Bull. Nov. Comp. Center, Comp. Science, 25 (2006), 13–24

Equilibrium prices in economy of the GRID resource allocation: Part one

S.V. Bredikhin, A.B. Khutoretsky

Abstract. In this paper, we consider Computational Grid as market of the two commodities: CPU time and disc storage. Market agents are suppliers and users (consumers). The suggested market model refines the model proposed by R. Wolski et al. For the refined model, we discuss the possible application of the price-adjustment mechanism elaborated by J. Ma and F. Nie.

1. Introduction

We consider a computational Grid (Grid thereinafter). We assume that users pay for the Grid resources. These payments should cover, as a minimum, the maintenance and development costs. The more is resource's utility for the user, the more is the user's willingness to pay. Therefore, suppliers will have incentive to increase supply of more useful resources. Prices generate fair rules of access to the Grid resources. Equilibrium prices equalize demand and supply, thus making it possible to obtain a feasible resource allocation as a result of decentralized decision making by market agents [1].

Under this approach, the resource management system (RMS) should be driven primarily by the users' tendency to a maximum utility and suppliers' tendency to a maximum profit, while technical criteria (as average residence time) become essentially less important. Hence, the minimal "market" role of RMS is in creation of infrastructure that would allow both suppliers and users (to be exact, their representatives program-brokers) to "find each other" and agree on volumes and prices of transactions. Such an approach is proposed, for example, in [2].

Many authors give the RMS a more active role. They assume that program-broker, that represents a market agent, "knows" his budget restriction and preferences on a set of resource bundles. Interacting with program-brokers, the RMS can compute, if possible, the equilibrium prices. Then "brokers" detect demand and supply at these prices. If mutually disjoint optimal resource bundles can be chosen for all market agents (in a decentralized way or with participation of the RMS), then an equilibrium allocation will be attained.

Such a market model of CPU time and disk storage allocation is proposed in [3, 4]. Here we analyze and modify this model. Next, based on [5], we propose an approach to calculate prices and discuss some related problems.

2. Initial model

Let us briefly describe the market model proposed in [3, 4]. Each consumer expresses his demand in the form of "jobs". A job description specifies size and occupancy duration for each required resource. Each consumer is endowed by a budget, which is periodically renewed. However, "money" not spent during the current period cannot be used to pay for resources in the next period. These conditions stimulate full spending the current budget and using the system resources for jobs of minor importance at the end of the budget period, instead of saving money and priority execution of the most important jobs.

When a user wishes to purchase resources for a job, he indicates the required size of each commodity, but not the duration. At the time a supplier agrees to sell, he fixes a price that will be charged to the user until the job completes [3, p. 757]. Such a forward contract with an indefinite term of the resource supply at a fixed price seems unrealistic.

Instead of the users' utility functions, the authors suggest the following algorithm to calculate an individual demand function at the current price vector $\boldsymbol{p}(t)$.

At each time, the consumer has a queue of non-started jobs. At the beginning of the unit period $[t, t+1)$, he calculates $C_1 = \big(\sum_i w_i p_i(t)\big)/t$ and $C_2 = I(t)/(T_b - t)$, where w_i is the total amount of the resource i used by the consumer from some initial moment 0, $p_i(t)$ is a current price of the resource i, $I(t)$ is a current budget of the consumer, and T_b is the nearest time of budget replenishment.

Here C_1 is average expenditures (per time unit) that the consumer would have in the period t if he always purchased resources at current prices; C_2 expresses maximal expenditures per time unit, which are possible for the user in the period $[t, T_b)$ at uniform money expending. If $C_1 > C_2$, then demand is equal to zero. Otherwise, the user "demands as many jobs from the queue as he can afford" [3, p. 759]. The question arises: whether such a demand function reflects the rational user's preferences?

Let us assume that there is only one resource in the market and the first job in the queue requires 10 units of this resource. Let $I(1) = 30$ and $T_b = 3$. Assume that the user has purchased 10 units of the resource during the period $[0, 1)$. Then, at time 1 at price 2, we have $C_1 = 20 > C_2 = 15$, the demand is zero. After this $I(2) = 30$, and the user has purchased the same 10 units of the resource during the period $[0, 2)$. At moment 2 at price 3, we have $C_1 = 15 < C_2 = 30$, so the job should be included in the demand. Hence, the user prefers the execution of the job later and at a higher price. Such a preference pattern does not seem to be highly rational.

There are two commodities in the market studied in [3, 4]: the CPU time and disk storage. A memory supplier has a certain number of "files" of a fixed size for sale; the following algorithm describes its supply function. At the moment t, each supplier calculates his average income from one file per time unit during the period $[0, t)$: $C = R/(tS)$, where R is the total revenue obtained by this supplier in the same period. If the current price is smaller than C, then the supply is zero. Otherwise, a supply in the period $[t, t + 1)$ includes all the files intended for sale.

The unit of CPU time is “slot”, a certain number of CPU cycles, the processor speed share. Each CPU supplier owns one processor and agrees to sell to the Grid no more then N slots. To determine supply at the given price and time, each CPU calculates $C = R/(tN)$, where R is defined as above. If the current price is less than C, then the supply is zero. Otherwise supply in the period $[t, t + 1)$ includes all N slots.

With such definitions of supply, it is easy to construct situations in which a supplier behaves irrationally: prefers to sell later and at a smaller price.

3. A modified model

Let us modify the original market model to eliminate some of its drawbacks noted above. We consider the period $[1, T]$, where T is sufficiently large. The unit period $[\tau - 1, \tau)$ will be called “moment τ”.

3.1. Agents and goods. There are $I = I_U + I_C + I_M$ agents on the market: users indexed by $1, \ldots, I_U$; the CPU time suppliers indexed by $I_U + 1, \ldots, I_U + I_C$; and the disk storage suppliers indexed by $I_U + I_C + 1, \ldots, I$. Let us assign the numbers $1, \ldots, J = I_C + I_M$ to resources, assuming that a resource j is owned by the supplier $j + I_U$. The supplier s ($I_U < s \le I$) owns $V(s)$ units of the corresponding resource at each moment. In particular, for suppliers of CPU time, this means that processors have, generally speaking, different clock rates and, hence, provide different amounts of slots per time unit.

It is convenient for further reasoning to “individualize” the resource units: let us assign the numbers $k \in \{1, \ldots, V(s)\}$ to units of the resource owned by the supplier s; “the unit resource (j, k, τ)” is abbreviation for “unit k of the resource j at the moment τ”. Now any collection A of resources with timing can be represented by a characteristic vector

$$\boldsymbol{x} = \boldsymbol{x}(A) = \big(x_{jk}(\tau) \mid 1 \le j \le J,\ 1 \le k \le V(j + I_U),\ 1 \le \tau \le T\big),$$

where $x_{jk}(\tau) = 1$ if the unit resource (j, k, τ) is included into A, and $x_{jk}(\tau) = 0$, otherwise. Let $\boldsymbol{x}$ represent the set A in the above sense, and let $D^R_{j\tau}(\boldsymbol{x})$ be the amount of the resource j at the moment τ included into A. Then

$$D^R_{j\tau}(\boldsymbol{x}) = \sum_{k=1}^{V(j+I_U)} x_{jk}(\tau).$$

3.2. Suppliers. When allocating a resource unit per unit period to a user, the supplier s bears the maintenance costs $c_s > 0$. He chooses a vector $\boldsymbol{q}^s = (q^s_\tau)^T_{\tau=1}$ (q^s_τ is a supply at the moment τ) and a number m_s (demand for money); $q^s_\tau \leq V(s)$. A utility function of the supplier s describes his profit in the period under consideration:

$$U_s(\boldsymbol{q}^s, m_s) = m_s - c_s \sum_\tau q^s_\tau. \tag{1}$$

At the prices $\boldsymbol{p} = (p_j(\tau) \mid 1 \leq j \leq J,\ 1 \leq \tau \leq T)$, the supplier s has the budget restriction

$$m_s \leq \sum_\tau p_j(\tau) q^s_\tau, \tag{2}$$

where $j = s - I_U$: the amount received cannot exceed the total revenue from the resource sales.

A supplier's demand set is defined as follows. Suppose that, when offering q^s_τ resource units at the moment τ to users, the supplier s thus presents the demand for the remaining $V(s) - q^s_\tau$ units. The binary vector $\boldsymbol{x}$ of appropriate dimensionality is a characteristic vector for a resource bundle of the supplier s, if and only if for all τ

$$\begin{aligned} &D^R_{j\tau}(\boldsymbol{x}) \in [0, V(s)] \quad \text{if } j = s - I_U; \\ &D^R_{j\tau}(\boldsymbol{x}) = 0 \quad \text{otherwise.} \end{aligned} \tag{3}$$

(the supplier demands only his own resource). The demand set X_s of the supplier s is a set of all vectors satisfying (3).

3.3. Consumers (users). The user i wishes to execute one job (with number i). The job i requires K_i units of the CPU time (slots) to be completed. Also, it requires M_i memory units (files) at each moment from start to completion. The set of resource units used by a job does not change during each unit period, but the job can use different resource units at different unit periods. The change of resources happens, if necessary, on the border of a unit period immediately and free of charge. At each moment, the job can use files from different suppliers, but only one processor.

The user i assigns the sum B_i (budget) for execution of his job. He chooses a resource bundle (which corresponds to some characteristic vector $\boldsymbol{x}^i$) and a number m_i (demand for money). Zero vector signifies the refusal to execute the job. The vector $\boldsymbol{x}^i$ specifies the demand $D^C_\tau(\boldsymbol{x}^i)$ for the CPU time and the demand $D^M_\tau(\boldsymbol{x}^i)$ for disc storage at each time τ:

$$D^C_\tau(\boldsymbol{x}^i) = \sum_{j=1}^{I_C} D^R_{j\tau}(\boldsymbol{x}^i); \qquad D^M_\tau(\boldsymbol{x}^i) = \sum_{j=I_C+1}^{J} D^R_{j\tau}(\boldsymbol{x}^i).$$

The job i is executed at the moment τ according to the plan $\boldsymbol{x}^i$ if it uses a processor: $D^C_\tau(\boldsymbol{x}^i) > 0$. Let us describe the set X_i of all vectors $\boldsymbol{x}^i$ corresponding to the resource bundles admissible for the user i. The moments of start and completion of the job i according to the plan $\boldsymbol{x}^i$ will be denoted by $t^0(\boldsymbol{x}^i)$ and $t^1(\boldsymbol{x}^i)$, respectively; $t^1(\boldsymbol{x}^i) = +\infty$ means that the job i could not be completed in the period $[0, T]$.

1. If $\sum_\tau D^C_\tau(\boldsymbol{x}^i) = 0$ (the job is not executed in the period $[0, T]$), then $\boldsymbol{x}^i \in X_i$; set $t^0(\boldsymbol{x}^i) = t^1(\boldsymbol{x}^i) = +\infty$.

2. Assume that $\sum_\tau D^C_\tau(\boldsymbol{x}^i) > 0$. Then the value $t^0(\boldsymbol{x}^i) = \min\{\tau \mid D^C_\tau(\boldsymbol{x}^i) > 0\}$ is determined.

2.1. If there are no $t \in [t^0(\boldsymbol{x}^i), T]$ such that

$$\sum_{\tau=t^0(\boldsymbol{x}^i)}^{t} D^C_\tau(\boldsymbol{x}^i) \geq K_i; \tag{4}$$

$$D^M_\tau(\boldsymbol{x}^i) \geq M_i, \quad \forall\, \tau \in [t^0(\boldsymbol{x}^i), t]; \tag{5}$$

$$\begin{gathered} D^R_{j\tau}(\boldsymbol{x}^i) > 0 \to D^R_{k\tau}(\boldsymbol{x}^i) = 0, \quad \forall j \in \{1, \dots, I_C\}, \\ \forall k \in \{1, \dots, I_C\} \setminus \{j\}, \quad \forall \tau \in [t^0(\boldsymbol{x}^i), t], \end{gathered} \tag{6}$$

then $\boldsymbol{x}^i \notin X_i$; set $t^1(\boldsymbol{x}^i) = +\infty$. Condition (4) means that the job receives sufficient CPU time in the period $(t^0(\boldsymbol{x}^i) - 1, t]$. Condition (5) requires that the job have enough memory for this period. Condition (6) forbids the job using more than one processor at each moment $\tau \in [t^0(\boldsymbol{x}^i), t]$.

2.2. There exist $t \in [t^0(\boldsymbol{x}^i), T]$ satisfying (4)–(6). Then $\boldsymbol{x}^i \in X_i$; set $t^1(\boldsymbol{x}^i)$ equal to the least one of such t.

The utility function of the user i is quasilinear. It has the form

$$U_i(\boldsymbol{x}^i, m_i) = u_i(\boldsymbol{x}^i) + m_i, \tag{7}$$

where $u_i(\boldsymbol{x}^i) = g_i(t^1(\boldsymbol{x}^i))$ and $g_i(\tau)$ is a pecuniary valuation of utility for the user the job completion at the time τ. Let us assume that the function $g_i(\tau)$ is monotonically decreasing and tends to zero as $\tau \to +\infty$.

At the prices $\boldsymbol{p} = (p_j(\tau) \mid 1 \leq j \leq J,\ 1 \leq \tau \leq T)$, the user i has the budget restriction

$$m_i + \sum_{j,\tau} D^R_{j\tau}(\boldsymbol{x}^i) p_j(\tau) \leq B_i. \tag{8}$$

3.4. Equilibria. The supplier s at the prices $\boldsymbol{p}$ maximizes function (1) subject to conditions (2) and $0 \le q^s_\tau \le V(s)$ for all τ. Let $j = s - I_U$. Constraint (2) is active at optimum: $m_s = \sum_\tau p_j(\tau) q^s_\tau$. So, the supplier's problem takes the form

$$\sum_\tau (p_j(\tau) - c_s) q^s_\tau \to \max \quad s.t. \quad 0 \le q^s_\tau \le V(s), \ \ \forall \tau. \tag{9}$$

As described in Subsection 3.2, each vector $\boldsymbol{q}^s$ represents a resource bundle of the supplier s. Some characteristic vector $\boldsymbol{x}^s \in X_s$, in turn, corresponds to this resource bundle (see Subsection 3.1). Then

$$\begin{aligned}\sum_\tau (p_j(\tau) - c_s) q^s_\tau &= \sum_\tau (p_j(\tau) - c_s)(V(s) - D^R_{j\tau}(\boldsymbol{x}^s)) \\ &= C + u_s(\boldsymbol{x}^s) - \sum_\tau p_j(\tau) D^R_{j\tau}(\boldsymbol{x}^s),\end{aligned}$$

where the constant $C = \sum_\tau (p_j(\tau) - c_s) V(s)$ and $u_s(\boldsymbol{x}^s) = c_s \sum_\tau D^R_{j\tau}(\boldsymbol{x}^s)$. Therefore, (9) is equivalent to the following problem:

$$\begin{gathered} v_s(\boldsymbol{x}^s, \boldsymbol{p}) = u_s(\boldsymbol{x}^s) - \sum_\tau p_j(\tau) D^R_{j\tau}(\boldsymbol{x}^s) \to \max \\ s.t. \quad 0 \le D^R_{j\tau}(\boldsymbol{x}^s) \le V(s), \ \ \forall \tau. \end{gathered} \tag{10}$$

Let $D_s(\boldsymbol{p})$ denote a set of all solutions to this problem. Then $\boldsymbol{p} \mapsto D_s(\boldsymbol{p})$ is a supply correspondence for the supplier $s \in \{I_U + 1, \dots, I\}$.

The user i at the prices $\boldsymbol{p}$ solves the problem of maximizing function (7) at conditions (8) and $\boldsymbol{x}^i \in X_i$. Budget restriction (8) should be active in any optimal solution. The substitution $m_i = B_i - \sum_{j,\tau} D^R_{j\tau}(\boldsymbol{x}^i) p_j(\tau)$ in (7) after constants elimination yields the following problem:

$$\begin{gathered} v_i(\boldsymbol{x}^i, \boldsymbol{p}) = u_i(\boldsymbol{x}^i) - \sum_{j,\tau} D^R_{j\tau}(\boldsymbol{x}^i) p_j(\tau) \to \max \\ s.t. \quad \boldsymbol{x}^i \in X_i. \end{gathered} \tag{11}$$

Let $D_i(\boldsymbol{p})$ denote a set of all solutions to this problem. Then $\boldsymbol{p} \mapsto D_i(\boldsymbol{p})$ is a demand correspondence for the user $i \in \{1, \dots, I_U\}$.

Set $\boldsymbol{e} = (1, \dots, 1)$. A pair $(\boldsymbol{x}, \boldsymbol{p})$ is an equilibrium if

$$\boldsymbol{x} = (\boldsymbol{x}^i)_1^I \in D_1(\boldsymbol{p}) \times \dots \times D_I(\boldsymbol{p}) \quad \text{and} \quad \textstyle\sum_i \boldsymbol{x}^i = \boldsymbol{e}.$$

In this case, $\boldsymbol{x}$ is an equilibrium resources distribution and $\boldsymbol{p}$ is an equilibrium price vector.

In other words, specification of a demand vector for each market agent gives an equilibrium distribution if it is feasible (each resource unit at each moment is allocated to just one agent, either to the user or to the resource owner) and provides each agent with a maximum utility within his budget restriction at some (equilibrium) prices.

4. The price adjustment mechanisms

The best-known model of detecting equilibrium prices is the Walras–Samuelson process [6, p. 2]. It assumes the existence of a certain central player, the “Walrasian auctioneer” [7, p. 746]. Market agents, knowing the current prices vector $\boldsymbol{p}$, inform the auctioneer about their demand / supply vectors. The auctioneer calculates the vector $\boldsymbol{E}(\boldsymbol{p})$ of an aggregated excess demand (demand minus supply). He increases the price of the commodity i if $E_i(\boldsymbol{p}) > 0$ (the demand is greater than the supply) and decreases it if $E_i(\boldsymbol{p}) < 0$. It is a *tâtonnement* mechanism: transactions take place only at equilibrium prices (when $E_i(\boldsymbol{p}) = 0$).

4.1. The Euler scheme. A version of this process in continuous time is described by the Euler vector differential equation

$$\dot{\boldsymbol{p}} = \boldsymbol{E}(\boldsymbol{p}). \tag{12}$$

Let $\boldsymbol{p}(t)$ be a solution to (12). The limit $\boldsymbol{p}(t)$ at $t \to +\infty$ exists and gives the equilibrium prices (see [8, 9]) if the function $\boldsymbol{E}$ satisfies the condition of “gross substitutes”:

$$\frac{\partial E_i}{\partial p_i} < 0, \quad \forall\, i; \qquad \frac{\partial E_i}{\partial p_j} > 0, \quad \text{for } i \neq j. \tag{13}$$

Condition (13) means that the markets of different commodities are tightly related: the price increase on the market immediately causes a money outflow from this market, and some part of this money comes to the market of each commodity $j \neq i$. If (13) is not satisfied, then process (12), generally speaking, does not converge [10]. There is an essential complementarity of resources in the considered model (see Section 3), so (13) is not indeed fulfilled.

4.2. S. Smale’s scheme and its use in [3, 4]. A more universal mechanism is given by Smale’s scheme

$$\boldsymbol{J}_{\boldsymbol{E}}(\boldsymbol{p}) \cdot \dot{\boldsymbol{p}} = -\lambda \boldsymbol{E}(\boldsymbol{p}), \tag{14}$$

where $\boldsymbol{J}_{\boldsymbol{E}}(\boldsymbol{p})$ is Jacobian of the vector-function $\boldsymbol{E}(\boldsymbol{p})$. If an excess demand function $\boldsymbol{E}$ is twice continuously differentiable and satisfies some additional technical conditions, then process (14) converges to equilibrium prices [13]. This mechanism is *tâtonnement* as well. The discrete version of (14)

$$\boldsymbol{J}_{\boldsymbol{E}}(\boldsymbol{p}(t)) \cdot \Delta \boldsymbol{p}(t) = -\boldsymbol{E}(\boldsymbol{p}(t)) \tag{15}$$

is used in [3, 4]. The authors propose to calculate an excess demand at a moment t (on the basis of the current prices), and then find the prices for the next moment by solving equations (15). But a demand function in their model is defined in such a way (see Section 2), that an excess demand at each moment should depend on prices and volumes of the preceding transactions. Hence, some resources were distributed and used outside the equilibrium. This means that some non-*tâtonnement* mechanism but not Smale's one is realized.

Also, it should be noted, that the approach suggested in [3, p. 760]to approximation of derivatives of the function $\boldsymbol{E}$ with respect to prices in (15) seems unconvincing, and we do not know a satisfactory solution to this problem.

4.3. J. Ma and F. Nie's mechanism. In [5], the model of the market of indivisible goods is studied under the following assumptions: there is exactly one (indivisible) unit of each commodity in the market and the market agents' utility functions are quasilinear. For economics, the authors offer a pricing mechanism (*MN-mechanism*), which generalizes the discrete form of the Euler scheme (12). It consists in the following.

Let us arbitrarily fix a number $b > 0$ and divide the segment $[0, b]$ into n equal intervals of the length $\Delta_n = b/n$ by points $t_0 = 0, t_1, \ldots, t_n = b$ (the partition uniformity is not significant; we suppose it for simplicity). Let us also fix an arbitrary price vector $\boldsymbol{p}_0 \geq \boldsymbol{0}$ and set $\boldsymbol{p}(0) = \boldsymbol{p}_0$. For all $l \in \{1, \ldots, n\}$, we calculate

$$\boldsymbol{p}(t_l) = \boldsymbol{p}(t_{l-1}) + \Delta_n[\boldsymbol{f}(\boldsymbol{p}(t_{l-1})) - \boldsymbol{e}], \tag{16}$$

where $\boldsymbol{f}(\boldsymbol{p}) = \sum_i \boldsymbol{f}_i(\boldsymbol{p})$, $\boldsymbol{f}_i(\boldsymbol{p})$ is any convex linear combination of optimal resource vectors of the agent i at prices $\boldsymbol{p}$, and all components of the vector $\boldsymbol{e}$ are equal to unity. For a market of the considered type, the term $\boldsymbol{f}(\boldsymbol{p}) - \boldsymbol{e}$ generalizes an excess demand at the prices $\boldsymbol{p}$.

After completing calculations by formula (16), we obtain the vector-function $\boldsymbol{p}_n(t)$ defined at the points t_l. The following results were proved in [5]:

1. A sequence of the functions $\boldsymbol{p}_n(t)$ uniformly converges to a certain function $\boldsymbol{p}^*(t)$ defined on the interval $[0, b]$.
2. At any choice of $\boldsymbol{p}_0$ and $\boldsymbol{f}(\cdot)$ in (16), the vector $\boldsymbol{p}^*(b)$ approximates equilibrium prices, in a sense, as close as possible.
3. If equilibrium exists, then $\boldsymbol{p}^*(b)$ is a vector of equilibrium prices.

Thus, the sequence $\boldsymbol{p}_n(b)$ converges in any case to a "good" system of prices $\boldsymbol{p}^*(b)$.

One can easily verify that the market model described in Section 3 (called "model G") is a special case of the "economy" considered in [5]. However, the use of the MN-mechanism to allocate resources in the model G encounters serious technical difficulties, which will be discussed below.

5. Applying the MN-mechanism to model G

The MN-mechanism considers each resource unit as an independent "unit resource". Therefore, a price vector in the model G must be of the form $\boldsymbol{p} = (p^{\tau}_{jk})_{j,k,\tau}$, where p^{τ}_{jk} is the price of the unit resource (j, k, τ). Assuming that different units of the resource j at the moment τ can have different prices in the model G, we obtain a "disaggregated" model denoted by DG. Problems (10) and (11) for the model DG have the form

$$\begin{aligned} & v_s(\boldsymbol{x}^s, \boldsymbol{p}) = u_s(\boldsymbol{x}^s) - \sum_{k,\tau} p^{\tau}_{jk} x^s_{jk}(\tau) \to \max \\ \text{s.t.} \quad & 0 \le \sum_k x^s_{jk}(\tau) \le V(s), \ \ \forall\tau, \ \text{ where } j = s - I_U, \end{aligned} \tag{17}$$

and, respectively,

$$\begin{aligned} v_i(\boldsymbol{x}^i, \boldsymbol{p}) = u_i(\boldsymbol{x}^i) - \sum_{j,k,\tau} x^i_{jk}(\tau) p^{\tau}_{jk} \to \max \\ \text{s.t.} \quad \boldsymbol{x}^i \in X_i. \end{aligned} \tag{18}$$

Theorem. *If $(\boldsymbol{x}, \boldsymbol{p})$ is an equilibrium in the model DG, then there exists a vector of prices $\hat{\boldsymbol{p}}$, such that $(\boldsymbol{x}, \hat{\boldsymbol{p}})$ is an equilibrium in the model DG and $\hat{p}^{\tau}_{jk}$ does not depend on k (the prices of all units of the resource j at the moment τ are equal).*

The proof of this theorem (see [11]) gives an efficient method for constructing the price system $\hat{\boldsymbol{p}}$. It is clear that $\hat{\boldsymbol{p}}$ gives equilibrium prices for the model G, that is, if DG has an equilibrium, then G has an equilibrium as well. The reverse is also certainly true. However, an equilibrium for DG is not necessarily an equilibrium for G.

Equations of system (16) for the model DG take the form

$$p^{\tau}_{jk}(t_l) = p^{\tau}_{jk}(t_{l-1}) + \Delta_n[f^{\tau}_{jk}(\boldsymbol{p}(t_{l-1})) - \boldsymbol{e}], \tag{19}$$

where $f^{\tau}_{jk}(\boldsymbol{p})$ is a corresponding component of the vector-function $\boldsymbol{f}(\boldsymbol{p})$.

5.1. Choosing the functions $\boldsymbol{f}(\cdot)$. The easiest way of fixing the functions $\boldsymbol{f}(\cdot)$ is to select $\boldsymbol{x}^i(\boldsymbol{p}) \in D_i(\boldsymbol{p})$ for each $\boldsymbol{p}$ and i, and set $\boldsymbol{f}(\boldsymbol{p}) = \boldsymbol{f}_1(\boldsymbol{p}) = \sum_i \boldsymbol{x}^i(\boldsymbol{p})$. Then, to realize procedure (16), one should be able to find solutions to both problems (17) and (18) for any $\boldsymbol{p}$.

At such a choice of the function $\boldsymbol{f}(\cdot)$, the prices of different units of the same resource at the same moment can differ both during the process and in the limit. If an equilibrium exists and the corresponding prices are constructed, we will be able to "correct" them in accordance with the theorem.

If there is no equilibrium or we have not found a good approximation to equilibrium prices, two units of the same resource at the same moment are, nevertheless, equivalent for all market agents, and it is desirable that their prices should be equal. We propose to provide such an equality at all iterations of the MN-mechanism by choosing the function $\boldsymbol{f}(\cdot)$ in the following way.

For any $\boldsymbol{p}$, let $\boldsymbol{x}^i(\boldsymbol{p})$ be equal to the arithmetic mean of all terms of $D_i(\boldsymbol{p})$ (this is a finite set) and set $\boldsymbol{f}(\boldsymbol{p}) = \boldsymbol{f}_2(\boldsymbol{p}) = \sum_i \boldsymbol{x}^i(\boldsymbol{p})$. Then $f^\tau_{jk}(\boldsymbol{p})$ in (19) is equal to the frequency ν^τ_{jk} of occurrence of the unit resource (j,k,τ) in the consumption bundles from $D_i(\boldsymbol{p})$. Hence, to realize the MN-mechanism, it is sufficient to know, at each iteration, the number of optimal resource bundles $n(i)$ for each market agent i and the number $m(i,j,k,\tau)$ of the optimal resource bundles including the unit resource (j,k,τ) (for all i, j, k, τ).

Let us assume that the price vector $\boldsymbol{p}(t_{l-1})$ provides *a uniform price* of the resource j at the moment τ for all j and τ ($p^\tau_{jk}(t_{l-1})$ does not depend on k). Then the replacement of the unit resource (j,k_1,τ) by the unit resource (j,k_2,τ) does not change utility of a resource bundle for a market agent. Thus, the values $m(i,j,k,\tau)$ do not depend on k; then $f^\tau_{jk}(\boldsymbol{p})$ and $\boldsymbol{p}(t_l)$ in (19) do not depend on k, too. In other words, if $\boldsymbol{f}(\cdot) = \boldsymbol{f}_2(\cdot)$ and the initial price vector $\boldsymbol{p}(0)$ provides uniform prices, then uniformity of prices is maintained at all iterations. In particular, one can set $p^\tau_{jk}(0) = c_s$ for $s = j + I_U$ (the price of each resource unit is equal to the put-up price of the supplier). In this case, all equations in (19) corresponding to given j and τ are identical; it is sufficient to keep only one of them.

5.2. Selecting the optimal resource bundles for market agents. It follows from the preceding section that implementation of the MN-mechanism requires (depending on the choice of the function $\boldsymbol{f}(\cdot)$) either one element from each $D_i(\boldsymbol{p})$ or the frequency of occurrence of each unit resource in the consumption bundles from $D_i(\boldsymbol{p})$ ($1 \le i \le I$). In addition, when the prices $\boldsymbol{p}^*$ are found, it is desirable to have full descriptions of the sets $D_i(\boldsymbol{p}^*)$.

Problem (17) can be easily solved. Let us fix s and $\boldsymbol{p}$, and set $j = s - I_U$. Substituting the expression for u_s (see Subsection 3.4) into (17), we reduce this problem to the following form:

$$\sum_{k,\tau}(c_s - p^\tau_{jk})x^s_{jk}(\tau) \to \max \quad \text{s.t.} \quad 0 \le \sum_k x^s_{jk}(\tau) \le V(s), \ \ \forall\tau.$$

Let K^+, K^-, and K^0 be a set of unit resources (j,k,τ) such that $p^\tau_{jk} > c_s$, $p^\tau_{jk} < c_s$, and $p^\tau_{jk} = c_s$, respectively. Then each resource bundle in $D_s(\boldsymbol{p})$ does not include the unit resources from K^+, but includes all unit resources from K^- and some resources from K^0. The number of such bundles is $2^{|K^0|}$, and each unit resource from K^0 participates in half of them. Hence, ν^τ_{jk} is equal to 0, 1, or 0.5 if the unit resource (j, k, τ) belongs to K^+, K^-, or K^0, respectively.

Let us consider problem (18). Apparently, it is an awkward combinatorial problem in the general case. However, the problem can be considerably simplified if we assume, for instance, that each job is either not executed or executed at a constant speed and without interruptions.

Let us make another simplifying supposition: all processors are identical, and each of them can provide only one slot per unit time. This means that $V(s) = 1$ for $I_U < s \leq I_U + I_C$. An algorithm to solve problem (18) providing this assumption is described in [11]. This algorithm makes a partial enumeration of integer-valued pairs (a, b), $1 \leq a \leq b \leq T$. It allows us to find an optimal solution in a reasonable time. However, the calculation of the number of optimal bundles and, moreover, listing all such bundles, generally speaking, result in exponential complexity.

6. Conclusion

Let us assume that we have constructed the limit vector of prices $\boldsymbol{p}^*$, using the MN-mechanism. To allocate resources at these prices, it is necessary to describe the sets $D_i(\boldsymbol{p}^*)$. If all of them are single-element sets, then an equilibrium exists (see [5, Theorem 8]). One can obtain the corresponding resources distribution by allocating resources to each market agent i according to the unique demand vector included into $D_i(\boldsymbol{p}^*)$. With multi-element sets $D_i(\boldsymbol{p}^*)$, the distribution of resources becomes a difficult combinatorial problem, even if an equilibrium exists.

The literature gives many examples of the absence of equilibrium in the markets of indivisible goods. It might be supposed that in our model, as well, the indivisibility of files and slots will make an exact equality between demand and supply a rare fact. Therefore, the statement of the problem of resources distribution at prices $\boldsymbol{p}^*$ should take into account the fact that these prices, possibly, do not equilibrate the market.

The problem can be formulated as follows. A system $(D_i \mid i \in I)$ of families of subsets of a certain finite set (of resource units) M is given. It is necessary to construct a set of pairwise disjoint terms of the set $\bigcup_i D_i$ containing no more than one representative from each D_i and maximal in a certain sense (for example, in the number of subsets, in the union of subsets power, or in the aggregate priority of the "represented" numbers i). If equilibrium exists, then this set determines the equilibrium distribution.

It should be noted that the proposed model does not fully reflect the Grid dynamics. The allocation problem discussed is dynamic in that it gives the timing of jobs execution. We assumed, however, that sets of jobs and unit resources do not change. This is not the case in reality, and the resources have to be redistributed time from time due to appearance / disappearance of users and/or suppliers. The following proposal is obvious: the system gathers requests for the jobs execution in the period $[t, \infty)$ up to the moment $t-\varepsilon$, and distributes the resources for these jobs during the interval $[t-\varepsilon, t)$. However, the solution to the resources distribution problem can require so many resources that there will be nothing to distribute. It is possible that a quick procedure like auction would be more adequate to the situation being considered.

References

[1] Buyya R., Abramson D., Giddy J. An economy driven resource management architecture for global computational power grids // Intern. Conf. on Parallel and Distributed Processing Techniques and Applications. PDPTA-2000, Las Vegas, USA, June. — 2000. — Vol. 5. — P. 3052–3060.

[2] Buyya R., Murshed M. A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing // Concurrency and Computation: Practice and Experience. — 2002. — Vol. 14. — P. 1175–1220.

[3] Wolski R., Brevik J., Plank J., Bryan T. Grid resource allocation and control using computational economies // Grid Computing. Making the Global Infrastructure a Reality / F. Berman, G. Fox, and T. Hey, Eds. — Chichester: John Wiley & Sons, 2003. — Ch. 32. — P. 747–773.

[4] Wolski R., Plank J., Bryan T., Brevik J. Analyzing market-based resource allocation strategies for the computational grid // Intern. J. of High Performance Computing Applications. — 2000. — Vol. 15, No. 3.

[5] Ma J., Nie F. Walrasian equilibrium in an exchange economy with indivisibilities // Mathematical Social Science. — 2003. — Vol. 46. — P. 159–192.

[6] Van den Elzen A. Adjustment Processes for Exchange Economies and Noncooperative Games. — Berlin: Springer, 1993. — (Lecture Notes in Economics and Mathematical Systems; 402).

[7] Hahn F. Stability // Handbook of Mathematical Economics / K. Arrow and M. Intriligator, Eds. — Amsterdam, 1993. — Vol. II, Ch. 16. — P. 745–703.

[8] Arrow K., Hurwicz L. The stability of the competitive equilibrium I // Econometrica. — 1958. — Vol. 26. — P. 522–552.

[9] Arrow K., Hurwicz L. The stability of the competitive equilibrium II // Econometrica. — 1959. — Vol. 27. — P. 82–109.

[10] Scarf H. Some examples of global instability of the competitive equilibrium // Intern. Econ. Review. — 1960. — Vol. 1. — P. 157–172.

[11] Bredikhin S., Vjalkov I., Savchenko I., Khutoretsky A.B. Allocation of network resources with price adjustment: two models // Siberian J. of Industrial Mathematics. — 2006. — Vol. IX, No. 1(25). — P. 28–46 (In Russian).