# Automated Negotiation with Decommitment for Dynamic Resource Allocation in Cloud Computing

Bo An, Victor Lesser, David Irwin
Dept. of Computer Science
University of Massachusetts, Amherst, USA
{ban,lesser,irwin}@cs.umass.edu

Michael Zink
Dept. of Electrical and Computer Engineering
University of Massachusetts, Amherst, USA
zink@ecs.umass.edu

## ABSTRACT

We consider the problem of allocating networked resources in dynamic environment, such as cloud computing platforms, where providers strategically price resources to maximize their utility. Resource allocation in these environments, where both providers and consumers are selfish agents, presents numerous challenges since the number of consumers and their resource demand is highly dynamic. While numerous auction-based approaches have been proposed in the literature, this paper explores an alternative approach where providers and consumers automatically negotiate resource leasing contracts. Since resource demand and supply can be dynamic and uncertain, we propose a distributed negotiation mechanism where agents negotiate over both a contract price and a decommitment penalty, which allows agents to decommit from contracts at a cost. We compare our approach experimentally, using representative scenarios and workloads, to both combinatorial auctions and the fixed-price model used by Amazon's Elastic Compute Cloud, and show that the negotiation model achieves a higher social welfare.

## Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems

## General Terms

Algorithms, Economics, Experimentation

## Keywords

Cloud computing, automated negotiation, negotiation strategy

## 1. INTRODUCTION

Cloud computing platforms enable consumers to programmatically rent multiple types of Internet-accessible computing resources. In many cases, these platforms use recent advances in virtualization to make the resources appear to the consumer as raw hardware components, such as machines, storage block devices, sensors, or network links. For example, Amazon currently operates both the Elastic Compute Cloud (EC2) and the Elastic Block Store (EBS), where consumers programmatically rent virtual machines and block devices, respectively. As another example, GENI [1] is a recent NSF initiative that uses a similar paradigm but incorporates a wider range of hardware components, including not only machines and block devices, but also sensors, mobile devices, and the network links connecting them, from a wider range of providers, including universities and industry research labs.

**Cite as:** Automated negotiation with decommitment for dynamic resource allocation in cloud computing, Bo An, Victor Lesser, David Irwin, Michael Zink, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. 981-988

There are many reasons why market-oriented mechanisms are attractive for regulating resource supply and demand for these platforms. Amazon's goal is to make a profit by renting their resources to consumers for more than it costs to purchase and operate them. While GENI is initially operated as a non-profit platform, it allocates resources from multiple providers that dynamically donate and withdraw them, which makes centralized allocation difficult as the number of providers scales. Additionally, market-oriented allocation mechanisms are attractive since they encourage providers to contribute resources to GENI in exchange for (real or virtual) currency that increases their access. Recent work has explored a variety of both system [9, 11] and market [6, 12] structures for resource allocation in market-oriented cloud computing platforms.

In this paper, we focus on a general resource allocation problem that matches the characteristics of cloud computing platforms and their consumers. Namely, multiple self-interested agents supply or consume multiple types of resources, where 1) consumers dynamically enter and leave the market, 2) consumers have some bounded flexibility over when they require resources, and 3) a single provider cannot satisfy consumers' resource requirements. The first two characteristics are evident in current cloud platforms that are available to the general public, which use them to execute tasks that may or may not have hard deadlines. The motivation for 3) is natural for an infrastructure like GENI that allocates networked resources from multiple providers, and is also becoming more prevalent for profit-making enterprises like Amazon as competitors, such as RackSpace Cloud, become more prominent.

Given these characteristics, we consider the design of a market structure that allocates resources to their most *efficient* use. A straightforward approach would have all consumers submit both their resource requirements and bids to a single super agent that runs an auction, such as the well-known VCG auction [7], to allocate resources. Since VCG is not necessarily strategy-proof in dynamic settings, this approach does not necessarily result in the most efficient usage [15]. While efficient online mechanisms have been proposed for dynamic environments, they only work in constrained settings and often rely on strong assumptions about agents' knowledge [15]. Further, finding an auctioneer that selfish agents will trust and comply with is difficult. Alternatively, each consumer could run the VCG auction separately, but a provider may not truthfully report its information due to the existence of other auctions.

In this paper, we present a negotiation mechanism in which agents make contracts for resource leases, which bind a set of resources from a provider to a consumer for a fixed time interval. To accommodate the highly dynamic nature of cloud computing platforms, we introduce a negotiation mechanism where an agent is able to decommit from a contract by paying a penalty to the other contract party. Thus, an agent may find it advantageous to decommit from existing contracts. Rather than setting decommitment penalties exogenously, we propose negotiating simultaneously over contract prices and decommitment penalties since it is difficult for system designers to decide the *optimal* contract prices and decommitment penalties that maximize the social welfare in dynamic environments involving multiple agents. We show that allowing decommitment improves the efficiency of the resource allocation mechanism.

Negotiation with uncertainty is both the most challenging problem in the negotiation literature [10], and the most practical problem for real cloud computing platforms. The literature provides a limited number of closed form results with narrow uncertainty settings using bilateral bargaining that considers only one type of uncertainty, such as a negotiation deadline [10] or reserve price [3]. In contrast, we consider negotiation between multiple agents in dynamic environments where there are multiple types of uncertainty that increases the difficulty of computing agents' rational equilibrium strategies. As a result, we bound agents' rationality and design negotiation strategies for them following the *negotiation decision functions* paradigm [5, 8]. Our negotiation problem is complex due to market dynamics, uncertainty, multiple contracting opportunities, resource competition, and decommitment. Rather than explicitly model these inter-dependent factors and determine each agent's best decisions through an intractable combined optimization, we connect these inter-dependent factors indirectly and develop a set of heuristics to approximate agents' decision-making during negotiation. The distinguishing characteristic of our negotiation agents is their flexibility to adjust their decisions, such as making offers, by reacting to changing negotiation status, while also considering the time constraints, resource competition, and resource cost.

We evaluate our negotiation mechanism on a simulation testbed against two well-known mechanisms—combinatorial auctions and Amazon's fixed-price model. Experimental results show that our negotiation mechanism achieves a higher social welfare than either mechanism in a wide range of scenarios. Further, we show that setting penalties through negotiation achieves a higher social welfare than exogenous mechanisms for setting penalties. Section 2 introduces the resource allocation problem using GENI as the motivation, while Section 3 formalizes the problem. We detail the negotiation strategies of our providers and consumers in Section 4 and Section 5, respectively. Finally, Section 6 experimentally evaluates the efficiency of the negotiation model, while Section 7 concludes.

## 2. RESOURCE ALLOCATION IN GENI

We explore our resource allocation problem in the context of NSF's GENI initiative [1], which is building a prototype of a shared experimental infrastructure to investigate next-generation Internet applications. GENI is similar to other cloud platforms in that it exposes network-accessible APIs for consumers to lease virtualized hardware components, although GENI offers a more diverse collection of resources donated by many providers, such as universities and industry research labs. The intent is for researchers to experiment with new Internet protocols and applications by reserving collections of geographically distributed hardware components and the network links connecting them, e.g., via Internet2 or NLR. A core concept for GENI and other cloud computing platforms is *resource leasing*.

Since GENI allocates resources from multiple providers, it uses one or more *Clearinghouses* to mediate the allocation. Providers delegate the right to allocate their resources to these Clearinghouses, which aggregate the resources and allocate them to researchers. As with Amazon's EC2 and EBS, GENI allocates virtualized hardware components to leverage statistical multiplexing and allow multiple researchers to use one hardware component simultaneously.

GENI consumers acquire resources from one or more Clearinghouses that broker the transactions for multiple providers. The initial intent is for the GENI Project Office to operate a small number of Clearinghouses, but, in general, there may be multiple Clearinghouses operated by governments, companies, or university-led consortiums. While the initial prototype's scale does not warrant market-based mechanisms, reaching GENI's goal for Internet-scale operation—allocating millions of components—motivates a market-oriented approach. Further, decentralizing resource allocation among multiple Clearinghouses gives GENI's architecture the flexibility to introduce market-oriented approaches incrementally in only a few Clearinghouses initially. We chose GENI as our motivation because its decentralized design is amenable to incrementally introducing market-oriented approaches and its structure is still open for debate. Further, we believe GENI's goal as a platform for experimental research should also include research on its own resource allocation mechanisms.

## 3. THE NEGOTIATION MODEL

### 3.1 The Resource Allocation Problem

We treat each consumer as a buyer and each provider as a seller, where $\mathcal{B}$ denotes the set of buyers and $\mathcal{S}$ denotes the set of sellers. Each buyer $\mathbf{b} \in \mathcal{B}$ has a high level task $\tau$, such as an experiment. The task $\tau$ of buyer $\mathbf{b}$ has the following attributes:

- A resource set $\mathcal{R}_\mathbf{b}$ and the quantity of units $\tau$ requires. For a resource $r \in \mathcal{R}_\mathbf{b}$, $\tau$ requires $q(\mathcal{R}_\mathbf{b}, r)$ units of resource $r$.
- Task generation time $tg(\mathbf{b})$ when the task is generated.
- Earliest start time $est(\mathbf{b})$ where task $\tau$ cannot start before time $est(\mathbf{b})$. Generally $est(\mathbf{b}) > tg(\mathbf{b})$ and $\mathbf{b}$ can use the time between $est(\mathbf{b})$ and $tg(\mathbf{b})$ to acquire resources.
- The period $pd(\mathbf{b})$ of resource usage, such that $\mathbf{b}$ must use resources $\mathcal{R}(\mathbf{b})$ for a period of length $pd(\mathbf{b})$.
- Deadline $dl(\mathbf{b})$ that indicates the latest start time of the task of a buyer $\mathbf{b}$. Since $dl(\mathbf{b}) \geq est(\mathbf{b})$. If $dl(\mathbf{b}) > est(\mathbf{b})$, the buyer has the flexibility to determine the start time of the experiment. Note that the task must finish before $dl(\mathbf{b}) + pd(\mathbf{b})$, and a rational buyer will not negotiate after $dl(\mathbf{b})$.
- Value $v_\mathbf{b}(t)$ represents the value $\mathbf{b}$ attaches to task completion as a function of completion time $t$. Following [13], $\mathbf{b}$ has its maximum value at time $est(\mathbf{b}) + pd(\mathbf{b})$ and its minimum value at time $dl(\mathbf{b}) + pd(\mathbf{b})$.

Each seller $\mathbf{s} \in \mathcal{S}$ has different types of resources $\mathcal{R}_\mathbf{s}$ in varying quantities, $q(\mathcal{R}_\mathbf{s}, r)$ units of resource $r \in \mathcal{R}_\mathbf{s}$, and suffers a cost $c_\mathbf{s}(r)$ for providing each unit of resource $r \in \mathcal{R}_\mathbf{s}$ for a unit time period. This model follows GENI in that sellers have different types of resources, although we simplify our problem by allowing only one "plan" for each task. While we specify only a single set of resources to satisfy each task, in general, multiple different types of resources may be able to satisfy a task. For example, a researcher may either plan an experiment with a small number of resources for a long duration, or a large number of resources and a short duration. In these cases, we can extend our formulation to include multiple plans.

We assume each buyer is able to discover the set of resources each seller provides. This assumption is reasonable since each seller is willing to let others to know its capability, and, from a single agent's perspective, knowing other agents' information may help it to develop appropriate strategies. For example, if a buyer knows that the resource competition is low, it may offer a lower price. We assume that 1) each buyer knows each seller's expected cost $c_\mathbf{b}(r)$ of providing a resource $r$ ; and 2) each agent has knowledge about the demand/supply ratio $\psi(r)$ of resource $r$ over time. This assumption is not more restrictive than related work [5, 14]. Further, in dynamic markets, a buyer can estimate a seller's cost and market competition by analyzing its negotiation history. We explore the sensitivity of this assumption in our experiments.

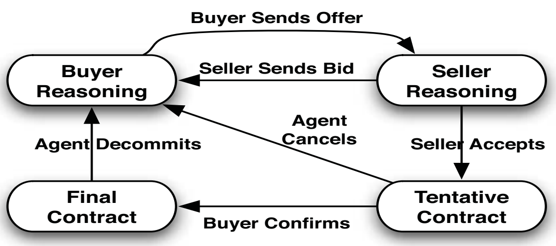

**Figure 1: Finite state machine for the negotiation protocol**

## 3.2 Negotiation Protocol

This work extends the alternating offers protocol [16], which has been widely used for bilateral bargaining. Before we formally define the protocol, we first define agents' possible actions:

- $\mathit{offer}[o]$, where $o$ is buyer $\mathbf{b}$'s offer to a seller $\mathbf{s}$. An offer $o$ is of the form $\langle pr, pe, \mathcal{R}, est, pd, dl\rangle$ where $pr$ is the offering price, $pe$ is the decommitment penalty, $\mathcal{R}$ specifies the set of required resources and their quantities, $est$ is the earliest start time of providing resources, $pd$ is the duration, and $dl$ is the latest time for providing resources. Note that when $dl > est$, the buyer provides a flexible schedule and the receiving seller is able to decide the exact start time of providing resources.
- $accept[o]$. When a seller $\mathbf{s}$ receives an offer $o'$, $\mathbf{s}$ is able to accept the offer resulting in the two agents reaching a *tentative* agreement. If $dl(o') > est(o')$, $\mathbf{s}$ must decide the exact start time of providing $\mathcal{R}(o')$ and $dl(o)$ should be equal to $est(o)$.
- $bid[\mathcal{Q}]$. When a seller $\mathbf{s}$ receives an offer $o'$ and the offer is not acceptable, $\mathbf{s}$ can send quotes $\mathcal{Q}$ for its available resources. Each quote $Q \in \mathcal{Q}$ describes the quantity of resource $r \in \mathcal{R}(o')$ and its asking price.
- $\mathit{confirm}[o]$. When a seller $\mathbf{s}$ accepts an offer $o$, two agents reach a *tentative* agreement and the buyer can *confirm* the tentative agreement. If $\mathbf{b}$ confirms the tentative agreement, then the agreement becomes a *final* agreement.
- $cancel[o]$. After two agents make a tentative agreement, any agent can *cancel* the agreement *without* paying a penalty. Then, negotiation between the two agents fails with no agreement.
- $decommit[o]$. After a final agreement is made, an agent has the opportunity to decommit from the agreement and the decommiting agent pays the penalty to the other party. Note that after time $est(o)$, no agent can decommit from the agreement. Furthermore, the decommiting agent pays the penalty after decommitment happens.

Figure 1 shows the finite state machine for the negotiation between $\mathbf{b}$ and $\mathbf{s}$. The initial state is "*buyer reasoning*" in which $\mathbf{b}$ decides how to make the offer. After $\mathbf{b}$ sends an offer to $\mathbf{s}$, the state is "*seller reasoning*" in which $\mathbf{s}$ is deciding whether to accept the offer or make a bid. If $\mathbf{s}$ accepts the offer, a tentative agreement is made. Otherwise, $\mathbf{s}$ sends a bid to $\mathbf{b}$ and then it is $\mathbf{b}$'s turn to decide its offer. If $\mathbf{b}$ confirms a tentative agreement, the negotiation is in the "*final agreement*" state. If one agent cancels a tentative agreement or decommits from a final agreement, their negotiation fails and $\mathbf{b}$ can restart to make an offer.

An important feature of our negotiation model is that many buyer-seller pairs can negotiate simultaneously. In addition to the decommiting action, we also introduce another pair of actions *"confirm"* and *"cancel"*. With the two actions, if a seller accepts an offer, a buyer can still have the chance to *"decommit"* from the agreement without paying a penalty. Assume that $\mathbf{b}$ only needs one resource. In absence of the action cancel, if $\mathbf{b}$ makes offers to multiple sellers that all accept, $\mathbf{b}$ must buy multiple items or decommit from agreements by paying penalties. Accordingly, $\mathbf{b}$ may only propose to one seller. In presence of actions $cancel$ and $\mathit{confirm}$, $\mathbf{b}$ can choose only one contract while negotiating with multiple sellers simultaneously.

Next we formalize the notion of utility. The utility of buyer $\mathbf{b}$ depends on its task completion time and its payment, including 1) its payment for getting resources, and 2) penalties it pays to other agents and receives from other agents. $\mathbf{b}$'s utility at time $t$ is

$$u_{\mathbf{b}}(t) = \begin{cases} v_{\mathbf{b}}(t) + \rho_{\mathbf{b}} & \text{if } \mathbf{b}\text{'s task is finished} \\ \rho_{\mathbf{b}} & \text{otherwise} \end{cases}$$

where $\rho_{\mathbf{b}}$ is the balance of the buyer $\mathbf{b}$—the difference between the payment received and the payment paid to other agents.

The total utility of each seller $\mathbf{s} \in \mathcal{S}$ from time 0 to time $t$ is $u_{\mathbf{s}}(t) = \rho_{\mathbf{s}} - c_{\mathbf{s}}$ where $\rho_{\mathbf{s}}$ is the balance of the seller $\mathbf{s}$ at time $t$ and $c_{\mathbf{s}}$ is seller $\mathbf{s}$'s cost for providing resources from the beginning to time $t$.

## 4. BUYERS' NEGOTIATION STRATEGY

Before formally defining a buyer $\mathbf{b}$'s negotiation strategy, we first discuss other important factors we consider:

- **Deadline Pressure**. $\mathbf{b}$ must satisfy its resource requirements by the deadline $dl(\mathbf{b})$, which is a hard constraint.
- **Sellers' Cost**. A rational seller will not accept a price lower than its cost. A buyer needs to offer different prices for different resources which have different costs.
- **Single Provider**. If $\mathcal{R}_{\mathbf{b}} \subseteq \mathcal{R}_{\mathbf{s}}$, $\mathbf{b}$ can make a *full* agreement with a single seller $\mathbf{s}$ which can satisfy $\mathbf{b}$'s resource requirements. Otherwise, it must request resources from different sellers and make a set of *partial* agreements, each of which can only satisfy part of $\mathbf{b}$'s resource requirements. The negotiation for the latter case is more complex since $\mathbf{b}$ must have contracts with multiple sellers, and making no agreement may be better than making agreements which cannot satisfy $\mathbf{b}$'s requirements. Furthermore, if $\mathbf{b}$'s resource requirements are satisfied through a set of contracts, the set of contracts should be *compatible* in that all contracts should provide resources during the same time frame.

In summary, a buyer agent's optimal action at each time point is affected by many factors and it is impossible to construct an integrated framework in which all these factors are optimized concurrently. Instead, this work connects those inter-dependent factors indirectly and develops a set of heuristics to approximate agents' decision making. In what follows we first introduce buyer $\mathbf{b}$'s strategy (Algorithm 1) informally and then present it formally.

One distinguishing feature of $\mathbf{b}$'s negotiation strategy is that it always tries to make two sets of agreements both of which can satisfy its resource requirements. Therefore, if a set of agreements is decommited, $\mathbf{b}$ can use the other agreement set to satisfy its resource requirements. If both set of agreements are not decommited, when one set of agreements starts execution, $\mathbf{b}$ can decommit from the other set of agreements. If the start time of two sets of agreements are the same, $\mathbf{b}$ will choose one set of agreements to decommit before the execution starts. Specifically, $\mathbf{b}$ is always trying to make a final full agreement and a set of partial final agreements both of which can satisfy its resource requirements. In case no single seller can satisfy $\mathbf{b}$'s resource requirements, $\mathbf{b}$ makes two sets of partial final agreements. In addition, $\mathbf{b}$ sets a small penalty for each partial agreement and thus it only needs to pay a small penalty for decommiting from any partial agreement. While a buyer can make more agreements to increase the probability that its task can be finished, it has to pay more for those agreements since for each unnecessary agreement, it has to pay either the penalty or the agreement price. Alternatively, if a buyer only makes one set of agreements, it may be difficult to find another set of agreements to satisfy the buyer's resource requirements when some

**Algorithm 1**: Negotiation strategy of buyer $\mathbf{b}$

```
Set $\mathcal{A}_\mathbf{b} = \emptyset$ , $\mathcal{TA}_\mathbf{b} = \emptyset$, $t$ is the real time (initially, $t = tg(\mathbf{b})$).
Let $estp = est(\mathbf{b})$, $dlp = dl(\mathbf{b})$ be the earliest start time and deadline for
negotiating partial contracts.
Let $\kappa$ (e.g., 4) be the total number of times to try different execution schedules of
partial agreements. Let $T_{bk} = (dl(\mathbf{b}) - tg(\mathbf{b}))/\kappa + tg(\mathbf{b})$.
while $t < dl(\mathbf{b})$ and the task has not started to run do          /* main loop */
    foreach $\mathbf{s} \in \mathcal{S}$ such that $\mathcal{R}_\mathbf{s} \cap \mathcal{R}_\mathbf{b} \neq \emptyset$ and $\mathbf{b}$ is not negotiating with $\mathbf{s}$ do
        send offer GENERATE_OFFER($\mathcal{A}_\mathbf{b}, \mathcal{TA}_\mathbf{b}, \mathbf{s}$) to seller $\mathbf{s}$;
    end
    if seller $\mathbf{s}$ sends a bid $\mathcal{Q}$ then
        update the bid set;
    end
    if seller $\mathbf{s}$ accepts offer $o$ then
        EVALUATE_ACCEPT($\mathcal{A}_\mathbf{b}, \mathcal{TA}_\mathbf{b}, o, \mathbf{s}$);
    end
    if $t > estp$ then
        decommit (cancel) agreements $\mathcal{A}_\mathbf{b} - \mathcal{A}^f_\mathbf{b}$ $(\mathcal{TA}_\mathbf{b} - \mathcal{TA}^f_\mathbf{b})$;
        set $estp = \max\{t, est(\mathbf{b})\}$, $dlp = dl(\mathbf{b})$;
    end
    if $t \geq T_{bk}$ then
        $\kappa - -$, $T_{bk} = (dl(\mathbf{b}) - t)/\kappa + t$;
        if $R(\mathcal{A}_\mathbf{b} + \mathcal{TA}_\mathbf{b} - \mathcal{A}^f_\mathbf{b} - \mathcal{TA}^f_\mathbf{b}) \subset \mathcal{R}_\mathbf{b}$ then
            decommit (cancel) agreements $\mathcal{A}_\mathbf{b} - \mathcal{A}^f_\mathbf{b}$ $(\mathcal{TA}_\mathbf{b} - \mathcal{TA}^f_\mathbf{b})$;
            set $estp = \max\{t, est(\mathbf{b})\}$, $dlp = dl(\mathbf{b})$;
        end
    end
    if seller $\mathbf{s}$ decommits from agreement $o$ then
        remove $o$ from $\mathcal{A}_\mathbf{b}$;
    end
    if seller $\mathbf{s}$ cancel tentative agreement $o$ then
        remove $o$ from $\mathcal{TA}_\mathbf{b}$;
    end
    if seller $\mathbf{s}$ has not responded to $\mathbf{b}$'s proposing $o$ for a period $\epsilon$ then
        send offer GENERATE_OFFER($\mathcal{A}_\mathbf{b}, \mathcal{TA}_\mathbf{b}, \mathbf{s}$) to seller $\mathbf{s}$;
    end
    if seller $\mathbf{s}$ has not responded to $\mathbf{b}$'s accepting offer $o$ for a period $\epsilon$ then
        cancel the tentative agreement $o$ and remove $o$ from $\mathcal{TA}_\mathbf{b}$;
    end
end
cancel from all tentative agreements $\mathcal{TA}_\mathbf{b}$ ;
if the task has started to run then
    decommit from each agreement $o \in \mathcal{A}_\mathbf{b}$ if $o$ is useless and $pr(o) > pe(o)$;
else
    decommit from each agreement $o \in \mathcal{A}_\mathbf{b}$ if $pr(o) > pe(o)$;
end
```

**Algorithm 2**: **GENERATE_OFFER**$(\mathcal{A}_\mathbf{b}, \mathcal{TA}_\mathbf{b}, \mathbf{s})$

```
if $|\mathcal{A}^f_\mathbf{b} \cup \mathcal{TA}^f_\mathbf{b}| = 0$ and $\mathcal{R}_\mathbf{b} \subseteq \mathcal{R}_\mathbf{s}$ then
    Let price be $pr = pr(\mathcal{R}_\mathbf{b}, \max\{t, est(\mathbf{b})\}, dl(\mathbf{b}), t)$ using Eq. (1);
    Let the penalty be $pe = v_\mathbf{b}(\max\{t, est(\mathbf{b})\} + pd(\mathbf{b})) - pr$;
    return offer $o = \langle pr, pe, \mathcal{R}_\mathbf{b}, \max\{t, est(\mathbf{b})\}, pd(\mathbf{b}), dl(\mathbf{b})\rangle$;
else
    Let $\mathcal{R} = \mathcal{R}_\mathbf{s} \cap (\mathcal{R}_\mathbf{b} - R(\mathcal{A}_\mathbf{b} + \mathcal{TA}_\mathbf{b} - \mathcal{A}^f_\mathbf{b} - \mathcal{TA}^f_\mathbf{b}))$;
    if $\mathcal{R} = \emptyset$ then
        return offer $o = null$;
    end
    if $estp \neq dlp$ then
        If possible, set the value of $estp = dlp > t + \sigma$ based on bids from
        sellers such that the available resource from $estp$ to $estp + pd(\mathbf{b})$ can
        satisfy $\mathbf{b}$'s resource requirements;
    end
    Let price be $pr = pr(\mathcal{R}, estp, dlp, t)$ using Eq. (2);
    Let the penalty be $pe = \alpha \cdot pr$ (e.g., $\alpha = 0.05$);
    return offer $o = \langle pr, pe, \mathcal{R}, estp, pd(\mathbf{b}), dlp\rangle$;
end
```

**Algorithm 3**: **EVALUATE_ACCEPT**$(\mathcal{A}_\mathbf{b}, \mathcal{TA}_\mathbf{b}, o, \mathbf{s})$

```
Let new offer $o' =$ GENERATE_OFFER($\mathcal{A}_\mathbf{b}, \mathcal{TA}_\mathbf{b}, \mathbf{s}$);
if 1) $\mathcal{R}(o) = \mathcal{R}(o') = \mathcal{R}_\mathbf{b}$, $|\mathcal{A}^f_\mathbf{b} \cup \mathcal{TA}^f_\mathbf{b}| = 0$, and $pr(o) \leq pr(o')$ or 2)
$\mathcal{R}(o) = \mathcal{R}(o')$, $est(o) = est(o') = estp = dlp$, and $pr(o) \leq pr(o')$ then
    confirm agreement $o$ and add it to $\mathcal{A}_\mathbf{b}$;
else
    cancel agreement $o$;
end
```

agreements are decommited. Experimental results show that making two sets of agreements is better than making only one set of agreements and making more than two sets of agreements.

Another distinguishing feature of $\mathbf{b}$'s negotiation strategy is that while deciding the offering price $pr(\mathcal{R}, est, dl, t)$ of requesting resources $\mathcal{R}$ at time $t$ with earliest start time $est$ and latest start time $dl$, the following factors are considered. First, the pressure of deadline. The buyer makes more concessions when the deadline approaches. Such time-dependent concession strategies have been widely used in the literature [5,8]. Second, the cost $c_\mathbf{b}(r)$ of resource $r$. Intuitively, a buyer needs to pay more for a resource with a higher cost. Third, the demand/supply ratio $\psi(r)$ of a resource $r$. The higher the ratio, the higher the price for the resource. Market (resource) competition has the largest effect on the equilibrium price [4]. Formally, the offering price $pr(\mathcal{R}_\mathbf{b}, est, dl, t)$ for all resources $\mathcal{R}_\mathbf{b}$ is defined as

$$c(\mathcal{R}_\mathbf{b}) + \big(RP(est, dl) - c(\mathcal{R}_\mathbf{b})\big)\Big(\frac{t - tg(\mathbf{b})}{dl(\mathbf{b}) - tg(\mathbf{b})}\Big)^{\varepsilon} \tag{1}$$

where $c(\mathcal{R}_\mathbf{b}) = \sum_{r \in \mathcal{R}_\mathbf{b}} c_\mathbf{b}(r) q(\mathcal{R}_\mathbf{b}, r) pd(\mathbf{b})$ is the expected cost of resources $\mathcal{R}_\mathbf{b}$ and $RP(est, dl)$ is the expected value of finishing the task with earliest start time $est$ and latest start time $dl$. Formally,

$$RP(est, dl) = \begin{cases} \frac{\int_{est}^{dl} v_\mathbf{b}(pd(\mathbf{b}) + t')dt'}{dl - est} & \text{if } dl \neq est \\ v_\mathbf{b}(pd(\mathbf{b}) + est) & \text{otherwise} \end{cases}$$

When $t = tg(\mathbf{b})$, $pr(\mathcal{R}_\mathbf{b}, est, dl, t) = c(\mathcal{R}_\mathbf{b})$, which is the lowest offer acceptable to sellers. When $t = dl(\mathbf{b})$, $pr(\mathcal{R}_\mathbf{b}, est, dl, t) = RP(est, dl)$, which is the highest offering price of $\mathbf{b}$ since the buyer will get negative utility if it pays more than its value of finishing the task. Parameter $\varepsilon > 0$ is used to model how the buyer $\mathbf{b}$ increases its offering price with the increase of time $t$. With infinitely many values of $\varepsilon$, there are infinitely many possible strategies in making concessions with respect to the remaining time. However, they can be classified into: *1) Linear*: $\varepsilon = 1$, *2) Conciliatory*: $0 < \varepsilon < 1$, and *3) Conservative*: $\varepsilon > 1$ [8]. We adopt the linear strategy for $\mathbf{b}$.

By considering both resources' costs and market competition, the buyer's offering price for $\mathcal{R} \subset \mathcal{R}_\mathbf{b}$ is calculated in the following way:

$$pr(\mathcal{R}, est, dl, t) = \sum_{r \in \mathcal{R}} q(\mathcal{R}, r) pd(\mathbf{b}) pr(r, est, dl, t) \tag{2}$$

$$pr(r, est, dl, t) = c_\mathbf{b}(r) + \frac{\big(pr(\mathcal{R}_\mathbf{b}, est, dl, t) - c(\mathcal{R}_\mathbf{b})\big)\psi(r) c_\mathbf{b}(r)}{pd(\mathbf{b}) \sum_{r \in \mathcal{R}_\mathbf{b}} \psi(r) c_\mathbf{b}(r) q(\mathcal{R}_\mathbf{b}, r)}$$

$pr(r, est, dl, t)$ is the price for one unit of resource $r$ and it increases with its cost $c_\mathbf{b}(r)$ and the demand/supply ratio $\psi(r)$.

Let $\mathcal{A}_\mathbf{b}$ be $\mathbf{b}$'s final agreements and $\mathcal{TA}_\mathbf{b}$ be $\mathbf{b}$'s tentative agreement set. Let $\mathcal{A}^f_\mathbf{b} \subseteq \mathcal{A}_\mathbf{b}$ ($\mathcal{TA}^f_\mathbf{b} \subseteq \mathcal{TA}_\mathbf{b}$, respectively) be the set of final full (tentative, respectively) agreements. Let $R(\mathcal{A})$ be the set of resources provided by the agreement set $\mathcal{A}$. If $\mathbf{b}$ has no full agreement, i.e., $|\mathcal{A}^f_\mathbf{b} \cup \mathcal{TA}^f_\mathbf{b}| = 0$, it will request for all resources $\mathcal{R}_\mathbf{b}$ from sellers which can satisfy its full resource requirements and request for resources $\mathcal{R}_\mathbf{s} \cap (\mathcal{R}_\mathbf{b} - R(\mathcal{A}_\mathbf{b} + \mathcal{TA}_\mathbf{b} - \mathcal{A}^f_\mathbf{b} - \mathcal{TA}^f_\mathbf{b}))$ from each seller $\mathbf{s}$ which can only satisfy part of its resource requirements. If $\mathbf{b}$ has a full agreement, it will request for resources $\mathcal{R}_\mathbf{s} \cap (\mathcal{R}_\mathbf{b} - R(\mathcal{A}_\mathbf{b} + \mathcal{TA}_\mathbf{b} - \mathcal{A}^f_\mathbf{b} - \mathcal{TA}^f_\mathbf{b}))$ from each seller $\mathbf{s} \in \mathcal{S}$.

When buyer $\mathbf{b}$ wants to acquire resources $\mathcal{R}$ from seller $\mathbf{s}$ at time $t$, in addition to specifying the offering price, it also decides the decommitment penalty $pe$, and the task execution period. First consider the case in which $|\mathcal{A}^f_\mathbf{b} \cup \mathcal{TA}^f_\mathbf{b}| = 0$ and $\mathcal{R} = \mathcal{R}_\mathbf{b}$. In this case, $\mathbf{b}$ simply requests its earliest execution start time $est(\mathbf{b})$, deadline $dl(\mathbf{b})$, and the execution period $dl(\mathbf{b})$. The seller will decide the exact start time. We use a simple rule to decide the decommitment penalty: the lower

**Algorithm 4**: Negotiation strategy of seller $\mathbf{s}$

```
Set $\mathcal{A}_\mathbf{s} = \emptyset, \mathcal{TA}_\mathbf{s} = \emptyset$;
if buyer $\mathbf{b}$ decommits from agreement $o$ then
    remove $o$ from $\mathcal{A}_\mathbf{b}$;
end
if buyer $\mathbf{b}$ cancels tentative agreement $o$ then
    remove $o$ from $\mathcal{TA}_\mathbf{b}$;
end
if buyer $\mathbf{b}$ confirms tentative agreement $o$ then
    remove $o$ from $\mathcal{TA}_\mathbf{b}$ and add $o$ to $\mathcal{A}_\mathbf{b}$;
end
if buyer $\mathbf{b}$ sends an offer $o$ then
    run the greedy algorithm for $\mathbf{OPT}_\mathbf{s}(\mathcal{RA}_\mathbf{s}, \mathcal{A}_\mathbf{s}, \mathcal{TA}_\mathbf{s}, \mathcal{O}_\mathbf{s})$;
end
if buyer $\mathbf{b}$ has not responded to $\mathbf{s}$'s proposing $o$ for a period $\epsilon$ then
    send offer to buyer $\mathbf{s}$ with a price $c_\mathbf{s}(r)\varphi(r,t)$ for each resource $r$;
end
if buyer $\mathbf{b}$ has not responded to $\mathbf{s}$'s accepting offer $o$ for a period $\epsilon$ then
    cancel the tentative agreement $o$ and remove $o$ from $\mathcal{TA}_\mathbf{s}$;
end
```

the price $pr(\mathcal{R}_\mathbf{b}, est, dl, t)$, the higher the penalty. In other words, $\mathbf{b}$ does not want a cheap full agreement to be decommited. One example rule to set the penalty is $pe = v_\mathbf{b}(\max\{t, est(\mathbf{b})\} + pd(\mathbf{b})) - pr$.

We also consider the case $|\mathcal{A}^f_\mathbf{b} \cup \mathcal{TA}^f_\mathbf{b}| > 0$ or $\mathcal{R} \neq \mathcal{R}_\mathbf{b}$. In this case, $\mathbf{b}$ must decide what time period to request resources since different sellers need to provide resources in the same time period and this decision making is difficult due to uncertainty and agents' selfishness. In this work, $\mathbf{b}$ decides the task execution schedule for partial agreements based on its information about sellers' available resources, which can be obtained from the bid messages and acceptance messages from sellers. Note that there is no guarantee that $\mathbf{b}$ can get part or all of $\mathbf{s}$'s available resources due to the market dynamics. Specifically, $\mathbf{b}$ searches from time $\max\{t + \sigma, est(\mathbf{b})\}$ until $dl(\mathbf{b})$ and sets the task start time $est$ as the earliest time point from which sellers' available resources from time $est$ to $est + pd(\mathbf{b})$ can satisfy the buyer's resource requirements. We use the parameter $\sigma > 0$ to allow the buyer the flexibility to negotiate for resources. We choose this simple rule for two reasons. First, since a buyer's value of finishing a task generally decreases with the task start time, the buyer can potentially achieve a higher utility if it negotiates for a set of agreements with an early task start time. Second, due to market dynamics and agents' strategic interaction, it is impossible to determine the best start time. If there is no start time for which the buyer's resource requirements can be satisfied, the buyer simply sets $est = est(\mathbf{b})$ and $dl = dl(\mathbf{b})$ and it will not confirm any partial agreement. Using our simple rule, we set the decommitment penalty in this case to $pe = \alpha \cdot pr$, where $0 < \alpha < 0.2$.

Once the task execution schedule of partial agreements is determined, buyer $\mathbf{b}$ will request resources from sellers according to the task execution schedule. However, the selected task execution schedule may cause buyer $\mathbf{b}$ to fail to find agreements to satisfy its resource requirements. Therefore, it is important for buyer $\mathbf{b}$ to try other task execution schedules if it fails to get agreements with the current schedule. In other words, buyer $\mathbf{b}$ should have the "backtracking" ability of changing its task execution schedule. In this work, a buyer agent will change its task execution schedule if it fails to satisfy its resource requirements for a given time. When a buyer $\mathbf{b}$ changes its task execution schedule, it will first decommit from its other partial agreements.

## 5. SELLERS' NEGOTIATION STRATEGY

Our negotiation strategy for the seller (Algorithm 4) has two features. First, the seller adopts a "myopic" negotiation strategy in the sense that it accepts an offer if and only if it can gain some *immediate* payoff by accepting the offer, and will not consider the effect of its current action on the future utilities. In addition, when a seller receives an offer, it will first make acceptance and decommitment decisions, and then generate bids to the buyer if the offer is not acceptable. Second, the seller decides the acceptable price for a set of resources based on resource competition and cost of resources.

If the competition of a resource is high, a seller has an expectation that it will receive a high price for the resource. When a seller receives an offer $o$, it first generates a threshold price $\phi(o)$. If $pr(o) < \phi(o)$, it will not accept the offer. The threshold price $\phi(o)$ is defined as

$$\phi(o) = \textstyle\sum_{r\in\mathcal{R}(o)} q(\mathcal{R}(o), r)c_\mathbf{s}(r)(1 + \psi(r))pd(o)$$

in which $c_\mathbf{s}(r)(1+\psi(r))pd(o)$ is seller $\mathbf{s}$'s "asking" price for one unit of resource $r$. Obviously, $\phi(o) > cost(o) = \sum_{r\in\mathcal{R}(o)} q(\mathcal{R}(o), r)$ $c_\mathbf{s}(r)pd(o)$. If an offer is not acceptable, the seller will simply report its available resources in the buyer's request, as well as the unit price of each resource $r$ as $c_\mathbf{s}(r)(1 + \psi(r))$.

Since an agent can decommit from a final agreement, a seller can make more agreements than its capacity. In this case, a seller can decommit from an unsatisfiable agreement before the resource providing time. However, since an agent does not know whether the other agent will decommit from an agreement and the seller may pay a high penalty, we have chosen a seller's strategy where the seller may not make agreements beyond its capability. That is, without decommiting from any final agreement or canceling any tentative agreement, the seller must be able to fulfill its current running agreements $\mathcal{RA}_\mathbf{s}$, final agreements $\mathcal{A}_\mathbf{s}$, and tentative agreements $\mathcal{TA}_\mathbf{s}$. This strategy also implies that when a seller receives a message indicating confirmation of an agreement, it can fulfill the agreement without decommiting from any final agreement or cancel any tentative agreement.

The most difficult decision problem for the seller is how to handle a set of acceptable offers, which can be formulated as an optimization problem $\mathbf{OPT}_\mathbf{s}(\mathcal{RA}_\mathbf{s}, \mathcal{A}_\mathbf{s}, \mathcal{TA}_\mathbf{s}, \mathcal{O}_\mathbf{s})$: Given the running agreements $\mathcal{RA}_\mathbf{s}$, final agreements $\mathcal{A}_\mathbf{s}$, tentative agreements $\mathcal{TA}_\mathbf{s}$, and offers $\mathcal{O}_\mathbf{s}$, compute the set $\mathcal{KA}_\mathbf{s}$ of final agreements not to decommit, the set $\mathcal{KTA}_\mathbf{s}$ of tentative agreements to not to cancel, and the set $\mathcal{AO}_\mathbf{s}$ of offers to accept to maximize the following objective function

$$\sum_{o\in\mathcal{KA}_\mathbf{s}\cup\mathcal{KTA}_\mathbf{s}\cup\mathcal{AO}_\mathbf{s}} pr(o) - cost(o) - \sum_{o\in\mathcal{A}_\mathbf{s}-\mathcal{KA}_\mathbf{s}} pe(o)$$

with the constraint that $\mathbf{s}$ can fulfill final agreements $\mathcal{KA}_\mathbf{s}$ and tentative agreements $\mathcal{TA}_\mathbf{s} \cup \mathcal{AO}_\mathbf{s}$.

THEOREM 1. *The optimization problem* $\mathbf{OPT}_\mathbf{s}(\mathcal{RA}_\mathbf{s}, \mathcal{A}_\mathbf{s}, \mathcal{TA}_\mathbf{s}, \mathcal{O}_\mathbf{s})$ *is* $\mathcal{NP}$*-complete.*

The theorem's proof is a straightforward reduction from the 0-1 Knapsack problem. Thus, we propose a greedy algorithm to handle this computationally costly optimization problem. First, an agreement is treated as an offer and let $\Omega = \mathcal{A}_\mathbf{s} \cup \mathcal{TA}_\mathbf{s} \cup \mathcal{O}_\mathbf{s}$ be the set of offers that must be considered. $\mathbf{s}$'s revenue of accepting offer $o \in \Omega$ is

$$rv(o) = \begin{cases} pr(o) - cost(o) + pe(o) & \text{if } o \in \mathcal{A}_\mathbf{s} \\ pr(o) - cost(o) & \text{otherwise} \end{cases}$$

Next, all the offers $\Omega$ are sorted by decreasing revenue and offers are greedily picked in this order, starting with the first offer, and until no offers remain. Let $\Omega' = \emptyset$ be the set of accepted offers. When an offer $o$ is picked, add $o$ to $\Omega'$ and check whether the seller is able to fulfill all agreements $\Omega'$. Note that if $o \in \mathcal{O}_\mathbf{s}$ and $dl(o) > est(o)$, the seller must decide the schedule (the start time of providing resources and end time of providing resources) for providing resources specified in the offer. If the seller can fulfill all the agreements in $\Omega'$ and $o$ is an offer from a buyer, the seller will send an acceptance message to the buyer. If the seller cannot fulfill all the agreements in $\Omega'$, remove $o$ from $\Omega'$. If $o \in \mathcal{A}_\mathbf{s}$, then send a decommitment message to the buyer

**Table 1: Variables**

| *Variables* | Values |
|---|---|
| *Number of sellers* | $[5, 20]$ |
| *No. of resource types per seller* | $[2, 8]$ |
| *Quantity of a resource per seller* | $[2, 20]$ |
| *Unit cost of a resource* | $[10, 100]$ |
| *No. of resource types per buyer* | $[2, 6]$ |
| *Quantity of a resource per buyer* | $[2, 8]$ |
| *Value/cost ratio* | $[1.2, 5]$ |
| $pd(\mathbf{b})$ | $[10, 50]$ |
| $\frac{dl(\mathbf{b})-pd(\mathbf{b})-est(\mathbf{b})+1}{pd(\mathbf{b})}$ *(task execution flexibility)* | $[0, 7]$ |
| $\frac{est(\mathbf{b})-tg(\mathbf{b})+1}{pd(\mathbf{b})}$ *(negotiation time ratio)* | $[1, 8]$ |
| *resource demand/supply ratio* $\psi(r)$ | $[0.2, 10]$ |

involved in the agreement and pay the penalty. If $o \in \mathcal{TA}_\mathbf{s}$, send a cancel message to the buyer involved in the agreement.

## 6. EMPIRICAL EVALUATION

To evaluate the performance of our mechanism, we implement a simulation testbed consisting of a discrete time virtual marketplace and a set of trading agents. We generate all seller agents before the market opens and buyers dynamically enter the market, which matches real-world environments with a fixed number of well-known sellers.

### 6.1 Different Mechanisms

*Negotiation mechanism (NG)*: When a buyer enters the market, it negotiates with sellers following the protocol described in Section 3.2. At each time point, first all buyers are triggered and then all sellers are triggered. All agents employ the negotiation strategies described in Sections 4 and 5. A buyer quits the market when its task is finished or it fails to satisfy its resource requirements by its deadline. For comparison, we also implemented two other widely used mechanisms:

- *Combinatorial reverse auction (CRA)*: In combinatorial auctions [7], a large number of items are auctioned concurrently. In combinatorial reverse auctions, a buyer buys goods from many competing sellers. When a buyer enters the market, it announces its resource requirements, and sellers submit bids indicating the set of resources and their prices. Finally, the buyer determines the set of contracts. The buyer uses the well-known strategy-proof Vickrey auction mechanism. We assume that each seller has no knowledge of other buyers and, thus, each seller truthfully reports its available resources and their costs.
- *Fixed price scheme (Amazon) [2]*: Amazon EC2 is a web service that provides resizable compute capacity. The primary pricing mechanism for Amazon is a fixed price scheme with hourly charges per virtual machine. While using the Amazon scheme, a seller sets its price for each resource in advance and sellers constantly update their available resources. When a buyer enters the market, it decides the set of resources to buy. In our experiments, we tried different methods for setting price of each resource, where the price/cost ratio is 1, 2, 3, or 5.

The Amazon scheme is similar to *CRA*, except that in the Amazon scheme, a seller's payment from buyers is decided by the seller. In contrast, a seller's payment in *CRA* is the opportunity cost that its presence introduces to all the other agents. Note that when the price/cost in the Amazon scheme is 1, the Amazon scheme is equivalent to *CRA* in terms of the allocation since each seller will only charge its cost. Our negotiation model is also similar to *CRA* since sellers' accepting offers in the negotiation model are equivalent to submitting bids in the auction model. There are two main differences between our negotiation model and the other two models. First, in the negotiation model, agents are allowed to decommit from agreements. Second, there is a dynamic bargaining process in the negotiation model.

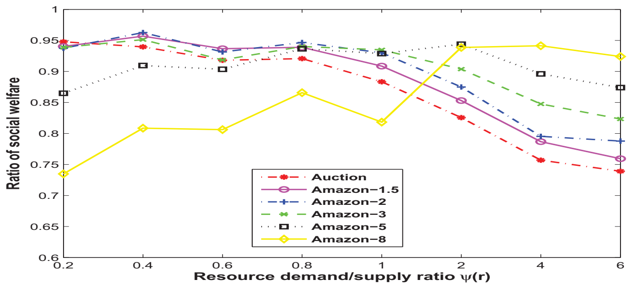

**Figure 2: Social welfare and resource competition**

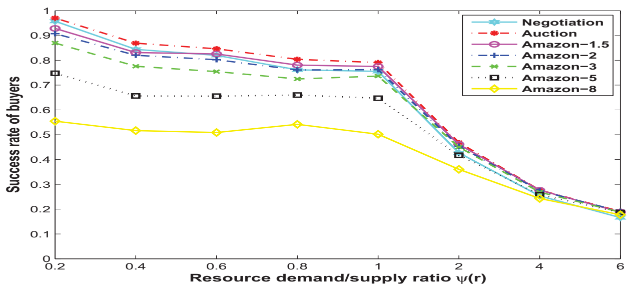

**Figure 3: Success rate and resource competition**

### 6.2 Experimental Settings and Measures

We performed a series of experiments in a variety of test environments using the parameters from Table 1. The parameters are inspired by the current design of the GENI infrastructure [1]. In the experiments, the number of sellers are in the range of $[5, 20]$, where each seller can provide 4 to 8 different types of resources. The quantity of a resource a seller can provide is in the range of $[2, 20]$. The cost of a resource per unit time is in the range of $[10, 100]$. Each buyer needs 2 to 6 different types of resources, and for each type of resource, a buyer needs 2 to 6 units. The length of resource usage is in the range of $[10, 50]$. The ratio $\frac{dl(\mathbf{b})-pd(\mathbf{b})-est(\mathbf{b})+1}{pd(\mathbf{b})} \in [0, 7]$ describes a buyer's flexibility of deciding when to start its task. Similarly, ratio $\frac{est(\mathbf{b})-tg(\mathbf{b})+1}{pd(\mathbf{b})} \in [1, 8]$ represents a buyer's time to negotiate for resources. We assume that each buyer has a linear value function in which the buyer gets the highest value when the task starts from $est(\mathbf{b})$ and the buyer gets the lowest value when the task starts at $dl(\mathbf{b})$. Value/cost ratio is used to generate a buyer's maximum value and minimum value based on sellers' cost of providing resources. $\psi(r) \in [0.2, 10]$ is the ratio of total resource requirements to total resource supply through the whole experiment horizon.

The main performance measure is the social welfare—the sum of all agents' utilities. Since the social welfare of a mechanism in different settings could be significantly different, we report the ratio of the social welfare of *CRA* and the Amazon mechanism to the social welfare of *NG*. We also report the success rate of different mechanisms—the percentage of buyers which successfully complete their tasks.

### 6.3 Results

Extensive stochastic simulations were carried out for all the combinations of variables in Table. 1. For each combination, we randomly generated over 5000 experiments and for each experiment, and tried all the three mechanisms and generated average performance measures. Even though extensive stochastic simulations were carried out for all the situations, due to space limitations, we only present the representative results. The length of each experiment is 1000 time units. We found that the confidence interval for each average value is very tight around the value, so the confidence intervals are not reported.

#### 6.3.1 *Performance of the negotiation mechanism*

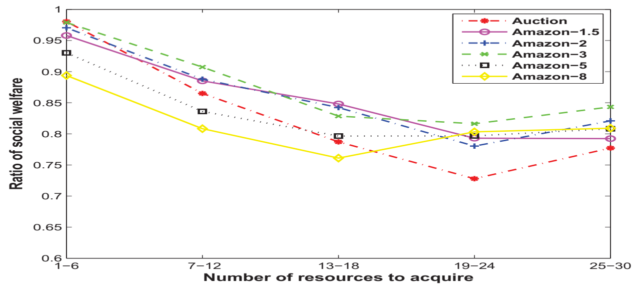

**Figure 4: Social welfare and number of resource to acquire**

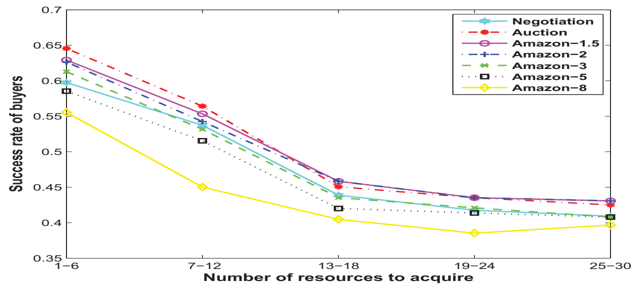

**Figure 5: Success rate and number of resource to acquire**

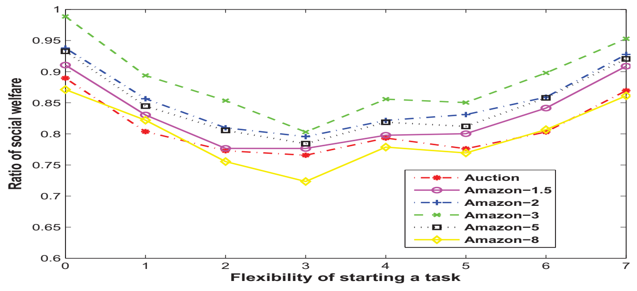

**Figure 6: Social welfare and the flexibility of starting a task**

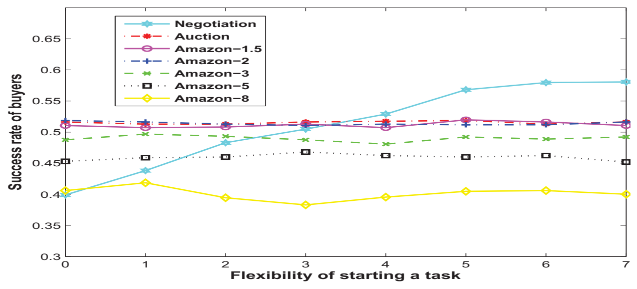

**Figure 7: Success rate and the flexibility of starting a task**

*Observation 1*: *NG* achieved about $13\%$ higher social welfare than any other evaluated mechanism. Figure 2 shows how the social welfare of different mechanisms changes with resource demand/supply ratio $\psi(r)$. We can observe that in all situations, *NG*'s social welfare is always higher than any other mechanism. Furthermore, when $\psi(r)$ is small (e.g., 0.2), *CRA* or the Amazon scheme with lower prices (e.g., Amazon-1.5) achieved higher social welfare than with higher prices (e.g., Amazon-8). In contrast, when $\psi(r)$ is large (e.g., 6), the Amazon scheme with higher prices (e.g., Amazon-8) achieved higher social welfare than *CRA* or Amazon scheme with lower prices. This observation is intuitive: When the resource competition is low, there are plenty of resources and each buyer can find them. However, when the resource competition is high, a mechanism can achieve a high social welfare if tasks with high revenues can be completed. If the price of each resource is low, a task with low revenue may get resources and a task with high revenue may fail to get resources since the resource were prematurely committed to the low revenue buyer and there was no way to decommit from the decision. In contrast, if a high price is set for each resource, only tasks with high revenues can get resources.

Figure 3 shows how the success rates of different mechanisms change with resource demand/supply ratio. First, a mechanism with a higher price has a lower success rate than that of a mechanism with a lower price. *NG*'s success rate is lower than some mechanisms with lower prices due to fact that in negotiation, each agent will not accept or offer any offer worse than its expectation. Second, with the increase of resource competition, the success rate of each mechanism decreases, which corresponds to the intuition that with higher resource competition, it is more difficult to acquire resources.

*Observation 2*: Figure 4 shows how the social welfare changes with the average number of resources acquired by buyers, which is $\sum_{r\in\mathcal{R}_\mathbf{b}} q(\mathcal{R}_\mathbf{b}, r)$. We can observe that the advantage of *NG* over other mechanisms increases with the number of resources to acquire. Fig. 5 shows that the success rate decreases with the number of resources to acquire, which is intuitive since it is difficult to acquire more resources which have to be provided during the same period.

*Observation 3*: In some cases, the difference between a deadline and the earliest start time is large and each buyer has more flexility of deciding when to start its task. A buyer $\mathbf{b}$ can use the time between $est(\mathbf{b})$ and $dl(\mathbf{b})$ to negotiate for resources. As shown in Figure 7, the success rate of *NG* increases when buyers have more flexibility to decide when to start task execution. However, an agreement's probability of being decommited increases with more flexibility. Accordingly, a buyer may fail to get resources due to the decommitment. Figure 6 shows that, with the increase of the flexibility, the advantage of *NG* over the other mechanisms increases at the beginning and slightly decreases when buyers have a lot of flexibility to decide when to start task execution, which is mainly due to sellers' decommitment.

*Observation 4*: A buyer $\mathbf{b}$ can start negotiation at time $tg(\mathbf{b})$ and its task cannot start before $est(\mathbf{b})$. Figure 9 shows that *NG*'s success rate increases with $(est(\mathbf{b}) - tg(\mathbf{b}))/pd(\mathbf{b})$ since a buyer has more time to negotiate for resources. However, as shown in Figure 8, the advantage of *NG* does not strictly increase with negotiation time: its advantage decreases when buyers have a long negotiation time. The reason is that a buyer's agreements made at an early stage may be decommited by sellers when there is a long negotiation deadline.

*Observation 5*: In addition to a fully distributed auction (*CRA*), we also designed a super buyer which receives requests from buyers and buys resources for buyers. The super buyer runs the auction when it has received a certain number of requests or one requesting buyer's deadline is approaching, whichever occurs first. Experimental results show that *NG* still beat the centralized *CRA* by $11\%$. The centralized *CRA* beat the distributed *CRA* by no more than $2\%$.

### *6.3.2 Evaluating agents' negotiation strategies*

*Observation 6*: Since it is impossible to find out agents' equilibrium strategies in the complex bargaining game, we designed strategies for agents by taking into account some important factors which are considered in the literature. While negotiation agents with the strategies achieved higher social welfare than other mechanisms, one may ask whether agents have an incentive to switch to other strategies. To answer this question, we tried some other strategies as follows: 1) each buyer makes only one set of agreements, 2) each buyer makes three sets of agreements, 3) when an agent decide to decommit from an agreement $o$, it decommits before $est(o)$ rather than decommits immediately, 4) and a seller makes contracts beyond its capability.

We found that making these changes did not improve either utilities of agents with new strategies or *NG*'s performance. Always making two sets of contracts is a good choice due to the tradeoff between failing to finish the task and paying too much. While delaying decommitment does not "hurt" an agent directly, it hurts the agent "indirectly"

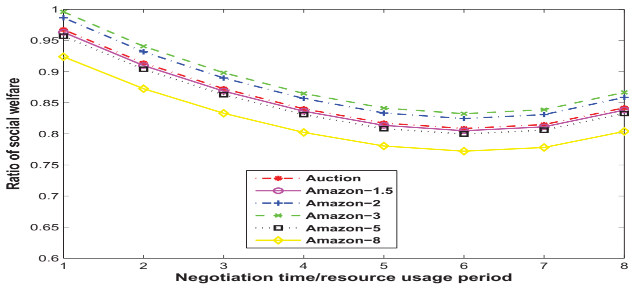

**Figure 8: Social welfare and negotiation time**

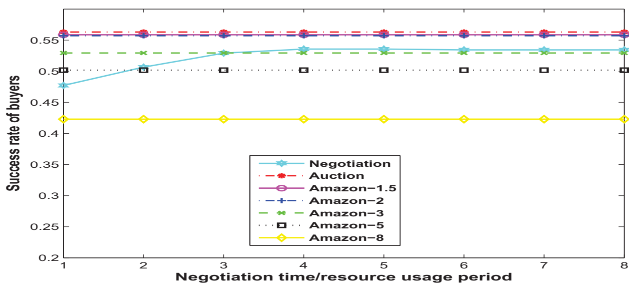

**Figure 9: Success rate and negotiation time**

since resource competition will increase if each agent holds more contracts. Further, it is better for a seller not to make contracts beyond its capability: if the resource competition is low, generally a seller cannot make contracts beyond its capability, and if the resource competition is high, a buyer is less likely to decommit from a contract and a seller may have to pay more penalties for contracts beyond its capability.

### *6.3.3 Sensitivity analysis*

We also did additional experiments to explore the sensitivity of our experimental results to changes to the parameters of our experimental environments or assumptions about our negotiation model.

*Observation 7*: This paper assumes that each agent knows the demand/supply ratio of each resource. In reality, an agent may not know the demand/supply ratio. We tested the negotiation model without this assumption and alternatively, each agent predicts the demand/supply ratio through its interaction with buyers. Specifically, a seller can estimate the competition of a resource according to 1) the requests for the resource from all the buyers in the last $\lambda$ time points and 2) the total number of resources provided by other sellers. A buyer can estimate the competition of a resource according to bids from sellers. In this case, we found that the social welfare of *NG* is still 10% higher than other mechanisms.

*Observation 8*: This paper also assumes that each agent knows each seller's cost of a resource. We found that that the accuracy of this information does have a slight effect on agents' negotiation performance. When the believed cost is less than half of the actual cost, the average social welfare of *NG* is 6% lower than that of *NG* in which each buyer knows the actual cost.

*Observation 9*: We also compared setting penalties through negotiation with exogenous mechanisms for setting penalties [5, 14], e.g., fixed penalties (e.g., $\{0, 10, 20, 40\}$) or penalty as a percentage ($e.g.$, $\{0.1, 0.3, 0.5\}$) of a contract price. We found that setting penalties through negotiation achieved higher social welfare than other exogenous mechanisms for setting penalties.

## 7. CONCLUSION

This paper presents the design and implementation of a negotiation mechanism for dynamic resource allocation problem. In the negotiation model, multiple buyers and sellers are allowed to negotiate with each other concurrently and an agent is allowed to decommimt from an agreement at the cost of paying a penalty. This paper also presents negotiation strategies for both buyers and sellers considering important factors widely studied in the literature. An extensive set of experiments were carried out and it is shown that the proposed negotiation model outperforms combinatorial auction mechanisms and Amazon's fixed price model. In general, the proposed mechanism can be applied in wide range of dynamic resource allocation problems.

Finally, we outline a future agenda for this work. First, in the current design, an agent will make its decision immediately after it receives a message. As future work, we will consider the role of delaying making decisions. Second, while it is impossible to derive agents' equilibrium strategies in such dynamic resource allocation game, it would be interesting to investigate agents' rational strategies in some simplified scenarios [4]. Third, we will implement and evaluate the negotiation mechanism using the current GENI prototype [1].

## 8. ACKNOWLEDGMENT

This work was supported primarily by the Engineering Research Centers Program of the National Science Foundation under NSF Cooperative Agreement No. EEC-0313747. Any Opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the National Science Foundation. This work was supported in part by the NSF udner grant No. CNS-0834243.

## 9. REFERENCES

[1] GENI System Overview. http://www.geni.net/.

[2] Amazon Elastic Compute Cloud. http://aws.amazon.com/ec2/.

[3] B. An, N. Gatti, and V. Lesser. Bilateral bargaining with one-sided two-type uncertainty. In *Proc. of the 9th IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pages 403–410, Sep. 2009.

[4] B. An, N. Gatti, and V. Lesser. Extending alternating-offers bargaining in one-to-many and many-to-many settings. In *Proc. of the 9th IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pages 423–426, Sep. 2009.

[5] B. An, V. Lesser, and K. M. Sim. Decommitment in multi-resource negotiation. In *Proc. of the Int. Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 1553–1556, 2008.

[6] A. A. Auyoung, B. N. Chun, A. C. Snoeren, and A. Vahdat. Resource allocation in federated distributed computing infrastructures. In *Proc. of the 1st Workshop on Operating System and Architectural Support for the Ondemand IT Infrastructure (OASIS 2004)*, 2004.

[7] P. Cramton, Y. Shoham, and R. Steinberg, editors. *Combinatorial Auctions*. MIT Press, 2006.

[8] P. Faratin, C. Sierra, and N. R. Jennings. Negotiation decision functions for autonomous agents. *Int. Journal of Robotics and Autonomous Systems*, 24(3-4):159–182, 1998.

[9] Y. Fu, J. Chase, B. Chun, S. Schwab, and A. Vahdat. Sharp: an architecture for secure resource peering. In *Proc. of the 19th ACM Symposium on Operating System Principles*, pages 133–148, 2003.

[10] N. Gatti, F. D. Giunta, and S. Marino. Alternating-offers bargaining with one-sided uncertain deadlines: an efficient algorithm. *Artificial Intelligence*, 172(8-9):1119–1157, 2008.

[11] D. Irwin, J. Chase, L. Grit, A. Yumerefendi, and D. Becker. Sharing networked resources with brokered leases. In *Proc. of the 2006 USENIX Annual Technical Conference*, pages 199–212, 2006.

[12] K. Lai, L. Rasmusson, L. Z. E. Adar, and B. Huberman. Tycoon: An implementation of a distributed, market-based resource allocation system. *Multiagent and Grid Systems*, 1(3):169–182, 2005.

[13] C. Lee and A. Snavely. Precise and realistic utility functions for user-centric performance analysis of schedulers. In *Proc. of the 16th Int. Symp. on High Performance Distributed Computing*, pages 107–116, 2007.

[14] T. D. Nguyen and N. R. Jennings. Managing commitments in multiple concurrent negotiations. *Electronic Commerce Research and Applications*, 4(4):362–376, 2005.

[15] D. Parkes. *Algorithmic Game Theory*, chapter Online Mechanisms. Cambridge University Press, 2007.

[16] A. Rubinstein. Perfect equilibrium in a bargaining model. *Econometrica*, 50(1):97–109, 1982.