of assigning tasks to robots in a MRS is to use a Market-Based Auction approach [8]. In this

method, tasks are auctioned off to robots with the highest bids. Bids are typically based on the ability of the robot to accomplish the task, while considering the additional cost of traveling to the task site. While this method is not guaranteed to find optimal solutions, it is efficient and can lead to near optimal solutions.

Altruism: In the literature on robotics, there are extensive treatments of cooperation among robots (e.g. [4], [3], [12], and [10]), but little mention of altruistic behaviors. Cooperation may in fact involve altruism, but it is generally not described in those terms.

Work directly involving altruism includes that of [13] and [9] who describe robot behaviors in terms of a "satisfaction index" and transmission/reception of signals from other robots. A robot's progress in a given task can be measured by its "satisfaction" in the task, which corresponds to the fitness or performance index indicated above. Thus, a robot needing help with a task may emit an attraction ("please help me") signal. Robots receiving this signal may stop their current task and assist the caller, thus exhibiting altruistic behavior. Clearly, such behaviors can be quantified. Lucidarme et. al. propose an "altruism vector" based on the satisfaction index of a robot and the signals emitted by other robots; a given robot decides on altruistic behavior based on the magnitude of this vector.

Similarly, [1] and [2] describe a software architecture for robot colonies based on robot tropisms, defined as their "likes" and "dislikes". Reinforcement of particular behaviors strengthens them in future scenarios. Here too, a robot can call for help to other robots when it needs assistance in moving heavy objects beyond its capability.

The emergence of cooperative behaviors has been studied extensively in game theory, under the name "Prisoner's Dilemma", e.g.[11]. However, the winning strategies in this situation call for cooperation, not altruism.

In [5], altruism and spite are modeled within a traffic routing setting, considering traffic flow in a graph. Latency is shown to be bounded for general topologies. As opposed to [5], the work here aims to use control and estimation theory to place bounds on the negative effects of selfish behavior.

Reputation Management: Reputation Management (RM) occurs when an agent evaluates the actions of other agents, forms opinions about those agents, and then uses these opinions to adjust its own actions. The field of RM involves analysis of such processes with applications ranging from interpersonal relationships to economics. A survey of RM with an emphasis on its application to the online marketplace is presented in [7].

A related example can be found in [16], where reputation management is applied to the general area of "Electronic Communities". This work demonstrates the positive development of altruistic relationships in which the *trust* of other agents can be built up over time. This has close similarities to the application within a MRC, but [16] uses the trust to assess the quality of information from other agents. Here, we use this trust to determine if robots should be altruistic to one another, thereby improving individual performance over

the long-term.

III. PROBLEM DEFINITION

In this work, a Multi-Robot Community (*MRC*) is defined as a set of n robots that can interact through some shared workspace W :

$$MRC = \{r_1, r_2, \dots, r_n\}. \quad (1)$$

In this community, each robot r_i will have a set of L_i individual tasks to accomplish described by the possibly dynamic set:

$$T_i = \{t_{i1}, t_{i2}, \dots, t_{iL_i}\}. \quad (2)$$

A task t_{ij} can be of several types. Example task types include visiting a location in the workspace to obtain a sensor measurement (e.g. for surveillance, scientific sampling, mapping), manipulating some object in the workspace (e.g. picking up trash, delivering light), and moving to a desired location (e.g. for delivery tasks). Task types are assumed to have a corresponding definition of completion, with some quantifiable measure of completeness. Also, it is assumed that a particular tasks may be completed by any robots within the *MRC* that can fulfill the level of completeness on the task. Therefore, once tasks are assigned, each robot r_i plans a sequence of tasks S_i to complete, ideally in some order that maximizes the robot's individual fitness function:

$$S_i = \{t_{kn}, t_{lo}, \dots, t_{mp}\}. \quad (3)$$

Where indices k, l, m, n, o, p are arbitrary, reflecting the possibility that robot r_i 's task sequence S_i may include any of the n^{th} , o^{th} , or p^{th} tasks belonging to any of the k^{th} , l^{th} , or m^{th} robots within the community.

A. The Fitness Function

To characterize the performance of individual robots, we define a fitness function F_i , as the difference between the rewards R_i gained from tasks in T_i being completed and the costs C_i incurred to complete such tasks. If a robot completes only those tasks assigned to it, its fitness function would be:

$$F_i = \sum_{t_{io} \in T_i} R_i(t_{io}) - \sum_{t_{io} \in T_i} C_i(t_{io}). \quad (4)$$

However, if robots can perform each other's tasks, then the fitness function generalizes to the rewards gained from all tasks in T_i being completed minus the costs incurred completing all tasks in S_i :

$$F_i = \sum_{t_{io} \in T_i} R_j(t_{io}, S_j) - \sum_{t_{kp} \in S_i} C_i(t_{kp}, S_i). \quad (5)$$

It is important to note that the rewards for having a task completed will be dependent on which robot is completing the task. This is highlighted in (5) by the fact that the fitness for robot i is based on how well robot j completes robot i 's task t_{io} to produce reward R_j . Consider a surveillance task where a robot must obtain images from different locations in

an environment. The reward could be based on the quality of the image (e.g. pixel resolution) so that if robot 0's camera obtains a 640x480 image and robot 1's camera obtains a 1280x960 image, the resulting rewards for obtaining the same image would satisfy $R_1 > R_0$.

Furthermore, costs to complete tasks are dependent on the robot completing the task. The fitness for robot i is dependent on the cost C_i incurs on itself when accomplishing a task t_{kp} that belongs to the k^{th} robot.

It is also noted that both rewards R_j and costs C_i in (5) are functions not just of the tasks, but of the task sequences. For example, if robot 2 is to complete task t_{43} , it will be required to move to a new location in the workspace. The additional energy cost incurred in moving to this location will clearly be a function of the robot's current path dictated by visiting the locations of other tasks in the sequence S_2 .

B. Community Classes

Generally, each robot will have altruistic relationships with some of the robots in the *MRC* and will act selfishly toward the others. Thus, the general case will involve some altruistic imbalance in the community. Yet, to facilitate analysis and discussion, the following classes will be used:

1) *Balanced Communities*: In balanced communities, each robot r_i , $i = 1..n$ has the capability to perform altruistic actions for all other robots r_j , $j = 1..n, j \neq i$ in the robot community. Fig. 3a provides an example of a balanced robot community.

2) *Unbalanced Communities*: In unbalanced communities, each robot r_i , $i = 1..n$ has the capability to perform altruistic actions for β other robots in the robot community, where $\beta < n - 1$.

Such communities occur when some robots simply cannot accomplish other robot's tasks. Fig. 3b provides an example of an unbalanced community in which each robot has altruism towards only two of the four other robots.

3) *Maximal Unbalanced Communities*: In maximal unbalanced communities, each robot r_i , $i = 1..n$ has the capability to perform altruistic actions for only one other robot r_j in the community, and r_j cannot perform altruistic actions for r_i . The case involving 2 robots and one UAV in Section I is one such example.

This scenario exemplifies a case of maximal unbalanced altruistic relationships and no pairwise relationship can

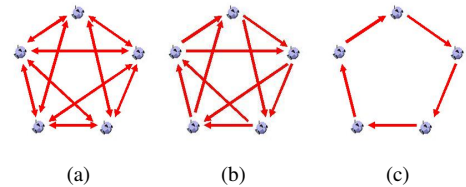


Fig. 3. Community classes as graphs. (a) The balanced class occurs when each edge is bi-directional. (b) The unbalanced class presents itself when only one directed edge connects any two particular robots. In (c), the maximal unbalanced class occurs when each robot has an indegree of 1 and an outdegree of 1, with each edge to a different robot in the *MRC*.

evolve. In graph theory terms, when the altruism graph becomes a single-cycle connected graph the MRC has entered this case (see Fig. 3c). In order for altruism to exist in this situation, the robots must allow some form of reciprocity toward the community or else allow the MRC to disband into disjoint sets of non-interacting robots.

IV. TASK ALLOCATION USING ALTRUISTIC BIDDING

Proposed is an auction-based bidding scheme for MRC. Each robot r_j can auction any of its own tasks from T_j . Other robots can *choose* to bid on robot r_j 's tasks, thus acting in an altruistic manner if they win the bid and complete the task. This choice to bid is based on the level of altruism, termed α , the robot r_i may have towards the robot r_j :

$$bid_i = \begin{cases} \sum Rewards - \sum Costs & \text{if } \alpha > C_i \\ -\infty & \text{else} \end{cases}. \quad (6)$$

Where C_i is the additional cost incurred by the altruistic robot in order to complete the other robot's task. The robot with the highest bid will win the auction and be assigned the task. This auction is conducted for each task in the MRC, however, tasks are auctioned and completed in epochs, or batches. During each epoch the robot cost functions are reset to zero. The tasks may repeat or change between epochs, although they remained the same for this analysis.

A. Altruistic Controllers

The level of altruism used during bidding is updated at the end of each epoch using a linear controller. The goal is to set the level of altruism to match the reciprocating robot's altruism. Two different controllers of altruism are proposed. To be precise, we use the two following definitions, one for each controller:

Definition 1: One-to-One The level of altruism robot r_i has towards robot r_j is $\alpha_{ij} \in [0, \infty]$, which equals the decrease in fitness robot i is willing to incur in completing robot j 's task (see Fig. 1a).

Definition 2: One-to-Many The level of altruism robot r_i has towards a community *MRC* is $\alpha_{iw} \in [0, \infty]$, which equals the decrease in fitness robot i is willing to incur in completing tasks belonging to any robot from a subset of the *MRC* (see Fig. 1b).

Each of the two proposed controllers model altruism differently, yet both controllers update based on a standard discretized linear time-invariant state space model:

$$\alpha^{t+1} = A\alpha^t + B\mathbf{u}^t. \quad (7)$$

At this stage of research, A and B are set to 1. By setting \mathbf{u} as a proportional controller, the update step for each individual altruistic value variable can be:

$$\alpha^{t+1} = \alpha^t + K(\alpha_{reciprocal}^t + \epsilon - \alpha^t). \quad (8)$$

The purpose of this controller is to match the level of work (i.e. altruistic actions) a robot is willing to do for another robot, or set of robots, to the level of work reciprocated,

termed $\alpha_{reciprocal}$ (see Fig. 4). The ϵ allows for a small amount of work that the robot is willing to do beyond the reciprocal work, allowing α to grow over time and K is the control gain constant. Each of the two different controllers use (8), but have different meaning for α , $\alpha_{reciprocal}$, and ϵ .

V. ONE-TO-ONE ALTRUISTIC CONTROLLER

This controller applies the theory of reciprocal altruism on a strictly pairwise basis. Relationships use only the reciprocal altruism of the other robot when updating their respective level of altruism. The controller protects against selfish robots and is guaranteed to increase the level of altruism over time when the other robot reciprocates. See [6] for the stability analysis and extensive treatment of this controller.

A. One-to-One Control Law

In the one-to-one control strategy, each robot r_i will attempt to set its altruistic level α_{ij} towards another robot r_j to be that which the other robot has toward it (α_{ji}). The controller follows the proportional control law:

$$\alpha_{ij}^{t+1} = \alpha_{ij}^t + K(\alpha_{ji}^t + \epsilon - \alpha_{ij}^t). \quad (9)$$

The control gain $K > 0$ determines the rate at which α_{ij} approaches the desired value of $\alpha_{ji} + \epsilon$. The first term in this desired value is the reciprocal altruism that robot r_j has towards robot r_i . The second term, $\epsilon > 0$, indicates how much more altruistically robot r_i will act towards robot r_j .

B. One-to-One Bidding

Each robot r_i will, or may, be given an opportunity to bid on each of r_j 's tasks based on the auctioning protocol for the MRC. Robot r_i , using the one-to-one controller, will bid using the following bidding law:

$$b_{ij} = \begin{cases} \sum R_i - \sum C_i & \text{if } \alpha_{ij} > C_i \\ -\infty & \text{else} \end{cases}. \quad (10)$$

This law allows for a certain level of altruistic action because if robot r_i wins the auction it will incur some cost, which may not be recouped from robot r_j .

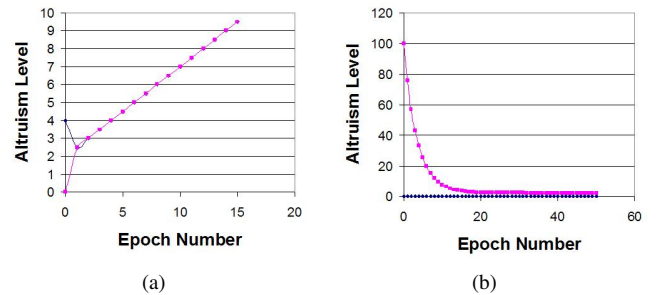


Fig. 4. Example of reciprocal altruism and response to selfishness. (a) The α values should grow when reciprocal altruism exists. (b) The α values for an (initially) altruistic robot decrease when paired with a selfish robot. Note: specific slopes and steady state values depend on the control law parameters. Each timestep is an epoch.

VI. ONE-TO-MANY ALTRUISTIC CONTROLLER

This controller applies the theory of reciprocal altruism from one robot to a *set* of robots. The relationships form between individual robots and the set of all other robots in the MRC. The controller does not directly protect against isolated selfish robots, yet it does protect against a selfish set of robots. A robot r_i determines a set of robots to be selfish when no robot in the set offers non-zero altruism towards r_i .

A. One-to-Many Control Law

In the one-to-many control strategy, each robot r_i will try to set its altruistic level α_{iw} towards the world (the ‘w’ signifies the set of robots in the MRC or world) to be that which the set of robots has toward it (α_{wi}). The total altruism that the robots in the MRC has toward robot r_i equates to:

$$\alpha_{wi} = \sum_{j=1}^n \alpha_{ji}, \quad j \neq i. \quad (11)$$

Where each of the values α_{ji} in (11) represent the altruistic level of robot r_j toward robot r_i . This leads to the proportional control law to track α_{iw} to α_{wi} :

$$\alpha_{iw}^{t+1} = \alpha_{iw}^t + K(\alpha_{wi}^t + \xi\epsilon - \alpha_{iw}^t). \quad (12)$$

Where the value of ξ equals the number of robots to which robot r_i applies the one-to-many controller. In other words, for a MRC of n robots, there are $n - (\xi + 1)$ robots that robot r_i cannot be altruistic towards. For example, consider a MRC of 7 robots. If robot A has no altruistic relationship to 4 robots (possibly due to inability to complete tasks), but uses the one-to-many for the 2 remaining robots, then $\xi = 2$.

Because α_{iw} contains the altruism of all ξ robots, the altruism level for each of these robots must be averaged:

$$\alpha_{ij} = \frac{1}{\xi} \alpha_{iw}, \quad j \neq i. \quad (13)$$

B. One-to-Many Bidding

Each robot r_i will, or may, be given an opportunity to bid on each of r_j 's tasks based on the auctioning protocol for the MRC. Robot r_i , using the one-to-many controller, will bid using the following bidding law:

$$b_{ij} = \begin{cases} \sum R_i - \sum C_i & \text{if } \frac{1}{\xi} \alpha_{iw} > C_i \\ -\infty & \text{else} \end{cases}. \quad (14)$$

C. Stability of the One-to-Many Controller

In the one-to-many control strategy, each robot r_i will try to set its altruistic nature α_{iw} towards a community of n robots as shown in (12). Consider the resulting state model of the altruistic relationship between robot r_i and the remaining $n - 1$ robots in C .

$$\begin{pmatrix} \alpha_{1w} \\ \alpha_{2w} \\ \dots \\ \epsilon \end{pmatrix}^{t+1} = \begin{pmatrix} 1 - K & \frac{K}{\xi} & \dots & K\xi \\ \frac{K}{\xi} & 1 - K & \dots & K\xi \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix} \begin{pmatrix} \alpha_{1w} \\ \alpha_{2w} \\ \dots \\ \epsilon \end{pmatrix}^t \quad (15)$$

The stability of the system can be evaluated through a coordinate transformation $e_i = \alpha_{iw} - \alpha_{wi}$ for $i = 1 \dots n$. Given this transformation, the system can be restated in error dynamics:

$$e_{iw}^{t+1} = \left(1 - K \frac{(1 + \xi)}{\xi}\right) e_{iw}^t. \quad (16)$$

Hence if $|1 - K \frac{1 + \xi}{\xi}| < 1$ the error dynamics will be stable and it follows that $(\alpha_{iw} - \alpha_{wi}) \rightarrow 0$ as $t \rightarrow \infty$.

If we consider a desired rate of change of altruism to be $K\epsilon$, then the error in the rate of change of altruism is $e_{i+n}^{t+2} = (\alpha_{wi}^{t+1} - \alpha_{wi}^t) - K\epsilon$ and it can be shown that as $t \rightarrow +\infty$, then $e_{i+n} \rightarrow e_{iw} \rightarrow 0$. More explicitly, the rate of change of altruism $(\alpha_{iw}^{t+1} - \alpha_{iw}^t)$ stabilizes to $K\epsilon$. Thus, for gain conditions $0 < K < 2 \frac{\xi}{1 + \xi}$, the mutual altruisms α_{iw} and α_{wi} will both match each other and grow over time in a stable fashion.

VII. RESULTS

A. Simulation

A series of experimental simulations were run involving communities of five differential drive robots operating in a 8m x 8m 2D workspace. Each experiment consisted of 25 epochs (batches), where altruism value updates occur after each epoch. In each epoch, 40 tasks, each at a random location, were individually assigned to random vehicles. These vehicles are committed to the completion of the tasks assigned to it, but they may auction them off to the highest bidder. However, the bidding occurs at the beginning of each epoch only. Within these experiments, fitness functions F_i were calculated with rewards R_j being equal to 1 for each task completed and costs C_i being equal to additional euclidean path costs. The controller gains K were set to 0.25 and ϵ was set to 0.25. Experiments were run 500 times and path cost results below are the average over these 500 experiments.

Within these experiments, the two approaches to controlling altruism, one-to-one and one-to-many, were applied to the three different cases. These cases corresponded to the three different community classes described above: balanced, unbalanced and maximal unbalanced.

Fig. 5 shows the resulting path costs from these experiments. In the balanced community case (a), both the one-to-one and one-to-many demonstrated almost identical decreases in path cost as the altruistic relationships evolved (shown in Fig. 5a). Although not optimal in the MTSP sense, over time the balanced communities lead to the lowest path costs possible in the MRC market bidding scheme.

To demonstrate controller performance in an unbalanced community, all five vehicles were only allowed to act altruistically to 2 of the 4 other robots. As shown in Fig. 5b, the one-to-many controller reduces the path cost an additional 15% when compared to the one-to-one controller.

To demonstrate controller performance in a maximal unbalanced community, all five vehicles were only allowed to act altruistically to 1 other robot. As shown in Fig. 5c, the

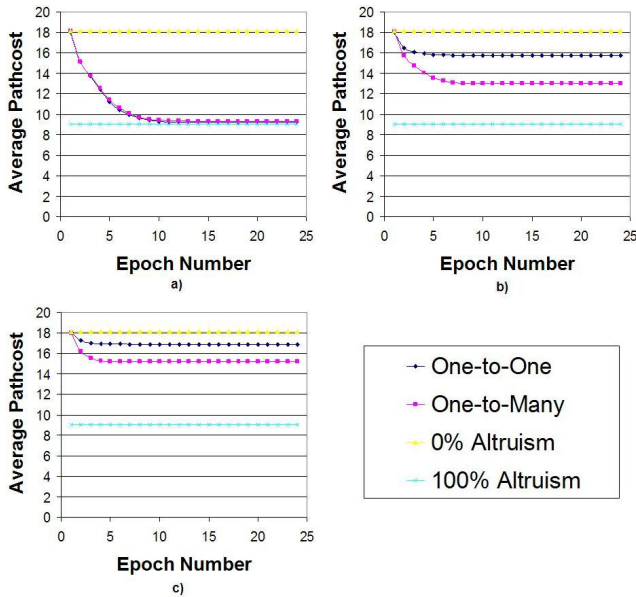


Fig. 5. Average path costs from simulations. Both the one-to-one and one-to-many controllers' path costs are shown for (a) balanced, (b) the unbalanced, and (c) the maximal unbalanced communities.

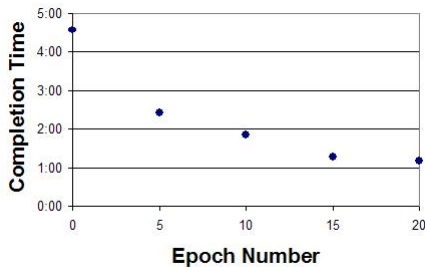


Fig. 6. Average task completion time on Dr. Robot X80 robots. Using a balanced community, the average completion time decreases as the altruism increases.

one-to-many controller continues to reduce the path cost over the one-to-one controller. The one-to-one controller still does have some path cost reduction because robots are willing to perform ϵ more work than the reciprocating robot.

B. On-Robot Tests

Preliminary tests were conducted using 4 Dr. Robot X80 mobile robots, with a total of 15 random task locations assigned per epoch. One-to-one altruism control was used in a balanced community. The average time required to complete all tasks in an epoch decreased considerably as time went by and the altruism increased (see Fig. 6).

VIII. CONCLUSION AND FUTURE WORK

This paper presents the idea of a Multi-Robot Community in which several robots sharing a common workspace are attempting to complete their individual tasks. It was shown that altruistic actions, where robots assist with each other's tasks, can lead to decreased path costs for individual robots.

Two controllers were compared in three different cases: balanced communities, unbalanced communities and maximal unbalanced communities. In all cases the new one-to-many controller did as good as or outperformed the one-to-one controller. However, it is postulated that the one-to-one controller will outperform the one-to-many controller when dealing with selfish robots that try to exploit the system. Previous work demonstrated the one-to-one controller's ability to protect against selfish robots, yet, more work is required to maintain protection when using the one-to-many controller.

Other future work includes developing better methods of estimating another robot's altruism achieved, dealing with situations with uneven task distributions (i.e. 5 tasks for one robot and 100 for another; in which case the 5-task robot may be disadvantaged by acting altruistically), adding uncertainty into task completions (i.e. probability of completion rather than binary), and further testing on real robots. In addition, a controller consisting of a weighted average of the controllers presented here may deal with their limitations.

REFERENCES

- [1] Agah A, Bekey GA (1997) Tropism-based cognition: a novel software architecture for agents in colonies. *Journal of Experimental and Theoretical Artificial Intelligence* 9: 393-404.
- [2] Agah A, Bekey GA (1997) Phylogenetic and ontogenetic learning in a colony of interacting robots. *Autonomous Robots* 4:85-100.
- [3] Arkin RC, Balch T (1998). Cooperative multiagent robotic systems, in *Artificial Intelligence and Mobile Robots*. D. Kortenkamp, R. P. Bonasso and R. Murphy, AAAI Press/MIT Press: 277-295.
- [4] Brooks R. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation* RA-2: 14-23.
- [5] Chen PA, Kempe D (2008) Altruism, Selfishness, and Spite in Traffic Routing. In *Proceedings of EC*.
- [6] Clark CM, Morton R, Bekey GA (2008) Altruistic Relationships for Optimizing Task Fulfillment in Robot Communities, *Proc. of Distributed Autonomous Robot Systems (DARS 08)*.
- [7] Dellarocas C (2006) Reputation Mechanisms. *Handbook on Information Systems and Economics*. Hendershott T (ed). Elsevier Publishing, 154-165.
- [8] Goldberg D, Cicirello V, Dias MB, Simmons R, Smith S, Stentz A (2003) Task Allocation Using a Distributed Market-Based Planning Mechanism. *International Conference on Intelligent Agents*.
- [9] Lucidarme P, Simonin O, Liegeois A (2002) Implementation and evaluation of a satisfaction/altruism based architecture for multi-robot systems. *Proc. IEEE Int. Conf. on Robotics and Automation* 1007-1012.
- [10] Mataric M (1997) Behavior based control: Examples from navigation, learning and group behavior. *Jour. of Experimental and Theoretical Artificial Intelligence* 9(2-3): 67-83.
- [11] Nowak MA, May RM, Sigmund K (1995) The arithmetics of mutual help. *Scientific American* 272(6):76-81.
- [12] Parker LE (1998) ALLIANCE: An architecture for fault tolerant multirobot cooperation. *IEEE Trans. on Robotics and Automation* 14(No. 2): 220-240.
- [13] Simonin O, Ferber J (2000) Modeling self-satisfaction and altruism to handle action and reactive cooperation. *SAB'00 Proceedings*: 314-323.
- [14] Sofge D, Schultz A, De Jong K (2002) Evolutionary Computational Approaches to Solving the Multiple Traveling Salesman Problem Using a Neighborhood Attractor Schema. *Applications of Evolutionary Computing - EvoWorkshops*.
- [15] Trivers R (1971) The evolution of reciprocal altruism. *Quarterly Review of Biology* Vol 46, 35-57.
- [16] Yu B, Singh MP (2000) A Social Mechanism of Reputation Management in Electronic Communities. *Cooperative Information Agents (CIA) 2000*. Springer-Verlag, Berlin Heidelberg 154-165.
- [17] McElreath A, Boyd R (2007) *Mathematical Models of Social Evolution: A guide for the perplexed*. University of Chicago Press.