IV international scientific conference
Stockholm. Sweden
19-20.09.2023

Scientific conferences

THEORETICAL AND PRACTICAL PERSPECTIVES OF MODERN SCIENCE

Proceedings of the I International Scientific and Practical Conference

19-20 September 2023

STOCKHOLM. SWEDEN

2023

UDC 001.1

BBC 1

IV International Scientific and Practical Conference «Theoretical and practical perspectives of modern science», September 19-20, 2023, Stockholm. Sweden. 42 p.

ISBN 978-91-65423-34-3

DOI https://doi.org/10.5281/zenodo.8376505

Publisher: «SC. Scientific conferences»

Main organization: artmedia24

Editor: Hans Muller

Layout: Ellen Schwimmer

The sample of the citation for publication is Sevda Aghayeva Aydin kizi, Aynur Rzayeva Elman kizi FAMOUS AZERBAİJAN WOMEN IV International Scientific and Practical Conference «Theoretical and practical perspectives of modern science», September 19-20, 2023, Stockholm. Sweden. Pp.19-22, URL: https://sconferences.com

Contact information

Website: https://sconferences.com

E-mail: info@sconferences.com

Content

Biological sciences

Economic sciences

Mathematical sciences

Philological sciences

Psychological sciences

Technical sciences

Matehmatical sciences

UDC 004

ASYNCHRONOUS MODE OF DISTRIBUTED COMPUTING WITH A LIMITED NUMBER OF COPIES OF A PROGRAM RESOURCE

P.A. Pavlov
PhD, Associate professor
Polessky state university, Republic of Belarus, Pinsk
N.S. Kovalenko
Dr. Sc., Professor
Belarusian state university, Republic of Belarus, Minsk

Abstract
The mathematical model of distributed computations is constructed, the problems of finding the minimum execution time of heterogeneous, homogeneous and equally distributed processes competing for the use of a limited number of copies of a software resource in asynchronous mode in cases of unlimited and limited parallelism in the number of processors of a multiprocessor system are solved. In this case the ideas of structuring a program resource into linearly ordered blocks with their further pipelining over processes and processors of a multiprocessor system are used.

Keywords: *multiprocessor system, structuring, pipelining, process, software resource, Gantt diagram, heterogeneous system, Bellman-Johnson functional, asynchronous mode, unlimited parallelism, limited parallelism, vertex-weighted graph, homogeneous system, equally distributed system.*

1. Introduction

Distributed computing is a promising and dynamically developing area of parallelism organisation. This term is usually used in case of parallel data processing on a set of processing devices remote from each other and in which data transmission via communication lines leads to significant time delays. The listed conditions are typical, for example, when organising colculations in multi-machine computing complexes, computing clusters, multiprocessor systems (MS), local or global networks. Creation of high-performance multiprocessor systems and computing complexes is characterised by a wide penetration of fundamental principles of parallelisation and pipelining of computations into hardware and software. In this regard there is a process of revision of mathematical methods and algorithms of solving problems in various subject areas up to revision of the whole algorithmic baggage of applied mathematics, new requirements to construction and research of mathematical models concerning various aspects of parallel and pipeline organisation of computations are put forward. The necessity and urgency of research in these directions is also connected with the fact that the principles of structuring, parallelisation and pipelining are rather general and inherent in processes of different nature [1-4].

The case when there is a single copy of a programme resource (PR) in the shared memory of the MS has been studied from different points of view in works [5-14]. The problems of finding the minimum total execution time of distributed competing processes using the PR structured into blocks in different modes of processes interaction, processors and blocks were solved; a comparative analysis of modes of processes interaction, processors and blocks was carried out; criteria of efficiency and optimality of software resource structuring were obtained; a number of optimisation problems on calculation of the number of processes, processors, etc. were solved. The study of these and other problems relating to the optimal organisation of distributed parallel computations becomes especially important in the case when a limited number of copies of a program resource can be simultaneously placed in the shared memory of MS. Such generalisation is of principal character because it reflects the main features of multiconveyor processing and also allows us to compare the efficiency of conveyor and parallel processing.

2. Mathematical model of distributed computing

Constructive elements for building mathematical models that implement distributed computing methods are the concepts of process and software resource. We will consider a process as a sequence of *sets of commands (procedures, blocks, subroutines)* $I_s = (1,\ 2,\ \ldots,\ s)$. *Processes that affect each other's behaviour by exchanging information are called cooperative processes. A programme or its part repeatedly executed in a multiprocessor system will be called a software resource, and a set of corresponding processes will be called competing processes.*

By solving the problem of software resource allocation between processes, as a consequence, the problems of efficient use of the main computing resources - processors, RAM, I/O channels, etc. - are implicitly solved. From this point of view, a software resource is an integrated means of requesting computational resources. On the other hand, effectively solving the problem of software resource allocation we solve the problem of reducing the time of realisation of given volumes of calculations.

The mathematical model of a distributed processing system of competing cooperative processes under a limited number of copies of a software resource includes:

- $p \ge 2$, *processors of a multiprocessor system that have access to shared memory;*
- $n \ge 2$, *distributed competing processes;*
- $s \ge 2$, *blocks of a software resource structured into blocks;*
- *matrix* $T = [t_{ij}]$, $i = \overline{1,n}$, $j = \overline{1,s}$, *execution times of the software resource blocks by distributed cooperating competing processes;*
- $2 \le c \le p$, *the number of copies of the software resource structured into blocks that can be simultaneously located in the RAM available for all processors;*
- $\theta > 0$ – *is a parameter characterising the time of additional system costs associated with the organisation of the conveyor mode of using blocks of a structured software resource by a set of interacting competing processes during distributed processing.*

We will also assume that the number of blocks of the programme resource $s \le p_c = [p/c]$, *where* $[x]$ – *where is the integer part of the number, where is the integer part of the number, and the number of processes* n *is a multiple of the number of copies* c *of the structured software resource, i.e.* $n = mc$, $m \ge 2$, *and that the interaction of processes, processors and blocks of the software resource is subject to the following conditions:*

1) no processor can process more than one block at a time;

2) processes are executed in parallel-conveyor mode in groups, i.e. simultaneous (parallel) execution c *of copies of each block in combination with pipelining of a group* c *of copies of the* Q_j *–th block by processes and processors, where* $j = \overline{1,s}$;

3) processing of each block of the software resource is carried out without interruptions;

4) distribution of blocks Q_j, $j = \overline{1,s}$, *of the programme resource to processors for each of the processes* $i = c(l-1)+q$, $l = \overline{1,m}$, $q = \overline{1,c}$, *where* $m = n/c$, *is performed cyclically according to the rule: the block with number* j *is distributed to the processor with number* $c(j-1)+q$.

Let's introduce the following modes of conveyor realisation of interaction of processes, processors and blocks taking into account the presence of software resource copies.

The asynchronous mode of interaction between processes, processors and blocks of a structured software resource implies that the beginning of execution of a copy of the next block Q_j, $j = \overline{1,s}$, *is determined by the presence* c *of processors and the readiness of the block copy for execution, while a software block is considered ready for execution if it is not executed on any of the processors*

Fig. 1 is a linear Gantt chart illustrating the asynchronous mode of execution $n = 4$ *of distributed competing processes using* $c = 2$ *copies of a software resource structured into* $s = 3$ *blocks in a multiprocessor system with* $p = 7$ *processors and a given matrix of block execution times taking into account additional system costs* $T^{\theta} = [t_{ij}^{\theta}]_{4\times3} = [t_{ij} + \theta]_{4\times3}$.

Under these assumptions, we consider solving the problems of obtaining mathematical relations for calculating the minimum total realisation time of a set of distributed competing processes under a limited number of copies of a software resource for the asynchronous mode.

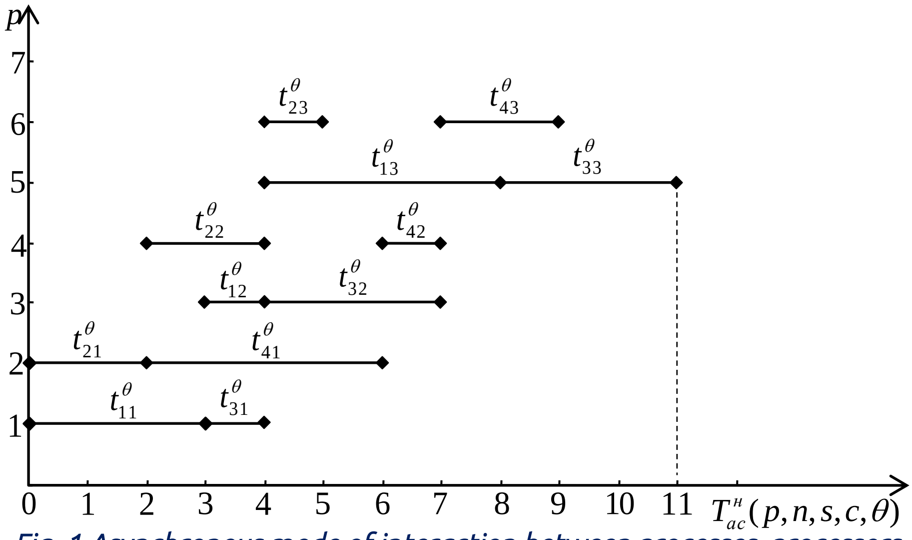


Fig. 1 Asynchronous mode of interaction between processes, processors and blocks of a software resource

3. Execution time of heterogeneous processes in conditions of unlimited parallelism

Let us consider the asynchronous mode of interaction between processes, processors and blocks of a structured software resource (Fig. 1).

Definition 1. A system n of distributed competing processes is called heterogeneous if the execution times of software resource blocks Q_1, Q_2, ..., Q_s depend on the volumes of processed data and/or their structure, i.e. different for different processes.

Let us denote the minimum total execution time n of heterogeneous distributed competing processes using c copies of a software resource structured into s blocks with a matrix of execution times T^{θ} in a multiprocessor system with p processors taking into account additional system costs θ in asynchronous mode by $T^{h}_{as}(p,n,s,c,\theta)$. For the computation $T^{h}_{as}(p,n,s,c,\theta)$, consider the cases of unbounded, i.e. $s \le p_c$, and bounded, when $s > p_c$, parallelism.

Let there be a system $n = mc$ of heterogeneous distributed competing processes, where $m \ge 2$, and $2 \le c \le p$, and the number of blocks s of the structured software resource does not exceed the number of groups of processors in each, i.e. $2 \le s \le p_c$. In this case, without restriction of generality, we can assume that each Q_j-th block, $j = \overline{1,s}$, of the i-th process, where $i = c(l-1)+q$, $l = \overline{1,m}$, $q = \overline{1,c}$, will be executed on the $(c(j-1)+q)$-th processor. Then it is enough to use cp_c processors to execute all n processes, and the rest $p - cp_c$ will be unused.

Let $T^{\theta} = [t^{\theta}_{ij}]$ – be the $n \times s$ –matrix of times of execution of blocks of the structured software resource by each of the i processes taking into account the parameter $\theta > 0$, where $t^{\theta}_{ij} = t_{ij} + \theta$, $i = \overline{1,n}$, $j = \overline{1,s}$. To calculate the minimum total time $T^{h}_{as}(p,n,s,c,\theta)$ we can use the functional of the Bellman-Johnson problem, which in our case will have the form:

$$T^{h}_{as}(p,n,s,c,\theta) = \max_{1\le q\le c}\left(\max_{1\le u_1 \le u_2 \le \ldots \le u_{s-1} \le m} \left[\sum_{i=1}^{u_1} t^{\theta}_{c(i-1)+q,1} + \sum_{i=u_1}^{u_2} t^{\theta}_{c(i-1)+q,2} + \ldots + \sum_{i=u_{s-1}}^{m} t^{\theta}_{c(i-1)+q,s} \right] \right), \quad (1)$$

where u_1, u_2, ..., u_{s-1} – are the integers.

Example 1. Let us consider the interpretation of formula (1) on a numerical example. Let $p=7$, $n=6$, $s=3$, $c=2$, *and the execution times of the processes of the blocks of the structured PR with regard to the parameter are given* θ *by the matrix*

$$T^{\theta}=\begin{bmatrix} t_{11}^{\theta}=3 & t_{12}^{\theta}=1 & t_{13}^{\theta}=4 \\ t_{21}^{\theta}=2 & t_{22}^{\theta}=2 & t_{23}^{\theta}=1 \\ t_{31}^{\theta}=1 & t_{32}^{\theta}=3 & t_{33}^{\theta}=3 \\ t_{41}^{\theta}=4 & t_{42}^{\theta}=1 & t_{43}^{\theta}=2 \\ t_{51}^{\theta}=3 & t_{52}^{\theta}=2 & t_{53}^{\theta}=1 \\ t_{61}^{\theta}=1 & t_{62}^{\theta}=4 & t_{63}^{\theta}=1 \end{bmatrix}.$$

Fig. 2 shows a Gantt chart illustrating the functioning of the distributed heterogeneous system according to the data of Example 1.

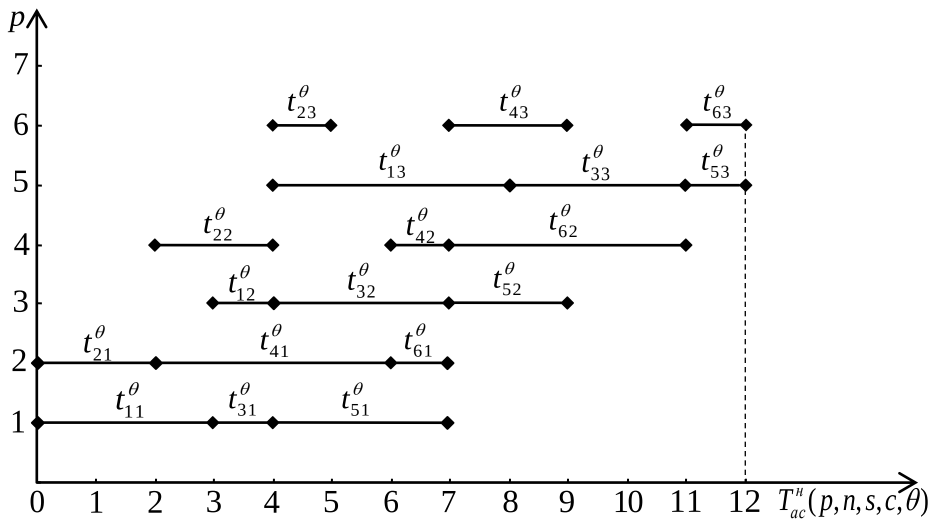


Fig. 2 Asynchronous mode of execution of heterogeneous processes at $2 \le s \le p_c$

Since $n=6$, *and* $c=2$, *then* $m=3$, *therefore, the functional (1) will take the form:*

$$T_{as}^{h}(7,6,3,2,\theta)=\max_{1\le q\le 2}\left(\max_{1\le u_1\le u_2\le 3}\left[\sum_{i=1}^{u_1} t_{2(i-1)+q,1}^{\theta}+\sum_{i=u_1}^{u_2} t_{2(i-1)+q,2}^{\theta}+\sum_{i=u_2}^{3} t_{2(i-1)+q,3}^{\theta}\right]\right)=$$

$$=[u_1=1 \Rightarrow u_2=\overline{1,3},\ u_1=2 \Rightarrow u_2=\overline{2,3},\ u_1=3 \Rightarrow u_2=3]=$$

$$= \max_{1\le q\le 2}\left(\max\left[\begin{array}{l} \sum_{i=1}^{1} t^{\theta}_{2(i-1)+q,1} + \sum_{i=1}^{1} t^{\theta}_{2(i-1)+q,2} + \sum_{i=1}^{3} t^{\theta}_{2(i-1)+q,3}, \\ \sum_{i=1}^{1} t^{\theta}_{2(i-1)+q,1} + \sum_{i=1}^{2} t^{\theta}_{2(i-1)+q,2} + \sum_{i=2}^{3} t^{\theta}_{2(i-1)+q,3}, \\ \sum_{i=1}^{u_1} t^{\theta}_{2(i-1)+q,1} + \sum_{i=u_1}^{3} t^{\theta}_{2(i-1)+q,2} + \sum_{i=3}^{3} t^{\theta}_{2(i-1)+q,3}, \\ \sum_{i=1}^{2} t^{\theta}_{2(i-1)+q,1} + \sum_{i=2}^{2} t^{\theta}_{2(i-1)+q,2} + \sum_{i=2}^{3} t^{\theta}_{2(i-1)+q,3}, \\ \sum_{i=1}^{2} t^{\theta}_{2(i-1)+q,1} + \sum_{i=2}^{3} t^{\theta}_{2(i-1)+q,2} + \sum_{i=3}^{3} t^{\theta}_{2(i-1)+q,3}, \\ \sum_{i=1}^{3} t^{\theta}_{2(i-1)+q,1} + \sum_{i=3}^{3} t^{\theta}_{2(i-1)+q,2} + \sum_{i=3}^{3} t^{\theta}_{2(i-1)+q,3} \end{array}\right]\right) =$$

$$= \max\left[\begin{array}{ll} \underline{t^{\theta}_{11}+t^{\theta}_{12}+t^{\theta}_{13}+t^{\theta}_{33}+t^{\theta}_{53}}, & t^{\theta}_{11}+t^{\theta}_{12}+t^{\theta}_{32}+t^{\theta}_{33}+t^{\theta}_{53}, \\ t^{\theta}_{11}+t^{\theta}_{12}+t^{\theta}_{32}+t^{\theta}_{52}+t^{\theta}_{53}, & t^{\theta}_{11}+t^{\theta}_{31}+t^{\theta}_{32}+t^{\theta}_{33}+t^{\theta}_{53}, \\ t^{\theta}_{11}+t^{\theta}_{31}+t^{\theta}_{32}+t^{\theta}_{52}+t^{\theta}_{53}, & t^{\theta}_{11}+t^{\theta}_{31}+t^{\theta}_{51}+t^{\theta}_{52}+t^{\theta}_{53}, \\ t^{\theta}_{21}+t^{\theta}_{22}+t^{\theta}_{23}+t^{\theta}_{43}+t^{\theta}_{63}, & t^{\theta}_{21}+t^{\theta}_{22}+t^{\theta}_{42}+t^{\theta}_{43}+t^{\theta}_{63}, \\ t^{\theta}_{21}+t^{\theta}_{22}+t^{\theta}_{42}+t^{\theta}_{62}+t^{\theta}_{63}, & t^{\theta}_{21}+t^{\theta}_{41}+t^{\theta}_{42}+t^{\theta}_{43}+t^{\theta}_{63}, \\ \underline{t^{\theta}_{21}+t^{\theta}_{41}+t^{\theta}_{42}+t^{\theta}_{62}+t^{\theta}_{63}}, & \underline{t^{\theta}_{21}+t^{\theta}_{41}+t^{\theta}_{61}+t^{\theta}_{62}+t^{\theta}_{63}} \end{array}\right] = 12 .$$

Consequently, the minimum total execution time $n=6$ of heterogeneous distributed interacting competing processes using $c=2$ copies of the program resource structured on $s=3$ blocks in a multiprocessor system with $p=7$ processors in asynchronous mode, taking into account the parameter θ of additional system costs associated with the organization of the conveyor mode of using blocks, will be $T^{h}_{as}(7,6,3,2,\theta)=12$, which coincides with the time on the Gantt chart (Fig. 4). In this case 6 processors will be involved.

In [9, 13] we consider an algorithm that allows us to solve the problem of determining the minimum total execution time of heterogeneous distributed competing processes in asynchronous mode with a limited number of copies of a structured program resource much more efficiently using the apparatus of vertex-weighted graphs. According to the given number of blocks of PR s, the number of its copies c and the matrix of block execution times $T^{\theta}=[t^{\theta}_{c(i-1)+q,j}]$, $i=\overline{1,m}$, $q=\overline{1,c}$, $j=\overline{1,s}$, a c-layer vertex-weighted graph is constructed G^{1}_{ac}. Each q layer (subgraph) of the graph G^{1}_{ac}, $q=\overline{1,c}$, will consist of vertices $t^{\theta}_{c(i-1)+q,j}$, $i=\overline{1,m}$, $j=\overline{1,s}$, which are located in the nodes of the rectangular $m\times s$-lattice, with t^{θ}_{q1} - input vertices and $t^{\theta}_{c(m-1)+q,s}$ - output vertices. The arcs in each layer q of the graph reflect the linear order of execution of program resource blocks Q_j, $j=\overline{1,s}$, by processes $i=c(l-1)+q$, $l=\overline{1,m}$, $q=\overline{1,c}$, on processors with numbers $c(j-1)+q$. The following theorem holds.

T h e o r e m 1. The minimum total execution time of $n=mc$, $m\ge 2$, heterogeneous distributed processes using $2\le c\le p$ copies of structured $s\ge 2$ block PR in MS with $p\ge 2$ processors with block execution time matrix taking into account additional system costs $[t^{\theta}_{ij}]$, $i=\overline{1,n}$, $j=\overline{1,s}$, in asynchronous *mode in the case when* $s \le p_c$*, is determined by the length of the critical path in the* c*-layer vertex-weighted graph* G_{ac}^1 *from the initial vertex* t_{q1}^{θ} *to the final vertex* $t_{c(m-1)+q,s}^{\theta}$*,* $q = \overline{1,c}$ *(Fig. 3).*

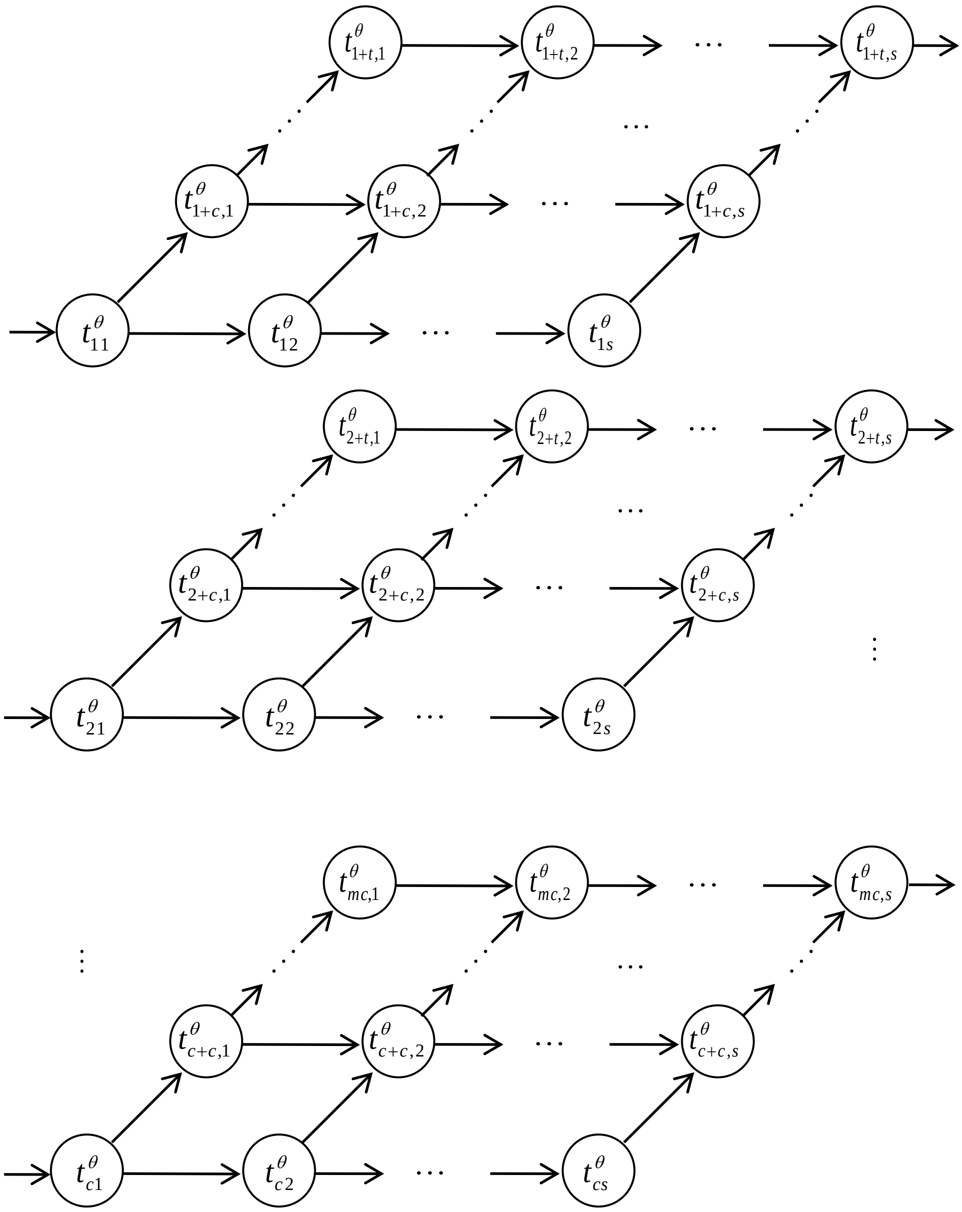


Fig. 3 Vertex-weighted graph G_{ac}^1

Example 2. On the data from Example 1, find using $T_{as}^h(p,n,s,c,\theta)$ *the algorithm to find the critical path in graph* G_{ac}^1*.*

Given $m = n/c = 3$*,* $s = 3$*, and a matrix* T^{θ}*, we construct a 2-layer vertex-weighted graph* G_{ac}^1 *(Fig. 4).*

Each layer contains 3×3 *vertices. The length of the critical path in the graph is 12, which coincides with the minimum total time of Example 1.*

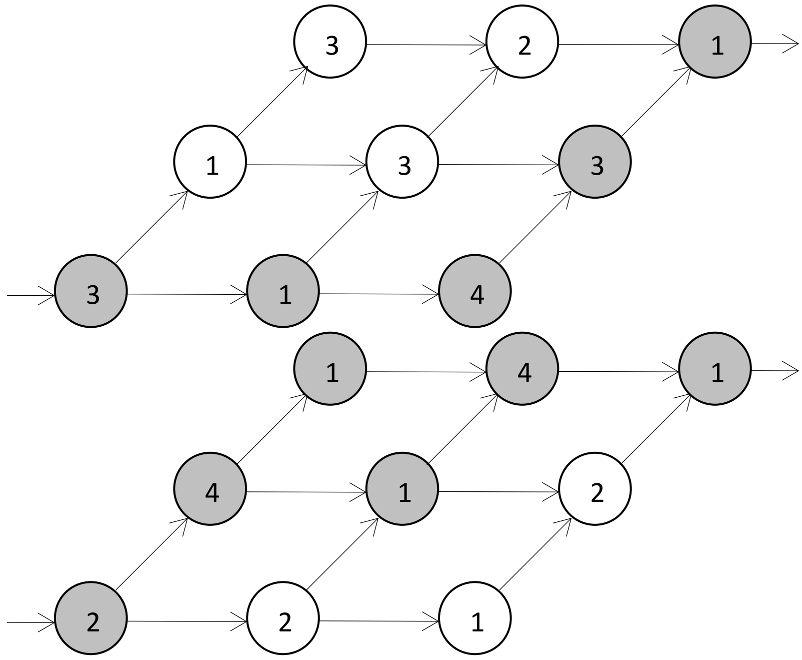


Fig. 4 2-lager vertex-weighted graph G^1_{ac}

4. Asynchronous mode of distributed processes execution under conditions of limited parallelism

Consider the case of limited parallelism, i.e. when the number of blocks of a structured program resource s *is greater than the number of groups* c *of processors in each, i.e.* $s > p_c$, $s = kp_c + r$, $k \geq 1$, $1 \leq r < p_c$.

As in the case where there is a single copy of the program resource in the MS shared memory, we will split the entire s *set of blocks into* $(k+1)$ *groups, each containing a* p_c *block(s), except for the last one* p_c, *which will contain only* r *the PR block(s). Then the matrix of block execution times* $T^\theta = [t^\theta_{ij}]$, $i = \overline{1,n}$, $j = \overline{1,s}$, *will be decomposed into a* $(k+1)$ *submatrix* T^θ_φ *of the form (2) of dimension* $n \times p_c$ *each,* $\varphi = \overline{1,k+1}$. *Moreover, the latter matrix* T^θ_{k+1} *will contain* s *at non-multiples* p_c *only* r *non-zero columns, and the remaining columns* $p_c - r$ *will be zero [9].*

$$T^\theta_\varphi = \begin{bmatrix} t^\theta_{1,(\varphi-1)p_c+1} & t^\theta_{1,(\varphi-1)p_c+2} & \cdots & t^\theta_{1,\varphi p_c} \\ t^\theta_{2,(\varphi-1)p_c+1} & t^\theta_{2,(\varphi-1)p_c+2} & \cdots & t^\theta_{2,\varphi p_c} \\ \vdots & \vdots & \ddots & \vdots \\ t^\theta_{n,(\varphi-1)p_c+1} & t^\theta_{n,(\varphi-1)p_c+2} & \cdots & t^\theta_{n,\varphi p_c} \end{bmatrix}, \ \varphi = \overline{1,k},\ T^\theta_{k+1} = \begin{bmatrix} t^\theta_{1,kp_c+1} & t^\theta_{1,kp_c+2} & \cdots & t^\theta_{1,kp_c+r} & 0 & \cdots & 0 \\ t^\theta_{2,kp_c+1} & t^\theta_{2,kp_c+2} & \cdots & t^\theta_{2,kp_c+r} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ t^\theta_{n,kp_c+1} & t^\theta_{n,kp_c+2} & \cdots & t^\theta_{n,kp_c+r} & 0 & \cdots & 0 \end{bmatrix}. \quad (2)$$

For each of the submatrices T^θ_φ, $\varphi = \overline{1,k+1}$, *we construct a linear Gantt chart, which will reflect the execution time of the next* p_c *blocks of the structured software resource on the* cp_c *processors by* n *all processes using a limited number of copies of the structured software resource* c, *and the* $(k+1)$ *-th chart will reflect the execution of the last* r *blocks on the* cr *processors.*

E x a m p l e 2. Construct linear Gantt charts for MS with parameters: $p = 7$, $n = 4$, $s = 8$, $c = 2$,

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

Figure 5 shows the unaligned Gantt charts.

The total execution time $n=4$ of the processes using $c=2$ copies of the PR will in this case be defined as the sum of the lengths of the critical paths in each of the consecutive unbiased Gantt charts. However, this time can be reduced, if we sequentially combine Gantt diagrams, starting from the second diagram, along the time axis from right to left by the maximum possible value without violating the technological conditions of asynchronous mode. As a result of overlapping we obtain the resulting combined Gantt chart (Fig. 6).

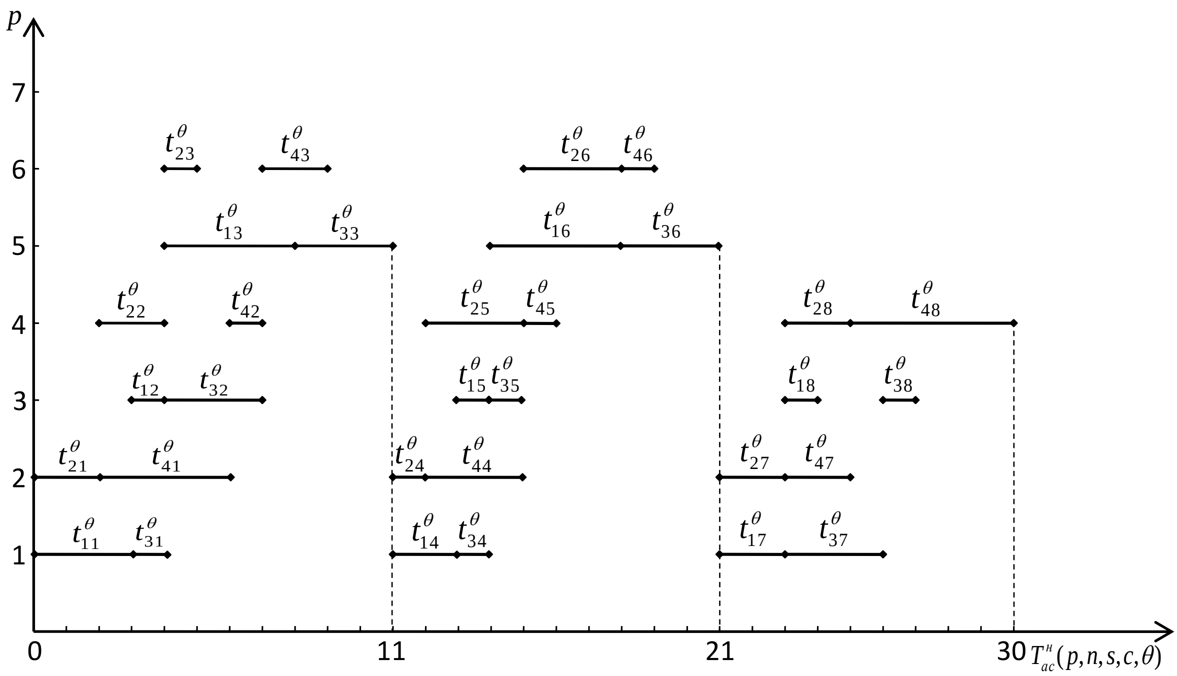


Fig. 5. Unbiased Gantt charts under limited parallelism

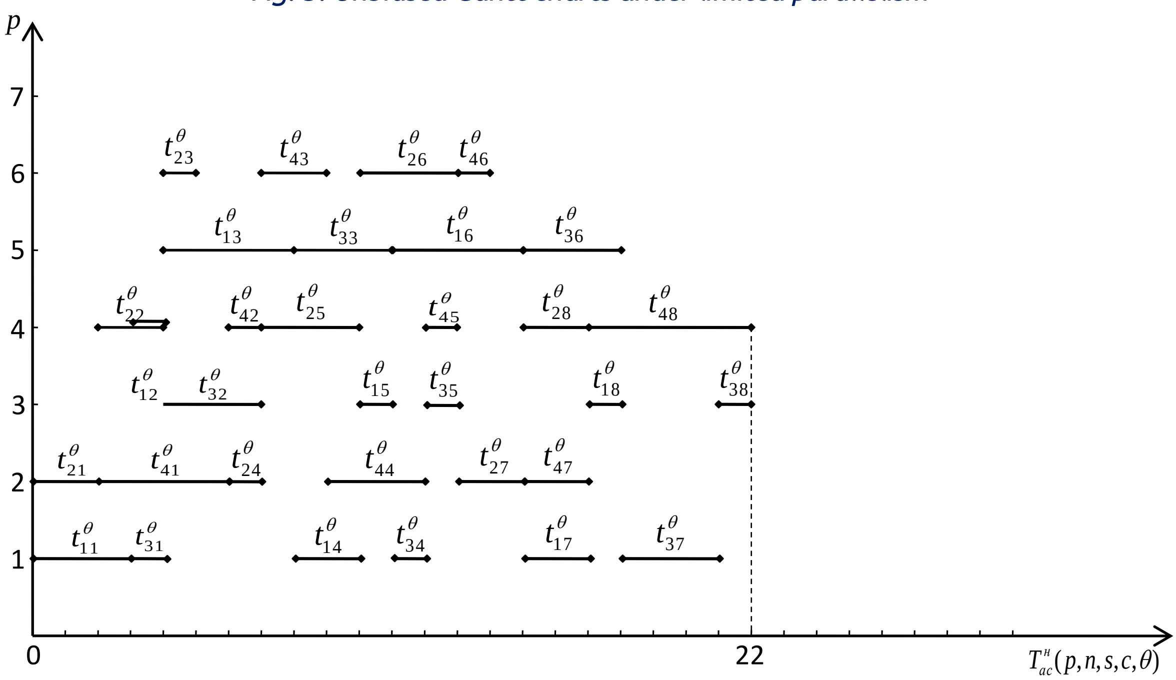


Fig. 6. Combined Gantt charts

The obtained structure of the resulting combined Gantt chart will be defined by a matrix T of the form (3), which consists of submatrices T_1^θ, T_2^θ, ..., T_{k+1}^θ. The matrix T takes into account both horizontal and vertical links between blocks, as well as linkages between blocks from different Gantt charts. Note also that the resulting matrix T will have dimension $(k+1)n \times (k+1)p_c$, $k=\lceil cs/p \rceil$, will be block, upper diagonal, symmetrical about the main diagonal of Hankel type.

$$T=\begin{bmatrix} T_1^\theta & T_2^\theta & T_3^\theta & \cdots & T_k^\theta & T_{k+1}^\theta \\ T_2^\theta & T_3^\theta & T_4^\theta & \cdots & T_{k+1}^\theta & 0 \\ T_3^\theta & T_4^\theta & T_5^\theta & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ T_k^\theta & T_{k+1}^\theta & 0 & \cdots & 0 & 0 \\ T_{k+1}^\theta & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}. \qquad (3)$$

By the matrix T *we construct a* c *-layer vertex-weighted graph* G_{ac}^2 *similar to the graph* G_{ac}^1. *The vertices of each subgraph* q *of the* G_{ac}^2 *graph will be assigned weights* $t_{c(i-1)+q,j}^{\theta}$, $q=\overline{1,c}$, $i=\overline{1,(k+1)m}$, $j=\overline{1,(k+1)p_c}$, $k=[cs/p]$. *The vertices* t_{q1}^{θ} *will be the input vertices, and* $t_{c((k+1)m-1)+q,(k+1)p_c}^{\theta}$ *will be the output vertices,* $q=\overline{1,c}$. *The following theorem holds.*

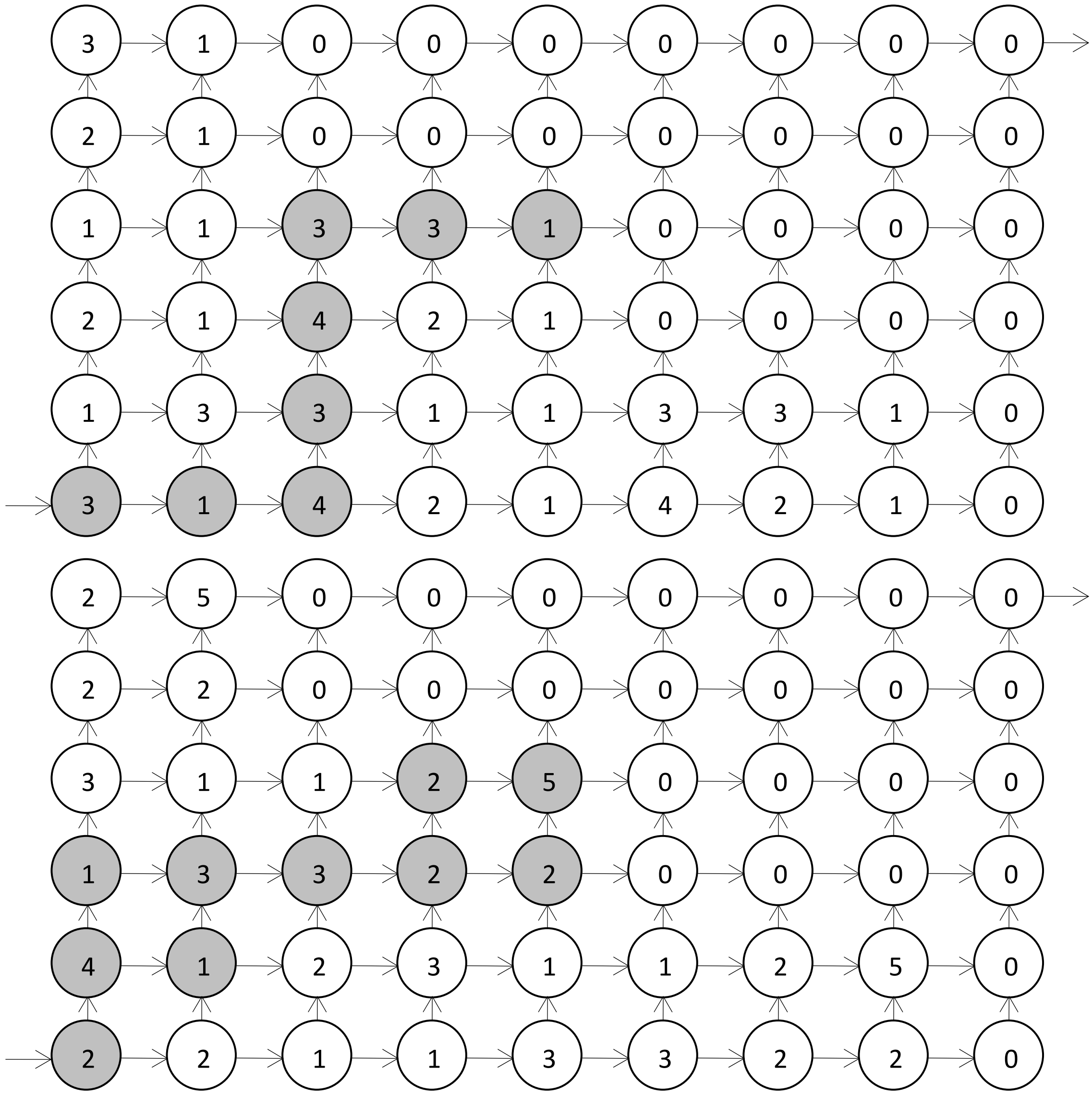
Theorem 2. Minimum total execution time of $T_{as}^h(p,n,s,c,\theta)$ *heterogeneous* n, $n\geq 2$, *distributed competing processes using linearly structured on* s, $s\geq 2$, *blocks program resource with block execution times taking into account additional system costs* $\theta>0$ *given by the matrix* $T^\theta=[t_{ij}^\theta]$, $i=\overline{1,n}$, $j=\overline{1,s}$, *in a multiprocessor system with* p, $p\geq 2$, *processors and* $c\geq 2$ *copies of the structured program resource in asynchronous mode in the case of limited parallelism, i.e. when* $s>p_c$, *is determined by the length of the critical path from the initial vertex* t_{q1}^θ *to the final vertex* $t_{c((k+1)m-1)+q,(k+1)p_c}^{\theta}$, $q=\overline{1,c}$, *of the graph* G_{ac}^2.

Example 3. Using the algorithm for finding the critical path in a c *-layer vertex-weighted graph* G_{ac}^2, *find the values* $T_{as}^h(p,n,s,c,\theta)$ *of the MS parameters of example 2.*

Since $s=k\left[\dfrac{p}{c}\right]+r=2\cdot 3+2$, *then and* $k=2$, $r=2$. *We partition the matrix* T^θ *into submatrices* T_φ^θ, $\varphi=\overline{1,3}$, *of dimension* 4×3 *each. The matrix* T *will be of dimension and have the form:*

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

We construct a 2-layer vertex-weighted graph G_{ac}^2 *by the matrix* T *:*



Consequently, the minimum total execution time of heterogeneous distributed competing processes is determined by the length of the critical path , which coincides with the time in fig. 6.

6. Conclusion

This paper generalises the mathematical model with a single software resource to the case of a limited number of software resources, which allows us to establish interrelations between multiconveyor processing and similar processing with a single PR, to obtain analytical estimates of the total execution time of competing processes with a sufficient number of processors and limited parallelism for the asynchronous mode. The obtained results can be used in the study of synchronous modes of interaction of processes, processors and blocks of a structured software resource, in the comparative analysis of different modes of distributed computations, in the mathematical study of efficiency and optimality of multiconveyor organisation of computations in solving the problems of building the optimal layout of blocks of a software resource and finding the optimal number of processors providing the directive time of execution of given amounts of computations, etc. The results can be used in the study of synchronous modes of interaction of processes, processors and blocks of a structured software resource.

References

1. Kovalenko N.S., Pavlov P.A. Organization of synchronous execution of parallel distributed processes on flow lines / N.S. Kovalenko, P.A. Pavlov // Problems of forecasting and state regulation of socio-economic development: Proceedings of the IX International Scientific Conference, Minsk, October 16-17, 2008. – Minsk: Research Institute of the Ministry of Economy of the Republic of Belarus, 2008. – PP. 482–496.

2. Kaplun V.V., Pavlov P.A., Shtepa V.N. Resource-process approach to building a mathematical model of a microenergy system / V.V. Kaplun, P.A. Pavlov, V.N. Shtepa // Bulletin of Kyiv National University of Technology and Design. – 2016. – №2. – C. 48–60.

3. Zaiets N., Shtepa V., Pavlov P., Elperin I., Hachkovska M. Development of a resource-process approach to increasing the efficiency of electrical equipment for food production / N. Zaiets, V. Shtepa, P. Pavlov, I. Elperin, M. Hachkovska // Eastern-European Journal of Enterprise Nechnologies. – 2019. – №8 (101). – C. 59–65.

4. Kaplun V.V., Pavlov P.A., Shtepa V.N., Prokopenya O.N. Resource-process model of energy management of a local facility with several energy sources / V.V. Kaplun, P.A. Pavlov, V.N. Shtepa, O.N. Prokopenya // Bulletin of Brest State Technical University. – 2019. – №4 (117). – PP. 86–91.

5. Pavlov P.A. Optimality of program resources structuring at pipeline distributed processing / P.A. Pavlov // Software products and systems. – 2010. – №3.– PP. 79–85.

6. Pavlov P.A. Efficiency of distributed computing in scalable systems / P.A. Pavlov // Scientific and Technical Vedomosti SPbSPU: Informatics. Telecommunications. Management. – 2010. – №1.– PP. 83–89.

7. Pavlov P.A. Organization of homogeneous competing processes in distributed conveyor processing / P.A. Pavlov // Management issues. – 2010. – №3.– PP. 66–75.

8. Pavlov P.A. Scalable distributed systems of competing interacting processes and their optimality / P.A. Pavlov // Bulletin of S.P. Korolev Samara Aerospace University (National Research University). – 2010. – №1.– PP. 234–242.

9. Pavlov P.A., Коваленко Н.С. Distributed computing with a limited number of copies of a software resource / P.A. Pavlov, N.S. Kovalenko // Software products and systems. – 2011. – №4.– PP. 155–163.

10. Pavlov P.A. The problem of optimizing the number of processors in distributed processing / P.A. Pavlov // Bulletin of the State Samara Academician S.P. Korolev Aerospace University (National Research University). – 2011. – №4.– PP. 230–240.

11. Pavlov P.A. The optimality of software resources structuring through the pipeline distributed processing of competitive cooperative processes / P.A. Pavlov // International Journal of Multimedia Technology (IJMT). – 2012. – Vol.2, №1. – PP. 5–10.

12. Kovalenko N.S., Pavlov P.A., Ovseec M.I. Problems of optimizing the number of processors and building the optimal layout of distributed systems / N.S. Kovalenko, P.A. Pavlov, M.I. Ovseec // Bulletin of BSU. Series 1: Physics. Mathematics. Informatics. – 2012. – №1. – PP. 119–126.

13. Kovalenko N.S., Pavlov P.A., Ovseec M.I. Asynchronous distributed computations with a limited number of copies of a structured program resource / N.S. Kovalenko, P.A. Pavlov, M.I. Ovseec // Cybernetics and systems analysis. – 2012. – Vol.48, №1. – PP. 86–98.

14. Kovalenko N.S., Pavlov P.A. Optimal Grouping Algorithm of Identically Distributed Systems / N.S. Kovalenko, P.A. Pavlov // Programming and Computer Software. – 2012. – Vol.38, №3. – PP. 143–150.