Solving the Auction-Based Task Allocation Problem in an Open Environment

David Sarne[1] **and Sarit Kraus**[1,2]
[1]Department of Computer Science, Bar-Ilan University, Ramat-Gan, 52900 Israel
[2]Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742
{sarned, sarit}@macs.biu.ac.il

Abstract

In this paper we analyze the process of allocating tasks to self-interested agents in uncertain changing open environments. The allocator in our model is responsible for the performance of dynamically arriving tasks using a second price reverse auction as the allocation protocol. Since the agents are self-interested (i.e. each agent attempts to maximize its own revenue), previous models concerning cooperative agents aiming for a joint goal are not applicable. Thus the main challenge is to identify a set of equilibrium strategies - a stable solution where no agent can benefit from changing its strategy given the other agents' strategies - for any specific environmental settings. We formulate the model and discuss the difficulty in extracting the agents' equilibrium strategies directly from the model's equations. Consequently we propose an efficient algorithm to accurately approximate the agents' equilibrium strategies. A comparative illustration through simulation of the system performance in a closed and open environments is given, emphasizing the advantage of the allocator operating in the latter environment, reaching results close to those obtained by a central enforceable allocation.

Introduction

Allocating tasks to agents in Multi-Agent Systems (MAS) is a fundamental problem that has attracted the attention of many authors in the field of AI. Obviously, the best allocation (given any efficiency criteria) can be reached when a non-computational bounded allocator assigns tasks to agents it fully controls, while having complete information concerning tasks and and the agents' performance capabilities. Nevertheless, since such a scenario is principally non-realistic, many mechanisms have been suggested for enhancing the task allocation process in environments where agents are not necessarily cooperative (Vulkan & Jennings 2000) or when a centralized mechanism is infeasible due to uncertainty, incomplete information, communication costs, computational complexities, etc. (Shehory & Kraus 1998).

In this paper we consider the problem of a self-interested agent ("central manager") responsible for performing different types of tasks which arrive dynamically along time. This central manager may be defined as a government, a municipality, a company, a project manager, etc., operating in an uncertain environment. The central manager can reassign any of the arriving tasks to other self-interested agents, operating in its environment. The incentive for reassigning a task is either insufficient required resources for performing the task by itself or the possibility of performing it with a smaller cost by one of the agents. Nevertheless, since the central manager does not own these agents (i.e. the agents represent different organizations and/or individuals) or cannot acquire full control over them (Vulkan & Jennings 2000), their willingness to perform a task is associated with a payment they demand in return. Therefore, the central manager needs to come up with a negotiation mechanism (protocol), defining the payments it is willing to pay in exchange for performing each task, and the rules for determining the performer among the agents that are willing to perform it.

We focus on open environments, allowing the entrance and exit of agents. New agents are not always available, but rather arrive dynamically, and choose to enter only if they find the process profitable. Similarly, agents leave the environment upon being assigned tasks (willingly) or if their expected net revenue at the current time is negative (i.e. loss).

Different agents have different sets of basic capabilities. In the context of our model, an agent's capability for performing a given task depends on the specific world state it needs to operate in. By being assigned a task, an agent receives an immediate payment. However it needs to allocate resources in order to perform the task, thus incapable of competing for additional tasks in the near future (possibly associated with better world states and/or smaller competition). Hence, each agent determines its negotiation strategy according to the tradeoff between the immediate gains and the loss of future opportunities. Both, are affected by the protocol set by the central manager, the current world state and its beliefs concerning the other agents' strategies in current and future world states.

Though the allocation is central, the control of the central manager over the final result is limited to the selection of the allocation mechanism. In order to evaluate a specific allocation protocol, the central manager needs to be able to extract the strategies used by different agents given such a protocol (Vulkan & Jennings 2000). For any environment and specific settings, a stable solution is a set of strategies, derived from an equilibrium where no agent can benefit from changing its strategy given the other agents' strategies. The main performance measure used by the central manager for the evaluation of the achieved allocation, using a specific protocol, is the average expense per task.

This research was supported in part by NSF under grant #IIS-0208608.
Copyright © 2005, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

Two typical applications associated with the above model are described in (Sarne, Hadad, & Kraus 2004). The first is an under-water exploration mission (where different companies are encouraged to compete for different tasks, such as underwater surveys, inspections, mapping, pollution prevention and recovery, using their own Remotely Operated Vehicles (ROVs)). The second involves self interested servers, with different configurations and changing loads, competing for the execution of jobs arriving from an external source, such as universities. Additional typical applications in this domain include exploration of remote planets, urban search, and rescue (Dias 2004). In all these applications the agents' capabilities (i.e. costs) to perform any given task are dissimilar in different world states. Thus, upon the arrival of a new task, each agent can calculate its own cost (given the current world state) and assess the distribution of costs among the other agents for performing it. Based on this information the agents have to promptly decide their negotiation strategies.

In this paper, we focus on a specific negotiation mechanism used by the central manager - an auction. Auctions provide an efficient way to resolve one-to-many negotiations, particularly in automated agents based environments (Vulkan & Jennings 2000). Specifically, the central manager in our model uses a reverse Vickrey auction[1].

The proposed model and the analysis are partially based on the framework introduced in (Sarne, Hadad, & Kraus 2004). Nevertheless, the solution methodology given there is limited to closed environments where the entrance of new agents is prohibited (i.e. once an agent is awarded a task, the number of remaining agents always decreases by one). Using such a permissive assumption allowed a backward induction based solution, which we cannot use in our open environment based model. While in closed environments there are no mutual dependencies between the strategies applied in different auctions, in our model the strategy taken in any of the world states affects all other possible world states' strategies. This significantly complicates our problem.

The main contributions of this paper are threefold: First, we formally model and analyze the task allocation problem using a specific auction based mechanism in an open environment with self-interested agents (which is more compatible to real-life applications). Second, we prove that the use of the mechanism in open environments results in lower bids of the agents (i.e. less expenses for the central manager) in comparison to closed environments in similar world states. Finally, we supply efficient algorithms to approximate the equilibrium strategies, thus the mechanism's performance can be evaluated for any specific open environment.

Related Work

The main objective of task allocation is to decide who does what and how to collaborate with others. Generally, task allocation mechanisms in MAS can be divided according to a bi-dimensional classification. The first dimension is the distribution level of the mechanism, ranging from a central allocator for the entire system (Gerkey & Mataric 2002) to a complete distributed approach where agents have initial tasks which they can reallocate through negotiations (Sandholm 1993). While the distributed algorithms do not necessarily reach the optimum allocation, they have the advantage of decreasing the communication and coordination requirements, as well as eliminating the need for a neutral central allocator. The second dimension is the level of cooperation between the agents in the system, ranging from fully cooperative agents (Dias 2004; Shehory & Kraus 1998) to self-interested agents (Vulkan & Jennings 2000; Walsh & Wellman 1999). The latter can be found also in the wide Contract Net protocol literature (Sandholm 1993). While cooperative agents share the same goals or have no notion of individual utilities or preferences, when considering self-interested agents, there is a possibility that some might have an incentive to deviate from the requested cooperation. Thus equilibrium considerations (which become significantly complex in open environments) are the basic infrastructure of the self interested case.

Our model resides in the domain of centralized allocation to self-interested agents, thus requiring a market based approach - an auction. Market based allocation methods in competitive environments are not new (Walsh & Wellman 1999; Vulkan & Jennings 2000). However, they mainly focus on static environments where tasks and other agents are known. Our model integrates an open environment where both agents and tasks arrive dynamically, thus equilibrium considerations become much more complex. The same holds for the analysis given for auctions in ecommerce domains. Here, the main emphasis is (basically due to the large number of participating agents) on maximizing the utility of a single agent that faces multiple dynamic opportunities (Shehory 2002), rather than long term equilibrium analysis.

The Model

We consider an open environment with a central manager and a changing number of self interested agents. The central manager is responsible for allocating tasks which arrive from an external source dynamically, at some inter-arrival time (assumed as a single time unit, for simplification) between two subsequent occurrences. We assume both the central manager and the agents are rational and seek to maximize their net revenue (minimize the costs in the case of the central manager). The dynamic nature of the environment suggests possible entrance of new agents (either former auction winners once they have completed their tasks, or brand new ones). The potential number of agents entering the environment between two subsequent auctions is associated with a probability function. We assume new agents enter the environment sequentially, right after an auction, and only if their expected net revenue in this environment is positive.

An agent's capability of performing a specific task is associated with a cost derived by its basic capabilities, the task's characteristics and the world state. This cost can be modeled as drawn from a specific probability function, shared by all agents (Sarne, Hadad, & Kraus 2004). Additionally, each agent is associated with a cost per time unit, while waiting idly for a task (common to all agents, as similar resources need to be spent).

Upon the arrival of a new task, the central manager initiates a reverse Vickrey auction. Regardless of the bids made

[1]A reverse Vickrey auction is a sealed bid auction in which the winner is payed the lowest amount bid by a loser

for any specific auction, the maximum payment to an agent for performing a task is limited by a value set by the central manager. This can be seen as the cost of performing the task by the central manager itself or an external fixed-cost contractor, thus the central manager is willing to delegate a task to any agent for a payment smaller or equal to this cost. If all agents in the environment bid above this value, the task is not allocated but rather performed by the central manager or the contractor with a cost equal to the maximum payment.

We assume all agents are acquainted with the total number of agents in the environment at the current time, the costs distribution function, the cost per time unit in an idle state, the maximum payment set by the central manager, the interarrival time between tasks, and the entrance rate of new agents. Thus, within a given auction, each agent can evaluate its own cost to perform the proposed task and knows the distribution associated with the other agents' costs.

Problem Formulation

We base our problem formulation on the definitions given in (Sarne, Hadad, & Kraus 2004) and extend them to better reflect our open environment model, where agents are allowed to enter and exit. We consider a set $\mathcal{A}$ of k self interested agents. We denote an agent g by A_g. An agent's cost associated with the performance of a given task in world state s_t, is denoted $c^{A_g}(s_t)$, drawn from a probability function $P_c(x)$ defined over an N discrete values interval $[c_{min}, \ldots, c_{max}]$. An agent's cost per time unit in an idle state, is denoted C. The maximum payment to an agent for performing a task is M. The probability of having z new agents arriving to the environment within a time unit is given by $P_{new}(z)$, $z = 0, ..., m$, $\sum P_{new}(z) = 1$ where m is the maximum number of new agents considering entrance. For any specific environment and given a total of k agents in a world state s_t, our problem is finding the equilibrium bid, denoted $B^k(c^{A_g}(s_t))$, for agent A_g associated with a cost $c^{A_g}(s_t)$.

Model Analysis

When analyzing an open environment with agents' entrances and exits, a key issue under consideration is the highest possible number of participants an agent might encounter in an auction, denoted K. Obviously the value K is derived from the internal forces forming the equilibrium rather than set by the central manager[2]. From the single agent's perspective, the increase in the number of competitors within the environment has a two-fold negative effect. First, the increased competition in each specific auction results in a smaller revenue as the margin between its bid (upon winning) and the second best bid decreases. Second, the expected number of auctions the agent needs to participate in until winning, increases, thus the expected cost of being in an idle state increases. The value K can be seen as the number of agents that once reached, no additional agent will have an incentive to join such an auction mechanism, as its expected revenue (as well as the other agents' expected revenues) from the process is negative. Similarly, the existence of K suggests that none of the agents will leave the environment intentionally (unless assigned a task) as long as the number of agents in the environment is smaller or equal to K. Formally, the existence of K can be proved by using $K > \frac{2(M-c_{min})}{C} - 1$. Here each of the agents in the environment will undoubtedly gain a negative revenue, as the lower bound for the agent's expected cost is greater than the upper bound for its expected payment.

[2] The central manager would never decide to limit the number of agents participating in an auction because any increase in the number of agents competing for a task enhances rigorous competition and thus reduces the overall expected cost paid eventually for any given number of tasks being performed.

The usage of K is critical for the completeness of the equilibrium analysis as it bounds the number of equations that needs to be handled simultaneously and prevents the existence of scenarios where agents still compete in multi-participant auctions where it is obvious that their expected long term revenue is negative (loss). In the rest of this section we present the appropriate modifications of the equilibrium equations given in (Sarne, Hadad, & Kraus 2004), adjusted to reflect a revenue based entrance of new agents into the environment, and discuss the equilibrium structure.

Notice that given the above model's assumptions, the agents' strategy is stationary, i.e., any agent A_1 associated with a cost $c^{A_1}(s_1)$ and k competing agents in a given auction will bid the same as agent A_2 associated with $c^{A_2}(s_2)$ and k competing agents, where $c^{A_1}(s_1) = c^{A_2}(s_2)$. Thus in the rest of this paper, we will refer to all costs $c^{A_g}(s_t)$ satisfying $c^{A_g}(s_t) = c_i \in [c_{min}, \ldots, c_{max}]$ $(A_g \in \mathcal{A})$ as c_i. Similarly, we denote the equilibrium bid $B^k(c_i)$ as B_i^k.

Based on the above, consider an agent which is about to attend an auction with a total of k ($k \leq K$) participating agents. We denote the expected revenue of this agent by R^k. The expected revenue of the agent currently participating in an auction, where its cost for the proposed task is c_i is denoted by $R^k_{c_i}$. Thus the expected revenue R^k can be calculated as:

$$R^k = -C + \sum_{y \in [c_{min}, c_{max}]} R^k_y P_c(y) \qquad (1)$$

An agent winning an auction, when bidding B_i^k, will be awarded the second bid value (bounded by M). Otherwise, it will move on to the next auction where its expected revenue will be either (assuming k agents in the last auction) $\sum_{j=0}^{m} P_{new}(j) R^{min(j+k-1,K)}$, if one of the other agents won this auction; or $\sum_{j=0}^{m} P_{new}(j) R^{min(j+k,K)}$, if all agents used a bid higher than M. For simplification, in the rest of this paper we will use: $R^{k+p(j)}$ to denote $\sum_{j=0}^{m} P_{new}(j) R^{min(j+k,K)}$. The probability, $P_{new}(j)$ is closely related to the eagerness of the agents to win an auction. Any increase in this parameter's mean results in lower expected bids within any auction and a greater number of tasks assignments per time unit. Consequently, such an increase has an opposite affect on K (the equilibrium value of K decreases as the entrance rate increases).

The basic rationale and analysis given in (Sarne, Hadad, & Kraus 2004) for the bidding strategies of the different agents, given a world state s_t and a total of k competing agents, remains valid in our model using the above modifications. Consequently we can prove that in equilibrium the agents are divided according to their cost, c_i, into 3 continuous groups. The first consists of agents with a cost c_i for performing the current task, satisfying $c_i < M - R^{k+p(j)-1}$. These agents (Type I) will always bid $B_i^k = R^{k+p(j)-1} + c_i$ and their equilibrium expected net revenue is given by:

$$R^k_{c_i} = \sum_{y\in[c_{i+1},c_{max}]} (min(B^k_y, M) - c_i)(P_c(c \geq y)^{k-1} - P_c(c > y)^{k-1}) + P_{eq}(B^k_i - c_i) + (1 - P_c(c \geq c_{i+1})^{k-1} - P_{eq})R^{k+p(j)-1} \quad (2)$$

where $P_{eq} = \sum_{j=1}^{k-1} \binom{k-1}{j} \frac{P_c(c_i)^j P_c(c>c_i)^{k-j-1}}{j+1}$ is the probability the agent will win the auction when one or more additional agents have the same cost c_i.

The second group (Type II) consists of agents bidding M as their equilibrium strategy. The expected revenue of an agent from this group, associated with a cost c_i is:

$$R^k_{c_i} = (M - c_i)P_{win} + (1 - P_{win})R^{k+p(j)-1} \quad (3)$$

where $P_{win} = \sum_{j=0}^{k-1} \binom{k-1}{j} \frac{P_c(\underline{c}\leq c\leq \overline{c})^j P_c(c>\overline{c})^{k-j-1}}{j+1}$ is the probability the agent will win the auction when bidding M, and $\underline{c}$ and $\overline{c}$ denote the lowest and highest costs associated with an M bidding strategy, respectively.

The last group (Type III), is of agents associated with a cost $c_i > \overline{c}$. These agents will bid $B_i^k > M$, as their preferred strategy, given their cost c_i is to wait for the next auction. The expected net revenue of these agents is given by:

$$R^k_{c_i} = P_c(c > \overline{c})^{k-1} R^{k+p(j)} + (1 - P_c(c > \overline{c})^{k-1}) R^{k+p(j)-1} \quad (4)$$

At this point, two major obstacles prevent a solution. First, we do not have any means for calculating K, thus we cannot finalize the set of simultaneous equations of types (1-4) that needs to be solved. Second, even if we did have the value of K, the complexity of the equations and the mutual dependencies of the different strategies suggest a major computational challenge that needs to be overcome using an algorithmic approach. For this purpose we propose three algorithms, each built on top of the other, that can facilitate the calculation of the equilibrium. The first algorithm is designed to calculate the different agents' equilibrium bids in an auction with $k \leq K$ participants, given a value K and the expected revenues that can be obtained in any future auction, $R^{k_i}, \forall k_i \neq k$. This algorithm is used as an infrastructure for the second algorithm which calculates the equilibrium R^k values, thus evaluating the validness of the value used for K. Finally, we show how the value of K can be bounded efficiently, and searched over the proposed interval.

Algorithm 1 *calculating equilibrium bids for a specific auction*
Input: *ρ - precision level for the algorithm; K - Maximum number of participants in equilibrium ; k - number of participants in current auction ; M - maximum payment; $c[1:N]$, $P_c[1:N]$ - Vectors of the possible discrete costs and their associated probabilities, respectively; $P_{new}(x)$ - entrance rate; $R^1, ..., R^{k-1}, R^{k+1}, ..., R^K$ - expected revenues in future auctions.*
Output: *$B[1:N]$,R^k - Array of equilibrium bids, and the expected revenue of this auction.*

1 Set $R^k = \frac{-C+\sum_{j=1}^{m} P_{new}(j) R^{min(j+k,K)}}{1-P_{new}(0)}$

2 Set $B[i] = R^{k+p(j)-1} + c[i]$, $\forall i = 1, ..., N$;

3 Find the first element, i, in $B[\,]$, satisfying $B[i] > M$. If $R[i]$ calculated using Equation (3) is greater than when calculated using Equation (4) then set $B[i] = M$. Repeat this stage until reaching an element $\overline{c}$ for which the above condition is not satisfied;

4 Calculate $R[i]$ using equations (2-4) , $\forall i = 1, ..., N$;

5 Calculate R^k using equation (1);

6 Set $B[i] = min(R^{k+p(j)-1} + c[i], M)$, $\forall i = 1, ..., \overline{c}$;

7 Find the last element, $i \in B[\,]$, satisfying $B[i] = M$. If $R[i]$ calculated using Equation (3) is smaller than when calculated using Equation (4) then set $B[i] = M + 1$. Repeat this stage until reaching an element $\overline{c}$ for which the above condition is not satisfied;

8 If the condition in step 7 was satisfied at least once, or the difference between the last two calculations of R^k is greater than ρ then goto 4. Else, stop and return B[1:N];

Theorem 1 *(a) Algorithm 1 always terminates in finite time. (b) The array $B[1:N]$ stores the equilibrium bids with a precision ρ after the algorithm execution is completed*[3].

Sketch of Proof:
Since the detailed proof is quite extensive only its general flow is presented. First, we prove that R^k calculated in step 1 is a lower bound for the equilibrium R^k. Then we prove that using the bids calculated in step 2, the execution of step 3 will lead to a lower bound for $\underline{c}$ and an upper bound for $\overline{c}$ (defining the interval of agents bidding M). For this purpose, we prove and use a proposition, stating that if an agent's optimal bid is M (type (II)), then any other agent associated with a smaller cost and not complying with the condition for type I agents, will bid M as well. Finally, we prove that in the loop executed in steps 4-8: (a) The value of R^k inevitably increases over each calculation; (b) The value of $\underline{c}$ ($\overline{c}$) derived from each execution of steps 7-8 can only decrease (increase), respectively; (c) It is suffice to check the stability of M-value bids downward. Thus over each execution of the main loop, the bid values as well as the value of R^i and the division of types defined by $\underline{c}$ and $\overline{c}$ converge to their equilibrium values. Also, since all parameters' values are either strictly increasing or strictly decreasing throughout the algorithm execution, a stable configuration (in terms of the division into different types) eventually is reached. The accuracy of the bids' values in equilibrium is determined by the parameter ρ. □

The complexity of the algorithm is $o(\frac{MN}{\rho})$. Any attempt to find the equilibrium bids for k agents using direct computation of equations (1-4) will require solving $\frac{N(N+1)}{2}$ permutations of N simultaneous linear equation sets. Each such set can be solved using Gaussian Elimination with a complexity of $o(N^3)$. Since N is highly correlated with the number of possible world states, we expect the ratio between M and ρ to be smaller in its magnitude compared to N^2.

Finding the System Equilibrium

Denoting the expected revenue of an agent in a closed environment by $\overline{R^k}$ and the expected revenue of the agent currently participating in an auction where its cost for the proposed task is c_i by $\overline{R^k_{c_i}}$, we introduce the following theorem.

Theorem 2 *For any k value satisfying $k \leq K$, the expected revenue of an agent participating in an auction in a closed*

[3]Notice that the discrete essence of the environments also suggests rare scenarios where an equilibrium does not exist. Nevertheless, the algorithm can be extended to handle such scenarios.

environment is an upper bound for the expected revenue of an agent participating in an action with $k' \geq k$ agents in an open environment. Formally: $\overline{R^k} \geq R^{k+i} \forall i = 0, ..., K-k$.

Sketch of Proof: Proof by induction. Consider agent $A_{\overline{g}}$ operating in a closed environment and agent A_g operating in an open environment. When $k = 1$, agent $A_{\overline{g}}$ can use the same strategy as A_g (in any k-agents' auction), resulting in an equal or better revenue as its cost components are similar to A_g's, while its expected payment, M, is an upper bound for the expected payment to A_g. Thus $\overline{R^1}$ is an upper bound for R^k (k=1, ..., K). Similarly for k=2 each agent uses higher bids in the open environment, since its alternative expected revenue (i.e. if it does not win the current auction) is greater (because $\overline{R^1} \geq R^{k+i}, \forall i = 0, ..., K-k$). Thus we obtain that $\overline{R^2}$ is an upper bound for R^k ($k = 2, ..., K$). And so on.□

The above theorem suggests that the solution for the closed environment can be a good starting point for finding the equilibrium in an open environment.

Theorem 3 *Given a value K and a set $R = (\widehat{R^1}, ..., \widehat{R^K})$ where $\widehat{R^i}$ is an upper bound to the equilibrium R^i, if a subset of new upper bounds R' can be found where $R^{i'} \leq \widehat{R^i}$ $\forall (R^{i'} \in R')$, then any $R^{j'}$ calculated by substituting $R^i = min(\widehat{R^1}, R^{i'}) \forall R^i \in R$ (where $i \neq j$) in equation (1) is also an upper bound for R^j satisfying $R^{j'} \leq \widehat{R^j}$.*

Sketch of Proof: Using equation (1) we prove that as long as any of the R^i values used is an upper bound to the real values, the calculated value is also an upper bound. □

The above two theorems result in a structured method for checking if a given $K = k$ is the equilibrium maximum number of agents in an auction, and if so, for calculating the agents' equilibrium strategies. This concept is used in the following algorithm.

Algorithm 2 *An algorithm for checking the validity of K.*
Input*: similar to algorithm 1, excluding the R^k values which are not necessary for this algorithm*
Output*: $B[1:K][1:N]$ - An array of equilibrium bids (if exist).*

01 *Calculate R=($R^1, ..., R^K$) using* alg. 1 *with $P_{new}(x)$=0, $\forall x$;*
02 *Repeat* {
03   *Set $R^{i^*} = R^i \quad \forall i \leq K$;*
04   ***For*** *(j=1;j $\leq$ K;j++) calculate R^j and $B[j][\,]$ using* alg. 1*;*
06   ***If*** *$(R^{i^*} - R^i) < \rho \quad \forall i \leq K$ then* {
07   *Calculate R^{K+1} using* algorithm 1*;*
08   ***If*** *$R^{K+1} < 0$ then return(null);*
09   ***Else return*** *$B[\,][\,]$; } };*

Theorem 4 *(a) Algorithm 2 will always terminate in finite time. (b) If K is the equilibrium value, then $B[\,][\,]$ will store the equilibrium bids with a precision ρ after the algorithm execution is completed.*

Sketch of Proof: We use theorem 2 to establish any R^i in step 1 as an upper bound for the equilibrium expected net revenue given i agents $\forall i = 1, ..., K$. Then, according to theorem 3, $R[\,]$'s elements will always contain decreased upper bounds, converging to the equilibrium strategy, given K. □

Notice that while in steps 4-5 of the algorithm we use a simple heuristic by which we calculate R^j sequentially, many alternative heuristics can be used. For example, a heuristic that starts with R^K and calculates the different R^j values sequentially backward, or one that incorporates some level of logic in identifying the next element which will have the maximum affect over future calculations. As long as the basic concept of continuously updating the expected net revenue value is maintained, any heuristic concerning the order by which the different elements are updated will result with equilibrium, in a finite time.

Finally, by using the above algorithm 2, we can outline an additional algorithm that finds the equilibrium K value, by exploring the interval $(1, ..., K_{upper})$, where K_{upper} is an upper bound for K (e.g. the bound given at the beginning of the former section). The search for K in this interval can be done using a binary search, since we know that below this value the system will always yield a positive net revenue and above this value a negative one. Obviously this algorithm, as well as the former two algorithms, can be executed offline prior to the agent's entrance to the environment as they supply the full set of strategies (for all possible world states).

Simulation Results

In this section we aim to illustrate the performance of the open environment compared to those that can be obtained in the closed environment model given in (Sarne, Hadad, & Kraus 2004) and in a central enforceable allocation model. We use an environment where the costs are uniformly distributed in the interval $[10, 50]$ with 100 discrete values, and the parameters: $M = 100$, $C = 2$. The entrance probability used is $P(0) = 1 - \alpha$ and $P(1) = \alpha$, thus $E[P_{new}] = \alpha$.

Figure 1, considers the agents' strategies when reaching a specific auction with $k = 4$ participants within the sequence of auctions. The four curves depict the expected bid as a parameter of the agent's cost for performing the task, c_i (the horizontal axis), in a closed environment and for different entrance rates ($\alpha = 0.1, 0.5, 0.9$) in an open environment. As expected, the bids of the agents in the closed environment are always higher than in the open environment, as these agents confront less competition.

Figure 2 depicts the central manager's average expense per task, as a function of the entrance rate of new agents. In the absence of a common ground for the different models we used the following comparative method which equalize the testing conditions for the open and closed environment. For each α value, we extracted the appropriate equilibrium K^α value in an open environment (results ranged from $K^{0.05} = 42$ to $K^{0.95}$=7). Then, for each (α,K^α) pair we simulated a closed environment, starting with K^α agents and obtained (using 10,000 runs each time) the expected number of tasks performed (until running out of agents), n^α_{tasks}, and the expected average cost per task (assuming non-assigned tasks are performed by an external contractor with a cost M). The latter parameter is described by the most upper curve in the graph, and the changes in its value are associated with the different K^α starting points used, as inherently it is not influenced by α. Then we used a simulation of an open environment (with new agents entering according to α), starting with K^α agents, and checked the expected average cost per task (using 10,000 runs), for performing n^α_{tasks} tasks (described by the middle curve). As expected, the increase in the the entrance rate (associated with α) increases the difference between the performance

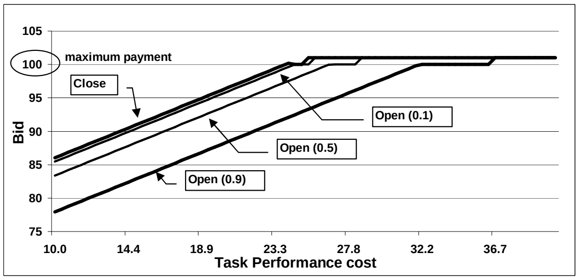

Figure 1: Agents' bids in different environments

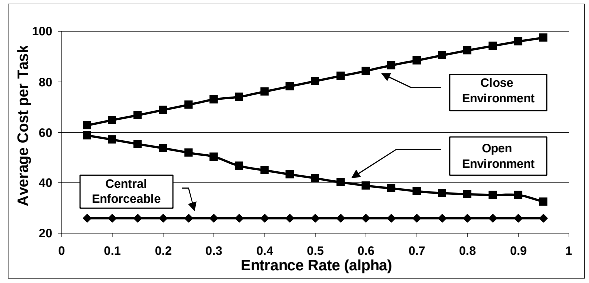

Figure 2: Central Manager's average expense as a function of α

achieved in the two environments. The lower horizontal line, represents the expected cost per task, when the central manager fully controls the agents. Here the central manager only needs to pay the cost C per an idle time of the agents it hires and the actual cost c_i of the agent assigned an arriving task (if any). Thus the central manager's problem is finding the optimal number of agents, K_{opt} to be hired prior to the arrival of a new task. This can be extracted by finding the K value minimizing the expected cost function (per task), $R(K)$, of the central manager for this case:

$$R(K) = C \cdot K + \sum_{y=c_{min}}^{c_{max}} min(y, M)(P(x \geq y)^K - P(x > y)^K) \quad (5)$$

Notice that the second term on the left hand side of the equation is actually the expected minimum of a K-size sample. For our environment we found that the optimum is $K_{opt} = 4$ and the associated expected cost is 26.

A complete analysis concerning the improvement achieved as a function of the different model parameters would require further detailed and more comprehensive scenarios and environments. Nevertheless, such an analysis is beyond the scope of this paper, as our main focus is on the introduction of the general model and its solution method.

Discussion and Conclusions

Scenarios in which an agent or a central manager have limited control over the agents they wish to cooperate with and reallocate tasks, are common in MAS environments. An important sub-class of these scenarios is where all agents are self-interested and attempt to maximize their net revenue. In such case, the performance evaluation of any negotiation protocol towards allocation should be derived from an equilibrium analysis. Here, each agent's strategy should take into consideration both the other agents' long term strategies and the influence changes in its own strategy will have on these strategies. Such an analysis implies a significant complexity, which increases further in open changing environments, thus algorithmic based computational approaches are required. We find the growing interoperability between different systems and environments to be an important factor, leading towards open environments rather than traditional closed ones. We cannot think of a scenario where a central allocator will reject new arrivals as such a strategy will necessarily reduce competition and will result with greater costs per task.

In this paper we focused on the use of an important specific allocation protocol - initiating a second price reverse auction for each arriving task. Former analysis and results that are available for a closed environment model (Sarne, Hadad, & Kraus 2004), can be considered as a specific case of our general open model. By presenting a solution for open environments, we significantly extend the applicability of our model, as many MAS environments are inherently open. Obviously, as new entries are allowed, the possible strategy space becomes infinite and a bound must be placed on the number of agents in the environment. The advantage of our model is that the restriction over the number of agents in the environment emerges from the internal balance according to equilibrium considerations (i.e., the existence of an expected positive revenue), rather than an external limit. This concept fits well into our algorithmic-based solution approach, bypassing the complexities of any attempt of solving the problem using permutation based equation sets.

The solution methodology and the different algorithms given in the former sections, are an important milestone in the process of finding the best negotiation protocol the central manager should use, in terms of the performance measure defined in the introduction. In future work we intend to explore the performance of additional negotiation protocols to be set by the central manager, using a similar equilibrium-based analysis. Additionally, as suggested in the solution methodology sections, we are currently evaluating alternative heuristics that can be integrated in the proposed mechanism to further improve the computation process of the equilibrium strategies given different environmental parameters.

References

Dias, M. 2004. *TraderBots: A New Paradigm for Robust and Efficient Multirobot Coordination in Dynamic Environments*. Ph.D. Dissertation, Robotics Institute, CMU.

Gerkey, B., and Mataric, M. 2002. Sold!: Auction methods for multi-robot control. *IEEE Transactions on Robotics and Automation* 18(5):758768.

Sandholm, T. 1993. An implementation of the contract net protocol based on marginal cost calculations. In *Proc. of AAAI-93*, 256–262.

Sarne, D.; Hadad, M.; and Kraus, S. 2004. Auction equilibrium strategies for task allocation in uncertain environments. In *CIA04*, 271–285.

Shehory, O., and Kraus, S. 1998. Methods for task allocation via agent coalition formation. *AIJ* 101:165–200.

Shehory, O. 2002. Optimal bidding in multiple concurrent auctions. *IJCIS* 3-4:315–327.

Vulkan, N., and Jennings, N. 2000. Efficient mechanisms for the supply of services in multi-agent environments. *Decision Support Systems* 28:5–19.

Walsh, W., and Wellman, M. 1999. Efficiency and equilibrium in task allocation economies with hierarchical dependencies. In *IJCAI'99*, 520–526.