Appl Intell (2006) 25:147–158
DOI 10.1007/s10489-006-9651-8

Multi economic agent interaction for optimizing the aggregate utility of grid users in computational grid

Li Chunlin · Li Layuan

Abstract This paper investigates the interactions between agents representing grid users and the providers of grid resources to maximize the aggregate utilities of all grid users in computational grid. It proposes a price-based resource allocation model to achieve maximized utility of grid users and providers in computational grid. Existing distributed resource allocation schemes assume the resource provider to be capable of measuring user's resource demand, calculating and communicating price, none of which actually exists in reality. This paper addresses these challenges as follows. First, the grid user utility is defined as a function of the grid user's the resource units allocated. We formalize resource allocation using nonlinear optimization theory, which incorporates both grid resource capacity constraint and the job complete times. An optimal solution maximizes the aggregate utilities of all grid users. Second, this paper proposes a new optimization-based grid resource pricing algorithm for allocating resources to grid users while maximizing the revenue of grid providers. Simulation results show that our proposed algorithm is more efficient than compared allocation scheme.

Keywords Computational grid · Resource allocation · Agent · Optimization

1. Introduction

A grid computing environment is one in which applications can utilize multiple computational resources that may be distributed at widespread geographic locations [1–3]. Resource management and scheduling is a complex undertaking due to large-scale heterogeneity present in resources, management policies, users, and applications requirements in these environments [12, 19, 20]. The resource owners and end-users have different goals, objectives, strategies, and demand patterns. Grid resources are owned and managed by different organizations with different access policies and cost models that vary with time, users, and priorities. There are many grid users distributed in the grid, which will be competing for the use of the available grid resources. The issues that grid allocation mechanism should be addressed are as follows. (1) efficient grid resource allocation to the different grid users taking into account their different needs and performance requirements; (2) the crucial notion of fairness; (3) the ability to implement the allocation scheme in a distributed manner with minimal communication overheads; and (4) the issue of pricing the resources in such a way that the grid providers' revenue will be maximized if the grid users are allocated resources according to (1) and (2) above. This paper is target to solve above issues by using utility-based optimization scheme.

In the context of networks, pricing has been extensively used in the literature as a means to arbitrate resource allocation [13–15]. In order to adopt a price-based approach, utility functions are used to characterize the resource requirements and the degree of satisfaction of individual users. The goal of the network is to appropriately allocate resources to maximize an objective function that depends on user utilities. For example, the total aggregated utility over all users may be maximized (called the social welfare) subject to certain resource constraints. Both of Kelly's work [9, 11] and Low's work [10] are to maximize the aggregate user utilities over flow rate subject to capacity constraints. In order to eliminate the coupling of users through shared links,

L. Chunlin (✉) · L. Layuan
Department of Computer Science, Wuhan University of Technology, Wuhan 430063, P. R. China
e-mail: chunlin74@tom.com or jwtu@public.wh.hb.cn

system optimization is decomposed into subsidiary optimization problems for users and networks respectively, by using price per unit flow as a Lagrange multiplier that mediates between these two sub problems. The main difference between Kelly' s work [9, 11] and Low's work [10] is that they propose different mechanisms. Kelly allows the users to decide their payments and the network allocates the rate, while in Low's approach, users decide the rate and pay what the network charges. But unfortunately, the utility functions used in Kelly [9] and Low [10] are abstract utility function without concrete parameters, so all behaviors of utility function only are analyzed in qualitative sense, and can not conduct experiments to evaluate effects of utility function on system performance. Thus, although reasonable assumptions can be made on the behavior of utility functions, such an approach cannot be used to provide concrete numerical answers. Hence, This paper proposes to consider measurable characteristics to formulate utility functions. Two important attributes that affect grid task agents, cost and completion time are considered in this paper.

In this paper, we propose a price-based resource allocation model to achieve maximized aggregated utility of grid users. Our original contributions are: First, we propose to use measurable characteristics to formulate utility functions, rather than abstract utility function used in [9, 10]. The grid user utility is defined as a function of the grid user's the resource units allocated. The function value can be understood as the perceived quality, user satisfaction, etc. Second, we present a new optimization-based grid resource-pricing algorithm for allocating resources to grid users while maximizing the revenue of grid providers. Through simulation results, we show that our proposed algorithm is more efficient than conventional allocation scheme.

The remainder of the paper is structured as followings. Section 2 analyses related works. Section 3 describes grid agents in grid market. Section 4 presents grid resource allocation for joint grid user and grid provider optimization. Section 5 describes an iterative algorithm that computes the price and resource allocation. In section 6 the experiments are presented and discussed. Section 7 concludes the paper.

2. Background and related works

This section reviews the current status of different economic models for trading resources to manage resources based on a computing economy. To realize the full potential of Grid economy, the Gridbus project [4] has been developing technologies that provide end-to-end support for allocation of resources based on resource providers and consumers quality of service (QoS) requirements. One of the key components of Gridbus system is the Grid Market Directory (GMD). R. Wolski [5] investigates 'G-commerce' computational economies for controlling resource allocation in Computational Grid settings. He defines hypothetical resource and resource producers, then measure the efficiency of resource allocation under two different market conditions: commodities markets and auctions. In [13], Richard J. La et al investigate achieving the system optimal rates in the sense of maximizing aggregate utility in a communication network. This is done by decomposing the overall system problem into subproblems for the network and for the individual users by introducing a pricing scheme. The users are to solve the problem of maximizing individual net utility, which is the utility less the amount they pay. In [14] Yi Cui and Yuan Xue target the problem of optimal network resource allocation in overlay multicast. They propose a distributed algorithm, which maximizes the aggregate utility of all multicast members, subject to both network and data constraints. They then implement the algorithm in a series of protocols purely depending on the coordination of end hosts. In [15] Nan Feng et al considered a radio resource management problem with user centric and network centric objectives. They used a utility function as the user-centric metric and for the network-centric counterpart. They introduced an explicit pricing mechanism to mediate between the user-centric and network-centric resource management problems. Users adjusted their powers in a distributed fashion to maximize the difference between their utilities and their payments. Carsten Ernemann [17] addresses the idea of applying economic models to the scheduling task. In [17] a scheduling infrastructure and a market-economic method is presented. The efficiency of this approach in terms of response and wait time minimization as well as utilization is evaluated by simulations with real workload traces. In [18], H. Yaiche et al present a game theoretic framework for bandwidth allocation for elastic services in high-speed networks. The framework is based on the idea of the Nash bargaining solution from cooperative game theory, which not only provides the rate settings of users that are Pareto optimal from the point of view of the whole system. They conclude the pricing of elastic connections based on users' bandwidth requirements and users' budget. The bargaining framework can be used to characterize a rate allocation and a pricing policy that takes into account users' budget in a fair way and such that the total network revenue is maximized. Buyya [19, 20] have proposed and developed a distributed computational economy-based framework, called the Grid Architecture for Computational Economy (GRACE), for resource allocation and to regulate supply and demand of the available resources. This economic-based framework offers an incentive to resource owners for contributing and sharing resources; and motivates resource users to think about trade-offs between the processing time (e.g., deadline) and computational cost (e.g., budget), depending on their QoS requirements. In [21], Jonathan Bredin et al formulate the hosts' resource-allocation problem as a game with the players being agents competing for a resource from a common server. They show how to compute the unique positive Nash equilibrium explicitly under perfect information when there are two or more players. They develop an optimal agent bidding strategy that plans an agent's expenditure over multi-task itineraries. Their bidding strategy minimizes execution time while preserving a prespecified budget constraint. In [22], O. Ercetin et al. study the caching model in the framework of Content Delivery Networks. The objective is to minimize the user latency by intelligently distributing the content and serving the user requests from the most efficient surrogates. They use price-directed market based algorithms to achieve this goal. They model the agents with a self-maximizing behavior and define the problem as a non-cooperative game played among the publishers and the surrogates. In [30], the proposed model adopts three types of agents: grid resource agents, grid user agent and grid service agents. Grid service agents act as both a buyer of grid resources and a seller of grid services for grid users. Grid resource allocation optimization is distributed to two market levels: service market and resource market. Interactions between three agent types are mediated by two level market mechanisms. Compared with [30], this paper applies grid resource agents and grid user agents to negotiate with each other, the main objective is to maximize the utility of grid users and the revenue of resource providers.

The above economic models for network and grid computing fields can be improved both conceptually and computationally. Our target is to solve the problem of optimal resource allocation in computational grid. An optimal solution should maximize the aggregate utilities of all grid users, subject to various constraints, such as the grid resource capacity, job complete times. At same time, the grid provider adjusts the unit price in order to maximize its revenue, which is measured as the sum of the individual payments. The grid user utility is defined as a function of the grid user's the resource units allocated. It was also described how the agents can be assigned proper utility functions, with which they made a natural trade-off between money and resource. Furthermore, based on our theoretical framework, we propose an iterative algorithm that computes the price and resource allocation.

3. Grid agents in grid market

The grid market consists of two economic agent types: the grid resource agents that represent the economic interests of the underlying resources of the computational grid, the grid task agents that represent the interests of grid user using the grid to achieve goals. Grid market has information about the locations of current resource providers in the grid and their prices. Whenever a grid resource agent in the grid decides to sell its resources, change it's pricing structure, or update available capacity, it will spawn an agent to find grid markets and update the advertised information. The grid market then provides this information to other agents wishing to know about resource providers. Whenever a new grid task agent is created, it is first given an endowment of electronic cash to spend to complete its task. If that agent either refuses to make a purchase under that level of availability or that price structure, or if the task agent does not purchase all of the available capacity, the resource agent offers the remaining capacity to the next task agent [6–8]. We assume that when a task agent purchases a portion of the resource, it is guaranteed that the task agent continues to receive resource uninterrupted from the resource agent until its task is completed. The price that the agent pays, per second of resource capacity, is the same for as long as he continues to use the purchased resource. The agent makes no guarantee to the resource provider and may leave the queue or leave the processor at any time. The user makes this decision by keeping up-to-date on the resources and prices offered by other resource providers on the grid. This can be done by periodically spawning agents that travel to grid markets and return with price and resource quotes. A grid resource agentis used at the source node in the grid and is deployed at the entry node. The Grid resource agents have varied computational resource capacity, and the computational resource capacity is shared among the grid task agents. The grid resource agents charge the task agents for the portion of the computational resource capacity occupies. We assume that the grid resource agents of a grid does not cooperate, due to high messaging and processing overheads associated with cooperative allocating. Instead, they act non-cooperatively with the objective of maximizing their individual profits [16, 21, 22]. The grid resource agents compete among each other to serve the task agents. The task agents do not collaborate either, and try to purchase as much computational resource as possible with the objective of maximizing their net benefit. The agents communicate by means of a simple set of signals that encapsulate offers, bids, commitments, and payments for resources. We couple the resources and payments with the offers and requests respectively. This reduces the number of steps involved in a transaction (committing agents to their payments and offers ahead of the market outcome), and so increases the speed of the system's decision making. To enforce these rules the interactions between the two agent types are mediated by means of market mechanisms. In our market mechanisms, agent communication is restricted to setting a price on a single unit of a known grid resource. Therefore, agents set their prices solely on the basis of their implicit perception of supply and demand of grid resource at a given time. When a resource is scarce, grid task agents have to increase the prices they are willing to buy, just as resource agents decrease the price at which they are willing to offer the resource. In our model, agents perceive supply and demand in the market through resource pricing algorithm that will be described in Section 5.

Grid resource agents sell the underlying resources of the grid. A grid task agent that represents the grid user makes buying decisions within budget constraints to acquire computation resources. The offers placed on the grid market by grid resource agents are allocated to grid task agent. Grid task agents buy computation resources solely on the basis of the most recent price information they have. Grid task agents initiate compete for grid resource by signaling that they wish to buy resources to complete certain tasks. The task agent retains a vector of prices that it is willing to pay for resources [24, 25, 29, 31]. The task agenttries to maintain its resources at a level optimal units that is discovered through gradient climbing adaptation to the behavior of the market described in Section 4. The price paid for each resource agent should be as low as possible without failing to obtain the resource. Therefore the task agent makes a request for each resource that it needs separately. If a request was rejected, the agent increases the price it will send to resource agent at the next negotiation. If a request was accepted, the agent reduces the price it pays for that resource in subsequent negotiations.

Grid task agents and grid resource agents do not communicate directly with one another or among themselves. All interactions are by the means of grid market. The grid market also broadcasts the prices at which trades are agreed, so the agents have more information upon which to base their trading behavior. The negotiation between agents is mediated by means of a grid market. It allows multiple grid task agents and grid resource agents to negotiate simultaneously, it provides a dense set of market price information and it allows supply and demand to be reconciled at the same time. Grid markets provides a means to complete institutionally mediated bargaining in one shot that would take an indeterminate time using iterated market allocation algorithms. The grid markets use price-directed allocation algorithm that will be described in Section 4. In this algorithm an initial set of prices is announced to the task agent. In each iteration, grid resource tasks individually determine their optimal allocation and communicate their results to the grid resource agent. Grid resource agents then update their prices and communicate the new prices to the task agents and the cycle repeats. Prices are then iteratively changed to accommodate the demands for resources until the total demand equals to the total amount of resources available. The task agent's utility maximization is also considered.

4. Grid resource pricing and allocation optimization

In this section, we set up the mathematical models for optimal grid resource allocation and pricing based on the framework developed in the above section.

First gives notations to be used in the following sections:

- c_j: capacity of grid resource j represented by grid resource agent j
- x_i^j: resource units allocated to task agent i by grid resource agent j
- t_i^j: the time taken by the i-th grid task agent to complete j-th job, the measurement unit is second
- T_i: time limits given by the i-th grid task agent to complete all jobs
- u_i^j: money paid to the grid resource agent j by grid task agent i, the measurement unit is grid dollars
- q_{ij}: is the size of $i$ts grid task agent's jth job
- p_j: the price of the resource unit in resource agent j.
- E_i: the endowment given to a task agent i

4.1. Problem formulation

Suppose c_j is the capacity in computational units of jth grid resource agent, which were shared by grid users. Let x_i^j be the resource units allocated to task agent i by resource agent j. We assume that grid users receive a utility equal to $U(x_i^j)$ if the allocated grid resource unit is x_i^j. Given complete knowledge and centralized control of the system, a natural problem for the grid market to try to solve is maximizing the aggregated utility function. We now formulate the problem of optimal resource allocation in computational grid as the following constrained non-linear [13–15]:

$$(\text{Grid}) \quad \begin{aligned} &\text{Max} \sum_i U\left(x_i^j\right) \\ &\text{s.t. } c_j \geq \sum_i x_i^j \end{aligned} \tag{4.1}$$

The constraint implies that the aggregate resource units do not exceed the total capacity of grid resource. By non-linear optimization theory, there exists a maximizing value of argument x_i^j for the above optimization problem (4.1), and we can apply the Lagrangian method to solve such a problem. Let us consider the Lagrangian form of this optimization problem:

$$\begin{aligned} L\left(x_i^j;\lambda\right) &= \sum_i U\left(x_i^j\right) - \lambda\left(\sum_i x_i^j - c_j\right) \\ &= \sum_i \left(U\left(x_i^j\right) - \lambda x_i^j\right) + \lambda c_j \end{aligned} \tag{4.2}$$

where λ is Lagrangian multiplier. Hence, at a maximum of L over x_i^j the following conditions hold: $U'(x_i^j) = \lambda$ Thus, given that the grid knows the utility functions U of all the grid task agents, this optimization problem can be mathematically tractable through the above procedure. However, in practice, it is not likely to know all the U, and it is also infeasible for computational grid to compute and allocate resource in a centralized fashion. Solving the objective function (4.1) requires global coordination of all grid users, which is impractical in distributed environment such as the computational grid. In order to achieve a distributed solution, we decompose the problem into the following two problems GU (4.3) and GP (4.4), seek a distributed solution where the grid provider does not need to know the utility functions of individual grid user. Suppose that grid task agent i may choose an amount to pay per unit time, u_i^j, and receives grid resource allocation x_i^j proportional to u_i^j, the relation can be represented as

$$x_i^j = \frac{u_i^j}{p_j}$$

where p_j is regarded as a charge per unit resource for grid task agent i. For a completed time, the task agent optimization problem (GU) can be written as (4.3).

$$\text{(GU)} \quad \begin{array}{l} \text{Max} U\left(u_i^j\right) \\ \text{s.t. } T_i \geq \sum_j t_i^j \end{array} \tag{4.3}$$

Constraint is a completed time constraint, which says that the aggregate sum of all times of each task agent cannot exceed its total time limits. T_i is time limits given to a grid task agent. GU objective is to choose optimal u_i^j. The grid resource provider, given the amounts that the grid task agents are willing to pay, $u = (u_1, u_2 \dots u_n)$, attempts to maximize the function $\sum u_i^j \log(x_i^j + 1)$. So the Grid resource provider's optimization problem can be formulated as follows:

$$\text{(GP)} \quad \begin{array}{l} \text{Max} \sum u_i^j \log\left(x_i^j + 1\right) \\ \text{s.t. } c_j \geq \sum_i x_i^j \end{array} \tag{4.4}$$

4.2. Optimal grid resource allocation

Grid task agents want to complete a set of jobs in a given sequence by purchasing resources from grid resource agents located throughout the grid. An agent begins with an endowment of E_i to spend to complete its task and wishes to minimize the total time taken to complete a sequence of jobs given its budget constraint. We assume that there are K types of resources and that each agent may needs many types to complete a job. Assume that there is a set K $= \{1, 2, \dots K\}$ of different types of resources that the grid allocates at each grid task agent in order to complete the task. For example, if computing tools, databases, are the two types of resources that the grid allocates in order to complete the task, then $K = \{1,2\}$. In this case, $k = 1$ refers to storage systems and $k = 2$ refers to databases. The agent's task can be represented as the sequence $\{q_{ij}\}_{j=1}^{j=K}$, where q_{ij} is the size of ith task agent's jth job.

Let u_i^j be the price paid to jth resource agent per time unit by the ith task agent. Let u_i be the total investment of the ith task agent, which is defined in (4.5). N grid task agents compete for grid resources with finite capacity. The resource is allocated using a market mechanism, where the partitions depend on the relative payments sent by the grid task agents. We assume that each task agent submits u_i^j to the grid resource agent. Then, $u^j = [u_1^j \dots u_N^j]$ represents all payments of grid task agents for jth resource agent. Let p_j denote the price of the unit computational resource in resource agent j. Let the pricing policy, $p = (p_1, p_2, \dots, p_n)$, denote the set of unit computational resource prices of all the resource agents in the grid.

$$u_i = \sum_j u_i^j \tag{4.5}$$

Let x_i^j be the fraction of resource units allocated to task agent i by resource agent j. If ith task agent's payment in the jth resource agent is u_i^j, then the total computation resource units allocated to task agent i is

$$x_i^j = \frac{u_i^j}{p_j}. \tag{4.6}$$

The i-th agent receives resources proportional to its payment relative to the sum of the resource agent's revenue, c_j is the capacity in computational units of jth grid resource agent. u_i^j is the amount that the ith agent pays for resource j, r_i^j is the capacity that i-th agent pays receives.

$$r_i^j = c_j \frac{u_i^j}{p_j} \tag{4.7}$$

Grid task agent needs to complete a sequence of jobs in a specified amount of time, T_i, while minimizing the cost accrued. The goal of each task agent is to complete its job as quickly as possible when spending the least possible amounts of money. q_{ij} is the size of ith task agent's jth job. c_j is the capacity in computational units of jth grid resource agent. The time taken by the ith agent to complete its job is:

$$t_i^j = \frac{q_{ij}}{c_j x_i^j}$$

The utility function $U(u_i^j)$ of the grid task agent is defined as (4.8).

$$U\left(u_i^j\right) = -\sum_j u_i^j - K\left(\sum_{j=1}^{N} \frac{q_{ij} p_j}{c_j u_i^j} - T_i\right) \tag{4.8}$$

Where u_i^j is the money which task agent i paid to the resource agent j, K is the relative importance of costs and times to complete grid task, an agent with larger value of K would indicate a greater preference to reduce its completion time. When K is 1, meaning that costs and times are equally important. For a completed time, the task agent optimization problem (GU) can be written as (4.9).

$$\text{(GU)} \quad \begin{array}{l} \text{Max}\left(-\sum_j u_i^j - K\left(\sum_{j=1}^{N} \frac{q_{ij} p_j}{c_j u_i^j} - T_i\right)\right) \\ \text{s.t. } T_i \geq \sum_j t_i^j \end{array} \tag{4.9}$$

Grid system optimization is distributed to two subproblems (4.3) and (4.4): optimization of task agent and resource agent. The following Theorem 1 is to prove that (4.1) can be solved by two subproblems optimization. u_i^{j*} is the solution to the task agent optimization problem (4.3). x_i^{j*} is the solution to grid resource provider's optimization problem (4.4).

Theorem 1. *There exist x_i^{j*}, u_i^{j*}, p_j such that* (1) u_i^{j*} *solves GU (p_j);* (2) x_i^{j*} *solves GP (u_i^j)*

Proof:

$$\text{For} \quad \text{(GU)} \quad \begin{array}{l} \text{Max} U\left(u_i^j\right) \\ \text{s.t. } T_i \geq \sum_j t_i^j \end{array}$$

We take derivative and second derivative of $U(u_i^j)$ with respect to u_i^j:

$$\begin{aligned} U'\left(u_i^j\right) &= \frac{dU\left(u_i^j\right)}{du_i^j} = K\sum_{j=1}^{N} \frac{q_{ij} p_j}{c_j \left(u_i^j\right)^2} - 1 \\ U''\left(u_i^j\right) &= \frac{d^2 U\left(u_i^j\right)}{d\left(u_i^j\right)^2} = -K\sum_{j=1}^{N} \frac{p_j q_{ij}}{c_j \left(u_i^j\right)^3} \end{aligned} \tag{4.10}$$

$U''(u_i^j) < 0$ is negative due to $0 < u_i^j$. The extreme point is the unique value minimizing the agent's cost under completed time limits. The Lagrangian for the task agent's utility is $L(u)$ (4.11).

$$L\left(u_i^j\right) = -\sum_j u_i^j - K\left(\sum_{j=1}^{N} \frac{q_{ij} p_j}{c_j u_i^j} - T_i\right) + \lambda\left(\sum_j t_i^j\right) \tag{4.11}$$

where λ is the Lagrangian constant. From Karush-Kuhn-Tucker Theorem we know that the optimal solution is given $\partial L(u)/\partial u = 0$ for $\lambda > 0$.

$$\partial L\left(u_i^j\right)\Big/\partial_{u_i^j} = -1 + K\frac{q_{ij} p_j}{c_j \left(u_i^j\right)^2} - \lambda\frac{q_{ij} p_j}{c_j \left(u_i^j\right)^2} \tag{4.12}$$

Let $\partial L(u_i^j)\big/\partial_{u_i^j} = 0$ to obtain (4.13)

$$u_i^j = \left(\frac{(K-\lambda) q_{ij} p_j}{c_j}\right)^{1/2} \tag{4.13}$$

Using this result in the constraint equation, we can determine $\theta = K - \lambda$ as

$$(\theta)^{-1/2} = \frac{T_i}{\sum_{k=1}^{N} \left(\frac{p_k q_{ik}}{c_k}\right)^{1/2}} \tag{4.14}$$

We substitute (4.13) into (4.14) to obtain (4.15)

$$u_i^{j*} = \left(\frac{q_{ij} p_j}{c_j}\right)^{1/2} \frac{\sum_{k=1}^{N} \left(\frac{q_{ik} p_k}{c_k}\right)^{1/2}}{T_i} \tag{4.15}$$

u_i^{j*} is the unique optimal solution to the optimization problem (GU).

$$\text{For (GP)} \quad \begin{array}{l} \text{Max} \sum u_i^j \log\left(x_i^j + 1\right) \\ \text{s.t. } cj \geq \sum_i x_i^j \end{array} \tag{4.16}$$

We take derivative and second derivative with respect to x_i^j:

$$U'\left(x_i^j\right) = u_i^j \Big/ x_i^j + 1 \quad U''\left(x_i^j\right) = -u_i^j \Big/ \left(x_i^j + 1\right)^2 \tag{4.17}$$

$U''(x_i^j) < 0$ is negative due to $0 < x_i^j$. The extreme point is the unique value maximizing the revenue of grid resource provider. The Lagrangian for *GP* problem is $L(x)$

$$L\left(x_i^j\right) = \sum u_i^j \log\left(x_i^j + 1\right) + \lambda\left(c_j - \sum_i x_i^j\right) \tag{4.18}$$

$$= \sum\left(u_i^j \log\left(x_i^j + 1\right) - \lambda x_i^j\right) + \lambda c_j$$

Where λ is the Lagrangian constant. From Karush-Kuhn-Tucker Theorem we know that the optimal solution is given $\partial L(x)/\partial x = 0$ for $\lambda > 0$.

$$\partial L(x_i^j)\Big/\partial x_i^j = u_i^j\Big/x_i^j + 1 - \lambda \tag{4.19}$$

Let $\partial L(x)/\partial x = 0$ to obtain (4.20)

$$x_i^j = u_i^{j-\lambda}\Big/\lambda \tag{4.20}$$

Using this result in the constraint equation, we can determine λ as

$$c_j = \frac{1}{\lambda}\sum_{k=1}^{n} u_k^j - n \tag{4.21}$$

$$\lambda = \frac{\sum_{k=1}^{n} u_k^j}{c_j + n} \tag{4.22}$$

We substitute (4.22) into (4.20) to obtain (4.23)

$$x_i^{j*} = \frac{u_i^j(c_j + n)}{\sum_{k=1}^{n} u_k^i} - 1 \tag{4.23}$$

x_i^{j*} is the unique optimal solution to the optimization problem (GP). □

5. Grid resource pricing scheme

The decomposition of *Grid* problem (4.1) into *GU* (4.3) and *GP* (4.4) problems suggests that solving *Grid* problem can be achieved by solving *GU* (4.3) and *GP* (4.4) problems via an iterative algorithm. In each iteration, the grid userindividually solves its fees to pay (4.9), adjusts its grid resource demand and notifies the grid about this change. After the new grid resource demand is observed by the grid resource agent, it updates its price accordingly and communicates the new prices to the grid task agent, and the cycle repeats. To illustrate how grid task agentadjusts its fees to pay, we define the demand function $D(p) : R \to R$, which is defined as the quantity of resource that the agent would desire if the price is p. $D(p)$ can be obtained by optimal solution u_i^{j*} to *GU* (4.3) problem.

$$D(p) = u_i^{j*}\Big/p_j$$

The iterative algorithm that computes the price and resource allocation is then given as follows.

Algorithm 1 Grid resource unit price calculation and resource allocation

This algorithm is consistent with the law of supply and demand: if the demand for grid resource exceeds the capacity supply C_j, then the price $p_j^{(n+1)}$ is raised; otherwise, the price is reduced.

Grid resource agent j at iteration n
Receives grid computation demand x_i^j from grid task agents;
If $c_i \geq \sum_j x_i^j$
Then
  $pc_j^{(n+1)} = \max\{\varepsilon, pc_j^{(n)} + \eta(x^j pc_j^{(n)} - c_j)\}$; // Computes a new price
  // $x^j = \sum_i x_i^j$, $\eta > 0$ is a small step size parameter, n is iteration number.
  Return new price $pc_j^{(n+1)}$ to all grid task agents;
Else Return Null;

Grid task agent i at iteration n
Receives from the grid resource agent j the price pc_j;
$u_i^{j*} = \mathrm{Max}\{U(u_i^{j*})\}$; // calculates u_i^{j*} to maximize $U(u_i^j)$
If $E_i \geq \sum_j u_i^j + \sum_k v_i^k$
Then $x_i^j(n+1) = u_i^{j*(n)}\Big/pc_j^{(n)}$; //Calculates its optimal computation resource demand $x_i^j(n+1)$
Return $x_i^{j(n+1)}$ to grid resource agents;
Else Return Null;

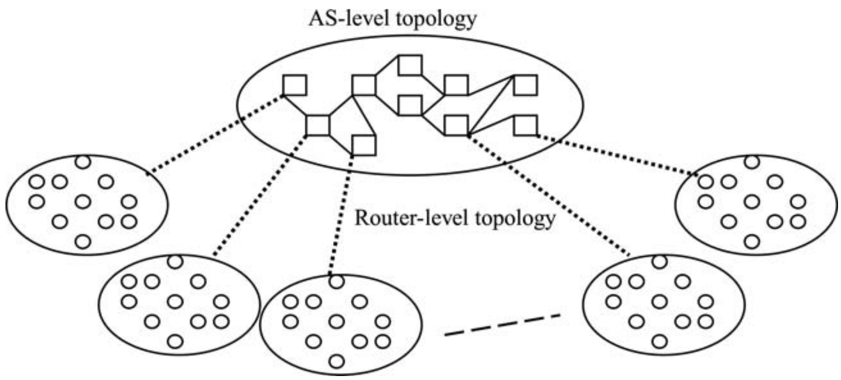

Fig. 1 Experimental Topology

6. Experiments

6.1. Experiments settings

Simulation studies were carried out to evaluate the performance of proposed grid resource pricing and allocation algorithm. We provide performance analysis comparing our pricing algorithm and R. Buyya proposed deadline and budget constrained scheduling algorithm [26, 27]. We provide simulation results using an ns-2 simulator. Grid task agents have a utility function defined as $U(u_i^j) = -\sum_j u_i^j - (\sum_{j=1}^{N} \frac{q_{ij} p_j}{c_j u_i^j} - T_i)$, which is a simple version of Eq. (4.8) where $K = 1$. The resource agent updates its price per unit every 200 msec. The resource agent forwards the price to task agents, the resource price is put in a packet. Whenever the new price packet passes to task agent, the task agent calculates the utility. According to allocation policy, if the price becomes higher than its maximum willingness to pay, task agent does not buy grid resource. The task agent can be informed the price for the next iteration by the next price packets. We use the BRITE generator [28] to setup network topology (Fig. 1). We first generate an AS-level topology consisting of 10 nodes. Each node in the AS-level topology generates a router-level topology of 50 nodes. Therefore, the size of our experimental network is 500 nodes. The bandwidths of all links are uniformly distributed between 1 and 10 Mbps. In this experiment there were resources with the different price and capability. The grid resources

are shared among task agents. Each task agent has an associated time limit, before which it should finish its job. Processor capacity can be expressed as MIPS (Million Instructions Per Second). The resource costcan be expressed in grid dollar that can be defined as processing cost per MIPS. Processor capacity varies from 220 to 580 MIPS. The initial value of the task price denoted by p varies from 10 to 500 grid dollars. All parameters are summarized in Table 1. In our experiments the following parameters will be varied: task agent's budget denoted by B, the completed time limit denoted by T, and the size of grid nodes denoted by S. The descriptions of task agents are listed in Table 2. There are eight types of task agents with different budget and time limits and task size. During the time of experiment, grid resource requests are generated by the grid user agents. After this initial period, the number of tasks that is statistically expected to be generated during an interval of 100 time units is considered in the result. To allow grid task agents to complete tasks, an additional margin of 300 time units is provided. Each measurement is run 6 times with different seeds. These experimental configurations are to bring up performance of resource allocation algorithm as many as possible. Completion times and resource allocation efficiency are two measurement criteria to measure in the experiment. Completion times measure the time observed by the grid task agent to access the requested grid resources and complete the task. It

Table 1 Experiment parameters

Num of cluster	Processor capacity (MIPS)	Initial price (grid dollar)
1	370	300–500
2	220	10–100
3	370–380	200–500
4	510–580	100–500
5	340–390	20–200
6	270–275	10–300
7	410–475	100–400
8	220	20–300
9	370–380	200–500
10	270–275	10–300

Table 2 The description of the task agents

Types of task agent	Time limits (T)(ms)	Budget (B) (grid dollar)	Task size (Kb)
1	100	1000	10
2	200	500	30
3	300	100	20
4	400	500	50
5	100	500	50
6	200	1000	60
7	300	500	40
8	400	100	30

is influenced by the size of the grid and resource capacity, budget, and processing delay. Processing delay means the time elapsed during the task was processed.Resource allocation efficiency indicates the ratio of accepted grid resource requests to all sent grid resource requests. It is influenced by resource capacity, budget, time limit and the size of the grid.

6.2. Comparison results

In this subsection we compare deadline and budget constrained (DBC) scheduling algorithm, which was presented by R. Buyya in [26, 27] and non-market algorithm Round-Robin scheduling algorithm with our method to investigate performance of our joint optimization pricing method. The experiments are to study characteristics of algorithm in terms of task completion time and resource allocation efficiency. Resource allocation based on a deadline and budget constraint-scheduling algorithm (DBC) is intent to complete the task as quickly as possible, within the budget available. A description of the core of the algorithm follows:

1 For each resource, calculate the next completion time for an assigned job, taking into account previously assigned jobs.
2 Sort resources by next completion time.
3 Assign one job to the first resource for which the cost per job is less than or equal to the remaining budget per job.
4 Repeat all steps until all jobs are assigned.

Graphs corresponding to our method are labeled as "optimal", deadline and budget constrained scheduling is denoted as "DBC", and Round-Robinscheduling algorithm is denoted as "Robin".

The value of system load expresses the extent to which the whole system is busy. If in a certain period of time the number of jobs submitted to the system is small and the lengths of jobs are short, then the system load is light; otherwise, the system load is heavy. System load influences the performance of scheduling inherently. Some experiments are done under different system loads to investigate the performance of the algorithms. First two experiments are to measure effect of system load on completion times and allocation efficiency respectively. Load factor vary from 0.05 to 0.9. It can be seen from Figs. 2 and 3 that price based strategy has better resource allocation efficiency and use less time to complete tasks

when compared to the Robin and DBC strategy especially at higher loads. Before load factor reach 0.5, price based schemes and DBC perform well. When load factor reach 0.5, the completion time of Round-Robin increase sharply. After load factor reach 0.5, the completion time using price-directed allocation can be as much as 15% shorter than that

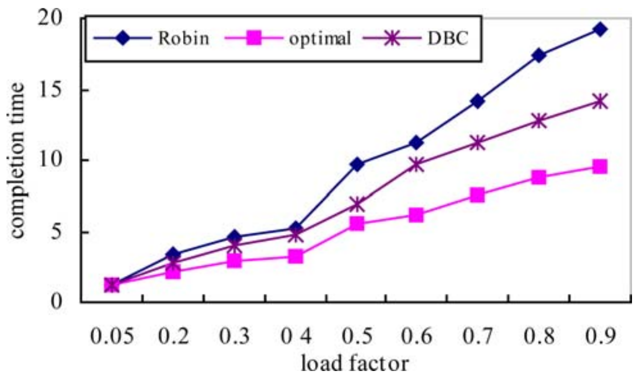

Fig. 2 Completion time comparison

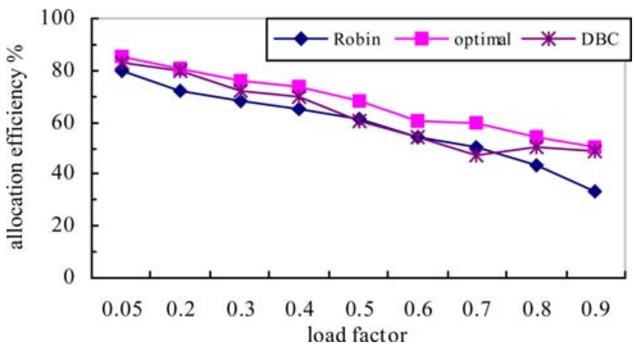

Fig. 3 Resource allocation efficiency comparison

using the DBC schedule. The reason is that at low loads, the task entering the grid is less than grid resource available. In such case, the task can be accepted and executed at the same as it is submitted. However, at higher loads, priced based strategy selects the best available resource for a task, which in this case is the least loaded and therefore the fastest. It helps acquiring higher grid resource utility and revenues. Round-Robin performs worst because resources are allocated arbitrarily.

The following four experiments are to measure effect of network latency on completion times and allocation efficiency under different grid size respectively. Network latency refers to the time elapsed between the sending of a message to a router and the return of that message. For the completion time criteria in general, lower network latency leads to faster completion times. The X-axis shows a change in network latency. From the results in Fig. 4, under small size grid ($S = 150$), After network latency reach 0.005,the completion time using price based allocation can be as much as 8% shorter than that using the DBC allocation. When changing the size of grid by $S = 500$ (Fig. 5), Round-Robin allocation takes more time to allocate appropriate resources, the completion time is as much as 30% longer than that using the price based allocation. This effect could be caused by an increasing network latency leads to longer times to complete tasks; as the times get shorter with decreasing grid size, the overall processing gets faster. Considering the resource

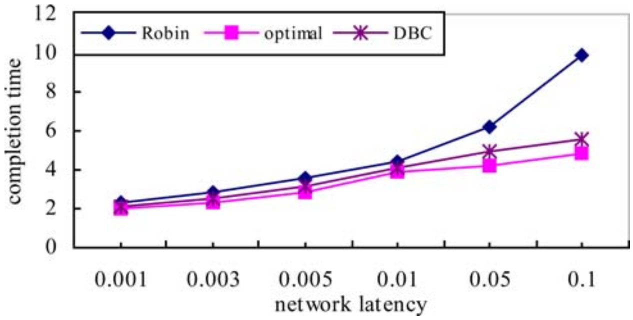

Fig. 4 Network latency effect on completion time under $S = 150$

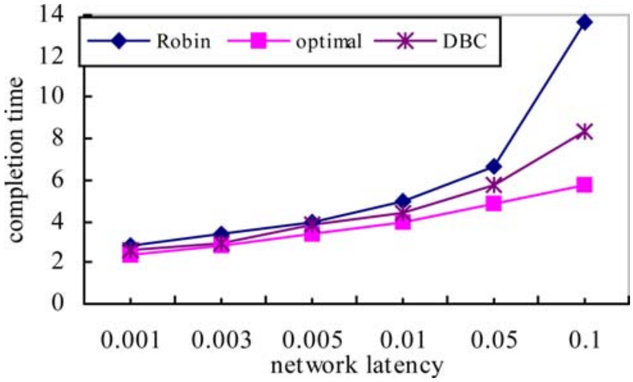

Fig. 5 Network latency effect on completion time under $S = 500$

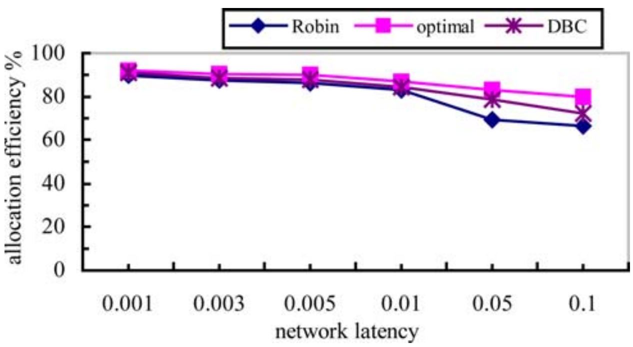

Fig. 6 Network latency effect on allocation efficiency under $S = 150$

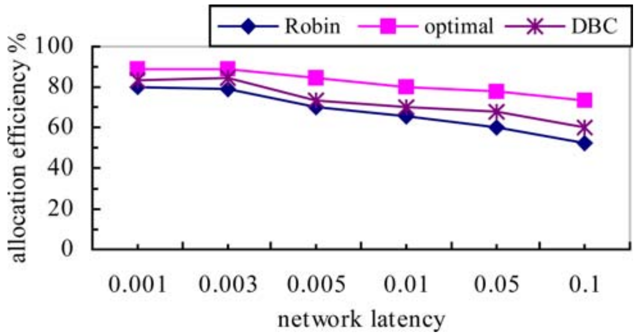

Fig. 7 Network latency effect on allocation efficiency under $S = 500$

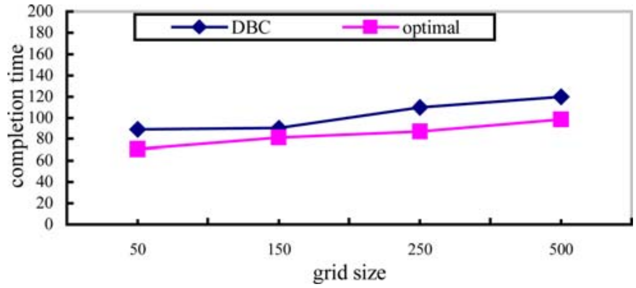

Fig. 8 Completion time under $B = 1000$, $T = 100$

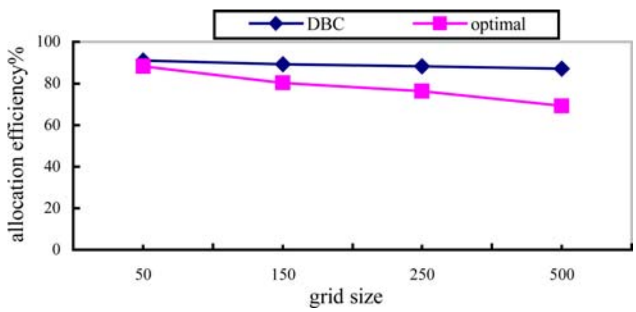

Fig. 9 Allocation efficiency under $B = 1000$, $T = 100$

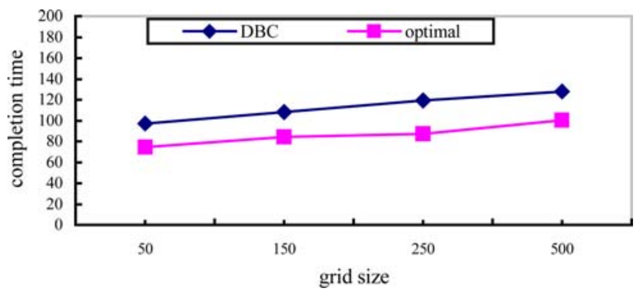

Fig. 10 Completion time under $B = 500$, $T = 400$

allocation efficiency, from the results in Fig. 6 and Fig. 7, under small size grid ($S = 150$), the resource allocation efficiency of price-directed allocation as much as 5% larger that using the DBC allocation. When increasing the size of grid by $S = 500$, the resource allocation efficiency of Round-Robin allocation is as much as 17% less than that using the price-directed allocation. When network latency reach 0.1, the resource allocation efficiency reduce to nearly 25% in all scenarios.

The last four experiments are to measure the effect of different combinations of budget and time limits constraint on completion times and allocation efficiency under different grid size respectively. The X-axis shows changes in grid size values. Firstly, considering the completion time, as shown in Fig. 8, Fig. 10, both strategies spend more time to complete tasks when grid size increases. For both strategies, smaller grid size leads to faster completion times. From the results in Fig. 8, budget and time limits constraint are set by $B = 1000$ and $T = 100$ respectively, this represents large budget and low time limit. Under small size grid ($S = 100$), completion time of two method is near, when the size of grid increases ($S = 500$), the completion time using price method can be as much as 18% shorter than that using DBC. When changing

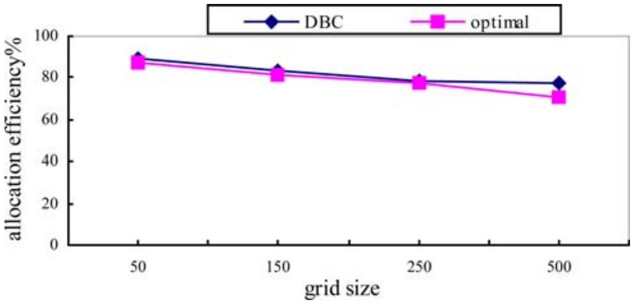

Fig. 11 Allocation efficiency under $B = 500$, $T = 400$

budget and time limits constraint by $B = 500$ and $T = 400$ (Fig.10), DBC based scheme takes more time to allocate appropriate resources, the completion time is as much as 18% longer than that using price based method. This is because DBC scheme wants to minimize the cost of the usage of grid resources within the budget limit, so they take longer times to complete tasks. Considering the resource allocation efficiency, In Fig. 9, for large budget and low time limit ($B = 1000$, $T = 100$), under small size grid ($S = 50$), the resource allocation efficiency of price based scheme as much as 10% more than that using DBC scheme. In Fig. 11, the resource allocation efficiency of price based scheme is nearly close to DBC scheme.

7. Conclusions

This paper targets the problem of optimal resource allocation in computational grid. The grid market adopts two economic agent types: the grid resource agents that represent the economic interests of the underlying resources of the computational grid, the grid task agents that represent the interests of grid user using the grid to achieve goals. The grid system optimization is decomposed to two subproblems: joint optimization of resource user and resource provider in grid market. Grid optimization problem can be formulated into utility optimization problem based on dynamic programming. This paper proposes a new optimization-based grid resource pricing algorithm for allocating resources to grid users while maximizing the revenue of grid providers. Simulation results show that our proposed algorithm is more efficient than compared allocation scheme.

Acknowledgments The authors thank anonymous reviewers for their useful comments and suggestions. The work is supported by National Science Foundation of China under grants (No. 60402028), Wuhan Younger Dawning Foundation under grants (No. 20045006071-15).

References

1. Foster I, Kesselman C (1999) The Grid: Blueprint for a new computing infrastructure. Morgan Kaufmann
2. Chapin S, Karpovich J, Grimshaw A (1999) The legion resource management system. In: Proc. of the 5th Workshop on Job Scheduling Strategies for Parallel Processing, Springer Verlag, pp 162–178
3. H. Casanova, G. Obertelli, F. Berman et al (2000) The AppLeS parameter sweep template: User-level middleware for the grid. In: Proc. of Super Computing, Springer Verlag, pp 75–76
4. The Gridbus Project: http://www.gridbus.org
5. Wolski R, Plank J, Brevik J, Bryan T (2001) Analyzing market-based resource allocation strategies for the computational grid. Intern J High-performance Comp Appl, Sage Publications 15(3):258–281
6. Chunlin Li, Layuan Li (2004) Agent framework to support computational grid. J Syst Software, Elsevier 70(1–2):177–187
7. Chunlin Li, Zhengding Lu, Layuan Li, Shuzhi Zhang (2002) A mobile agent platform based On tuple space coordination. J Adv Enging Software, Elsevier 33(4):215–225
8. Chunlin Li, Zhengding Lu, Layuan Li (2003) Design and implementation of a hybrid agent platform. Program Comp Soft, Kluwer 29(1):28–42
9. Kelly F, Maulloo A, Tan D (1998) Rate control for communication networks: shadow prices, proportional fairness and stability. J Operational Res Soc 49(3):237–252
10. Low SH, Lapsley DE (1999) Optimization flow control, I: Basic algorithm and convergence. IEEE/ACM Trans Network 7(6):861–874
11. Kelly FP (1997) Charging and rate control for elastic traffic. Europ Trans Telecomm 8:33–37
12. Gomoluch J, Schroeder M (2003) Market-based resource allocation for grid computing: A model and simulation. 1st International Workshop on Middleware for Grid Computing (MGC2003), Rio de Janeiro, Brazil
13. La RJ, Anantharam V (2002) Utility-based rate control in the internet for elastic traffic. IEEE/ACM Trans. on Network 10(2):272–286
14. Cui Yi, Xue Yuan, Nahrstedt K (2003) Optimal resource allocation in overlay multicast. In: Proc. of IEEE ICNP
15. Feng N, Mau S-C, Mandayam NB (2002) Pricing and power control for joint network—centric and user-centric radio resource management. CISS'2002, Princeton NJ
16. Chunlin L, Layuan L (2004) The use of economic agents under price driven mechanism in grid resource management. J Syst Archit, Elsevier 50(9):521–535
17. Ernemann C (2002) Economic scheduling in grid computing. In: Proc of 8th International Workshop Job Scheduling Strategies for Parallel Processing, Lecture Notes in Computer Science, UK, pp 128–152
18. Yaiche H, Mazumdar R, Rosenberg C (2000) A game theoretic framework for bandwidth allocation and pricing in broadband networks. IEEE/ACM Trans Network l8(5):667–678
19. Buyya R, Abramson D, Giddy J (2000) Nimrod/G: An architecture for a resource management and scheduling system in a global computational grid. HPC Asia 2000, Beijing, China. IEEE Computer Society Press, USA
20. Buyya R, Giddy J, Abramson D (2001) A case for economy grid architecture for service-oriented grid computing. In: 10th IEEE International Heterogeneous Computing Workshop (HCW 2001), San Francisco, California, USA
21. Bredin J, Maheswaran RT, Imer C (2000) A game-theoretic formulation of multi-agent resource allocation. In: Proceedings of the Fourth International Conference on Autonomous Agents, Barcelona
22. Ercetin O, Tassiulas L (2003) Market based resource allocation for content delivery in the internet. IEEE Trans. on Computers 52(12)
23. Layuan L, Chunlin L (2003) A distributed QoS-Aware multicast routing protocol. Acta Informatica. Springer-Verlag, Heidelberg 40(3):211–233

24. Chunlin L, Layuan L (2004) Competitive proportional resource allocation policy for computational grid. Future Generation Computer Systems, Elsevier 20(6):1041–1054
25. Chunlin L, Zhengding L, Layuan L (2003) Apply market mechanism to agent-based grid resource management. Intern J Software Enging & Knowl Enging. World Scientific Publishing 13(3):327–340
26. Buyya R, Murshed R, Abramson D (2002) A deadline and budget constrained cost-time optimization algorithm for scheduling task farming applications on global grids. In: Int. Conf. on Parallel and Distributed Processing Techniques and Applications, Las Vegas, NV, USA
27. Buyya R, Murshed M (2002) GridSim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing. The Journal of Concurrency and Computation: Practice and Experience (CCPE). Wiley Press
28. http://www.cs.bu.edu/brite
29. Chunlin Li, Layuan Li (2003) Applying agents to build grid service management. J Network & Computer Appl Elsevier 26(4):323–340
30. Chunlin Li, Layuan Li (2005) A utility-based two level market solution for optimal resource allocation. In: Computational Grid, ICPP'05, Norway. IEEE Computer Society Press, pp 23–30
31. Chunlin Li, Layuan Li (2005) A distributed utility-based two level market solution for optimal resource scheduling. In: Computational Grid. Parallel Computing, Elsevier, USA vol. 31 No. 3–4, pp 332–351

Li Chunlin received the ME in computer science from Wuhan Transportation University in 2000, and PhD degree in Computer Software and Theory from Huazhong University of Science and Technology in 2003. She now is an associate professor of Computer Science in Wuhan University of Technology. Her research interests include computational grid, distributed computing and mobile agent. She has published over 15 papers in international journals.

Li Layuan received the BE degree in Communication Engineering from Harbin Institute of Military Engineering, China in 1970 and the ME degree in Communication and Electrical Systems from Huazhong University of Science and Technology, China in 1982. Since 1982, he has been with the Wuhan University of Technology, China, where he is currently a Professor and PhD tutor of Computer Science, and Editor in Chief of the Journal of WUT. He is Director of International Society of High-Technol and Paper Reviewer of IEEE INFOCOM, ICCC and ISRSDC. His research interests include high speed computer networks, protocol engineering and image processing. Professor Li has published over 150 technical papers and is the author of six books. He also was awarded the National Special Prize by the Chinese Government in 1993.