Market-based Approaches to Optimization

Maria Karlsson[†], Fredrik Ygge[‡], and Arne Andersson[‡]

[†]Uppsala University and Växjö University
School of Mathematics and Systems Engineering
Växjö University
SE - 351 95 Växjö, Sweden
www.msi.vxu.se/~mka
Maria.Karlsson@msi.vxu.se

[‡]Trade Extensions and Uppsala University
Computing Science Department
Information Technology
SE - 751 05 Uppsala, Sweden
www.user.it.uu.se/~{ygge,arnea}
{ygge,arnea}@it.uu.se

Abstract. We present a general discussion of what constitutes a market-oriented approach to optimization. We demonstrate how a general framework can be used to conceptually improve two well-known approaches from the literature, and discuss computational properties of the different approaches. We also show how existing theory could be adjusted to be directly applicable to the theory of the two approaches, thus proving special theory unnecessary.
We want to bring out the pedagogical aspects of markets mechanisms, in order to take full advantage of its potential.

1 Introduction

The use of markets has been proposed for different resource allocation/optimization problems. The main arguments found in the literature for applying the market concept to these types of problems are:

- The numerous similarities between economic systems and distributed computer systems suggest that models and methods previously developed within the field of mathematical economics can serve as *blueprints* for engineering similar mechanisms in distributed computer systems [1].
- Auction algorithms are highly intuitive and easy to understand; they can be explained in terms of economic competition concepts, which are grounded in everyday experience [2].
- Market approaches enable a natural decomposition, from a software engineering perspective as well as from a computational perspective [3, Chapter 15] and [4].

- Market approaches are very flexible in that they allow for ongoing addition and deletion of agents. No global changes are required—merely the demand/supply relation is altered [3, Chapter 15].
- Markets are informationally efficient in terms of information dimensionality [5], and the abstractions used are the most natural ones for the user [3, Chapter 15].
- Trading resources for some sort of money enables evaluation of local performance and valuation of resources, so that it becomes apparent which resources are the most valuable and which agents are using the most of these [3, Chapter 15].

This article focuses on the presentation of two approaches to optimization in the literature, based on market abstractions. Kurose and Simha [1] investigate a file allocation problem. In this approach, agents report their marginal utility (for having a certain amount of storage) and its derivative to an auctioneer, which reallocates the resource using a resource-oriented Newton-Raphson algorithm until an equilibrium has been reached in which the marginal utilities of all agents are the same. This procedure leads to an optimal allocation and is reported to be computationally efficient.

The problem of allocating n objects to n users has been investigated by Bertsekas [2]. Each user has a valuation of each object, and cannot be assigned more than one object. It is reported that each user can be seen as an economic agent, and it is shown how an auction (which essentially is an English auction) results in an equilibrium. The assumed agent behavior is that an agent bids for the object that is most profitable given the agent's valuation of the objects and the current prices. The bidding increment depends on the valuation and the price of the most preferred object in relation to the valuation and the price of the second most preferred object. When the equilibrium is reached, the allocation is no more than a limited term away from being optimal.

Even though the two examples use ideas that are inspired by microeconomics, they rely on rather unrealistic agent strategies. Therefore we argue that neither of these approaches can be seen as fully market-oriented. For developing this argument, we give a general definition of what a market-oriented approach to optimization is, and show how the approaches under investigation can be conceptually improved to fit the more general framework. We also present some general theory (which interestingly also applies to non-separable resource allocation problems) and show how the rather ad-hoc theory of the examples constitute special cases of the general theory. Thus, the aim of the paper is to bring out the pedagogical advantages of markets. We want to emphasize the concepts that support this view even though we show that this might result in a loss of computational efficiency.

The article is organized as follows. In Section 2 we shortly describe the basic properties of market-oriented programming. In Section 3 the two examples of market-based applications are described in more detail. In Section 4 we generalize the theory, while Section 5 includes discussion and conclusions.

2 Market-oriented programming

Wellman [6] has introduced market-oriented programming as a general approach to deriving solutions to distributed resource allocation problems by a computational economy. Based on the models introduced by Wellman, we require that, for an optimization approach to be considered market-oriented, it must at least fulfill the following basic requirements:

- There must be a well defined *market mechanism* which includes some notion of prices (which often are expressed in terms of some monetary unit). The market mechanism regulates how negotiations and trade are performed among the participating agents and hence determines how certain commodities can be traded for certain other commodities.
- There must be some arguments for why the *agent strategies are reasonably realistic*, given the market model. That is, assume we have some model of information available for the different agents and a well-defined model for how they interact (the market mechanism). Then the strategies must be consistent with the agents' attempt to maximize utility, given bounded rationality.

We are used to different market mechanisms from our daily experience. At least in the industrial countries, *fixed price* markets are the most frequent. In a fixed price market, the customer faces preset prices for different commodities and has only the choice whether to buy or not. This type of market is however of limited interest for application to optimization problems. Efficient market-oriented optimization must generally be based on a *dynamic price* market mechanism. In a dynamic price market the agents do not only choose which commodities to buy/bid for, but also at what prices.

Another important distinction is whether the commodities can be traded for in *discrete* or *continuous* amounts. An example where only discrete amounts make sense is an auction for flight tickets. Many other markets, however, allow for continuous amounts to be traded. Examples of such markets are markets for electricity, bandwidth and pollution rights.

An example of a dynamic price mechanism for discrete amounts is the English auction. In its most simple form, the auction consists of an object to be sold, an auctioneer (which keeps track of the currently highest/winning bid), and a number of bidders. The auction may start at some minimum reserve price. A bidder can place a new bid by bidding at least ϵ above the currently winning bid. When no agent wants to make any further bid the auction stops. Every object is allocated to the agent with the highest bid for that object at the price reported by the winning bid. This type of market mechanism is very common and can, for example, be seen at a number of Internet sites.

An example of a dynamic price mechanism for continuous amounts is equilibrium (or clearing) markets [6]. An equilibrium market establishes a price at which supply meets demand. Each agent then typically declares a *demand/supply function*, telling how much it wants to buy/sell at different prices. The role of the auctioneer is then to search for the price such that all demand matches all supply. When the price is found the search terminates and the resources are reallocated as described by the bids and the clearing price. This type of market is less common, but one example of a market based on these principles is the main power market in the Nordic countries, NordPool [7].

Below we will illustrate the use of both discrete and continuous market mechanisms in the context of optimization problems.

In market-oriented programming the main purpose is not to prove game-theoretic properties of the behavior of agents. Rather, we will restrict ourselves to the weaker assumption of reasonably realistic, intuitive and understandable behavior based on everyday experience. This follows the trend of the AI-community. Further, we make the simplifying assumption that agents do not speculate about how their own behavior or that of other agents will affect the market. This is reasonable if the market is of sufficient size and/or uncertainty is sufficiently large [8, 9]. Thus, the only claim made in this paper is that the agent's attempt to maximize its utility should be reasonable, given the information available by the mechanism. This behavior need not to be game-theoretically optimal.

Our way of reasoning is very consistent with the view of Wellman [10]. He claims that the knowledge, preferences and abilities that must be attributed to the agents will limit their rational behavior. This ”rationality abstraction” is common to artificial intelligence and economics, according to Wellman.

Since we deal with the modeling of optimization problems as markets, we assume the designer to have full control over agent strategies. This means that in practice even unrealistic agent behavior (like in the algorithm by Bertsekas [2]) can be considered for computional reasons. However, to fulfill the requirements of market-oriented programming, the actions of the agents should be grounded in everyday experience. Realistic behavior will give an algorithm which is easy to understand, and thus also easy to implement and use. This paper will show that the conceptual gains of making the behavior more realistic can result in a loss in computional efficiency. We consider the trade-off between these two concepts an interesting question.

If the model is to be used in systems where the designer does not have control over agent strategies, then in the general case computional properties etc. must be demonstrated using game-theoretic reasoning. However, since we make the assumption of non-speculating agents, we consider the potential use in an open system to be another motivation for the need of realistic behavior.

Optimality properties of some basic auctions, similar to the one described by Bertsekas [2] and the English auction described above, are discussed by Demange et al. [11]. More recently, auction-based algorithms for more complicated agent preferences have been examined. Combinatorial auctions, primal-dual algorithms and Vickrey-Clarke-Groves mechanisms are discussed in e.g. [12–15]. Background on basic microeconomic definitions and auction design can be found in [16] or [17].

3 Examples of market-based optimization

3.1 The file allocation problem

Kurose and Simha [1] consider the file allocation problem in a distributed computer system. One file is fragmented into records so that it can be distributed to the different computers in a network. The fragmentation is done in a way that minimizes the communication costs and the average processing delay associated with a file access.

The system consists of n nodes interconnected through a communication network. The network is assumed to be fully connected. The processes running at each of the nodes generate accesses (queries and updates) to the file resource. If a process generates an access request which cannot be satisfied locally, the access is transmitted to the node in the network that holds the requested information.

To be able to state the problem more formally, we make the following definitions:

- x_i is the fraction of the file resource at node i. Since there is only a single divisible resource, $\sum_{i=1}^{n} x_i = 1$. Assuming accesses are made on a uniform basis, x_i also represents the probability that a file access (from anywhere in the network) will be transmitted to node i for processing.
- λ_i is the average rate at which node i generates accesses to the file resource. The networkwide access generation rate is defined $\lambda = \sum_{i=1}^{n} \lambda_i$.
- c_{ij} is the communication cost of transmitting an access from node i to node j and transmitting the answer back to i from j. c_{ii} is taken to be zero.
- C_i is the average systemwide communication cost of making an access at node i. We take this as the weighted sum of the individual communication costs:
$$C_i = \sum_{j \in N} \frac{\lambda_j}{\lambda} c_{ji}.$$
- $1/\mu_i$ is the average service time for an access at node i. We will assume $\mu_i = \mu$ for all nodes i.
- T_i is the expected time delay associated with satisfying an access at node i. In terms of λ, x_i, and μ, we have:
$$T_i = \frac{1}{\mu - \lambda x_i}.$$
(μ is the average number of jobs processed at one node every time unit. λx_i is the average number of accesses to node i per time unit. Thus, $\mu - \lambda x_i$ is the decrease in the length of the queue of waiting jobs at node i per time unit and so $1/(\mu - \lambda x_i)$ is the expected time a job will have to spend in the queue at node i.)
- K is the relative importance of communication costs and access delays. K is here taken to be 1, meaning that communication costs and access delays are equally important.

Now, the problem can be described as follows. Let x_i be the fraction of the file resource stored at node i. The goal is then to find the optimal $(x_1, x_2, \ldots, x_n)$. The overall expected cost of access to the file system is given by

$$\begin{aligned} C &= \sum_{i \in N} (\text{cost of access to } x_i)\text{prob}(\text{accessing } x_i) \\ &= \sum_{i \in N} (C_i + KT_i)x_i \\ &= \sum_{i \in N} \Big(C_i + \frac{K}{\mu - \lambda x_i}\Big)x_i. \end{aligned}$$

In order to adapt to microeconomic theory, Kurose and Simha choose the equivalent problem formulation of maximizing the negative of the cost function. Thus, the utility function $U = -C$ is to be maximized.

Kurose and Simha present two approaches to the problem. The first approach starts with an arbitrary initial feasible allocation. Each node i computes its marginal utility, evaluated at the current allocation. Then each node transmits this value to a central node, which computes the average value. The result is broadcast back to the individual nodes and the file resources are reallocated. Nodes with marginal utility below average utility gets a proportional decrease in file resource, while nodes with marginal utility above average gets a proportional increase. These steps are iterated until the termination condition is met, i.e. the algorithm terminates when the difference between the marginal utilities of two nodes is below some ϵ for every pair of nodes. At each reallocation the algorithm must make sure that no node gets a negative allocation.

Let U_i' be $\partial U(x_1, \ldots, x_n)/\partial x_i$. Furthermore, let A be the set of nodes that could participate in a reallocation without getting a negative allocation. (For a description on how the set A is constructed, see below.) Now, we can state this algorithm more formally:

1. Initialization. An arbitrary feasible allocation $x_i, i \in N$ is made.
2. Iteration.
   DO - Each node $i \in N$ calculates U_i' evaluated at the current allocation and sends U_i' and x_i to the designated central agent.
   - The central agent computes the change in allocation for each node by:

$$\Delta x_i = \alpha\Big(U_i' - \frac{1}{|A|}\sum_{j \in A} U_j'\Big), \qquad \forall i \in A \tag{1}$$

   where α is the stepsize parameter. $\forall i \notin A \Rightarrow \Delta x_i = 0$.
   - Each node $i \in A$, sets $x_i = x_i + \Delta x_i$.

   UNTIL $|U_i' - U_j'| < \epsilon, \forall i, j \in A$.

The construction of the set A is required to ensure that no node's allocation goes below zero in the reallocation process. The algorithm for computing A at each iteration:

1. For all i sort U_i'.
2. Set $A' = \{i | \text{node } i \text{ has largest } U_i'\}$.
3. Do step 4. for each $j, j \notin A'$ in descending order of U_j'.
4. If j would receive a positive allocation x_j as a result of the reallocation defined by Equation (1) with $A = A' \cup \{j\}$, then set $A' = A' \cup \{j\}$.
5. Set $A = A'$.

An alternative approach is to let the reallocation depend on the second partial derivatives. Then equation (1) in the algorithm and the construction of A above is replaced by

$$\Delta x_i = \alpha k_i \Big(U_i' - \frac{\sum_{i \in A} k_i U_i'}{\sum_{i \in A} k_i} \Big), \qquad \forall i \in A \tag{2}$$

where $k_i = 1/|\partial^2 U / \partial x_i^2|$.

Theorem 1. *Both algorithms have the following properties:*

- *When the algorithm converges the allocation is optimal.*
- *For sufficiently small values of the stepsize parameter, α, the algorithm converges to the optimal file allocation.*
- *Each iteration of the algorithm results in a strict increase of the systemwide utility.*
- *The algorithm maintains a feasible file resource allocation at each iteration.*

The two first properties were proved by Heal [18] while the two remaining ones were proved by Kurose and Simha [1].

Conceptual improvements Despite the title of their paper ("A Microeconomic Approach to Optimal Resource Allocation") Kurose and Simha's approach is—apart perhaps from the notions of utility and marginal utility—not very microeconomic. First, the algorithm of computing an allocation such that the marginal utility of each agent is the same for all agents cannot really be called a market mechanism. To motivate a similar behavior the concept of money is needed. Secondly, there is no motivation for why an agent should truthfully report its marginal utility to the auctioneer.[1] As we will see below, there is a very natural way to formulate this approach as a proper market, giving a number of conceptual improvements.

Market mechanism We use a *clearing* or *equilibrium market* as market mechanism, cf. [6, 3]. In an equilibrium market, a price is established such that supply meets demand. This computation is based on a demand function, $z(p)$, which represents an agent's requested *change* in allocation at different prices, further described below. (A supply is represented by a negative demand.) Given the

[1] Admittedly, we also made mistakes similar to the ones made by Kurose and Simha in some of our earlier work, e.g. [19]

demand function for each agent, $z_i(p)$, the clearing (or equilibrium) price is the solution to

$$\sum_{i=1}^{n} z_i(p) = 0, \tag{3}$$

where $z_i(p)$ is the demand function of agent i.

We now turn our attention to how $z_i(p)$ is determined.

Agent strategy Let $U_i(x_i)$ represent U's dependency on x_i, i.e. U as a function of x_i given some allocation to all nodes except i. Given $U_i(x_i)$, we introduce a utility function for each agent defined by

$$u_i(x_i, m_i) = U_i(x_i) + m_i, \tag{4}$$

where m_i represents some "money". In microeconomics, x_i denotes the allocation after a trade. The initial allocation (called endowment) is denoted e, and the difference between the initial and the final allocation is denoted z, i.e. $z = x - e$. The local optimization problem of an agent is then to maximize $u_i(x_i, m_i)$ subject to the constraint that what is bought (z_i) must be paid for in m_i. Since the monetary term is linear we can hence write the optimization problem for an agent as

$$\max_{z_i} U_i(e_i + z_i) - p \cdot z_i.$$

As U_i is a concave function, the demand as a function of the price, $z_i(p)$, is obtained from[2]

$$\begin{aligned}
&\tfrac{\partial U_i(e_i+z_i)-p\cdot z_i}{\partial z_i} = 0 \Leftrightarrow \\
&U_i'(e_i + z_i) = p \Leftrightarrow \\
&(U_i')^{-1}(p) = e_i + z_i \Leftrightarrow \\
&z_i(p) = (U_i')^{-1}(p) - e_i.
\end{aligned} \tag{5}$$

In the Kurose and Simha case we get (with variables as defined above)

$$z_i(p) = \frac{1}{\lambda}\left(\mu - \sqrt{\frac{\mu K}{-C_i - p}}\right) - e_i.$$

Discussion From Equation (3) and Equation (5) we see that the competitive equilibrium is equivalent to the outcome of the Kurose and Simha algorithm. Hence, by formulating the problem properly we have arrived at an approach (with equivalent allocations), which rightfully can be called market-oriented; it is based on a well-defined market mechanism and assumed realistic agent strategies given the mechanism. This gives some conceptual advantages. For example, it is fairly easy for most people to understand the principles: a number of agents make a trade-off between their usage of a file resource and money, and the resource is reallocated among the agents in such a way that supply meets demand. The

[2] This formulation takes the price as given, even though an agent's choice actually has an impact on the prices.

introduction of money will promote truthful bidding. Misreporting might result in an agent having to pay more than it gains in utility or, conversely, it might not be allowed to buy as much as it had wanted to maximize its utility.

Computational considerations The most straightforward way to implement the above mechanism is to let every agent submit a (sampled) demand function ($z(p)$ from Equation (5)), compute the clearing price (p from Equation (3)), and assign the resource in accordance with the bids. For example, if agent one submits $z(p) = 1/p - 4$ and agent two submits $z(p) = 1/p$, the clearing price is $\frac{1}{2}$ and 2 units is sold by agent one to agent two for 1 ($p \cdot z$) unit of money.

The search for the clearing price can be performed in many different ways of which the bisection method [20] (or binary search) is among the simplest, but often sufficiently fast in comparison to e.g. communication delays. More efficient algorithms include standard Newton-Raphson methods [20], e.g.

$$p^{it+1} = p^{it} - \alpha \frac{z(p^{it})}{z'(p^{it})}, \tag{6}$$

where *it* is an iteration index and α is a step size. In heavily distributed environments it often also pays off to use some hierarchical algorithm [21]. One important design issue of all Newton-Raphson methods is the management of the step-size, here denoted α. Kurose and Simha derive a step-size for which the algorithm converges. This is however unnecessary since there are algorithms that automatically adjust the step-size through standard backtracking [20]. This does not only have the advantage of relieving the developer from determining the step-size at design time, but it also speeds up convergence significantly when the algorithm approaches the solution.

In the formulation above, the price was used as the free search parameter. This is commonly referred to as a price-oriented method. However, as seen from Equation (1) and Equation (2), Kurose and Simha use a different method, commonly referred to as a *resource-oriented* method. In the Kurose and Simha algorithms the *allocations* (i.e. resources) are updated until all marginal utilities are equal. There is also a natural market interpretation of these algorithms: at the current allocation, ask the agents how much they are willing to pay for an additional infinitesimal amount of resource. Then the auctioneer tries different reallocations until the price that every agent is willing to pay for an infinitesimal amount of resource is the same. A standard resource-oriented Newton-Raphson approach [20] formulated in market terms looks like [22]

$$z_i^{it+1} = z_i^{it} - \alpha \frac{p(z_i^{it}) - \frac{\sum_{j=0}^{n} \frac{p(z_j^{it})}{p'(z_j^{it})}}{\sum_{j=0}^{n} \frac{1}{p'(z_j^{it})}}}{p'(z_i^{it})}, \tag{7}$$

where—as before—*it* is an iteration index and α is a step-size. As we see from Equation (5), $p(z_i) = U_i'(e_i + z_i)$, and hence Equation (7) is equivalent to Equation (2).

A detailed comparison between price-oriented and resource-oriented approaches can be found elsewhere [3], but some pros and cons of these two methods are outlined here:

- The price functions are closely related to the agents utility functions, and therefore the price function, $p(z)$, can be computed faster than the demand function, $z(p)$, in cases where the demand function cannot be analytically derived.
- As resource-oriented algorithms merely reallocates the existing resource in each step, the reallocation of each iteration is feasible. Hence, gradually better allocations can be performed during the search for the optimal allocation. However, since for most networks the number of messages is more critical than the size of the messages, it is typically preferable to send the demand function as a sample vector. With such an approach, no (communication) iterations are required for computing the solution and the optimal solution will be very rapidly computed compared to the communication delays, and therefore there is no point in sending allocation updates to the agents during the search for the optimal solution.
- For resource-oriented algorithms the management of boundaries is a cumbersome issue. For each iteration, the auctioneer needs to make sure that no agents are assigned an allocation that is outside their boundaries. (This is often the result when the Newton-Raphson algorithms are used, particularly early in the search.) If the resource of one agent is outside its boundaries the excess (or shortage) must be reallocated among the other agents to obtain a feasible allocation.
- The resource-oriented algorithms are more complicated, cf. Equation (6) vs. Equation (7).
- Constant factors are higher for resource-oriented algorithms and therefore price-oriented approaches are significantly faster if the demand functions can be derived analytically.

The conclusion for the Kurose and Simha example is that—since the demand can be analytically derived—a price-oriented approach is highly preferred to a resource-oriented approach (such as the one used by Kurose and Simha).

Discussion Above we have studied the file allocation approach by Kurose and Simha and described how this approach could be improved both conceptually and computationally.

The conversion from an approach that aims at finding an allocation such that all partial derivatives are equal is replaced by a market mechanism that establishes a price such that supply meets demand. It is also described how the agents can be assigned proper utility functions, with which they make a natural trade-off between money and resource. Furthermore, given the market mechanism and the utility functions, an agent strategy leading to a globally optimal solution is realistic from an economics point of view (in bright contrast to the original approach, which gave no arguments for why the agents should report their true marginal utility). It is our belief that this type of improvement makes the system easier to understand and implement (as well as giving a number of other advantages as described in Section 1).

It was also shown that since the demand function can be derived analytically, a price-oriented approach can be used (instead of the resource-oriented approach proposed by Kurose and Simha). Since, price-oriented approaches are less complicated and more computationally efficient, they are typically highly preferable [3], and are recommended for the application at hand.

3.2 The assignment problem

Bertsekas [2] studies the *symmetric assignment problem*; matching n persons and n objects on a one-to-one basis, in such a way that the total benefit is maximized.

Original model The *market mechanism* is not explicitly given by Bertsekas [2]. However, there is one price p_j per object j, which is public to all bidders. Hence, the mechanism is most reasonably interpreted as an open bid *English auction*, i.e. an auction in which the current winning bid for each object is public and the winner, at termination, pays p_j for object j.

The *agent strategy* is to bid the current running price of the most preferred object (given the market prices) plus the surplus (i.e. the agent's valuation of having the object minus the current running price) of the most preferred object minus the surplus of the second most preferred object. Somewhat more formally: Let $\mathcal{A}$ be the set of agent-object pairs (i, j) that can be matched. For each person i, the set of objects that can be matched with i is $A(i) = \{j|(i, j) \in \mathcal{A}\}$. Furthermore, let a_{ij} be the benefit of matching person i with object j. Then the surplus of matching object j to person i is $a_{ij} - p_j$, $(i, j) \in \mathcal{A}$. The surplus of the most preferred object, v_i is

$$v_i = \max_{j \in A(i)} \{a_{ij} - p_j\} \tag{8}$$

and the surplus of the second most preferred object, w_i, is[3]

$$w_i = \max_{k \in A(i), k \neq j} \{a_{ik} - p_k\}.$$

Hence, in each iteration an agent bids for object j (of Equation (8)) at price $p_j + v_i - w_i$.

The auction is run in iterations and, in each iteration, every object is assigned to the agent that make the highest bid for that object. When every agent has been assigned an object the algorithm terminates. Unfortunately, this naive auction algorithm does not always work. If more than two objects offer maximum value

[3] If j is the only object in $A(i)$, then w_i is defined to be $-\infty$ (i.e. in practical computer implementations some large negative constant).

for the bidder i, the bidding increment is zero. Thus, several persons could compete for some objects without raising their prices and the algorithm would be trapped in an infinite loop.

The solution to this problem is to let every agent bid for its most preferred object, but at price $p_j + v_i - w_i + \epsilon$.

With this protocol, the following theorem holds.

Theorem 2. *A feasible assignment according to the auction algorithm together with some price vector is within $n\epsilon$ of being optimal [2].*

This means that when the algorithm stops the cost of the final assignment is within $n\epsilon$ of being optimal. An immediate conclusion is that if all benefits a_{ij} are integers and if $\epsilon < 1/n$ then the auction algorithm terminates in a finite number of iterations with an optimal assignment. This is true because if $n\epsilon < 1$ then any assignment within $n\epsilon$ of being optimal must be optimal. (This generalizes in a straightforward manner to any discretization, not only integers.)

With the mechanisms described above, if the problem is infeasible, the algorithm never terminates (since termination only occurs with a feasible assignment). When problems are not known to be feasible, detection of infeasibility is crucial. Bertsekas [2] shows several ways of handling this.

The aim of Bertsekas was to create a highly efficient and close to optimal algorithm, based on an auction interpretation of a primal-dual algorithm. However, from the viewpoint of market-oriented programming, it has one main conceptual flaw: *the agent strategy is not realistic given the market mechanism.* Hence, it is not possible to view each person as an economic agent acting in its own best interest. The rational (dominant) strategy for a self-interested agent for the given market mechanism would be to bid p_j plus some minimal increment for the most preferred object (given the prices). For example, assume that we have two agents with valuations $a_{11} = 800$ and $a_{12} = 100$ for agent one and $a_{21} = 400$ and $a_{22} = 300$ for agent two. Then with the (sequential variant of) the above scheme, agent one first bids 700 for object one, and agent two then bids 600 for object two. However, with rational agent strategies for the given mechanism, the prices would have become 100 (optionally plus some minimal bid increment) for object one and zero for object two. (They would have only marginally overbid each other for object one until the price reached 100 and then agent two would bid for object two.) This algorithm would of course be very computionally inefficient. In the next sections we will discuss how a more realistic mechanism and more realistic strategies can be introduced at different computational expenses.

Mechanism variant 1 For every object, let there be two public prices, p_j^1 and p_j^2, which reflect the highest and the second highest bids for the object respectively. Further, require that a new bid for an object is accepted only if it is at least $p_j^1 + \epsilon$. Bids cannot be withdrawn. When no more bids are received assign the objects to the highest bidder of the respective objects, but *charge* only p_j^2. (That is, use a second price auction [23].)

Now, with one exception that we return to below, the above agent strategy (with p_j^1 instead of p_j) is very realistic. An agent does not gain anything by bidding below $p_j^1 + v_i - w_i$, since it own bidding price does not affect what it will be charged. The only result of bidding too low is that there is a greater chance of losing the object.

The exception mentioned above is that an economic agent can be assumed to bid $p_j^1 + \max(v_i - w_i, \epsilon)$, rather than $p_j^1 + v_i - w_i + \epsilon$, i.e. it will not increase the bid more than can be motivated by its preferences and the market mechanism—the latter saying that a bid must be an increase by at least ϵ. However, this typically has a relatively small effect on algorithm performance.

Hence, we have derived a new mechanism that better conforms to the market-oriented view, while making small sacrifices in terms of algorithm efficiency. This is in bright contrast to the original model, which has been referred to as an auction algorithm, motivated by the fact that it can be explained in competitive economics terms couched in our everyday experience, but where the agent strategies do not reflect our everyday experience of how people bid in standard auctions.

Though the new market mechanism indeed has made the assumed strategy much more realistic, it could still be in an agent's best interest to deviate from the above strategy. The problem is that if an agent with sufficiently high probability predicts that no other agent places a simultaneous bid, it can be beneficial to bid *too high*. For example, assume that the sequential mechanism suggested by Bertsekas [2] is used. Then only one agent bids at the time, and it is a dominant strategy to bid ∞ for the most preferred object, as you then get to pay the current price for it and no other agent can outbid you. Of course, if agents can place simultaneous bids, there is an enormous risk in doing this—if another agent uses the same strategy it could be fatal. So if, for example, all agents update their bids synchronously in parallel, this need not be a problem.

In the next section we introduce an algorithm that eliminates this problem but has lower computational efficiency.

Mechanism variant 2 For every object, let there be one public price, p_j^2, which reflects the second highest bid for the object. Further, require that a new bid for an object only is accepted if it is at least $p_j^2 + \epsilon$. When putting in a bid, the agent is told whether it is winning the object or not. When no more bids are received, assign the objects to the highest bidder of the respective objects, but charge p_j^2.

Then it is very realistic to assume that every agent bids $p_j^2 + v_i - w_i$ for its most preferred object j, since any price between p_j^2 and $p_j^2 + v_i - w_i$ will maximize agent i's surplus. There is no point for the agent to put in a higher bid than $p_j^2 + v_i - w_i$. If the unknown p_j^1 is higher than $p_j^2 + v_i - w_i$ then the agent will have to pay p_j^1 and thus it will be more profitable for him to bid for another object.

Theorem 3. *In our variant 2, when an agent is allocated an object, this object is the optimal object for the agent, given the final prices. Furthermore, the algorithm will terminate.*

Proof. Let i be any agent. Set $v_i^2 = \max_{j \in A(i)}\{a_{ij} - p_j^2\}$ and $w_i^2 = \max_{k \in A(i), k \neq j}\{a_{ik} - p_k^2\}$.

Let x be $arg \max_{j \in A(i)}\{a_{ij} - p_j^2\}$. Let y be any other object. Agent i bids $p_x^2 + v_i^2 - w_i^2$ for object x. Now assume that $p_x^2 + v_i^2 - w_i^2 > p_x^1$, which means that agent i gets object x. The surplus of agent i will then be $v_i = a_{ix} - p_x^1 > a_{ix} - (p_x^2 + v_i^2 - w_i^2) = a_{ix} - (p_x^2 + a_{ix} - p_x^2 - w_i^2) = w_i^2 \geq a_{iy} - p_y^2 > a_{iy} - p_y^1$. Thus $arg \max_{j \in A(i)}\{a_{ij} - p_j^2\} = arg \max_{j \in A(i)}\{a_{ij} - p_j^1\}$, i.e. the agent would have preferred the object it is winning even with complete information available.

Clearly, it will also prefer the object it is winning given the highest bids on all objects.

The algorithm will terminate since a new bid has to be at least $p_j^2 + \epsilon$, and p_j^2 of this scheme cannot be above p_j^1 of the above scheme. □

In section 4 we will show how this theorem will imply the optimality of this variant of the mechanism.

This approach has lower computational efficiency than the one above. This is not surprising: Generally we cannot expect mechanisms that are based on the assumption that the agent only reveals the information that is most profitable for them to be as efficient as mechanisms where the agents' preferences are publicly available knowledge.

Discussion In this section we have discussed different aspects of Bertsekas' auction algorithm. We first discussed the fact that no mechanism was originally given that motivated the proposed agent strategies. Then we introduced a new scheme (with slightly lower computational efficiency) that made the strategies more realistic and therefore made the entire approach more intuitive from an economics point of view. We showed that under certain circumstances we needed to modify the mechanism and agent strategies further to obtain an economically realistic model. The different mechanisms were implemented and tested for computational efficiency. The results, shown in Fig 1, indicates that methods that are more realistic (like *realistic* and *variant 2*) are less effective than the less realistic methods (*original* and *variant 1*).

The main conclusion from this exercise is that there sometimes is a trade-off between computational efficiency and economic realism. In the case of a standard optimization problem with publicly available information, economic realism (with associated conceptual advantages) can of course be sacrificed for computational efficiency, by imposing more or less unrealistic strategies on the agents. However, if we assume that we are to assign the objects to real self-interested persons, we must on the other hand assume that they act in their own best interest. Then we cannot generally hope to obtain efficient methods.

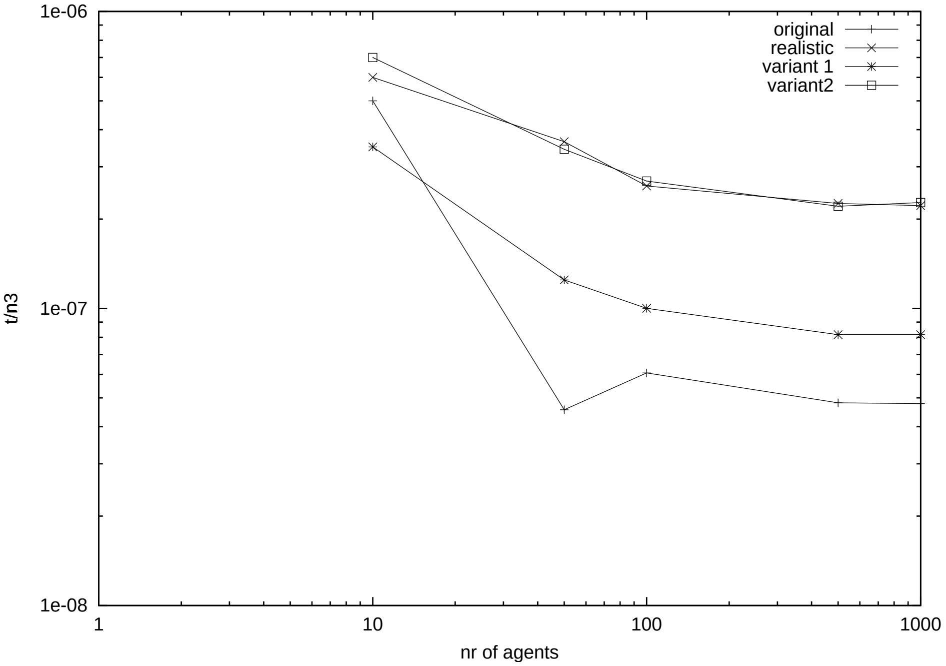

Fig. 1. Test of the computional efficiency of the different algorithms, comparing n, the number of agents, with the time of the computation, t, divided by n^3. *Original* is Bertsekas original model with bidding increment $v_i - w_i + \epsilon$, *realistic* has bidding increment ϵ, *variant 1* has bidding increment $\max(v_i - w_i, \epsilon)$ and finally in *variant 2* the new bid is $p_j^2 + \max(v_i - w_i, \epsilon)$. The agents valuations for the respective items are (independently) drawn from a rectangular distribution.

4 Generalizing the theory

The theory of the two examples [1, 2] examined in this article is very specific to the examples. In this section we show that properly formulating the optimization problems as market problems not only has conceptual advantages; it also allows us to use a more general theory. The needed theory exists in the literature of economics and linear programming. Thus, the specialized theory of the examples is unnecessary. One of the main advantages of market-oriented programming was stated in section 1 as the ability to use theories from economics in distributed computer systems. By using the general theory, this advantage could be applied to the examples. This brings out the close connection to economic theory, which was not clear from the original papers.

In this section the existing theory is adjusted to exactly suit the examples. We also show that some of the theory applies to the case of non-separable allocation problems (as the one by Kurose and Simha).

We first define a type of optimization problems to which the theory of this section can be applied.

Definition 1. *The maximization problem MP is given by*

$$\begin{array}{ll}\max_{\mathbf{x}} \sum_{i=1}^{n} f_i(\mathbf{x}) \\ s.t. \quad \sum_{i=1}^{n} \mathbf{x}_i = \mathbf{X},\end{array}$$

$\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{ik}]$, $x_{ij} \in S_{ij}$, $\mathbf{S}_i = [S_{i1}, S_{i2}, \ldots, S_{ik}]$, $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$, $\mathbf{S} = [\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_n]$, $\mathbf{X} = [X_1, X_2, \ldots, X_k]$, $f_i : \mathbf{S} \mapsto \Re$, S_{ik} is any set of reals (e.g., all reals, all positive integers or $\{1.25, 2.77, 4\frac{1}{3}\}$) and X_i is the total available amount of commodity i. The problem is interpreted as a resource allocation problem with n agents and $f_i(\mathbf{x})$ as agent i's valuation of the allocation $\mathbf{x}$.

Next we give a proper market definition corresponding to the optimization problem of definition 1.

Definition 2. *With the notation from MP, the market M is a market of n agents and k commodities. All agents want to maximize their utility function defined by*

$$u_i(\mathbf{x}, m_i) = f_i(\mathbf{x}) + m_i, \tag{9}$$

$m_i \in \Re$, i.e. $u_i(\mathbf{x}, m_i)$ is quasi-linear with respect to m_i.

$\mathbf{X}$ is distributed as endowment among the agents. Let the endowment be $\mathbf{e} = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n]$, $\mathbf{e}_i = [e_{i1}, e_{i2}, \ldots, e_{ik}]$ and let the endowment of m be $\mathbf{m}^\circ = [m_1^\circ, m_2^\circ, \ldots, m_n^\circ]$. Let the market price of m be 1, i.e. let

$$m_i = m_i^\circ - \sum_{j=1}^{k} (x_{ij} - e_{ij}) p_j$$

where $p_j \in \Re$ denotes the respective market price for the k commodities.

Then we can state the following theorem.

Theorem 4. *Any Pareto-optimal allocation in M is a solution to MP.*

Proof. Let $\Delta f = \sum_{i=1}^{n} f_i(\mathbf{x}^b) - f_i(\mathbf{x}^a)$, $\sum_{i=1}^{n} \mathbf{x}_i^a = \mathbf{X}$, and $\sum_{i=1}^{n} \mathbf{x}_i^b = \mathbf{X}$. For any allocation $\langle \mathbf{x}^a, \mathbf{m}^a \rangle$ in M, $\langle \mathbf{x}^b, \mathbf{m}^b \rangle$ (where $m_i^b = m_i^a + u(\mathbf{x}^a, m_i^a) - u(\mathbf{x}^b, m_i^a) + \frac{\Delta f}{n}$) is a Pareto-improvement if $\Delta f > 0$. ($\sum_{i=1}^{n} m_i^b = \sum_{i=1}^{n} m_i^a$, because $\sum_{i=1}^{n} u_i(\mathbf{x}^a, m_i^a) - u_i(\mathbf{x}^b, m_i^a) = -\Delta f$.) In other words: if an allocation is not a solution to MP it is not Pareto-optimal in M. Thus, if an allocation is Pareto-optimal in M it is a solution to MP. □

Hence, Theorem 4 implies that if we can devise a market mechanism that exploits all possible profitable reallocations, it will lead to a globally optimal solution. An important remark is of course that even though it is sometimes possible to find such a mechanism, the reallocation problem in general is $\mathcal{NP}$-hard [24], i.e. can be computationally intractable. Furthermore, the conditions that make problems hard to solve with markets also tend to make them hard to solve by conventional means. As an example, concave objective functions (i.e. the functions $f_i(\mathbf{x})$ above) ensure the existence of an equilibrium and enable straightforward market implementation. If the functions are non-concave, however, the problem is less straightforward to formulate in market terms and at the same time harder to solve conventionally [24].

Non-separable functions A utility function is said to be *separable* if the utility of one agent only depends on the same agent's allocation. The definitions and theorems above also capture the case of non-separable functions. This deserves some further discussion.

Starting with Theorem 4, the notion of Pareto optimality is commonly discussed in the context of separable functions. However, the concept of Pareto optimality makes perfect sense also with non-separable functions. With separable functions Theorem 4 captures the fact that if there is another allocation with higher total utility, then there is always some agent (or set of agents) that is so much better off by changing its allocation, that it can compensate the agent (or set of agents) that gets worse off. With non-separable functions Theorem 4 instead tells us that if there is another allocation with higher total utility, then there is always some agent (or set of agents) that is so much better off by changing its *or any other agent's* allocation, that it can compensate the agent (or set of agents) that gets worse off.

In a corresponding manner, also Theorem 5 below (and its corollaries) has a straightforward interpretation in the non-separable case.

When generalizing the theory for the optimality of the assignment problem of Section 3.2, we have made the following interesting observation: When generalizing the problem to a high enough level, it becomes trivial. When having the general theorem, the theory of the investigated assignment problem (as well as some other theory from the literature) is obtained as trivial corollaries.

Theorem 5. *With basic definitions from Definition 1, assume that we have a resource reallocation problem, where the current allocation* $\mathbf{e}$*, is to be reallocated to a new allocation* $\mathbf{x}$*. Furthermore, let there be a cost (that may be positive or negative) for each agent associated with moving from* $\mathbf{e}$ *to* $\mathbf{x}$*. Call this cost* $c_i(\mathbf{e}, \mathbf{x}_i)$*. Define the objective of the resource allocation as*

$$\begin{array}{ll} \max_{\mathbf{x}} & \sum_{i=1}^{n} f_i(\mathbf{x}) - c_i(\mathbf{e}, \mathbf{x}_i) \\ s.t. & \sum_{i=1}^{n} \mathbf{x}_i = \mathbf{X}. \end{array}$$

An upper bound on the objective is

$$\begin{array}{ll} \max_{\mathbf{x}} & \sum_{i=1}^{n} f_i(\mathbf{x}) - c_i(\mathbf{e}, \mathbf{x}_i) \\ s.t. & \sum_{i=1}^{n} \mathbf{x}_i = \mathbf{X} \end{array} \leq \sum_{i=1}^{n} \begin{array}{l} \max_{\mathbf{x}} f_i(\mathbf{x}) - c_i(\mathbf{e}, \mathbf{x}_i) \\ s.t. \sum_{i=1}^{n} \mathbf{x}_i = \mathbf{X}, \end{array}$$

Proof. The theorem is obvious; the maximized sum is clearly smaller than or equal to the sum of the maximized terms. □

Corollary 1. *If $c_i(\mathbf{e}, \mathbf{x}_i) = \mathbf{p}(\mathbf{x}_i - \mathbf{e}_i)$, then*

$$\begin{array}{ll} \max_{\mathbf{x}} \sum_{i=1}^{n} f_i(\mathbf{x}) \\ s.t. \quad \sum_{i=1}^{n} \mathbf{x}_i = \mathbf{X} \end{array} \leq \begin{array}{l} \sum_{i=1}^{n} \max_{\mathbf{x}} f_i(\mathbf{x}) - \mathbf{p}(\mathbf{x}_i - \mathbf{e}_i) \\ s.t. \sum_{i=1}^{n} \mathbf{x}_i = \mathbf{X}, \end{array}$$

since $\sum_{i=1}^{n} \mathbf{p}(\mathbf{x}_i - \mathbf{e}) = 0$ (i.e. all payments sum to zero, everything is transferred between the agents). In other words, if the only costs are the payments between the agents, the optimal solution is smaller than or equal to the sum of the utilities of the optimal choice for the respective agents, for any given price vector.

Corollary 2. *By adding $\sum_{i=1}^{n} f_i(\mathbf{e})$ to the expression in Corollary 1, we get*

$$\begin{array}{ll} \max_{\mathbf{x}} \sum_{i=1}^{n} f_i(\mathbf{x}) - f_i(\mathbf{e}) \\ s.t. \quad \sum_{i=1}^{n} \mathbf{x}_i = \mathbf{X} \end{array} \leq \begin{array}{l} \sum_{i=1}^{n} \max_{\mathbf{x}} f_i(\mathbf{x}) - f_i(\mathbf{e}) - \mathbf{p}(\mathbf{x}_i - \mathbf{e}_i) \\ s.t. \sum_{i=1}^{n} \mathbf{x}_i = \mathbf{X}. \end{array}$$

That is, the total improvement can not be greater than the sum of the desired improvements of the respective agents at a given price vector.

Corollary 3. *Our version of the Kurose and Simha file allocation problem is optimal.*

Corollary 4. *Theorem 2, the proof for upper bound on distance to global optimality for the assignment problem, is a consequence of Corollary 2. So is the theorem on upper bound on distance to optimality by Walsh and Wellman [25].*

Corollary 5. *Our version 2 of the assignment problem is optimal.*

Proof. This follows directly from Corollary 2 and Theorem 3. □

There is another interesting comment to make about Corollary 2. If we generalize the notion of competitive equilibrium to properly incorporating nonseparability, we directly see that the competitive equilibrium is globally optimal. So, we generalize the definition of competitive equilibrium from "a price vector and allocation such that supply meets demand and at the given prices, no agent's utility can be increased with respect to those prices by changing its allocation (i.e. buy or sell the different commodities)" to "a price vector and allocation such that supply meets demand and at the given prices, no agent's utility can be increased with respect to those prices by changing its *or any other agent's* allocation". We observe that Corollary 2 captures the optimality of the generalized equilibrium.

Hence, we see that we have arrived at a very general theory, and that formulating the applications of Kurose and Simha and Bertsekas in proper market terms make them fit into a broader theoretical framework.

5 Discussion and Conclusions

In this article we have discussed market-oriented approaches to optimization. We have presented some general definitions of what a market-oriented approach to optimization is and argued that the basic requirements are that it should rely on some well-defined *market mechanism* and some *agent strategies* that are reasonably realistic given the market mechanism.

To illustrate the general framework for market-oriented approaches to optimization, two examples from the literature—the microeconomic approach to the file allocation problem by Kurose and Simha [1] and the auction algorithm for the assignment problem by Bertsekas [2]—have been analyzed. Some theory from the literature was reformulated to match the examples, thus showing specialized theory unnecessary.

It was shown that a number of conceptual improvements could be made to the Kurose and Simha approach, making the model more market-like from the perspective of how people generally understand markets. Some computational discussions showed that the file allocation problem could be managed by a *price-oriented* approach rather than the *resource-oriented* approach presented by Kurose and Simha. Generally, price-oriented approaches are simpler (easier to implement) and more computationally efficient when the demand functions can be derived analytically (as could be done in the analyzed case).

With respect to the auction algorithm for the assignment problem, different mechanisms were discussed and reasonable strategies introduced. It was shown that the original approach does not reflect agent behavior in real auctions (as suggested by the motivation of the algorithm), but that there are more reasonable mechanisms and agent strategies with high computational efficiency. These mechanisms are hence also better suited for applications in real markets (i.e. when the different agents truly represent different interests).

In the case of the file allocation application, we showed that one could define market mechanisms and reasonable agent strategies that are highly efficient. However, there is a catch preventing straightforward application of these methods to real markets. The problem is the utility function. In Equation (4) and just above that equation, we simply assigned a utility function to each agent. However, in a market setting, this would not be the utility function of a self-interested agent (since the current utility function is based on the notion of global utility). The main problem is that the utilities are *non-separable*, i.e. dependent on the allocations of the other agents. In Section 4 we show that existing theory could be generalized to capture even this case.

One of the most interesting results from the comparison between different auction mechanisms for the assignment problem is that we found a trade-off between computational efficiency and economic realism. Whereas the economical efficiency aspects of strategic behavior are relatively well understood and described (see e.g. [26, 27] for good introductory overviews), the computational aspects of strategic behavior are much less so. The different variants above suggest that there may be an interesting connection that seems interesting to look into further.

References

1. J. F. Kurose and R. Simha. A microeconomic approach to optimal resource allocation in distributed computer systems. *IEEE Transactions on Computers*, 38(5):705–717, 1989.
2. D. Bertsekas. Auction algorithms for network flow problems: A tutorial introduction. *Computational Optimization and Applications*, (1):7–66, 1992.
3. F. Ygge. *Market-Oriented Programming and its Application to Power Load Management*. PhD thesis, Department of Computer Science, Lund University, 1998. ISBN 91-628-3055-4, CODEN LUNFD6/(NFCS-1012)/1-224/(1998) (Available from http://www.enersearch.se/ygge).
4. A. Th. Schreiber, J. M. Akkermans, and al. *Knowledge Engineering and Management*. The MIT Press, Cambridge, MA, 1999. In Press.
5. J. S. Jordan. The competitive allocation process is informationally efficient uniquely. *Journal of Economic Theory*, 28:1–18, 1982.
6. M. P. Wellman. A market-oriented programming environment and its application to distributed multicommodity flow problems. *Journal of Artificial Intelligence Research (www.jair.org/)*, (1):1–23, 1993.
7. NordPool. The elspot market—The spot market, October 1998. (Available from www.nordpool.no).
8. D. J. Roberts and A. Postlewaite. The incentives for price-taking behavior in large exchange economies. *Econometrica*, 44(1):115–127, 1976.
9. T. W. Sandholm and F. Ygge. On the gains and losses of speculation in equilibrium markets. In *Proceeding of the Fifteenth International Joint Conference on Artificial Intelligence, IJCAI 97*, pages 632–638, August 23–29, 1997. (Available from http://www.enersearch.se/ygge).
10. M. P. Wellman. Market-oriented programming: Some early lessons. In S. Clearwater, editor, *Market-Based Control: A Paradigm for Distributed Resource Allocation*, chapter 4. World Scientific, 1996. (More info about the book at http://www.wspc.co.uk/wspc/Books/compai _hilites.html#mbc The paper is available from http://ai.eecs.umich.edu/people/wellman/Publications.html).
11. G. Demange, D. Gale, and M. Sotomayor. Multi-item auctions. *Journal of Political Economy*, 94(4):863 – 872, 1986.
12. S. de Vries and R. Vohra. Combinatorial auctions: A survey. *INFORMS Journal on Computing*, 15(3):284–309, 2003.
13. David C. Parkes and Lyle H Ungar. An ascending-price generalized Vickrey auction. Technical report, Harvard University, 2002.
14. Debasis Mishra and David C. Parkes. Ascending price Vickrey auctions using primal-dual algorithms. Technical report, Harvard Univeristy, 2004.
15. N. Devanur, C. Papadimitriou, A. Saberi, and V. Vazirani. Market equilibrium via a primal-dual-type algorithm. In *43rd Symposium on Foundations of Computer Science, FOCS 2002*, pages 389–395, November 2002.
16. A. Mas-Colell, M. Whinston, and J. R. Green. *Microeconomic Theory*. Oxford University Press, 1995.
17. H. R. Varian. *Intermediate Microeconomics—A Modern Approach*. W. W. Norton and Company, New York, 1999. Fifth Edition.
18. G. Heal. Planning without prices. *Review of Economic Studies*, 36:346–362, 1969.
19. F. Ygge and J. M. Akkermans. Power load management as a computational market. In M. Tokoro, editor, *Proceedings of the Second International Conference on Multi-Agent Systems, ICMAS'96*, pages 393–400. AAAI Press, Menlo Park, CA, December 9–14 1996. (Available from www.enersearch.se/ygge).

20. W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipies in C*. Cambridge University Press, 1994. Second Edition.
21. A. Andersson and F. Ygge. Managing large scale computational markets. In H. El-Rewini, editor, *Proceedings of the Software Technology track of the 31th Hawaiian International Conference on System Sciences (HICSS31)*, volume VII, pages 4–14. IEEE Computer Society, Los Alamos, January 1998. ISBN 0-8186-8251-5, ISSN 1060-3425, IEEE Catalog Number 98TB100216. (Available from http://www.enersearch.se/ygge).
22. F. Ygge and J. M. Akkermans. On resource-oriented multi-commodity market computations. *Autonomous Agents and Multi-Agent Systems*, 3(1), 2000. In press.
23. W. Vickrey. Counter speculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.
24. A. Andersson and F. Ygge. Efficient resource allocation with non-concave objective functions. Technical report, EnS 99:1, EnerSearch AB, 1999. Submitted for journal publication. (Available from http://www.enersearch.se).
25. W. Walsh and M. Wellman. Efficiency and equilibrium in task allocation economies with hierarchical dependencies. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI 99)*, pages 520 – 526. Morgan Kaufmann Publishers, Inc., July 1999.
26. R. B. Myerson. Bayesian equilibrium and incentive compatibility: An introduction. In L. Hurwicz, D. Schmeidler, and H. Sonnenschein, editors, *Social Goals and Social Organization*, chapter 8, pages 229–259. Cambridge University Press, 1985.
27. E. Muller and M. Satterthwaite. Strategy-proofness: The existence of dominant-strategy mechanisms. In L. Hurwicz, D. Schmeidler, and H. Sonnenschein, editors, *Social Goals and Social Organization*, chapter 8, pages 131–171. Cambridge University Press, 1985.