Comput Econ (2008) 31:381–395
DOI 10.1007/s10614-008-9122-x

A Pricing Mechanism for Resource Management in Grid Computing

Panos Parpas · Berç Rustem

Received: 18 May 2007 / Accepted: 7 January 2008 / Published online: 22 February 2008

Abstract We consider the problem of efficient resource allocation in a grid computing environment. Grid computing is an emerging paradigm that allows the sharing of a large number of a heterogeneous set of resources. We propose an auction mechanism for decentralized resource allocation. The problem is modeled as a multistage stochastic programming problem. Convergence of the auction allocations to the social optimum is established. Numerical experiments illustrate the efficacy of the method.

Keywords Grid computing · Decentralized resource allocation · Multistage stochastic programming

1 Introduction

Grid computing is an emerging paradigm that allows the sharing of a large number of a heterogeneous set of resources. Computational grids were inspired by electricity grids. Grids are emerging in many environments including research, academic, as well as commercial institutions. The areas where grid computing methodologies can be used includes all fields where substantial computational power and resources are required. As a result this paradigm from computer science has found many applications, for example physics (e.g. turbulence simulations), cosmology, bio-informatics, genetics, and the life sciences (Distributed European Infrastructure Supercomputing Applications; The London e Science Centre). Managing distributed, loosely-coupled, heterogeneous resources is an extremely difficult task. The grid computing community is trying to address these issues, but there are a lot of open issues that need to be addressed. The purpose of this paper is to propose a quantitative framework, based on

P. Parpas (✉) · B. Rustem
Department of Computing, Imperial College, London SW7 2AZ, UK
e-mail: pp500@doc.ic.ac.uk

tools from operations research and economics literature, in order to address the issue of managing such a network of resources.

As in physical systems, one can think of resource management as consisting of three layers: the micro-scopic, meso-scopic, and macro-scopic layer (Birge and Dempster 1996). In the context of grid computing the micro-scopic layer will consist of detailed decisions e.g. routing paths for data, error correction and retransmission policies. In the meso-scopic layer we need to address higher level problems e.g. which machine will run a particular job-request, what encryption algorithm to use etc. In the macro-scopic layer one encounters strategic issues. For example, whether an agreement with another grid owner will need to be made in order meet user QoS requirements. Obviously there is a lot of interaction between these three layers. The problem addressed in this paper belongs to the meso-scopic layer. However, the method we chose to address the resource allocation problem can be used in the other layers too. In particular we will use a pricing mechanism to solve a grid resource management problem. One of the inputs to the proposed mechanism is an error tolerance parameter. The error arises from the fact that we use a decentralized resource management strategy. With the proposed approach, problems in the micro-scopic layer (that need to be solved in almost real time) can be solved with lower accuracy, while problems in the micro-scopic layer can be solved to optimality.

This is not the first time that pricing has been used as an aid to managing computer resources. Pricing of resources started in 1968 with the paper by Sutherland (1968). Even though very little work has been done on grid computing resource management, there is a lot of related work from models proposed to manage networks, and the Internet in particular. Contributions in this direction using concepts from game theory and microeconomics have been made by various authors, see Papadimitriou (2001) for a review. In this line of research one assumes that each stakeholder (e.g. ISP, router, etc.) is an agent that selfishly tries to optimize its own utility function. One then tries to calculate the loss of efficiency in the network by comparing the 'selfish' strategies with the social optimum. To compute the social optimum one optimizes (typically) a sum consisting of all the different utility functions of each agent. The social optimum or welfare problem requires a central authority to have complete knowledge over the state of the system. This assumption is unrealistic in real-world networks, and is unrealistic in the context of grid computing as well.

More related to this paper is work dealing with decentralized resource management. When managing computing resources, and grids of IT resources in particular, one cannot address these problems using a centralized approach. This is the motivation behind decentralized resource management. In Gupta et al. (1997), a model for achieving efficient utilization of Internet resources using an equilibrium approach was proposed. Gupta et al. used pricing in order to 'drive' the community of agents towards equilibrium. Convergence of the pricing mechanism towards social welfare maximization was not studied. Thomas et al. (2002) used Scarf's algorithm Scarf (1967, 1973) to establish convergence between the decentralized model and the social welfare problem. In the decentralized model each agent tries to optimize its own utility function; while in the social welfare the sum of all the utility functions is optimized. Stoenescu and Teneketzis (2002) removed some of the assumptions of the Thomas et al. paper. The papers of Thomas et al. (2002), and Stoenescu and Teneketzis (2002) are the ones mostly related to this work in the sense that we too formulate a social welfare problem, we then use pricing to guide the set of selfish agents towards the social optimum.

The obvious variation of this paper from the work mentioned above is that we apply similar ideas to context of grid computing. However, this paper makes more contributions too. In Thomas et al. (2002) and Stoenescu and Teneketzis (2002) Scarf's algorithm is used as the basic principle behind the auction mechanism. It is well known that Scarf's algorithm can be inefficient. Consequently it is not well suited as the pricing mechanism for managing grid computing resources. We use a decomposition algorithm, developed in Holmberg and Kiwiel (2006), as the mechanism to guide the agents to behave in such a way that the social optimum is achieved. Another contribution of this paper is the inclusion of stochastics and dynamics in the realm of decentralized resource allocation. While we do not address the decomposition of the subproblems at the level of individual agents it is clear that decomposition algorithms can be used to solve each agents subproblems. In summary, this paper makes contributions to the grid computing community by providing a model for optimal decentralized resource allocation. We also contribute to the resource management literature by developing an efficient algorithmic paradigm. We are primarily concerned with the problem of decentralized resource allocation under uncertainty. However, one can also pose the problem in the form of a multi-stage stochastic programming problem. Consequently, the proposed algorithm can be viewed as a decomposition algorithm for a class of large scale stochastic programming problems.

The rest of this paper is structured as follows: in Sect. 2 we present our model. We first define the social welfare problem and then we go on to define the problem solved by each agent. We start with a static model and then go on to define a dynamic model that also includes uncertainty. In Sect. 3 we define the auction mechanism, and establish convergence to the social optimum. Numerical experiments are given in Sect. 4. We conclude by discussing possible extensions of this work in Sect. 5.

2 A Model for Resource Management

Let $r = \{1, \dots, R\}$ represent the resources available on the grid; and let $k = \{1, \dots, K\}$ represent the agents competing for these resources. We use $x \in X \subset \mathbb{R}^{RK}$ to denote the levels made available to each agent for each resource. These resources may represent CPU cycles available to a particular machine, the maximum bandwidth available on a link, and so on. When dealing with problems in grid computing it is paramount to deal with services rather than low level hardware/software resources. For this reason we postulate the existence of a concave function: $A_k : \mathbb{R}^{RK} \to \mathbb{R}^B$, that translates a particular configuration of resources into *basic computing blocks*; where $b = \{1, \dots, B\}$ represents the different basic blocks. Typically this function will be linear. For example, if $\hat{x} \in X$ is a particular configuration of the resources then $A(\hat{x})$ will represent the 'output' of the grid in terms of computing blocks. These blocks can be thought of as the basic components needed to run a service, examples include random number generation, solution of an optimization problem, image rendering and so on.

In the grid community there is a tendency to move away from resource management of hardware resources to the management of services. This is the motivation for distinguishing between resources (e.g. a web-server) and computing blocks. By computing blocks we mean the resources that need to be put together to deliver a service (e.g. a web service). For example, a web-service may provide access to a database over a browser. The computing blocks required to deliver the service include multiple resources (e.g. database server, software, web server, and security software).

We postulate the existence of K agents. These agents accept requests for services from users. The agents use the resources on the grid to construct services which they deliver to their users. Each agent is therefore competing for the resources on the grid. Agents derive their benefit by serving their users and meeting the demand. We believe that the modeling choice we made for representing the grid resources as blocks of computation represents a reasonable compromise for dealing with low level resources (e.g CPU cycles), and the high level requests for services (e.g. pricing derivatives for a financial application). We use $q = \{1, \dots, Q\}$, to denote the different services requested by users. Again we assume that services can be mapped into basic computing blocks using a convex function $B_i : \mathbb{R}^Q \to \mathbb{R}^B$. A constraint on the owner of the grid is to meet the demand of the users, we therefore must have:

$$A_k(x) - B_k(y_k) \geq 0 \quad k = 1, \dots, K,$$

where A_k maps the configuration of the grid resources to basic blocks that will be made available to agent k; y_k represents the request for services made by agent k. The convexity of the model will play a key role in our convergence proof. Given the demand constraint above it is natural to assume that A_k, and B_k are linear. We will use A_{kb}, and B_{kb} to denote the bth block of the kth agent for the two mappings for resources and services, respectively.

Each agent must also meet the demand from its users. We will assume that when users make high level requests for packages of services the concave function h_k maps services into packages. We must then have:

$$h_k(y_k) - d_k \geq 0,$$

where d_k represents the demand of agent k. We also assume that y_k belongs to a compact convex set Y_k.

In our hypothetical grid market we will use two types of agents: the *grid agent* which will be responsible for managing the resources. The second type we call the *market agents* and these will be responsible for meeting the demands from the users. The basic idea is that one central authority (the grid agent) manages the resources. The market agents accept requests from users and try to secure resources on the grid in order to perform the required computations. If the grid agent decides to make allocation $\hat{x}$ to the market agents then the grid agent will incur some cost. This cost will be modeled using a convex utility function c_0. The grid agent has access to a bounded amount of resources. This feature of the problem will be modeled with the following constraint: $\sum_j x_{ij} \leq r_i \quad i = 1, \dots, R$. Where x_{ij} denotes the amount of the ith resource allocated to the jth market agent; r_i will be used to denote the total amount of resource i.

Similarly, if the kth market agent decides to request $\hat{y}_k$ then the agents will derive some benefit which will be modeled using a concave function c_k. Putting all these ideas together we can formulate the social welfare problem as follows:

$$
\begin{aligned}
\min\quad & c_0(x) - \sum_{k=1}^{K} c_k(y_k) \\
& A_k(x) - B_k(y_k) \geq 0 \quad k = 1, \ldots, K \\
& \sum_j x_{ij} \leq r_i \quad i = 1, \ldots, R \\
& h_k(y_k) - d_k \geq 0 \quad k = 1, \ldots, K \\
& y_k \in Y_k,\ k = 1, \ldots, K \quad x \in X.
\end{aligned}
\tag{SW-A}
$$

The exact benefit the agent will derive from meeting the demand is never exactly known in advance. More importantly, demand for services is in general random, and decisions need to be taken over several time periods. For these reasons we propose to study the problem as a multistage stochastic programming problem. Even though a chance constraint formulation will also be relevant, we will use a recourse formulation. The reason for considering multiple time periods is that work services done on grids usually have temporal relationships that we would like to capture. For example, the output from one computation may be input to another, or that one process may not be able to start unless another finishes etc. Before we present our model in full, it is pertinent to explain the basic idea behind stochastic programming. We refer the interested reader to Birge and Louveaux (1997) for a textbook treatment of these issues. Since resources are not perfectly divisible, the whole problem should be modeled as an integer programming problem. However, the resulting problem would be much more difficult to solve. This is a definite limitation of the proposed model.

Typically resource management problems need to be addressed over several time periods $t = \{1, \ldots, T\}$. These time periods may represent the different demands in peak and off-peak times. It is also possible to represent work-flows in this manner. The uncertainty in the model enters in two ways. Firstly, the agents are unsure of the exact benefit they will derive from a particular strategy. Secondly, the agents are unsure of the demand they will encounter in the next time periods. These uncertainties can be represented as a scenario tree. The root of the tree represents the state of the world that is deterministic. At the root node the grid agent decides x_0, the resource allocation for the first time period. The kth market agent decides y_{0k}, the demand for services in the first time period. All benefits, costs, and demands are known at this stage. As we move down the scenario tree, different events represent different realizations of the uncertainties. Each level of the tree represents a different time period. The path from the root to a leaf node is termed as a scenario. Thus a node on the tree (i.e an event) can be indexed using (s, t). We will use $a(s, t)$ to denote the parent node of scenario s at time t. The basic framework can be described as follows: the grid agent makes a decision x_0 at time 0; then at time $t = 1$, the agent is faced with different possible scenarios concerning the cost of a particular strategy. The agent then makes a decision based on the new observation. At time t, under scenario s, the cost incurred by the grid agent is represented by $c_0(x_t^s, x_k^{a(s,t)}, \xi_t^s)$. In other words, if the grid agent decided to follow strategy $\hat{x}^{a(s,t)}$ in the previous time period, then observed the random variable ξ_t^s and decided to follow strategy $\hat{x}_t^s$ the costs incurred by this choice of strategy is given by: $c_0(\hat{x}_t^s, \hat{x}^{a(s,t)}, \xi_t^s)$, where ξ_t^s is a random variable having its support in a finite set $\Xi = \Xi_1 \times \cdots \times \Xi_T$. Decisions in stochastic programming need to be non-anticipative. This means that decisions must depend on the past and not the future. In the context of discrete probability distributions the concept of non-anticipativity can be represented using a compact or a split-view formulation (Rockafellar and Wets 1991). In the compact formulation the data of the problem can be mapped directly onto a tree structure as described above. We chose to use a split variable formulation. In this framework new decision variables are introduced so that the large-scale problem is decomposed into many different subproblems. Conceptually, using this approach the non-anticipativity constraints are completely relaxed. In order to enforce these constraints new constraints are introduced that 'rebuild' the links between nodes. We will use X_t^s, and Y_t^s to denote the set of non-anticiaptivity constraints for the grid agent and market agents, respectively.

The grid agent will want to compute a strategy that performs the best on average over all time periods and scenarios. Applying similar reasoning to each of the market agent's problem we find that the objective function of the stochastic social-welfare problem should be given by:

$$
\begin{aligned}
C(x, y) \triangleq c_0(x_0) - \sum_{k=1}^{K} c_k(y_{0k}) + \sum_{t=1}^{T}\sum_{s=1}^{S} p_t^s (c_0(x_t^s, x^{a(s,t)}, \xi_t^s) \\
- \sum_{k=1}^{K} c_k(y_{kt}^s, y_k^{a(s,t)}, \omega_t^s)),
\end{aligned}
$$

where p_t^s represents the probability of scenario s occurring at time t. As before y_{kt}^s represents the kth agent's decision at time t under scenario s. ω_t^s is a random variable with support in the finite set $\Omega = \Omega_1 \times \cdots \times \Omega_T$.

Using the same ideas for the constraints of the model, the following multistage stochastic programming problem will be considered:

$$
\begin{aligned}
\min \quad & C(x, y) \\
& A_k(x_0) - B_k(y_{0k}) \geq 0 \quad k = 1, \ldots, K \\
& \sum_j x_{ij} \leq r_i \quad i = 1, \ldots, R \\
& h_k(y_{0k}) - d_{0k} \geq 0 \quad k = 1, \ldots, K \qquad \text{(SW-B)} \\
& y_{0k} \in Y_{0k}, \; k = 1, \ldots, K, \quad x_0 \in X. \\
& A_k(x_{kt}^s) - B_k(y_{kt}^s) \geq 0 \quad k = 1, \ldots, K \quad \forall (s, t)
\end{aligned}
$$

$$\sum_j x^s_{ktij} \leq r_i \quad i = 1, \ldots, R \quad \forall (s,t)$$
$$h_k(y^s_{kt}) - d^{st}_k \geq 0 \quad k = 1, \ldots, K \quad \forall (s,t)$$
$$y^s_{kt} \in Y^s_{kt},\ k = 1, \ldots, K\ x^s_t \in X^s_t \quad \forall (s,t).$$

3 An Auction Mechanism

Solving (SW-B) using a conventional algorithm (e.g. Newton's method) is an unrealistic approach in the context of grid computing. Firstly, the problem will be too large for conventional algorithms that do not take advantage of the specific structure. Secondly, and more importantly, solving (SW-B) using a standard algorithm requires a central authority to have complete knowledge over the preferences and demand expectations of the grid agents. Thus for reasons of computational efficiency, and practical applicability, we propose to use an auction mechanism. The basic idea is that a central authority (the grid agent) selects a configuration of the resources, along with a price for each market agent. The prices and configuration are then announced to the agents. The market agents then announce their demands given the price information. The whole process is then repeated. The auction mechanism we chose to implement to adjust prices and demands is essentially the Mean Value Cross (MVC) decomposition algorithm of Holmberg and Kiwiel (2006). A similar approach was taken by Thomas et al. (2002), but they used Scarf's algorithm to achieve a similar result. The MVC decomposition algorithm is computationally efficient, easy to analyze and has a nice game theoretic interpretation.

We will use x, and y to denote the aggregate vectors for the grid agent's, and grid market's decision vectors, respectively. By π^s_{kt} we denote the dual variables associated with the

$$A_k(x^s_{kt}) - B_k(y^s_{kt}) \geq 0,$$

constraints. π will be used to denote the vector incorporating all the π^s_{kt}. The auction is initialized with four parameters $(\hat{\pi}, \hat{y}, \epsilon, \nu)$. $\hat{\pi}$ is the initial vector of prices that will be announced to each user. $\hat{y}$ is the starting aggregate demand announced by each user. ν is a positive scalar representing the maximum number of price announcements the market agent will make. Finally, ϵ is a positive scalar indicating an error tolerance. In this section we show that under appropriate conditions, and if ν is allowed to be large enough then ϵ can be taken to be arbitrarily small. Let $\delta(j)$ be any sequence such that:

$$\delta(j) \in (0, 1],\ \delta(j) \to 0,\ \sum_{j=0}^{\infty} \delta(j) = \infty.$$

The auction mechanism can be described as follows:

Step 0: Let $j = 1, U = \infty,\ L = -\infty$. Let $(\hat{\pi}, \hat{y}, \epsilon, \nu)$ be given.

Step 1: The grid agent announces the price $\hat{\pi}(j)^k$ to the kth market agent:

$$\hat{\pi}_k(j)^k = \delta_k(j)\pi_k(j) + (1-\delta_k(j))\hat{\pi}_k(j-1).$$

Step 2: Given the price information the market agents calculate their demands:[1]

$$\begin{aligned} y_k(j) \in \arg\max_y \quad & \psi_k(j) = c_k(y_{0k}) + \sum_{st} p_t^s c_k(y_{kt}^s, y_k^{a(s,t)}, \omega_t^s) \\ & \qquad - \sum_b \hat{\pi}(j)_{kb} B_b(y_{0k}) \\ & \qquad - \sum_{stb} \hat{\pi}(j)_{kb}^{st} B_b(y_{kt}^s) \\ \text{s.t.} \quad & h_k(y_{0k}) - d_{0k} \geq 0 \\ & h_k(y_{kt}^s) - d_{kt}^s \geq 0 \\ & y_{0k} \in Y_{0k},\ y_{kt}^s \in Y_{kt}^s. \end{aligned}$$

The market agents then announces $y_k(j)$ and $\psi_k(j)$ to the grid agent.

Step 3: Given the information above, the grid agent solves the following problem:

$$\begin{aligned} x(j) \in \arg\min_x \quad & \chi(j) = c_0(x^1) + \sum_{st} p_t^s c_0(x_t^s, x^{a(s,t)}, \xi_t^s) \\ \text{s.t.} \quad & A_k(x_0) - B_k(\hat{y}_{0k}(j)) \geq 0 \quad k = 1, \ldots, K \\ & A_k(x_t^s) - B_k(\hat{y}_{kt}^s(j)) \geq 0 \quad k = 1, \ldots, K\ \forall(s,t) \\ & x_0 \in X,\ x_t^s \in X_t^s,\ \forall(s,t), \\ & \sum_j x_{ij} \leq r_i \quad i = 1, \ldots, R \\ & \sum_j x_{tij}^s \leq r_i \quad i = 1, \ldots, R\ \forall(s,t) \end{aligned}$$

where

$$\hat{y}_{kt}^s(j) = \delta_k(j)y(j)^k + (1-\delta_k(j))\hat{y}(j-1)^k.$$

Let $\pi(j)$ denote the Lagrange multiplier vector of the problem above.

Step 4: The grid agent tests for convergence:

$$U^j = \chi(j) + \sum_k c_k(\hat{y}_k^1(j)) + \sum_{kst} p_t^s c_k(\hat{y}_{kt}^s(j), \hat{y}_k^{a(s,t)}(j), \omega_t^s), \quad (3.1)$$

$$L^j = \sum_k \psi_k(j) + X(j), \quad (3.2)$$

[1] Note that the market agents try to maximize their net profit.

where

$$\begin{aligned} X(j) = \min_x \quad & c_0(x_0) + \sum_{t=1}^{T}\sum_{s=1}^{S} p_t^s c_0(x_t^s, x^{a(s,t)}, \xi_t^s) \\ & - \sum kb\hat{\pi}_1^{kb}(j)A_{kb}(x_0) - \sum_{bkst} \hat{\pi}_{st}^{kb}(j)A_{kb}(x_t^s) \\ \text{s.t.} \quad & x_0 \in X,\ x_t^s \in X_t^s,\ \forall (s,t) \\ & \sum_j x_{ij} \le r_i \quad i = 1,\dots,R \\ & \sum_j x_{tij}^s \le r_i \quad i = 1,\dots,R \ \forall (s,t). \end{aligned}$$

If $U^j < U$ let $U = U^j$.
If $L^j > L$ let $L = L^j$. If $U - L < \epsilon$ or $j > \nu$ stop, else set $j := j + 1$ and go to Step 1.

We will follow Holmberg and Kiwiel (2006) in order to prove that the procedure described above does indeed converge to the social optimum. The argument is based on results from Belenky et al. (1976).

Definition 3.1 (*Belenky et al. 1976*) A convex game of K players is given by $G = \{Z_k, f_k\}_{k=1}^K$. Where Z_k and f_k are the strategy (feasible) set, and payoff function of the kth player, respectively. Let $Z = Z_1 \times \cdots \times Z_K$, then any choice $z \in Z$ is called a play of the game. The game is zero-sum if:

$$\sum_{k=1}^{K} f_k(z) = 0 \quad \forall z \in Z.$$

Let, $f_k(z_k; \hat{z}) = f_k(\hat{z}_1, \dots, z_k, \dots, \hat{z}_n)$ where $z_k \in Z_k$, and $\hat{z} \in Z$. The game will be called convex if Z is convex, compact, and f_k is concave with respect to z_k and convex with respect to $\hat{z}$.

Definition 3.2 (*Belenky et al. 1976*) By a game process we will mean the iterative play of a convex game given as follows:

$$z(j+1) = (1 - \delta(j))z(j) + \delta(j)u(j),$$

where $u(j) \in \Phi(z(j))$, and

$$\Phi(z(j)) \in \arg\max_z \sum f_k(z_k; z(j)).$$

Finally, $\delta(j)$ satisfies:

$$\delta(j) \in (0, 1],\ \delta(j) \to 0,\ \sum_{j=0}^{\infty} \delta(j) = \infty.$$

The main result that we will use to prove convergence of the auction is given in the following result. The proof can be found in Belenky et al. (1976, Theorem 10).

Theorem 3.3 *If G is a convex zero-sum game then the game process given in definition 3.2 is convergent to the set of equilibrium points of the game G.*

We will follow Holmberg and Kiwiel (2006) and place the auction procedure described above into the same framework as the game process in Theorem 3.3. Our analysis will make use of the following Slater type constraint qualification.

Assumption 3.4 For every $(\hat{y}_{0k}, \hat{y}_{kt}^s) \in Y_{0k} \times Y_{kt}^s$, satisfying:

$$h_k(y_{0k}) - d_{0k} \geq 0$$
$$h_k(y_{kt}^s) - d_k^{st} \geq 0.$$

There exists an $(\hat{x}_{0k}, \hat{x}_t^s) \in X \times X_t^s$ such that:

$$A_k(\hat{x}_0) - B_k(\hat{y}_{0k}) > 0$$
$$A_k(\hat{x}_t^s) - B_k(\hat{y}_{kt}^s) > 0.$$

For completeness we summarize our assumptions concerning the data of the problem. To simplify notation we will assume that the random variables appearing in (SW-B) are ν-dimensional.

Assumption 3.5 The functions defining the problem in (SW-B):

$$\begin{aligned}
&c_0 : \mathbb{R}^{RK} \to \mathbb{R} \\
&c_k : \mathbb{R}^{Q} \to \mathbb{R},\ k = 1, \ldots, K \\
&c_0 : \mathbb{R}^{RK} \times \mathbb{R}^{RK} \times \mathbb{R}^{\nu} \to \mathbb{R} \\
&c_k : \mathbb{R}^{Q} \times \mathbb{R}^{Q} \times \mathbb{R}^{\nu} \to \mathbb{R},\ k = 1, \ldots, K
\end{aligned}$$

are convex. Moreover:

$$\begin{aligned}
&A_k : \mathbb{R}^{RK} \to R^B,\ k = 1, \ldots, K \\
&B_k : \mathbb{R}^{Q} \to R^B,\ k = 1, \ldots, K,
\end{aligned}$$

are concave and convex, respectively. The sets X, Y_k^1, Y_{kt}^s, and X_t^s are convex and compact. Finally, the demand functions:

$$h_k : \mathbb{R}^{Q} \to R,\ k = 1, \ldots, K,$$

are concave.

Consider the following partial Lagrangian associated with (SW-B):

$$
\begin{aligned}
L(x, y, \pi) \triangleq c_0(x_0) - \sum k c_k(y_{0k}) + \sum_{st} p_t^s \Big[c_0(x_t^s, x^{a(s,t)}, \xi_t^s) \\
- \sum k c_k(y_{kt}^s, y_k^{a(s,t)}, \omega_t^s) \Big] - \sum_{kb} \pi_b^k \Big[A_{kb}(x_0) - B_{kb}(y_{0k})) \\
- \sum_{kbst} \pi_{bt}^{ks} (A_{kb}(x_t^s) - B_{kb}(y_{kt}^s) \Big].
\end{aligned}
$$

Note that the optimization performed in Step 2 and Step 4 can be combined into a single optimization problem as follows:

$$
\begin{aligned}
\min_y \Phi(y, \hat{\pi}) \triangleq \ & \sum_k M_k(y_k, \hat{\pi}_k) + N(\hat{\pi}) \\
\text{s.t.} \quad & (y_k^1, y_{kt}^s) \in Y_k^1 \times Y_{kt}^s,
\end{aligned}
\tag{3.3}
$$

where

$$
\begin{aligned}
M_k(y_k, \hat{\pi}_k) \ &\triangleq c_k(y_{0k}) + \sum_{st} p_t^s c_k(y_{kt}^s, y_k^{a(s,t)}, \omega_t^s) + \sum_b \hat{\pi}_b^k B_{kb}(y_k^1) \\
&+ \sum_{stb} \hat{\pi}_{kb}^{st} B(y_{kt}^s) \\
N(\hat{\pi}) \ &\triangleq \min_x \ c_0(x_0) + \sum_{st} p_t^s c_0(x_t^s, x^{a(s,t)}, \xi_t^s) - \sum_{kb} \pi_1^{kb} A_{kb}(x^0) \\
&- \sum_{kb} \pi_{st}^{kb} A_{kb}(x_t^s) \\
&\text{s.t.} \ \ x^1 \in X, \ \ x_t^s \in X_t^s.
\end{aligned}
$$

In order to improve readability we have assumed, with out loss of generality, that the constraints:

$$
\begin{aligned}
&\sum_j x_{ij} \le r_i \quad i = 1, \dots, R \\
&\sum_j x_{tij}^s \le r_i \quad i = 1, \dots, R \quad \forall (s, t),
\end{aligned}
$$

have been incorporated into X, and X_t^s, respectively.

Using Assumption 3.4 it can easily be seen that the optimization problem in Step 3 can be reformulated as follows:

$$\max_{\pi\geq 0} \Phi(\hat{y},\pi) \triangleq \left[\min_{x}\; c_0(x_0) + \sum_{st} p_t^s c_0(x_t^s, x^{a(s,t)}, \xi_t^s) \right.$$
$$\left. - \sum_{kb} \pi_1^{kb}(A_{kb}(x^1) - B_{kb}(\hat{y}_k^1)) - \sum_{kb} \pi_{st}^{kb}(A_{kb}(x_t^s) - B_{kb}(\hat{y}_{kt}^s)) \right]$$
$$+ \sum_{k} c_k(\hat{y}_{0k}) + \sum_{kst} p_t^s c_k(\hat{y}_{kt}^s, \hat{y}_k^{a(s,t)}, \omega_t^s). \tag{3.4}$$

Note that we added the constant terms from (3.1) in Step 4 of the algorithm in order to derive Eq. 3.4. In particular, in order to obtain (3.4) the partial Lagrangian of the problem in Step 3 was taken, with respect to the $A(x) - B(y) \geq 0$ constraints. The equality follows from the convexity of the functions involved.

We are now ready to state the main result of this section.

Theorem 3.6 *Let S^* denote the optimal value of the social welfare problem in* (SW-B). *Suppose that* 3.4 *and* 3.5 *are satisfied. Then:*

$$\lim_{j\to\infty} U^j = \lim_{j\to\infty} L^j = S^*.$$

Proof We will follow Holmberg and Kiwiel (2006) and formulate the auction mechanism as a game satisfying the conditions of Theorem 3.3. The result will then follow. In order to place the game in an appropriate form, note that we can view the model as having only two players. The first is the grid agent, and the second is the sum of all market agents. By π, and y denote the decisions of the grid and market agent(s), respectively.

The problem in (3.4) can be written as follows:

$$\mathcal{Z}_1(\hat{y}) = \max_{\pi\in\Pi}\; \Phi(\hat{y},\pi).$$

Note that we have used the fact that the problem satisfies the Slater condition of Assumption 3.4 in order to contain the prices in a compact set Π. Under our assumptions this can be done with no loss of generality. We can write (3.3) as follows:

$$\mathcal{Z}_2(\hat{\pi}) = \min_{y}\; \Phi(y,\hat{\pi}).$$

We now have:

$$\mathcal{Z}_1(\hat{y}) - \mathcal{Z}_2(\hat{\pi}) = \max_{y,\pi} \Phi(\hat{y},\pi) - \Phi(y,\hat{\pi}).$$

Where $\Phi(\cdot,\pi)$ is the objective function of the grid agent with y fixed. While, $\Phi(y,\cdot)$ represents the objective function of the sum of the market agent with π fixed. In terms of the notation of Theorem 3.3, the vector (y, π) corresponds to z, while Φ corresponds to $\mathcal{M}$.

Let $\Delta((\hat{y}, \hat{\pi}), (y, \pi)) = \Phi(\hat{y}, \pi) - \Phi(y, \hat{\pi})$ and $\mathcal{Z}(y, \pi) = \mathcal{Z}_1(y) - \mathcal{Z}_2(\pi)$. Using all these we can formulate the optimization problems solved during the auction as the following single optimization problem:

$$\mathcal{Z}(\hat{y}, \hat{\pi}) = \max_{y,\pi} \Delta((\hat{y}, \hat{\pi}), (y, \pi)).$$

It is straightforward to show that $\Delta((y, \pi), (\hat{y}, \hat{\pi}))$ is concave in $(\hat{y}, \hat{\pi})$ and convex in (y, π). Moreover $\Delta((y, \pi), (y, \pi)) = 0$. Note that at every iteration j of the auction the following optimization problem is solved:

$$\mathcal{M}(\hat{y}(j), \hat{\pi}(j)) \in \arg\max \Delta((y, \pi), (\hat{y}(j), \hat{\pi}(j))).$$

Finally, the $\delta(j)$'s used in the auction satisfy the conditions of the game process used in Theorem 3.3. We have thus formulated the auction mechanism as the game process in 3.3. The result now follows. □

4 Numerical Experiments

In this section we discuss the numerical performance of the proposed auction mechanism. The purpose of our numerical study was to empirically test how many iterations of the auction are required in order to approximate the social optimum. In order to perform our simulations we generated random instances of the model in (SW-B). For ease of implementation we used linear utility functions and constraints. Moreover, the model was assumed to have only two stages. The algorithm was implemented in C, and GLPK (2006) was used to solve the LP subproblems. All the results were obtained on a Linux machine with a 3Ghz processor, and 2Gb of RAM. In Table 1 we tabulate the data of the test problems. In Figure 1 we plot the error calculated as: $(U^j - L^j)/|SW^*|$; where SW^* denotes the optimal solution of the social welfare problem. The error as taken as the average of one hundred instances of the problems given in Table 1. We used $\delta_j = \alpha/(j + \beta)$, with $\alpha = 7$ and $\beta = 5$ for the updates required in Steps 2 and 3 of the algorithm.

As it can be seen from Fig. 1 the proposed method works well when an approximate answer is sought in a few iterations. The results show that if one wants to operate near the optimal value then a few iterations of the auction can achieve this goal. The results

Table 1 Problem data

Problem	No. of market agents	No. of blocks	No. of resources	No. of services	No. of scenarios
P1	5	10	5	10	70
P2	15	15	12	14	50
P3	20	25	100	40	10
P4	10	12	10	10	60

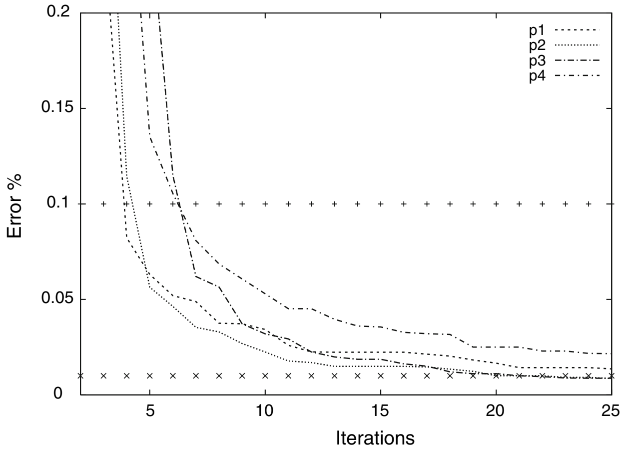

Fig. 1 Problems 1–4

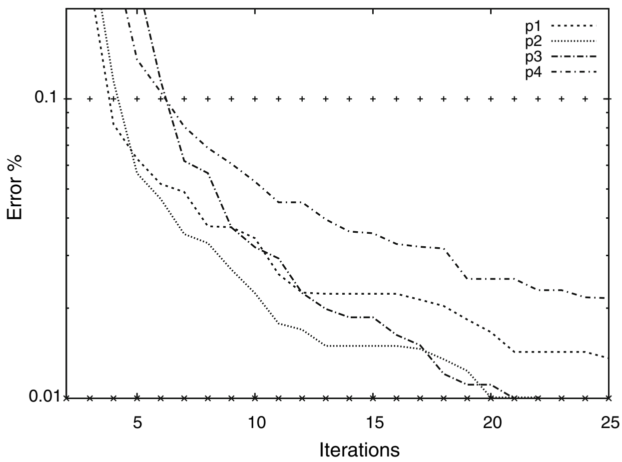

Fig. 2 Log plot of problems 1–4

also show that without further refinement of the method it may be too expensive when an exact answer is required. However, we see from the semi-log plot in Fig. 2 that the algorithm has a convergence rate that is less than linear. Thus the algorithm can be useful in obtaining a good solution after a few iterations, but if an exact answer is required then the algorithm becomes inefficient. Tuning the update parameter δ_j can help towards increasing the efficiency of the method. The efficient use of intelligent starting points is another direction that we plan to explore.

5 Conclusions

In this paper we proposed an auction mechanism for decentralized resource allocation in the context of grid computing. An auction mechanism based on a decomposition algorithm was proposed and its convergence established. The numerical results seem to indicate that the proposed approach is feasible. We emphasize that the focus of this paper is to propose a framework that can be used for managing resources in a grid computing environment. Even though research on this problem is at an embryonic stage we believe that some of its main features (e.g. the need for decentralized resource management) have been captured in the auction mechanism adumbrated in the previous sections. There are still many issues that will need to be addressed. For example, we have assumed the decisions can take real values. While this assumption enabled us to to use results from convex games to establish the convergence of the algorithm, it must be relaxed before the method can be applied to real world grid environments. One possibility is to integrate the proposed algorithm with a branch and bound framework. Another possibility is to use tools from semi-definite programming to find, potentially useful, relaxations of the large scale combinatorial problem. Another simplifying assumption we made concerned the existence of the A and B functions used to convert resources to blocks, and services to blocks, respectively. For the method to become useful the functional form of these functions will need to be identified. We believe that the correct modeling and efficient solution of the grid scheduling problem is extremely complicated. Some fundamental properties were incorporated into this paper but many issues are still open.

References

Belenky, V. Z., Ivankov, S. A., & Volkonsky, V. A. (1976). On a certain approach to the investigation of the convergence of iterative processes. In *Computing equilibria: How and why (Proc. Internat. Conf., Comput. Centre, Polish Acad. Sci., Toruń, 1974)* (pp 225–247). North-Holland, Amsterdam.

Birge, J. R., & Dempster, M. A. H. (1996). Stochastic programming approaches to stochastic scheduling. *Journal of Global Optimization, 9*(3–4), 417–451.

Birge, J. R., & Louveaux, F. (1997). *Introduction to stochastic programming*. Springer Series in Operations Research. New York: Springer-Verlag.

Distributed European infrastructure supercomputing applications. http://www.deisa.org. Project information available online.

Gnu linear programming kit, version 4.9, January 2006. http://www.gnu.org/software/glpk/

Gupta, A., Stahl, D. O., & Whinston, A. B. (1997). A stochastic equilibrium model of internet pricing. *Journal of Economic Dynamics & Control, 21*(4–5), 697–722.

Holmberg, K., & Kiwiel, K. C. (2006). Mean value cross decomposition for nonlinear convex problems. *Optimization Methods & Software, 21*(3), 401–417.

Papadimitriou, C. H. (2001). Algorithms, games, and the internet. In *STOC*.

Rockafellar, T., & Wets, R. (1991). Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research, 16*(1), 119–147.

Scarf, H. (1967) The approximation of fixed points of a continuous mapping. *SIAM Journal on Applied Mathematics, 15*, 1328–1343.

Scarf, H. (1973). *The computation of economic equilibria*. New Haven, Conn: Yale University Press. With the collaboration of Terje Hansen, Cowles Foundation Monograph, No. 24.

Stoenescu, T. M., & Teneketzis, D. (2002). A pricing methodology for resource allocation and routing in integrated-services networks with quality of service requirements. *Mathematical Methods of Operations Research, 56*(2), 151–167.

Sutherland, I. E. (1968). A futures market in computer time. *Communications of the ACM, 11*(6), 449–451.

The London e Science Centre. http://www.lesc.ic.ac.uk/. Project information available online.

Thomas, P., Teneketzis, D., & MacKie-Mason, J. K. (2002). A market-based approach to optimal resource allocation in integrated-services connection-oriented networks. *Operations Research, 50*(4), 603–616.