Trust Networks in Multi-Robot Communities

Kevin T. Mantel and Christopher M. Clark

Abstract— **It has been shown that it can be beneficial for robots, which share a common workspace but do not share common goals, to work together and help each other complete their tasks. However, current systems designed to facilitate such actions can be manipulated by dishonest robots seeking to maximize their own benefit. The successful implementation of a system that protects against such activity could benefit groups of robots working near each other, even if the robots are deployed by different and possibly competing organizations. To address this issue, this paper develops and analyzes a distributed trust estimation framework that allows robots to estimate the trustworthiness of other robots in the community and to use this information to determine whether they should cooperate with the same robots. This framework is applied to a specific multi-robot exploration problem, involving the determination of cell types in a discretized workspace. Results show that a team of robots can explore more quickly when other trustworthy robots are present, without sacrificing performance when untrustworthy robots are also present.**

I. Introduction

Cooperation within multi-robot systems has seen much attention in the research community due to the potential for increased performance over single robot systems when applied to tasks including search, exploration, spatio-temporal sampling, and construction. The majority of the cooperative robot systems developed assume that robots will inherently be able to trust one another. This is a fair assumption in systems deployed by a single operator. However, as robots become ubiquitous and are deployed in a workspace that is shared with other robots that are deployed by other users, there is no reason to assume that all robots within the workspace are trustworthy. The possibility for robots to compete and misinform one another exists.

This work aims to develop a trust network framework that promotes cooperation between trustworthy robots. The ultimate goal is to improve robot performance (e.g. minimizing task completion time) through cooperation with previously unknown robots, despite the presence of robots that aim to deceive and take advantage of trustworthy robots. Presented below is a framework that accomplishes this goal, with a proven ability to estimate the trustworthiness of robots, and demonstrated performance gains in a multi-robot exploration task.

II. Background

There has been much work done in reputation management in non-robotic areas; for example, a user on the online auction website, eBay, has a rating based on the votes of other users who have shared interactions. [1], [2] and [3] both discuss reputation management in product marketplaces, and incorporate some communal information in a user's rating of other users. Peng et al. [4] deal with this idea in a general sense; they use a use trust system, that updates trust based on quality of agent interactions, using communal estimates only when no other information is available, and weights decisions according to these trust values.

Reputation management is useful in other fields, including peer-to-peer networks [5] and in robotics. While much work has been done in the field of robotic multi-agent systems, concerning cooperation among teams of robots that are deployed together or have similar goals [6], reputation management has not been necessary. However, it is useful in related work, as in Morton, et. al. [7], which involves different teams in a Multi-Robot Community (*MRC*) working together. This work examines robots exhibiting altruism, or completing tasks for other robots without expectation of immediate reciprocation, as well as motion planning and task-allocation systems. This paper integrates these ideas, improving upon [7] by handling cases in which a group member attempts to provide false information or to improve the reputation of itself or a confederate.

III. A Framework for Trust Networks

Proposed is a Trust Network framework that incorporates reputation management into *MRC*s. Each robot uses its trust, modeled as a variable, of another robot in order to estimate the likelihood that a statement from the other robot is true. A Multi-Robot Community $MRC = \{r_1, r_2, ..., r_n\}$ is defined as a set of n robots that can communicate and interact through some shared workspace W, but are deployed by different operators and may have different objectives

It is assumed that each robot in the *MRC* runs a fixed-time control loop that continually listens for messages broadcasted from other robots. Before using information from these messages to improve task completion performance, the truth of the messages must be evaluated. It is also assumed that robots broadcast at minimum two types of messages. The first are *statement* messages which can either be true or false. The second are *trust* messages, which indicate one robot's estimate of another robot's likelihood to broadcast true statements.

Within the *MRC* then, let $\beta_j \in [0, 1]$ be the likelihood that any general statement communicated from robot r_j is true. One robot's *trust* of another robot is defined here as $\hat{\beta}_{i,j,t} \in [0, 1]$, which is robot r_i's estimate of β_j at time t. Also, let $z_{i,j} \in [0, 1]$ be the fraction of robot r_j's statements

K. Mantel is a student in the Dept. of Computer Science, Princeton University, Princeton, NJ 08544, USA kmantel@princeton.edu
C. Clark is with the faculty in the Engineering Dept. at Harvey Mudd College, Claremont, CA 91711, USA clark@hmc.edu

that are true, according to the sensory observations of robot r_i. Finally, denote $\hat{\beta}'_{i,j,t}$ as the value robot r_i *broadcasts* to other robots, which represents its estimate of the likelihood that r_j communicates truthful statements. To note, $\hat{\beta}'_{i,j,t} = \hat{\beta}_{i,j,t}$ only if robot r_i broadcasts truthful statements.

In evaluating the ability of robots within the *MRC* to estimate β_j values, the error $e_{i,j}$ is defined as the difference between actual and estimate β values, i.e. $e_{i,j,t} = \beta_j - \hat{\beta}_{i,j,t}$. Parameters that may vary between *MRC* experiments include the number of deceivers $m \leq n$, the magnitude ϵ of which a deceiver inflates its broadcasted truth, (i.e. $\epsilon_j = \hat{\beta}'_{j,j,t} - \beta_j$), the communication range R_C within which any two robots may communicate, the sensor range R_S within which one robot may observe another robot's state, and the convergence parameters τ_t and τ_β where an estimator of any β_j is said to have converged at time t_c if $|\hat{\beta}_{i,j,t} - \hat{\beta}_{i,j,avg}| < \tau_\beta$ for $t \in [t_c - \tau_t, t_c]$.

A. Trust Estimation

As robots within an *MRC* communicate statements to other robots, the trust values are estimated a using feedback control based approach that has guarantees on performance. At each time step, each robot's estimator uses two pieces of information, 1) the trust values $\hat{\beta}'_{i,j,t}$ broadcast from other robots, and 2) its personal observations $z_{i,j,t}$ of other robots likelihood to broadcast true statements. The following update rule for trust is proposed:

$$\hat{\beta}_{i,j,t+1} = \frac{1}{|C|} \sum_{k \in C} \hat{\beta}'_{k,j,t} + K \left(z_{i,j,t} - \frac{1}{|C|} \sum_{k \in C} \hat{\beta}'_{k,j,t} \right) \quad (1)$$

In (1), C is the set of all robots within communication range R_C of robot r_i, and K is a constant proportional gain. The first term averages the trust of all robots within R_C and is used as a predictive element. The second term uses the robot's observations as a corrective element.

Theorem 1: Given n robots implementing the proposed estimator in (1) with infinite communication range, infinite sensor range, perfect sensing, and gain constant $0 < K < 2$, the estimation error $e_{i,j}$ converges to $(K-1)/K \cdot \bar{\epsilon}_j$, where $\bar{\epsilon}_j$ is the average inflation of trust $\epsilon_{k,j,t} = \hat{\beta}'_{k,j,t} - \hat{\beta}_{k,j,t}$ communicated by robots.

Proof: The broadcasted trust values $\hat{\beta}'_{k,j,t}$ in the update rule from (1) can be isolated and replaced with the sum of trust inflation and actual trust estimate $\epsilon_{k,j,t} + \hat{\beta}_{k,j,t}$ for every robot k.

$$\hat{\beta}_{i,j,t+1} = K \cdot z_{i,j,t} + \frac{1-K}{n} \sum_{k=1}^{n} (\epsilon_{k,j,t} + \hat{\beta}_{k,j,t}) \quad (2)$$

The assumption of perfect sensing and infinite sensor range leads to the approximation $z_{i,j,t} \approx \beta_j$. Including this assumption, along with the simplification of $\bar{\epsilon} = \Sigma_{k=1}^{n} \epsilon_{k,j,t}/n$ yields:

$$\hat{\beta}_{i,j,t+1} \approx K \cdot \beta_j + \frac{1-K}{n} \sum_{k=1}^{n} \hat{\beta}_{k,j,t} + (1-K)\bar{\epsilon_j} \quad (3)$$

Now, define $\gamma_{i,j,t+1}$, which will be shown to converge to 0 over time.

$$\gamma_{i,j,t+1} = e_{i,j,t+1} - \frac{K-1}{K}\bar{\epsilon_j} = \left(\beta_j - \hat{\beta}_{i,j,t+1}\right) - \frac{K-1}{K}\bar{\epsilon_j}$$

Substituting $\hat{\beta}_{i,j,t+1}$ from (3):

$$\gamma_{i,j,t+1} = \beta_j - \left[K \cdot \beta_j + \frac{1-K}{n} \sum_{k=1}^{n} \hat{\beta}_{k,j,t} + (1-K)\bar{\epsilon_j} \right] - \frac{K-1}{K}\bar{\epsilon_j}$$

$$\gamma_{i,j,t+1} = \frac{1-K}{n} \sum_{k=1}^{n} \gamma_{k,j,t} \quad (4)$$

This is a linear, discrete-time dynamical system, and can be expressed in matrix form as:

$$G_{t+1} = TG_t$$

where

$$G_t = \begin{bmatrix} \gamma_{1,1} & \gamma_{1,2} & \cdots & \gamma_{1,n} \\ \gamma_{2,1} & \gamma_{2,2} & \cdots & \gamma_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{n,1} & \gamma_{n,2} & \cdots & \gamma_{n,n} \end{bmatrix}_t$$

and

$$T = \begin{bmatrix} \frac{1-K}{n} & \frac{1-K}{n} & \dots & \frac{1-K}{n} \\ \frac{1-K}{n} & \frac{1-K}{n} & \dots & \frac{1-K}{n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1-K}{n} & \frac{1-K}{n} & \dots & \frac{1-K}{n} \end{bmatrix}$$

The significant eigenvalue of this transition matrix T is $1-K$. So, for $0 < K < 2$ (that is, $|1-K| < 1$), $\gamma_{i,j}$ converges to 0 over time. Therefore, robot r_i will estimate β_j with an error $e_{i,j}$ that converges to $(K-1)/K \cdot \bar{\epsilon}_j$. ■

IV. Stationary Robots Experiments

Several experimental simulations were conducted in order to validate the claims made in the previous section and to examine how different variables affected our system. Each experiment consisted of $n = 5$ stationary robots positioned in a line in a virtual grid environment. Simulations were run for a maximum of 2000 iterations (time steps). Each robot r_j was assigned a constant $\beta_j \in [0, 1]$.

At each time step, each robot broadcasted a message stating its position within the grid environment. A robot r_j would broadcast its actual position with probability β_j, or an incorrect position with probability $1 - \beta_j$. Robots also broadcasted truth estimates $\hat{\beta}'_{i,j,t}$ $\forall j$, to enable truth estimation.

Observations of truth values were simulated using the following equation:

$$z_{i,j,t+1} = \begin{cases} \dfrac{1 + t \cdot z_{i,j,t}}{t+1} & \text{if } r_j\text{'s statement is observed true} \\ \dfrac{t \cdot z_{i,j,t}}{t+1} & \text{if } r_j\text{'s statement is observed false} \\ \dfrac{t \cdot z_{i,j,t}}{t+1} + \dfrac{\frac{1}{|C|}\sum_{k \in C} \hat{\beta}'_{k,j,t}}{t+1} & \text{if Distance}(r_j,\, r_i) > \text{sensor range} \end{cases} \tag{5}$$

where C is the set of all robots within communication range of robot r_i, and the *sensor range* is the maximum distance in grid cells that a robot can "see." After message exchange, robot trust estimates were updated using (1), ($K = 0.99$).

A. Case 1: No Deceivers

First, the ability to accurately estimate β_j values is validated when no deceivers are present, i.e. $\epsilon_j = 0$, $\forall j$. Infinite communication and sensor range are assumed. Each robot r_j's β_j is set to a different value according to $\beta_j = j/n$, i.e. $\beta_1 = .2$, $\beta_2 = .4$, etc.. Shown in Fig. 1a, the trust value estimates converge the actual values, as expected from Theorem 1.

Next, simulations with no deceivers and the same values of β_j as above were used, but sensor and communication range were varied. While no noticeable correlation between convergence time and communication range was observed, a distinct positive correlation between the sensor range and convergence time, was exhibited in Fig. 1b. At low sensor ranges, robots on opposite ends of the grid are not able to verify each other's statements, and must rely on other robots for their trust estimates. This leads to smaller changes in estimates at each step, because the observed value becomes the average shared estimate (see (5)).

Fig. 1c shows that in the no-deceiver case, the error is dependent on the sensor range. Intuitively, as more robots are able to verify statements being made, more robots should be able to have accurate estimates of the trust value. There is room for future work to develop a version of Theorem 1 that does not require the assumption of infinite sensor range.

B. Case 2: Single Deceiver

Next, a single robot r_d is itself attempting to deceive other robots by communicating a $\hat{\beta}'_{d,d,t} > \beta_d$. The effects of r_d's placement within the line of robots are investigated. For this experiment, each robot's truth fraction is set as before, $\beta_i = (i+1)/n$. In one experiment, the deceiver is located at the end of the line of $n = 5$ robots, at position P_1. In another experiment, the deceiver is located at the center of the line, at position P_3.

It can be observed that there is no clear relation between the communication range and the error. However, the sensor range significantly affects the error when the deceiver is on the end of the line, (see Fig. 2). This is due to a lack of "good" information entering the network in the form of observations. When the sensor range is high, there is no such deficiency. Appropriately, we see in Fig. 2 that the average error with a peripheral deceiver at sensor ranges of 4 or 5 is very similar to the error at any sensor range with the deceiver in the center. Also notable, the convergence time increases with sensor range.

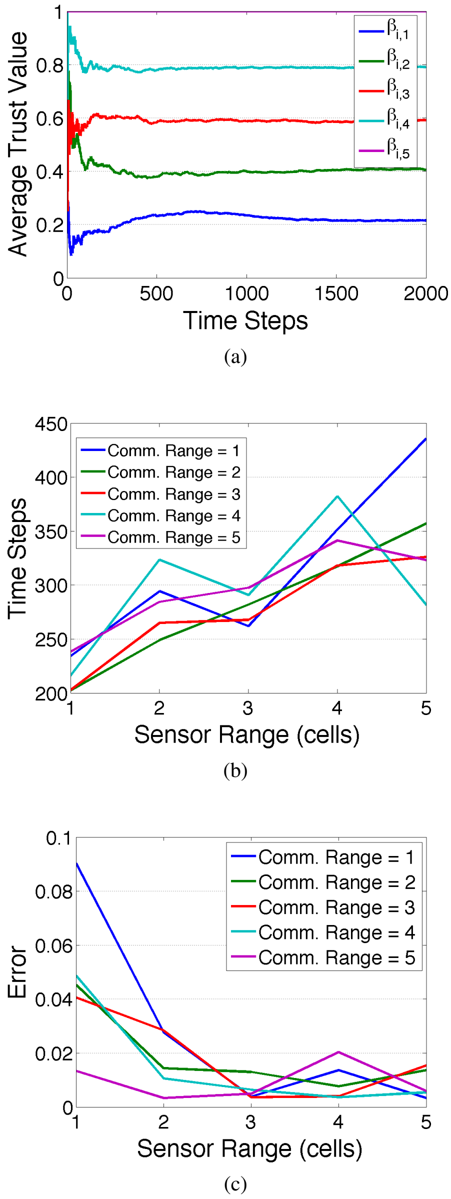


Fig. 1: No Deceivers Case: In (a), the average trust values are converging over time. Convergence Time when varying sensor range is plotted in (b). In (c), the average error after convergence is plotted as a function of sensor range.

C. Case 3: Multiple Deceivers

In this case, m robots attempt to deceive the *MRC* by artificially boosting a single robot r_d's trust value, e.g. $\hat{\beta}'_{j,d,t} > \hat{\beta}_{j,d,t}$ for $j = 1..m$. Theorem 1 is tested for a variety of deception magnitudes $\epsilon_{j,d}$. Infinite communication and sensor range are assumed. The gain $K = 0.99$. The truth fractions β_i of all robots are set to i/n. The number of deceivers, m, was varied from 0 to n. Also, $\epsilon_{j,d}$ for $j = d, 1..m-1$ was varied in increments of 0.1 from 0

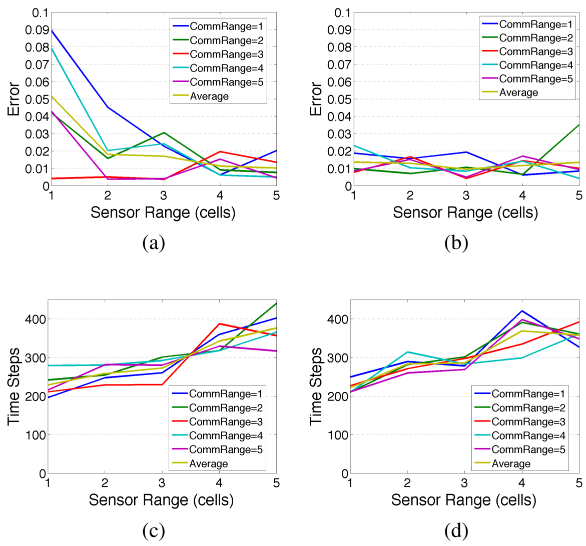


Fig. 2: Modifying the position of a single deceiver: In (a) and (b), the error as a function of sensor range is plotted for a deceiver at position P_1 and P_3 respectively. In (c) and (d), the convergence time as a function of sensor range is plotted for a deceiver at position of P_1 and P_3 respectively.

to 1. Robot r_d was chosen to be a peripheral robot, and any other deceiver positions were chosen randomly from the remaining robots.

In Fig. 3a, it is clear that the error is linearly dependent on both the number of deceivers and the deception size. This is precisely the relation that Theorem 1 provides. Fig. 3b shows the the average steady state difference between the experimental and theoretical value for $e_{i,d}$ as calculated from Theorem 1. It is clear that the difference has decreased to approximately zero.

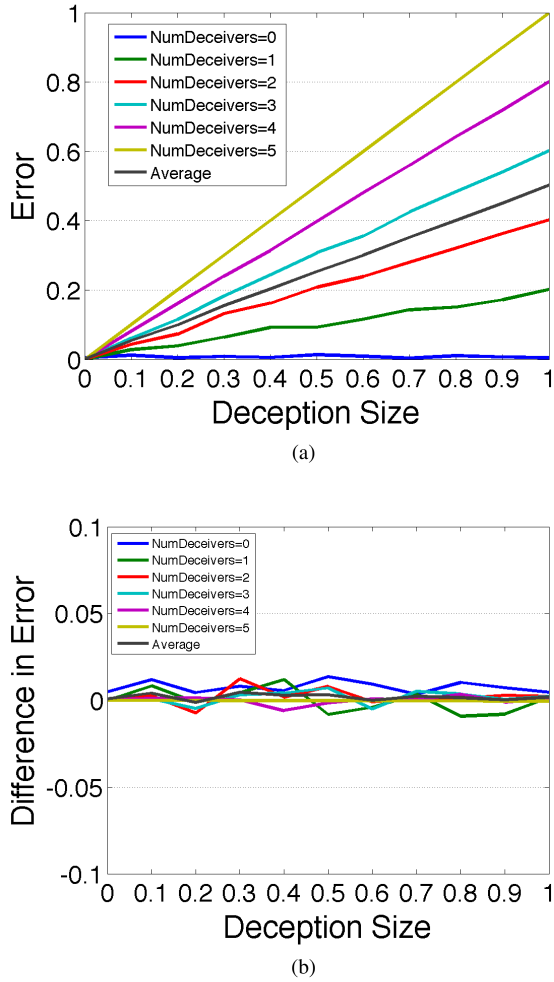


Fig. 3: Average trust estimation error varied over m and deception size (a). In (b), the difference between experimental and theoretical error.

V. Multi-Robot Exploration Experiments

In the following experiments, the trust network framework is applied to a typical multi-robot exploration problem to validate the improvement of team performance through cooperation when not all robots in an *MRC* can be trusted.

Simulations of a team $T \subseteq MRC$ of $p < n$ robots are tasked with cooperatively exploring an environment. The workspace W is evenly discretized into an o x o grid of cells. Within W, cells at each x, y location can be one of two types, i.e. $type(c_{x,y}) \in \{0, 1\}$ $\forall c_{x,y} \in W$. Fig. 4a illustrates a sample environment with two team members (blue) and two non-team members (red). Each cell is colored to indicate robot 0's belief of the cell type. Pure black and pure white demonstrate belief with high confidence of type 0 and type 1 respectively. Intermediate shades of grey show lower confidence. The exploration task requires the robots to determine the cell type of all cells by cooperatively visiting cells and locally observing their type. The term $O_i(c_{x,y})$ is defined to be robot r_i's average observation of cell type for $c_{x,y}$.

In applying the trust network framework to this application, each robot r_j broadcasts messages including statements about the cell types, $O'_j(c_{x,y})$, and truth value estimates $\hat{\beta}'_{i,j,t}$. Observations of *trust* values are set according to (6)), where cell $c_{x,y} \in I$ iff both robot r_i and robot r_j have observed $c_{x,y}$.

$$z_{i,j} = 1 - \frac{1}{|I|} \sum_{c_{x,y} \in I} \left| O_i(c_{x,y}) - O'_j(c_{x,y}) \right| \qquad (6)$$

To maximize *team* cooperation, all robots within the team communicate true statements with each other, i.e. $\beta_i = 1$ $\forall r_i \in T$. Non team members are not guaranteed to do so, and may use deception, i.e. $\beta_j \in [0, 1]$ $\forall r_j \notin T$. Moreover, team robots always communicate their best estimate of truth values with other team members, i.e. $\hat{\beta}'_{i,j,t} = \hat{\beta}_{i,j,t}$ $\forall r_i, r_j \in T$, while this is not necessarily true for non-team members, $\epsilon_j \in [0, 1]$ $\forall r_j \notin T$.

As robots broadcast their messages, they can update their individual belief of each cell's type. Specifically,

$B_i(type(c_{x,y}) = \tau)$ is defined as robot r_i's belief that the cell type is $\tau \in \{0, 1\}$, calculated using observations from many robots:

$$B_i(type(c_{xy}) = \tau) = 0.5 + 0.5 b_\tau sign(0.5 - \tau) \tag{7}$$

$$B_i(type(c_{xy}) = \neg\tau) = 1 - B_i(type(c_{xy}) = \tau) \tag{8}$$

$$b_\tau = 1 - \prod_{r_j \in R_\tau} (1 - 2\hat{\beta}_{i,j} \left|0.5 - O'_j((c_{xy})\right|) \tag{9}$$

In the above equations, R_τ is the set of all robots that have an observation of c_{xy} and that have average observation closer to τ than $\neg\tau$. That is, if $O\left(c_{xy}\right)_i \in [0, .5)$ then $r_i \in R_0$ and if $O\left(c_{xy}\right)_i \in (.5, 1]$ then $r_i \in R_1$.

In (9), the product is the likelihood that all robots in R_τ are incorrect in observing τ as the correct cell type. Subtracting this product from 1 yields the probability that at least one robot $r_j \in R_\tau$ is correct in observing τ as the correct cell type.

The simulation invokes an iterative algorithm that at each step updates the position of each robot, simulates inter-robot communication, updates each robot's belief of cell type, and updates trust values. Alg. 1 provides an overview.

Algorithm 1 Multi-Robot Simulation

if any robot team member is not finished **then**
  Allocate task cell to each robots
  for all robots **do**
    Move
    Observe cells within range
    Share cell observations, trust values with other robots in range
    Update beliefs of all cells in the grid
    Update trust values of other robots based on new information
  end for
end if

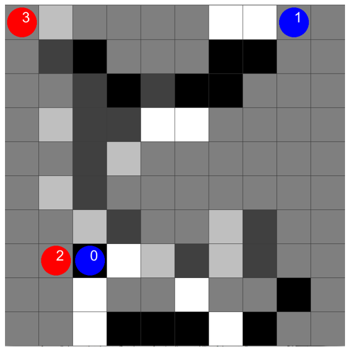
A team member r_i is defined to be unfinished if there is some cell $c_{x,y}$ for which $B_i(type(c_{x,y})) < 0.9$. If there exists any unfinished robots, a new task (i.e. desired cell to visit) is assigned to each robot whose previously assigned task has not been completed. A greedy approach to task assignment is used: visit the closest *unallocated* cell $c_{x,y}$ that has $B_i(type(c_{x,y})) < 0.9$. To note, the goal of this work is not to implement an optimal exploration algorithm, but demonstrate the usefulness of the trust network for a typical approach to exploration.

After each robot moves one cell toward its task cell, robots exchange information so they can subsequently update their beliefs of cell types and trust values. To note, the simulation assumes robots have infinite communication, but can only observe cell type for those cells within 1 cell of the robot's current cell.

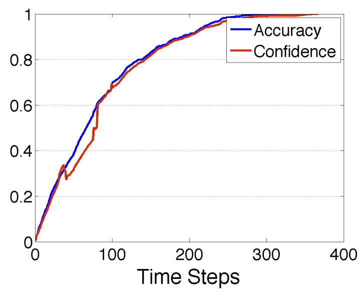
A. *Validating Convergence of Trust Values and Accuracy*

This experiment is used to evaluate the ability of the trust network to estimate trust values and ensure accurate cell type estimation by the end of the exploration task. The experiment involved $n = 8$ robots, only one of which was a team member. The grid was of size 40x40. The non-team robots have truth fractions chosen from the set $\{0/8, 1/8, 2/8, ..., 8/8\}$.

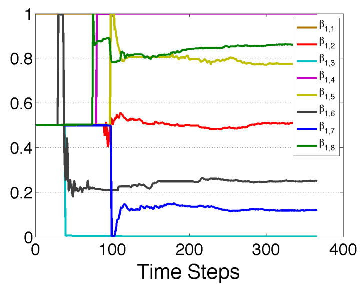
Fig. 4 shows data from the team robot r_0. The accuracy generally improves as time goes on, even though it receives incorrect information from non-team members with low β_j values who may be supplying false information. However, as time passes and trust values are estimated accurately, information from non-trustworthy robots is weighted less when calculating cell type beliefs.



(a)



(b)



(c)

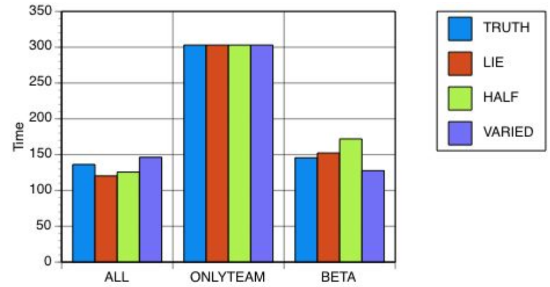
Fig. 4: Convergence of Trust Estimation: In (a), a visualization of the simulated workspace is shown. In (b), the progression of the team member's belief accuracy over time. In (c), the corresponding progression of the team member's trust values of other robots over time

B. Establishing Performance Improvements with the Trust Network

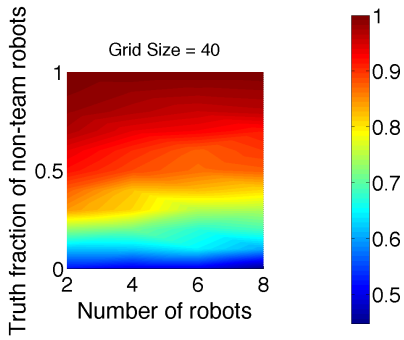
To demonstrate the benefits of the trust network when a team of robots is operating within a *MRC* with potential deceivers, three approaches to cooperative exploration are compared. In the first approach called "All", team robots trust non-team robots completely, i.e. $\hat{\beta}_{i,j} = 1 \; \forall r_i \in T, \forall r_j \in MRC$. In the second approach, called "Only Team", the team robots simply ignore non-team robots, i.e. $\hat{\beta}_{i,j} = 0 \; \forall r_i \in T, \forall r_j \notin T$. In the third approach, called "Beta", the team robots use the proposed trust network update system described above.

The experiments involved a total of 4 team and 4 non-team robots exploring a 40x40 grid. Four different truth value settings were implemented: 1) Truth, where all non-team robots always broadcast their observations truthfully, 2) Lie, where all non-team robots always broadcast incorrect observations, 3) Half, where all non-team robots broadcast truthful observations with probability 0.5, and 4) Varied, where non-team robots have truth fractions chosen from the set $\{0/8, 1/8, 2/8, ..., 8/8\}$.

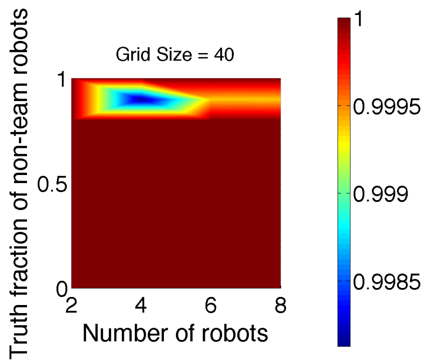
Fig. 5 illustrates the exploration time for the different trust value settings and different approaches to cooperation. The Only Team approach takes roughly twice as long to explore the workspace, which is expected considering the team robots do not use any information from non-team robots. The proposed trust network approach (i.e. Beta), takes only slightly longer than the All approach. However, in comparing the accuracy of these two approaches (Fig. 5b and c), there is a clear advantage to using the proposed trust network.



(a)



(b)



(c)

Fig. 5: Trust Network Performance: Exploration time of different trust schemes are shown in (a). In (b) and (c), the corresponding cell type accuracies are shown for the All and Beta approaches. Note differences in scale from (b) to (c).

VI. Conclusions

This work proposes a method for robots to cooperate with possibly untrustworthy robots sharing the same workspace. By continually monitoring and estimating the level of trust a robot can place in other robots, the information communicated from the other robots can be leveraged appropriately. The estimation of a robot's truthfulness is shown to converge to a bounded error, and the the performance benefits are clearly demonstrated in that accuracy of a typical exploration task can be maintained while decreasing exploration time - despite the presence of deceptive robots that provide false information.

References

[1] Sabater, J.; Sierra, C. (2002) Reputation and social network analysis in multi-agent systems, Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1, pp. 475-482

[2] Dellarocas, Chrysanthos N.; Dini, Federico; Spagnolo, Giancarlo. (2006) Designing Reputation (Feedback) Mechanisms. HANDBOOK OF PROCUREMENT, Nicola Dimitri, Gustavo Piga, Giancarlo Spagnolo, eds., Cambridge University Press.

[3] Yu, Bin; Singh, Munindar P. (2000). A Social Mechanism of Reputation Management in Electronic Communities. In Proceedings of Fourth International Workshop on Cooperative Information Agents. 154-165

[4] Peng, M., Xu, Z., Pan, S., Li, R., Mao, T. (2012). AgentTMS: A MAS Trust Model based on Agent Social Relationship. Journal Of Computers, 7(6), 1535-1542

[5] Mahapatra, A.; Tarasia, N.; Ajay, A.; Ray, S. (2011) A fuzzy approach for reputation management in Bittorrent P2P network, Electronics Computer Technology (ICECT), 2011 3rd International Conference on , vol.1, no., pp.360-364

[6] 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: 277295.

[7] Morton, R.D.; Bekey, G.A.; Clark, C.M. (2009) Altruistic task allocation despite unbalanced relationships within Multi-Robot Communities, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009.