

АСИНХРОННЫЕ РАСПРЕДЕЛЕННЫЕ ВЫЧИСЛЕНИЯ ПРИ ОГРАНИЧЕННОМ ЧИСЛЕ КОПИЙ ПРОГРАММНОГО РЕСУРСА

П.А. Павлов

Полесский государственный университет

Республика Беларусь, г.Пинск, ул. Днепровской Флотилии, 23

E-mail: p.a.pavlov@mail.ru

Решены задачи нахождения минимального общего времени выполнения распределенных конкурирующих процессов при ограниченном числе копий программного ресурса в условиях неограниченного и ограниченного параллелизма.

THE ASYNCHRONOUS DISTRIBUTED CALCULATIONS AT THE LIMITED NUMBER OF COPIES OF THE PROGRAM RESOURCE / P.A. Pavlov (Polesky state university, st. Dneprovskoy Flotilii 23, Pinsk, Belarus). In work problems of a finding of minimum general time of performance of the distributed competing processes are solved at the limited number of copies of a program resource in the conditions of the unlimited and limited parallelism.

Введение. Во многих приложениях, связанных с оптимальной организацией параллельных вычислений в многопроцессорных системах (МС) и вычислительных комплексах (ВК), значительный интерес представляют задачи, когда множество конкурирующих процессов могут использовать не одну, а несколько копий структурированного программного ресурса (ПР). Случай, когда в общей памяти многопроцессорной системы имеется одна копия ПР, с различных точек зрения был изучен в работах [1–11]. При этом были решены задачи определения минимального общего времени выполнения распределенных конкурирующих процессов, использующих структурированный на блоки программный ресурс в различных режимах взаимодействия процессов, процессоров и блоков [1–3], получены критерии эффективности и оптимальности структурирования программных ресурсов [4], проведен сравнительный анализ режимов взаимодействия процессов, процессоров и блоков [5–6], решен ряд оптимизационных задач по расчету числа процессов, минимального числа процессоров и др. [7–11]. Изучение этих и других задач, относящихся к оптимальной организации параллельных вычислений, приобретает особую актуальность в случае, когда в общей памяти МС могут быть одновременно размещены $c \geq 2$ копий программного ресурса. Такое обобщение носит принципиальный характер в виду того, что отражает основные черты мультиконвейерной обработки, а также позволяет сравнить эффективность конвейерной и параллельной обработки.

В данной работе строится и исследуется математическая модель организации конкурирующих процессов, использующих ограниченное число копий программного ресурса. При этом, используя идеи метода структурирования программных ресурсов на блоки с их последующей конвейеризацией по процессам и процессорам, исследуются оптимальные временные характеристики такой организации.

1. Математическая модель распределенных вычислений при ограниченном числе копий программного ресурса. Конструктивными элементами для построения математических моделей систем распределенных вычислений являются понятия процесса и программного ресурса.

Процесс будем рассматривать как последовательность блоков (команд, процедур) Q_1, Q_2, \dots, Q_s , для выполнения которых используется множество процессоров (процессорных узлов, обрабатывающих устройств, интеллектуальных клиентов). При этом процесс называется *распределённым*, если все блоки или часть из них обрабатываются разными процессорами. Для ускорения выполнения процессы могут обрабатываться параллельно, взаимодействуя путем обмена информацией. Такие процессы называются *кооперативными* или *взаимодействующими* процессами.

Понятие *ресурса* используется для обозначения любых объектов вычислительной системы, которые могут быть использованы процессами для своего выполнения. *Рентерабельные (многократно используемые)* ресурсы характеризуются возможностью одновременного использования несколькими вычислительными процессами. Для параллельных систем характерной является ситуация, когда одну и ту же последовательность блоков или ее часть необходимо процессорам выполнять многократно, такую последовательность будем называть *программным ресурсом*, а множество соответствующих процессов – *конкурирующими*.

Математическая модель распределенной обработки конкурирующих взаимодействующих процессов при ограниченном числе копий программного ресурса включает в себя: p , $p \geq 2$, процессоров многопроцессорной системы, которые имеют доступ к общей памяти; n , $n \geq 2$, распределенных конкурирующих процессов; s , $s \geq 2$, блоков структурированного на блоки программного ресурса; матрицу $T = [t_{ij}]$, $i = \overline{1, n}$, $j = \overline{1, s}$, времен выполнения блоков программного ресурса распределенными взаимодействующими конкурирующими процессами; $2 \leq c \leq p$, число копий структурированного на блоки программного ресурса, которые могут одновременно находиться в оперативной памяти, доступной для всех p процессоров, причем $\left\lceil \frac{p}{c} \right\rceil \geq 2$; $\varepsilon > 0$ – параметр, характеризующий время дополнительных системных расходов, связанных с организацией конвейерного режима использования блоков структурированного программного ресурса множеством взаимодействующих конкурирующих процессов при распределенной обработке.

Будем также предполагать, что число процессов n кратно числу копий c структурированного программного ресурса, т.е. $n = mc$, $m \geq 2$, и что взаимодействие процессов, процессоров и блоков программного ресурса подчинено следующим условиям:

- 1) ни один из процессоров не может обрабатывать одновременно более одного блока;
- 2) процессы выполняются в параллельно-конвейерном режиме группами, т.е. осуществляется одновременное (параллельное) выполнение c копий каждого блока в сочетании с конвейеризацией групп из c блоков по процессорам и процессам;
- 3) обработка каждого блока программного ресурса осуществляется без прерываний;
- 4) распределение блоков программного ресурса по процессорам для каждого из процессов $i = lc + q$, $i = \overline{1, n}$, $l \geq 0$, $q = \overline{1, c}$, осуществляется циклически по правилу:

блок с номером $j = k \left\lceil \frac{p}{c} \right\rceil + r$, $j = \overline{1, s}$, $k \geq 0$, $r = \overline{1, \left\lceil \frac{p}{c} \right\rceil}$, распределяется на процессор с номером $q + c(r - 1)$.

Введем следующие режимы взаимодействия процессов, процессоров и блоков с учетом наличия c копий программного ресурса:

- *асинхронный режим*, при котором начало выполнения очередной группы из c копий блока Q_j , $j = \overline{1, s}$, определяется наличием c процессоров и готовностью этой группы блоков к выполнению (программный блок считается готовым к выполнению, если он не выполняется ни на одном из процессоров);
- *первый синхронный режим*, обеспечивающий линейный порядок выполнения блоков программного ресурса внутри каждого из процессов без задержек, т.е. для каждого процесса $i = lc + q$, $i = \overline{1, n}$, $l \geq 0$, $q = \overline{1, c}$, момент завершения выполнения j -го блока на $(q + c(r - 1))$ -м процессоре совпадает с моментом начала выполнения следующего $(j + 1)$ -го блока на $(q + cr)$ -м процессоре, $j = \overline{1, s - 1}$, $r = 1, \left\lceil \frac{p}{c} \right\rceil$;
- *второй синхронный режим*, при котором c копий каждого блока непрерывно переходит по группам из c процессов, т.е. момент окончания обработки c копий

текущего блока совпадает с моментом начала их обработки на следующей группе из c процессоров.

На рис. 1–3 представлены диаграммы Ганта, иллюстрирующие выполнение $n = 4$ распределенных конкурирующих процессов, использующих $c = 2$ копии структурированного программного ресурса в МС с $p = 7$ процессорами в рассмотренных выше режимах и с заданной матрицей времен выполнения блоков ПР с учетом дополнительных системных

$$\text{расходов } T^{\varepsilon} = \begin{bmatrix} 3 & 1 & 4 \\ 2 & 2 & 1 \\ 1 & 3 & 3 \\ 4 & 1 & 2 \end{bmatrix}.$$

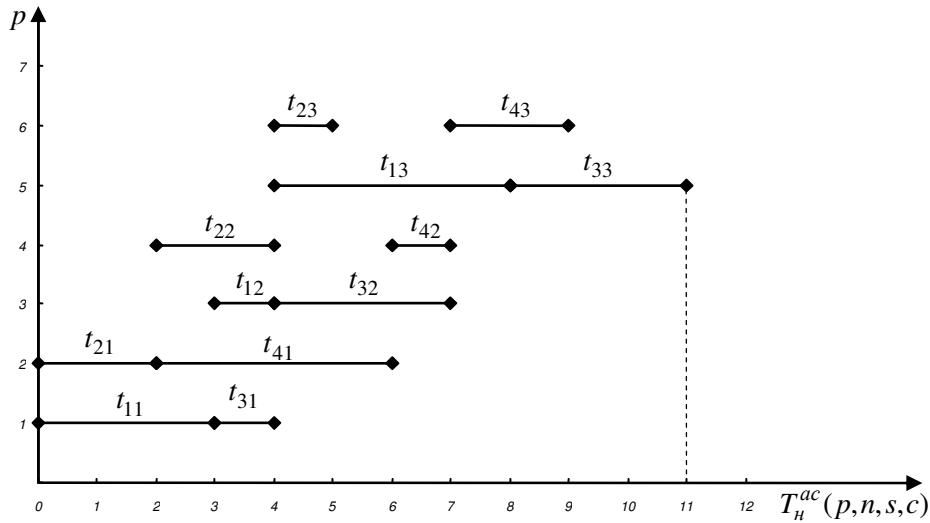


Рис.1. Асинхронный режим

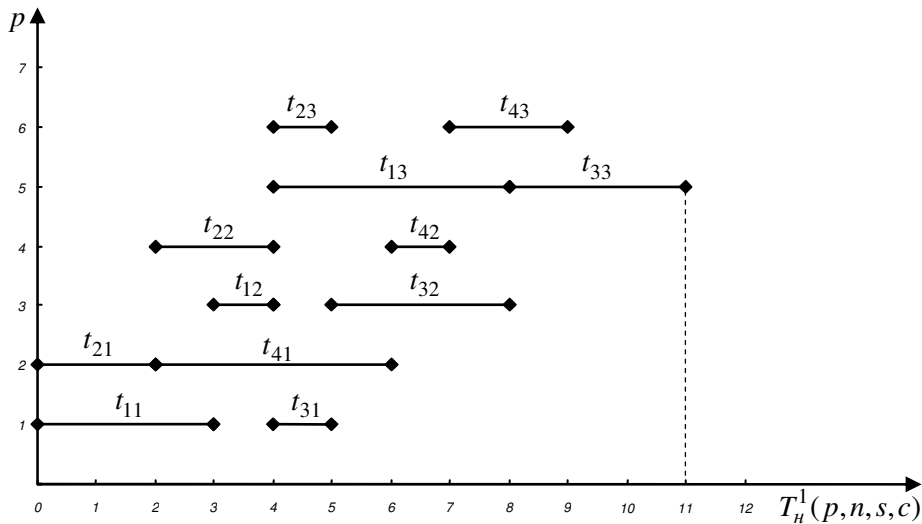


Рис. 2. Первый синхронный режим

Определение 1. Система n распределенных конкурирующих процессов называется *неоднородной*, если времена выполнения блоков программного ресурса Q_1, Q_2, \dots, Q_s зависят от объемов обрабатываемых данных и/или их структуры, т. е. разные для разных процессов.

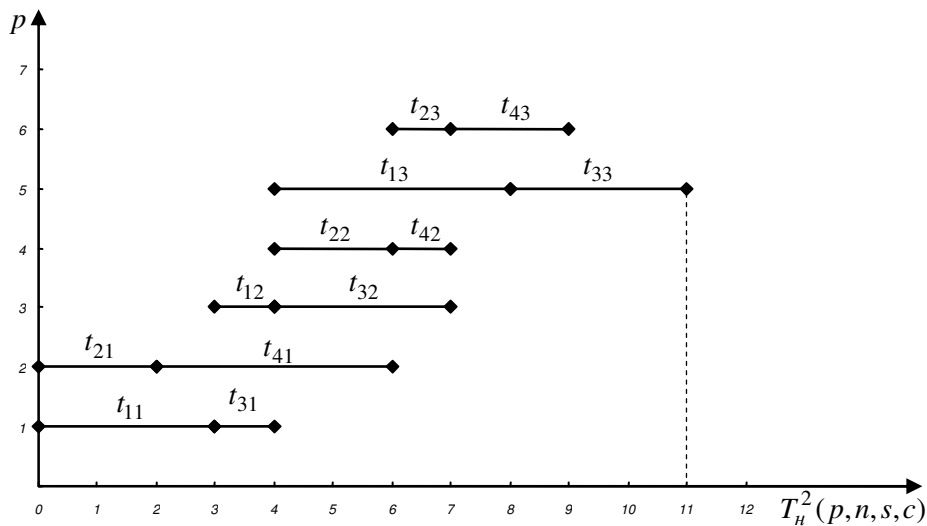


Рис. 3. Второй синхронный режим

Определение 2. Систему распределенных конкурирующих процессов будем называть *однородной*, если времена выполнения Q_j -го блока каждым из i -х процессов равны, т. е. $t_{ij} = t_j$, $i = \overline{1, n}$, $j = \overline{1, s}$.

Определение 3. Систему конкурирующих процессов будем называть *одинаково распределенной*, если времена t_{ij} выполнения блоков Q_j , $j = \overline{1, s}$, программного ресурса каждым из i -х процессов совпадают и равны t_i для всех $i = \overline{1, n}$, т. е. справедлива цепочка равенств $t_{i1} = t_{i2} = \dots = t_{is} = t_i$ для всех $i = \overline{1, n}$.

Заметим, что для системы однородных процессов в матрице времен выполнения блоков конкурирующими процессами будут одинаковы все строки, а для одинаково распределенной системы – столбцы.

2. Минимальное общее время выполнения неоднородных распределенных процессов в асинхронном режиме при достаточном числе процессоров. Обозначим минимальное общее время выполнения n неоднородных распределенных конкурирующих процессов при ограниченном числе c копий программного ресурса в многопроцессорной системе с p процессорами в асинхронном режиме, с учетом параметра ε , через $T_n^{ac}(p, n, s, c)$.

Для вычисления $T_n^{ac}(p, n, s, c)$ рассмотрим случаи *неограниченного* $\left(s \leq \left\lfloor \frac{p}{c} \right\rfloor\right)$ и *ограниченного* $\left(s > \left\lfloor \frac{p}{c} \right\rfloor\right)$ параллелизма.

Пусть имеется система $n = mc$, $m \geq 2$, $2 \leq c \leq p$, неоднородных распределенных конкурирующих процессов, причем число блоков s структурированного программного ресурса не превосходит числа групп процессоров по c процессоров в каждой, т. е. $2 \leq s \leq \left\lfloor \frac{p}{c} \right\rfloor$.

В этом случае без ограничения общности можно считать, что каждый Q_j -й, $j = \overline{1, s}$, блок i -го процесса, где $i = lc + q$, $i = \overline{1, n}$, $l \geq 0$, $q = \overline{1, c}$, закреплен за $(q + c(r - 1))$ -м процессором, $r = 1, \left\lfloor \frac{p}{c} \right\rfloor$. Тогда для выполнения n процессов достаточно взять $p = \left\lfloor \frac{p}{c} \right\rfloor s$ процессоров, а остальные $p - \left\lfloor \frac{p}{c} \right\rfloor s$ процессоров будут не задействованы.

Пусть $T^\varepsilon = [t_{ij}^\varepsilon]$ – $n \times s$ -матрица времен выполнения блоков программного ресурса каждым из i -х процессов с учетом параметра $\varepsilon > 0$, где $t_{ij}^\varepsilon = t_{ij} + \varepsilon$, $i = \overline{1, n}$, $j = \overline{1, s}$. Для вычисления минимального общего времени $T_n^{ac}(p, n, s, c)$ можно воспользоваться функционалом задачи Беллмана–Джонсона, который в нашем случае будет иметь вид:

$$T_n^{ac}(p, n, s, c) = \max_{1 \leq u_1 \leq u_2 \leq \dots \leq u_{s-1} \leq m} \left[\sum_{i=1}^{u_1} t_{q+(i-1)c,1}^\varepsilon + \sum_{i=u_1}^{u_2} t_{q+(i-1)c,2}^\varepsilon + \dots + \sum_{i=u_{s-1}}^m t_{q+(i-1)c,s}^\varepsilon \right], \quad (1)$$

где $m = \frac{n}{c}$, $t_{q+(i-1)c,j}^\varepsilon = t_{q+(i-1)c,j} + \varepsilon$, $q = \overline{1, c}$, $i = \overline{1, m}$, $j = \overline{1, s}$, а u_1, u_2, \dots, u_{s-1} – целые числа.

Пример 1. Рассмотрим интерпретацию формулы (1) на числовом примере. Пусть $p = 7$, $n = 6$, $s = 3$, $c = 2$, а времена выполнения блоков процессами заданы матрицей

$$T^\varepsilon = \begin{bmatrix} 3 & 1 & 4 \\ 2 & 2 & 1 \\ 1 & 3 & 3 \\ 4 & 1 & 2 \\ 3 & 2 & 1 \\ 1 & 4 & 1 \end{bmatrix}. \text{ Тогда } m = 3, \text{ следовательно, функционал (1) примет вид:}$$

$$T_n^{ac}(p = 7, n = 6, s = 3, c = 2) = \max_{1 \leq u_1 \leq u_2 \leq 3} \left[\sum_{i=1}^{u_1} t_{q+(i-1)2,1}^\varepsilon + \sum_{i=u_1}^{u_2} t_{q+(i-1)2,2}^\varepsilon + \sum_{i=u_2}^3 t_{q+(i-1)2,3}^\varepsilon \right], \quad q = \overline{1, 2}.$$

1) При $q = 1$, имеем:

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

Если $u_1 = 1$, $u_2 = \overline{1, 3}$, если $u_1 = 2$, $u_2 = 2, 3$, если $u_1 = 3$, $u_2 = 3$, тогда:

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

$$\begin{aligned}
&= \max \begin{bmatrix} \overline{t_{1,1}^\varepsilon + t_{1,2}^\varepsilon + t_{1,3}^\varepsilon + t_{3,3}^\varepsilon + t_{5,3}^\varepsilon}, & \overline{t_{1,1}^\varepsilon + t_{1,2}^\varepsilon + t_{3,2}^\varepsilon + t_{3,3}^\varepsilon + t_{5,3}^\varepsilon}, \\ \overline{t_{1,1}^\varepsilon + t_{1,2}^\varepsilon + t_{3,2}^\varepsilon + t_{5,2}^\varepsilon + t_{5,3}^\varepsilon}, & \overline{t_{1,1}^\varepsilon + t_{3,1}^\varepsilon + t_{3,2}^\varepsilon + t_{3,3}^\varepsilon + t_{5,3}^\varepsilon}, \\ \overline{t_{1,1}^\varepsilon + t_{3,1}^\varepsilon + t_{3,2}^\varepsilon + t_{5,2}^\varepsilon + t_{5,3}^\varepsilon}, & \overline{t_{1,1}^\varepsilon + t_{3,1}^\varepsilon + t_{5,1}^\varepsilon + t_{5,2}^\varepsilon + t_{5,3}^\varepsilon} \end{bmatrix} = \max \begin{bmatrix} \overline{3+1+4+3+1}, & \overline{3+1+3+3+1}, \\ \overline{3+1+3+2+1}, & \overline{3+1+3+3+1}, \\ \overline{3+1+3+2+1}, & \overline{3+1+3+2+1} \end{bmatrix} = \\
&= \max[12,11,10,11,10,10] = 12.
\end{aligned}$$

2) При $q = 2$, имеем:

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

Если $u_1 = 1$, $u_2 = \overline{1,3}$, если $u_1 = 2$, $u_2 = 2,3$, $u_1 = 3$, $u_2 = 3$, тогда:

$$\begin{aligned}
\max_{1 \leq u_1 \leq u_2 \leq 3} \left[\sum_{i=1}^{u_1} t_{2i,1}^\varepsilon + \sum_{i=u_1}^{u_2} t_{2i,2}^\varepsilon + \sum_{i=u_2}^3 t_{2i,3}^\varepsilon \right] &= \max \begin{bmatrix} \sum_{i=1}^1 t_{2i,1}^\varepsilon + \sum_{i=1}^2 t_{2i,2}^\varepsilon + \sum_{i=1}^3 t_{2i,3}^\varepsilon, & \sum_{i=1}^1 t_{2i,1}^\varepsilon + \sum_{i=1}^2 t_{2i,2}^\varepsilon + \sum_{i=2}^3 t_{2i,3}^\varepsilon, \\ \sum_{i=1}^1 t_{2i,1}^\varepsilon + \sum_{i=1}^3 t_{2i,2}^\varepsilon + \sum_{i=3}^3 t_{2i,3}^\varepsilon, & \sum_{i=1}^2 t_{2i,1}^\varepsilon + \sum_{i=2}^2 t_{2i,2}^\varepsilon + \sum_{i=2}^3 t_{2i,3}^\varepsilon, \\ \sum_{i=1}^2 t_{2i,1}^\varepsilon + \sum_{i=2}^3 t_{2i,2}^\varepsilon + \sum_{i=3}^3 t_{2i,3}^\varepsilon, & \sum_{i=1}^3 t_{2i,1}^\varepsilon + \sum_{i=3}^3 t_{2i,2}^\varepsilon + \sum_{i=3}^3 t_{2i,3}^\varepsilon \end{bmatrix} = \\
&= \max \begin{bmatrix} \overline{t_{2,1}^\varepsilon + t_{2,2}^\varepsilon + t_{2,3}^\varepsilon + t_{4,3}^\varepsilon + t_{6,3}^\varepsilon}, & \overline{t_{2,1}^\varepsilon + t_{2,2}^\varepsilon + t_{4,2}^\varepsilon + t_{4,3}^\varepsilon + t_{6,3}^\varepsilon}, \\ \overline{t_{2,1}^\varepsilon + t_{2,2}^\varepsilon + t_{4,2}^\varepsilon + t_{6,2}^\varepsilon + t_{6,3}^\varepsilon}, & \overline{t_{2,1}^\varepsilon + t_{4,1}^\varepsilon + t_{4,2}^\varepsilon + t_{4,3}^\varepsilon + t_{6,3}^\varepsilon}, \\ \overline{t_{2,1}^\varepsilon + t_{4,1}^\varepsilon + t_{4,2}^\varepsilon + t_{6,2}^\varepsilon + t_{6,3}^\varepsilon}, & \overline{t_{2,1}^\varepsilon + t_{4,1}^\varepsilon + t_{6,1}^\varepsilon + t_{6,2}^\varepsilon + t_{6,3}^\varepsilon} \end{bmatrix} = \max \begin{bmatrix} \overline{2+2+1+2+1}, & \overline{2+2+1+2+1}, \\ \overline{2+2+1+4+1}, & \overline{2+4+1+2+1}, \\ \overline{2+4+1+4+1}, & \overline{2+4+1+4+1} \end{bmatrix} = \\
&= \max[8,8,10,10,12,12] = 12.
\end{aligned}$$

Следовательно, минимальное общее время выполнения $n=6$ неоднородных распределенных взаимодействующих конкурирующих процессов, использующих $c=2$ копии структурированного на $s=3$ блока программного ресурса, в многопроцессорной системе с $p=7$ процессорами в асинхронном режиме составит $T_n^{ac}(p,n,s,c)=12$. При этом будет использовано 6 процессоров.

Рассмотрим алгоритм, который позволяет решить задачу определения минимального общего времени $T_n^{ac}(p,n,s,c)$ выполнения неоднородных распределенных конкурирующих процессов в асинхронном режиме значительно эффективнее.

По заданным $s, c, m = \frac{n}{c}$ и матрице $T^\varepsilon = [t_{q+(i-1)c,j}^\varepsilon]$, $q = \overline{1,c}$, $i = \overline{1,m}$, $j = \overline{1,s}$, строим c -слойный вершинно-взвешенный граф G_1^c . Каждый q -й, $q = \overline{1,c}$, слой графа G_1^c состоит из вершин $t_{q+(i-1)c,j}^\varepsilon$, $i = \overline{1,m}$, $j = \overline{1,s}$, которые расположены в узлах прямоугольной $m \times s$ -решетки, причем t_{q1}^ε – входные вершины, $t_{q+t,s}^\varepsilon$ – выходные, $q = \overline{1,c}$, $t = (m-1)c$ (рис. 4). Дуги в каждом слое q отражают линейный порядок выполнения блоков Q_j , $j = \overline{1,s}$, программного ресурса каждым из $(q+(i-1)c)$ -м процессом, $q = \overline{1,c}$, $i = \overline{1,m}$, а также линейный порядок

использования каждого блока всеми m процессами, Таким образом, имеет место следующая теорема.

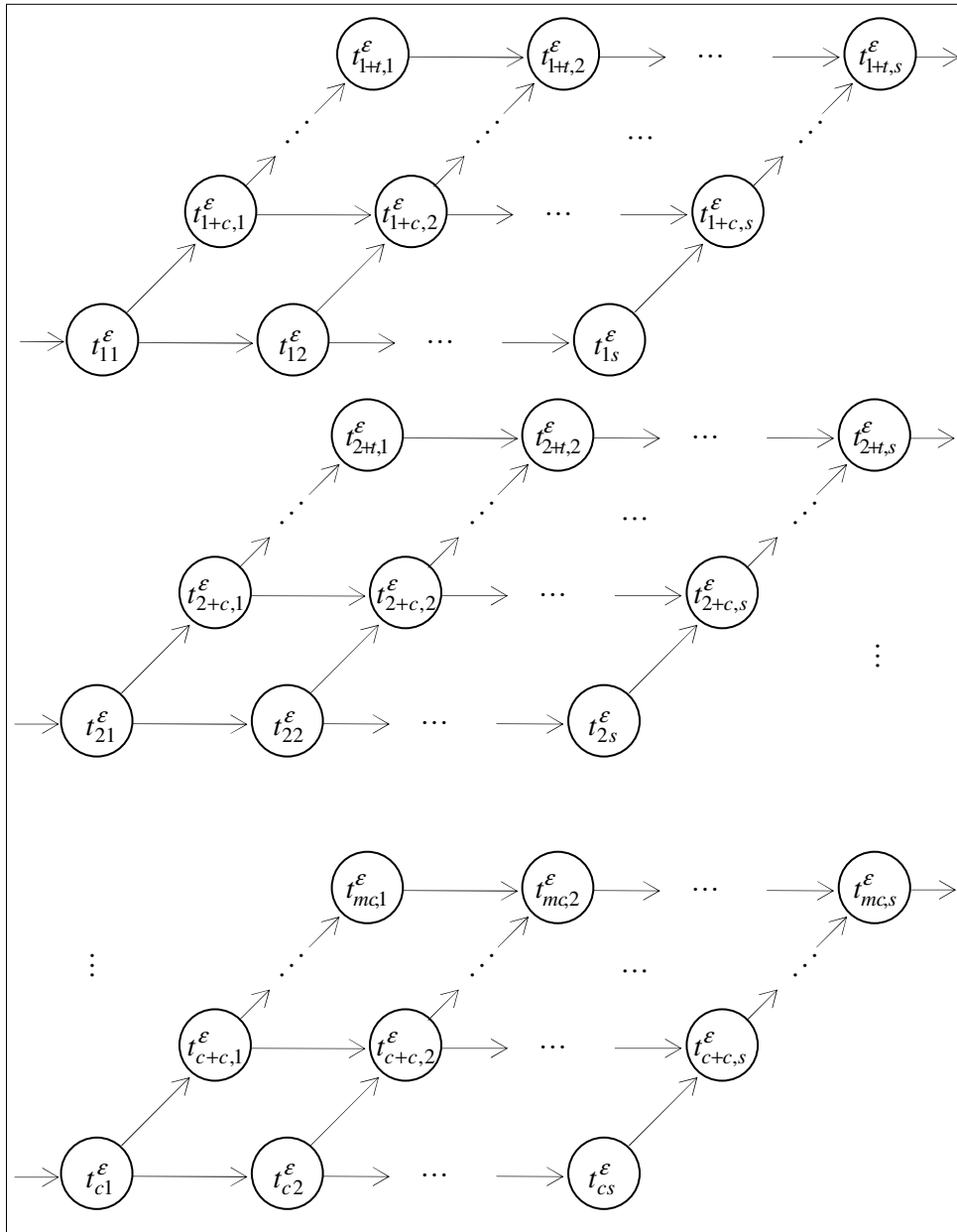


Рис. 4. Граф G_1^c

Теорема 1. Минимальное общее время выполнения $n = tc$, $t \geq 2$, неоднородных распределенных конкурирующих процессов, использующих $2 \leq c \leq p$ копии структурированного на $2 \leq s \leq \left\lceil \frac{p}{c} \right\rceil$ блока программного ресурса с временами выполнения блоков с учетом дополнительных системных расходов $[t_{ij}^\epsilon]$, $i = \overline{1, n}$, $j = \overline{1, s}$, в многопроцессорной системе с p , $\left\lceil \frac{p}{c} \right\rceil \geq 2$, процессорами в асинхронном режиме определяется длиной критического пути в c -слойном вершинно-взвешенном графе G_1^c из начальной вершины t_{q1}^ϵ в конечную $t_{q+(m-1)c,s}^\epsilon$, $q = \overline{1, c}$, $m = \frac{n}{c}$.

Пример 2. Используя данные примера 1, найти минимальное общее время $T_n^{ac}(p, n, s, c)$, используя алгоритм нахождения критического пути в c -слойном вершинно-взвешенном графе G_1^c .

По заданным $n=6$, $s=3$, и матрице T^ε строим 2-слойный ($c=2$) вершинно-взвешенный граф G_1^c (рис. 5). Каждый слой содержит ms вершин, где $m=n/c=3$. Длина критического пути в графе равна 12, что совпадает с минимальным общим временем, полученном в примере 1.

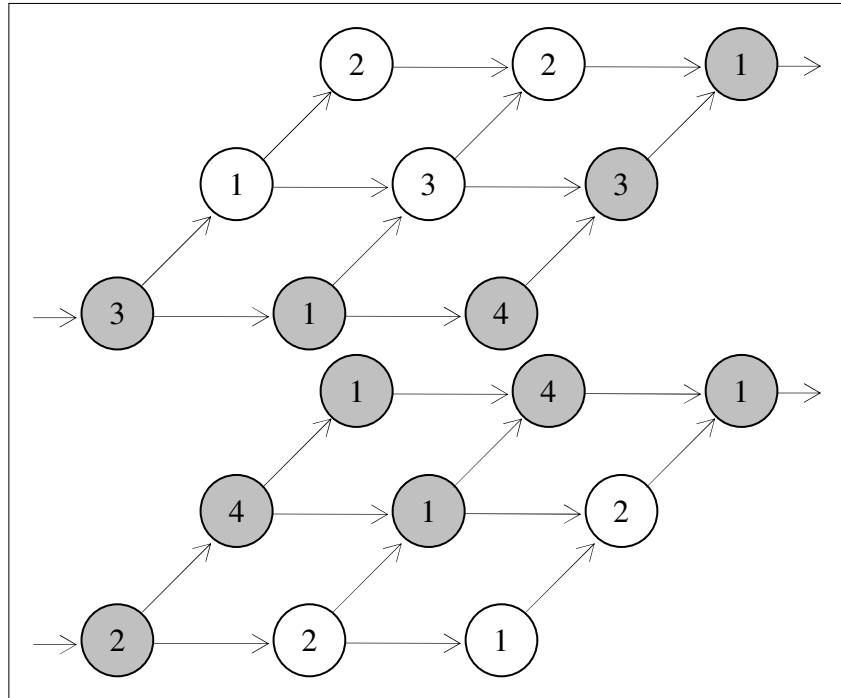


Рис. 5. c -слойный вершинно-взвешенный граф G_1^c

3. Асинхронный режим выполнения распределенных конкурирующих процессов в условиях ограниченного параллелизма. Рассмотрим случай *ограниченного* параллелизма, т.е. когда число блоков структурированного программного ресурса больше числа групп по c процессоров в каждой, т.е. $s > \left\lceil \frac{p}{c} \right\rceil$, $s = k \left\lceil \frac{p}{c} \right\rceil + r$, $k \geq 1$, $1 \leq r < \left\lceil \frac{p}{c} \right\rceil$.

Как и в случае, когда в общей памяти МС имеется одна копия программного ресурса, множество из s блоков разобьем на $(k+1)$ -ую группу по $\left\lceil \frac{p}{c} \right\rceil$ блока в каждой, за исключением $(k+1)$ -й группы, которая будет содержать r блоков [2]. Тогда матрицу $T^\varepsilon = [t_{ij}^\varepsilon]$, $i = \overline{1, n}$, $j = \overline{1, \left(k \left\lceil \frac{p}{c} \right\rceil + r \right)}$, времен выполнения блоков разбиваем на $(k+1)$ -ую подматрицу T_l^ε , $l = \overline{1, k+1}$, размерностью $n \times \left\lceil \frac{p}{c} \right\rceil$ каждая, за исключением последней T_{k+1}^ε , которая будет содержать при s не кратном $\left\lceil \frac{p}{c} \right\rceil$ только r столбцов, а остальные $\left\lceil \frac{p}{c} \right\rceil - r$ столбцов будут нулевыми. По каждой из подматриц T_l^ε , $l = \overline{1, k+1}$, строим $(k+1)$ -ую линейную диаграмму

Ганта, каждая из которых отображает во времени выполнение очередных $\left[\frac{p}{c}\right]$ блоков структурированного программного ресурса на $c\left[\frac{p}{c}\right]$ процессорах всеми n процессами, использующими ограниченное число c копий структурированного программного ресурса. При $r \neq 0$, $(k+1)$ -я диаграмма будет отражать выполнение последних r блоков на cr процессорах. На рис. 6 представлена диаграмма Ганта для МС с параметрами: $p=7$, $n=4$,

$$s=3, c=2, T^\varepsilon = \begin{bmatrix} 3 & 1 & 4 & 2 & 1 & 4 & 2 & 1 \\ 2 & 2 & 1 & 1 & 3 & 3 & 2 & 2 \\ 1 & 3 & 3 & 1 & 1 & 3 & 3 & 1 \\ 4 & 1 & 2 & 3 & 1 & 1 & 2 & 5 \end{bmatrix}.$$

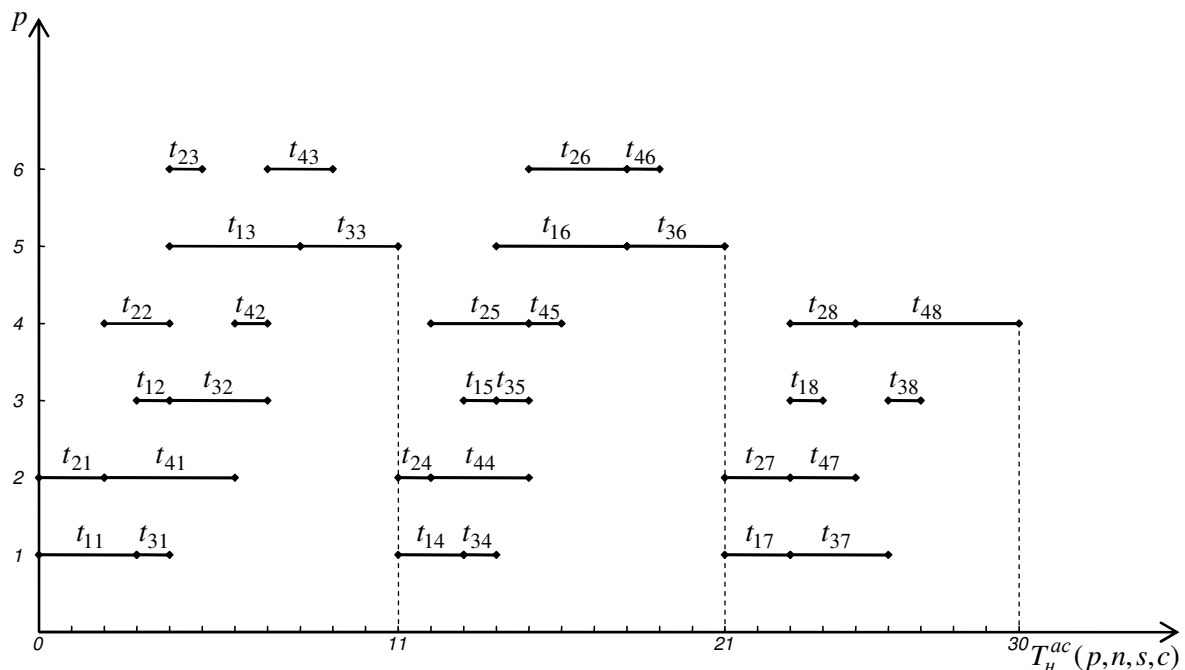


Рис. 6. Несовмещенная диаграмма Ганта

Суммарное время выполнения всех процессов, использующих c копий ПР, в этом случае будет определяться как сумма длин критических путей в каждой из подряд идущих несомещенных диаграммах Ганта. Однако, это время можно сократить, если последовательно поблочко совмещать диаграммы Ганта, начиная со второй диаграммы, по оси времени справа налево на максимально возможную величину не нарушая технологических условий асинхронного режима. В результате совмещения получим результирующую совмещенную диаграмму Ганта (рис. 7).

В [1] показано, что полученная структура результирующей совмещенной диаграммы Ганта определяется матрицей T^* (2), которая состоит из подматриц $T_1^\varepsilon, T_2^\varepsilon, \dots, T_{k+1}^\varepsilon$. В матрице T^* учтены как все горизонтальные, так и все вертикальные связи между блоками, а также связи между блоками из разных диаграмм Ганта. Отметим также, что результирующая матрица T^* будет иметь размерность $(k+1)n \times (k+1)\left[\frac{p}{c}\right]$, будет являться блочной, симметричной, верхней диагональной относительно второй диагонали, типа Ганкелевой порядка $k+1$.

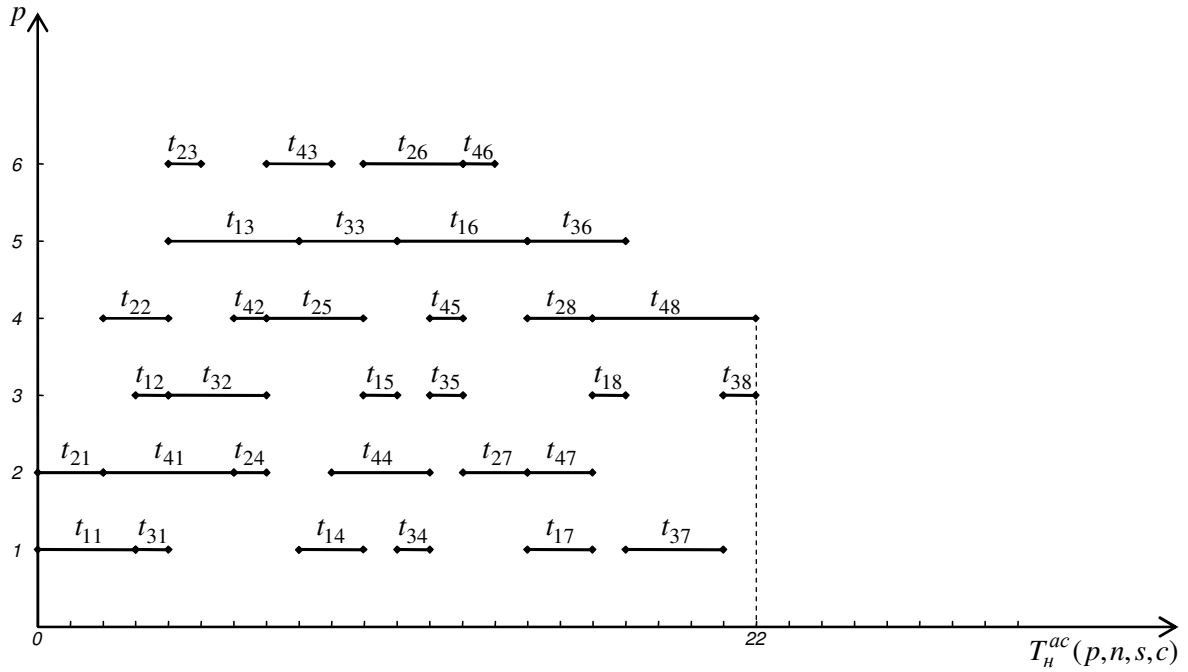


Рис. 7. Совмещенная диаграмма Ганта

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

где T_l^ε и T_{k+1}^ε матрицы вида:

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

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

По матрице T^* построим c -слойный вершинно-взвешенный граф G_2^c аналогичный графу G_1^c . Вершинам каждого слоя q размерности $(k+1)m \times (k+1) \left\lceil \frac{p}{c} \right\rceil$ графа G_2^c будут приписаны веса $t_{q+(i-1)c,j}^\varepsilon$, $q = \overline{1,c}$, $i = \overline{1,(k+1)m}$, $j = \overline{1,s}$. Вершины t_{q1}^ε будут являться входными, а $t_{q+((k+1)m-1)c,s}^\varepsilon$ – выходными, $q = \overline{1,c}$. Имеет место следующая теорема.

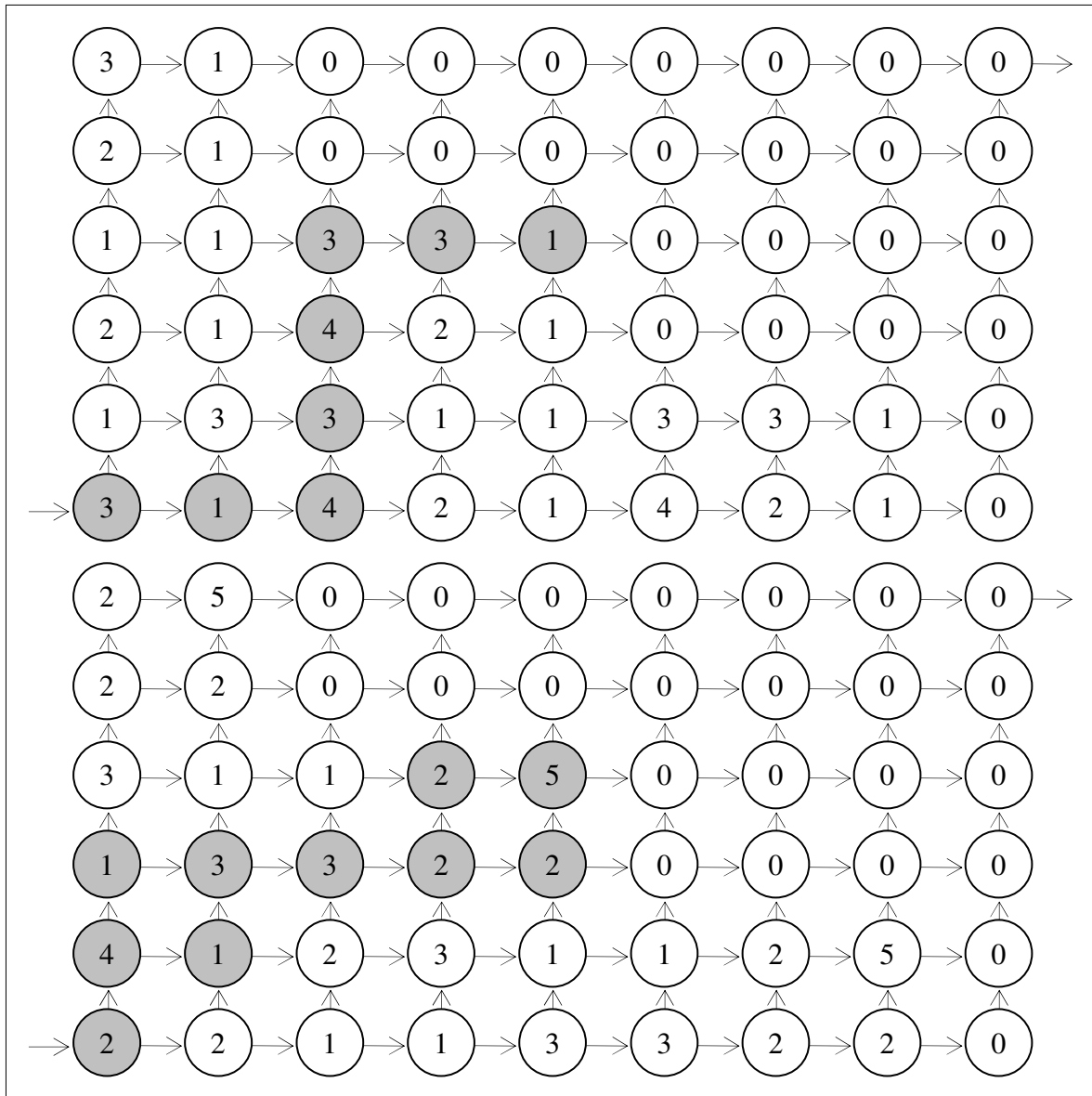
Теорема 2. Минимальное общее время $T_n^{ac}(p,n,s,c)$ выполнения n , $n \geq 2$, неоднородных распределенных конкурирующих процессов, использующих линейно структурированный на s , $s \geq 2$, блоков программный ресурс, с временами выполнения блоков с учетом дополнительными системными расходами $\varepsilon > 0$, задаваемыми матрицей $T^\varepsilon = [t_{ij}^\varepsilon]$, $i = \overline{1,n}$, $j = \overline{1,s}$, в многопроцессорной системе с p , $p \geq 2$, процессорами и $c \geq 2$ копиями структурированного программного ресурса $\left(\left\lceil \frac{p}{c} \right\rceil \geq 2 \right)$, в асинхронном режиме в случае неограниченного параллелизма $\left(s > \left\lceil \frac{p}{c} \right\rceil \right)$, определяется длиной критического пути из начальной вершины t_{q1}^ε в конечную вершину $t_{q+((k+1)m-1)c,s}^\varepsilon$, $q = \overline{1,c}$, графа G_2^c .

Пример 3. Используя значения параметров диаграммы Ганта, изображенной на рис. 6, найти минимальное общее время $T_n^{ac}(p,n,s,c)$, используя алгоритм нахождения критического пути в c -слойном вершинно-взвешенном графе G_2^c .

Так как $\left\lceil \frac{p}{c} \right\rceil = 3$, то $8 = s = k \left\lceil \frac{p}{c} \right\rceil + r = 2 \cdot 3 + 2$, следовательно, $k = 2$, $r = 2$. Матрицу T^ε разбиваем на подматрицы T_l^ε , $l = \overline{1,3}$, размерностью 4×3 каждая. Матрица T^* будет размерности $(k+1)n \times (k+1) \left\lceil \frac{p}{c} \right\rceil = 12 \times 9$ и иметь вид:

$$T^* = \begin{bmatrix} \left[\begin{array}{ccc} 3 & 1 & 4 \\ 2 & 2 & 1 \\ 1 & 3 & 3 \\ 4 & 1 & 2 \end{array} \right] \left\| \left[\begin{array}{ccc} 2 & 1 & 4 \\ 1 & 3 & 3 \\ 1 & 1 & 3 \\ 3 & 1 & 1 \end{array} \right] \left\| \left[\begin{array}{ccc} 2 & 1 & 0 \\ 2 & 2 & 0 \\ 3 & 1 & 0 \\ 2 & 5 & 0 \end{array} \right] \\ \left[\begin{array}{ccc} 2 & 1 & 4 \\ 1 & 3 & 3 \\ 1 & 1 & 3 \\ 3 & 1 & 1 \end{array} \right] \left\| \left[\begin{array}{ccc} 2 & 1 & 0 \\ 2 & 2 & 0 \\ 3 & 1 & 0 \\ 2 & 5 & 0 \end{array} \right] \left\| \left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{array} \right] \\ \left[\begin{array}{ccc} 2 & 1 & 0 \\ 2 & 2 & 0 \\ 3 & 1 & 0 \\ 2 & 5 & 0 \end{array} \right] \left\| \left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{array} \right] \left\| \left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{array} \right] \end{bmatrix}.$$

Построим по матрице T^* 2-слойный вершинно-взвешенный граф G_2^c (рис. 8). Длина критического пути равна 22.

Рис. 8. Граф G_2^c

4. Время выполнения однородных и одинаково распределенных конкурирующих процессов. Согласно определению 2 систему распределенных конкурирующих процессов будем считать *однородной*, если времена выполнения каждого блока Q_j , $j = \overline{1, s}$, каждым из процессов равны, т. е. $t_{ij}^\varepsilon = t_j^\varepsilon$, $i = \overline{1, n}$, $j = \overline{1, s}$.

На рис. 9 представлена диаграмма Ганта, иллюстрирующая выполнение однородных распределенных конкурирующих процессов при ограниченном числе копий программного

ресурса в МС с параметрами: $p = 7$, $n = 4$, $s = 3$, $c = 2$, $T^\varepsilon = \begin{bmatrix} 3 & 1 & 4 \\ 3 & 1 & 4 \\ 3 & 1 & 4 \\ 3 & 1 & 4 \end{bmatrix}$.

Оценим общее время выполнения n однородных распределенных конкурирующих процессов в асинхронном режиме, использующих c копий структурированного программного ресурса. Пусть $(t_1^\varepsilon, t_2^\varepsilon, \dots, t_s^\varepsilon)$ – длительности выполнения каждого из блоков Q_j , $j = \overline{1, s}$, программного ресурса с учетом накладных расходов ε , $t_j^\varepsilon = t_j + \varepsilon$, $j = \overline{1, s}$.

Обозначим длительность выполнения всего программного ресурса каждым из процессов через $T_\varepsilon^s = \sum_{j=1}^s t_j^s$.

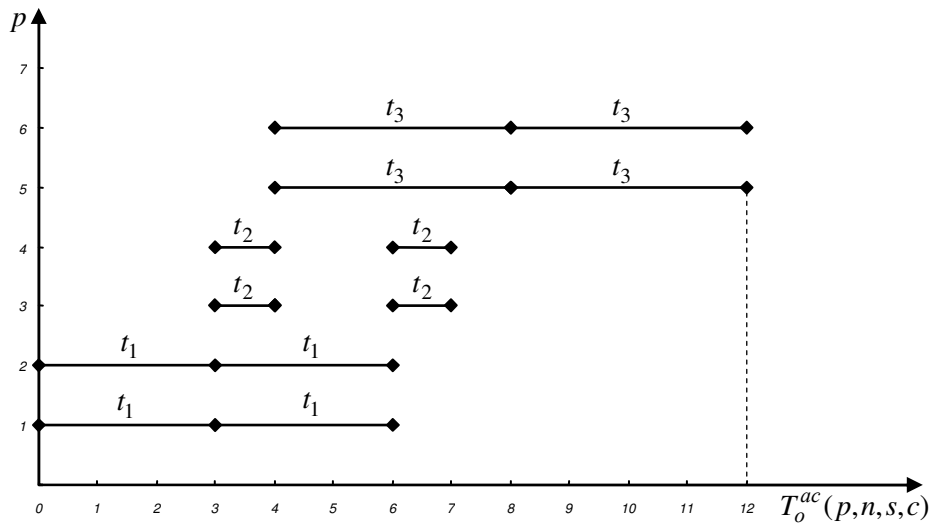


Рис.9. Асинхронный режим – однородные процессы

Покажем, что в этих условиях вычисление общего времени $T_o^{ac}(p, n, s, c)$ в случае неограниченного параллелизма сводится к нахождению общего времени выполнения однородных распределенных процессов при одной копии структурированного программного ресурса. При $n = mc$, $m \geq 2$, $2 \leq c \leq p$, выполнение c копий структурированного программного ресурса в асинхронном режиме равносильно выполнению c групп по m процессов, конкурирующих за использование одной копии программного ресурса на $\left\lceil \frac{p}{c} \right\rceil$ процессорах.

В силу формулы вычисления общего времени выполнения n однородных конкурирующих процессов, использующих одну копию структурированного программного ресурса, полученной в [1,3], и с учетом того, что $n = mc$, $m \geq 2$, $2 \leq c \leq p$, получаем:

$$T_o^{ac}(p, mc, s, c) = T_o^{ac}\left(\left\lceil \frac{p}{c} \right\rceil, m, s, 1\right) = T_\varepsilon^s + (m-1) \max_{1 \leq j \leq s} t_j^\varepsilon. \quad (2)$$

Для доказательства формулы (2) воспользуемся функционалом (1) задачи Беллмана–Джонсона, который для систем однородных конкурирующих процессов примет вид:

$$T_o^{ac}(p, n, s, c) = \max_{1 \leq u_1 \leq u_2 \leq \dots \leq u_{s-1} \leq m} \left[\sum_{i=1}^{u_1} t_1^\varepsilon + \sum_{i=u_1}^{u_2} t_2^\varepsilon + \dots + \sum_{i=u_{s-1}}^m t_s^\varepsilon \right] = \sum_{j=1}^s t_j^\varepsilon + (m-1) \max_{1 \leq j \leq s} t_j^\varepsilon,$$

где $m = \frac{n}{c}$, $t_j^\varepsilon = t_j + \varepsilon$, $j = \overline{1, s}$, а u_1, u_2, \dots, u_{s-1} – целые числа.

В случае, когда $s > \left\lceil \frac{p}{c} \right\rceil$, $s = k \left\lceil \frac{p}{c} \right\rceil$, $k > 1$, матрица времен выполнения блоков программного ресурса строится аналогично, как и матрица T^* . Отличие состоит в том, что в каждой из подматриц T_l^ε , $l = \overline{1, k}$, матрицы T^* все строки совпадают. Тогда, по аналогии с

теоремой 2, общее время $T_n^{ac}(p, n, s, c)$ выполнения n распределенных однородных конкурирующих процессов при ограниченном числе копий программного ресурса определяется длиной критического пути из начальной вершины в конечную соответствующего сетевого графа.

Если $s > \left\lceil \frac{p}{c} \right\rceil$, $s = k \left\lceil \frac{p}{c} \right\rceil + r$, $k \geq 1$, $1 \leq r < \left\lceil \frac{p}{c} \right\rceil$, то последняя подматрица T_{k+1}^ε матрицы T^* будет содержать $\left\lceil \frac{p}{c} \right\rceil - r$ нулевых столбцов.

Определение 4. Однородное структурирование программного ресурса на s блоков с временами выполнения $(t_1^\varepsilon, t_2^\varepsilon, \dots, t_s^\varepsilon)$, $\sum_{j=1}^s t_j^\varepsilon = T_\varepsilon^s$, будем называть *равномерным*, если $t_1^\varepsilon = t_2^\varepsilon = \dots = t_s^\varepsilon = t^\varepsilon$.

Следствие. В случае равномерного структурирования для вычисления минимального общего времени выполнения распределенных конкурирующих процессов при ограниченном числе копий программного ресурса имеют место формулы:

$$T_p^{ac}(p, n, s, c) = \begin{cases} (m + s - 1)t^\varepsilon, & \left\lceil \frac{p}{c} \right\rceil \geq \min\{m, s\}, \\ \left(km + \left\lceil \frac{p}{c} \right\rceil - 1 \right) t^\varepsilon, & \left\lceil \frac{p}{c} \right\rceil < \min\{m, s\}, s = k \left\lceil \frac{p}{c} \right\rceil, k > 1, \\ ((k + 1)m + r - 1)t^\varepsilon, & \left\lceil \frac{p}{c} \right\rceil < \min\{m, s\}, s = k \left\lceil \frac{p}{c} \right\rceil + r, k \geq 1, 1 \leq r < \left\lceil \frac{p}{c} \right\rceil. \end{cases}$$

Рассмотрим систему *одинаково распределенных* конкурирующих процессов. Времена выполнения всех блоков рассматриваемой системы с учетом накладных расходов ε каждым из i -х процессов совпадают и равны t_i^ε , т. е. справедлива цепочка равенств $t_{i1}^\varepsilon = t_{i2}^\varepsilon = \dots = t_{is}^\varepsilon = t_i^\varepsilon$, для всех $i = \overline{1, n}$.

На рис. 10 представлена диаграмма Ганта, иллюстрирующая выполнение одинаково распределенных конкурирующих процессов в МС с параметрами: $p = 7$, $n = 4$, $s = 3$, $c = 2$,

$$T^\varepsilon = \begin{bmatrix} 3 & 3 & 3 \\ 4 & 4 & 4 \\ 2 & 2 & 2 \\ 1 & 1 & 1 \end{bmatrix}, \text{ в случае неограниченного } \left(s \leq \left\lceil \frac{p}{c} \right\rceil \right) \text{ параллелизма.}$$

Обозначим через $T_\varepsilon^q = \sum_{i=1}^m t_{q+(i-1)c}^\varepsilon$ – суммарное время выполнения каждого из блоков Q_j , $j = \overline{1, s}$, всеми m процессами из q -й группы, а $t_{\max}^q = \max_{1 \leq i \leq m} t_{q+(i-1)c}^\varepsilon$ – максимальное время выполнения блока из этой группы, $q = \overline{1, s}$. Справедлива следующая теорема.

Теорема 3. Минимальное общее время выполнения n , $n \geq 2$, одинаково распределенных конкурирующих процессов, использующих структурированный на s , $s \geq 2$, блоков программный ресурс в многопроцессорной системе с p , $p \geq 2$, процессорами в асинхронном режиме при ограниченном числе копий программного ресурса составляет величину $T_{op}^{ac}(p, n, s, c)$, равную

$$T_{op}^{ac}(p, n, s, c) = \max_{1 \leq q \leq c} \begin{cases} T_{\varepsilon}^q + (s-1)t_{\max}^q, & \text{при } s \leq \left\lfloor \frac{p}{c} \right\rfloor, \text{ или } s > \left\lfloor \frac{p}{c} \right\rfloor, \text{ но } T_{\varepsilon}^q \leq \left\lfloor \frac{p}{c} \right\rfloor t_{\max}^q, \\ kT_{\varepsilon}^q + \left(\left\lfloor \frac{p}{c} \right\rfloor - 1 \right) t_{\max}^q, & \text{при } s = k \left\lfloor \frac{p}{c} \right\rfloor, k > 1 \text{ и } T_{\varepsilon}^q > \left\lfloor \frac{p}{c} \right\rfloor t_{\max}^q, \\ (k+1)T_{\varepsilon}^q + (r-1)t_{\max}^q, & \text{при } s = k \left\lfloor \frac{p}{c} \right\rfloor + r, k \geq 1, 1 \leq r < \left\lfloor \frac{p}{c} \right\rfloor \text{ и } T_{\varepsilon}^q > \left\lfloor \frac{p}{c} \right\rfloor t_{\max}^q. \end{cases}$$

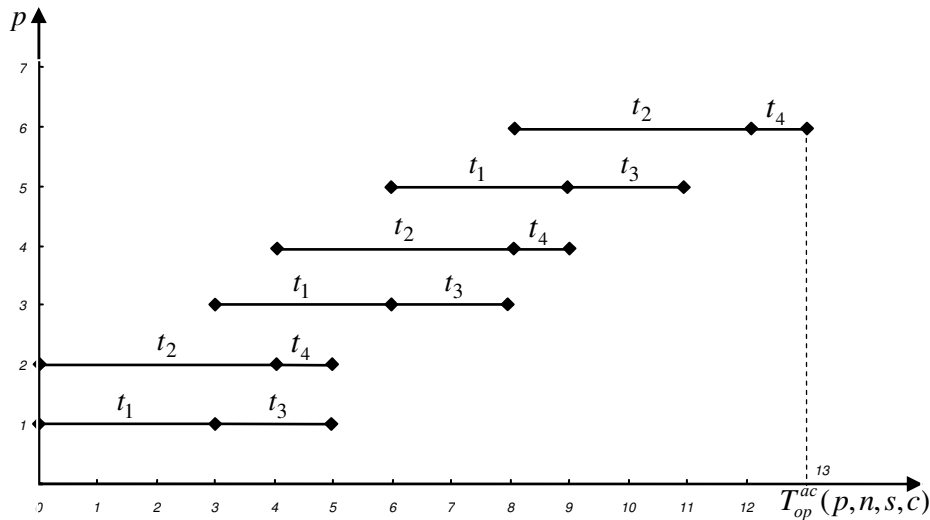


Рис.10. Асинхронный режим – одинаково распределенные процессы

Для доказательства рассмотрим сначала случай, когда $s \leq \left\lfloor \frac{p}{c} \right\rfloor$, или $s > \left\lfloor \frac{p}{c} \right\rfloor$, но $T_{\varepsilon}^q \leq \left\lfloor \frac{p}{c} \right\rfloor t_{\max}^q$. Воспользуемся функционалом (1) задачи Беллмана–Джонсона, который для систем одинаково распределенных конкурирующих процессов примет вид:

$$T_n^{ac}(p, n, s, c) = \max_{1 \leq u_1 \leq u_2 \leq \dots \leq u_{s-1} \leq m} \left[\sum_{i=1}^{u_1} t_{q+(i-1)c}^{\varepsilon} + \sum_{i=u_1}^{u_2} t_{q+(i-1)c}^{\varepsilon} + \dots + \sum_{i=u_{s-1}}^m t_{q+(i-1)c}^{\varepsilon} \right] = T_{\varepsilon}^q + (s-1) \max_{1 \leq i \leq m} t_{q+(i-1)c}^{\varepsilon},$$

где $m = \frac{n}{c}$, $t_{q+(i-1)c}^{\varepsilon} = t_{q+(i-1)c} + \varepsilon$, $q = \overline{1, c}$, $i = \overline{1, m}$, u_1, u_2, \dots, u_{s-1} – целые числа.

Рассмотрим далее случай, когда $s = k \left\lfloor \frac{p}{c} \right\rfloor$, $k > 1$ и $T_{\varepsilon}^q > \left\lfloor \frac{p}{c} \right\rfloor t_{\max}^q$. Вычисление общего времени $T_{op}^{ac}(p, n, s, c)$ в этом случае с помощью функционала задачи Беллмана–Джонсона приводит к формуле:

$$T_{op}^{ac}(p, n, s, c) = \max_{1 \leq u_1 \leq u_2 \leq \dots \leq u_{\left\lfloor \frac{p}{c} \right\rfloor - 1} \leq km} \left[\sum_{i=1}^{u_1} t_{q+(i-1)c}^{\varepsilon} + \sum_{i=u_1}^{u_2} t_{q+(i-1)c}^{\varepsilon} + \dots + \sum_{i=u_{\left\lfloor \frac{p}{c} \right\rfloor - 1}}^{km} t_{q+(i-1)c}^{\varepsilon} \right] = kT_{\varepsilon}^q + \left(\left\lfloor \frac{p}{c} \right\rfloor - 1 \right) t_{\max}^q.$$

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

В случае, когда $s = k \left\lceil \frac{p}{c} \right\rceil + r$, $k \geq 1$, $1 \leq r < \left\lceil \frac{p}{c} \right\rceil$ и $T_\varepsilon^q \geq \left\lceil \frac{p}{c} \right\rceil t_{\max}^q$, вычисление общего

времени с помощью функционала задачи Беллмана–Джонсона приводит к третьей формуле для вычисления общего времени выполнения одинаково распределенных конкурирующих процессов в случае ограниченного числа копий программного ресурса.

Заключение. В заключении отметим, что рассмотренное обобщение математической модели с одним программным ресурсом (конвейером) на случай ограниченного числа программных ресурсов позволяет установить взаимосвязи мультikonвейерной обработки с аналогичной обработкой при одном программном конвейере, получить аналитические оценки общего времени выполнения конкурирующих процессов при ограниченном параллелизме и провести математическое исследование эффективности и оптимальности мультikonвейерной организации конкурирующих процессов, вскрыть потенциальные возможности роста ускорения вычислений, выполнить сравнительный анализ различных режимов такой обработки.

Список литературы

1. Коваленко Н.С., Метельский В.М. О времени реализации конкурирующих процессов при распределенной обработке. 1996. №1. С. 54–64.
2. Иванников В. П., Коваленко Н. С., Метельский В. М. О минимальном времени реализации распределенных конкурирующих процессов в синхронных режимах // Программирование. 2000. №5. С. 44–52.
3. Павлов П.А. О времени реализации систем параллельных распределенных процессов // Вестник Полесского государственного университета. 2008. № 1. С. 60–67.
4. Коваленко Н.С., Павлов П.А. Эффективность и оптимальность структурирования программных ресурсов при распределенной обработке // Труды минского института управления. 2005. № 1. С. 104–107.
5. Павлов П.А. Анализ режимов организации одинаково распределенных конкурирующих процессов // Вестник БГУ. Сер. 1: Физ.Мат.Информ. 2006. № 1. С. 116–120.
6. Павлов П.А. Сравнительный анализ одинаково распределенных конкурирующих процессов с учетом дополнительных системных расходов // Вестник Фонда фундаментальных исследований. 2006. № 1. С. 55–58.
7. Капитонова Ю.В., Коваленко Н.С., Павлов П.А. Оптимальность систем одинаково распределенных конкурирующих процессов // Кибернетика и системный анализ. 2005. № 6. С. 3–10.
8. Коваленко Н.С., Павлов П.А. Эффективность систем конкурирующих процессов с учетом накладных расходов // Доклады НАН Беларуси. Сер. физ.–мат. наук. 2005. № 6. С. 32–36.
9. Коваленко Н.С., Павлов П.А. Системы одинаково распределенных конкурирующих процессов в условиях ограниченного параллелизма и их оптимальность // Доклады НАН Беларуси. Сер. физ.–мат. наук. 2006. № 2. С. 25–29.
10. Коваленко Н.С., Павлов П.А. Приемы ускорения вычислений при распределенной обработке // Управляющие системы и машины. 2006. № 4. С. 33–39.
11. Yu.V. Kapitonova, N.S. Kovalenko, P.A. Pavlov, Optimality of systems of identically distributed competing processes, Cybernetics and Systems Analysis. 2006. New York: Springer, p. 793–799.

