Cybernetics and Systems Analysis, Vol. 48, No. 1, January, 2012

ASYNCHRONOUS DISTRIBUTED COMPUTATIONS WITH A LIMITED NUMBER OF COPIES OF A STRUCTURED PROGRAM RESOURCE

N. S. Kovalenko,[a] **P. A. Pavlov,**[b] **and M. I. Ovseec**[c]

UDC 681.3.06

Abstract. *Problems of finding minimum total execution times of distributed competing processes are solved for a limited number of copies of a structured program resource under conditions of unbounded and bounded parallelism.*

Keywords: *distributed competing process, program resource, asynchronous mode, synchronous mode, limited parallelism, unlimited parallelism, homogeneous system, identically distributed system.*

INTRODUCTION

In many applications connected with designing multiprocessor systems (MSs), computer complexes (CCs), system and applied software [1], and optimal organization of parallel computational processes, of significant interest are problems in which many competing processes can use not one but several copies of a structured program resource (PR). The case when a common (shared) memory of a multiprocessor system contains one copy of this PR was investigated from different positions in [2–8]. In these works, problems of finding the minimum total execution time were solved for distributed competing processes using a program resource structured into blocks in different interaction modes of processes, processors, and blocks [2–6], criteria of efficiency and optimality of structurization of program resources [7] are obtained, a comparative analysis of interaction modes of processes, processors, and blocks [8] is conducted, and a number of optimization problems of calculating the number of processes, minimum number of processors, etc. were solved. The investigation of these and other problems specific to the optimal organization of parallel computations becomes especially topical in the case when $c \geq 2$ copies of a program resource can be simultaneously placed in the shared memory of an MS. This generalization is of fundamental importance since it reflects main features of multiconveyor processing and also allows one to compare the efficiency of conveyor and parallel processing.

The present article constructs and investigates a mathematical model of organization of competing processes using a limited number of copies of a program resource. Using ideas of the method of block structurization of program resources with their subsequent conveyorization between processes and processors, optimal time characteristics of this organization are investigated.

1. MATHEMATICAL MODEL OF DISTRIBUTED COMPUTATIONS FOR THE CASE WHEN THE NUMBER OF COPIES OF A PROGRAM RESOURCE IS LIMITED

Constructive elements for the construction of mathematical models of systems of distributed computations are the concepts of a process and a program resource.

A process is understood to be a sequence of blocks (commands or procedures) $Q_1, Q_2, \ldots, Q_s$ that are executed on many processors (processor nodes, processing devices, or intelligent clients). A process is called distributed if all its blocks

[a]Belarus State Economic University, Minsk, Republic of Belarus. [b]Polessky State University, Pinsk, Republic of Belarus, *pin2535@tut.by*. [c]Private Institute of Business, Minsk, Republic of Belarus. Translated from Kibernetika i Sistemnyi Analiz, No. 1, pp. 105–117, January–February 2012. Original article submitted May 25, 2010.

or some of them are processed by different processors [2]. To accelerate the execution of processes, computations can be performed simultaneously and interact to interchange data. Such processes are called cooperative or interacting processes.

The concept of a resource is used to denote any objects of a computing system that can be used by processes for their execution. Reenterable (reusable) resources can be simultaneously used by several computational processes. For parallel systems, the situation is typical when the same sequence of blocks or their portion should repeatedly be executed by processors. We call such a sequence a program resource and the set of corresponding processes competing processes.

When the number of copies of a program resource is limited, the mathematical model of distributed processing of competing interacting processes includes $p,\ p \geq 2$, processors of a multiprocessor system that have access to the shared memory; $n,\ n \geq 2$, distributed competing processes; $s,\ s \geq 2$, blocks of the structured program resource; a matrix $T = [t_{ij}]$, $i = \overline{1, n},\ j = \overline{1, s}$, of times of executing blocks of the program resource by distributed interacting competing processes; the number of copies $2 \leq c \leq p$ of the program resource structured into blocks that are simultaneously available in the main memory to all p processors, $\left[\frac{p}{c}\right] \geq 2$; $\varepsilon > 0$ is the parameter describing the time of additional system expenditures for the organization of the conveyor mode of using blocks of the structured program resource by a set of interacting competing processes in the course of distributed processing.

We also assume that the number of processes n is a multiple of the number of copies c of the structured program resource, i.e., $n = mc,\ m \geq 2$, and that the interaction of processes, processors, and blocks of the program resource satisfies the following conditions:

(1) any processor cannot simultaneously process more than one block;

(2) processes are executed by groups in the parallel–conveyor mode, i.e., c copies of each block are executed simultaneously (in parallel) in combination with the conveyorization of groups consisting of c blocks between processors and processes;

(3) each block of the program resource is processed without interruptions;

(4) for each process $i = lc + q,\ i = \overline{1, n},\ l \geq 0,\ q = \overline{1, c}$, blocks of the program resource are cyclically distributed among processors according to the following rule: the block with a number $j = k\left[\frac{p}{c}\right] + r,\ j = \overline{1, s},\ k \geq 0,\ r = \overline{1, \left[\frac{p}{c}\right]}$, is assigned to the processor with a number $q + c(r-1)$.

We introduce the following interaction modes for processes, processors, and blocks with allowance for the presence of c copies of the program resource:

(1) the asynchronous mode, in which the beginning of the execution of the next group consisting of c copies of the jth block $Q_j,\ j = \overline{1, s}$, is determined by the presence of c processors and readiness of this group of blocks for execution (a program block is considered to be ready for execution if it is not executed on any processor);

(2) the first synchronous mode providing the linear order of executing blocks of the program resource within each process without delays, i.e., for each process $i = lc + q,\ i = \overline{1, n},\ l \geq 0,\ q = \overline{1, c}$, the moment of termination of the jth block on the $(q + c(r-1))$th processor coincides with the moment of the beginning of executing the next $(j+1)$th block on the $(q + cr)$th processor, $j = \overline{1, s-1}$, $r = \overline{1, \left[\frac{p}{c}\right]}$;

(3) the second synchronous mode in which c copies of each block are continuously moved between groups consisting of c processes, i.e., the moment of termination of processing c copies of the current block coincides with the moment of the beginning of their processing on the next group consisting of c processors.

Figures 1–3 present Gantt charts illustrating the execution of $n = 4$ distributed competing processes using $c = 2$ copies of the structured program resource in an MS with $p = 7$ processors in the modes considered above with a specified matrix of execution times for PR blocks with allowance for additional system expenditures $T^{\varepsilon} = \begin{bmatrix} 3 & 1 & 4 \\ 2 & 2 & 1 \\ 1 & 3 & 3 \\ 4 & 1 & 2 \end{bmatrix}$.

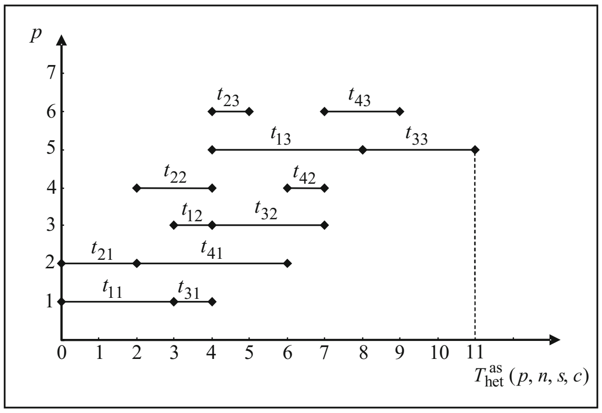

Fig. 1. Diagram of the asynchronous mode.

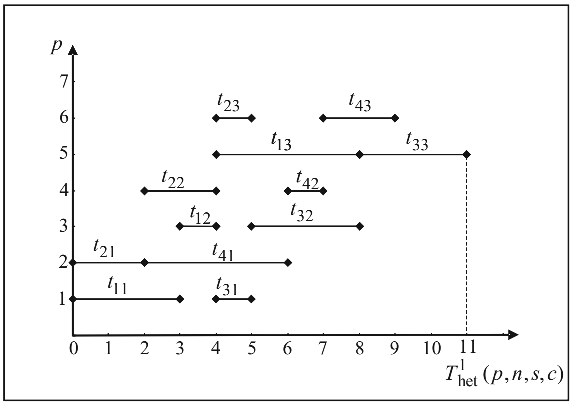

Fig. 2. Diagram of the first synchronous mode.

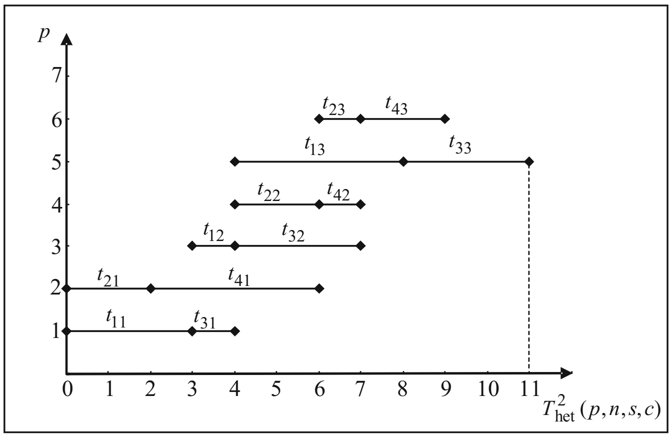

Fig. 3. Diagram of the second synchronous mode.

Definition 1. A system consisting of n distributed competing processes is called heterogeneous if execution times of blocks of the program resource $Q_1, Q_2, \ldots, Q_s$ depend on amounts of the data being processed and/or on their structure, i.e., they are different for different processes.

Definition 2. We call a system of distributed competing processes homogeneous if the times of execution of the Q_jth block by each ith process are equal to one another, i.e., $t_{ij} = t_j$, $i = \overline{1, n}$, $j = \overline{1, s}$.

Definition 3. A system of competing processes is understood to be identically distributed if the times t_{ij} of execution of the jth block Q_j, $j = \overline{1, s}$, of the program resource by each ith process coincide and are equal to t_i for all $i = \overline{1, n}$, i.e., a chain of equalities $t_{i1} = t_{i2} = \ldots = t_{is} = t_i$ takes place for all $i = \overline{1, n}$.

2. MINIMUM TOTAL EXECUTION TIME OF HETEROGENEOUS DISTRIBUTED PROCESSES IN THE ASYNCHRONOUS MODE WHEN THE NUMBER OF PROCESSORS IS SUFFICIENT

We denote by $T_{\text{het}}^{\text{as}}(p,n,s,c)$ the minimum total time of execution of n heterogeneous distributed competing processes when the number c of copies of the program resource is limited in a multiprocessor system with p processors in the asynchronous mode with allowance for a parameter ε. To compute $T_{\text{het}}^{\text{as}}(p,n,s,c)$, we consider the cases of unbounded ($s \le \left[\frac{p}{c}\right]$) and bounded ($s > \left[\frac{p}{c}\right]$) parallelism.

Let there be a system $n = mc$, $m \geq 2$, $2 \leq c \leq p$, of heterogeneous distributed competing processes, and let the number of blocks s of a structured program resource do not exceed the number of processor groups each of which consists of c processors, i.e., $2 \leq s \leq \left[\frac{p}{c}\right]$. In this case, without any loss of generality, we can consider that each Q_jth block, $j = \overline{1, s}$, of the ith process, where $i = lc + q$, $i = \overline{1, n}$, $l \geq 0$, $q = \overline{1, c}$, is assigned to the $(q + c(r-1))$th processor, $r = \overline{1, \left[\frac{p}{c}\right]}$. Then, to execute n processes, it suffices to use $p = \left[\frac{p}{c}\right] s$ processors, and the other $p - \left[\frac{p}{c}\right] s$ processors will be idle.

Let $T^{\varepsilon} = [t_{ij}^{\varepsilon}]$ be an $n \times s$ matrix of execution times of blocks of the program resource by each ith process with allowance for the parameter $\varepsilon > 0$, where $t_{ij}^{\varepsilon} = t_{ij} + \varepsilon$, $i = \overline{1, n}$, $j = \overline{1, s}$. To compute the minimum total time $T_{\text{het}}^{\text{as}}(p, n, s, c)$, the functional of the Bellman–Johnson problem can be used that, in this case, assumes the form

$$T_{\text{h}}^{\text{as}}(p, n, s, c) = \max_{1 \leq u_1 \leq u_2 \leq \ldots \leq u_{s-1} \leq m} \left[\sum_{i=1}^{u_1} t_{q+(i-1)c,1}^{\varepsilon} + \sum_{i=u_1}^{u_2} t_{q+(i-1)c,2}^{\varepsilon} + \ldots + \sum_{i=u_{s-1}}^{m} t_{q+(i-1)c,s}^{\varepsilon} \right], \tag{1}$$

where $m = \frac{n}{c}$, $t_{q+(i-1)c,j}^{\varepsilon} = t_{q+(i-1)c,j} + \varepsilon$, $q = \overline{1, c}$, $i = \overline{1, m}$, $j = \overline{1, s}$, and $u_1, u_2, \ldots, u_{s-1}$ are integers.

Example 1. We illustrate an interpretation of formula (1) by a numerical example. Assume that $p = 7$, $n = 6$, $s = 3$, and $c = 2$ and that times of execution of blocks by processes are specified by the matrix $T^{\varepsilon} = \begin{bmatrix} 3 & 1 & 4 \\ 2 & 2 & 1 \\ 1 & 3 & 3 \\ 4 & 1 & 2 \\ 3 & 2 & 1 \\ 1 & 4 & 1 \end{bmatrix}$. Then $m = 3$ and, hence, functional (1) assumes the form

$$T_{\text{het}}^{\text{as}}(p = 7, n = 6, s = 3, c = 2)$$

$$= \max_{1 \leq u_1 \leq u_2 \leq 3} \left[\sum_{i=1}^{u_1} t_{q+(i-1)2,1}^{\varepsilon} + \sum_{i=u_1}^{u_2} t_{q+(i-1)2,2}^{\varepsilon} + \sum_{i=u_2}^{3} t_{q+(i-1)2,3}^{\varepsilon} \right], \quad q = \overline{1, 2}.$$

When $q = 1$, we have

$$\max_{1 \leq u_1 \leq u_2 \leq 3} \left[\sum_{i=1}^{u_1} t_{q+(i-1)2,1}^{\varepsilon} + \sum_{i=u_1}^{u_2} t_{q+(i-1)2,2}^{\varepsilon} + \sum_{i=u_2}^{3} t_{q+(i-1)2,3}^{\varepsilon} \right] = \max_{1 \leq u_1 \leq u_2 \leq 3} \left[\sum_{i=1}^{u_1} t_{2i-1,1}^{\varepsilon} + \sum_{i=u_1}^{u_2} t_{2i-1,2}^{\varepsilon} + \sum_{i=u_2}^{3} t_{2i-1,3}^{\varepsilon} \right].$$

If $u_1 = 1$, then $u_2 = \overline{1, 3}$; if $u_1 = 2$, then $u_2 = 2, 3$; if $u_1 = 3$, then $u_2 = 3$. Then we have

$$\max_{1 \leq u_1 \leq u_2 \leq 3} \left[\sum_{i=1}^{u_1} t_{2i-1,1}^{\varepsilon} + \sum_{i=u_1}^{u_2} t_{2i-1,2}^{\varepsilon} + \sum_{i=u_2}^{3} t_{2i-1,3}^{\varepsilon} \right]$$

$$= \max \begin{bmatrix} \sum_{i=1}^{1} t_{2i-1,1}^{\varepsilon} + \sum_{i=1}^{1} t_{2i-1,2}^{\varepsilon} + \sum_{i=1}^{3} t_{2i-1,3}^{\varepsilon}, & \sum_{i=1}^{1} t_{2i-1,1}^{\varepsilon} + \sum_{i=1}^{2} t_{2i-1,2}^{\varepsilon} + \sum_{i=2}^{3} t_{2i-1,3}^{\varepsilon}, \\ \sum_{i=1}^{1} t_{2i-1,1}^{\varepsilon} + \sum_{i=1}^{3} t_{2i-1,2}^{\varepsilon} + \sum_{i=3}^{3} t_{2i-1,3}^{\varepsilon}, & \sum_{i=1}^{2} t_{2i-1,1}^{\varepsilon} + \sum_{i=2}^{2} t_{2i-1,2}^{\varepsilon} + \sum_{i=2}^{3} t_{2i-1,3}^{\varepsilon}, \\ \sum_{i=1}^{2} t_{2i-1,1}^{\varepsilon} + \sum_{i=2}^{3} t_{2i-1,2}^{\varepsilon} + \sum_{i=3}^{3} t_{2i-1,3}^{\varepsilon}, & \sum_{i=1}^{3} t_{2i-1,1}^{\varepsilon} + \sum_{i=3}^{2} t_{2i-1,2}^{\varepsilon} + \sum_{i=3}^{3} t_{2i-1,3}^{\varepsilon} \end{bmatrix}$$

$$= \max \begin{bmatrix} \underline{t_{1,1}^{\varepsilon} + t_{1,2}^{\varepsilon} + t_{1,3}^{\varepsilon} + t_{3,3}^{\varepsilon} + t_{5,3}^{\varepsilon}}, & t_{1,1}^{\varepsilon} + t_{1,2}^{\varepsilon} + t_{3,2}^{\varepsilon} + t_{3,3}^{\varepsilon} + t_{5,3}^{\varepsilon}, \\ t_{1,1}^{\varepsilon} + t_{1,2}^{\varepsilon} + t_{3,2}^{\varepsilon} + t_{5,2}^{\varepsilon} + t_{5,3}^{\varepsilon}, & t_{1,1}^{\varepsilon} + t_{3,1}^{\varepsilon} + t_{3,2}^{\varepsilon} + t_{3,3}^{\varepsilon} + t_{5,3}^{\varepsilon}, \\ t_{1,1}^{\varepsilon} + t_{3,1}^{\varepsilon} + t_{3,2}^{\varepsilon} + t_{5,2}^{\varepsilon} + t_{5,3}^{\varepsilon}, & t_{1,1}^{\varepsilon} + t_{3,1}^{\varepsilon} + t_{5,1}^{\varepsilon} + t_{5,2}^{\varepsilon} + t_{5,3}^{\varepsilon} \end{bmatrix}$$

$$= \max \begin{bmatrix} \underline{3+1+4+3+1}, & 3+1+3+3+1, \\ 3+1+3+2+1, & 3+1+3+3+1, \\ 3+1+3+2+1, & 3+1+3+2+1 \end{bmatrix} = \max[12, 11, 10, 11, 10, 10] = 12.$$

When $q = 2$, we have $\max\limits_{1 \le u_1 \le u_2 \le 3} \left[\sum\limits_{i=1}^{u_1} t_{q+(i-1)2,1}^{\varepsilon} + \sum\limits_{i=u_1}^{u_2} t_{q+(i-1)2,2}^{\varepsilon} + \sum\limits_{i=u_2}^{3} t_{q+(i-1)2,3}^{\varepsilon} \right]$

$$= \max_{1 \le u_1 \le u_2 \le 3} \left[\sum_{i=1}^{u_1} t_{2i,1}^{\varepsilon} + \sum_{i=u_1}^{u_2} t_{2i,2}^{\varepsilon} + \sum_{i=u_2}^{3} t_{2i,3}^{\varepsilon} \right].$$

If $u_1 = 1$, then $u_2 = \overline{1,3}$; if $u_1 = 2$, then $u_2 = 2, 3$; if $u_1 = 3$, then $u_2 = 3$. Then we obtain

$$\max_{1 \le u_1 \le u_2 \le 3} \left[\sum_{i=1}^{u_1} t_{2i,1}^{\varepsilon} + \sum_{i=u_1}^{u_2} t_{2i,2}^{\varepsilon} + \sum_{i=u_2}^{3} t_{2i,3}^{\varepsilon} \right]$$

$$= \max \begin{bmatrix} \sum\limits_{i=1}^{1} t_{2i,1}^{\varepsilon} + \sum\limits_{i=1}^{1} t_{2i,2}^{\varepsilon} + \sum\limits_{i=1}^{3} t_{2i,3}^{\varepsilon}, & \sum\limits_{i=1}^{1} t_{2i,1}^{\varepsilon} + \sum\limits_{i=1}^{2} t_{2i,2}^{\varepsilon} + \sum\limits_{i=2}^{3} t_{2i,3}^{\varepsilon}, \\ \sum\limits_{i=1}^{1} t_{2i,1}^{\varepsilon} + \sum\limits_{i=1}^{3} t_{2i,2}^{\varepsilon} + \sum\limits_{i=3}^{3} t_{2i,3}^{\varepsilon}, & \sum\limits_{i=1}^{2} t_{2i,1}^{\varepsilon} + \sum\limits_{i=2}^{2} t_{2i,2}^{\varepsilon} + \sum\limits_{i=2}^{3} t_{2i,3}^{\varepsilon}, \\ \sum\limits_{i=1}^{2} t_{2i,1}^{\varepsilon} + \sum\limits_{i=2}^{3} t_{2i,2}^{\varepsilon} + \sum\limits_{i=3}^{3} t_{2i,3}^{\varepsilon}, & \sum\limits_{i=1}^{3} t_{2i,1}^{\varepsilon} + \sum\limits_{i=3}^{3} t_{2i,2}^{\varepsilon} + \sum\limits_{i=3}^{3} t_{2i,3}^{\varepsilon} \end{bmatrix}$$

$$= \max \begin{bmatrix} t_{2,1}^{\varepsilon} + t_{2,2}^{\varepsilon} + t_{2,3}^{\varepsilon} + t_{4,3}^{\varepsilon} + t_{6,3}^{\varepsilon}, & t_{2,1}^{\varepsilon} + t_{2,2}^{\varepsilon} + t_{4,2}^{\varepsilon} + t_{4,3}^{\varepsilon} + t_{6,3}^{\varepsilon}, \\ t_{2,1}^{\varepsilon} + t_{2,2}^{\varepsilon} + t_{4,2}^{\varepsilon} + t_{6,2}^{\varepsilon} + t_{6,3}^{\varepsilon}, & t_{2,1}^{\varepsilon} + t_{4,1}^{\varepsilon} + t_{4,2}^{\varepsilon} + t_{4,3}^{\varepsilon} + t_{6,3}^{\varepsilon}, \\ \underline{t_{2,1}^{\varepsilon} + t_{4,1}^{\varepsilon} + t_{4,2}^{\varepsilon} + t_{6,2}^{\varepsilon} + t_{6,3}^{\varepsilon}}, & \underline{t_{2,1}^{\varepsilon} + t_{4,1}^{\varepsilon} + t_{6,1}^{\varepsilon} + t_{6,2}^{\varepsilon} + t_{6,3}^{\varepsilon}} \end{bmatrix}$$

$$= \max \begin{bmatrix} 2+2+1+2+1, & 2+2+1+2+1, \\ 2+2+1+4+1, & 2+4+1+2+1, \\ \underline{2+4+1+4+1}, & \underline{2+4+1+4+1} \end{bmatrix} = \max[8, 8, 10, 10, 12, 12] = 12.$$

(Underlined sums denote maximum execution times.)

Hence, the minimum total execution time $n = 6$ for heterogeneous distributed interacting competing processes using $c = 2$ copies of the program resource structured into $s = 3$ blocks in a multiprocessor system with $p = 7$ processors in the asynchronous mode amounts to $T_{\text{het}}^{\text{as}}(p, n, s, c) = 12$. In this case, six processors will be used.

Let us consider an algorithm that makes it possible to more efficiently solve the problem of finding the minimum total time $T_{\text{het}}^{\text{as}}(p, n, s, c)$ of execution of heterogeneous distributed competing processes in the asynchronous mode.

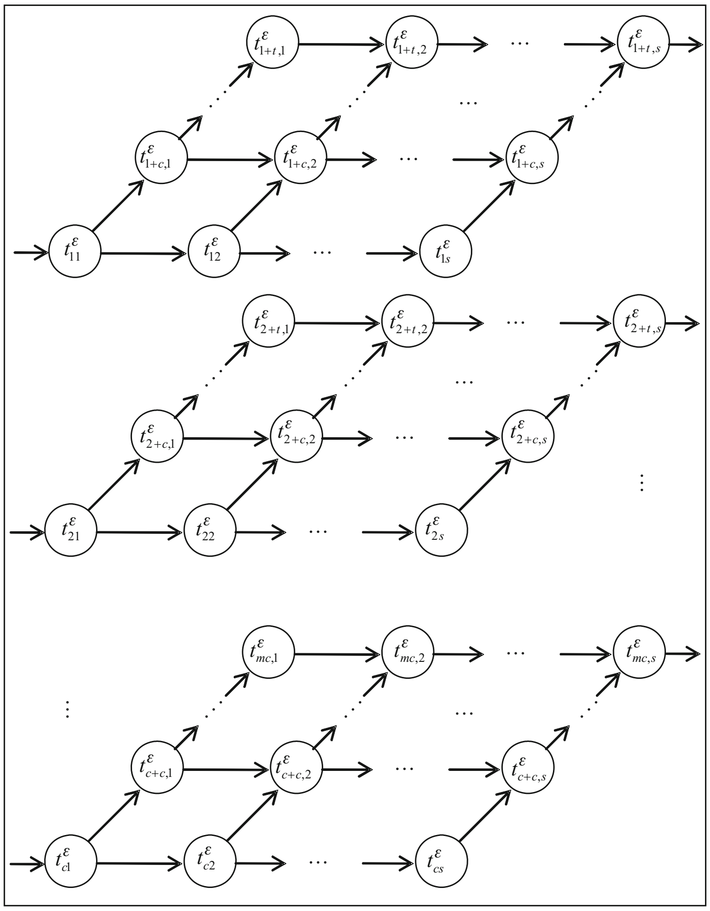
$t^{\varepsilon}_{1+t,1}$
$t^{\varepsilon}_{1+t,2}$
$t^{\varepsilon}_{1+t,s}$
$t^{\varepsilon}_{1+c,1}$
$t^{\varepsilon}_{1+c,2}$
$t^{\varepsilon}_{1+c,s}$
t^{ε}_{11}
t^{ε}_{12}
t^{ε}_{1s}
$t^{\varepsilon}_{2+t,1}$
$t^{\varepsilon}_{2+t,2}$
$t^{\varepsilon}_{2+t,s}$
$t^{\varepsilon}_{2+c,1}$
$t^{\varepsilon}_{2+c,2}$
$t^{\varepsilon}_{2+c,s}$
t^{ε}_{21}
t^{ε}_{22}
t^{ε}_{2s}
$t^{\varepsilon}_{mc,1}$
$t^{\varepsilon}_{mc,2}$
$t^{\varepsilon}_{mc,s}$
$t^{\varepsilon}_{c+c,1}$
$t^{\varepsilon}_{c+c,2}$
$t^{\varepsilon}_{c+c,s}$
t^{ε}_{c1}
t^{ε}_{c2}
t^{ε}_{cs}

Fig. 4. Graph G_1^c.

From given $s, c, m = \frac{n}{c}$, and a matrix $T^{\varepsilon} = [t^{\varepsilon}_{q+(i-1)c,j}]$, $q = \overline{1,c}$, $i = \overline{1,m}$, $j = \overline{1,s}$, we construct a c-layer vertex–weighted graph G_1^c. Each qth, $q = \overline{1,c}$, layer of the graph G_1^c consists of vertices $t^{\varepsilon}_{q+(i-1)c,j}$, $i = \overline{1,m}$, $j = \overline{1,s}$, that are located at nodes of a rectangular $m \times s$-lattice in which t^{ε}_{q1} are input vertices and $t^{\varepsilon}_{q+t,s}$ are output vertices, $q = \overline{1,c}$, $t = (m-1)c$ (Fig. 4). In each layer q, arcs reflect the linear order of execution of blocks Q_j, $j = \overline{1,s}$, of the program resource by each $(q+(i-1)c)$th process, $q = \overline{1,c}$, $i = \overline{1,m}$, and also the linear order of using each block by all m processes. Thus, the following theorem takes place.

THEOREM 1. The minimum total execution time $n = mc$, $m \ge 2$, of heterogeneous distributed competing processes using $2 \le c \le p$ copies of the program resource structured into $2 \le s \le \left[\frac{p}{c}\right]$ blocks with execution times of blocks taking into account system expenditures $[t^{\varepsilon}_{ij}]$, $i = \overline{1,n}$, $j = \overline{1,s}$, in a multiprocessor system with p, $\left[\frac{p}{c}\right] \ge 2$, processors in the asynchronous mode is determined by the length of the critical path in a c-layer vertex–weighted graph G_1^c from the initial vertex t^{ε}_{q1} to the terminal vertex $t^{\varepsilon}_{q+(m-1)c,s}$, $q = \overline{1,c}$, $m = \frac{n}{c}$.

Example 2. Using the data from Example 1, find the minimum total time $T^{\text{as}}_{\text{het}}(p,n,s,c)$ using an algorithm for finding the critical path in the c-layer vertex–weighted graph G_1^c.

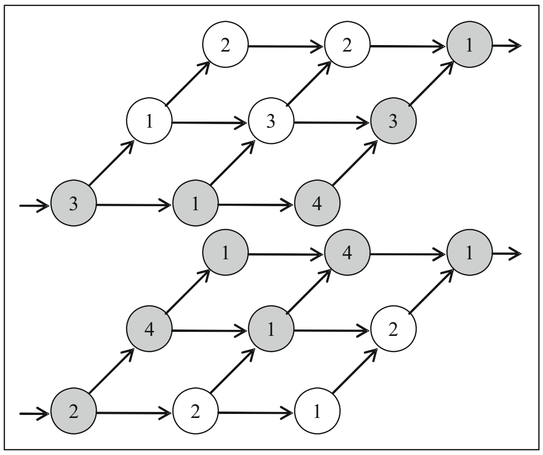

Fig. 5. C-layer vertex–weighted graph G_1^c.

From given $n = 6$, $s = 3$, and a matrix T^ε, we construct a 2-layer ($c = 2$) vertex–weighted graph G_1^c (Fig. 5). Each layer contains ms vertices, where $m = n / c = 3$. The length of the critical path in the graph is equal to 12, which coincides with the minimum total time obtained in Example 1.

3. ASYNCHRONOUS MODE OF EXECUTION OF DISTRIBUTED COMPETING PROCESSES UNDER CONDITIONS OF BOUNDED PARALLELISM

Let us consider the case of bounded parallelism i.e., when the number of blocks of a structured program resource is greater than the number of groups consisting of c processors, i.e., $s > \left[\frac{p}{c}\right]$, $s = k\left[\frac{p}{c}\right] + r$, $k \geq 1$, $1 \leq r < \left[\frac{p}{c}\right]$.

As in the case when the shared memory of an MS consists of one copy of the program resource, we divide the set consisting of s blocks into $(k+1)$ groups each of which consists of $\left[\frac{p}{c}\right]$ blocks except for the $(k+1)$th group that contains r blocks [6]. Then we divide the matrix $T^\varepsilon = [t_{ij}^\varepsilon]$, $i = \overline{1, n}$, $j = \overline{1, \left(k\left[\frac{p}{c}\right] + r\right)}$, of execution times of blocks into $(k+1)$ submatrices T_l^ε, $l = \overline{1, k+1}$, of size $n \times \left[\frac{p}{c}\right]$ except for the last T_{k+1}^ε that, when s is not a multiple of $\left[\frac{p}{c}\right]$, will contain only r columns, and the other $\left[\frac{p}{c}\right] - r$ columns will be zero. From each submatrix T_l^ε, $l = \overline{1, k+1}$, we construct the $(k+1)$th linear Gantt chart each of which reflects (in time) the execution of the next $\left[\frac{p}{c}\right]$ blocks of the structured program resource on $p\left[\frac{p}{c}\right]$ processors by all n processes using a bounded number c of copies of the structured program resource. When $r \neq 0$, the $(k+1)$th diagram reflects the execution of the last r blocks on cr processors. Figure 6 presents the Gantt chart for an MS with the parameters $p = 7$, $n = 4$, $s = 3$, $c = 2$, and $T^\varepsilon = \begin{bmatrix} 3 & 1 & 4 & 2 & 1 & 4 & 2 & 1 \\ 2 & 2 & 1 & 1 & 3 & 3 & 2 & 2 \\ 1 & 3 & 3 & 1 & 1 & 3 & 3 & 1 \\ 4 & 1 & 2 & 3 & 1 & 1 & 2 & 5 \end{bmatrix}$.

In this case, the total execution time of all processes using c PR copies is found as the sum of lengths of critical paths in each of successive nonoverlapped Gantt charts. However, this time can be reduced if these Gantt charts are successively and maximally superposed block-by-block beginning with the second diagram along the time axis from right to left without violating technological conditions of the asynchronous mode. As a result of overlapping, the resulting overlapped Gantt chart is obtained (Fig. 7).

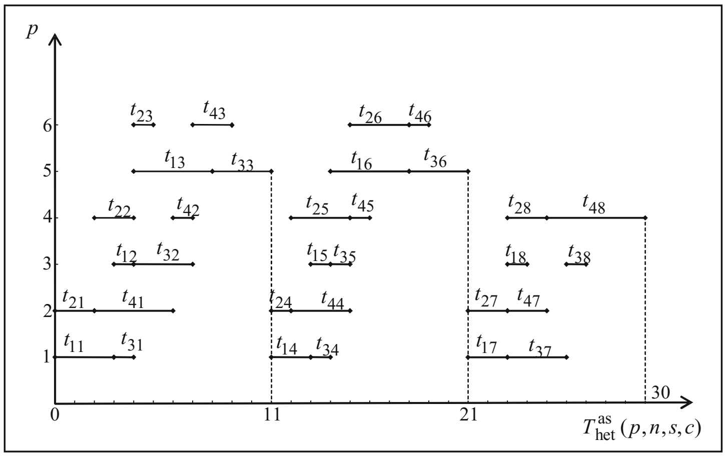

Fig. 6. Nonoverlapped Gantt chart.

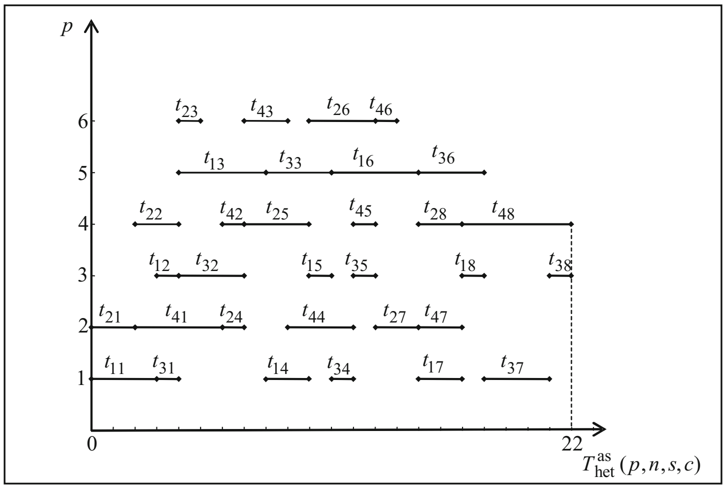

Fig. 7. Overlapped Gantt chart.

As is shown in [6], the obtained structure of the resulting overlapped Gantt chart is specified by the matrix T^* that is presented below and consists of submatrices $T_1^\varepsilon, T_2^\varepsilon, \dots, T_{k+1}^\varepsilon$. The matrix T^* takes into account all vertical and horizontal connections between blocks and also connections between blocks from different Gantt charts. We also note that the following resulting matrix T^* has size $(k+1)n \times (k+1)\left\lceil \frac{p}{c} \right\rceil$ and is considered to be a block, symmetric, upper diagonal with respect to the second diagonal, and Hankel-type matrix of order $k+1$:

$$T^* = \begin{bmatrix} T_1^\varepsilon & T_2^\varepsilon & T_3^\varepsilon & \cdots & T_k^\varepsilon & T_{k+1}^\varepsilon \\ T_2^\varepsilon & T_3^\varepsilon & T_4^\varepsilon & \cdots & T_{k+1}^\varepsilon & 0 \\ T_3^\varepsilon & T_4^\varepsilon & T_5^\varepsilon & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ T_k^\varepsilon & T_{k+1}^\varepsilon & 0 & \cdots & 0 & 0 \\ T_{k+1}^\varepsilon & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}, \tag{2}$$

where the submatrices T_l^ε and T_{k+1}^ε are of the form

$$T_l^{\varepsilon}=\begin{bmatrix} t^{\varepsilon}_{1,(l-1)\left[\frac{p}{c}\right]+1} & t^{\varepsilon}_{1,(l-1)\left[\frac{p}{c}\right]+2} & \cdots & t^{\varepsilon}_{1,l\left[\frac{p}{c}\right]} \\ t^{\varepsilon}_{2,(l-1)\left[\frac{p}{c}\right]+1} & t^{\varepsilon}_{2,(l-1)\left[\frac{p}{c}\right]+2} & \cdots & t^{\varepsilon}_{2,l\left[\frac{p}{c}\right]} \\ \vdots & \vdots & \ddots & \vdots \\ t^{\varepsilon}_{n,(l-1)\left[\frac{p}{c}\right]+1} & t^{\varepsilon}_{n,(l-1)\left[\frac{p}{c}\right]+2} & \cdots & t^{\varepsilon}_{n,l\left[\frac{p}{c}\right]} \end{bmatrix},\ l=\overline{1,k},$$

$$T_{k+1}^{\varepsilon}=\begin{bmatrix} t^{\varepsilon}_{1,k\left[\frac{p}{c}\right]+1} & t^{\varepsilon}_{1,k\left[\frac{p}{c}\right]+2} & \cdots & t^{\varepsilon}_{1,k\left[\frac{p}{c}\right]+r} & 0 & \cdots & 0 \\ t^{\varepsilon}_{2,k\left[\frac{p}{c}\right]+1} & t^{\varepsilon}_{2,k\left[\frac{p}{c}\right]+2} & \cdots & t^{\varepsilon}_{2,k\left[\frac{p}{c}\right]+r} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ t^{\varepsilon}_{n,k\left[\frac{p}{c}\right]+1} & t^{\varepsilon}_{n,k\left[\frac{p}{c}\right]+2} & \cdots & t^{\varepsilon}_{n,k\left[\frac{p}{c}\right]+r} & 0 & \cdots & 0 \end{bmatrix}.$$

With the help of the matrix T^*, we construct a c-layer vertex–weighted graph G_2^c similar to the graph G_1^c. Weights $t^{\varepsilon}_{q+(i-1)c,j}$, $q=\overline{1,c}$, $i=\overline{1,(k+1)m}$, $j=\overline{1,s}$, will be assigned to vertices of each layer q of size $(k+1)m\times(k+1)\left[\frac{p}{c}\right]$ in the graph G_2^c. Here, t^{ε}_{q1} and $t^{\varepsilon}_{q+((k+1)m-1)c,s}$ are input and output vertices, $q=\overline{1,c}$. The following theorem takes place.

THEOREM 2. The minimum total time $T^{\text{as}}_{\text{het}}(p,n,s,c)$ of execution of n, $n\geq 2$, heterogeneous distributed competing processes using the program resource linearly structured into s, $s\geq 2$, blocks with execution times of blocks taking into account additional system expenditures $\varepsilon>0$ specified by a matrix $T^{\varepsilon}=[t^{\varepsilon}_{ij}]$, $i=\overline{1,n}$, $j=\overline{1,s}$, in a multiprocessor system with p, $p\geq 2$, processors and $c\geq 2$ copies of the structured program resource ($\left[\frac{p}{c}\right]\geq 2$) in the asynchronous mode in the case of unbounded parallelism ($s>\left[\frac{p}{c}\right]$) is determined by the length of the critical path from the initial vertex t^{ε}_{q1} to the terminal vertex $t^{\varepsilon}_{q+((k+1)m-1)c,s}$, $q=\overline{1,c}$, of the graph G_2^c.

Example 3. Using the parameter values of the Gantt chart presented in Fig. 6, find the minimum total time $T^{\text{as}}_{\text{h}}(p,n,s,c)$ with the help of an algorithm for finding the critical path in the c-layer vertex–weighted graph G_2^c.

Since $\left[\frac{p}{c}\right]=3$, we have $8=s=k\left[\frac{p}{c}\right]+r=2\cdot 3+2$ and, hence, $k=2$ and $r=2$. We divide the matrix T^{ε} into submatrices T_l^{ε}, $l=\overline{1,3}$, of size 4×3. The matrix T^* has the size $(k+1)n\times(k+1)\left[\frac{p}{c}\right]=12\times 9$ and is as follows:

$$T^*=\begin{bmatrix} \begin{bmatrix}3&1&4\\2&2&1\\1&3&3\\4&1&2\end{bmatrix} & \begin{bmatrix}2&1&4\\1&3&3\\1&1&3\\3&1&1\end{bmatrix} & \begin{bmatrix}2&1&0\\2&2&0\\3&1&0\\2&5&0\end{bmatrix} \\ \begin{bmatrix}2&1&4\\1&3&3\\1&1&3\\3&1&1\end{bmatrix} & \begin{bmatrix}2&1&0\\2&2&0\\3&1&0\\2&5&0\end{bmatrix} & \begin{bmatrix}0&0&0\\0&0&0\\0&0&0\\0&0&0\end{bmatrix} \\ \begin{bmatrix}2&1&0\\2&2&0\\3&1&0\\2&5&0\end{bmatrix} & \begin{bmatrix}0&0&0\\0&0&0\\0&0&0\\0&0&0\end{bmatrix} & \begin{bmatrix}0&0&0\\0&0&0\\0&0&0\\0&0&0\end{bmatrix} \end{bmatrix}.$$

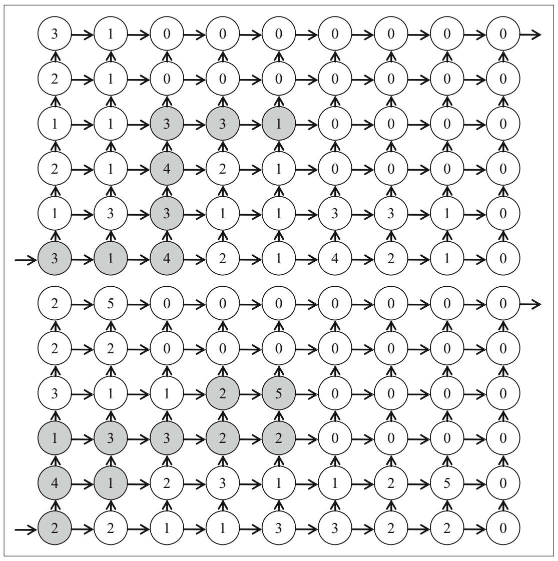

Fig. 8. Graph G_2^c.

From the matrix T^*, we construct the 2-layer vertex–weighted graph G_2^c (Fig. 8). The critical path length equals 22.

4. EXECUTION TIMES OF HOMOGENEOUS AND IDENTICALLY DISTRIBUTED COMPETING PROCESSES

According to Definition 2, a system of distributed competing processes is considered to be homogeneous if the times of execution of each block Q_j, $j=\overline{1,s}$, by each process are the same, i.e., $t_{ij}^{\varepsilon}=t_j^{\varepsilon}$, $i=\overline{1,n}$, $j=\overline{1,s}$.

Figure 9 presents the Gantt chart illustrating the execution of homogeneous distributed competing processes when the number of copies of the program resource is limited on an MS with the following parameters: $p=7$, $n=4$, $s=3$, $c=2$, and

$$T^{\varepsilon}=\begin{bmatrix}3&1&4\\3&1&4\\3&1&4\\3&1&4\end{bmatrix}.$$

Let us estimate the total execution time of n homogeneous distributed competing processes using c copies of the structured program resource in the asynchronous mode. Let $(t_1^{\varepsilon},t_2^{\varepsilon},\dots,t_s^{\varepsilon})$ be the duration of execution of each block Q_j, $j=\overline{1,s}$, of the program resource with allowance for overheads ε, $t_j^{\varepsilon}=t_j+\varepsilon$, $j=\overline{1,s}$. We denote by $T_{\varepsilon}^{s}=\sum_{j=1}^{s}t_j^{s}$ the duration of execution of the entire program resource by each process. Let us show that, under such conditions, the computation of the total time $T_{\text{hom}}^{\text{as}}(p,n,s,c)$ in the case of unbounded parallelism is reduced to the finding of the total execution time of homogeneous distributed processes for one copy of the structured program resource. When $n=mc$, $m\geq 2$, $2\leq c\leq p$, the execution of c copies of the structured program resource in the asynchronous mode is equivalent to the execution of c groups each of which consists of m processes competing for the use of one copy of the program resource on $\left[\frac{p}{c}\right]$ processors.

Based on the formula [6] for the computation of the total time of execution of n homogeneous competing processes using one copy of the structured program resource and taking into account that $n=mc$, $m\geq 2$, $2\leq c\leq p$, we obtain

$$T_{\text{hom}}^{\text{as}}(p,mc,s,c)=T_{\text{hom}}^{\text{as}}\left(\left[\frac{p}{c}\right],m,s,1\right)=T_{\varepsilon}^{s}+(m-1)\max_{1\leq j\leq s}t_j^{\varepsilon}. \tag{3}$$

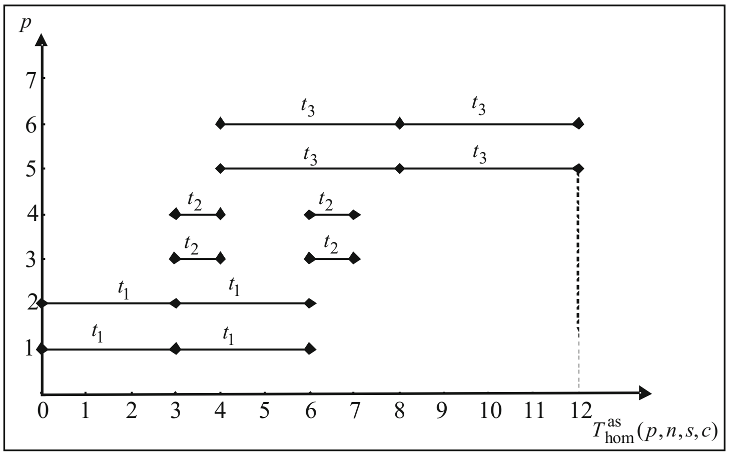

Fig. 9. Gantt chart of the asynchronous mode (homogeneous processes).

To prove formula (3), we use functional (1) of the Bellman–Johnson problem; for systems of homogeneous competing processes, this functional assume the form

$$T^{\text{as}}_{\text{hom}}(p,n,s,c) = \max_{1 \le u_1 \le u_2 \le \ldots \le u_{s-1} \le m} \left[\sum_{i=1}^{u_1} t_1^{\varepsilon} + \sum_{i=u_1}^{u_2} t_2^{\varepsilon} + \ldots + \sum_{i=u_{s-1}}^{m} t_s^{\varepsilon} \right] = \sum_{j=1}^{s} t_j^{\varepsilon} + (m-1) \max_{1 \le j \le s} t_j^{\varepsilon},$$

where $m = \frac{n}{c}$, $t_j^{\varepsilon} = t_j + \varepsilon$, $j = \overline{1,s}$, and $u_1, u_2, \ldots, u_{s-1}$ are integers.

In the case when $s > \left[\frac{p}{c}\right]$ and $s = k\left[\frac{p}{c}\right]$, $k > 1$, the matrix of execution times for blocks of the program resource is constructed by analogy with the matrix T^*. A difference is that, in each submatrix T_l^{ε}, $l = \overline{1,k}$, of the matrix T^*, all rows coincide. Then, by analogy with Theorem 2, the total time $T^{\text{as}}_{\text{hom}}(p,n,s,c)$ of execution of n distributed homogeneous competing processes when the number of copies of the program resource is limited is determined by the length of the critical path from the initial vertex to the terminal vertex of the corresponding network graph.

If $s > \left[\frac{p}{c}\right]$, $s = k\left[\frac{p}{c}\right] + r$, $k \ge 1$, $1 \le r < \left[\frac{p}{c}\right]$, then the last submatrix T_{k+1}^{ε} of the matrix T^* contains $\left[\frac{p}{c}\right] - r$ zero columns.

Definition 4. A homogeneous structurization of the program resource into s blocks with execution times $(t_1^{\varepsilon}, t_2^{\varepsilon}, \ldots, t_s^{\varepsilon})$, $\sum_{j=1}^{s} t_j^{s} = T_{\varepsilon}^{s}$, is called uniform if $t_1^{\varepsilon} = t_2^{\varepsilon} = \ldots = t_s^{\varepsilon} = t^{\varepsilon}$.

CONSEQUENCE. In the case of uniform structurization, the following formulas are used for computing the minimum total execution time of distributed competing processes when the number of copies of the program product is limited:

$$T_{\text{p}}^{\text{as}}(p,n,s,c) = \begin{cases} (m+s-1)t^{\varepsilon}, & \left[\frac{p}{c}\right] \ge \min\{m,s\}, \\ (km + \left[\frac{p}{c}\right] - 1)t^{\varepsilon}, & \left[\frac{p}{c}\right] < \min\{m,s\}, s = k\left[\frac{p}{c}\right], \ k > 1, \\ ((k+1)m + r - 1)t^{\varepsilon}, & \left[\frac{p}{c}\right] < \min\{m,s\}, s = k\left[\frac{p}{c}\right] + r, \ k \ge 1, \ 1 \le r < \left[\frac{p}{c}\right]. \end{cases}$$

Let us consider a system of identically distributed competing processes. With allowance for overheads ε for each ith process, the execution times of all blocks of the system being considered coincide and are equal to t_i^{ε}, i.e., the following chain of equalities takes place: $t_{i1}^{\varepsilon} = t_{i2}^{\varepsilon} = \ldots = t_{is}^{\varepsilon} = = t_i^{\varepsilon}$ for all $i = \overline{1,n}$.

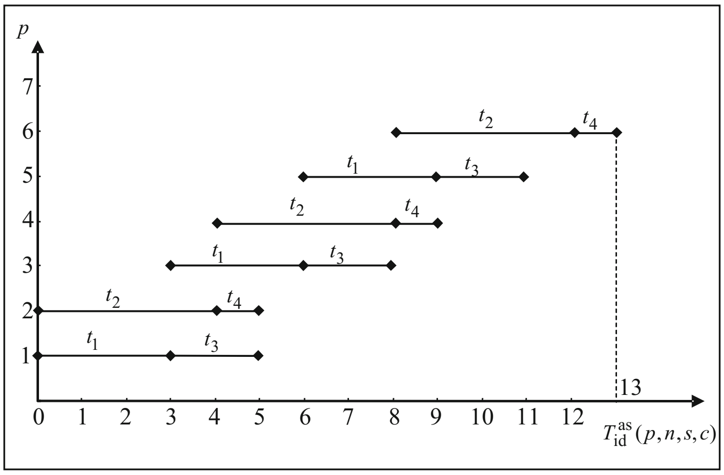

Fig. 10. Gantt chart of the asynchronous mode (identically distributed processes).

Figure 10 presents the Gantt chart illustrating the execution of identically distributed competing processes in an MS with the parameters $p=7$, $n=4$, $s=3$, $c=2$, and $T^{\varepsilon}=\begin{bmatrix}3&3&3\\4&4&4\\2&2&2\\1&1&1\end{bmatrix}$ in the case of unbounded ($s\le\left[\frac{p}{c}\right]$) parallelism.

We denote by $T_{\varepsilon}^{q}=\sum_{i=1}^{m}t_{q+(i-1)c}^{\varepsilon}$ the total time of execution of each block Q_j, $j=\overline{1,s}$, by all m processes from the qth group and by $t_{\max}^{q}=\max_{1\le i\le m}t_{q+(i-1)c}^{\varepsilon}$ the maximum execution time for a block from this group, $q=\overline{1,c}$. The following theorem is true.

THEOREM 3. The minimum total time of execution of n, $n\ge 2$, identically distributed competing processes using the program resource structured into s, $s\ge 2$, blocks on a multiprocessor system with p, $p\ge 2$, processors in the asynchronous mode when the number of copies of the program resource is limited amounts to $T_{\text{id}}^{\text{as}}(p,n,s,c)$,

$$T_{\text{op}}^{\text{as}}(p,n,s,c)=\max_{1\le q\le c}\begin{cases}T_{\varepsilon}^{q}+(s-1)t_{\max}^{q} \text{ when } s\le\left[\frac{p}{c}\right] \text{ or } s>\left[\frac{p}{c}\right], \text{ and } T_{\varepsilon}^{q}\le\left[\frac{p}{c}\right]t_{\max}^{q};\\ kT_{\varepsilon}^{q}+\left(\left[\frac{p}{c}\right]-1\right)t_{\max}^{q} \text{ when } s=k\left[\frac{p}{c}\right], k>1,\ T_{\varepsilon}^{q}>\left[\frac{p}{c}\right]t_{\max}^{q};\\ (k+1)T_{\varepsilon}^{q}+(r-1)t_{\max}^{q} \text{ when } s=k\left[\frac{p}{c}\right]+r,\ k\ge 1,\ 1\le r<\left[\frac{p}{c}\right],\ T_{\varepsilon}^{q}>\left[\frac{p}{c}\right]t_{\max}^{q}.\end{cases}$$

To prove the theorem, we first consider the case when $s\le\left[\frac{p}{c}\right]$ or when $s>\left[\frac{p}{c}\right]$ but, at the same time, $T_{\varepsilon}^{q}\le\left[\frac{p}{c}\right]t_{\max}^{q}$. We take advantage of functional (1) of the Bellman–Johnson problem, which assumes the following form for systems of identically distributed competing processes:

$$T_{\text{op}}^{\text{as}}(p,n,s,c)=\max_{1\le u_1\le u_2\le\ldots\le u_{s-1}\le m}\left[\sum_{i=1}^{u_1}t_{q+(i-1)c}^{\varepsilon}+\sum_{i=u_1}^{u_2}t_{q+(i-1)c}^{\varepsilon}+\ldots+\sum_{i=u_{s-1}}^{m}t_{q+(i-1)c}^{\varepsilon}\right]=T_{\varepsilon}^{q}+(s-1)\max_{1\le i\le m}t_{q+(i-1)c}^{\varepsilon},$$

where $m=n/c$, $t_{q+(i-1)c}^{\varepsilon}=t_{q+(i-1)c}+\varepsilon$, $q=\overline{1,c}$, $i=\overline{1,m}$, and $u_1,u_2,\ldots,u_{s-1}$ are integers.

Next, let us consider the case when $s = k\left[\frac{p}{c}\right]$, $k > 1$, and $T_\varepsilon^q > \left[\frac{p}{c}\right] t_{\max}^q$. In this case, the computation of the total time $T_{\mathrm{id}}^{\mathrm{as}}(p,n,s,c)$ with the help of the functional of the Bellman–Johnson problem leads to the formula

$$T_{\mathrm{id}}^{\mathrm{as}}(p,n,s,c) = \max_{1 \le u_1 \le u_2 \le \ldots \le u_{\left[\frac{p}{c}\right]-1} \le km} \left[\sum_{i=1}^{u_1} t_{q+(i-1)c}^{\varepsilon} + \sum_{i=u_1}^{u_2} t_{q+(i-1)c}^{\varepsilon} + \ldots + \sum_{i=u_{\left[\frac{p}{c}\right]-1}}^{km} t_{q+(i-1)c}^{\varepsilon} \right] = kT_\varepsilon^q + \left(\left[\frac{p}{c}\right] - 1 \right) t_{\max}^q .$$

Here, $t_{q+(i-1)c+km}^{\varepsilon} = t_{q+(i-1)c}^{\varepsilon}$, $q = \overline{1,c}$, $i = \overline{1,m}$.

In the case when $s = k\left[\frac{p}{c}\right] + r$, $k \ge 1$, $1 \le r < \left[\frac{p}{c}\right]$, and $T_\varepsilon^q \ge \left[\frac{p}{c}\right] t_{\max}^q$, the computation of the total time with the help of the functional of the Bellman–Johnson problem leads to the third formula for the computation of the total execution time of identically distributed competing processes when the number of copies of the program resource is limited.

CONCLUSIONS

The considered generalization of the mathematical model with one structured program resource (a conveyor) to the case of a limited number of program resources allows one to establish interrelations between multiconveyor processing and similar processing on one program conveyor. At the same time, analytical estimates of the total execution time can be obtained for competing processes under conditions of bounded parallelism, a mathematical investigation of the efficiency and optimality of the multiconveyor organization of competing processes can be carried out, potentialities of increasing the acceleration of computations can be determined, and a comparative analysis of different processing modes can be performed.

REFERENCES

1. Yu. V. Kapitonova and A. A. Letichevsky, Mathematical Theory of Computer System Design [in Russian], Nauka, Moscow (1988).
2. Yu. V. Kapitonova, N. S. Kovalenko, and P. A. Pavlov, "Optimality of systems of identically distributed competing processes," Cybernetics and Systems Analysis, Vol. 41, No. 6, 793–799 (2005).
2. Yu. V. Kapitonova, N. S. Kovalenko, and P. A. Pavlov, "Optimality of systems of identically distributed competing processes," Cybernetics and Systems Analysis, Vol. 41, No. 6, 793–799 (2005).
4. Yu. V. Kapitonova, N. S. Kovalenko, and M. I. Ovseets, "Efficiency of pipelining of competing processes when the number of program copies is limited," Cybernetics, Vol. 25, No. 3, 358–364 (1989).
5. V. P. Ivannikov, N. S. Kovalenko, and V. M. Metelsky, "Minimum time of realization of realization of distributed competitive processes in synchronous modes," Programmirovanie, No. 5, 44–52 (2000).
6. N. S. Kovalenko and V. M. Metelsky, "Execution time of competing processes in distributed processing," Cybernetics and Systems Analysis, Vol. 32, No. 1, 41–49 (1996).
7. N. S. Kovalenko and P. A. Pavlov, "Optimality of systems of identically distributed competing processes under conditions of bounded parallelism," Dokl. NAN Belarusi, Ser. Fiz.-Mat. Navuk, No. 2, 25–29 (2006).
8. P. A. Pavlov, "Analysis of modes of organization of identically distributed competing processes," Vestnik BGU, Ser. 1, No. 1, 116–120 (2006).