Distributed Multiagent Resource Allocation in Diminishing Marginal Return Domains

Yoram Bachrach Jeffrey S. Rosenschein
School of Engineering and Computer Science
The Hebrew University of Jerusalem, Israel
{yori, jeff}@cs.huji.ac.il

ABSTRACT

We consider a multiagent resource allocation domain where the marginal production of each resource is diminishing. A set of identical, self-interested agents requires access to sharable resources in the domain. We present a *distributed* and *random* allocation procedure, and demonstrate that the allocation converges to the optimal in terms of utilitarian social welfare. The procedure is based on direct interaction among the agents and resource owners (without the use of a central authority).

We then consider potential strategic behavior of the self-interested agents and resource owners, and show that when both act rationally and the domain is highly competitive for the resource owners, the convergence result still holds. The optimal allocation is arrived at quickly; given a setting with k resources and n agents, we demonstrate that the expected number of timesteps to convergence is $O(k \ln n)$, even in the worst case, where the optimal allocation is extremely unbalanced.

Our allocation procedure has advantages over a mechanism design approach based on Vickrey-Clarke-Groves (VCG) mechanisms: it does not require the existence of a central trusted authority, and it fully distributes the utility obtained by the agents and resource owners (i.e., it is *strongly* budget-balanced).

Categories and Subject Descriptors

F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity;
I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent Systems*;
J.4 [**Computer Applications**]: Social and Behavioral Sciences—*Economics*

General Terms

Algorithms, Theory, Economics

Keywords

Resource allocation, Multiagent systems

Cite as: Distributed Multiagent Resource Allocation in Diminishing Marginal Return Domains, Yoram Bachrach and Jeffrey S. Rosenschein, *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.),May,12-16.,2008,Estoril,Portugal,pp. 1103-1120.

1. INTRODUCTION

Multiagent resource allocation problems are an important research area in the field of multiagent systems [12, 1]. These problems deal with allocating resources to autonomous agents, who have preferences over alternative allocations.

Traditional work in mechanism design tries to maximize the sum of the agents' utilities, even when they rationally follow their own selfish goals. Such procedures typically require a *trusted central authority* to gather information and choose the proper outcome. However, an alternative approach is to use a *decentralized* procedure, based on direct interaction among agents.

1.1 Centralized vs. Decentralized Procedures

The mechanism design approach assumes that a *central mechanism* receives the agents' preferences and chooses an outcome, attempting to maximize social welfare. The problem is that agents have private information that the mechanism needs, in order to find the optimal solution. The mechanism can query the agents regarding this private information, but the agents may falsely reply, so as to increase their own utilities. We are interested in the design of incentive-compatible mechanisms (sometimes called strategy-proof, or truthful mechanisms), whose payment schemes motivate the participants to correctly report their private information.

One prominent mechanism design framework is that of Vickrey-Clarke-Groves (VCG) mechanisms [5]. VCG has pronounced advantages: it ensures that agents truthfully report their private information, and finds the optimal outcome; it does so by requiring certain payments from the agents. VCG has the disadvantage of being only *weakly* budget balanced—the net total payments the mechanism receives may be relatively large, and not all the utility generated is redistributed to the agents.

Another disadvantage of the common mechanism design approach is that a central mechanism may be inappropriate in some distributed environments; for example, it may not always be possible to establish a single trusted authority. Also, in centralized solutions, scalability is a major concern, as the central mechanism may soon become a bottleneck of the system [9, 13, 3].

In some cases we can have agents actively participate in choosing the outcome, without using a central mechanism. This is the approach we take. We present a *distributed* and *random* allocation procedure for the multiagent resource allocation problem in certain settings, and demonstrate that the allocation converges to the optimal in terms of utilitarian social welfare. The procedure is based on direct interaction among the agents and resource owners (without the use of a central authority). This type of solution is most appropriate when we cannot establish a trusted central authority.

1.2 Applications and Limitations

Convergence in our procedure is guaranteed only in diminishing marginal production domains, and the negative impact of strategic behavior is only eliminated in highly competitive settings. However, these conditions are quite suitable to certain real-world domains. For example, in grid computing, computational agents residing on nodes in a system typically require access to storage devices to perform a computation. The same data is duplicated across several such devices, which are directly attached to the network (rather than to a specific node). Each agent needs to wait for the data in order to perform the computation, and typically the waiting time increases as more agents share the same storage device. Thus, the marginal utility obtained by an additional agent added to the storage device decreases as more agents share it. When there is a wide selection of storage devices, for example in large grids, the setting is likely to be highly competitive for the owners of the storage devices.

1.3 Related Work

Multiagent resource allocation problems have been studied in the context of several applications, including procurement, manufacturing, allocation of satellite resources, and allocation of resources in grid architectures [4, 7, 10]. Chevaleyre et al. [1] have provided a good multiagent resource allocation survey.

Above, we contrasted our method of choosing an allocation with the mechanism design approach. VCG was developed in several papers (e.g., Groves' seminal paper [5]). VCG has many advantages, but relies on a central mechanism; we aspire to achieve similar results with no central mechanism. Understanding the behavior of a system (or market) as a whole, in the absence of a central mechanism, is an important topic in microeconomics [8]. However, economists generally attempt to *model the conditions* under which an optimal allocation is reached without a central mechanism. Our goal is the opposite—we attempt to *design a protocol* for interaction among the agents, thus *specifying* the appropriate conditions so that an optimal allocation is reached despite the strategic nature of the agents. In this sense, the work is related to *distributed mechanism design*, and similar to Feigenbaum and Shenker's approach [3].

An approach similar to ours was also taken by Heydenreich et al. [6], who discuss a scheduling domain where jobs choose the machine on which they are to be processed. That domain is quite different from ours, and attempts to minimize completion time, rather than maximizing production. Also, that work focuses on a "myopic best response" solution concept, and employs online analysis to reach a competitive algorithm, whereas we reach the *optimal* solution.

Several papers have analyzed the related issue of negotiations over resources [11, 2], but in these general domains, the optimal outcome is only *possible* to achieve (so a sub-optimal result may be reached).

1.4 Structure of the Paper

We consider domains where *identical* agents require access to *exactly one* resource, which may also be *shared* with other agents. Using a resource, agents can generate utility. Each resource has a production function, mapping the number of agents who share the resource to the total utility generated by all those agents (the formal model appears in Section 2).

Two principle questions arise. First, how do we allocate the resources to the agents so as to maximize the total utility generated? Second, how should the total utility produced be divided among the entities (agents and resource owners)?

We suggest, in Section 3, a *distributed*, *random*, and *market oriented* allocation procedure, which is composed of a sequence of interactions among potentially *self-interested* entities. The interaction proceeds in rounds, and is performed directly between agents and resource owners; our method does not require establishing and maintaining a central mechanism. During each round, only a polynomially bounded number of messages is sent, so our protocol method is scalable. We also suggest interaction strategies for agents and resource owners, in Section 4. When these strategies are used in particular domains, the allocation *converges* to the optimal allocation (demonstrated in Section 5). In Section 6, we consider strategic behavior, and show that in certain settings, which are *highly competitive for the resource owners*, no agent or resource owner has any incentive to deviate from the suggested strategy in a way that would affect the convergence result. Expected time to convergence is discussed in Section 7, and we conclude in Section 8.

2. PROBLEM FORMALIZATION

DEFINITION 1. *A single shareable resource allocation domain is composed of a set of* identical *agents* $Ag = \{a_1, a_2, \ldots, a_n\}$*; a set of resources* $R = \{r_1, r_2, \ldots, r_k\}$*; and a set of production functions* $P = \{p_1, p_2, \ldots, p_k\}$*. Each production function* $p_i : \mathbb{N} \to \mathbb{R}+$ *maps the number of agents who are sharing the resource* r_i *to the total utility produced on that resource. The production on a resource is 0 when no agents are using the resource, so for all* r_i *we have* $p_i(0) = 0$.[1]

An allocation is a function $A : Ag \to R$, mapping every agent to the resource it is using. We denote by $A_{r_i} = \{a_j | A(a_j) = r_i\}$ the set of agents that allocation A maps to resource r_i. Given an allocation A, we call $P_i(A) = p_i(|A_{r_i}|)$ the production in resource r_i in allocation A.

A *utility division function* for resource r_i is a function $d_i : Ag \to \mathbb{R}+$ that maps each of the agents who share resource r_i to its share in the utility produced in r_i. If $a_j \notin A_{r_i}$ then $d_i(a_j) = 0$ (i.e., when the agent a_j is not allocated the resource r_i, it has no share in the utility division of resource r_i). The utility division function cannot divide among the agents more than what was produced on the resource, so for all the resources r_i we have $\sum_{j=0}^{n} d_i(a_j) \leq P_i(A)$. The remainder of the utility produced is the share of the resource owner of r_i, and we denote $d_{r_i} = P_i(A) - \sum_{j=0}^{n} d_i(a_j)$. A *utility division* is the set of utility division functions for all the resources $D = \{d_1, d_2, \ldots, d_k\}$.

We denote by $P(A) = \sum_{i=0}^{k} P_i(A)$ the total production in allocation A. Since all of the production is distributed

[1]When the agents are *not identical*, the production functions are $p_i : 2^{Ag} \to \mathbb{R}+$, mapping the set of agents who are sharing the resource r_i to the total utility produced on that resource. We are considering the specific case of identical agents.

either to the agents or the resource owners, for any allocation A we have $P(A) = \sum_{i=0}^{k} \sum_{j=0}^{n} d_i(a_j) + \sum_{i=0}^{k} d_{r_i}$. The *marginal production* of a resource r_i when j agents are allocated to that resource is $m_i(j) = p_i(j) - p_i(j-1)$. We say a production function p_i has *diminishing marginal return* if for all $0 < a < b < n$ we have $m_i(b) \leq m_i(a)$.

Given such a domain, we want to achieve the optimal allocation A, maximizing $P(A)$. Given the production functions, $P = \{p_1, p_2, \ldots, p_k\}$, this is straightforward. With diminishing marginal returns, a greedy algorithm which iteratively adds an agent to the resource with the highest marginal return finds an optimal allocation. However, the production functions are known only to the resource owners. It is possible to ask each such resource owner to declare its production function, find the optimal allocation *for the declared values*, and divide the utility in any way we desire. However, the resource owners are self-interested and may declare false production functions in order to increase their own utilities, depending on the way we choose the utility division. How can we maximize $P(A)$ without knowing the production functions, while also taking into account the self-interested nature of the entities?

Below, we show that it is possible to reach an optimal allocation by allowing the agents and resource owners to interact using a certain protocol, that determines both the allocation and utility division. An alternative approach is the mechanism design approach, using the VCG framework. In order to highlight the advantages of our method over the VCG mechanism for this particular problem, we first briefly describe the VCG solution.

2.1 A Mechanism Design Approach

In the mechanism design solution, we construct a central trusted authority, the mechanism, which is in charge of eliciting private information and choosing an outcome based on this information. In our domain, the private information is the set of production functions, and the outcome is an allocation A and a utility division D for that allocation. A common framework for designing mechanisms is VCG. In VCG, the mechanism chooses the optimal allocation given the reported information a, but also requires payments from the agents. If a_i's value from the chosen outcome a is $v_i(a)$, the mechanism charges a_i the quantity $t_i = h_i(v_{-i}) - \sum_{j \neq i} v_j(a)$, where h_i is an arbitrary fixed function that does not depend on v_i. An important feature of VCG is that the payment rule results in truthful reports from rational utility-maximizing participants.

A VCG solution achieves the optimal allocation, but it has the disadvantage that the net payments to the mechanism may be positive. This means that not all the generated utility $P(A)$ is distributed among the agents and resource owners—some of it may need to stay in the mechanism's hands. Another disadvantage of VCG is that it requires a *central* authority that all agents trust. Thus, this method may be inappropriate for certain distributed domains.

3. ALLOCATION BY INTERACTION

Our method of obtaining an allocation is based on direct interaction between the agents and resource owners, using a particular protocol; we now define this protocol. For analysis purposes, we divide the interaction into discrete time units called *rounds*. In each round, an agent only has time to interact with a single resource owner. For simple analysis, we assume each agent *in turn* interacts with one resource.[2]

The protocol allows the following messages:

1. *Resource Request.* This message is sent from agent a_j to the owner of resource r_i, indicating that the agent is considering using the resource, and requests an offer for the payment it would get from the resource owner (other than the payments to the agents, the resource owner keeps the rest of the production generated on that resource).
2. *Payment Bid.* This message is sent from a resource owner r_i to an agent a_j, and includes a parameter $w_{i,j} \in R+$, indicating the payment from the resource owner to the agent if the resource is allocated to the agent. If the agent agrees to this bid, this determines the share of the production of that resource the agent would get. The share of the production not paid to any agent is the share of the resource owner. Thus, these messages determine the utility division functions at the end of the round.
3. *Accept.* This message is sent from an agent to a resource owner, indicating that the agent agrees to use the resource. The set of accept messages determines the allocation at the end of the round.
4. *Decline.* This message is sent from an agent to a resource owner, indicating that the agent does not agree to use the resource. It may be sent as a response to a Payment Bid, indicating that the agent wants to keep on using the resource that is currently allocated to him, or as a reaction to an interaction with a different resource, indicating that the agent does not want to use the current resource anymore, and is switching to a new resource.
5. *Payment Change.* This message is sent from resource owner r_i to an agent a_j, indicating that the resource owner changes the payment it is willing to offer the agent. The message includes a parameter $w_{i,j} \in R+$, the new payment the resource owner offers the agent. Such a message may only be sent by the resource owner as a result of receiving an accept message from some agent.
6. *Round Payment.* This is similar to the Payment Bid message, except it is sent only at the beginning of each round. The message is sent from a resource owner to all the agents that have accepted the bid by that resource, and have not sent a decline message to the resource. Like the payment bid message, it indicates the payment the resource owner is willing to offer the agent, and includes a parameter $w_{i,j} \in R+$, the new payment the resource owner offers the agent.

Each round proceeds as follows. First, each resource owner sends a Round Payment message to each of the agents who have accepted the resource's payment bid (by sending it an Accept message) and not yet declined it (by sending a Decline message). This message indicates the payment the agent would receive if it decides to keep using the sending resource, and not switch to a different resource. The Round Payment message contains the fee the resource owner is willing to pay agents who are allocated that resource. The rest of the utility generated on that resource would belong to the resource owner.

After the Round Payment messages, each agent *in turn* may submit *one* resource request to *one* resource owner.

[2]Allowing concurrent interaction speeds up the interaction, but requires some way of handling consistency and deadlocks. We avoid having to deal with these issues by allowing the agents to interact with the resources one at a time. It also simplifies analysis of convergence times.

This message indicates that the agent *considers* switching to that resource. A resource owner replies to the Resource Request message with a Payment Bid message, which contains the fee the resource owner is willing to pay agents who are allocated that resource. The rest of the utility generated on that resource belongs to the resource owner.

An agent replies to the Payment Bid message with either an Accept message (if it switches to the bidding resource) or with a Decline message (indicating the resource allocated to that agent does not change). If an Accept Message is sent, a Decline Message must be sent to the old resource (owner) the agent was using. Once an agent has accepted a resource owner's bid, the resource owner may change the payment it offers the agents currently allocated that resource, by sending them a Payment Change message. Such a message indicates that the payment offered to these agents during the next round would be different. Agents who get a Payment Change message reducing their payment have a chance of switching resources during the next round. This concludes the interaction for this agent, and the round continues with the next agent, who may submit his Resource Request. Once all the agents have finished their interaction, the round ends, and the process continues in the next round.

3.1 Chosen Allocation

We now define the allocation chosen after a round of interaction. If during round r agent a_j has sent an Accept message to resource r_i, then in the allocation A^r chosen at the end of that round, resource r_j is allocated to agent a_i so $A^r(a_j) = r_i$. Such an Accept message has been sent in response to a Payment Bid message from r_i to a_j, with a parameter $w_{i,j}$. Unless Payment Change messages have been sent later during that round from r_i to a_j, then this bid determines the share of a_j in the utility division of that resource: $d_i(a_j) = w_{i,j}$. If Payment Change messages have been sent from r_i to a_j later during that round, the parameter $w_{i,j}$ of the *last* Payment Change message sent determines a_j's payment, and $d_i(a_j) = w_{i,j}$.

If agent a_j replies with a Decline message to r_i's Payment Bid, then the allocation for a_j remains as in the previous round: $A^r(a_j) = A^{r-1}(a_j)$. In such a case, a_j remains allocated to some resource $r_x = A^{r-1}(a_j)$. If r_x has not sent any Payment Change messages to a_j during round r, then the payment that agent a_j gets from r_x is as it was in the parameter of the Round Payment message sent from r_x to a_j, at the beginning of the round. If r_x does send Payment Change messages to a_j during round r, then the parameter $w_{x,j}$ of the *last* Payment Change message sent determines a_j's payment, and $d_x(a_j) = w_{x,j}$. Note that only the bids that resulted in Accept messages and the *last* Payment Change messages determine the utility division.

The total production given allocation A is, as defined in Section 2, $P(A) = \sum_{i=0}^{k} P_i(A)$. Our goal is to choose an optimal allocation A^{opt} such that for any allocation A' we have $P(A^{opt}) \geq P(A')$. As also explained in Section 2, this maximizes social welfare, since $\sum_{i=0}^{k} \sum_{j=0}^{n} d_i(a_j) + \sum_{i=0}^{k} d_{r_i} = P(A)$. Note that the protocol determines not only the allocation, but also a utility division. When optimizing for utilitarian social welfare, we do not consider how production is distributed, and concern ourselves only with total production. Other definitions of social welfare (such as Nash product social welfare, egalitarian social welfare, etc.) additionally take into account the utility division among agents.

4. SUGGESTED STRATEGIES

We here suggest a strategy for the agents in our scenario, and a strategy for the resource owners. Later we show that these strategies have certain desirable properties.

A. Agents: Each agent keeps track of the current resource allocated to it, R_{cur}, and his share of the utility produced on that resource, *CurPayment*. On the first round *CurPayment* is set to 0, and R_{cur} indicates the agent is not allocated to any resource. In each round, each agent randomly chooses a resource and requests use of that resource by sending a Resource Request message. If the Payment Bid message from that resource indicates a higher utility than the agent currently has, it switches to that resource. The pseudocode for the agents' strategy is as follows:

1. Set *CurPayment* to the value of the Round Payment message sent at the beginning of the round.
2. For each Payment Change message received, update *CurPayment* to the value declared in this message.
3. Randomly choose a resource R_{new}, and send that resource a Resource Request message. The resource would reply with a Payment Bid message. Set *OfferedPayment* to the value of that message.
4. If *OfferedPayment* $>$ *CurPayment*:
   (a) Send the current resource a Decline message;
   (b) Send resource R_{new} an Accept message.
5. If *OfferedPayment* $\leq$ *CurPayment*:
   (a) Send resource R_{new} a Decline message.

B. Resource owners: Each resource owner r_i keeps a list of agents allocated that resource, A_{r_i}, and their number, $num_i = |A_{r_i}|$. These are the agents who have sent an Accept message to the resource, and have not yet sent a Decline message indicating that they have switched resources. During the first round, A_{r_i} is an empty list, and num_i is 0. At the beginning of each round, the resource owner sends the agents who are allocated that resource a message, indicating that their share of the utility is the current marginal production in that resource. The resource then waits for Resource Request messages, and replies to each such message with a payment bid of the *next* marginal production on that resource. Agents who send an Accept message are added to A_{r_i} (and thus the resource is allocated to them as well), and agents who send a Decline message are removed from A_{r_i} (and thus the resource is no longer allocated to them). If an agent switches to a resource, the resource owner updates the offered payment to all the agents it is allocated to be the new marginal production on that resource, after adding the new agent, by sending Payment Change messages. The pseudocode for the resource owners' strategy is as follows:

1. For each agent a_j in A_{r_i}, send a_j a Round Payment message with value of $w_{i,j} = m_i(num_i)$.
2. For each Resource Request message received from an agent a_j:
   (a) Reply with a Payment Bid message of the marginal production, assuming a_j would accept the offer: $w_{i,j} = m_i(num_i + 1) = p_i(num_i + 1) - p_i(num_i)$;
   (b) If agent a_j replies with a Decline message, ignore that message (nothing needs to be done);
   (c) If a_j replies with an Accept message:
      i. send a Payment Change message to any agent a_x in A_{r_i} (any agent who is currently allocated this resource), with a parameter $w_{i,x} = m_i(num_i + 1)$;
      ii. set $num_i = num_i + 1$.

We now consider what happens when the entities follow these suggested strategies. The first round occurs with an allocation A_0, when no agent is allocated any resource. In A_0 the production on any resource r_i is $p_i(0) = 0$; all the resource owners get $d_{r_i} = 0$, and all the agents get payments of $d_i(a_j) = 0$ for any resource i and agent j. The resources are not allocated to any agent, and thus no Round Payment messages need to be sent.

During round r, a new allocation A_r is constructed, by improving the previous allocation A_{r-1}. During round r, agents attempt to improve their own utilities, by switching to a resource that gives them a higher payment, according to their own self-interest. Each resource owner chooses a payment according to the marginal production on that resource, and updates all the agents to which it is allocated regarding this fee. Therefore, when an agent accepts a Payment Bid from a resource owner, this means the agent is now allocated a resource with a higher marginal production. The production in the agent's old resource (the one allocated to it during round $r-1$) decreases, since in round r fewer agents would use that resource, and the production in the new resource increases. However, the production gain in the new resource is greater than the production loss in the old resource, since the marginal production in the new resource is higher than in the old resource.

5. PROCEDURE CONVERGENCE

We now consider a single sharable resource allocation problem, where agents and resource owners interact using the allocation protocol defined above in Section 3. We have a set of *identical* agents $Ag = \{a_1, a_2, \dots, a_n\}$, and a set of resources $R = \{r_1, r_2, \dots, r_k\}$ with production functions $P = \{p_1, p_2, \dots, p_k\}$, which have diminishing marginal production. We prove that the suggested strategies result in convergence to an optimal allocation, and that in certain settings no rational strategic behavior changes this convergence result.

THEOREM 1 (MONOTONIC IMPROVEMENT). *Let A^{r-1} be the allocation chosen at round $r-1$. If during rounds r and $r-1$ the entities follow the suggested strategies, as defined in Section 4, then the allocation chosen at the end of round r, A^r, is no worse than the allocation in the previous round, so $P(A^r) \geq P(A^{r-1})$.*

PROOF. Each round is a series of interactions between an agent a_j and a resource owner r_i. Each such interaction either changes the allocation by allocating a different resource to a single agent (when that agent sends an Accept message to the new resource), or leaving the allocation as it is (when that agent sends a Decline message to the new resource). If the resource owners follow the suggested strategy, then after each such change in the allocation, the payment to the agents who are allocated that resource is changed to the marginal production on that resource (when the resource owner sends a Payment Change message).

Let A' be the allocation during round r, just before a_j's interaction with r_i. Let r_x be the resource allocated to a_j at the end of round $r-1$. Let num_x be the number of agents to whom resource r_x is allocated prior to the interaction between r_i and a_j. Let num_i be the number of agents to whom r_i is allocated at that time. The production on r_x is $p_x(num_x)$, and the payment that agent r_j gets is $d_x(a_j) = m_x(num_x) = p_x(num_x) - p_x(num_x - 1)$. The payment bid sent by r_i to a_j is sent with a parameter $w_{i,j} = m_i(num_i + 1) = p_i(num_i + 1) - p_i(num_i)$.

Agent a_j only accepts that bid if he gets a higher payment, so if that bid is accepted $m_i(num_i + 1) > m_x(num_x)$. Otherwise, a_j declines, and the allocation remains unchanged, and the total production remains the same. If a_j accepts, the allocation A' is changed to A'' by allocating a_j the resource r_i instead of r_x. The total production in r_x drops by $m_x(num_x)$, and rises in r_i by $m_i(num_i + 1)$. Since $m_i(num_i) > m_x(num_x)$, we have $P(A'') > P(A')$. Note that if a_j switches to r_i, the payment that the agents r_i are allocated to get decreases, and the payment that agents r_x are allocated to get increases (due to the new Round Payment set in the next round), but the *total production* increases. Since each round is composed of a series of interactions, each either changing the allocation in a way that increases the total production or not changing the allocation, we have $P(A^r) \geq P(A^{r-1})$. □

THEOREM 2 (STABILITY IN OPTIMUM). *Once the optimal allocation A^{opt} is reached, it never changes. If the allocation at the end of round r is $A^r = A^{opt}$, it remains the same during round $r+1$ and $A^{r+1} = A^{opt}$.*

PROOF. The allocation only changes when some agent accepts the bid of some resource owner. In such a case, as shown in Theorem 1, a new allocation A' is chosen, and the total production increases, so $P(A') > P(A^{opt})$. But that contradicts the fact that A^{opt} is an allocation with the highest possible total production. □

Theorem 1 shows the allocation in a given round never becomes worse than that of the previous round. That by itself is not enough to guarantee convergence to the global optimal allocation, as the procedure can be stuck in a *local optimum*. The next theorem shows that for the suggested strategies, there are no such local optima: once the protocol reaches an allocation that cannot be improved by any round using the protocol (a local optimum), it is an optimal allocation (global optimum).

DEFINITION 2. *Protocol stable allocation. Let Ag and R be sets of agents and resource owners, interacting using the protocol defined in Section 3. Let the strategies the agents are using be $s_{a1}, \dots, s_{an}$, and the strategies the resources are using $s_{r1}, \dots, s_{rk}$. Allocation A is a* protocol stable allocation *for that strategy profile if, once reached, no interaction between the agents and resource owners using these strategies would result in a change in A. [Unless otherwise stated, when discussing protocol stable allocations, we are referring to the suggested strategies, as defined in Section 4.]*

If A is a protocol stable allocation (for the suggested strategies), there is no agent a_j who can send a resource request message to some resource r_j which would result in a price bid that a_j would accept. Since the protocol proceeds in rounds, and each round is composed of a series of interactions between single agents and resource owners, this means that there is no *single* agent that would be better off by switching to a different resource alone.

THEOREM 3 (PROTOCOL STABLE ALLOCATION IS OPTIMAL). *If A is a protocol stable allocation (for the suggested strategies), then it is an optimal allocation.*

PROOF. Let A^{opt} be an optimal allocation. Let A be a protocol stable allocation, that differs from A^{opt}. We construct a series of allocations $A^{opt} = A^1, A^2, \ldots, A^m = A$. Each such allocation is the same as the previous one, except that it allocates one of the agents a different resource. We show that they all have equal production, so for all $2 \leq i \leq m$ we have $P(A^i) = P(A^{i-1})$. Therefore A is also an optimal allocation.

Let $A^1 = A^{opt}$. A differs from A^1, so there are some resources that are allocated to more agents in A^1 than in A. Thus, there are also some resources that are allocated to fewer agents in A^1 than in A (since both allocations allocate resources to the same number of agents). Let r_a be a resource that is allocated to more agents in A^1 than in A, and let r_b be a resource that is allocated to fewer agents in A^1 than in A. We denote $a = m_a(|A^1_{r_a}|)$, $b = m_a(A_{r_a})$, $c = m_b(|A_{r_b}|)$, $d = m_b(A^1_{r_b})$.

Since all the resources have diminishing marginal production, we have $a \leq b$, and $c \leq d$. Let $b_1 = m_a(|A_{r_a}| + 1)$, $b_2 = m_a(|A_{r_a}| + 2)$, $b_3 = m_a(|A_{r_a}| + 3)$ and so on, until for some i we have $b_i = m_a(|A_{r_a}| + i) = a$ (possibly $i = 1$ if this is the next marginal production on that resource). In the same way, let $d_1 = m_b(|A^1_{r_b}| + 1)$, $d_2 = m_b(|A^1_{r_b}| + 2)$, $d_3 = m_b(|A^1_{r_b}| + 3)$ and so on, until for some j we have $d_j = m_b(|A^1_{r_b}| + j) = c$. Again, due to diminishing marginal production, we have $b \geq b_1 \geq b_2 \geq \ldots \geq b_i = a$ and $d \geq d_1 \geq d_2 \geq \ldots \geq d_j = c$. Allocation A is protocol stable. In A, agents A_{r_a} obtain a payment of b, the marginal production on r_a, and agents A_{r_b} obtain a payment of c, the marginal production on r_b. Thus, $b_1 \leq c$, since otherwise (if $b_1 > c$) one agent from r_b could do better by switching to r_a. We thus have $a \leq b_1 \leq c \leq d_1 \leq d$, so $a \leq d_1$.

Let a_j be some agent that A^1 allocates the resource r_a. We define A^2 to be the same allocation as $A^1 = A^{opt}$, except that A^2 allocates a_j the resource r_b instead of r_a.

$$A^2(a_i) = \begin{cases} A^{opt}(a_i) & \text{if } a_i \neq a_j; \\ r_b & \text{if } a_i = a_j; \end{cases}$$

The difference in total production in A^2 and A^1 is only due to the change in allocation of the resource for a_j. A^2 produces less in r_a and more in r_b than A^1. In r_a, A^2 produces $m_a(|A^1_{r_a}| = a)$ less than A^1 (since we moved exactly 1 agent, and this was the marginal production in r_a). In r_b, A^2 produces $d_1 = m_b(|A^1_{r_b}| + 1)$ more than A^1, since we have added 1 agent to that resource. We showed that $a \leq d_1$, so $P(A^2) - P(A^1) = d_1 - a \geq 0$. But A^1 is an optimal allocation, with maximal production, so $P(A^2) = P(A^1) = P(A^{opt})$, so A^2 is also an optimal allocation.

We can continue the same process as long as there is a difference between the newly-built optimal allocation and A, our protocol stable allocation. Since there are a finite number of allocations, eventually, for some i we will have $A^i = A$, so A is also an optimal allocation. □

When entities use the suggested strategies, the allocation only improves in the next round, and if we have not reached the optimal allocation yet, there is a possible round which can increase the allocation quality. Since there are a finite number of allocations, this shows that our procedure eventually converges to an optimal allocation. However, agents may choose not to follow the suggested strategies, so we may converge to a sub-optimal allocation. In the next section, we consider such strategic behavior.

6. STRATEGIC BEHAVIOR

We now show that no agent or resource owner has an incentive to deviate from the suggested strategies in a way that would change the final allocation. We consider a domain with a set of agents Ag and a set of resources R. When the entities interact using the protocol defined above in Section 3, and using the suggested strategies defined above in Section 4, Theorem 3 showed that the allocation would converge into an optimal allocation A^{opt}, and a certain utility division $D = \{d_1, d_2, \ldots, d_k\}$. Let $a_j \in Ag$ be one of the agents. Let $r_x = A^{opt}(a_j)$, and the utility of a_j in A^{opt} is $u_{r_j}(A^{opt}) = d_x(a_j)$. Let $r_i \in R$ be one of the resources. The utility of r_i in A^{opt} is $u_{r_i}(A^{opt}) = d_{r_i} = P_i(A^{opt}) - \sum_{j=0}^{n} d_i(a_j)$.

THEOREM 4 (AGENT DEVIATIONS). *Let s'_{a_j} be some strategy for a_j. Let S' be the strategy profile where all the entities follow their suggested strategies, except a_j who follows strategy s'_{a_j}. If under S' the allocation converges to some protocol stable allocation A' such that $u_{a_j}(A') > u_{a_j}(A^{opt})$, then $P(A') = P(A^{opt})$, and A' is also an optimal allocation. In other words, if a_j used some strategy and managed to increase its utility, then the optimal allocation is still reached.*

PROOF. Allocation A' is a protocol stable allocation under S', and thus, after a certain round r, a_j is allocated a certain resource r_x, and never leaves it (otherwise A' is not a protocol stable allocation under S'). Let the allocations in the following rounds be $A^{1'}, A^{2'}, \ldots$ Due to the same reasons in Theorem 1, $P(A^{i+1'}) \geq P(A^{i'})$ (since a_j never switches to a different resource, and if any of the other agents switches, the production improves). When A' is reached, no agent except (maybe) a_j can do better by switching resources.

If there is some optimal allocation A^{opt} such that $A^{opt}(a_j) = r_x$, due to the same reason as in Theorem 3, A' is also an optimal allocation. If no such A^{opt} exists, take any optimal allocation A^{opt}. Starting in round r, a_j does not affect the considerations of the other agents: once a protocol stable allocation is reached, none of them would switch to r_x even if a_j was not allocated that resource, and since a_j is allocated that resource, the payment r_x offers any of them is even lower. The payment a_j gets on r_x is $d_x(r_j)$, lower than the next marginal production on any of the resources in A^{opt}. In A^{opt} agent a_j is allocated some resource r_y, and agent a_j gets the next marginal production on r_y, with a higher utility. Thus, if a_j sticks to some resource to which some agents are allocated in some optimal allocation, the protocol converges to some optimal allocation, and if a_j sticks to some resource r_x that no optimal allocation allocates any agent, his utility is smaller than in any optimal allocation. □

We define a condition under which resource owners have no incentive to deviate from the suggested strategy. Let S^* be the strategy profile where all the entities follow the suggested strategies. As shown in Section 3, under S^* the allocation converges to some optimal allocation A^{opt}. Let w^*_i be the payment that resource r_i offers agents who are allocated that resource in that allocation. Let the number of agents who are allocated r_i there be $l^*_i = |A^{opt}_{r_i}|$.

We later show that r_i has no incentive to offer any payment higher than w^*_i. However, in some cases r_i does have an incentive to offer a payment below w^*_i. When r_i offers a smaller payment, $w^*_i - \epsilon$, for some $\epsilon > 0$, every agent who finds a resource where the next marginal production is higher than $w_i^* - \epsilon$ would switch to that resource. Since the marginal productions diminish when a resource is allocated to more agents, eventually no such resources would be found, and we would once again reach a stable allocation. Thus, when the resource owner drops its suggested payment to $w^* - \epsilon$, some agents would switch to different resources, thus decreasing the number of agents who are allocated r_i from l_i^* to l_i^c (l_i^c being a function of the payment change ϵ, so l_i^c is shorthand for $l_i^c(\epsilon)$). The total payments r_i gives the agents would then drop from $l_i^* \cdot w_i^*$ to $l_i^c \cdot (w_i^* - \epsilon)$. However, the production also drops from $p_i(l_i^*)$ to $p_i(l_i^c)$.

For payment w_i^*, we know that the marginal production of each of the agents to which w_i is allocated in A^{opt} was higher than w_i^* (since the marginal production is diminishing, and w_i^* was the marginal production of the last agent on r_i: $w_i^* = m_i(l_i^*)$). We denote the utility r_i obtained from the l'th agent allocated that resource as $g_i(l) = m_i(l) - w_i^*$. So, $p_i(l_i^*) - p_i(l_i^c) = \sum_{l=l_i^c+1}^{l^*}(g_i(l) + w_i^*) = (l_i^* - l_i^c) \cdot w_i^* + \sum_{l=l_i^c+1}^{l^*} g_i(l)$. Thus, when r_i reduces the suggested payment by ϵ, from w_i^* to $w_i^* - \epsilon$, its utility increases by:

$\Pi_i = l_i^* \cdot w_i^* - l_i^c \cdot (w_i^* - \epsilon) - (p_i(l_i^*) - p_i(l_i^c)) =$
$(l_i^c + (l_i^* - l_i^c)) \cdot w_i^* - l_i^c \cdot w_i^* + l_i^c \cdot \epsilon - (p_i(l_i^*) - p_i(l_i^c)) =$
$l_i^* \cdot w_i^* - l_i^c \cdot w_i^* + l_i^c \cdot \epsilon - (p_i(l_i^*) - p_i(l_i^c)) =$
$l_i^* \cdot w_i^* - l_i^c \cdot w_i^* + l_i^c \cdot \epsilon - ((l_i^* - l_i^c) \cdot w_i^* + \sum_{l=l_i^c+1}^{l^*} g_i(l)) =$
$(l_i^* - l_i^c) \cdot w_i^* + l_i^c \cdot \epsilon - (l_i^* - l_i^c) \cdot w_i^* - \sum_{l=l_i^c+1}^{l^*} g_i(l)) =$
$l_i^c \cdot \epsilon - \sum_{l=l_i^c+1}^{l^*} g_i(l))$

Our protocol allows the resource owners to "compete" for agents, by allowing them to offer higher parts of the utility generated on the resource. When there are many resources with similar production functions, even a small reduction of ϵ in the payment that resource r_i offers the agents results in many agents leaving the resource and switching to other resources. The number of agents who would leave r_i when r_i reduces the payment by ϵ is $\Delta_l(\epsilon) = l_i^* - l_i^c(\epsilon)$. Agent r_i's competition are the resources whose next marginal productions are higher than $w_i^* - \epsilon$ (which determine $\Delta_l(\epsilon)$). Thus, the bigger Δ_l is, the smaller l_i^c is. Since $g_i(l)$ depends only on w_i^* and p_i, the larger $\Delta_l(\epsilon)$ is, the higher $\sum_{l=l_i^c+1}^{l^*} g_i(l)$ is. We now define a competition condition that causes resource owners to lose utility, when they reduce the offered payment below what they would offer under the convergence achieved when using the suggested strategies.

Definition 3. *Highly competitive settings. Let S^* be the strategy profile where all the entities follow the suggested strategies. If for every $\epsilon > 0$ we have $l_i^c \cdot \epsilon < \sum_{l=l_i^c+1}^{l^*} g_i(l)) = \sum_{l=l_i^c+1}^{l^*}(m_i(l) - w_i^*)$, we say the setting is* highly competitive *for r_i.*

If a setting is highly competitive for r_i, then r_i cannot gain by lowering the payment it offers agents who request to use its resource, since $\Pi_i = l_i^c \cdot \epsilon - \sum_{l=l_i^c+1}^{l^*} g_i(l)) < 0$.

Theorem 5 (Resource Deviations). *Consider a highly competitive setting for r_i. Let s'_{r_i} be some strategy for r_i. Let s' be the strategy profile where all entities follow the suggested strategies, except r_i who follows strategy s'_{r_i}. Denote by $S' = \{s_{a_1}, \dots, s_{a_n}, s_{r_1}, \dots, s'_{r_i}, \dots s_{r_k}\}$. If under S' we converge to some protocol stable allocation A' such that $u_{r_i}(A') > u_{r_i}(A^{opt})$, then $P(A') = P(A^{opt})$, and A' is also an optimal allocation. In other words, if r_i managed to increase its utility, then an optimal allocation is still reached.*

Proof. Let S^* be the strategy profile where all the entities follow the suggested strategies. We denote by $l_i^* = |A_{r_i}^{opt}|$ and by $w_i^* = m_i(l_i^*)$ the payment r_i offers agents to which it is allocated. Under S^*, we have $u_{r_i}^* = p_i(l_i^*) - w_i^* \cdot l_i^*$. Let A' be the protocol stable allocation reached under S'. We denote by $n_l = m_l(|A'_{rl}| + 1)$ the next marginal production on resource r_l, and $n' = max_{l=0}^n n_l$. We denote by w' the minimal payment r_i publishes at the beginning of any round after A' is reached.

A' is protocol stable, so we have $w' > n'$, since otherwise one of the agents may try to request access to some resource that would pay him more than he is offered on r_i. If $w_i^* > w' = w_i^* - \epsilon$ for some $\epsilon > 0$, then since the setting is highly competitive, $u_{r_i}(A') < u_{r_i}(A^{opt})$ for any A^{opt}. If $w_i^* < w' = w_i^* + \epsilon$ for some $\epsilon > 0$, we denote by l^a the maximal number of agents such that $m_i(l^a) > w'$. We denote by l^c the number of agents to which A' allocates r_i, so $l^c = |A'_{r_i}|$. Since the marginal production on r_i diminishes as more agents are allocated that resource, $l^a < l^*$. The utility of r_i is $u_{r_i}^c = p_i(l^c) - \sum_{j=1}^n d_i(a_j) < p_i(l^c) - w' \cdot l^c$.

If $l^c \leq l^a$, then
$u_{r_i}^* - u_{r_i}^c > p_i(l_i^*) - w_i^* \cdot l_i^* - (p_i(l^c) - w' \cdot l^c) =$
$p_i(l_i^*) - w_i^* \cdot l_i^* - p_i(l^c) + w' \cdot l^c =$
$\sum_{l=1}^{l^c} m_i(l) + \sum_{l=l^c+1}^{l_i^*} m_i(l) - w_i^*(l^c + (l_i^* - l^c)) - \sum_{l=1}^{l^c} m_i(l) + w' \cdot l^c =$
$\sum_{l=l^c+1}^{l_i^*} m_i(l) - w_i^* \cdot l^c - w_i^* \cdot (l_i^* - l^c) + w' \cdot l^c$.
But, $w' > w_i^*$, so $u_{r_i}^* - u_{r_i}^c > \sum_{l=l^c+1}^{l_i^*} m_i(l) - w_i^* \cdot (l_i^* - l^c)) = \sum_{l=l^c+1}^{l_i^*}(m_i(l) - w_i^*)$.
But $w_i^* = m_i(l_i^*)$, and the marginal production diminishes, so for any $l < l*$ we have $(m_i(l) - w_i^*) \leq 0$. Thus, $u_{r_i}^* - u_{r_i}^c > 0$, and r_i's utility only decreases in A'.

If $l^c > l^a$, then $p_i(l^c) - w' \cdot l^c = \sum_{l=1}^{l^c}(m_i(l) - w') = \sum_{l=1}^{l^a}(m_i(l) - w') + \sum_{l=l^a+1}^{l^c}(m_i(l) - w')$. But since resource l^a was the last one where $m_i(l^a) \geq w'$, and the marginal production diminishes, the second sum is of non-positive values. Thus, $u_{r_i}^c = p_i(l^c) - w' \cdot l^c \leq \sum_{l=1}^{l^a}(m_i(l) - w') = p_i(l^a) - w' \cdot l^a$. But $p_i(l^a) - w' \cdot l^a$ is the utility of r_i for a choice of $l^c = l^a$, and even for that choice we have shown that r_i's utility is not better than $u_{r_i}^*$.

Thus, r_i cannot gain by having $w' < w_i^*$ (due to the highly competitive setting for r_i), and cannot gain by having $w' > w_i^*$. So r_i chooses bids such that $w' = w_i^*$. Since w' is the minimal payment offered, if any agent is offered a higher payment, r_i's utility decreases. Thus, r_i has no incentive to deviate from the suggested strategy. □

The above theorems show that in highly competitive settings, strategic behavior cannot cause us to reach a sub-optimal allocation. Our results so far only show that rational behavior cannot cause us to converge to a protocol stable allocation that is non-optimal.

7. EXPECTED TIME TO CONVERGENCE

We now consider the expected time to convergence of the above protocol. Consider a setting with n agents Ag and k resources R, interacting using the suggested strategies. We first consider the case where the marginal returns for a certain resource $r_{k'} \in R$ are higher than any other resource, for any number of agents who are allocated this resource. That is, for any resource $r_k \in R$, $r_k \neq r_{k'}$ and any number of agents $0 \le i, j \le n$ we have $m_{k'}(i) \ge m_k(j)$. In this case the optimal allocation allocates $r_{k'}$ to all the agents: for any agent $a \in A$, $A^{opt}(a) = r_{k'}$.

Let X be a random variable indicating the number of rounds to convergence to A^{opt}, and X_i be a random variable indicating the number of rounds until agent a_i is allocated resource $r_{k'}$. Once it is allocated that resource, it is always allocated that resource, since the marginal production on any other resource is lower, so the agent would not switch. In the optimal allocation A^{opt}, all the agents are allocated the resource k', so we have $X = max_i\{X_i\}$.

LEMMA 1. $P(X_i > tk \ln n) \approx n^{-t}$

PROOF. Once a single agent requests access to the resource k' he switches to using that resource, and never switches to a different resource. Thus, an agent has a chance of $1 - \frac{1}{k}$ of missing the resource k' at any given round. Thus, $P(X_i > tk \ln n) = (1 - \frac{1}{k})^{tk \ln n} = (1 - \frac{1}{k})^{k^{t \ln n}} \approx e^{-t \ln n} = n^{-t}$ □

We now bound the probability that the convergence for *all* the agents takes more than $tk \ln n$ steps.

LEMMA 2. $P(X > tk \ln n) \approx n^{1-t}$

PROOF. We apply the union bound, and use Lemma 1. $P(X > tk \ln n) = P(max_i\{X_i\} > tk \ln n) = P(\exists X_i \text{ that } X_i > tk \ln n) \le \sum_{i=0}^{n} P(X_i > tk \ln n) = \sum_{i=0}^{n} n^{-t} = n^{1-t}$. □

This gives us a bound on the probability that the convergence takes more than $tk \ln n$ rounds. We now analyze the *expected* time to convergence. We define $q_j = P(max_i\{X_i\} = j)$ and $p_j = P(max_i\{X_i\} > j)$. We have $E(X) = \sum_{j=0}^{\infty} j \cdot q_j$.

THEOREM 6. *The expected time of convergence is $O(k \ln n)$, or more precisely, $E(X) < 4k \ln n$.*

PROOF. $E(X) = \sum_{j=0}^{\infty} j \cdot q_j = 0 \cdot q_0 + 1 \cdot q_1 + 2 \cdot q_2 + 3 \cdot q_3 + \ldots = (q_1 + q_2 + \ldots) + (q_2 + q_3 + \ldots) + (q_3 + q_4 + \ldots) + \ldots = \sum_{j=1}^{\infty} q_j + \sum_{j=2}^{\infty} q_j + \sum_{j=3}^{\infty} q_j + \ldots = p_1 + p_2 + p_3 + \ldots = \sum_{j=1}^{\infty} p_j = \sum_{j=1}^{sk \ln n} p_j + \sum_{j=sk \ln n+1}^{2sk \ln n} p_j + \sum_{j=2sk \ln n+1}^{3sk \ln n} p_j + \ldots$

The function p_i is monotonically dropping in i. Denote $x = sk \ln n$, and we get:

$E(X) \le \sum_{j=1}^{x} p_1 + \sum_{j=x+1}^{2x} p_x + \sum_{j=2x+1}^{3x} p_{2x} + \ldots \le \sum_{j=1}^{sk \ln n} 1 + \sum_{j=sk \ln n+1}^{2sk \ln n} n^{1-s} + \sum_{j=2sk \ln n+1}^{3sk \ln n} n^{1-2s} + \ldots$
$= (sk \ln n) \cdot (1 + \frac{n}{n^s} + \frac{n}{n^{2s}} + \frac{n}{n^{3s}} + \ldots)$

For $s = 2$ and $n > 2$ we get: $E(X) \le (2k \ln n) \cdot (1 + \frac{1}{n} + \frac{1}{n^3} + \frac{1}{n^5} + \ldots) \le (2k \ln n) \cdot 2 = 4k \ln n$. □

This proves a convergence time of $O(k \ln n)$ for an extremely unbalanced case. We now show that this unbalanced case is the worst case in terms of expected time to convergence. The analysis relies on Lemma 1. Since agents randomly choose a resource, if the agents are required to choose one specific resource, the probability of choosing that resource in a certain round is low. However, when the optimal allocation has several resources which are allocated to some agents, since the agents are identical, during the first rounds agents have a higher probability of randomly choosing one of these resources. Thus, the bounds for the probability that $X_i > tk \ln n$ decreases. Using the same methods as in Section 2, we can see that the probability that the procedure does not converge to the optimal allocation after $tk \ln n$ rounds, $P(X > tk \ln n)$, also decreases. Since $E(X)$ sums smaller values, it is also smaller. Thus, the procedure converges to the optimal allocation in $O(tk \ln n)$ rounds.

8. CONCLUSION

We have studied a setting of a multiagent resource allocation problem where the marginal production of each resource is diminishing, and suggested a market-based protocol for it. Rational action in highly competitive settings for resource owners would cause convergence to the optimal allocation, even for self-interested entities. The procedure has several desirable properties: it rapidly converges to the optimal allocation, and, as opposed to VCG, it is fully budget-balanced and operates without a central trusted authority.

It remains an open question to see whether similar approaches can be used in other domains as well.

9. ACKNOWLEDGMENT

This work was partially supported by Israel Science Foundation grant #898/05.

10. REFERENCES

[1] Y. Chevaleyre, P. E. Dunne, U. Endriss, J. Lang, M. Lemaitre, N. Maudet, J. Padget, S. Phelps, J. A. Rodriguez-Aguilar, and P. Sousa. Issues in multiagent resource allocation. *Informatica*, 30:3–31, 2006.

[2] U. Endriss, N. Maudet, F. Sadri, and F. Toni. Negotiating socially optimal allocations of resources. *JAIR*, 25:315–348, 2006.

[3] J. Feigenbaum and S. Shenker. Distributed algorithmic mechanism design: Recent results and future directions. In *6th Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pages 1–13. ACM Press, 2002.

[4] P. Gradwell and J. Padget. Distributed combinatorial resource scheduling. In *AAMAS Workshop on Smart Grid Technologies (SGT-2005)*, pages 295–308, 2005.

[5] T. Groves. Incentives in teams. *Econometrica*, pages 617–631, 1973.

[6] B. Heydenreich, R. Müller, and M. Uetz. Decentralization and mechanism design for online machine scheduling. In *10th Scandinavian Workshop on Algorithm Theory (SWAT 2006)*, 2006.

[7] G. Jonker, J.-J. C. Meyer, and F. Dignum. Towards a market mechanism for airport traffic control. In *12th Portuguese Conference on Artificial Intelligence (EPIA 2005)*, pages 500–511, 2005.

[8] A. Mas-Collel, W. Whinston, and J. Green. *Microeconomic Theory*. Oxford University Press, 1995.

[9] E. Ogston and S. Vassiliadis. A peer-to-peer agent auction. In *1st Int. Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 151–159, 2002.

[10] T. W. Sandholm. An implementation of the contract net protocol based on marginal cost calculations. In *12th Int. Workshop on Distributed Artificial Intelligence*, pages 295–308, 1993.

[11] T. W. Sandholm. Contract types for satisficing task allocation: Theoretical results. In *AAAI 1998 Spring Symposium: Satisficing Models*, 1998.

[12] O. Shehory and S. Kraus. Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101(1–2):165–200, 1998.

[13] B. Yu and M. P. Singh. Distributed reputation management for electronic commerce. *Computational Intelligence*, 18(4):535–549, 2002.