Towards Adaptive Multi-Robot Coordination Based on Resource Expenditure Velocity: Extended Version

Dan Erusalimchik and Gal A. Kaminka
The MAVERICK Group
Computer Science Department
Bar Ilan University, Israel

Technical Report MAVERICK 2008/02

Abstract

In the research area of multi-robot systems, several researchers have reported on consistent success in using heuristic measures to improve loose coordination in teams, by minimizing coordination costs using various heuristic techniques. While these heuristic methods has proven successful in several domains, they have never been formalized, nor have they been put in context of existing work on adaptation and learning. As a result, the conditions for their use remain unknown. We posit that in fact all of these different heuristic methods are instances of reinforcement learning in a one-stage MDP game, with the specific heuristic functions used as rewards. We show that a specific reward function—which we call *Effectiveness Index* (EI)—is an appropriate reward function for learning to select between coordination methods. EI estimates the *resource-spending velocity* by a coordination algorithm, and allows minimization of this velocity using familiar reinforcement learning algorithms (in our case, Q-learning in one-stage MDP). The paper analytically and empirically argues for the use of EI by proving that under certain conditions, maximizing this reward leads to greater utility in the task. We report on initial experiments that demonstrate that EI indeed overcomes limitations in previous work, and outperforms it in different cases.

1 Introduction

This paper begins with a puzzle. In the research area of multi-robot systems, several researchers have reported on consistent success in using heuristic measures—which for the moment we call *coordination cost* measures—to improve loose coordination in teams. Specifically, Goldberg et al. [4], Zuluaga and Vaughan [16], and Rosenfeld et al. [12] all report that minimizing their respective coordination cost measures lead to improved performance.

However, while these heuristic methods has proven successful in several domains, they have never been formalized to a degree that allowed comparison with other methods. Nor have they been put in context of existing work on adaptation and learning. As a result, their optimality and the appropriate conditions for their use remain open questions.

We posit that in fact all of these different heuristic methods are instances of reinforcement learning in a one-stage MDP game [7], with the specific heuristic functions used as rewards. We further argue that the different coordination cost measures are all variations on central theme: Reducing the time and/or resources spent on coordination. These variations can be recast as reward functions within the MDP game.

We show that a specific reward function—which we call *Effectiveness Index* (EI)—is an appropriate reward function for learning to select between coordination methods. EI estimates the *resource-spending velocity* by a coordination algorithm, and allows minimization of this velocity using familiar reinforcement learning algorithms (in our case, Q-learning in one-stage MDP game).

The paper analytically and empirically argues for the use of EI by proving that under certain conditions, maximizing this reward leads to greater utility in the task. We report on initial experiments that demonstrate that EI indeed overcomes limitations in previous work, and outperforms it in different cases.

2 Related Work

Most closely related to our work is earlier work on measures of coordination effort. Rosenfeld et al. [12], presented a method that adapts the selection of coordination methods by multi-robot teams, to the dynamic settings in which team-members find themselves. The method relies on a measuring the resources expended on coordination, using a measure called Combined Coordination Cost (*CCC*). The adaptation is stateless, i.e., has no mapping from world state to actions/methods. Instead, the CCC is estimated at any given point, and once it passes pre-learned (offline learning) thresholds, it causes dynamic re-selection of the coordination methods by each individual robot, attempting to minimize the CCC.

Interference [4] is a closely related measure to CCC, and can be seen as a special case of it: It measures the amount of time spent on coordination. Zuluaga and Vaughan [16] presented an method called *aggression* for reducing interference in distributed robot teams, to improve their efficiency. During movement, multiple robots frequently interfere with each other. When such interference occurs, each of the robots demonstrate its own level of aggression such that the robot with the highest level becomes the winner, while the loser concedes its place. Zuluaga and Vaughan have shown that choosing aggression level proportional to the robot's task investment can produce better overall system performance compared to aggression chosen at random. This result is compatible with Rosenfeld et al.'s conclusions that reducing total resource spending in coordination is highly beneficial.

We formulate and generalize Rosenfeld et al.'s work in terms of reinforcement learning in single-state MDP game (MDG). Based on this generalized formulation, we are able to explain the empirically-observed success of Rosenfeld et al.'s method (as a special case), and suggest novel learning methods that do not require an off-line learning phase.

Indeed, the contribution of our work lies in the introduction of a general reward function for coordination (and only for coordination). This reward function minimizes *the velocity of resource expenditure*. In contrast, most investigations of reinforcement learning in multi-robot settings have focused on other mechanisms (e.g., modifying the basic Q-learning algorithm), and utilized task-specific reward functions. We briefly discuss these below. Two recent surveys are provided in [15, 6].

Balch [1] discusses considerations for task-oriented reward functions for reinforcement learning in multi-robot settings. He shows that the choice of reward function influences the behavioral diversity, and group performance in a variety of tasks, including foraging and soccer. Kok and Vlassis [9] discuss a technique for propagating rewards among cooperative robots, based on the structure of the dependence between the robots. However, they too assume that the reward function is given as part of the task.

Mataric[10] discusses three techniques for using rewards in multi-robot Q-learning: A local performance-based reward (each robot receiving reward for its own performance, and per its own goals), a global performance-based reward (all robots receive reward based on achievement of team goals), and a heuristic strategy referred to as shaped reinforcement. Shaped reinforcement, which was developed by Mataric, provides a heuristic function that combines rewards based on local rewards, global rewards and coordination interference of the robots. in contrast to these investigations, we explore general reward functions, based on minimize resource use, and use them in selecting between coordination behaviors, rather than individual behaviors.

Kapetanakis and Kudenko [7] present the FMQ learning algorithm. This algorithm is intended for coordination learning in one-stage MDP games. FMQ is a modified regular Q-Learning method for one-stage games and this modification is based on the Boltzman's strategy. They then examine how an robot that uses FMQ learning technique may influence other robot's effectiveness of learning, when the latter uses a simple Q-learning algorithm [8]. This method does not use communication or monitoring of other robot's action, but based on the assumption that all of the robots are getting the same rewards.

The Q-learning algorithm used in these works has no states, similarly to the proposed method, but Kapetanakis and Kudenko's works are concentrating on improving effectiveness of the learning algorithm and assume that rewards were pre-defined before and thus, the robot just has to discover them. In opposite, we concentrate on the method of reward determination by the robot. In the real world we do not have predefined rewards and especially when distinguishing between rewards from behaviors with the same goal is needed. Therefore, Kapetanakis and Kudenko's work, like a many other works of Reinforcement Learning is concentrated on Q-learning algorithm modification and assume pre-definition of the rewards, should be considered as a complimentary work instead of an alternative to ours.

3 Maximizing Social Utility by Limiting Coordination Costs

We first cast the problem of selecting coordination algorithms as a reinforcement learning problem (Section 3.1). We then introduce the effective index (EI) in Section 3.2. We then discuss the conditions underwhich maximizing it leads to improved task performance, and provide a proof in Section 3.3.

3.1 Coordination Algorithm Selection as an RL Problem

Multilateral coordination prevents and resolves conflicts among robots in a multi-robot system (MRS). Such conflicts can emerge as results for shared resource (e.g., space), or as a result of violation of joint decisions by team-members. Many coordination algorithms (protocols) have been proposed and explored by MRS researchers [3, 4, 11, 13, 14]. Not one method is good for all cases and group sizes [12]. However, deciding on a coordination method for use is not a trivial task, as the effectiveness of coordination methods in a given context is not known in advance.

We focus here on loosely-coupled application scenarios where coordination is triggered by conflict situations, identified through some mechanism (we assume the existence of such mechanism exists, though it may differ between domains; most researchers simply use a pending collision as a trigger). Thus the normal routine of an robot's operation is to carry out its primary task, until it is interrupted by an occurring or potentially-occurring conflict with another robot, which must be resolved by a coordination algorithm. Each such interruption is called *a conflict event*. The event triggers a coordination algorithm to handle the conflict. Once it successfully finishes, the robots involved go back to their primary task. Such scenarios include multi-robot foraging, formation maintenance (coordinated movement), and delivery.

Let $A = \{\ldots, a_i, \ldots\}, 1 \leq i \leq N$ be a group of N robots, cooperating on a group task that started at time 0 (arbitrarily) lasts upto time T (A starts working and stops working on the task together). We denote by $T_i = \{c_{i,j}\}, 0 \leq j \leq K_i$ the set of conflict events for robot i, where $c_{i,j}$ marks the time of the beginning of each conflict. Note that each robot i may have been interrupted a different number of time, i.e., K_i may be different for different robots. For notational uniformity, $c_{i,K_i+1} = T$, and $c_{i,0}$ is defined as time 0.

The time between the beginning of a conflict event j, and up until the next event, the interval $I_{i,j} = [c_{i,j}, c_{i,j+1})$, can be broken into two conceptual periods: The *active* interval $I^a_{i,j} = [c_{i,j}, t_{i,j})$ (for some $c_{i,j} < t_{i,j} < c_{i,j+1}$) in which the robot was actively investing resources in coordination, and the *passive* interval $I^p_{i,j} = [t_{i,j}, c_{i,j+1})$ in which the robot no longer requires investing in coordination; from its perspective the conflict event has been successfully handled, and it is back to carrying out its task. By definition $I_{i,j} = I^a_{i,j} + I^p_{i,j}$. We define the *total active time* as $I^a = \sum_i \sum_j I^a_{i,j}$ and the *total passive time* as $I^p = \sum_i \sum_j I^p_{i,j}$.

Our research focuses on a case where the robot has a nonempty set M of coordination algorithms to select from. The choice of a specific coordination method $\alpha \in M$ for a given conflict event $c_{i,j}$ may effect the active and passive intervals $I^a_{i,j}, I^p_{i,j}$. To denote this dependency we use $I_{i,j}(\alpha), I^a_{i,j}(\alpha), I^p_{i,j}(\alpha)$ as total, active and passive intervals (respectively), due to using coordination method α.

Using this notation, we can phrase the selection of coordination algorithms as determining a policy for selecting between different coordination methods among those in M. We denote a robot i's selection at conflict event $c_{i,j}$ as $\Pi_{i,j}$. A sequence of these selections, for all events $j \leq K_i$, is denoted by Π_i; this defines an individual coordination policy. The set of individual policies of all robots in A is marked Π.

Formally, we define the problem of coordination algorithm selection as a one-stage Markov Decision Process (MDP) game, with a limited set of actions (selectable algorithms), and an individual reward for each robot (player) [7]. Each robot tries to maximize its own reward. Typically, reward functions are given, and indeed most previous work focuses on learning algorithms that use the reward functions as efficiently as possible. Instead, we assume a very basic learning algorithm (a simple Q-Learning variant), and instead focus on defining a reward function. The learning algorithm we use is stateless:

$$Q_t(a) = Q_{t-1}(a) + \rho(R_t(a) - \gamma Q_{t-1}(a))$$

Where ρ is the learning speed factor, and γ is a factor of discounting.

3.2 Effectiveness Index

We call the proposed general reward for coordination *Effectiveness Index* (EI). Its domain independence is based on its using three intrinsic (rather than extrinsic) factors in its computation; these factors depend only internal computation or measurement, rather than environment responses.

The time spent coordinating. The main goal of a coordination algorithm is to reach a (joint) decision that allows all involved robots to continue their primary activity. Therefore, the sooner the robot returns to its main task, the less time is spent on coordination, and likely, the robot can finish its task more quickly. Thus, smaller I^a_i is better.

The frequency of coordinating. If there are frequent interruptions—even if short-lived—to the robot's task, in order to coordinate, this would delay the robot. We assume (and the preliminary results show) that good coordination decisions lead to long durations of non-interrupted work by the robot. Therefore, the frequency of coordination method's use is not less important, than the time spent on conflict resolving. Thus, larger $I^p_{i,j}$ is better.

The cost of coordinating. Finally, in addition to speed of conflict resolution and frequency of calling, careful resource spending is a very important factor for behavior selection. Short-lived, infrequent calls to an expensive coordination method will not be preferable to somewhat more frequent calls to very cheap coordination method. It is thus important to consider the internal resources used by the chosen method. We argue that such internal estimate of resource usage is feasible.

First, some resource usage is directly measurable. For instance, energy consumption during coordinated movement (e.g., when getting out of a possible collision) or communications (when communicating to avoid a collision) is directly measurable in robots, by accessing the battery device before and after using the coordination algorithm.

Second, resource usage may sometimes be analytically computed. For instance, given a the basic resource cost of a unit of transmission, the cost of using a specific protocol may often be analytically computed (as it is tied directly to its communication complexity in bits).

Finally, the most general way is in using of a resources manager with capability to monitor resource usage by components of the robot system. The description of such manager is beyond of this work, though we note in passing that such managers exist already for general operating systems.

Rosenfeld et al. [12] have defined CCC as the total cost of resources spent on resolving conflicts (re-establishing coordination) before, during, and after a conflict occurs. Their definition of the cost consisted of a weighted sum of the costs of different resources. We denote by U_i^C the utility of coordination, of robot i, of which the cost of coordination, denoted C_i^C is components. By definition, $CCC = C_i^C$. It can be broken into the costs spent on resolving all conflicts T_i, $C_i^C = \sum_j CCC_{c_{i,j}}$.

Let us use a cost function $cost_i(\alpha, t)$ to represent the costs due to using coordination method $\alpha \in M$ at any time t during the lifetime of the robot. The function is not necessarily known to us a-priori (and indeed, in this research, is not).

Using the function $cost_i(\alpha, t)$ we redefine the $C_{i,j}^C$ of a particular event of robot i at time $c_{i,j}$ to be:

$$C_{i,j}^C(\alpha) = \int_{c_{i,j}}^{c_{i,j+1}} cost_i(\alpha, t) \, \mathrm{d}t \tag{1}$$

We remind the reader that $C_{i,j}^C$ is defined as the costs of applying the coordination algorithm during the active interval $[c_{i,j}, t_{i,j})$ and the passive interval $[t_{i,j}, c_{i,j+1})$. However, the coordination costs during the passive interval are zero by definition.

$$\begin{aligned} C_{i,j}^C(\alpha) &= \int_{c_{i,j}}^{t_{i,j}} cost_i(\alpha, t) \, \mathrm{d}t + \int_{t_{i,j}}^{c_{i,j+1}} cost_i(\alpha, t) \, \mathrm{d}t \\ &= \int_{c_{i,j}}^{t_{i,j}} cost_i(\alpha, t) \, \mathrm{d}t \end{aligned} \tag{2}$$

We define the *Active Coordination Cost* (ACC) function for robot i and method α at time $c_{i,j}$, that considers the *active time* in the calculation of coordination resources cost:

$$ACC_{i,j}(\alpha) = \int_{c_{i,j}}^{t_{i,j}} 1 + cost_i(\alpha, t) \, \mathrm{d}t \tag{3}$$

We finally define Effectiveness Index of a particular event of robot i at time $c_{i,j}$ due to using coordination method $\alpha \in M$:

$$EI_{i,j}(\alpha) = \frac{ACC_{i,j}(\alpha)}{I_{i,j}} = \frac{\int_{c_{i,j}}^{t_{i,j}} 1 + cost_i(\alpha, t) \, \mathrm{d}t}{I_{i,j}^a + I_{i,j}^p} \tag{4}$$

That is, the effectiveness index (EI) of an algorithm α during this event is the velocity by which it spends resources during its execution, amortized by how long a period in which no conflict occurs. Since greater EI signifies greater costs, we typically put a negation sign in front of the EI, to signify that greater velocity is worse; we seek to minimize resource spending velocity.

In this proposal we present the simple single-state Q-learning algorithm (see Algorithm 3.2) which uses the EI to select between coordination methods.

Algorithm 1 Stateless EI-Based Adaption

```
Input: $CBO$, a set of coordination algorithms
Input: $RES$, a set of resources available for coordination

Require: $\beta$, rate of exploration vs exploitation
Require: $\rho$ learning speed factor
Require: $\gamma$ learning discount factor
for all $b \in CBO$ do
    $Q(b) \leftarrow 0$
{We assume the robot starts with a conflict situation}
while robot is active do
    $r \leftarrow random([0,1])$
    if $r < \beta$ then
        $best \leftarrow random(b \in CBO)$
    else
        $best \leftarrow argmax_{b \in CBO}(Q(b))$
    $start_{time} \leftarrow CurrentTime()$
    Execute $best$ {Record $CCC_{best}$}
    $t_a \leftarrow CurrentTime() - start_{time}$
    $start_{time} \leftarrow CurrentTime()$
    WAIT FOR CONFLICT EVENT
    $t_p \leftarrow CurrentTime() - start_{time}$
    $EI_{best} \leftarrow -\frac{\int_{c_{i,j}}^{t_{i,j}} 1+cost_i(\alpha,t)\ \mathrm{d}t}{t_a+t_p}$
    $Q(best) \leftarrow Q(best) + \rho(EI_{best} - \gamma Q(best))$
```

3.3 An Analytical Look at EI

We now turn to discuss the conditions underwhich an EI-minimizing policy Π will lead to greater team performance on its group task.

Preliminaries. We use the following notations in addition to those already discussed. First, we denote by U_i is the individual utility of robot i. U_i^T marks its utility due to executing the task (*task utility*), and U_i^C marks its utility due to being coordinated with others at a conflict situation (*coordination utility*): $U_i = U_i^T + U_i^C$. Each such utility value can be broken into gains G and costs C: $U_i^T = G_i^T - C_i^T$ and $U_i^C = G_i^C - C_i^C$. The social utility U is the sum of all individual utilities of the robots: $U = \sum_{i=1}^{N} U_i$.

To maximize this sum, the robot can invest effort in maximizing the utility from the task, and/or the utility from coordination. In the same way, to maximize the social utility of the team, each robot can invest effort in maximizing the its own utility and/or the teammates' utility. We are interested in task-independent reward functions, and thus focus our attention on maximizing utility from coordination (social utility).

Let us use a function $cgain_i(\alpha, t)$ to denote the coordination gain at any time t during the lifetime of the robot i that uses method α. When a robot is handling a conflict event, it is not gaining anything from coordination (in fact, it is investing effort in re-establishing coordination). Thus, the $cgain_i(\alpha, t)$ function can be defined as a step function

$$cgain_i(\alpha, t) = \begin{cases} 0 & \text{robot } i \text{ in a conflict situation} \\ 1 & \text{other} \end{cases} \tag{5}$$

Using this function, we redefine the $G^T_{i,j}$ of a particular event of robot i at time $c_{i,j}$ to be:

$$G^C_{i,j}(\alpha) = \int_{c_{i,j}}^{c_{i,j+1}} cgain_i(\alpha, t)\ \mathrm{d}t = \int_{c_{i,j}}^{t_{i,j}} cgain_i(\alpha, t)\ \mathrm{d}t + \int_{t_{i,j}}^{c_{i,j+1}} cgain_i(\alpha, t)\ \mathrm{d}t$$

$$= 0 + \int_{t_{i,j}}^{c_{i,j+1}} cgain_i(\alpha, t)\ \mathrm{d}t = \int_{t_{i,j}}^{c_{i,j+1}} cgain_i(\alpha, t)\ \mathrm{d}t = \int_{t_{i,j}}^{c_{i,j+1}} 1\ \mathrm{d}t = I^p_{i,j}(\alpha) \tag{6}$$

Now, we can define two evaluation functions of coordination policy.

- *Social Utility* of team by using policy Π

$$U(\Pi) = \sum_i^N \sum_j^{K_i} U_{i,j}(\Pi_{i,j}) = \sum_i^N \sum_j^{K_i} U^T_{i,j}(\Pi_{i,j}) + G^C_{i,j}(\Pi_{i,j}) - C^C_{i,j}(\Pi_{i,j}) \tag{7}$$

- *Social ACC* of team by using policy Π

$$ACC(\Pi) = \sum_i^N \sum_j^{K_i} ACCc_{i,j}(\Pi_{i,j}) \tag{8}$$

Based on the above, we would ideally want to show that (1) minimizing EI with each event leads to improved coordination utility for the team, and that (2) this, in turn, leads to improved overall task performance of the team (greater social utility). The first part is in some sense already given, when we use the FMQ framework. As long as its conditions hold, we can expect individual rewards to be maximized (i.e., the coordination utility will be greater individually). However, the second part is more challenging.

It is possible to show, that if the coordination costs for the team are minimized (i.e., the sum of coordination costs for all robots is minimized), then the coordination utility of the team is greater (Lemma 1).

Lemma 1. *The* Coordination Utility *for policy* Π' *is better than* Coordination Utility *for policy* Π'' *if* Social ACC *for* Π' *is lower than* Social ACC *for* Π''.

$$ACC(\Pi') < ACC(\Pi'') \implies U^C(\Pi') > U^C(\Pi'')$$

Proof. Let us consider the two policies Π' and Π''. The ratio of Social ACC after time T (when the task is completed) with both policies is

$$ACC(\Pi') < ACC(\Pi'') \tag{9}$$

$$\sum_i^N \sum_j^{K_i} \int_{c_{i,j}}^{t_{i,j}} 1 + cost_i(\Pi'_{i,j}, t) \; \mathrm{d}t < \sum_i^N \sum_j^{K_i} \int_{c_{i,j}}^{t_{i,j}} 1 + cost_i(\Pi''_{i,j}, t) \; \mathrm{d}t \tag{10}$$

For both sides of equation (10), the following holds:

$$\begin{aligned} \int_{c_{i,j}}^{t_{i,j}} 1 + cost_i(\Pi_{i,j}, t) \; \mathrm{d}t &= \int_{c_{i,j}}^{t_{i,j}} 1 \; \mathrm{d}t + \int_{c_{i,j}}^{t_{i,j}} cost_i(\Pi_{i,j}, t) \; \mathrm{d}t \\ &= I^a_{i,j}(\Pi_{i,j}) + C^C_{i,j}(\Pi_{i,j}) \end{aligned} \tag{11}$$

So, we can rewrite equation (10) by using equality (11)

$$\sum_i^N \sum_j^{K_i} I^a_{i,j}(\Pi'_{i,j}) + C^C_{i,j}(\Pi'_{i,j}) < \sum_i^N \sum_j^{K_i} I^a_{i,j}(\Pi''_{i,j}) + C^C_{i,j}(\Pi''_{i,j}) \tag{12}$$

By using definitions of $I^a(\Pi)$ and $C^C(\Pi)$

$$I^a(\Pi) = \sum_i^N \sum_j^{K_i} I^a_{i,j}(\Pi_{i,j}) = T - I^p(\Pi)$$

$$C^C(\Pi) = \sum_i^N \sum_j^{K_i} C^C_{i,j}(\Pi_{i,j})$$

we can represent equation (12) as

$$\begin{aligned} I^a(\Pi') + C^C(\Pi') &< I^a(\Pi'') + C^C(\Pi'') \\ -I^a(\Pi') - C^C(\Pi') &> -I^a(\Pi'') - C^C(\Pi'') \\ T - I^a(\Pi') - C^C(\Pi') &> T - I^a(\Pi'') - C^C(\Pi'') \\ I^p(\Pi') - C^C(\Pi') &> I^p(\Pi'') - C^C(\Pi'') \end{aligned} \tag{13}$$

From definition (6) of $G^C_{i,j}(\alpha)$ for this research and equation (13)

$$\begin{aligned} G^C(\Pi') - C^C(\Pi') &> G^C(\Pi'') - C^C(\Pi'') \\ U^C(\Pi') &> U^C(\Pi'') \end{aligned} \tag{14}$$

□

Social (overall) utility is defined as $U(\Pi) = U^T(\Pi) + U^C(\Pi)$. The question therefore becomes under what conditions does an improved coordination utility policy leads to improved social utility; i.e., when does $U^C(\Pi') > U^C(\Pi'') \Rightarrow U(\Pi') > U(\Pi'')$? We consider several cases.

Case 1. $U_i^T(\Pi') \geq U_i^T(\Pi'')$**.** Here, the conflict solving methods do not affect individual task utility (or make it better), for *all* robots. In this case it is easy to see that the accumulated task utility is greater, and the greater task and coordination utilities, combined, result in greater overall utility. From equation (14) and the case assumption

$$\left(\sum_i^N U_i^T(\Pi')\right) + U^C(\Pi') > \left(\sum_i^N U_i^T(\Pi'')\right) + U^C(\Pi'')$$

$$U^T(\Pi') + U^C(\Pi') > U^T(\Pi'') + U^C(\Pi'') \tag{15}$$

$$U(\Pi') > U(\Pi'') \tag{16}$$

Suppose, however, that one robot's task utility under the policy Π' is actually made worse than other the competing policy. Does that automatically mean that the overall utility for the team is worse when using Π'? The answer is no; the robot might in fact be sacrificing its own task utility to maximize the team's (as collaborating robots might be expected to do [5]). The question is whether its sacrifice is compensated for by greater rewards to others.

Case 2. $U_i^T(\Pi') < U_i^T(\Pi'')$**, but** $U^T(\Pi') \geq U^T(\Pi'')$**.** For all reduction in task utility made by the choice of conflict solving method exists number of compensations in other conflicts of other robots in the team.

From equation (14) and the case assumption

$$U^T(\Pi') + U^C(\Pi') > U^T(\Pi'') + U^C(\Pi'') \tag{17}$$

$$U(\Pi') > U(\Pi'') \tag{18}$$

Finally, it might still be possible for the team to perform better with policy Π' even when task performance is made worse.

Case 3. $U^T(\Pi') < U^T(\Pi'')$**, but** $U^T(\Pi'') - U^T(\Pi') < U^C(\Pi') - U^C(\Pi'')$**.** In the case where the loss in team task utility from using policy Π' is smaller than benefit in team coordination utility that policy Π' provides, it is still true that $U^T(\Pi')+U^C(\Pi') > U^T(\Pi'') + U^C(\Pi'')$.

From the case assumption

$$0 < U^T(\Pi'') - U^T(\Pi') \tag{19}$$

and given the premise $U^T(\Pi'') - U^T(\Pi') < U^C(\Pi') - U^C(\Pi'')$,

$$U^T(\Pi') + U^C(\Pi') > U^T(\Pi'') + U^C(\Pi'') \tag{20}$$

and therefore,

$$U(\Pi') > U(\Pi'') \tag{21}$$

Tying these three cases above together, we now state the concluding theorem:

Theorem 2. *EI is a good individual reward for total social utility,* if *(i) either case 1, 2, or 3 above hold; and (ii) EI minimization policy leads to maximal ACC_i*

Proof. Given the use of the FMQ framework, repeated selection of methods that minimizes the EI will lead to minimizing ACC_i. Based on Lemma 1, this will maximize the individual coordination utility U_i^C. And given that one of the cases above holds, this guarantees that the utility of the team will be maximized.

$$\forall\, i, j \;\; EI_{i,j}(\Pi'_{i,j}) < EI_{i,j}(\Pi''_{i,j}) \Longrightarrow U(\Pi') > U(\Pi'')$$

□

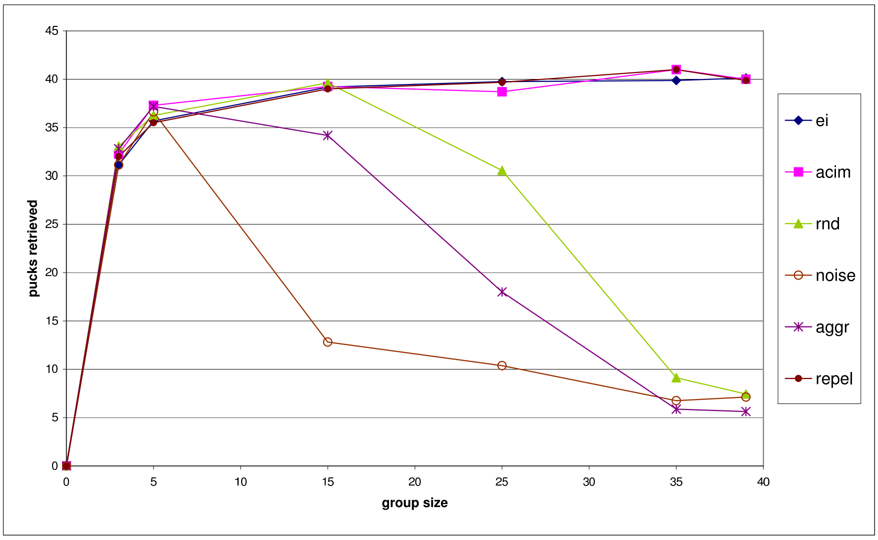
4 Experiments

We now turn to briefly survey a subset of empiric results supporting the use of EI and the stateless Q-learning algorithm in multi-robot team foraging. Here, robots locate target items (pucks) within the work area, and deliver them to a goal region. As was the case in Rosenfeld's work [12], we used the TeamBots simulator [2] to run experiments. Teambots simulated the activity of groups of Nomad N150 robots in a foraging area that measured approximately 5 by 5 meters. We used a total of 40 target pucks, 20 of which where stationary within the search area, and 20 moved randomly. For each group, we measured how many pucks were delivered to the goal region by groups of 3,5,15,25,35,39 robots within 10 and 20 minutes. We averaged the results of 16–30 trials in each group-size configuration with the robots being placed at random initial positions for each run. Thus, each experiment simulated for each method a total of about 100 trials of 10 and 20 minute intervals. We compare the EI method with three types of coordination methods appearing also in [12]: Noise (which essentially allows the robots to collide in their motion uncertainty does not prevent collision), Aggression [14], and Repel, in which robots move away (variable distance) to avoid an impending collision. We compare all of these to random coordination algorithm selection (RND), and to the method of Rosenfeld et al. (ACIM).

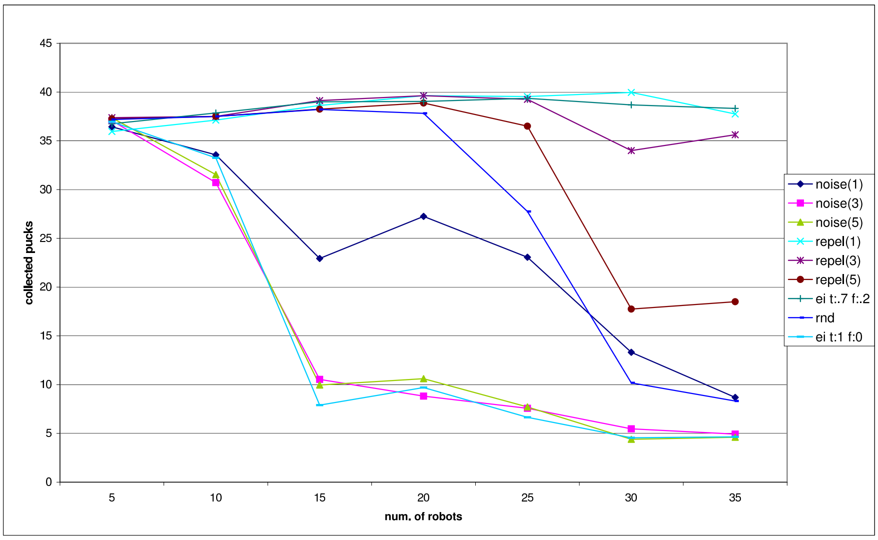
Figures 1(a)–1(d) show a subset of results. In all, the X axis marks the group size, and the Y axis marks the number of pucks collected. Figure 1(a) shows that given no resource limitations, the EI method is just as good as ACIM, though it has not used prior off-line learning. When resource limitations are applied (Figure 1(b)), the EI method is still the best among all the different variations. When resource limits are known a-priori the ACIM method provides the same result (or slightly superior) as EI (Figure 1(c). But when these resource limits are unknown, and methods spend more than advertised, the EI method leads to significantly improved results (Figure 1(d)).

5 Summary

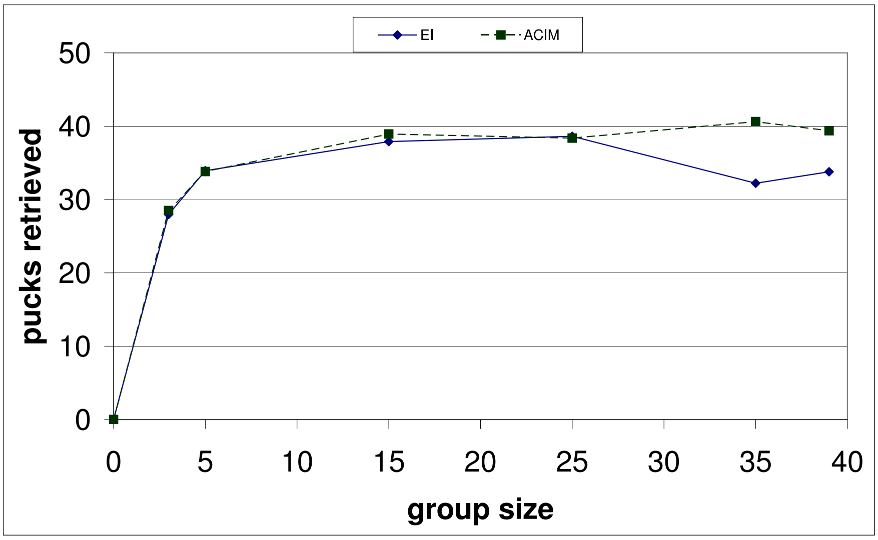
This paper examined in depth the success of previously-report heuristic methods in improving loose coordination in teams, by selecting between different coordination methods. We have shown that these methods can be cast as solving a multi-agent



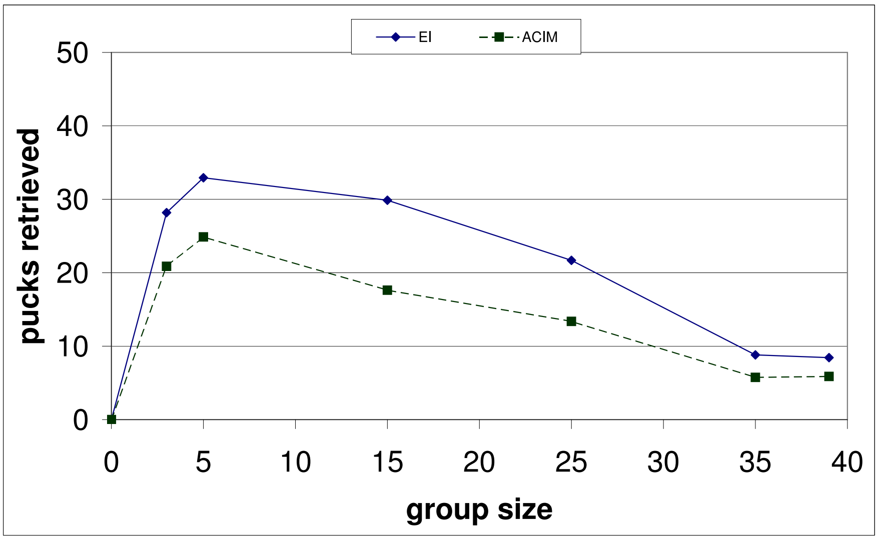
(a) $T = 20$, no resource limits.



(b) $T = 20$, severe fuel limits.



(c) $T = 20$, resource cost known.



(d) $T = 20$, resource cost unknown.

reinforcement learning problem (specifically, a one-stage MDP game), and that existing heuristics can be viewed as rudimentary reward functions.

We have argued for a more principled investigation of appropriate reward functions for this framework, and presented a novel reward function, called Effectiveness Index, which essentially measures the velocity in which resources are spent when reestablishing conflicts. We analytically examine the cases underwhich the use of this reward function leads to improved performance, and then empirically shown that indeed it leads to better performance then existing methods of adaptation. We plan to extend our analysis and empiric investigation to examine additional domains and team tasks.

References

[1] T. Balch. Reward and diversity in multirobot foraging. IJCAI-99 Workshop on Agents Learning, July 1999.

[2] T. Balch. www.teambots.org, 2000.

[3] M. Fontan and M. Matarić. Territorial multi-robot task division. *IEEE Transactions of Robotics and Automation*, 14(5):815–822, 1998.

[4] D. Goldberg and M. J. Mataric. Interference as a tool for designing and evaluating multi-robot controllers. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*, pages 637–642, Providence, RI, 1997. AAAI Press.

[5] B. J. Grosz and S. Kraus. Collaborative plans for complex group actions. *Artificial Intelligence*, 86:269–358, 1996.

[6] P. J. Hoen, K. Tuyls, L. Panait, S. Luke, and J. A. L. Poutré. An overview of cooperative and competitive multiagent learning. In K. Tuyls, P. J. Hoen, K. Verbeeck, and S. Sen, editors, *First International Workshop on Learning and Adaption in Multi-Agent Systems*, volume 3898 of *Lecture Notes in Computer Science*, pages 1–46. Springer, 2006.

[7] S. Kapetanakis and D. Kudenko. Reinforcement learning of coordination in cooperative multi-agent systems. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-02)*, pages 326–331, 2002.

[8] S. Kapetanakis and D. Kudenko. Reinforcement learning of coordination in heterogeneous cooperative multi-agent systems. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-04)*, pages 1258–1259, 2004.

[9] J. R. Kok and N. Vlassis. Collaborative multiagent reinforcement learning by payoff propagation. *Journal of Machine Learning Research*, 7:1789–1828, 2006.

[10] M. J. Matarić. Reinforcement learning in the multi-robot domain. *Auton. Robots*, 4(1):73–83, 1997.

[11] E. Ostergaard, G. Sukhatme, and M. Matarić. Emergent bucket brigading. In *Proceedings of the Fifth International Conference on Autonomous Agents (Agents-01)*, pages 29–30, 2001.

[12] A. Rosenfeld, G. A. Kaminka, S. Kraus, and O. Shehory. A study of mechanisms for improving robotic group performance. *Artificial Intelligence*, 172(6–7):633–655, 2008.

[13] M. Schneider-Fontan and M. Matarić. A study of territoriality: The role of critical mass in adaptive task division. In P. Maes, M. Matarić, J.-A. Meyer, J. Pollack, and S. Wilson, editors, *From Animals to Animats IV*, pages 553–561. MIT Press, 1996.

[14] R. Vaughan, K. Støy, G. Sukhatme, and M. Matarić. Go ahead, make my day: robot conflict resolution by aggressive competition. In *Proceedings of the 6th int. conf. on the Simulation of Adaptive Behavior*, Paris, France, 2000.

[15] E. Yang and D. Gu. Multiagent reinforcement learning for multi-robot systems: A survey. Technical Report CSM-404, University of Essex, 2004.

[16] M. Zuluaga and R. Vaughan. Reducing spatial interference in robot teams by local-investment aggression. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Edmonton, Alberta, August 2005.