*Market Oriented Grid and Utility Computing*, Edited by Rajkumar Buyya and Kris Bubendorfer.
ISBN 0-471-XXXXX-X 

# Chapter 22

# Auction-based Resource Allocation

***Björn Schnizler***

## 22.1 Introduction

Since the early 1990s, the integration of computers into parallel and distributed systems has become common practice. In this context, the Grid denotes an infrastructure in which computer resources (e.g. disk space, processors) are organized in a cohesive distributed system [1]. Within the Grid, distant computers are dynamically linked over either public or virtual private networks. The establishment of such Grids has major ramifications on the business, since organizations that have computational demand are not required to purchase and maintain computer resources on their own. Instead, it is possible that computation can be performed on demand by using resources from the Grid that are not under permanent control of the temporary user. The demand for computation is covered by resource owners that have temporarily idle computers. The suppliers of computer resources in Grids are either small-scale owners or large-scale owners who strive for an increased utilization of existing resources.

A lot of research effort has been devoted to the development of Grid middleware that provides the technical infrastructure to share resources over multiple geographic and administrative domains. However, the allocation of supplied resources to jobs is studied in less detail. Current state-of-the-art systems for resource management typically use idiosyncratic cost functions for scheduling jobs [2]. Those mechanisms are controlled centrally and only work well if information about supply and demand is truthfully reported. Since Grid addresses resource sharing not only within the borders of one organizational unit, but also cross organizational, centrally controlled mechanisms suffer untruthful revelation of job related data.

Market mechanisms are known to attain fairly efficient allocations in situations where the participating agents may conceal their private information about costs and valuations. If the market mechanism is properly defined, users may be provided incentives to express their true values for service requests and offers. This in turn marks the prerequisite for attaining an efficient allocation of services, which maximizes the sum of aggregate valuations [3].

In recent times, the idea of incorporating market mechanisms into Grid technology has increasingly gained attention. Despite this interest in market-based approaches, research regarding market mechanisms for Grid resources remains in its infancy. The canon of available market mechanisms only insufficiently copes with the requirements imposed by the Grid [3]. The contribution of this chapter is to tailor a market-based Grid exchange that can attain efficient allocations. This is achieved by a multi-attribute combinatorial exchange (MACE).

The remainder of this chapter is structured as follows: The economic and technical properties which the proposed market mechanism should satisfy are presented in Section 22.2 and reflected with regard to related work in Section 22.3. On the basis of the requirements, Section 22.4 describes the design of an auction mechanism for the Grid. In

Section 22.5, the proposed mechanism is evaluated by means of a numerical simulation. Section 22.6 concludes with a brief summary.

# 22.2 Requirements

The objective of an adequate market mechanism for the Grid is the efficient and reliable provision of resources to satisfy demand. A critical step in designing such a market is to understand the nature of the trading object [4]. This paper considers services which respect resource functionalities (e.g. storage) and quality characteristics (e.g. size), dependencies and time attributes. Relying on services instead of computational resources removes many technical problems. For instance, the resource CPU may technically not be offered without an appropriate amount of hard disk space on the same computer, while a computation service offering CPU cycles already includes the complementary resources. In the remainder of the paper, it is abstracted from those technical details by treating resources and services as synonyms.

The following subsections present the design objectives and the Grid specific requirements for the market mechanism.

## *22.2.1 Design Objectives*

The theoretical basis for designing mechanisms has emerged from a branch of game theory called mechanism design [5]. Within the scope of practical mechanism design, the primary design objective is to investigate a mechanism that has desirable properties. The following comprises common economic properties of a mechanism’s outcome:

**Allocative efficiency:** An allocation is efficient if the sum of individual utilities is maximized. A mechanism can only attain allocative efficiency if the market participants report their valuation truthfully. This requires incentive compatibility in equilibrium.

**Incentive compatibility:** A mechanism is incentive compatible if every participant's expected utility maximizing strategy in equilibrium with every other participant is to report his true preferences [6].

**Individual rationality:** The constraint of individual rationality requires that the utility following participation in the mechanism must be greater or equal to the previous utility.

**Budget balance:** A mechanism is budget balanced if the prices add up to zero for all participants [7]. In case the mechanism runs a deficit, it must be subsidized by an outside source and is therefore not feasible per-se.

**Computational tractability:** Computational tractability considers the complexity of computing a mechanism's outcome. With an increasing number of participants, the allocation problem can become very demanding and may delimit the design of choice and transfer rules [8].

For the scenario at hand, allocative efficiency meets the general design goal that the mechanism designer wants to achieve, whereas the remaining categories are constraints upon the objective.

### *22.2.2 Domain-Specific Requirements*

In addition to those mechanism properties pertaining to the outcome, the mechanism must also account for the underlying environment. The constraints of the market participants impose very rigid requirements upon the design [3]:

**Double-sided mechanism:** A Grid middleware usually provides a global directory enabling multiple service owners to publish their services and multiple service requesters to discover them. Since a market mechanism replaces these directories, it has to allow many resource owners (henceforth sellers) and resource consumers (buyers) to trade simultaneously.

**Language includes bids on attributes:** Participants in the Grid usually have different requirements for the quality characteristics of Grid services and require these in different time spans. For example, a data mining job could require a storage service with at least 250 GB of free space for 4 hours in any slot between 9 a.m. and 4 p.m. The Grid community takes these requirements into account by defining service level agreement protocols, e.g. WS-Agreement [9]. To facilitate the adherence of these agreement protocols, a market mechanism is required to support bids on multiple quality attributes of services as well as time objectives.

**Language includes combinatorial bids:** Buyers usually demand a combination of different Grid services as a bundle in order to perform a task [10]. As such, Grid services are complementarities, meaning that participants have super-additive valuations for the services, since the sum of valuations for single services is less than the valuation for the whole bundle. Suppose a buyer requires services for storage, computation and rendering. If any service is not allocated to him, the remaining bundle has no value for him. In order to avoid this exposure risk, the mechanism must allow for bids on bundles. In addition, the buyer may want to submit more than one bid on a bundle as well as many that exclude each other. In this case, the resources for the bundles are substitutes, i.e. participants have sub-additive valuations for the services. For instance, a buyer is willing to pay a high price for a service during the day and a low price if the service is executed at night. However, this service may only be computed once. To express this, the mechanism must support XOR bids to express substitutes. For simplicity, a seller's bid is restricted to a set of OR bids. This simplification can be justified by the fact that Grid services are non-storable commodities, e.g. a computation service currently available cannot be stored for a later time.

**Language includes co-allocation constraints:** Capacity-demanding jobs often require the simultaneous allocation of several homogenous service instances from different providers. For example, a large-scale simulation may require several computation services to be completed at one time. Literature often refers to the simultaneous allocation of multiple homogenous services as co-allocation. A mechanism for the Grid has to enable co-allocations and provide functionality to control it. In this context, two cases must be considered: It is desirable to limit the maximum number of service co-allocations, i.e. the maximum number of service divisions. It may be logical to couple multiple services of a bundle in order to guarantee that these resources are allocated from the same seller and will be executed on the same machine.

An adequate market mechanism for the Grid must satisfy these requirements stemming from the economic environment and ideally meet the design objectives.

# 22.3 Related Work

Waldspurger et al. [11] and Regev and Nisan [12] propose the application of Vickrey auctions for allocating homogenous computational resources in distributed systems.

Vickrey auctions achieve truthful bidding as a dominant strategy and hence result in efficient allocations.

Buyya et al. [13] were among the first researchers to motivate the transfer of market-based systems from distributed systems to Grids. Nonetheless, they propose classical one-sided auction types which cannot account for combinatorial bids. Wolski et al. [14] compare classical auctions with a bargaining market, coming to the conclusion that the bargaining market is superior to an auction-based market. This result is less surprising since the authors only consider classical auction formats where buyers cannot express bids on bundles. Eymann et al. [15] introduce a decentralized bargaining system for resource allocation in Grids. In their simulation the bargaining systems work fairly well; however, bids on bundles are largely ignored.

Subramoniam et al. [10] account for combinatorial bids by providing a tatonnement process for allocating and pricing Grid resources. Furthermore, Ng et al. [16] propose repeated combinatorial auctions as a microeconomic resource allocation in distributed systems. Nonetheless, the resources are still considered to be standardized commodities. Standardization of the resources would either imply that the number of resources is limited compared to the number of all possible resources or that there are many mechanisms which are likely to suffer due to meager participation. Both implications result in rather inefficient allocations.

Additionally, state-of-the-art mechanisms widely neglect time attributes for bundles and quality constraints for single resources. Hence, the use of these mechanisms in the Grid environment is considerably diminished. The introduction of time attributes redefines the Grid allocation problem as a scheduling problem. To account for time attributes, Wellman et al. [17] model single-sided auction protocols in order to allocate and to schedule resources with regard to different time constraint considerations. However, the proposed approach is single-sided. Installing competition on both sides is deemed superior, since no particular market side is systematically given an advantage. Demanding competition on both sides suggests the development of a combinatorial exchange. In research literature,

Parkes et al. [6] introduce the first combinatorial exchange as a single-shot sealed bid auction. As payment scheme, Vickrey discounts are approximated. The approach results in approximately efficient outcomes; however, it neither accounts for time nor for quality constraints. It is thus not directly applicable to the Grid allocation problem. Counteractively, Bapna et al. [18] propose a family of combinatorial auctions for allocating Grid services. Although the mechanism accounts for quality and time attributes and enables the simultaneous trading of multiple buyers and sellers, there is no competition on the sellers' side as all orders are aggregated to one virtual order. Moreover, the mechanism does not take co-allocation constraints into account.

In reviewing the related mechanisms according to the requirements presented in Section 22.2, it is revealed that no market mechanism installs competition on both sides, includes combinatorial bids, allows for time constraints, manages quality constraints, or considers co-allocation restrictions. This chapter intends to address these deficiencies by outlining the design of a multi-attribute combinatorial exchange for allocating and scheduling Grid services.

# 22.4 A Multi-Attribute Combinatorial Exchange

The design of MACE follows common assumptions of mechanism design and auction theory: Agents are assumed to be risk neutral, have quasi-linear utility functions as well as independent private valuations and reservation prices. The valuation functions of agents satisfy free-disposal and $v_i(\emptyset) = 0$. The valuation functions of sellers allow a linear transformation in case of partial executions. For instance, if a seller values a storage service with 300GB capacity with 10, he values a partial execution of the service with 150GB with 5. In contrast, buyers do not accept partial executions of their requests or their applications. Furthermore, it is assumed that buyers can specify their resource requirements in terms of quality characteristics and job duration. For instance, it is assumed that a buyer can specify the amount of storage space that is required for executing a job. In addition, the buyer can specify how long the job has to be executed. Likewise, a seller of resources can specify the characteristics of those resources that he can offer in the future. The elicitation of the resource characteristics can be supported by prediction models such as proposed by [19]. In addition, it is assumed that jobs can be paused and be resumed at a later time.

Resource allocations are interpreted as contracts. This means, that a seller has to provide the allocated resources. In case of failure, the seller has to offer alternative resources or compensate the buyer for the failed allocation.

As in any combinatorial auction, the design of MACE mainly affects three components: (i) the communication language which defines how bids can be formalized, (ii) the winner determination problem, and (iii) the pricing scheme to determine net payments. As such, the following description of MACE is structured as follows: First, a bidding language is introduced which supports multi-attribute combinatorial bids including co-allocation constraints. Second, a winner determination model (allocation rule) is proposed that attains an efficient allocation if agents bid truthfully. Finally, a pricing schema is outlined to incentivize agents to reveal their private information.

## *22.4.1 Bidding Language*

The design of an auction that meets the requirements requires an expressive bidding language. The following notation is used to define such a language:

Let $N$ be a set of $N$ buyers and $M$ be a set of $M$ sellers, where $n \in N$ defines an arbitrary buyer and $m \in M$ an arbitrary seller. There are $G$ discrete resources $G = \{g_1, \ldots, g_G\}$ with $g_k \in G$ and a set of $D$ bundles $S = \{S_1, \ldots, S_D\}$ with $S_j \in S$ and $S_j \subseteq G$ as a subset of resources. For instance, $S_j = \{g_k, g_l\}$ denotes that the bundle $S_j$ consists of two resources $g_k$ and $g_l$, where $g_k$ could be a computation service and $g_l$ a storage service.

A resource $g_k$ has a set of $A_k$ cardinal quality attributes $A_{g_k} = (a_1^k, \ldots, a_{A_k}^k)$ where $a_i^k \in A_{g_k}$ represents the $i.th$ attribute of the resource $g_k$. For instance, in the context of a Grid resource, a quality attribute can be the $size$ of a storage service.

A buyer $n$ can specify the minimal required quality characteristics for a bundle $S_j \in S$ with $q_n^N(S_j, g_k, a_i^k) \geq 0$, where $g_k \in S_j$ is a resource of the bundle $S_j$ and $a_i^k \in A_{g_k}$ is an attribute of the resource $g_k$. For instance, the minimal required size of a storage service $g_k \in S_j$ can be denoted by $q_n^N(S_j, g_k, a_i^k) = 200GB$. Accordingly, a seller $m$ can specify the maximum offered quality characteristics with $q_m^M(S_j, g_k, a_i^k) \geq 0$. The quality attributes are assumed to be cardinal numbers. The characteristics have to satisfy $q_n^N(\cdot) \geq \overline{q_n^N}(\cdot)$ if the first quality characteristic $q_n^N(\cdot)$ satisfies at least the second one $\overline{q_n^N}(\cdot)$. These quality characteristics are also used to specify a value for the agent's network connection. For instance, this can be used to denote the uplink and downlink rates of the given network connection.

For each resource $g_k \in S_j$, a buyer $n$ can specify the maximum number of co-allocations in each time slot with $\gamma_n(S_j, g_k) \geq 0$. This means, that a buyer $n$ can limit the number of sellers that provide the required resource $g_k$. Let $\gamma_n(S_j, g_k) = K$ if the resource $g_k$ has no divisibility restrictions, where $K$ is a large enough constant[1]. The coupling of two resources in a bundle is represented by the binary variable $\phi_n(S_j, g_k, g_l)$ where $\phi_n(S_j, g_k, g_l) = 1$ if resources $g_k$ and $g_l$ have to be allocated from the same bundle bid of a seller and $\phi_n(S_j, g_k, g_l) = 0$ otherwise. It is assumed that all resources offered in a bundle are located on the same machine.

Resources in the form of a bundle $S_j$ can be assigned to a set of maximal $T$ discrete time slots $T = (0,...,T-1)$, where $t \in T$ specifies one single time slot. A buyer $n$ can specify the minimum required number of time slots $s_n(S_j) \geq 0$ for a bundle $S_j$. The earliest time slot for any allocatable bundle $S_j$ can be specified by $e_n^N(S_j) \geq 0$ for a buyer $n$ and $e_m^M(S_j) \geq 0$ for a seller $m$; the latest possible allocatable time slot by $l_n^N(S_j) \geq 0$ for a buyer $n$ and by $l_m^M(S_j) \geq 0$ for a seller $m$.

---

[1]The constant $K$ has to be greater than the total number of seller bids.

A buyer $n$ can express the valuation for a single slot of a bundle $S_j$ by $v_n(S_j) \geq 0$, whereat $v_n(S_j)$ denotes the maximum price for which the buyer $n$ is willing to buy. The reservation price for allocating a single slot of a bundle $S_j$ is denoted by $r_m(S_j) \geq 0$. This price represents the minimum price for which the seller $m$ is willing to sell.

On the basis of these parameters, an atomic bid of a buyer is defined as follows:

**Definition 22.1 [MACE Atomic Buyer Bid]:** In MACE, an atomic bid $B_n$ of a buyer $n$ is defined as

$$\begin{aligned}B_n(S_j) = (&v_n(S_j), s_n(S_j), e_n^N(S_j), l_n^N(S_j),\\ &(q_n^N(S_j, \overline{g}_1, a_1^1), ..., q_n^N(S_j, \overline{g}_l, a_{A_{\overline{g}_l}}^l)), (\gamma_n(S_j, \overline{g}_1), ..., \gamma_n(S_j, \overline{g}_l)),\\ &(\phi_n(S_j, \overline{g}_1, \overline{g}_2), \phi_n(S_j, \overline{g}_1, \overline{g}_3), ..., \phi_n(S_j, \overline{g}_1, \overline{g}_l), ..., \phi_N agent(S_j, \overline{g}_{l-1}, \overline{g}_l))),\end{aligned}$$

where $G_{S_j} = \{\overline{g}_1, ..., \overline{g}_l\}$ are the resources of the bundle $S_j$.

It is to note that the atomic bid can also be represented in a more compact way. For instance, the encoding of the coupling conditions $\phi_n(\cdot)$ can be restricted to cases with $\phi_n(\cdot) = 1$. For a better readability, however, the atomic bid is formalized in this detailed way.

In order to allow buyers to express substitutes over a set of resources, MACE supports the submission of XOR concatenated atomic bids.

**Definition 22.2 [MACE XOR Buyer Bid]:** A XOR bid of a buyer $n$ is defined as

$$B_n = (B_n(S_j) \oplus ... \oplus B_n(S_k)).$$

The total number of atomic bids that are concatenated by the XOR operator can be restricted by the auctioneer.

The sellers' bids are formalized in a similar way to those of the buyers. However, they do not include maximum divisibility and coupling properties and assume that the number of time slots is equal to the given time range. An atomic bid for a seller is defined as follows:

**Definition 22.3 [MACE Atomic Seller Bid]:** An atomic bid $B_m$ for a seller $m$ is defined as

$$B_m(S_j) = (r_m(S_j), e_m^M(S_j), l_m^M(S_j), q_m^M(S_j, \bar{g}_1, ute_1^1), \ldots, (q_m^M(S_j, \bar{g}_l, a_{A_{\bar{g}_l}}^l))),$$

where $G_{S_j} = \{\bar{g}_1, \ldots, \bar{g}_l\}$ are the resources that are part of the bundle $S_j$.

For sellers as resource providers, a XOR operator is not necessary. Grid resources are non-storable commodities. For instance, a computation service currently available cannot be stored and used at a later time. As such, the bidding space for sellers is restricted to OR bids.

In the following subsections, it is assumed that the bid elicitation has already taken place. This means, buyers and sellers submitted their preferences by means of the bidding language to the auctioneer. For formulating bids, agents may use preference elicitation techniques to formalize their preferences [20] or may use an autonomous bidding agent that takes over their bidding strategies.

## *22.4.2 Winner Determination*

Based upon this bidding language, the winner determination problem of MACE (MACE allocation problem, MAP) can be formulated. Following the previous winner determination models, MAP is formulated as a linear mixed integer program.

For formalizing the model, the decision variables $x_n(S_j)$, $z_{n,t}(S_j)$, $y_{m,n,t}(S_j)$, and $d_{m,n,t}(S_j)$ have to be introduced. The binary variable $x_n(S_j) \in \{0,1\}$ denotes whether bundle $S_j$ is allocated to buyer $n$ $(x_n(S_j)=1)$ or not $(x_n(S_j)=0)$. Furthermore, the binary variable $z_{n,t}(S_j) \in \{0,1\}$ is assigned to a buyer $n$ and is associated in the same way as $x_n(S_j)$ with the allocation of $S_j$ in time slot $t$. For a seller $m$, the real-valued variable $y_{m,n,t}(S_j)$ with $0 \le y_{m,n,t}(S_j) \le 1$ indicates the percentage contingent of bundle $S_j$ allocated to the buyer $n$ in time slot $t$. For example, $y_{m,n,t}(S_j)=0.5$ denotes that 50 percent of the quality characteristics of bundle $S_j$ are allocated from seller $m$ to buyer $n$ in time slot $t$. Suppose a seller is offering a storage service $S_2=\{g_2\}$ with 30 GB of free space. A partial allocation of 15 GB from seller $m$ to buyer $n$ in time slot $t$ would lead to $y_{m,n,t}(S_2)=0.5$. The binary variable $d_{m,n,t}(S_j) \in \{0,1\}$ is linked with $y_{m,n,t}(S_j)$ and denotes whether the seller $m$ allocates bundle $S_j$ to buyer $n$ in time slot $t$ $(d_{m,n,t}(S_j)=1)$ or not $(d_{m,n,t}(S_j)=0)$.

By means of these variables, MAP is formulated as follows [21]:

$$\max \sum_{n\in N}\sum_{S_j\in S}\sum_{t\in T} v_n(S_j) z_{n,t}(S_j) - \sum_{m\in M}\sum_{n\in N}\sum_{S_j\in S}\sum_{t\in T} r_m(S_j) y_{m,n,t}(S_j) \quad (22.1)$$

$$\text{s.t.} \sum_{S_j\in S} x_n(S_j) \le 1, \forall n \in N \quad (22.2)$$

$$\sum_{t\in T} z_{n,t}(S_j) - x_n(S_j) s_n(S_j) = 0, \forall n \in N, \forall S_j \in S \quad (22.3)$$

$$\sum_{n\in N} y_{m,n,t}(S_j) \le 1, \forall m \in M, \forall S_j \in S, \forall t \in T \quad (22.4)$$

The objective function 1 maximizes the surplus $V^*$, which is defined as the difference between the sum of the buyers' valuations $v_n(S_j)$ and the sum of the sellers' reservation prices $r_m(S_j)$. Assuming bidders are truthful, the objective function reflects the goal of maximizing social welfare. The first Constraint 22.2 guarantees that each buyer $n$ can be allocated to one only bundle $S_j$. This constraint is necessary to fulfill the XOR constraint of a buyer bid. Constraint 22.3 ensures that for any allocated bundle $S_j$, a buyer $n$ receives exactly the required slots within the time set $T$. For each time slot $t$, Constraint 22.4 ensures that each seller cannot allocate more than the seller possesses. The formulation of this constraint implicates that a seller cannot fully allocate two resources to two different buyers in time slot $t$. For instance, suppose a seller offers the bundle $S_j = \{g_k, g_l\}$. An allocation of the resource $g_k$ to buyer 1 (with $y_{m,1,t}(S_j) = 1$) and an allocation of $g_l$ to buyer 2 (with $y_{m,2,t}(S_j) = 1$) is not possible. This restriction is applied to simplify the model. However, the above mentioned allocation can be attained by submitting an OR concatenated bid on the bundles $S_n = \{g_k\}$ and $S_i = \{g_l\}$.

The constraints 22.2 – 22.4 consider the basic allocation functionality of the exchange. In designing an adequate mechanism for the Grid, quality characteristics and dependencies between resources must also to be addressed:

$$\sum_{S_j \ni g_k} z_{n,t}(S_j) q_n^N(S_j, g_k, a_i^k) - \sum_{S_j \ni g_k} \sum_{m \in M} y_{m,n,t}(S_j) q_m^M(S_j, g_k, a_i^k) \le 0,$$
$$\forall n \in N, \forall g_k \in G, \forall a_i^k \in A_{g_k}, \forall t \in T \tag{22.5}$$

$$\sum_{S_j \ni g_k} \sum_{m \in M} d_{m,n,t}(S_j) - \sum_{S_j \ni g_k} \gamma_n(S_j, g_k) z_{n,t}(S_j) \le 0,$$
$$\forall n \in N, \forall g_k \in G, \forall t \in T \tag{22.6}$$

$$\sum_{S_j \ni g_k, g_l} \phi_n(S_j, g_k, g_l)\Big(\sum_{S_j \ni g_k} d_{m,n,t}(S_j) - \sum_{S_j \ni g_l} d_{m,n,t}(S_j)\Big) = 0,$$
$$\forall n \in N, \forall m \in M, \forall g_k, g_l \in G, \forall t \in T \tag{22.7}$$

$$\sum_{S_j \ni g_k, g_l} \phi_n(S_j, g_k, g_l)\Big(\sum_{S_j \ni g_k} \sum_{m \in M} d_{m,n,t}(S_j) + \sum_{S_j \ni g_l} \sum_{m \in M} d_{m,n,t}(S_j) - 2 z_{n,t}(S_j)\Big)$$
$$\le 0, \forall n \in N, \forall g_k, g_l \in G, \forall t \in T \tag{22.8}$$

Constraint 22.5 guarantees that for any allocated bundle in an arbitrary time slot $t$, all required resources have to be fulfilled in the same slot in at least the demanded qualities. Constraint 22.6 ensures that a resource will be provided by at most $\gamma_n(S_j, g_k)$ different suppliers. For simplicity, it is assumed that a resource $g_k$ with restricted co-allocations is not part of further XOR concatenated bids of the buyer $n$. Furthermore, resources with co-allocations cannot be allocated as free-disposal items. As an example, suppose a buyer $n$ values $S_j = \{g_k\}$ with $v_n(S_j) = 1$ and $S_i = \{g_l\}$ with $v_n(S_i) = 10$. For bundle $S_j$, the buyer has co-allocation restrictions with $\gamma_n(S_j, g_l) = 1$. A seller $m$ that offers $S_t = \{g_k, g_l\}$ cannot allocate the resource $g_l$ to buyer $n$ as this would imply a free-disposal allocation of the restricted resource $g_k$.

Constraint 22.7 and 22.8 are both responsible for the coupling of two resources. Constraint 22.7 ensures that two resources must be provided by the same seller, in case they should be coupled. This constraint alone does not suffice the coupling requirements since it would be possible for two sellers to co-allocate a coupled computation service with $3000$ MIPS and a storage service with $30$ GB in different quality characteristics. For instance, MAP could allocate a computation service with $2998$ MIPS and a storage service with $1$ GB from one seller, and a computation service with $2$ MIPS and a storage service with $29$ GB from another. To exclude these undesirable allocations, Constraint 22.8 imposes the restriction that coupled resources cannot be co-allocated. Simplifying the model, this also includes free-disposal resources. For instance, if the computation service with $3000$ MIPS and the storage service with $30$ GB are allocated from one particular seller as a bundle, another seller cannot allocate a bundle containing a rendering service and another storage service to the same buyer. However, the seller may allocate any bundle without a storage and computation service to the buyer, e.g., the rendering service alone. Furthermore, it is assumed that coupled resources are only part of one particular atomic bid $B_n(S_j)$ in case a buyer submits two XOR concatenated bids containing coupled resources.

The time restrictions of the bids are given by:

$$(e_n^N(S_j)-t)z_{n,t}(S_j) \le 0, \forall n \in N, \forall S_j \in S, \forall t \in T \quad (22.9)$$

$$(t-l_n^N(S_j))z_{n,t}(S_j) \le 0, \forall n \in N, \forall S_j \in S, \forall t \in T \quad (22.10)$$

$$(e_m^N(S_j)-t)\sum_{n \in N} y_{m,n,t}(S_j) \le 0, \forall m \in M, \forall S_j \in S, \forall t \in T \quad (22.11)$$

$$(t-l_m^M(S_j))\sum_{n \in N} y_{m,n,t}(S_j) \le 0, \forall m \in M, \forall S_j \in S, \forall t \in T \quad (22.12)$$

Essentially, constraints 22.9 – 22.12 indicate that slots cannot be allocated before the earliest and after the latest time slot of either a buyer (Constraint 22.9 and 22.10) or a seller (Constraint 22.11 and 22.13).

Finally, the establishment of the relationship between the real valued decision variable $y_{m,n,t}(S_j)$ and the binary variable $d_{m,n,t}(S_j)$ needs to be addressed and the decision variables of the optimization problem have to be defined:

$$y_{m,n,t}(S_j)-d_{m,n,t}(S_j) \le 0, \forall n \in N, \forall m \in M, \forall S_j \in S, \forall t \in T \quad (22.13)$$

$$d_{m,n,t}(S_j)-y_{m,n,t}(S_j) < 1, \forall n \in N, \forall m \in M, \forall S_j \in S, \forall t \in T \quad (22.14)$$

$$x_n(S_j) \in \{0,1\}, \forall n \in N, \forall S_j \in S \quad (22.15)$$

$$z_{n,t}(S_j) \in \{0,1\}, \forall n \in N, \forall S_j \in S, \forall t \in T \quad (22.16)$$

$$y_{m,n,t}(S_j) \ge 0, \forall n \in N, \forall m \in M, \forall S_j \in S, \forall t \in T \quad (22.17)$$

$$d_{m,n,t}(S_j) \in \{0,1\}, \forall n \in N, \forall m \in M, \forall S_j \in S, \forall t \in T \quad (22.18)$$

Constraints 22.13 and 22.14 incorporate an `if-then-else` constraint. If a seller $m$ partially allocates a bundle $S_j$ to a single buyer $n$ ($y_{m,n,t}(S_j) > 0$), the binary variable $d_{m,n,t}(S_j)$ has to be $d_{m,n,t}(S_j) = 1$ (Constraint 13); otherwise, it has to be $d_{m,n,t}(S_j) = 0$

(Constraint 22.14). Finally, the constraints 22.15 – 22.18 specify the decision variables of the optimization problem.

As multiple solutions may exist that maximize the objective function, ties are broken in favor of maximizing the number of traded bundles and then at random. A special case of tie breaking occurs if the total surplus is zero. This can be the case if buyers and sellers balance their payments or no possible trade can be matched. In such a scenario, the allocation with the balanced traders is selected.

Following related work on combinatorial auctions and exchanges [22], the presented winner determination problem is also $NP$-complete.

**Theorem 22.1 [MAP Complexity]:** The MACE allocation problem (MAP) is $NP$-complete.

*Sketch 22.1.* The combinatorial allocation problem (CAP) can be reduced to MAP. Obviously, any CAP instance (multiple buyers, one seller with a zero reservation price, no attributes and no coupling constraints) can be solved by MAP. CAP is known to be $NP$-complete [22]. As such, MAP is also $NP$-complete. □

### *22.4.3 Pricing*

The outcome of MAP is allocative efficient as long as buyers and sellers reveal their valuations truthfully. The incentive to set bids according to the valuation is induced by an adequate pricing mechanism.

The implementation of an adequate price mechanism for an exchange is a challenging problem. The VCG schema cannot be applied as it runs a deficit and requires outside subsidiary [23]. On the other hand, alternative pricing schemas such as the approximated VCG mechanism are budget-balanced and approximately efficient [6]. However, the pricing scheme still requires $I+1$ instances of MAP to be solved if $I$ agents are part of the allocation. As a consequence, an alternative pricing scheme is designed that is computationally more efficient and still attains desirable economic properties.

The underlying idea of the $k$-pricing scheme is to determine prices for a buyer and a seller on the basis of the difference between their bids [24]. For instance, suppose that a buyer $n$ wants to purchase a storage service for $v_n(\cdot)=5$ and a seller $m$ wants to sell a storage service for at least $r_m(\cdot)=4$. The difference between these bids is $\beta=1$, where $\beta$ is the surplus of this transaction that can be distributed among the participants.

For a single commodity exchange, the $k$-pricing scheme can be formalized as follows: let $v_n(S_j)=a$ be the valuation of a buyer $n$ and $r_m(S_j)=b$ be the reservation price of the buyer's counterpart $m$. It is assumed that $a \geq b$, which implicates that the buyer has a valuation for the commodity that is at least as high as the seller's reservation price. Otherwise, no trade would occur. The price for a buyer $n$ and a seller $m$ can be calculated by $p(S_j)=ka+(1-k)b$ with $0 \leq k \leq 1$.

The $k$-pricing schema can also be applied to a multi-attribute combinatorial exchange: In each time slot $t$ in which a bundle $S_j$ is allocated from one or more sellers, the surplus generated by this allocation is distributed among a buyer and the sellers. Suppose a buyer $n$ receives a computation service $S_1=\{g_1\}$ with 1000 MIPS in time slot $4$ and values this slot with $v_n(S_1)=5$. The buyer obtains the computation service $S_1=\{g_1\}$ by a co-allocation from seller 1 (400 MIPS) with a reservation price of $r_1(S_1)=1$ and from seller $2$ (600 MIPS) with $r_2(S_1)=2$. The distributable surplus of this allocation is

$\beta_{n,4}(S_1) = 5 - (1+2) = 2$. Buyer $n$ gets $k\beta_{n,4}(S_1)$ of this surplus, i.e. the price buyer $n$ has to pay for this slot $t = 4$ is

$$p^N_{k,n,4}(S_j) = v(S_1) - k\beta_{n,4}(S_1).$$

Furthermore, the sellers have to divide the other part of this surplus, i.e. $(1-k)\beta_{n,4}(S_1)$. This will be done by considering each proportion a seller's bid has on the surplus. In the example, this proportion $0 \le o_{m,n,t}(S_j) \le 1$ for seller 1 is $o_{1,n,4}(S_1) = \frac{1}{3}$ and for seller 2 is $o_{2,n,4}(S_1) = \frac{2}{3}$. The price a seller $m$ receives for a single slot $t = 4$ is consequently calculated as

$$p^M_{k,n,4}(S_j) = r_m(S_1) + (1-k)\beta_{n,4}(S_1)o_{m,n,4}(S_1).$$

Expanding this scheme to a set of time slots, co-allocations, and the allocation of different bundles to a buyer results in the following formalization: let $\beta_{n,t}(S_j)$ be the surplus for a bundle $S_j$ of a buyer $n$ with all corresponding sellers for a time slot $t$:

$$\beta_{n,t}(S_j) = z_{n,t}(S_j)v_n(S_j) - \sum_{m \in M}\sum_{S_l \in S} y_{m,n,t}(S_l)r_m(S_l)$$

The iteration over $\sum_{S_l \in S} y_{m,n,t}(S_l)r_m(S_l)$ is required, as one seller may allocate a subset $S_l$ of the required bundle $S_j$ to a buyer. For instance, this is the case if a buyer requires $S_3 = \{g_1, g_2\}$ and two sellers allocate $S_1 = \{g_1\}$ and $S_2 = \{g_2\}$.

For the entire job (i.e. all time slots), the price for a buyer $n$ is calculated as

$$p^N_{k,n}(S_j) = x_n(S_j)v_n(S_j)s_n(S_j) - k\sum_{t \in T}\beta_{n,t}(S_j).$$

This means, that the difference between the valuation for all slots $v_n(S_j)s_n(S_j)$ of the bundle $S_j$ and the $k$-th proportion of the sum over all time slots of the corresponding surpluses is determined.

The price of a seller $m$ is calculated in a similar way: First of all, the proportion $o_{m,n,t}(S_j)$ of a seller $m$ allocating a bundle $S_j$ to the buyer $n$ in time slot $t$ is given by

$$o_{m,n,t}(S_j) = \begin{cases} y_{m,n,t}(S_j) r_m(S_j) / m \in M \sum S_l \in S \sum y_{m,n,t}(S_l) r_m(S_l) & \text{if } y_{m,n,t}(S_j) r_m(S_j) > 0 \\ 0 & \text{otherwise.} \end{cases}$$

The formula computes the proportion of a seller's allocation compared to all other allocations made by any seller to the particular buyer $n$. In case a buyer is allocated a bundle $S_j$, it is ensured that it is not allocated any other bundle (XOR constraint). As a consequence, any allocation of a seller to buyer $n$ correlates with this bundle $S_j$.

Having computed $\beta_{n,t}(S_j)$ and $o_{m,n,t}(S_j)$, the price a seller receives for a bundle $S_j$ is calculated as:

$$p_{k,m}^{M}(S_j) = \sum_{n \in N} \sum_{t \in T} y_{m,n,t}(S_j) r_m(S_j) + (1-k) \sum_{n \in N} \sum_{S_l \in S} \sum_{t \in T} o_{m,n,t}(S_j) \beta_{n,t}(S_l).$$

Using the $k$-pricing schema, the exchange does not have to subsidize the participants, since it fulfills the budget-balance property in a way that no payments towards the mechanism are necessary.

**Theorem 22.2 [Budget-Balance and Individual Rationality]:** MACE is budget-balanced and individually rational [3].

Following the Myerson-Satterthwaite theorem [23], it is obvious that MACE cannot be incentive compatible. In order to evaluate these implications of the pricing schema in different settings, further analyses need to be investigated.

# 22.5 Evaluation

The application of the $k$-pricing schema implicates that agents can gain a higher utility by misrepresenting their private information. This raises the question if this utility gain can be measured and if it can serve as a metric for the loss of incentive compatibility: Let $\tilde{I}$ be a set of agents that can manipulate their valuations and reservation prices. In a benchmark scenario with an outcome $o$, all agents $\tilde{i} \in \tilde{I}$ honestly reveal their preferences. Consequently, their utility $u_{\tilde{i}}(o)$ from bidding truthfully can be calculated as $\sum_{\tilde{i} \in \tilde{I}} u_{\tilde{i}}(o)$. In a second setting with an outcome $\overline{o}$, agents $\tilde{i} \in \tilde{I}$ manipulate their bids, whereas the input parameters (i.e., the characteristics of the underlying bids) remain the same. The resulting utility due to manipulation is calculated as $\sum_{\tilde{i} \in \tilde{I}} u_{\tilde{i}}(\overline{o})$. Thus, the utility gained due to manipulation can be measured as

$$UG^{O}_{n,k}(S_j) = \sum_{\tilde{i} \in \tilde{I}} u_{\tilde{i}}(\overline{o}) - \sum_{\tilde{i} \in \tilde{I}} u_{\tilde{i}}(o), \qquad (22.19)$$

where $k$ stands for the $k$-pricing schema and $O$ stands for an optimal winner determination algorithm. The metric reflects the difference between the utility gained by manipulation and the utility gained in a truthful scenario. If this value is positive, agents have an incentive to manipulate their bids. In case the value is negative, agents do worse by manipulating.

## *22.5.1 Data Basis*

As a data basis, a random bid stream including Decay distributed bundles is generated. The Decay function has been recommended by Sandholm [8] because it creates hard instances of the allocation problem. At the beginning, a bundle consists of one random resource. Afterwards, new resources are added randomly with a probability of $\alpha = 0.75$. This procedure is iterated until resources are no longer added or the bundle already includes the same resource. The effects that can be obtained by the Decay distribution will be amplified. Hence, the Decay function is used to create a benchmark for upper bounds of the effects.

As an order, buyers and sellers submit an atomic bid, where a bundle is Decay-distributed from 5 possible resources. Each resource has two different attributes drawn from a uniform distribution within a range of $[1..2000]$. The time attributes are each uniformly distributed where the earliest and latest time slots each have a range of $[0..4]$ and the number of slots lies in $[1..3]$. For simplicity, neither co-allocation restrictions nor co-allocations constraints are taken into account. The corresponding valuations and reservation prices for a bundle are drawn from the same uniform distribution and multiplied by the number of resources in a bundle. In any problem instance, new orders for buyers and sellers are randomly generated. Subsequently, demand and supply are matched against each other, determining the winning allocation and corresponding prices.

### *22.5.2 Results*

Following Equation 22.19, the measured metric reflects the difference between the utility gained by manipulation and the utility gained in a truthful scenario. Consequently, the following results reflect absolute values. In case a manipulating agent $\tilde{i}$ is neither part of the allocation in the truthful scenario $o$ nor in the manipulating scenario $\overline{o}$, the resulting utilities ($u_{\tilde{i}}(o) = u_{\tilde{i}}(\overline{o}) = 0$) are neglected.

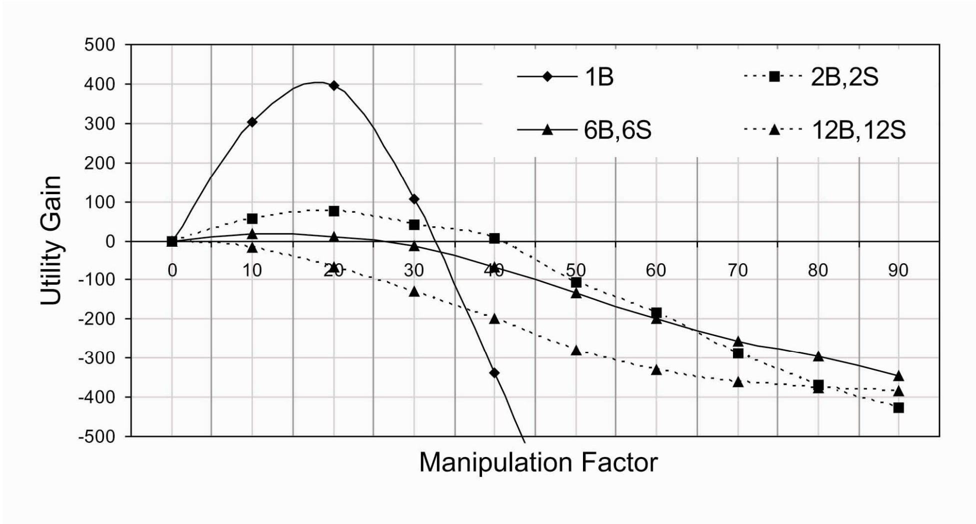


Figure 22.2: Utility gain of manipulating agents with the application of the $k$-pricing schema

Figure 22.2 depicts the utility gain of agents as a function that depends on the manipulation factor $\beta\%$. The input data is generated using the baseline setting $I_1$ for domain independent bids. The graph points out that agent can increase their utility by manipulation. For instance, if one agent underbids his valuation by $\lambda = 20\%$, his average utility gain is $UG_k^O = 394.15$. However, if the agent manipulates by more than $\lambda = 20\%$, his average utility gain continuously decreases. This is reasoned by the fact that he increases the risk of not getting allocated in the final outcome. In settings with a manipulation factor greater than $\lambda = 35\%$, he has a negative utility gain. Consequently, he has no incentive to underbid his valuation by more than $\lambda = 35\%$. Utility losses greater than 500 ($UG_k^O \leq -500$) are truncated in the graph.

If more agents deviate from bidding truthfully, the average utility gain of each agent decreases. In settings where half of the agents manipulate their bids by more than $\lambda = 30\%$, no agent has a positive utility gain on average. Moreover, if all agents manipulate their bids, none of them can attain a positive utility. This can be explained by the fact that the total number of potential counterparts decreases as the price span between buyers and sellers increases.

The results show that agents can gain a positive utility by manipulating their bids. The gains, however, are restricted to settings in which only few agents manipulate their bids by a low factor. Although a single agent can gain a positive utility by not revealing his true preferences, the gain decreases if more agents start to manipulate their bids. If only one agent manipulates by a low factor, the overall efficiency losses are small. The simulation results suggest that the $k$-pricing schema has accurate incentive properties resulting in fairly mild allocative efficiency losses. As a consequence, the $k$ – pricing schema can be a practical alternative to the VCG mechanism for combinatorial auction mechanisms.

## 22.6 Conclusion

This paper proposed the derivation of a multi-attribute combinatorial exchange. In contrast to other combinatorial approaches, the proposed mechanism also accounts accounts for time and quality attributes as well as allocation restriction. The mechanism provides buyers and sellers with a rich bidding language, the formulation of a winner determination model that can attain efficient allocations, and the derivation of the $k$-pricing schema for combinatorial mechanisms.

As the simulation illustrates, the applied pricing rule does not rigorously punish inaccurate valuation and reserve price reporting. Agents sometimes increase their individual utility by cheating. This possibility, however, is only limited to mild misreporting and a small number of strategic buyers and sellers. If the number of misreporting participants increases, the risk of not being executed in the auction rises dramatically. As a result, the $k$-pricing schema is a practical alternative to the VCG mechanism.

Aside from topics related to the computational issues of the mechanism, future research needs to extend the expressiveness of the current bidding language. For instance, MACE only supports the specification of cardinal resource attributes. Although this is sufficient for most practical cases, there may be settings in which nominal attributes are also required. In addition, the bidding language is based on pairs of attributes and values that syntactically describe resources and their quality attributes. Consequently, demand and supply is matched on the basis of attribute-based matching functions. This may be insufficient if an agent is not only interested in one particular resource configuration but is also willing to accept similar ones. To remedy this drawback, the use of ontology based bidding languages has to be considered in the future [25].

# References

1. I. Foster and C. Kesselman, *The Grid 2 - Blueprint for a New Computing Infrastructur*, 2nd Edition, Elsevier, 2004.
2. R. Buyya, *Economic-based Distributed Resource Management and Scheduling for Grid Computing*, PhD thesis, Monash University, Melbourne, Australia, 2002.
3. B. Schnizler, *Resource Allocation in the Grid - A Market Engineering Approach*, Studies on eOrganisation and Market Engineering, Universitätsverlag Karlsruhe, 2007.
4. C. Weinhardt, D. Neumann, and C. Holtmann, Computer-aided market engineering. *Communications of the ACM* 49(7), pp. 79-79, 2006.
5. P. Milgrom, *Putting Auction Theory to Work*, Cambridge University Press, 2004.
6. D. Parkes, J. Kalagnanam, and M. Eso, Achieving budget-balance with Vickrey-based payment schemes in exchanges, in *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pp. 1161-1168, 2001.
7. M.O. Jackson, Mechanism theory, *Encyclopedia of Life Support Systems*, Oxford, UK, Eolss Publishers, 2002.

8. T. Sandholm, Algorithm for optimal winner determination in combinatorial auctions, *Artificial Intelligence*, 135(1-2), pp. 1-54, 2002.
9. H. Ludwig, A. Dan, and B. Kearney, Cremona: An architecture and library for creation and monitoring of WS-Agreements, in *Proceedings of the 2nd International Conference on Service Oriented Computing*, pp. 65-74, 2004.
10. K. Subramoniam, M. Maheswaran, and M.Toulouse, Towards a micro-economic model for resource allocation in Grid computing systems, in *Proceedings of the 2002 IEEE Canadian Conference on Electrical & Computer Engineering*, 2002.
11. C. Waldspurger, T. Hogg, B. Huberman, J. Kephart, and W. Stornetta, Spawn: A distributed computational economy, *IEEE Transactions on Software Engineering* 18(2), pp. 103-117, 1992.
12. O. Regev and N. Nisan, The POPCORN market: Online markets for computational Resources, *Decision Support Systems*, 28(1-2), pp. 177-189, 2000.
13. R. Buyya, D. Abramson, J. Giddy, and H. Stockinger, Economic models for resource management and scheduling in Grid computing, *Concurrency and Computation: Practice and Experience,* 14(13-15), pp. 507-1542, 2002.
14. R. Wolski, J. Plank, J. Brevik, and T. Bryan, Analyzing market-based resource allocation strategies for the computational Grid, *International Journal of High Performance Computing Applications,* 15(3), pp. 258-281, 2001.
15. T. Eymann, M. Reinicke, O. Ardaiz, P. Artigas, L.D. de Cerio, F. Freitag, R. Messeguer, and L. Navarro, Decentralized vs. centralized economic coordination of resource allocation in grids, in *Proceedings of the 1st European Across Grids Conference*, 2003.
16. C. Ng, P. Buonadonna, N.B. Chun, A.C. Snoeren, and A. Vahdat, Addressing strategic behavior in a deployed microeconomic resource allocator, in *Proceedings of the 3rd Workshop on Economics of Peer-to-Peer Systems*, 2005.
17. M.P. Wellman, W.E. Walsh, P.R. Wurman, and J.K. MacKie-Mason, Auction protocols for decentralized scheduling, *Games and Economic Behavior*, 35, pp. 271-303, 2001.
18. R. Bapna, S. Das, R. Garfinkel, and J. Stallaert, A market design for Grid computing, *INFORMS Journal of Computing*, 20(1), pp. 100-111, 2008.
19. W. Smith, Improving resource selection and scheduling using predictions, *Grid Resource Management – State of the Art and Future Trends*, J. Nabrzyski, J.M. Schopf, and J. Weglarz, J.,(eds), Kluwer Academic Publishers, 2004.
20. W. Conen and T. Sandholm, Preference elicitation in combinatorial auctions, in *Proceedings of the Third ACM Conference on Electronic Commerce*, New York, ACM Press, pp. 256-259, 2001.
21. B. Schnizler, D. Neumann, D. Veit, and C. Weinhardt, Trading Grid services - a multi-attribute combinatorial approach, *European Journal of Operational Research*, 187 (3), pp. 943-961, 2008.
22. M. Rothkopf, A. Pekec, and R. Harstad, Computationally manageable combinational auctions, *Management Science*, 44, pp. 1131-1147, 1998.
23. R.B. Myerson and M.A., Satterthwaite, Efficient mechanisms for bilateral trading, *Journal of Economic Theory*, 28, pp. 265-281, 1983.

24. M.A. Sattherthwaite and S.R. Williams, The Bayesian theory of the k-double auction, *The Double Auction Market - Institutions, Theories, and Evidence*, D. Friedman and J. Rust (eds.), Addison-Wesley, pp. 99-123, 1993.
25. S. Lamparter and B. Schnizler, Trading services in ontology-driven markets, in *Proceedings of the 2006 ACM Symposium on Applied Computing*, ACM Press, pp. 1679-1683, 2006.