RANDOMIZED DISTRIBUTED ACCESS TO MUTUALLY EXCLUSIVE RESOURCES

MOSHE DROR, BRUCE HARTMAN, GARY KNOTTS, AND DANIEL ZENG

Many systems consist of a set of agents which must acquire exclusive access to resources from a shared pool. Coordination of agents in such systems is often implemented in the form of a centralized mechanism. The intervention of this type of mechanism, however, typically introduces significant computational overhead and reduces the amount of concurrent activity. Alternatives to centralized mechanisms exist, but they generally suffer from the need for extensive interagent communication. In this paper, we develop a randomized approach to make multiagent resource-allocation decisions with the objective of maximizing expected concurrency measured by the number of the active agents. This approach does not assume a centralized mechanism and has no need for interagent communication. Compared to existing autonomous-decentralized-decision-making (ADDM)-based approaches for resource-allocation, our work emphasizes achieving the highest degree of agent autonomy and is able to handle more general resource requirements.

1. Introduction

In many applications, a set of agents must acquire exclusive access to resources from a shared pool. Consider, for instance, a network of computers with access to shared peripheral devices [11]. Peripheral devices in such systems (e.g., tape drives and printers) often have characteristics which require that they be used by only one computer at a time. In addition, the high cost of these peripheral devices can create an economic incentive to keep them as busy as possible.

Consider another example in project management [1]. A project is typically broken down into a set of activities that correspond to the individual tasks that must be completed as part of the project. No activity can begin until the activities which logically precede it are completed and most activities must acquire resources before they can begin. Some of these resources may be such that they can be used for only one activity at a time. If one considers an agent to be equivalent to a project activity [8], then a project is similar to the network of computers described previously. Both are systems of agents where the agents require some number of mutually exclusive resources.

In the literature, the multiagent resource-allocation problem is generally being solved using two different approaches [6, 9]. The first approach assumes the presence of a centralized mechanism that takes as input agent resource requests and generates a system-wide optimal or near-optimal resource assignment. The second approach is based on decentralized control where individual agents or small groups of agents make resource-allocation decisions "locally" given partial information regarding resource availability and other agents' status.

A centralized approach is computationally expensive and therefore does not scale well as the number of the involved agents increases (e.g., when a large number of agents are to be involved in a complex negotiation scheme [2]). A decentralized approach that involves extensive interagent communications suffers similar problems with computation and communication overhead. In this paper, we focus on a special class of the decentralized approach that allows the agents to act autonomously and unilaterally in accessing resources with no interagent communication. This type of approach is called autonomous decentralized decision making (ADDM) in the literature [11].

The "slotted Aloha" protocol, widely used in computer networks, can be viewed as a special type of ADDM [12]. We summarize this approach in nontechnical terms. Several agents are competing for one single unit of resource. When resource congestion, that is, two or more agents wanting to reserve the same resource, occurs, each agent will wait for a random period of time before retrying. The key control variable is the one that governs the generation of random waiting times. When the waiting times are long on average, the probability of congestion recurring is low but resource utilization is also low. When the waiting times are short on average, the probability of congestion is high with potentially higher resource utilization.

Previous work done by Pasquale demonstrates that an ADDM process using randomization is an effective multiagent resource-allocation mechanism to deal with more than one unit of resources [11]. This mechanism allows the agents to act unilaterally in accessing resources, and therefore eliminates the need for a centralized allocation mechanism. Furthermore, the mechanism does not introduce the need for the agents to communicate directly with one another. Pasquale's method is based on the calculation of a probability α that determines the likelihood with which an agent will choose to bid or not bid for the resource it needs. Pasquale shows how each agent, using the number of agents in the system and the number of resources, can determine for itself an optimal value of α that guarantees maximum performance defined as the weighted difference between resource utilization and resource congestion.

The objective assumed in Pasquale's work is applicable to applications where resource utilization is the primary consideration. For instance, in the above example concerning the computers sharing networked peripheral devices, maximizing the utilization of these devices in a decentralized manner without communication can be achieved by Pasquale's method. In many other applications, however, objectives other than resource utilization are more relevant.

Consider the following two application settings.

(a) In project management, many activities may be competing for certain units of resources from a common pool (e.g., man power). A typical project management objective

in this context is to maximize the number of the activities that can acquire the resources needed successfully and in turn be completed.

(b) In parallel and distributed computing, a number of computational tasks are waiting to be executed. Each task needs exclusive access to a CPU and a certain amount of internal memory space. The most critical consideration in scheduling these tasks on given computational resources is to maximize the number of tasks that can be completed.

In the above two settings and many other practical resource-allocation applications, the objective is to maximize the number of activities or tasks that can acquire needed resources (as opposed to maximizing resource utilization). Pasquale's work, which is resource-centric by nature, is not applicable to these task- or activity-centric application settings.

In this paper, we study an ADDM-based multiagent resource-allocation mechanism aimed at maximizing the expected number of activities simultaneously in progress. The rest of the paper is structured as follows. Section 2 reviews Pasquale's work on ADDM in detail. Section 3 presents the model and decision procedures of our ADDM-based multiagent resource-allocation mechanism. This mechanism differs from the existing ADDM work in targeting a different system objective relevant to many important applications such as project management and parallel computing. Our mechanism is also capable of handling more general types of resource requirements than existing approaches. Related computational issues are discussed in Sections 4 and 5. Section 6 provides bounds for situations where the agents require multiple resources at the same time. Section 7 discusses the contribution of our research in the broader context of multiagent systems research and proposes two computational measures for agent autonomy. We conclude the paper in Section 8 with a summary of our research.

2. Review of Pasquale's ADDM model

Pasquale considers a system in which there are s units of a single resource and N agents compete for exactly one of the s units of resource [11]. The duration for which any unit of resource is utilized is one period long. The competition for resources is totally decentralized. The only information common to the agents at the instant of decision making is s, N, and β, the relative benefit of advancing by using a unit of resource relative to the cost of failing to proceed due to a conflict over a unit of the resource.

The ADDM procedure consists of a single decision making instant. Each agent independently follows a two-step strategy.

ADDM procedure. (1) The agent decides whether or not to bid at this time by randomizing the binary choice with probability α of bidding.

(2) If the agent decides to bid, then it selects one of the s units to bid for using a uniform probability distribution. That is, it chooses a unit to bid for by sampling with equal probability $1/s$ of choosing each unit. If the agent does not choose to bid, it waits until the next decision-making instant.

There are three possible resource states: *utilized*, *congested*, and *wasted*. A unit of resource for which there is only one bidding agent is *utilized*. If a unit is bid for by two or more agents, the unit is *congested*. Units for which no bid is received are *wasted*. Our

work focuses on the agents, so we provide three analogous definitions for the three possible states of an agent: *active*, *blocked by congestion*, and *blocked by choice*. If an agent is the only one to bid for a particular unit of resource, we will call this agent *active*. If an agent is a member of a set of two or more agents that bid for the same unit of resource, we will call this agent and all others in the set *blocked by congestion*. If an agent fails to bid, we will call it *blocked by choice*.

The *gain G* of the system is a random variable given by

$$G = U - \beta C, \tag{2.1}$$

where U is the number of units of resource utilized and C is the number of congested units. Pasquale's objective is to maximize the gain where β is a design parameter used to weight the relative benefit of utilization versus congestion. When $\beta = 0$, utilization is maximized with no concern for the amount of congestion. When $\beta = 1$, utilization and congestion are equally important.

The only decision variable in Pasquale's ADDM model is α, the probability with which any agent chooses to bid. He demonstrated how to calculate the optimal bid probability α^* for the agents using his notion of gain, and how to calculate the maximum expected gain EG^* as a function of N, s, and β [11]. A measure of the adequacy of resources he proposes is the maximum expected gain available per unit of resource, EG^*/s, as a function of the number of agents competing for the resource units, and he computes the optimal number of agents to compete for exactly s units of resource in terms of this expected gain per unit of resource.

3. ADDM for concurrency maximization

In this section, we develop a new ADDM model with the objective of maximizing the expected number of active agents, that is, expected concurrency. This model can also handle more general resource requirements than Pasquale's model.

3.1. Decision procedures. Let N be the total number of agents in the system and let n denote the number of agents which wish to be active where $1 \leq n \leq N$. Let s be the number of units of available resources and let t_i be the number of units required by the ith agent where $1 \leq t_i \leq s$ (we assume that the resource requirements of the individual agents are feasible). Observe that if $t_i = 0$, we can simply eliminate this agent from the set of those which will consider whether or not to bid. We define a revised decision-making procedure for an individual agent as follows.

Revised ADDM procedure. (1) Decide with probability α whether or not to bid for sufficient units of resource to proceed.

(2) Select at random a subset S_i of size t_i from the set S of s units of the resource which are available. In the random selection, assign some positive probability mass to each possible subset of size t_i. (Various probability mass assignments are possible. In this paper, we discuss only one type of assignment. See Section 4.2.)

The revised ADDM procedure is readily applicable to situations where all agents are independent. Whenever resource-allocation decisions need to be made, this procedure

can be invoked to assign resources to needing agents. Agents that fail to acquire needed resources will bid in later invocations if necessary. Every invocation will be conducted as completely independent of each other.

This procedure can also be easily adapted to applications where simple forms of agent interdependencies, such as precedence constraints, are present (e.g., typical of project management). In such cases, only agents that have no uncompleted predecessors will participate in bidding of resources. Agents that are waiting for some predecessors to finish first will participate in later rounds of bidding when these predecessor agents are completed.

3.2. Objective function. The objective function appropriate for our situation is the expected number of active agents. In particular, we are interested in the expected number of agents which are active at the same point in time. Thus, we define EC, the expected concurrency, as an adaptation of the gain function defined in Section 2. We define A to be the total number of agents which are active such that A is equal to the sum of independent 0–1 random variables A_i where A_i is one when the ith agent is active and zero otherwise. We define B as the total number of agents blocked by congestion such that B is equal to the sum of independent 0–1 random variables B_i where B_i is one when the ith agent is blocked by congestion and zero otherwise. We define X to be the total number of agents blocked by choice such that X is equal to the sum of independent 0–1 random variables X_i where X_i is one when the ith agent is blocked by choice and zero otherwise. Note that when $X_i = 1$, $B_i = 0$ by definition. However, when $X_i = 0$, B_i may be either zero or one. We let $D_i = B_i + X_i$ and observe that $D_i \in \{0, 1\}$. We define $D = B + X$ and define the gain as

$$G = A - \beta D. \tag{3.1}$$

When $t_i = 1$ for all i, this model is similar to Pasquale's model. U and A represent the same quantity in our model and Pasquale's, since for Pasquale each agent requires exactly one unit of resource. The blocked agents are those which conflict over a single resource unit (i.e., they congest the resource). However, $B \neq C$, because several agents could congest the same resource.

In addition, in our gain definition, we include the number of agents that do not bid, since they adversely affect our objective. One could reduce the "gain" by assigning some positive, nonzero value to β. Observe, however, that $D = N - A$ at all times, since each agent either bids or does not, and if it bids, it becomes either active or blocked by congestion. Thus, $G = A - \beta(N - A)$ and maximizing its expected value is equivalent to maximizing the expected number of concurrently active agents EC. In other words, the value of β will have no impact on the decision procedure.

4. Calculation of the probability of agent activation

In this and the next section, we derive explicit solutions for our ADDM model. We first determine the probability that an agent will be active under ADDM. We begin with the

assumption that each agent demands one unit of resource and then extend this to more general resource requirements.

4.1. Each agent demands a single unit of resource. We begin by determining the probability of activation, Pr(activation), for the case where we consider only those agents that have decided to bid then extend the result to consider all agents in the system.

Observation 4.1. In the case where we consider only the agents that have chosen to bid, the probability of a particular agent's activation is given by

$$\Pr(\text{activation}) = \left(1 - \frac{1}{s}\right)^{k-1},\tag{4.1}$$

where $k \leq N$ is the number of agents in the bidding set, s is the number of resources, all units of resource are interchangeable, $k, N, s \geq 1$, and each agent needs exactly one unit of resource.

Proof. First, consider that this problem is analogous to having each of k agents draw one of s different numbers from a hat where each number in the hat represents one unit of resource and these units are interchangeable. Since all agents select resources simultaneously, this is equivalent to drawing the numbers with replacement. Based on the analogy, we prove that the probability that any particular agent will get the resource it needs without conflict with any other agent is $(1 - 1/s)^{k-1}$.

Assume that agents are indexed by i and resources by j. The probability that the ith agent will draw the jth resource is $1/s$. After the ith agent draws a resource, it is returned to the hat and the $(i+1)$th agent draws next. The probability that the $(i+1)$th agent will draw a resource different than the one drawn by the ith agent is the total number of resources less one divided by the total number of resources; $(s-1)/s$.

After the $(i+1)$th agent draws, the resource is again returned to the hat prior to the draw of the next agent. This continues until all agents have drawn a resource. Given that the drawings are independent events, we find that the probability that all agents will draw a resource different from that of the ith agent is the product of the probabilities that each individual agent will draw something other than the resource drawn by the ith agent; $(1 - 1/s)^{k-1}$. $\qquad\square$

Observation 4.2. In the case where we consider all agents in the system (i.e., bidding and nonbidding), the probability of activation of a given agent is:

$$\Pr(\text{activation}) = \alpha \sum_{k=0}^{N-1} \binom{N-1}{k} \alpha^k (1-\alpha)^{N-1-k} \left(1 - \frac{1}{s}\right)^k,\tag{4.2}$$

where α is the probability that an agent will choose to bid and is the same for all agents in the system, k is the number of agents which choose to bid including the given agent, N is the total number of agents in the system, s is the number of resources, all units of resource are interchangeable, $k, N, s \geq 1$, and each agent needs exactly one unit of resource.

Proof. Observation 4.1 gives the probability that an agent will be activated given that only bidding agents were under consideration. In that result, we saw that the probability of activation of any particular agent was a function of the number of other agents in the bidding set. However, the number of other agents in the bidding set cannot be known prior to bidding since it is a function of the probability α that agents will choose to bid. Thus, to extend our previous result to the case where we do not know the number of bidding agents, we need a means by which to determine the likelihood that a particular number of agents will choose to bid. The process of agent activation as described thus far is a Bernoulli process. A Bernoulli process must satisfy the following conditions.

(1) There is some number of identical trials of a random experiment for which there are only two possible outcomes.
(2) The possible outcomes are mutually exclusive (i.e., the occurrence of one outcome guarantees the "nonoccurence" of the other outcome).
(3) Each trial is independent from all other trials (i.e., the outcome of any particular trial does not change the probabilities associated with the outcomes of any other trial).

The outcome of each trial is generally referred to as a success or failure where success and failure can be assigned arbitrarily to the possible outcomes. In our case, each agent behaves identically in reaching a decision to bid or not bid and the choice each agent makes is equivalent to one trial of a random experiment with the two possible outcomes. If we refer to a decision to bid as a success, then we find that the probability of success is given by α, and the probability of not bidding, in other words failure, is $1 - \alpha$. Given this, the bid decision process consists of $N - 1$ identical trials of a random experiment. Furthermore, any agent that chooses to bid cannot also choose not to bid and vice versa. Thus, the possible outcomes are mutually exclusive. Finally, all agents make the decision to bid independent of all others. Thus, the trials are independent.

Given that we have a Bernoulli process, the binomial theorem provides a formula by which we can determine the probability of a specified number of successes in a given number of trials. Applying the binomial formula, we have

$$\Pr\left(\text{exactly } k \text{ of } N - 1 \text{ bid} \mid \Pr(\text{bid}) = \alpha\right) = \binom{N-1}{k} \alpha^k (1 - \alpha)^{N-1-k}, \qquad (4.3)$$

where k is some arbitrary number of agents such that $0 \leq k \leq N - 1$. We use $N - 1$ because the agent which wishes to determine the probability of its own activation will do so in consideration of the number of the agents *other than itself* which might choose to bid. With this in hand, we now observe that for the agent which has chosen to bid, the probability of activation is the product of the probability that k agents choose to bid and the probability that all k choose a resource different from that selected by the particular agent, which is given by

$$\Pr\left(\text{activation}\right) = \Pr\left(\text{exactly } k \text{ of } N - 1 \text{ bid} \mid \Pr(\text{bid}) = \alpha\right) \left(1 - \frac{1}{s}\right)^k. \qquad (4.4)$$

Now substituting for $\Pr(\text{exactly } k \text{ of } N - 1 \text{ bid} | \Pr(\text{bid}) = \alpha)$ from the binomial formula, we have

$$\Pr(\text{activation}|k \text{ agents choose to bid}) = \binom{N-1}{k} \alpha^k (1-\alpha)^{N-1-k} \left(1 - \frac{1}{s}\right)^k. \quad (4.5)$$

The previous result gives us the probability that a particular agent will be activated given that exactly k other agents choose to bid. To arrive at the desired result, we must now consider all possible values of k from 0 to $N - 1$. Since these possibilities are disjoint events, our previous result summed over all possible values of k yields

$$\Pr(\text{activation}) = \sum_{k=0}^{N-1} \binom{N-1}{k} \alpha^k (1-\alpha)^{N-1-k} \left(1 - \frac{1}{s}\right)^k. \quad (4.6)$$

Finally, we recognize that the probability of agent activation is also a function of its own decision to bid and therefore the previous result is multiplied by α to yield

$$\alpha \sum_{k=0}^{N-1} \binom{N-1}{k} \alpha^k (1-\alpha)^{N-1-k} \left(1 - \frac{1}{s}\right)^k. \quad (4.7)$$

$\square$

4.2. Each agent demands multiple units of resource. In this section, we extend our results to consider a system in which agents $1, 2, \ldots, N$ require $t_1, t_2, \ldots, t_N$ units of resource, respectively.

As we discussed previously, in the ADDM procedure some probability of selection of each subset must be determined. This is because each autonomous agent does not know the needs of the other agents. Therefore it must make some prior probability assumption. We assume that the values of t_i are such that $1 \le t_i \le s$, but that $\sum_i t_i > s$.

In this paper, we assume that each agent assigns the probability mass in two steps.

   (i) First, the s possible values of the demand by other agents, from 1 to s, are equally likely. (A requirement of zero units is not considered, because the analysis is restricted to agents which require resources.)
  (ii) Second, independently, each subset of the size k chosen is equally likely. Since there are $\binom{s}{k}$ such subsets, the second-stage probability is the reciprocal of that quantity.

Other methods are also possible. For instance, one could assume that all possible subsets of the s units of resources are equally likely. This yields a different probability mass for each subset. For example, consider $s = 3$. If each subset is equally likely, then the probability of each is 1/8 (or 1/7 if we do not allow the empty subset); so the probability that $t_i = 2$ is 3/8 (or 3/7). If each size is equally likely, then $\Pr(t_i = 2) = 1/4$ (or 1/3 if size 0 is not allowed). Since there are 3 subsets of size 2, the probability of a specific subset with two members is $(1/4)(1/3) = 1/12$ (or $(1/3)(1/3) = 1/9$). So the two probability mass functions are not the same. They therefore will give different results for the expected concurrency and for the optimal value of α. In this paper, we analyze only the density produced by the bulleted randomization procedure above.

In the following, let $C_{y,x}$ denote the number of ways of selecting x items from y items at a time where

$$C_{y,x} = \binom{y}{x} = \frac{y!}{x!(y-x)!}. \tag{4.8}$$

Observation 4.3. In the case where agents may require more than one unit of resource, the probability of agent activation for the first agent is given by

$$
\begin{aligned}
\Pr(\text{activation}) = \ & \alpha \binom{N-1}{0} \alpha^0 (1-\alpha)^{N-1} \left(\frac{1}{s}\right)^0 \\
& + \alpha \binom{N-1}{1} \alpha^1 (1-\alpha)^{(N-1)-1} \left(\frac{1}{s}\right)^1 \sum_{t_2=1}^{s} \frac{C_{s-t_1,t_2}}{C_{s,t_2}} \\
& + \alpha \binom{N-1}{2} \alpha^2 (1-\alpha)^{(N-1)-2} \left(\frac{1}{s}\right)^2 \left[\sum_{t_2=1}^{s} \sum_{t_3=1}^{s} \frac{C_{s-t_1,t_2} C_{s-t_1-t_2,t_3}}{C_{s,t_2} C_{s,t_3}} \right] \\
& + \cdots + \alpha \binom{N-1}{N-1} \alpha^{N-1} (1-\alpha)^{(N-1)-(N-1)} \left(\frac{1}{s}\right)^{N-1} \\
& \times \left[\sum_{t_2=1}^{s} \sum_{t_3=1}^{s} \cdots \sum_{t_N=1}^{s} \frac{C_{s-t_1,t_2} C_{s-t_1-t_2,t_3} \cdots C_{s-t_1-t_2-\cdots-t_{N-1},t_N}}{C_{s,t_2} C_{s,t_3} \cdots C_{s,t_N}} \right],
\end{aligned}
\tag{4.9}
$$

where α is the probability that an agent will choose to bid and is the same for all agents in the system, N is the total number of agents in the system, s is the total number of resources, all units of resource are interchangeable, $N, s \geq 1$, and each agent $1, 2, \ldots, N$ requires exactly $t_1, \ldots, t_N$ units of resource, respectively.

Proof. First, we consider only bidding agents and we further assume that there are only two bidding agents. Agent 1 needs t_1 units of resource and agent 2 needs t_2 units of resource. Agent 1 wishes to determine the probability that it will be activated. Agent 1 knows the value of t_1 and we will assume for the moment that it also knows the value of t_2. Under these circumstances, agent 1 can calculate the probability of activation as

$$\Pr(\text{activation}\,|\,\text{agent 2 needs } t_2) = \frac{C_{s,t_1} C_{s-t_1,t_2}}{C_{s,t_1} C_{s,t_2}} \tag{4.10}$$

which can be simplified to

$$\Pr(\text{activation}\,|\,\text{agent 2 needs } t_2) = \frac{C_{s-t_1,t_2}}{C_{s,t_2}}. \tag{4.11}$$

C_{s-t_1,t_2} is the number of ways that agent 2 can select t_2 units of resource from the $s - t_1$ units of resource that remain after agent 1 has made its selection of t_1 units of resource from the pool of s units of resource. C_{s,t_2} is the number of ways that agent 2 can select t_2 units of resource from a total pool of s units of resource. The ratio of these is the probability of activation.

Of course agent 1 will not know the value of t_2; thus, for agent 1 to determine its probability of activation, it must find

$$\Pr(\text{activation}|\text{agent 2 bids}) = \Pr(\text{activation}|\text{agent 2 needs } t_2)$$
$$\times \Pr(\text{agent 2 needs } t_2|\text{agent 2 bids}). \tag{4.12}$$

This value is determined by considering all possible values of t_2 that agent 2 might need, multiplied by the probability that agent 2 will choose that particular value of t_2, under the assumption that agent 2 is certain to bid.

In the following, we assume that the possible values of $t_2 \in \{1,2,\dots,s\}$ are equally likely. Again, we point out that other decision rules are possible, such as treating every subset as equally likely, so that subsets of size t_2 would have probability C_{s,t_2}^{-1} of occurring, but we calculate only for our chosen rule.

In addition, we claim that summing over the values of t_2 from 1 to s is valid since $C_{s-t_1,t_2} = 0$ for all $t_2 > s - t_1$ and, therefore, has no impact on the result of the calculation.

Having stated these facts, we conclude that

$$\Pr(\text{activation}|\text{only agent 2 bids}) = \left(\frac{1}{s}\right) \sum_{t_2=1}^{s} \frac{C_{s-t_1,t_2}}{C_{s,t_2}}. \tag{4.13}$$

Now we extend the previous result to three agents so that the general form for N agents will be apparent. Again, we note that summing over the values of t_2 and t_3 from 1 to s is valid since $C_{s-t_1,t_2} = 0$ for all $t_2 > s - t_1$ and $C_{s-t_1-t_2,t_3} = 0$ for all $t_3 > s - t_1 - t_2$. Thus, we get

$$\Pr(\text{activation}|\text{only agents 2 and 3 bid}) = \frac{1}{s^2} \sum_{t_2=1}^{s} \sum_{t_3=1}^{s} \frac{C_{s-t_1,t_2} C_{s-t_1-t_2,t_3}}{C_{s,t_2} C_{s,t_3}}. \tag{4.14}$$

We now state the general form for the probability of agent activation considering N bidding agents:

$$\Pr(\text{activation}|N-1 \text{ other agents bid})$$
$$= \frac{1}{s^{N-1}} \sum_{t_2=1}^{s} \sum_{t_3=1}^{s} \cdots \sum_{t_N=1}^{s} \frac{C_{s-t_1,t_2} C_{s-t_1-t_2,t_3} \cdots C_{s-t_1-t_2-\cdots-t_{N-1},t_N}}{C_{s,t_2} C_{s,t_3} \cdots C_{s,t_N}}. \tag{4.15}$$

To this point, we have assumed that we know the number of agents that may choose to bid, but as was the case earlier, this cannot be known prior to bidding. So our formulation must consider every possible number of bidding agents where the number of bidding agents is the result of $N-1$ Bernoulli trials and the possibility of success in each trial is given by α. As was the case for Observation 4.2, this can be done using the binomial theorem to determine the probability that 0 to $N-1$ agents will choose to bid multiplied by the probability that the bid of the ith agent will be disjoint from the other bidding agents. This is summed over all possible numbers of bidding agents from 0 to $N-1$. Finally, we recognize that the previous formulation is based on the assumption that the ith agent has chosen to bid. Thus, to get the final probability of activation, we multiply by the probability of bidding α to yield the result of the observation. □

4.3. Multiple resource pools. We have assumed to this point that agents compete for resources by placing bids for the number of units they require where all resources are in a single pool available to all agents and that all resources within the pool are interchangeable. A more realistic assumption for our purposes is that agents will require different types of resources which may not be interchangeable. Thus, we consider multiple pools of resources where individual units within pools are interchangeable, but units are not interchangeable across pools.

Assume a total of M pools of resources; each pool will contain some total number of units, $s_1, s_2, \ldots, s_M$. Agents may place separate bids for resources from any number of these pools. The quantities they bid for within each pool may be from 0 up to the total number of units within the pool. Under these circumstances, an agent will be activated if and only if all of its bids for resources are successful.

Observation 4.4. In the case where agents can bid for resources from multiple resource pools, the probability of agent activation for the ith agent is given by

$$\Pr(\text{activation}) = \prod_{m \in M_i} \Pr(\text{activation}|m), \qquad (4.16)$$

where $M_i \subset M$ consists of all resource pools from which the ith agent requires resources and $\Pr(\text{activation}|m)$ is the probability that the ith agent's bid for resources from the mth pool will be disjoint from the bids of all other agents which bid for resources from the mth pool. The value of $\Pr(\text{activation}|m)$ is computed exactly as was done for the case of a bid for a quantity of t_i resources from a single pool of resources defined in the previous section.

Proof. Since the bids for resources from multiple pools are independent events, Observation 4.4 is true based on the product rule for joint probabilities which states that for any two independent events A and B, the probability for the occurrence of both, $\Pr(A, B)$ is the product of the probability that either will occur, $\Pr(A)\Pr(B)$.   □

5. Calculation of expected concurrency and optimal bid probability

5.1. Expected concurrency. Given the assumption that the number of units of resource required by other agents are equally likely (i.e., all possible values of $t_i \in \{1, 2, \ldots, s\}$ are equally likely for all i agents), the expected number of concurrently active agents, or EC, is given by the sum of the probabilities for each agent, since the agents operate independently, and each one activated contributes 1 to the random variable A. The optimal bid probability α^* can be determined by differentiating (4.9) with respect to α and solving the first-order condition, a polynomial equation in α. Since it is not possible to find roots of polynomials of degree greater than 3, the following technique can be used to estimate α^*. First fit a fourth-degree polynomial to the probability function above. Its derivative is a cubic polynomial and there is a closed-form formula for its roots. Since the probability is 0 if $\alpha = 0$ and the probability is also 0 if $\alpha = 1$ (since every agent will bid, and there are not enough resources to go around, by assumption), any root between 0 and 1 will be a maximum. This procedure is fast enough for convenient implementation in an agent.

As mentioned previously, instead of assuming that $t_i \in \{1,2,\ldots,s\}$ are all equally likely, one could alternatively assume that every subset of required resources is equally likely. In this case, we derive EC differently. We use indexes i, j, k for agents. Each bidding agent i satisfies its resource requirements by selecting a subset S_i from the set of available resources S where $|S| = s$. The total number of resources selected by agent i is $|S_i| = t_i$. Given a system of two agents, i and k, where both agents choose to bid for resources in the set S, we wish to know the likelihood that the two will independently select disjoint subsets of resources S_i and S_k (i.e., $S_i \cap S_k = \varnothing$). The answer can be formulated as follows:

$$\phi(i,k) = \Pr\{S_i \cap S_k = \varnothing \mid i,k \text{ bid}\} = \frac{C_{s-t_n,t_i}}{C_{s,t_k}}. \tag{5.1}$$

Clearly, ϕ as formulated here is based on the condition that both i and k choose to bid. However, we wish to arrive at a formulation conditional only on i bidding. We can find this by recalling that the probability an agent will bid for resources is the same for all agents in the system and is given by α. This observation yields

$$\alpha\phi(i,k) = \Pr\{k \text{ bids}\}\Pr\{S_i \cap S_k = \varnothing \mid i,k \text{ bid}\} = \Pr\{S_i \cap S_k = \varnothing \mid i \text{ bids}\}. \tag{5.2}$$

We may now formulate a choice function η that represents the probability that either (a) S_i is disjoint from S_k given that k bids, or (b) k does not bid which is independent of whether i bids. (If k chooses not to bid, then $S_k = \varnothing$ and is, therefore, disjoint from S_i by definition.) To do this, first recognize that the probability that k will choose not to bid is $1 - \alpha$. Using this and the result previously obtained, we have

$$\eta(i,k,\alpha) = \begin{cases} 1 - \alpha + \alpha\phi(i,k) & \text{if } i \neq k, \\ 1 & \text{otherwise.} \end{cases} \tag{5.3}$$

We note that the actions of agents are independent and, therefore, the probability that the bid of agent i is disjoint from the bids of all other agents is the product of the probabilities that the bids of all pairs of agents are disjoint. Thus, we have

$$W(i,\alpha) = \Pr\{\forall k,\ S_i \cap S_k = \varnothing \mid i \text{ bids}\} = \prod_{k=1}^{N} \eta(i,k,\alpha) \tag{5.4}$$

which represents the probability that all other S_k are disjoint from S_i conditional on i bidding.

Observe that W is a decreasing function of α for any i. This is because the ϕ are between 0 and 1, so as α increases, the factors $\alpha(\phi - 1)$ decrease in size while still nonnegative. The product must therefore decrease. Figure 5.1 graphs for $N = 7$, $t = (1,2,3,4,2,1,1)$, and $s = 10$ the first four agents' W functions.

We define

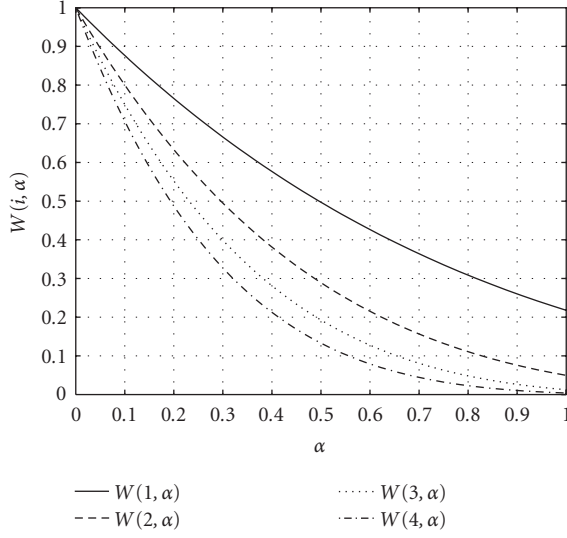$$EC_n(\alpha) = \Pr\{n \text{ is active}\} = \alpha W(n,\alpha) \tag{5.5}$$

Figure 5.1. Behavior of W as a function of α for agents with resource requirements of 1, 2, 3, and 4 when $N = 7$, $s = 10$.

and the total expected concurrency

$$EC = \alpha \sum_{j=1}^{N} W(j, \alpha). \tag{5.6}$$

Observe that the random variable $D_i = 1 - A_i$, for if an agent is not active then it must be either blocked by congestion or blocked by choice. Thus $ED = N - EC$. Then

$$EG(\alpha, \beta) = EC(\alpha) - \beta ED(\alpha) = (1 + \beta)EC(\alpha) - \beta N. \tag{5.7}$$

Observe that $EG(\alpha, 0) = EC(\alpha)$, and for positive β the graph of EG is just a negative translate of an expansion less than two times that for EC. The gain is maximized for the same bid probability α as that for EC.

Figure 5.2 shows the expected concurrency for the example of $N = 7$, resource requirements $t = (1, 2, 3, 4, 2, 1, 1)$, and $s = 10$. Figure 5.3 shows the expected concurrency for the first four individual agents which demand 1, 2, 3, and 4 units of resource, respectively.

5.2. Optimal bid probability. While the function EC can easily be computed even for agent sets with disparate resource requests, the agents need to determine α readily. Differentiating $EC(\alpha)$ and setting the first-order condition equal to zero is not easily accomplished. A large (degree $N - 1$) polynomial results whose roots between 0 and 1 must be determined.

We know that $EC(0) = 0$, because if no agents bid then none can become active. Also $EC(1) \in [0, N]$, because if all bid, one can imagine little success (for $s = 1$) or a lot of success when s is very much larger than the requirements of the agents.
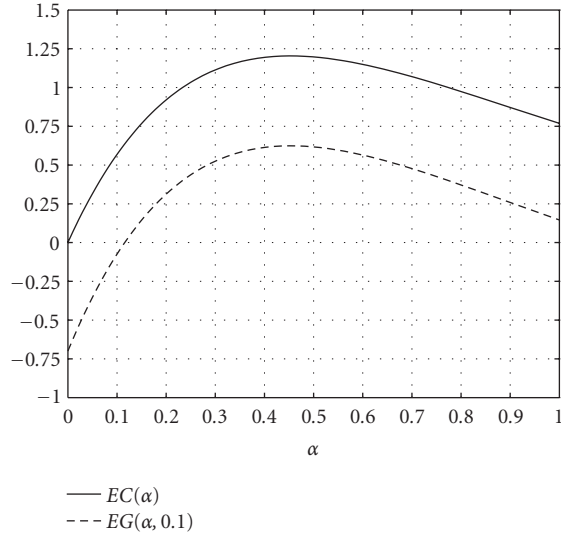
Figure 5.2. Behavior of EC as a function of α for agents with resource requirements of 1, 2, 3, and 4 when $N = 7$, $s = 10$, and $t = (1,2,3,4,2,1,1)$.
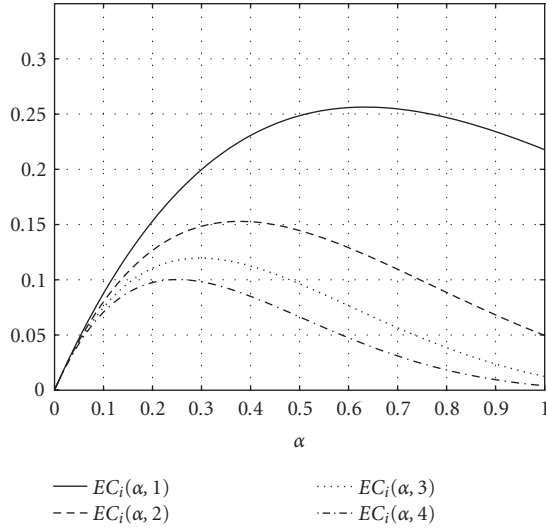

Figure 5.3. Individual $EC_n(\alpha)$ for agents with resource requirements of 1, 2, 3, and 4 when $N = 7$, $s = 10$, and $t = (1,2,3,4,2,1,1)$.

However, since the function EC is smooth, it is relatively easy to fit a fourth-degree polynomial $P(x)$ to an array of values of $EC(\alpha)$ for a modest number of values of $\alpha = x$. This allows us to estimate α^* from the five-polynomial coefficients $c_0 \approx 0,\ldots,c_4$ determined by least-squares polynomial regression using, say, 21 values of α on $[0,1]$.
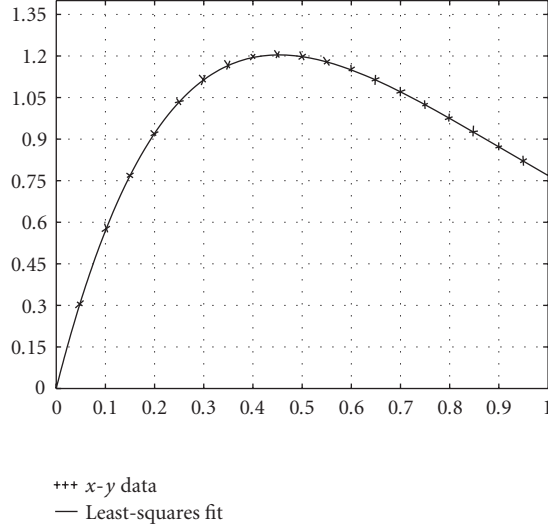
+++ x-y data
— Least-squares fit

Figure 5.4. The best fitting quartic polynomial to 21 marked values of $EC_n(\alpha)$, for agents with re-
source the requirements of 1, 2, 3, and 4 when $N = 7$, $s = 10$, and $t = (1,2,3,4,2,1,1)$.

The first-order condition is a cubic polynomial $P'(x) = c_1 + 2c_2x + 3c_3x^2 + 4c_4x^3$, whose
roots are easily determined with a closed-form calculation of complexity $O(1)$. Observe
that the degree of P is the highest possible for which a closed-form calculation of the
roots of the derivative exists. If there is no real root between 0 and 1, then the optimal bid
probability is $\alpha^* = 1$, since $\alpha = 0$ must be a minimum on $[0,1]$. The second-order suf-
ficient condition is a quadratic polynomial $P''(x) = 2c_2 + 6c_3x + 12c_4x^2$. Note that $c_4 < 0$
from the shape of the function being fit.

Figure 5.4 shows the best fourth-degree polynomial fit to the expected concurrency
for the example of $N = 7$, $t = (1,2,3,4,2,1,1)$, and $s = 10$, using 21 values of α evenly
spaced on $[0,1]$. The fit has an $R^2 = 1$ and residuals less than 0.0027 over the interval.
The estimated value of the α^* is 0.45.

The accuracy of an estimate obtained in this fashion is very satisfactory for the agents.
Since they will use it for sampling to decide whether or not to bid, it need not be exact to
give a good probability of the agent becoming active.

6. Determination of bounds for the multiple-resource case

Now we proceed to the multiple-resource scenario, in which there are r resources indexed
by q; each agent requires t_i^q units of the qth resource.

We assume for our lower bound that agents decide to bid independently for each
resource, using the optimal α^q for each resource calculated as in the previous section.
Then the probability that agent i will get all the resources it needs to proceed is

$$\Pr\{i \text{ is active}\} = \prod_{q=1}^{r} \alpha^q W^q(i, \alpha^q). \tag{6.1}$$

Define $m = \min_i \Pr\{i \text{ is active}\}$. Then the distribution of the number of active agents arising from independent trials is given by the binomial distribution with individual probability m, so that in $M = \lceil 1/m \rceil$ attempts, the expected number of times that an arbitrary agent would be able to start will be at least 1.

Since each agent is proceeding completely independently, we can write the expected concurrency as

$$EC = \sum_{i=1}^{N} \Pr\{i \text{ is active}\} = \sum_{i=1}^{N} \prod_{q=1}^{r} \alpha^q W^q(i, \alpha^q). \tag{6.2}$$

Observe that this strategy requires the bidders to calculate a whole set of values of α^q, one for each resource; this loop increases complexity by a factor of $O(r)$.

The complexity could be lowered by calculating a single α to use for all resources and all agents, as in the previous case. One could use the most congested resource measured by the ratio of the sum of number of units demanded by each agent to the total number of units available.

7. ADDM and agent autonomy

The ADDM mechanism provides a means by which agents can make independent decisions in regard to accessing resources from a set of shared resources. In the case where the agent actions can be coordinated without a central controller or interaction with other agents, their behavior is autonomous.

There have been numerous definitions for autonomy in the literature. Rozenblit defines autonomy as the ability of a system to function independently subject to its own laws and control principles [13]. Jennings and Wooldridge define it to be an ability to solve problems without interaction with humans or other entities [7]. Davidssonet et al. define it as an ability to interact independently with the environment through sensors and effectors [3]. Haddawy defines it as an ability to manipulate the environment in the course of satisfying needs and desires [5]. Maes defines it as an ability to operate without intervention in the process of relating sensory inputs to motor outputs in such a way as to achieve goals [10].

Although all the previous definitions capture the same essence of autonomy, none is particularly well suited for the purpose of deriving a practical, objective measure. A definition which is easier to operationalize can be found in [11], where it is suggested that coordinating mechanisms which best promote autonomy are those which support fast decision making with limited shared information in a minimally conflicting manner. If we focus on computer-based environments, Pasquale's statement appears to be based on the generally accepted notion that we wish to use time and space as wisely as possible in any computer-based system. Furthermore, we can infer from the definitions of autonomy given earlier that 100% autonomy exists only where no amount of time or space is devoted to the coordination of agents. Thus, we provide the following two measures of autonomy.

(i) Relative space autonomy. Given a specific problem setting, the relative space autonomy is a function of the amount of computer memory required for the coordination of

agents relative to the total amount of memory required by the system. This can be computed as $1 - CM/TM$, where CM is the memory required for agent coordination and TM is the total memory. To make the measure more precise, we compute the *relative space autonomy* using the worst-case analysis; that is, for a given problem description, given a problem size of, say, n agents, what is the minimum value of $1 - CM/TM$ taken over all problems of size n? This term is expressed as a function of n.

(ii) Relative time autonomy. Given a specific problem setting, the relative time autonomy is a function of the amount of CPU time required for the coordination of agents relative to the total amount of CPU time required by the system. This can be computed as $1 - CT/TT$ where CT is the CPU time required for agent coordination and TT is the total CPU time required by the system. As in relative space autonomy, this measure is made more precise by computing the *relative time autonomy* using the worst-case analysis; that is, for a given problem description, given a problem size of, say, n agents (represented precisely by the problem input requirements), what is the minimum value of $1 - CT/TT$ taken over all problems of size n? This term is expressed as a function of n.

These measures of autonomy are obviously based on ideas derived from the field of algorithm analysis as used to determine the computational complexity of algorithms. One of the primary advantages of this approach is the existence of well-defined formalisms for conducting such analyses. Although it is beyond the scope of this paper, we believe that future research should be dedicated to furthering the establishment of a generally accepted objective measure of autonomy.

8. Conclusion

In this work, we have developed a randomized ADDM approach to make multiagent resource-allocation decisions with the objective of maximizing the number of the active agents. We have shown the means by which an optimal value of decision variable (selected by each agent) α can be calculated for this problem and extended it to more general resource requirements. This idea has potential for application to resource-constrained project scheduling problems and parallel computing where maximization of active agents is of essence [4, 8]. We have also introduced two preliminary objective measures for the strength of autonomy of a system of agents. Such measures can potentially serve as the foundation of evaluating and comparing various multiagent resource-allocation mechanisms in a computational context.

References

[1]   J. Blazewicz, W. Cellary, R. Slowinski, and J. Weglarz, *Scheduling Under Resources Constraints: Deterministic Models*, Baltzer, Basel, 1986.

[2]   B. Chaib-Draa, B. Moulin, R. Mandiau, and P. Millot, *Trends in distributed artificial intelligence*, Artif. Intell. Rev. **6** (1992), 35–66.

[3]   P. Davidsson, E. Astor, and B. Ekdahl, *A framework for autonomous agents based on the concept of anticipatory systems*, In Cybernetics and Systems '94 (R. Trapl, ed.), vol. II, World Scientific Publishing, New Jersey, 1994, pp. 1427–1434.

[4]   H. El-Rewini and T. G. Lewis, *Distributed and Parallel Computing*, Manning Publishing, Cambridge, 1997.

[5]   P. Haddawy and L. Rendell, *Planning and decision theory*, Knowl. Eng. Rev. **5** (1990), no. 1, 15–33.

[6]   L. Hurwicz, *The design of mechanisms for resource allocation*, Am. Econ. Rev. (AER) **63** (1973), no. 2, 1–30.

[7]   N. Jennings and M. Wooldridge, *Applying agent technology*, Int. Journal of Applied Artificial Intelligence **9** (1995), no. 4, 357–369.

[8]   G. Knotts, M. Dror, and B. Hartman, *A project scheduling methodology derived as an analogue of digital circuit technology*, Ann. Oper. Res. **82** (1998), 9–27.

[9]   J. Kurose and R. Simha, *A microeconomic approach to optimal resource allocation in distributed computer systems*, IEEE Trans. Comput. **38** (1989), no. 5, 705–717.

[10]   P. Maes, *Modeling adaptive autonomous agents*, Artificial Life: An Overview, Langton, 1995, pp. 135–162.

[11]   J. Pasquale, *Randomized coordination in an autonomous decentralized system*, Proceedings of ISADS '93, Japan, 1993, pp. 77–82.

[12]   A. Polydoros and J. Silvester, *Slotted random access spread spectrum networks: An analytical framework*, IEEE J. Sel. Areas Comm. (J-SAC) **SAC-5** (1987), 989–1002.

[13]   J. Rozenblit, *Design for autonomy: An overview*, Int. Journal of Applied Artificial Intelligence **6** (1992), no. 1, 1–18.

Moshe Dror: Department of Management Information Systems, University of Arizona, Tucson, AZ 85721, USA
E-mail address: mdror@bpa.arizona.edu

Bruce Hartman: Department of Mathematics and Computer Science, Saint Mary's College of California, Moraga, CA 94575, USA
E-mail address: bhartman@stmarys-ca.edu

Gary Knotts: Department of Management Information Systems, University of Arizona, Tucson, AZ 85721, USA
E-mail address: gknotts@bpa.arizona.edu

Daniel Zeng: Department of Management Information Systems, University of Arizona, Tucson, AZ 85721, USA
E-mail address: zeng@bpa.arizona.edu

Special Issue on
Time-Dependent Billiards

Call for Papers

This subject has been extensively studied in the past years for one-, two-, and three-dimensional space. Additionally, such dynamical systems can exhibit a very important and still unexplained phenomenon, called as the Fermi acceleration phenomenon. Basically, the phenomenon of Fermi acceleration (FA) is a process in which a classical particle can acquire unbounded energy from collisions with a heavy moving wall. This phenomenon was originally proposed by Enrico Fermi in 1949 as a possible explanation of the origin of the large energies of the cosmic particles. His original model was then modified and considered under different approaches and using many versions. Moreover, applications of FA have been of a large broad interest in many different fields of science including plasma physics, astrophysics, atomic physics, optics, and time-dependent billiard problems and they are useful for controlling chaos in Engineering and dynamical systems exhibiting chaos (both conservative and dissipative chaos).

We intend to publish in this special issue papers reporting research on time-dependent billiards. The topic includes both conservative and dissipative dynamics. Papers discussing dynamical properties, statistical and mathematical results, stability investigation of the phase space structure, the phenomenon of Fermi acceleration, conditions for having suppression of Fermi acceleration, and computational and numerical methods for exploring these structures and applications are welcome.

To be acceptable for publication in the special issue of Mathematical Problems in Engineering, papers must make significant, original, and correct contributions to one or more of the topics above mentioned. Mathematical papers regarding the topics above are also welcome.

Authors should follow the Mathematical Problems in Engineering manuscript format described at http://www.hindawi.com/journals/mpe/. Prospective authors should submit an electronic copy of their complete manuscript through the journal Manuscript Tracking System at http://mts.hindawi.com/ according to the following timetable:

Manuscript Due	December 1, 2008
First Round of Reviews	March 1, 2009
Publication Date	June 1, 2009

Guest Editors

Edson Denis Leonel, Departamento de Estatística, Matemática Aplicada e Computação, Instituto de Geociências e Ciências Exatas, Universidade Estadual Paulista, Avenida 24A, 1515 Bela Vista, 13506-700 Rio Claro, SP, Brazil ; edleonel@rc.unesp.br

Alexander Loskutov, Physics Faculty, Moscow State University, Vorob'evy Gory, Moscow 119992, Russia; loskutov@chaos.phys.msu.ru