# The Eager Bidder Problem: A Fundamental Problem of DAI and Selected Solutions

Michael Schillo
German Research Center for Artificial Intelligence (DFKI)
Stuhlsatzenhausweg 3
66123 Saarbrücken, Germany
schillo@dfki.de

Christian Kray
German Research Center for Artificial Intelligence (DFKI)
Stuhlsatzenhausweg 3
66123 Saarbrücken, Germany
kray@dfki.de

Klaus Fischer
German Research Center for Artificial Intelligence (DFKI)
Stuhlsatzenhausweg 3
66123 Saarbrücken, Germany
kuf@dfki.de

## ABSTRACT

The *contract net protocol* is a widely used protocol in DAI, as it proved to be a flexible and low communication interaction protocol for task assignment. It is however not clear in which manner agents participating in a contract net should allocate their resources if a large number of contract net protocols is performed concurrently. If the agent allocates too many resources too early, e.g. when making a bid, it may not get any bid accepted and resources have been allocated while other negotiations have come to an end and it is no longer able to make bids for them. If it allocates resources too late, e.g. after being awarded the contract, it may have made bids for more tasks than its resources allow for, possibly all being accepted and resulting in commitments that cannot be kept. We call this dilemma the *Eager Bidder Problem*. Apart from resource allocation this problem is of further importance as it constitutes the "dual" problem to engaging in multiple simultaneous first-price sealed-bid auctions.

We present an ad hoc solution and two more complex strategies for solving this problem. Furthermore, we introduce a new method based on a statistical approach. We describe these mechanisms and how they deal with the concept of commitment at different levels. There is no optimal solution for every problem setting, but each has advantages and disadvantages. Our discussion concludes with criteria for the decision which of these mechanisms is best selected for a given problem domain.

## 1. INTRODUCTION

The assignment of tasks to agents and the (re-)allocation of tasks in a multiagent system (MAS) is one of the key features of automated negotiation systems [20]. The contract net protocol, originally proposed in [18], and other more general auction mechanisms can be widely applied to resource and task allocation problems. The contract net has been applied e.g. to online dispatching in the transportation domain [1, 5], meeting scheduling [6, 16] and flexible manufacturing [17, 11, 9]. Although researchers tried to deal with probabilistically known future events, little work has been done on specifying strategies that help an agent to make reasonable decisions when several contract nets are concurrently active. With recent developments of small transaction commerce on the Internet for purchasing goods and information, this problem will become relevant and the trend to virtual enterprises and agile manufacturing (cf. [11]) will make this even more demanding. In these coordination processes, the notion of commitment (and its semantics) is central [8]. In these settings the agents will face the situation that they have to decide in how many of the concurrent negotiations they intend to participate and how they should handle their commitments with respect to the local resources at hand. Travel agencies doing flight reservations are a practical example of multiple contracting negotiations that are going on in parallel. The overbooking of seats in airplanes is an application where the companies in addition to mathematical tools use experience to calibrate coefficients for the risk estimation.

For the application of MAS technology this means that if there is only one source for tasks, i.e. there is only one manager who announces tasks, then the bidders do not have much of a choice. They will try to do their best by participating in the CNPs that are initiated by this source. The usual strategic bidding [12] will then depend on whether the pure CNP is used or alternative, truth-revealing mechanisms such as for example the Vickrey auction [19] (which in turn is known to be only truthrevealing depending on certain assumptions). The problem changes dramatically if we assume that there are several sources of tasks and that the agents have only limited resources to actually execute tasks. In this case the agent has to decide how many of the CNPs that are active it is actually going to participate and which kind of offer it wants to send to the managers of those CNPs. The following sections present a systematic discussion on mechanisms that can be used to solve the problem and how they deal with the concept of commitment at different levels. There is no optimal solution for every problem setting, but the protocols have advantages and disadvantages. Our discussion concludes with criteria for the decision on which of these mechanisms is best selected for a given problem domain. Although our discussion concentrates on the CNP, it is obvious that the general results can be transferred to other single-shot auctions.


*AAMAS'02,* July 15-19, 2002, Bologna, Italy.

# 2. THE EAGER BIDDER PROBLEM

## 2.1 The contract net protocol

Assume we have an agent that has a task that needs to be done but it does not have the ability or the resources to do so. The contract net protocol [18] was designed to describe the communication protocol to determine some other agent to do the task. Our discussion is based on the FIPA interpretation of the contract net [3], which is a minor modification of the original protocol in that it adds rejection and confirmation speech acts. Currently, this interpretation is the standard for a whole range of prominent agent platform implementations [4, 7, 21].

In order to comply with the FIPA standards, we call the agent with the task *initiator* (*manager* in the original), agents that compete for acquiring the task *participants* (*bidder*, respectively). In general, the procedure requires the initiator to send a call for proposals including a task description to all participants. They can specify their required costs for this task in a proposal or refuse to do the task at all. The initiator then accepts one of these proposals, and rejects all others. The agent who got his bid accepted is then required to inform the participant about the result of the task (or its failure). Note that this protocol requires the initiator to know when it has received all replies. In the case that a contractor fails to reply with either a propose or a refuse act, the initiator may potentially be left waiting indefinitely. To guard against this, the call for proposals includes a deadline by which replies should be received by the initiator. Proposals received after the deadline are automatically rejected with the given reason that the proposal was late [3].

## 2.2 Problem definition

The contract net protocol was designed for distributing one task among a number of agents. As long as the participants are not engaged in any other activity, this mechanism will find the agent, which the initiator prefers most, and will create a commitment of the accepted agent to perform the task.

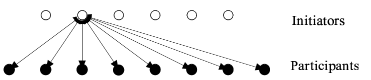


**Figure 1: CNP communication.**

However, if we assume a large number of initiators and bounded resources for each of the participants (as is generally the case in applied multiagent systems), new problems arise. While several initiators are requesting bids (communication paths from all white circles to all black circles in Figure 1), it is a hard problem for each agent to decide when to allocate the resources for which task. Imagine that among the agents in a large-size multiagent system there are $m$ agents with tasks (initiators) and $n$ providers of services (participants). While a participant is waiting for being awarded to do tasks by a possibly large number of initiators, it may still receive more calls for proposals. The participant may at this stage not have received any *reject-proposal* messages as the initiators are either idle waiting for incoming bids until the end of the deadline (expiration time) or still busy evaluating the proposals. This is a problem, in that the bid it would send back to any of the initiators depends on the availability of its resources, in some domains (e.g. the transportation domain) the cost for a task even depends on the set of tasks already scheduled. Waiting for an *accept-proposal* or reject-proposal message before making any further bids will result in deadlocks, or timing out of the protocol on the side of the initiator, thus terminating the protocol with fewer proposals than possible. In short, the problem consists in:

> deciding on which resources to allocate in a distributed setting with more than one initiator in order to create bids to all incoming call-for-proposals before the first reject or accept notification is returned.

There exist several alternatives, none of them are satisfying. If the agent allocates too many resources too early (e.g. when making a bid), it may not get its bid accepted and therefore allocate the resources, which then are not available for other incoming calls for proposals. If it allocates too late (e.g. when receiving the *accept-proposal* message), it may have committed to more tasks than it has resources, thus causing repeatedly propagating failure to the system level.

## 2.3 The Ad Hoc Solution

Let us consider the case where the agent allocates resources at the time of sending the bid. We call this solution the ad hoc solution. This solution is conservative in that it makes sure that only correct assignments of tasks to agents are created, i.e. that every agent only commits to the tasks it can perform. This is also used as a conservative solution in open cry auctions, where agents make at most concurrent $k$ bids, if they want to purchase $k$ goods (cf. e.g. [10]). However, if we want to reduce communication and use a single shot auction or, as in our case, a contract net protocol, then agents are only allowed to make a single bid per protocol instance, and it is not unlikely that several participants send their few proposals to a small set of initiators. The result is that only some of them get a task assigned, while others remain idle. Therefore, the ad hoc procedure is not complete in that it will not compute solutions that could be found with better approaches.

For illustration, consider using the conservative approach in a setting with 100 initiators, each having one task to assign and 100 participants, each capable of performing one task. Further consider that the deadlines are set such that the participants cannot reply to the calls sequentially (otherwise there would be no problem). If in this case every participant just sends one bid, the chance of getting a bid accepted assuming lottery on the side of the initiator is ca. 0.64[1]. If other agents make more than one bid, the probability is even lower. So in more than one third of all cases, the available resources of the participant will be idle due to the conservative strategy. Correspondingly, the same number of initiators will be left with unassigned tasks, as they did not get any bids for their tasks.

# 3. CLASSIFICATION OF SOLUTIONS

Beyond the safe but highly inefficient ad hoc solution, we will now present a classification of three different approaches to the Eager Bidder Problem: (i) the leveled commitment approach, (ii) the protocol redesign approach and

[1]The computation of this probability is out of scope here, but from the problem chosen, it is in any case clear that the probability is below 1.

(iii) the statistical approach. What distinguishes the three approaches is the way they treat commitment: the leveled commitment approach spends resources on negotiation of penalties for breaking a commitment. This solves the problem by introducing a second level (or meta) commitment about the penalty of breaking a first level commitment. The protocol design method approaches the problem by delaying the commitment time further in the future to achieve high efficiency. The statistical method assumes the commitment is important enough to minimize the risk of breaking it, but it is not essential to the overall task of the system to guarantee full completion of all tasks.

# 4. APPROACH 1: LEVELED COMMITMENTS

Sandholm and Lesser proposed a leveled commitment contracting protocol to give self-interested agents in the context of automated negotiation the possibility to retract commitments when they face a situation where the future evolves in an uncertain manner [13, 14]. They show that this leveled commitment protocol increases Pareto efficiency of deals and that it can make contracts individually rational when no full commitment contract can.

This protocol allows an agent to participate in several contract net protocols in a sequential manner. The agent is able to decommit from commitments to earlier contracts when it finds out that a new contract is more attractive with respect to the local payoff for the agent. In doing so, the decommitment penalty has of course to be taken into account. The leveled commitment approach has been extended by Excelente-Toledo et al. in that (among other enhancements) the ongoing cost of participating in the coordination process is incorporated in the decommitment penalty [2]. However, Sandholm and Lesser's discussion does not include the concurrent participation in several contract nets. If the agent participates in too many contract net protocols at the same time, they risk to pay too many penalty fees which would not be an individually rational strategy. We come back to this problem in Section 6.

# 5. APPROACH 2: PROTOCOL REDESIGN

The second approach to solve the Eager Bidder Problem is based on a redesign the protocol to postpone the time of commitment as far as possible. The major inefficiency in the CNP is that in every execution of the protocol *all* participating agents need to commit themselves to do the job, although only *one* of them will actually be awarded to do the task. We now present the *contract net with confirmation protocol* (CNCP) taken from [15], which precisely addresses this issue and improves the CNP procedure by drastically reducing the number of commitments made.

## 5.1 Procedure

The CNCP (cf. Figure 2) is very similar to the CNP. It starts with a *call for proposals*, gathers the responses from the participants, until the initiator received messages from all participants or the deadline has passed. As in the contract net protocol, this deadline safeguards that singular message dropouts do not prevent the whole protocol from termination. In the original contract net, the participant makes its commitment in the bidding stage. In the CNCP this is not the case: the commitment is only made when the initiator requests that the participant should take over the task. For this purpose the initiator arranges all bids in a sorted list and sends *request*s to all participants starting with the best bid to find out if they can actually do the job. This is now easy to decide for the participant, as it knows it will get the task awarded if it agrees to do it and there is no harm to make a commitment at this stage. The next participant is sent a request message if the last participant has sent a *refuse* or a deadline has passed. This iteration stops as soon as one participant replies with an *agree* message. All other agents are sent a *reject-proposal* message (those who have already received the request and sent the refuse do not need this message, but this depends on the agent implementation and it does not interfere with the basic protocol properties). The participant only needs to commit at the time of sending the *agree* message. In order to trigger task execution and to correspond to the CNP, it is required that the agent sends an *accept-proposal* while the participant will reply as it does in the CNP with *failure*, *inform-done*, or *inform-ref*.

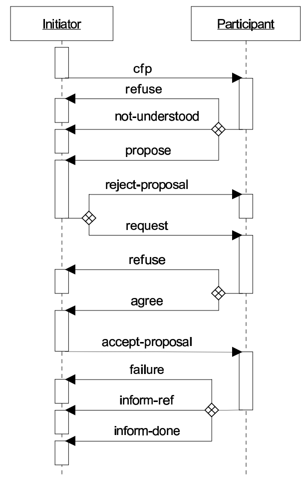


**Figure 2: The Contract Net with Confirmation Protocol.**

## 5.2 Discussion and Analysis

The proposed procedure needs $O(n)$ messages, where $n$ is the number of participants. In the best case, the CNCP requires only two more messages (the request for confirmation and the reply to it) while still solving the resource allocation problem of the participant. In the worst case, the initiator needs to contact all participants to find out that no one can do the task. Although this results in an excess of $2n$ messages for the CNCP, its great advantage is that it only requires one single agent to make one commitment. This is achieved by using the confirmation stage in the protocol, to postpone the commitment and allow the participants to reply to all incoming *call for proposals* without need to already allocate the resources at this early stage of interaction or to risk penalties for multiply allocating resources. A minor disadvantage of this approach is that the initiator possibly needs some overhead to sort the list of participants according to their bids, while the CNP only requires it to find the maximum. However, with careful implementation this additional computational effort for finding the next best participant is only required at the point where the original CNP already would have failed.

In order to guarantee termination even in the case of faulty participants the second deadline of the protocol is necessary. It makes sure that the next best participant can be sent a request message and has a chance to receive the task.

# 6. APPROACH 3: BIDDING STRATEGIES BASED ON RISK ANALYSIS

The general idea of this new approach is to risk some broken commitments, as long as the probability of this event can be evaluated and can in the long run be guaranteed to be below a certain threshold. This is similar to the statistical approaches already mentioned in flight booking systems, which involve the risk of overbooking (as is the common experience with frequent flyers) but in general work quite well and have high acceptance in settings where efficient usage of resources is important. In contrast to these systems, we do not assume databases containing long-term, past experiences (as they are available in flight booking), but make some basic assumptions about the distribution of agent behavior.

## 6.1 Risk estimation

As already mentioned, the aim of this approach is to determine for a given risk threshold the number of bids an agent $X$ can make beyond the amount of resources an agent has at its disposal. One assumption necessary to cope with the complexity of the problem is to assume that all agents apart from $X$ will use the same strategy, i.e. will make identical choices, albeit different from the choice of $X$ (later in the discussion section we will show how this assumption can be relaxed). We do not assume these agents make the same choice as $X$.

Let $m$ be the number of initiators, $n$ the number of participants, $N_X$ the number of bids an agent $X$ is making ($N_X < m$), and let $N_A$ be the number of bids of all other agents ($N_A < m$).

The first goal we want to achieve is to determine the probability for agent $X$ of getting *one* bid accepted when sending it to a randomly chosen initiator. Once we achieve to compute this probability, we will also be able to evaluate the risk of getting more bids accepted than we can perform tasks (for the extension of this approach if the capacity is greater than one see the discussion section below).

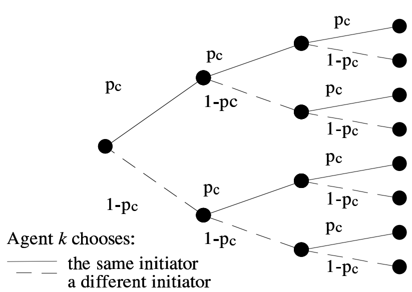


**Figure 3: All permutations of other agents choosing the same or a different initiator in an example probability tree.**

Let us first look at the choice any other agent than agent $X$ is going to make. The number of bids they are going to make is assumed to be $N_A$. From a probabilistic point of view, deciding whom of the $m$ initiators to send a bid is therefore equal to choosing a tuple of the length $N_A$ with $m$ different possibilities to choose each member of the tuple. If we want to know the probability $p_c$ of an agent to choose a specific initiator for making a bid, we know that one entry of this tuple is going to be this initiator (with $N_A$ possibilities for the position of this entry in the tuple). The other $N_A - 1$ entries will contain the permutations of the other $m-1$ initiators, thus ensuring that the entries must be mutually different. To get the probability for this event, we divide the number of these tuples by the number of all possible $(m)_{N_A}$ tuples:

$$\begin{aligned} p_c &= N_A(m-1)_{N_A-1}\frac{1}{(m)_{N_A}} \\ &= N_A\frac{(m-1)!(m-N_A)!}{(m-1-N_A+1)!m!} \end{aligned}$$

Evaluating this expression yields:

$$p_c = \frac{1}{m}N_A \tag{1}$$

With this tool at hand we will now compute the probability that agent $X$ gets its bid to one specific initiator accepted. Firstly, it is certain that it gets the award, if no one else sent a bid to this initiator. The probability for this case is the product of multiplying the probability that one agent apart from $X$ did not choose this initiator, which is $1 - p_c$, $n-1$ times multiplied with itself:

$$(1-p_c)^{n-1} \tag{2}$$

This corresponds to the lowest branch in 3. Secondly, we know that the probability to be chosen by the initiator is one over the number of agents, which made a bid, as we here assume a lottery (for strategic bidding see below). We can also compute the probabilities for all of the cases "one agent made a bid to this initiator", "two agents made a bid", "three agents made a bid" etc.

These probabilities are the product of the probability for choosing the initiator to the power of the number of cases, multiplied with the probability for the counter event to the power of the number of these cases, multiplied with the number of all possible permutations. This means we get as formula a single sum evaluating the probability $p_a$ for agent $X$ of getting a single bid accepted, taking into account that with certain probabilities either zero, one, two, or up to $n-1$ (i.e. all but $X$) agents may have made a bid to the same initiator:

$$p_a = \sum_{i=0}^{n-1}\binom{n-1}{i}p_c^i(1-p_c)^{n-1-i}\frac{1}{i+1}. \tag{3}$$

The case for zero other participants is already integrated here ($i = 0$) as the probability for the case with no other bids (cf. Formula 2) to this initiator can be appropriately integrated into the formula. Once we have the probability of getting a bid accepted by one initiator, we can estimate the probability (or the risk) to get more than one bid accepted if we decide to make bids to more than one initiator. Making a number of bids is like a chain of experiments, each with the same probability of being successful (in the sense that the bid is accepted). Therefore we can compute the probability of having more than one bid accepted as a *Bernoulli-chain*:

$$P(T \geq 2) = \sum_{i=2}^{N_X}\binom{N_X}{i}p_a^i(1-p_a)^{N_X-i}. \tag{4}$$

Now agent $X$ can, given a risk threshold $\tau$, compute the greatest $N_X$ that will still imply a risk $R_X(N_X, N_A) = P(T \geq 2)$ smaller than $\tau$.
In order to give provide intuition for this result, we will now discuss an example configuration and show the range of this risk for agent $X$ depending on the number of bids made by others and itself.

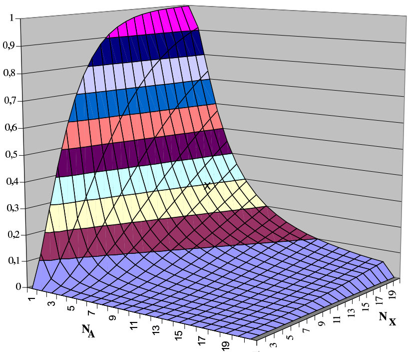


**Figure 4: The distribution of risk $R_X(N_X, N_A)$, depending on the number $N_X$ of bids made by agent $X$ and the number of bids $N_A$ made by each other agent.**

## 6.2 Example

Assume we have thirty initiators ($m = 30$) and eighty participants ($n = 80$), all initiators with one task, all participants with the resources for a single task (for more than one task, see the discussion section below). If we compute $P(T \geq 2)$ for all $N_X$ and $N_A$ with this configuration, we get the risk distribution $R_X(N_X, N_A)$ shown in Figure 4. The distribution displays two behaviors we can also predict by analysis: if $N_A$ is fixed, the risk is increasing monotonously with increasing $N_X$, and decreasing monotonously with increasing $N_A$, if $N_X$ is fixed. This reflects the fact that agent $X$ is risking more when making more and more bids and that the risk for agent $X$ is decreasing if the other agents make more bids and thus decrease the probability that $X$ is chosen.
This distribution has an impact on the practical use of a statistical risk taking approach in the contract net procedure. Let us assume rational behavior of all agents, the same knowledge about the number of participants, and the same risk threshold $\tau$. In this case, choosing the number of bids to make can be described as a game in normal form and we are able to show important properties for this game.

## 6.3 Game Theoretic Analysis

The given interaction problem can be transformed into a game in normal form. The set of players $S$ of the game is $S = \{i | i \text{ is a participant}\}$ ($X$ being one of them). The set of options of the agents is the number of bids they are making. This is the set of integers up to the number of initiators (agents are not allowed by the protocol to make more than one bid to an initiator): $O = \{k | k \leq m\}$. The pay-off matrix $M$ is constructed by inserting the risk computed by Formula 4. Setting up the matrix including all players would result in an $n$-dimensional matrix, making the analysis difficult beyond the scope of this work. Thus, we use the simplification we made earlier: we assume that all players apart from $X$ use the same strategy $N_A$ (we show in the discussion session that this assumption does not interfere with the general argument).
The risk distribution is symmetric in the sense that it does not depend on which agent we choose for our analysis and the entries in the pay-off matrix are strictly monotonous. Figure 5 shows a diagonal cut through the distribution of Figure 4, displaying the risk for any agent, given that all other agents choose to make the same number of bids. It is rational for each agent to make as many bids as possible in order to increase the expected utility. The upper bound for the number of bids is the $N_X$ where the resulting risk exceeds $\tau$. In our example $N_X$ would be six, if we assume a threshold of five percent risk (analogously for different $\tau$). Is there a reason for a single agent to deviate from this solution? Increasing $N_X$ is not rational, as for a fixed $N_A$ risk is increasing with increasing $N_X$. On the other hand decreasing $N_X$ is certainly not rational, as making less bids decreases the expected utility. Thus, this choice is a pure strategy *Nash equilibrium* of the game (see our proof in the appendix for a more precise treatment). Furthermore, due to the symmetry of the risk distribution and its monotonic behavior, this is also a Pareto optimal solution. There is no other combination of choices that will get any agent more pay-off without reducing the pay-off of some other agent, while preserving the risk threshold. For the game theoretic discussion of the interaction it is very important to have a single pure strategy Nash equilibrium that is also Pareto optimal, as it stabilizes the agent behavior.

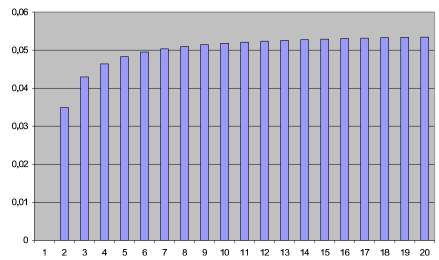


**Figure 5: The risk for each agent $X$, if all agents make the same number $N_X$ of bids.**

## 6.4 Discussion

Of course this approach is acceptable only in domains which allow for a small number of unallocated tasks, as dropouts will occur with the specified probability. This approach is applicable, if the number of tasks is high, the value of single tasks is low or the tasks contain redundancy.

This mechanism is not restricted to scenarios where initiators have only one task, or participants have only resources of capacity one. More than one task per initiator is in terms of the risk computation equal to the case of an additional amount of initiators as different *call for proposals* are treated independently and the identity of the initiator does not play a specific role. The increase of resources on the side of the participants results in a change in the estimation of the "bad" outcomes of the *Bernoulli-Experiments*. The "bad" outcome now is if strictly more than the available capacity $c$ bids get accepted which yields for the risk of the individual agent $X$:

$$P(T \geq c + 1) = \sum_{i=c+1}^{N_X} \binom{N_X}{i} p_a^i (1 - p_a)^{N_X - i}.$$

Another assumption we can relax is the assumption that all agents apart from $X$ are choosing the same number of bids to make. Note that for the design of a single agent this assumption is still worth considering, because as long as $X$ has not collected enough data about the behavior of other agents to discriminate between them, it is well worth considering that they act in an equal manner. To show that this mechanism does not sink or swim with this assumption, we present its extension.

The assumption is important for Formula 1, where we evaluate the probability $p_c$ of an initiator being chosen by a participant. This probability relies on $N_A$. If we cannot assume identical choices for all agents apart from $X$, we need to introduce $N_{A,i}$ denoting the choice of agent $i$. This results in different probabilities $p_{c,i}$ for an initiator to be chosen by a certain participant. This change does not affect our risk distribution directly, but through the probability $p_a$ for a participant to be accepted by the initiator. This probability is now more difficult to write up, as we need to take into account permutations (in the case with our assumption all probabilities were the same and could be summarized more easily). If we look at Figure 3, the change in the argument lies in the now individually different weights of the branches in the probability tree. As before in Formula 3 the sum now consists of the sum of all probabilities for the case of $0..n-1$ participants choosing this initiator weighed by $\frac{1}{i+1}$ to care for the probability of being chosen in this case. This sum itself resists brief notation, but it is a sequence of multiplications and sums and is easy to compute. The result of this computation is a constant $p_a$, which can be used with Formula 4 to compute the risk distribution as already discussed in the previous section.

## 7. CONCLUSIONS

Whenever a multiagent approach with multiple concurrent CNPs is used for resource allocation, as for example in multiagent job shop scheduling problems, the system designer faces a variation of the *Eager Bidder Problem*. We presented an ad hoc solution as well as three different advanced approaches, each with different properties. None of them is the "silver bullet", but each has its own pros and cons.

The leveled commitment approach introduces the concept of penalties and the option of "legally" decommitting. It has maximum flexibility and includes the option of decommitting during run-time with the drawback of more communication (and implementation) effort. Failures result in penalties paid, restarting the protocol with the assigned but failed task is assumed. The protocol redesign is a specific change of the CNP and uses a drastic reduction of the number of commitments to reach its goal, while in the worst case it only requires $2n$ more messages. The statistical method presented here is easy to compute and the risk estimation in uncertain environments is straightforward. It requires no changes to the protocol. This interaction has a pure strategy Nash equilibrium that can be computed easily and is also Pareto optimal. This is particularly important as it increases the stability of this approach.

The following is a classification of different task assignment settings where we provide information on which of the mentioned approaches from our point of view are satisfying the settings' requirements best.

**The protocol is not under control of the designer / the protocol cannot be changed.** This is the case with most open or semi open settings of multiagent systems. In this case either the conservative strategy or the statistical approach apply (depending on whether dropouts are acceptable or not). The other approaches require changes in the protocol.

**Flexibility during runtime as the possibility of decommitting is necessary.** Only the leveled commitments approach extents the possibility to decommit to the execution time of a task. It explicitly allows for dealing with failing agents and their possibility to decommit to perform a task during runtime.

**A quick initial result is desired, as post optimization is an inherent part of the overall procedure.** Here the statistical approach will help, as during post-optimization the failure cases can be eliminated. It is also possible here to let the agents bid with no commitment at all, but this will return worse solutions than the statistical approach. Note that every protocol stage guarded with a deadline is a potential source of delay and the leveled commitment approach, and to some extent the CNCP are disadvantageous in this respect.

**Low amount of messages is most important.** The conservative approach is best in terms of messages sent, as bids can only be made for the amount of resources available. All other approaches make more bids. If the allocation should be more efficient than with the conservative approach, the statistical approach applies. The statistical approach will produce only a limited number of messages more than the conservative approach, which can be configured by choosing the threshold. The CNCP is less applicable, as it gives incentives for making bids to every incoming cfp, as there is always a chance to be awarded the task, but it reduces the communication overhead to a minimum and provides better results. The leveled commitment approach produces most messages, as there is an incentive to bid everywhere and to communicate about the penalties involved, including the possibility of counterproposals.

**Robust behavior when facing failing agents/dropouts.** Only the leveled commitment approach deals with the case of agents failing during task execution time explicitly. All approaches provide the possibility to restart the procedure in case an accepted participant fails to perform the task. The CNCPs advantage is that it stores the list of possible alternatives without requiring a commit for the participant to be in this list. So a call for proposals does not have to be sent to all participants but only those remaining in the list. Deadlines are part of all the procedures, as well as the FIPA version of the CNP. Thus, in case a single agent fails to reply, there is no danger of a deadlock. However, as the leveled commitment approach allows for many messages being sent, the increased number of messages requires an equally high number of deadlines, possibly causing long delays when facing dropouts.

As a conclusion, it is up to the designer to choose for a given domain the most appropriate approach and we believe we provided some helpful arguments for this decision.

## 8. FUTURE WORK

The presented work is restricted to the CNP, where there is only a task that needs to be assigned to a single agent. We are interested in repeating our work with the case where the initiator assigns a composed task, where it is not clear at the start of the protocol, which group (you might want to say coalition) of agents is capable of solving this task.

For the analysis in the statistical approach it was necessary for us to assume that no agent makes use of further information for strategic bidding. However, in many cases, the height of the bid is not arbitrary as some assumptions about costs and the bids of others may be made. Therefore, in our future work we want to take into account the strategic choice of the bid and reevaluate the probabilities for getting a bid accepted given some distribution over an interval of possible bids. Currently we are working on extending this approach to include a sanctioning mechanism, i.e. we introduce a coefficient $\rho$ that expresses which part of the pay-off the agent looses if it committed to a task it cannot perform in the end. Here, $\rho$ can be smaller or greater one. We will then repeat the analysis presented here with the expected total pay-off including the expected sanction, without considering the risk threshold. We believe that this describes more accurately the agents' and the system's behavior in the long run.

## 9. ACKNOWLEDGEMENTS

We owe thanks to Michael Rovatsos for many fruitful discussions on early versions of this work and to the anonymous reviewers for important and constructive comments.

## 10. REFERENCES

[1] H. Bürkert, K. Fischer, and G. Vierke. Holonic transport scheduling with TELETRUCK. *Applied Artificial Intelligence*, 14(7):697–726, 2000.

[2] C. Excelente-Toledo, C. Bourne, and N. Jennings. Reasoning about commitments and penalties for coordination between autonomous agents. In *Proceedings of the Fifth International Conference on Autonomus Agents*, pages 131–138. ACM, 2001.

[3] FIPA. (Foundation for Intelligent Agents), http://www.fipa.org/repository/ips.html.

[4] FIPA-OS. http://fipa-os.sourceforge.net/.

[5] K. Fischer, J. P. Müller, and M. Pischel. A model for cooperative transportation scheduling. In *Proceedings of the 1st International Conference on Multiagent Systems (ICMAS'95)*, pages 109–116, San Francisco, June 1995.

[6] L. Garrido and K. Sycara. Multiagent meeting scheduling: Preliminary experimental results. In *Proceedings of the 2nd International Conference on Multiagent Systems (ICMAS'96)*, 1996.

[7] JADE. http://www.sharon.cselt.it/projects/jade/.

[8] N. R. Jennings. Commitments and conventions: The foundation of coordination in multi-agent systems. *The Knowledge Engineering Review*, 8(3):223–250, 1993.

[9] V. Parunak. Manufacturing experience with the contract net. In M. Huhns, editor, *Distributed Artificial Intelligence*, pages 285–310. Pitman, London, 1995.

[10] C. Preist and A. Byde and C. Bartolini. Economic Dynamics of Agents in Multiple Auctions. In Proceedings of the Fifth International Conference on Autonomous Agents (Agents'01), pages 545–551, 2001.

[11] R. Rabelo, L. Camarinha-Matos, and H. Afsarmanesh. Multi-agent perspectives to agile scheduling. In R. Rabelo, L. Camarinha-Matos, and H. Afsarmanesh, editors, *Intelligent Systems for Manufacturing*, pages 51–66. Kluwer Academic Publishers, 1998.

[12] J. Rosenschein and G. Zlotkin. *Rules of Encounter*. The MIT Press, Cambridge, Mass., 1994.

[13] T. Sandholm. *Negotation among Self-Interested Computationally Limited Agents*. PhD thesis, University of Massachusetts at Amherst, Department of Computer Science, 1996.

[14] T. Sandholm and V. Lesser. Advantages of a leveled commitment contracting protocol. In *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI-96)*, pages 126–133, 1996.

[15] M. Schillo, K. Fischer, and T. Knabe. The contract net with confirmation protocol: A solution to a fundamental problem of DAI. Technical memo TM-01-01, DFKI, 2001.

[16] S. Sen and E. Durfee. On the design of an adaptive meeting scheduler. In *Proceedings of the 10th IEEE Conference on AI Applications*, 1994.

[17] W. Shen and D. Norrie. An agent-based approach for dynamic manufacturing scheduling. In *Proceedings of the 3rd International Conference on the Practical Applications of Agents and Multiagent Systems*, 1998.

[18] R. G. Smith. The contract net protocol: High level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, Series C-29(12):1104–1113, 1980.

[19] W. Vickrey. Counter speculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16(1):8–37, 1961.

[20] G. Weiß, editor. *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*. The MIT Press, Cambridge, MA, 1999.

[21] ZEUS. http://www.btexact.com/projects/agents/.

## APPENDIX

The following will prove the existence of one pure strategy nash equilibrium that is also Pareto optimal for the statistical approach as described in Section 6. The proof consists of several parts. After some definitions we will first show that there is one pure strategy Nash equilibrium in the game and that it is the only pure strategy Nash equilibrium. We then show that this Nash equilibrium is also Pareto optimal. This proof also applies to the $n$-player game as described in Section 6.4.

**STEP 1:**(Preliminary assumption) We assume here $n = 2$, i.e. we will first look at a two player game between agents $X$ and $Y$ (See Step 8 for the relaxation of this assumption). Think of Figure 6 as the birds-eye-view of Figure 4. The arrows indicate that the risk $R_X(N_X, N_Y)$ is decreasing for increasing $N_X$ and for decreasing $N_Y$. This means that the risk for agent $X$ is increasing with making more bids and decreases with other agents making more and more bids.

**STEP 2:** (Location of the Nash equilibrium, formulation of the theorem) Only those cells of the matrix are *acceptable* for both players where the following conditions hold:

$$R_X(N_X^*, N_Y) < \tau \text{ and } R_Y(N_Y^*, N_X) < \tau.$$

It is each agents' aim to maximize the chance of getting a bid accepted and therefore to make as many bids as possible, while restricting the risk of getting more bids accepted than capacity available to a threshold $\tau$.

**Theorem:** Let $c(i, i)$ be the cell on the diagonal, with greatest $i$ limiting the risk $R_X(N_X, N_Y)$ to $\tau$ for both agents. Then $c$ is the single pure strategy Nash equilibrium of the game and $c$ is also Pareto optimal.

**STEP 3:** ($c$ is a pure strategy Nash equilibrium)

**Lemma:** Cell $c$ is the choice in the game with Nash equilibrium. It holds for both agents that, with fixed opponent choice, every cell in the grid either a) yields a lower chance to get bids accepted than cell $c$ or b) has a risk above the risk threshold $\tau$.

**Proof:** Without loss of generality we choose to prove the lemma for agent $X$. Due to the symmetry of the risk distribution, the same holds for agent $Y$. The proof is based on refuting the contradiction and a discrimination into two cases. For both we will assume that there is a $c'$ that contradicts the theorem and then refute this assumption. As stated in the theorem, $N_Y$ is fixed (for agent $Y$ with fixed $N_X$).

**Case 1:** Let $c'$ be a cell with $N_X$ (number of bids) smaller than $i$. It follows that the chance to get a bid accepted is smaller. Therefore $c'$ is not a better choice than $c$, hence supporting the theorem.

**Case 2:** Let $c'$ be a cell with number of bids greater than $i$ (light gray cells marked $2b$ in Figure 6, $2a$ for the proof with agent $Y$, respectively). The higher number of bids increases the chance of getting bids accepted. To render $c'$ a better choice than $c$, it is necessary to fulfil part b) as well: maintaining the risk threshold. If any cell with higher bids for agent $X$ (any of the light gray cells) is below the threshold, then, according to the monotony on the $N_X$-axis, the neighbor $c''$ of $c$ must be below the threshold as well.

When choosing $c$, we found that $c'''$ was a cell above the threshold: remember that $c$ was constructed to be the cell on the diagonal yielding maximum bids accepted, while respecting the threshold. According to the monotony on the $N_Y$-axis, if $c'''$ is above the threshold and not acceptable, then $c''$ is above the threshold and not acceptable as well. Therefore $c''$ and $c'$ cannot be better results to the game for $X$, again supporting the theorem.

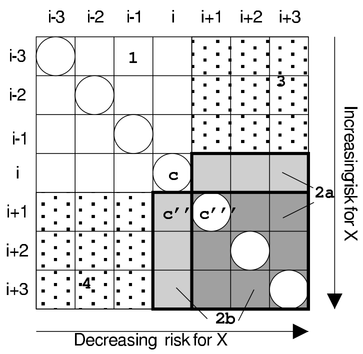


**Figure 6: Overview on the risk distribution matrix.**

For both cases the negation of the theorem was refuted. We will now show in steps 4 to 6 that no other cell is acceptable *and* a pure strategy Nash equilibrium. For this part of the proof, we divide the matrix into different parts and prove for each that there is no other Nash equilibrium.

**STEP 4:** (Quadrant 1) In quadrant 1 there cannot be another Nash equilibrium as the cell $c(i, i)$ according to the monotony of the risk distribution fully dominates every cell in this quadrant.

**STEP 5:**(Quadrants 2a/2b) In Figure 6 the areas $2a$ and $2b$ are shown in light gray, their overlapping area is in dark gray. From Case 2 of the proof of the theorem of Step 3 it follows that the light gray areas are not acceptable: the condition of Step 1 does not hold for them. As in this proof, the other cells of the diagonal are not acceptable, causing that the neighbors to their right and below the diagonal are also not acceptable.

**STEP 6:** (Quadrants 3 and 4) For our proof, these two areas are the two most problematic ones as they contain some cells for which the condition holds. These cells are not easy to identify, as this depends not only on the basic properties of the risk distribution, but on the threshold $\tau$ and the actual value of the risk distribution. However, we can show that each of the remaining two areas is not acceptable for one of the two players (though they are acceptable for the other). Let us first look at quadrant 3.

For agent $X$ any of these cells are not a pure strategy Nash equilibrium, as they are dominated by the cells to the right of $c$: $c^*(i..m, i)$. For $X$ these cells are acceptable as $c$ is acceptable for $X$ and the risk is decreasing for $X$ from $c$ to its neighbors on the right. Thus, $X$ has an incentive to deviate from any chosen cell in quadrant 3. The same holds for quadrant 4 in analogy.

From steps 4 to 6 it follows that there exists no other pure strategy Nash equilibrium apart from the one shown in step 3.

**STEP 7:** ($c$ is Pareto optimal) To show this, we need to prove that there exists no other cell $c^*$ that would give any player higher pay-off without causing one player to loose pay-off. We show that $c^*$ is not in any quadrant of the matrix.

**Quadrant 1:** Due to the monotonic behavior of the risk distribution for player $X$, the cells in column $k$ are dominated by the cells $(k, i)$. These cells are dominated for $X$ by $c$. In analogy, for player $Y$, the cells in row $k$ are dominated by the cells $(i, k)$. These cells are dominated for $Y$ by $c$. From this it follows that no cell in the quadrant is dominating $c$, which would be the prerequisite for $c$ not being Pareto optimal.

**Quadrant 2a/b:** As shown in Step 5, these cells are not acceptable for both players. Thus they cannot contain $c^*$.

**Quadrants 3 and 4:** Any cell in this quadrant can be written as cell $(i + l, i + k)$, with $k$, $l$, being positive integers. For agent $X$ the pay-off is decreasing from cell $(i, i)$ to cell $(i + k, i)$ and decreasing again from cell $(i + k, i)$ to cell $(i + k, i + l)$. Quadrant 4 is the analog case of quadrant 3 for agent $Y$. As for any cell in these quadrants at least for one player pay-off is decreasing, $c^*$ cannot be in quadrant 3 or 4.

None of the quadrants can contain $c^*$, therefore $c$ is Pareto optimal.

**STEP 8:** (For $n$-players) The only assumptions we made here are the fundamental monotony and symmetry properties of the risk distribution. Introducing more players adds new dimensions to the matrix, while the monotony of the risk distribution is preserved: making more bids still implies higher risk for any agent $X$, other agents making more bids implies a decreasing risk for agent $X$. □