

**УЛУЧШЕННИЙ АЛГОРИТМ НАЗНАЧЕННЯ  
ДЛЯ ПЛАНИРОВЩИКОВ ЗАДАНИЙ В НЕОДНОРОДНЫХ  
РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ**

**А.М. СЕРГИЄНКО, В.П. СИМОНЕНКО, А.В. СИМОНЕНКО**

Рассмотрены основы проектирования пространственных планировщиков для глобальных, неоднородных, распределенных вычислительных систем. Представлены теоремы, позволяющие для двудольных графов, отображающих претендование заявок на ресурсы, уменьшить временную сложность венгерского алгоритма с  $O(n^3)$  до  $O(n^{1.5} \log n)$ . Подход применяется в алгоритме адаптивного мультианализа, который основан на предварительном анализе и коррекции графа паросочетаний. При его применении к матрицам графов с коэффициентом заполнения меньше 30% алгоритм имеет статистическую временную сложность, которая близка к линейной.

**ВВЕДЕНИЕ**

Распределенные вычислительные системы являются новым поколением вычислительных систем, которые используются в основном для научных вычислений. По мере развития таких систем возникают проблемы эффективного распределения задач в них [1, 2, 5]. Одной из проблем большой распределенной системы является вовлечение в вычислительный процесс максимально большего количества ресурсов и, соответственно, распределение большего количества задач по ресурсам. В большинстве систем [3, 6, 11] используются планировщики потокового типа, что определяют либо случайный выбор узла для задачи, либо выбор узла в соответствии с системой приоритетов. Можно назвать следующие знаковые проекты, получившие известность: PBS [7], LSF [4], NQE [7], I-SOFT [8], EASY [9], LoadLeveler [10]. При этом влияние назначения заявки на ресурс при последующих назначениях не учитывается.

Неэффективность такого подхода особенно сильно проявляется в глобальных вычислительных системах, так как в системах такого типа ресурсы и заявки являются неоднородными.

Идея рассматриваемого подхода заключается в разделении составления расписания на предварительный анализ исходной информации, определение стратегии поиска решения и поиск варианта решения с использованием результатов анализа. Этап предварительного анализа существенно уменьшает время решения по сравнению с классическими методами решения задачи планирования [12].

Требования заявок на захват ресурсов вычислительной системы можно разделить на обязательные  $C_x$ ,  $x=1, \dots, p$ , и оптимизирующие  $O_y$ ,  $y=1, \dots, k$ . С помощью обязательных требований  $x$ -го типа  $C_x^{i,j} \in \{0,1\}$  анализируется принципиальная возможность предоставления  $i$ -й заявке  $j$ -го ресурса [13]. Оптимизирующие требования  $y$ -го типа  $O_y^{i,j} \in [0,1]$  определяют степень предпочтения (приоритет)  $j$ -го ресурса для назначения на него  $i$ -й заявки по требованиям  $O_y^{i,j}$ . Для определения приоритета  $j$ -го ресурса для назначения на него  $i$ -й заявки по всем требованиям можно использовать выражение

$$O_{i,j} = \prod_{x=1}^p C_x^{i,j} \prod_{y=1}^p R_y O_y^{i,j}, \quad (1)$$

где  $R_y$  — весовой коэффициент оптимизирующего требования  $y$ -го типа.

Если система планирования учитывает обязательные и оптимизирующие требования, то коэффициент претендования в матрице связности вычисляется из выражения (1) и находится в диапазоне  $O_{i,j} \in [0,1]$ , а при диспетчеризации с учетом только обязательных требований  $O_{i,j} = \prod_{x=1}^p C_x^{i,j}$  и  $O_{i,j} \in \{0,1\}$ . В этом случае матрица связности, отображающая претендование заявок на ресурсы, примет булевый вид. Единичный элемент в этой матрице означает, что ресурс подходит для размещения заявки, т.е. есть достаточный объем памяти, процессор имеет необходимые производительность, программы, данные и т.д.

Для системы из  $N$  ресурсов в некоторый момент времени имеется  $M$  работ. Пул  $M$  задач, для которых система выделяет ресурсы, может быть ограничен наличием свободных ресурсов, т.е. при условии  $M = N$ . Требования работ на захват ресурсов представлены булевой матрицей связности (МС)  $C[i, j]$ ,  $i, j=1, \dots, N$ . Необходимо определить отношения работа – ресурс  $A = \{(V_i, W_j)\}$ ,  $V_i \in V = \{1, 2, \dots, N\}$ ,  $W_j \in W = \{1, 2, \dots, N\}$  так, чтобы  $\forall V_i, W_j (C[V_i, W_j] = 1, V_i \neq W_j)$ .

Введем дополнительные условия: ресурс может обслужить только одну заявку; процесс обслуживания заявки не может быть прерван; каждая работа имеет индивидуальные характеристики и может претендовать на захват только некоторых системных ресурсов; нет очереди к ресурсам; одна работа может быть обслужена только одним ресурсом. При этом отсутствие очередей определяется тем, что на данном уровне планирования планировщик более высокого уровня выбирает из общего потока заявок такое количество заявок, которое соответствует количеству свободных ресурсов.

Единичный элемент в МС  $C$  соответствует паре (ресурс( $j$ ), задача( $i$ )) или  $(R^j, J^j)$  соответствует выполнению всех  $K$  обязательных требований, предъявляемых к системе обработки заявки на этом ресурсе. Нулевой элемент МС означает невозможность обслуживания. При такой постановке решение задачи распределения заявок по ресурсам сводится к задаче поиска максимального паросочетания в невзвешенном двудольном графе. Для ее решения традиционно применяется алгоритм направленного поиска (АНП) на основе метода Карпа–Хопкрофта.

В работе рассмотрены основы АНП с алгоритмом адаптивного мультианализа (АМА), который обеспечивает ускоренный поиск максимального паросочетания.

### ЭЛЕМЕНТЫ ТЕОРИИ УСКОРЕНИЯ РАСПРЕДЕЛЕНИЯ РАБОТ ПО РЕСУРСАМ В НЕОДНОРОДНЫХ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

В основу наиболее известных алгоритмов [15–23] поиска максимального паросочетания в произвольном графе положены два основных подхода [16]: сведение задачи к поиску максимального потока в сети и поиска увеличивающего пути от свободных вершин. В основу поиска увеличивающего чередующегося пути положена схема Диница [16] и разработанный на его основе алгоритм Хопкрофта–Карпа [22]. Наилучшие известные алгоритмы, реализующие этот подход, имеют полиномиальную временную сложность. Однако эти алгоритмы и реализующие их программы имеют сложную структуру или предназначены для частных случаев и не обеспечивают приемлемых временных показателей, что ограничивает применение их в системах диспетчирования параллельных систем.

Для алгоритмов, использующих поиск увеличивающего чередующегося пути, правильность выбора начального варианта паросочетания в значительной степени влияет на количество шагов при поиске такого пути. Многие авторы [15–19] подчеркивают, что уникальные свойства двудольного графа позволяют уменьшить временную сложность алгоритмов.

Для выявления особенностей двудольного графа, влияющих на временную сложность, выполнено статистическое исследование программной модели базового алгоритма Хопкрофта–Карпа с временной сложностью  $O(n^{2.5})$ ,  $n = N$ . Результаты моделирования приведены на рис. 1 для графов размерности 10.

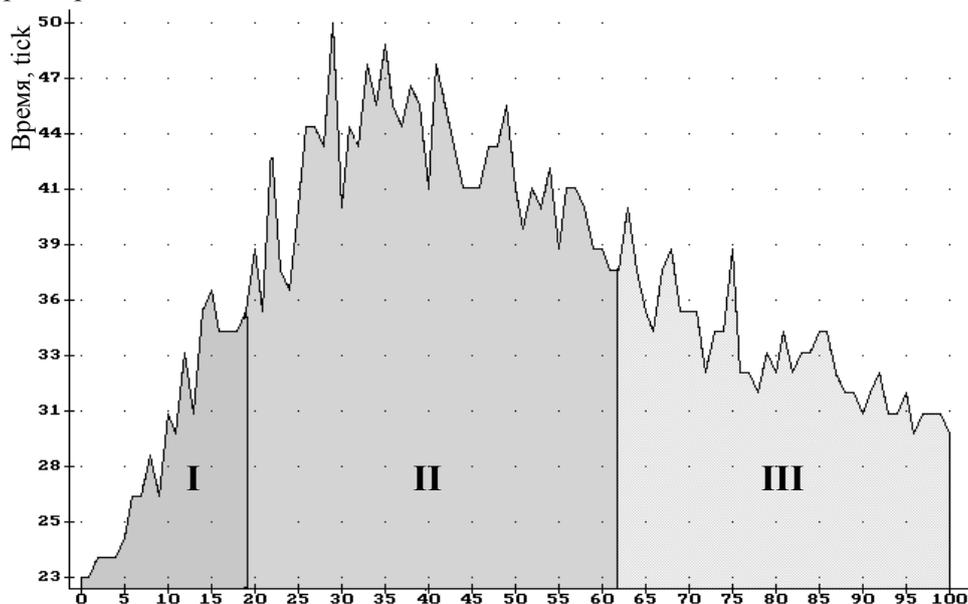


Рис. 1. Зависимость времени решения задачи поиска максимального паросочетания от коэффициента заполнения МС при  $N = 10$

При моделировании время вычисления максимального паросочетания вычислялось в относительных временных единицах (тиках). Как видно из этого графика, время решения задачи назначения зависит от коэффициента заполнения МС  $C$  и ее размерности  $N$ . При этом в графике можно выделить три зоны (I, II, III) с существенно различным временем решения. Исследования показали зависимость размера и положения этих зон от размерности задачи (рис. 2). Значительные различия временных затрат в выделенных зонах обусловило необходимость исследования в них свойств двудольного графа. Кроме этого, анализ базового алгоритма показал зависимость времени решения задачи от правильности выбора исходного (базового) варианта решения.

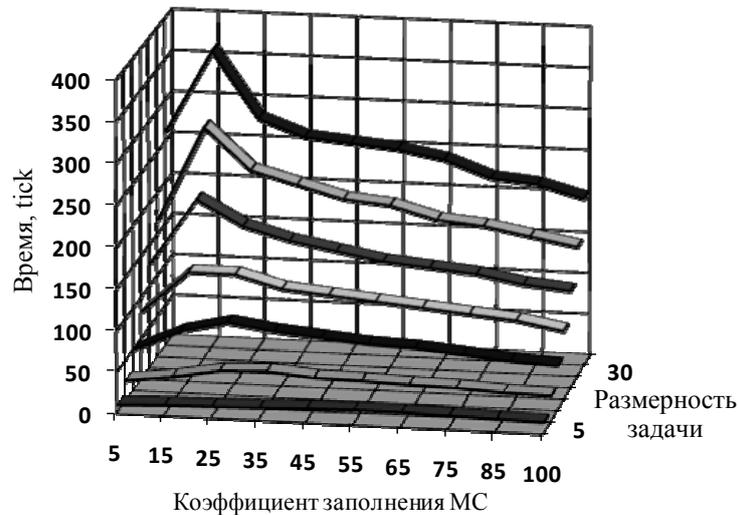


Рис. 2. Зависимость времени решения задачи планирования от ее размерности

Анализ алгоритмов поиска максимального паросочетания, а также анализ процесса поиска решения в выделенных зонах показывает, что наибольшие трудности, влияющие на время поиска максимального паросочетания, возникают в двудольных графах, в которых перманент МС близок к единице или равен нулю. Эти трудности вызваны тем, что поиск максимального паросочетания основан на центральной теореме Кенига–Холла о существовании паросочетания [15, 16] и теореме Берга [17]. Поэтому все известные алгоритмы предусматривают попытки поиска увеличивающегося пути от свободных вершин после генерирования базового варианта даже в том случае, если этого пути нет, что существенно увеличивает время поиска.

## ЭЛЕМЕНТЫ ТЕХНОЛОГИИ ПРЕДВАРИТЕЛЬНОГО АНАЛИЗА СТРУКТУРЫ ДВУДОЛЬНОГО ГРАФА

Чтобы упростить решение задачи поиска максимального паросочетания, предлагается разделить его на несколько этапов [13, 14], когда собственно решению предшествует быстрый анализ исходной информации и выработка рекомендаций для ее дальнейшего решения. Добавление дополнительных шагов значительно меньшей временной сложности, чем сам алгоритм, не влияет на временную сложность алгоритма в целом, однако позволяет

уменьшить размерность решения задачи за счет выделения назначений, которые обязательно нужно выполнить и выделить назначения, которые выполнять не нужно, и за счет этого избежать лишних проверок на возможность включения их в решение.

Кроме этого, на этапе подготовки исходной информации возможно вычисление мощности максимального паросочетания. Имея численное значение такой мощности, можно избежать поиска увеличивающего пути от свободных вершин при достижении расчетной мощности паросочетания на очередном шаге поиска решения.

Задача назначения требует определения условий возможности ее решения, т.е. возможности полного распределения всех заявок по ресурсам. Необходимые условия существования такого решения формулируются следующим образом:

$$\sum_{i=1}^N C[i, j] \neq 0, \quad i = 1, \dots, M, \quad \sum_{i=1}^M C[i, j] \neq 0, \quad i = 1, \dots, N. \quad (2)$$

Анализ появления конфликтных ситуаций и условий их возникновения показывает, что выполнение условий (2) необходимо для получения варианта размещения, но недостаточно, так как они не оценивают взаимосвязи возможных мест размещения заявок и влияния возможного назначения на последующие.

При выполнении исследований используются следующие определения.

**Определение 1.** Задан невзвешенный двудольный граф  $G = (V_R, V_J, E)$ , где  $V_R = \{R_1, R_2, \dots, R_N\}$ ,  $V_J = \{J_1, J_2, \dots, J_N\}$  — вершины графа, отвечающие ресурсам и задачам соответственно;  $E = \{E_1, E_2, \dots, E_d\}$  — дуги графа,  $E_k = (R^*, J^*)$ , где  $R^* \in V_R$  и  $J^* \in V_J$ ,  $k = 1, \dots, d$ ,  $0 \leq d \leq N^2$ . Матрица  $C_{RJ}$  называется *матрицей связности* графа  $G$ , если

$$C_{RJ}[i, j] = \begin{cases} 1 & \text{при } (R_i, J_j) \in E, \\ 0 & \text{при } (R_i, J_j) \notin E. \end{cases} \quad i = 1, \dots, N, \quad j = 1, \dots, N,$$

**Определение 2.** Подмножество  $A = \{a_1, a_2, \dots, a_n\}$ ,  $a_k = (R^k, J^k)$ ,  $R^k \in V_R$ ,  $J^k \in V_J$ ,  $k = 1, \dots, n$ ,  $n \in N$  называется *паросочетанием* графа  $G$  или его матрицы  $C_{RJ}$ , если

$$A = \{(R^1, J^1), \dots, (R^n, J^n)\}, \quad A_R = \{R^1, \dots, R^n\}, \quad A_J = \{J^1, \dots, J^n\}, \quad k = 1, \dots, n$$

и  $A_R \subseteq V_R$ ,  $A_J \subseteq V_J$ .

Другими словами, подмножество  $A$  ребер графа  $G = (V, E)$  называется паросочетанием, если никакие два ребра из  $A$  не имеют общей вершины.

**Определение 3.** Пара  $a_k = (R^k, J^k)$ ,  $k = 1, \dots, n$  называется *назначением* для ресурса  $R^k \in V_R$  и для задания  $J^k \in V_J$ .

**Определение 4.** Пусть  $X = \{A^1, \dots, A^z\}$ ,  $z \in N$  — множество результатов всех возможных паросочетаний для графа  $G = (V, E)$  или его матри-

цы  $C_{RJ}$ . Паросочетание  $A^*$  называется *максимальным паросочетанием* или *решением* для данного графа  $G$ , если

$$|A^*| = n^* \text{ и } n^* = \max\{|A^1|, \dots, |A^z|\}.$$

## ТЕОРЕТИЧЕСКОЕ ОБОСНОВАНИЕ ВЫДЕЛЕНИЯ ОБЯЗАТЕЛЬНЫХ НАЗНАЧЕНИЙ

Анализ свойств невзвешенного двудольного графа при решении задачи поиска максимального паросочетания, а также анализ известных алгоритмов позволяют сформулировать следующую теорему.

**Теорема 1.** Если в матрице  $C_{RJ}[i, j]$  графа  $G = (V_R, V_J, E)$  существует решение  $A$  мощностью  $n = N$  и существует такое назначение  $(p, q)$ , что  $C_{RJ}[p, q] = 1$ ,  $C_{RJ}[p, j] = 0$ ,  $j \in \{1, \dots, N\} \setminus q$  и/или  $C_{RJ}[i, q] = 0$ ,  $i \in \{1, \dots, N\} \setminus p$ , тогда это назначение  $(p, q)$  всегда участвует в решении  $A$ , т.е.  $(p, q) \in A$ .

Если для невзвешенного двудольного графа существует совершенное паросочетание и в графе есть вершина со степенью один, то ребро, инцидентное этой вершине, и вершины, инцидентные этому ребру обязательно входят в совершенное паросочетание.

**Определение 5.** Назначение  $(p, q)$  по теореме 1 называется *обязательным*.

**Следствие.** Если существует решение  $A^*$ , то задачу назначения можно разделить на две части: в первой части участвуют только обязательные назначения  $(p, q)$ , во второй части — оставшиеся назначения из новой матрицы связности  $C_{RJ}^*$ , которую получают после удаления строк и столбцов, соответствующих обязательным назначениям, определенным по теореме 1. Причем ввиду редукции графа и соответствующей коррекции МС это следствие может применяться рекуррентно.

**Теорема 1\*.** Любая из вершин в невзвешенном двудольном графе, имеющая степень, равную единице, всегда участвует в одном из вариантов максимального паросочетания.

Теорема 1\* справедлива как для вершин-заявок, так и для вершин-ресурсов. В том случае, если вершины со степенью 1 образуют веер, то теорема 1\* справедлива для любой вершины, входящей в веер и каждая из них может быть взята в паросочетание, а проверку остальных вершин на возможность получения увеличивающего пути выполнять не следует.

**Теорема 2.** Если в матрице  $C_{RJ}[i, j]$  существует веер  $E_{FA}$ :

$$E_{FA} = \{(R^1, q), \dots, (R^f, q)\}, C_{RJ}[R^k, q] = 1, k = 1, \dots, f$$

и  $C_{RJ}[R^k, J^k] = 0$ ,  $J^k \neq q$  или  $E_{FA} = \{(p, J^1), \dots, (p, J^f)\}$ ,  $C_{RJ}[p, J^k] = 1$ ,  $k = 1, \dots, f$  и  $C_{RJ}[R^k, J^k] = 0$ ,  $J^k \neq p$ , тогда любая из вершин из  $E_{FA}$  вхо-

дит в один из вариантов максимального паросочетания, совершенное паросочетание не может быть получено и мощность максимального паросочетания определяется из выражения  $M < N - f + 1$ .

**Следствие 1.** Мощность решения задачи поиска максимального паросочетания может быть уменьшена на количество пар вершин, определенных по теоремам 1 и 1\*, и поиск паросочетания должен вестись в новом суграфе.

**Следствие 1\*.** Размерность решения задачи поиска максимального паросочетания должна быть уменьшена на количество вершин, входящих в веер, а поиск паросочетания должен вестись в новом суграфе.

**Следствие 2.** Смежные ребра, инцидентные вершинам, определенным по теореме 1, должны быть удалены из дальнейшего рассмотрения, а исходный граф редуцирован и преобразован в новый суграф.

**Следствие 3.** Смежные ребра, инцидентные вершинам, определенным по теореме 2, должны быть удалены из дальнейшего рассмотрения, а исходный граф редуцирован и преобразован в новый суграф.

## ТЕОРЕТИЧЕСКИЕ ПРЕДПОСЫЛКИ ВЫДЕЛЕНИЯ КОНФЛИКТНЫХ НАЗНАЧЕНИЙ

Базовый или начальный вариант решения задачи назначения обычно выполняется случайным образом. Анализ свойств двудольного графа показывает, что наряду с обязательными назначениями в нем можно выделить конфликтные, попадание которых в решение неминуемо приводит к появлению свободных вершин, а значит и к дополнительным шагам при формировании окончательного решения. Для выявления конфликтных назначений предлагается выполнить анализ исходного графа, а именно его МС.

В результате выполнения этой операции находятся все основные подматрицы в МС  $C'_{RJ} = I - C_{RJ}$ , где  $I$  — квадратная матрица единичных элементов. Это соответствует выделению в исходной МС нулевых подматриц. Локализация таких подматриц реализуется целевой перестановкой строк и столбцов МС. При разработке алгоритма назначения учитывались следующие очевидные утверждения.

**Утверждение 1.** Так как каждая строка МС представляет собой отражение возможного распределения заявок на ресурсы, то перемена местами двух строк МС с запоминанием нового расположения заявок не влияет на качественную характеристику МС и исходного графа.

**Утверждение 2.** Так как каждый столбец МС представляет собой отражение претендования заявок на захват данного ресурса, то перемена местами двух столбцов МС с запоминанием нового расположения ресурсов не влияет на качественную характеристику МС и исходного графа.

Утверждения 1 и 2 не требуют доказательств, так как они следуют из определения изоморфности графов, полученных путем перестановки любых вершин. При эквивалентном преобразовании МС путем перестановки строк и столбцов делается попытка выполнить квазиназначения и получить один

из вариантов решения задачи назначения соответствующего элементам главной диагонали МС. Это не противоречит одному из определений максимального паросочетания [15]: максимальное паросочетание — это максимальное множество единиц матрицы связности, у которых не совпадают координаты.

Алгоритм изоморфного преобразования исходного графа основан на упорядочении МС. Строки упорядочиваются по возрастанию, а столбцы — по убыванию сумм единиц с попыткой получить единичную главную диагональ, что соответствует квазиназначению данной заявки на ресурс. Следует отметить, что для уменьшения временной сложности алгоритма сами перестановки не выполняются, а запоминается новый порядок строк и столбцов по результатам сортировки. В основу алгоритма положены модифицированный метод разложения ориентированного графа на максимально сильносвязанные подграфы и алгоритм поиска минимального подмножества сочленения, предложенные Мальгранжем [15].

**Теорема 3.** Если после эквивалентного преобразования удастся получить терм-ранг МС, равный рангу МС, то все заявки имеют места назначения, соответствующие единицам в главной диагонали, и найдено полное паросочетание, получен один из вариантов решения задачи назначения заявок на ресурсы.

**Доказательство.** Задача о паросочетании по определению, приведенном в работе [15], состоит в нахождении в исходном графе  $G$  максимального паросочетания. Если мощность паросочетания равна  $\lfloor V/2 \rfloor$ , т.е. наибольшему возможному значению в графе с  $V$  вершинами, то паросочетание называется полным. Так как терм-ранг матрицы равен  $\lfloor V/2 \rfloor$ , то утверждение верно.

**Теорема 4.** Если в матрице  $C_{RJ}[i, j]$  можно выделить подматрицу  $C_{MM}[k, l]$ ,  $k = 1, \dots, T$ ,  $l = (N - S + 1), \dots, N$ , где  $S + T > N$  и  $C_{MM}$  — нулевая матрица, тогда задача назначения не имеет полного решения (рис. 3, а–в).

**Доказательство.** Предположим, что существует полное решение  $A = \{(R^1, J^1), \dots, (R^n, J^n)\}$ ,  $n = N$ . Так как решение полное, имеем

$$A_R = \{R^1, \dots, R^n\} = V_R = \{R_1, \dots, R_N\} = \{1, 2, \dots, N\};$$

$$A_J = \{J^1, \dots, J^n\} = V_J = \{J_1, \dots, J_N\} = \{1, 2, \dots, N\}.$$

Рассмотрим вершины  $V_{RT} = \{R_1, \dots, R_T\}$ , которые принадлежат подматрице  $C_{MM}$ . Поскольку они также входят в состав полного решения  $A$ , всегда имеем  $T$  соответствующих пар  $\{(R_1, J_1^*), \dots, (R_T, J_T^*)\}$  в  $A$  и  $C_{RJ}[R_k, J_k^*] = 1$ , для  $k = 1, \dots, T$ ,  $\{J_1^*, \dots, J_T^*\} = V_{JT} \subset V_J$ . Очевидно, что для таких  $T$  ресурсов  $R_k$  имеются по крайней мере  $T$  заданий  $J_k^*$ , которые имеют доступ к ним.

Поскольку  $C_{MM}$  — нулевая матрица, имеем  $C_{MM}[k, l] = C_{RJ}[k, l] = 0$ . Это означает, что есть  $S$  вершин-заданий  $J_k^*$ , которые не имеют доступа к  $T$  ресур-

сам  $R_k$ . Поэтому осталось максимально  $N - S$  заданий, которые возможно имеют доступ к  $T$  ресурсам  $R_k$ . Однако, исходя из условия, что  $S + T > N$ , получаем  $T > N - S$ , что является противоречием. Поэтому предложение о существовании полного решения  $A$  неверно.

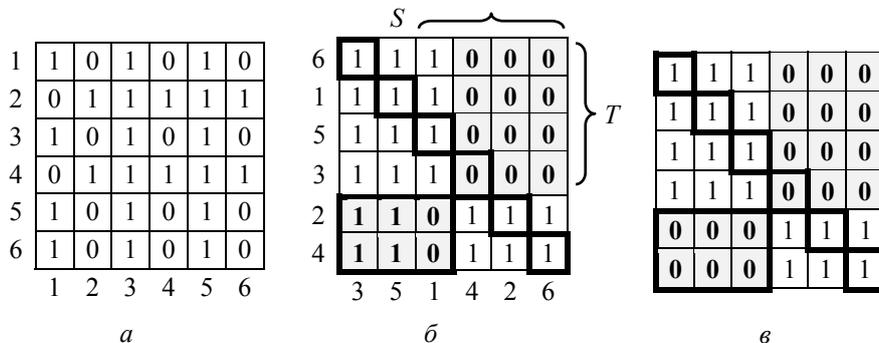


Рис. 3. К теореме 4:  $a$  — исходная МС;  $b$  — МС после преобразования;  $v$  — МС после коррекции:  $\blacksquare$  — конфликтная зона

Теорема 4 справедлива не только для матрицы  $C_{MM}$ , которая находится над главной диагональю матрицы  $C_{RJ}$ , но также для любой позиции  $C_{MM}$  в  $C_{RJ}$ . Из утверждений 1 и 2 очевидно, что всегда существует такое эквивалентное преобразование матрицы, когда матрица  $C_{MM}$  может быть перенесена из любой позиции в правый верхний угол матрицы  $C_{RJ}$ .

**Примечание.** Доказательство теоремы 4 можно выполнить на основе теоремы Фробениуса–Кенига и Минка о вычислении перманента 0,1-матрицы.

**Теорема 5.** Если в матрице  $C_{RJ}$  можно выделить подматрицу  $C_{MN}[k, l]$ ,  $k = 1, \dots, T$ ,  $l = (N - S + 1), \dots, N$ , где  $S + T = N$  и  $C_{MN}$  — нулевая матрица, тогда

$$R_i \in \{1, \dots, N - S\}, J_j \in \{T + 1, \dots, N\}, (R_i, J_j) \notin A$$

и все элементы  $(R_i, J_j)$  должны быть обнулены и исключены из рассмотрения при поиске полного решения (рис. 4).

**Доказательство.** Предположим, что в  $C_{RJ}$  имеется подматрица  $C_{MT}$ , в которой есть элемент  $C_{MT}[i^*, j^*] = 1$  и  $(i^*, j^*) \in A$ ,  $i^* \in \{T + 1, \dots, N\}$ ,  $j^* \in \{1, \dots, N - S\}$ . Выделение подматрицы  $C_{MT}$  делит МС на подматрицы:  $S \times T$ ,  $(N - S) \times T$ ,  $(N - T) \times S$ . Так как по условию  $S + T = N$ , то подматрицы  $(N - S) \times T$  и  $(N - T) \times S$  — квадратные. Тогда предположение, что  $(i^*, j^*) \in A$ , где  $i^* \in \{T + 1, \dots, N\}$ ,  $j^* \in \{1, \dots, N - S\}$ , приводит к тому, что в подматрице  $(N - S) \times N$  должны присутствовать  $T + 1$  назначений, входящих в  $A$ . Но так как  $T + 1 > N - S$  и  $S + T = N$ , то это предположение неверно. Аналогичное доказательство можно привести и для подматрицы  $(N - T) \times S$ .

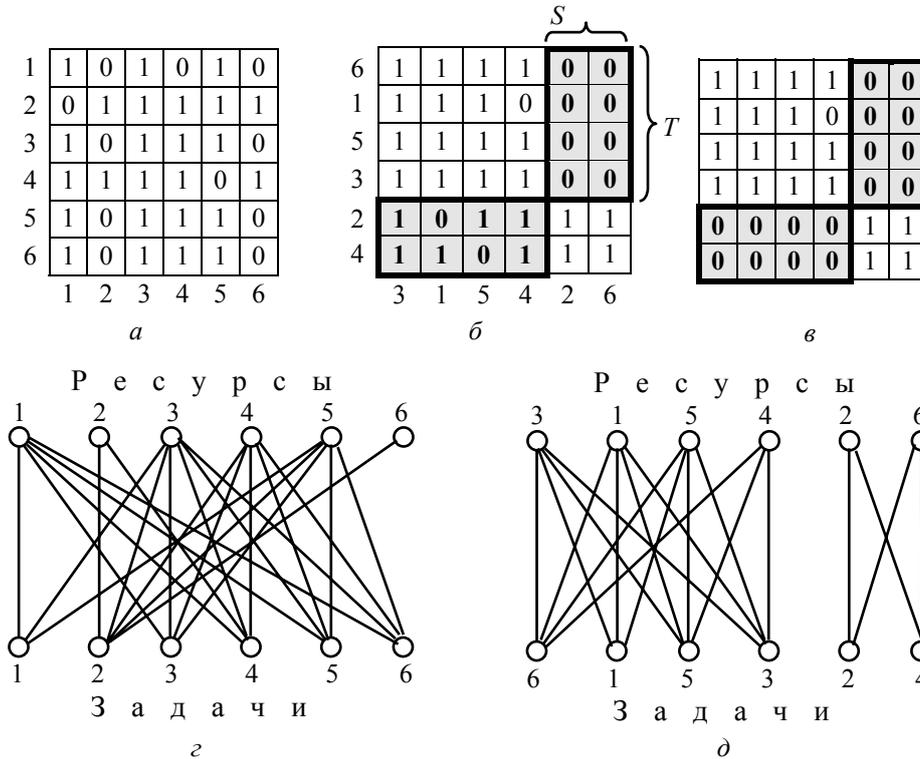


Рис. 4. К теореме 5: исходная МС: а — МС после преобразования; б — МС после коррекции; в — исходный граф; г — граф после редукции

**Теорема 6.** Если в матрице  $C_{MT}[i, j]$  можно выделить несколько подматриц, удовлетворяющих теореме 5, то все соответствующие им подматрицы, симметричные относительно главной диагонали, являются конфликтными и должны быть обнулены (рис. 5, а–д).

Доказательство теоремы аналогично доказательству теоремы 5 и справедливо для каждой подматрицы отдельно.

**Теорема 7.** Если в матрице  $C_{RJ}$  существует подматрица  $C_{MT}$  такая, что  $C_{MT}[k, l] = 0$ ,  $k = 1, \dots, T$ ,  $l = (N - S + 1), \dots, N$ , кроме элемента  $C_{MT}[T, N - S + 1] = 1$ , причем  $S + T = N + 1$ , тогда назначение  $(T, N - S + 1)$  является ключевым и всегда входит в состав одного из вариантов решения.

**Доказательство.** Из условия теоремы в подматрице  $C_{MT}$  можно выделить две подматрицы  $C_{MN1}$  и  $C_{MN2}$  такие, что

$$C_{MN1}[k, l] = 0, \quad k = 1, \dots, T - 1, \quad l = N - S + 1, \dots, N;$$

$$C_{MN2}[k, l] = 0, \quad k = 1, \dots, T - 1, \quad l = N - S + 2, \dots, N. \quad (3)$$

Применив теорему 6 для этих подматриц, получаем две подматрицы  $C_{MN1}^*$  и  $C_{MN2}^*$ , симметричные им относительно главной диагонали, которые являются конфликтными и должны быть обнулены. Тогда

$$C_{MN1}^*[k,l]=0, k=T,\dots,N, l=1,\dots,N-S;$$

$$C_{MN2}^*[k,l]=0, k=T+1,\dots,N, l=1,\dots,N-S+1. \tag{4}$$

Рассмотрим строку  $T$  и столбец  $N-S+1$ . Из условия теоремы, а также (3) и (4) имеем:  $C_{RJ}[T,N-S+1]=1, C_{RJ}[T,k]=0, k=1,\dots,N$  и  $k \neq T, C_{RJ}[k,N-S+1]=0, k=1,\dots,N$  и  $k \neq 1$ . Но по теореме 1 и следствию 1 теоремы 2 назначение  $(T,N-S+1)$  является обязательным и всегда участвует в составе одного из вариантов решения максимального паросочетания.

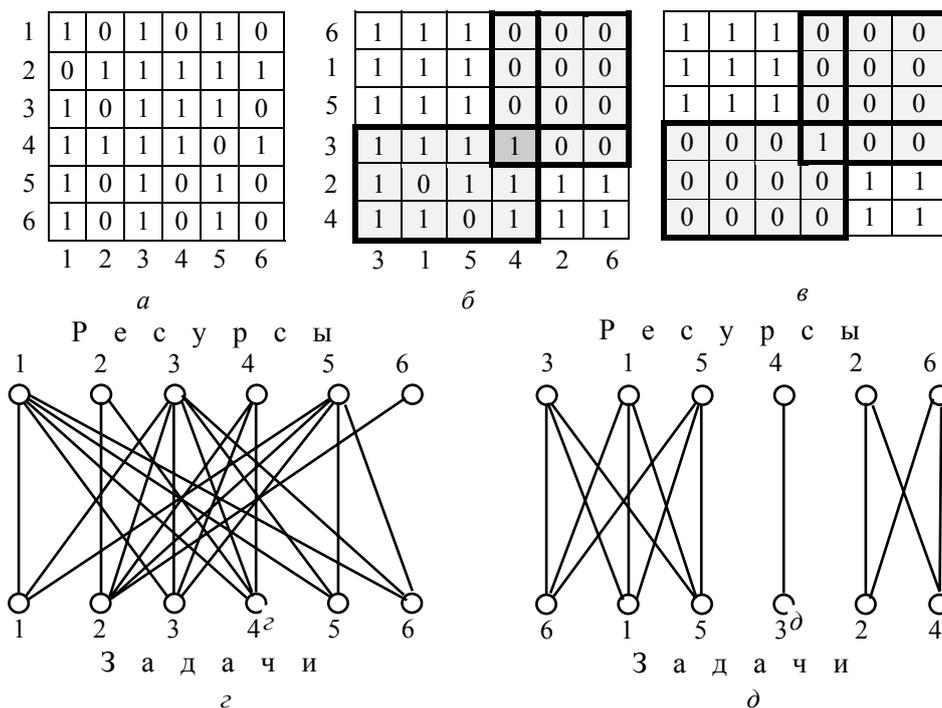


Рис. 5. К теореме 6:  $a$  — исходная МС;  $b$  — МС после преобразования;  $c$  — МС после коррекции;  $z$  — исходный граф;  $d$  — граф после преобразования и коррекции

**Следствие.** Если МС представляет собой нижнюю диагональную матрицу, то задача имеет единственное решение, соответствующее единичной главной диагонали (рис. 6,  $a-d$ ). Доказательство следует из теорем 5, 6.

На основе теорем 1–7 и их следствий разработан алгоритм адаптивного мультианализа (АМА), который принадлежит к категории алгоритмов пошагового конструирования решения задачи назначения [13, 14].

### ОЦЕНКА ВРЕМЕННОЙ СЛОЖНОСТИ МОДИФИЦИРОВАННОГО ВЕНГЕРСКОГО АЛГОРИТМА

Основой алгоритма направленного поиска решения задачи о назначениях является венгерский метод, имеющий временную сложность  $O(n^3)$ , где  $n$  — количество вершин графа матрицы  $C_{RJ}$  заданий-ресурсов. Тради-

ционный АНП основан на методе Карпа–Хопкрофта. Его временная сложность известна из работы [22] и оценивается как  $O(n^{0.5}m)$ , где  $m$  — количество дуг графа. Но с учетом того, что  $n \leq m \leq n^2$ , алгоритм Карпа–Хопкрофта имеет временную сложность  $O(n^{2.5})$ . Однако для процедуры поиска максимального паросочетания используется разработанный алгоритм поиска максимального паросочетания вместо традиционного, который был использован в венгерском методе.

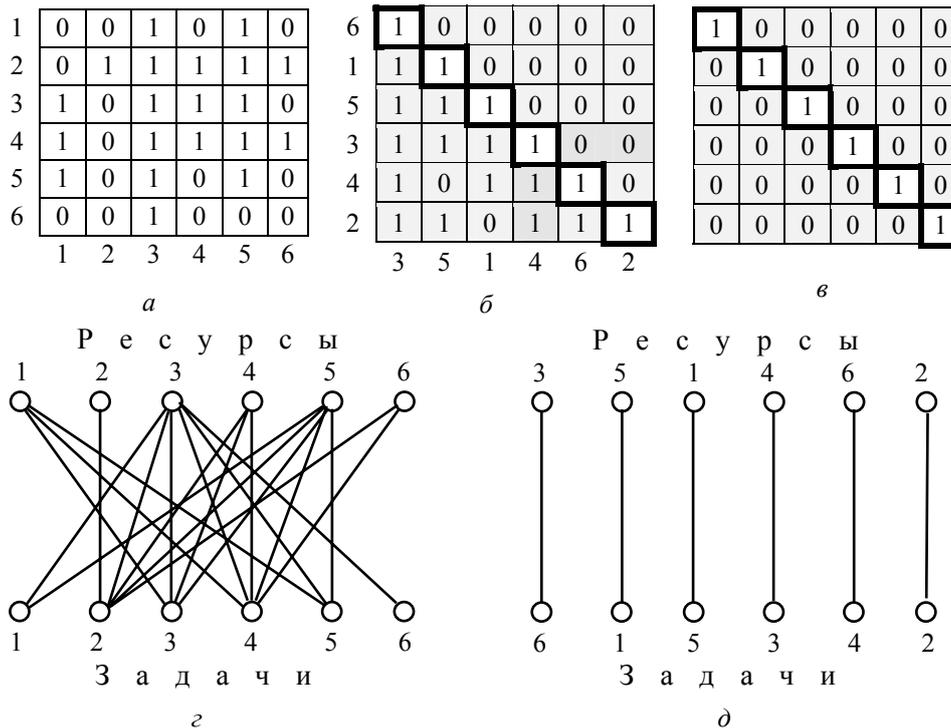


Рис. 6. К теореме 7: а — исходная МС; б — МС после преобразования; в — МС после коррекции; г — исходный граф; д — граф после коррекции

Для оценки временной сложности АНП, т. е., венгерского метода с использованием АМА вместо алгоритма Карпа–Хопкрофта, выполним оценку временной сложности венгерского метода без алгоритма Карпа–Хопкрофта и оценку временной сложности АМА отдельно.

**Временная сложность венгерского метода без алгоритма Карпа–Хопкрофта.** Алгоритм АНП состоит из 4 главных процедур. Проведем оценку временной сложности для каждой из них:

1. Процедура подготовки данных и поля определения поиска имеет временную сложность  $O(E)$  или в худшем случае  $O(n^2)$ .
2. Процедура определения внешней границы зоны поиска имеет временную сложность  $O(2n)$ .
3. Процедура ориентированного поиска оптимального расписания в выделенной линии поиска имеет временную сложность  $O(X)$ , которую определим позже.

4. Процедура определения новой линии поиска имеет временную сложность  $O(n + 2n^2)$  для маркировки и  $O(3n^2)$  для перестановки. В итоге ее общая временная сложность равна  $O(Y)$ , где  $Y = n + 5n^2$ . Для уменьшения временной сложности этой процедуры в АНП формируются дополнительные векторы степеней вершин и линии поиска. В этом случае  $Y = n + 2n + 5n^2 = 7n$ .

При сравнительной оценке различных алгоритмов процедура формирования исходных данных исключается из рассмотрения, так как ее временная сложность одинакова. При поиске решения имеется цикл из процедур 4 и 3. Допустим что этот цикл выполняется  $z$  раз и имеет временную сложность  $Z$ . Так как венгерский метод имеет временную сложность  $O(n^3)$ , то временная сложность процедур внутреннего цикла будет равна  $O(n^3)/W$ . Поскольку для процедуры 3 используется алгоритм Карпа–Хопкрофта, то  $X = O(n^{0,5}m)$  или  $X = O(n^{0,5})$ ,  $Y = O(n + 5n^2)$ . С другой стороны, временную сложность  $W$  можно принять как  $W = O(\max(X, Y)) = O(n^{0,5}m)$  или  $O(n^{2,5})$ . Отсюда  $O(n^3)/Z = \max(X, Y)$ . Таким образом,  $Z = O(n^3)/O(n^{2,5}) = O(n^{0,5})$ .

Временная сложность АНП без процедуры 3, т.е. без алгоритма Карпа–Хопкрофта, оценивается как  $O(n^2 + 2n + n^{0,5}(n + 5n^2))$ , а в общем случае —  $O(n^2 + n^{0,5} \max(n^{0,5}m, n))$ . Без учета операций подготовки исходной информации временная сложность АНП равна  $O(n^{0,5} \max(n^{0,5}m, n))$ .

**Временная сложность АНП с АМА.** Временная сложность алгоритма АНП, реализованного на основе предложенного метода пошагового конструирования, состоит из суммы оценок сложности шагов 1–4. Сложность процедуры подготовки исходной информации и анализа (шаг 1) зависит от количества ребер в исходном графе, а так как при этом выполняются обычные операции по формированию матрицы связности или списков инцидентности, то сложность выполнения этого шага равна  $o(m)$ .

Ввиду того, что в предлагаемом алгоритме, кроме матрицы связности, формируются еще и векторы степеней вершин графа, то временная сложность этой процедуры увеличивается на  $O(2n)$ . Тогда общая временная сложность выполнения этой процедуры равна  $O(m + 2n)$ . При выполнении грубого анализа ищутся изолированные вершины и для этого необходим одноразовый просмотр векторов, что определяет временную сложность выполнения этого шага как  $O(n)$ .

Выполнение шага 2 для поиска обязательных назначений имеет временную сложность  $O(n)$  и в случае каждого обнаружения вершины со степенью 1 граф подвергается редукции с временной сложностью  $O(2n)$ . В случае единственного решения после выполнения  $n$  шагов получается полное решение. Тогда временная сложность нахождения полного решения равна  $O(n(n - 1)/2 + m)$ . Общая временная сложность этой процедуры равняется  $O(n + 2n) = O(3n)$ .

Временная сложность выполнения шага 3 определяется выбранным алгоритмом сортировки одномерного массива и сложностью анализа преобразованной МС и ее коррекции и равна  $O(2n \log n + m/2)$ .

Общая временная сложность АНП  $X_{\text{АНП}} = O(3m + 2n(3 + \log n))$ . Ввиду того, что на общую оценку сложности алгоритма влияет выполнение шагов 1 и 4, а значение  $m$  в худшем случае равно  $n^2$ , то сложностью остальных шагов можно пренебречь. Тогда временная сложность алгоритма АНП равна  $O(m + n(3 + \log n))$ . Однако анализ алгоритмов, выполняющих поиск максимального паросочетания на основе теоремы Бержа, и примеров [18, 19, 20], с помощью которых обычно иллюстрируется работоспособность предлагаемых алгоритмов, позволяет сделать вывод, что для большинства вариантов не требуется выполнение шага 4 и, следовательно, сложность описанного алгоритма получения решения можно уменьшить для этого вида графов до  $O(m)$  при использовании списков инцидентности, а при использовании МС — до  $O(n)^2$ .

Таким образом, временная сложность венгерского метода в общем случае  $O(n^{0.5} \max(X, n))$  или  $O(n^{2.5} \max(X, n)/m)$ , где  $X = X_{\text{АНП}} = O(m + n \log n)$ . Тогда временная сложность АНП с АМА равна  $O(n^{2.5} \log n)$  при  $m = n$ , или  $O(n^{1.5} \log n)$  при  $m = n^2$ .

Следует отметить, что временная сложность АНП с АМА определялась для худшего случая, т.е. когда МС является заполненной. Но и в этом случае временная сложность АНП равна  $O(n^{1.5} \log n)$ , что меньше  $O(n^3)$ .

Достоверность расчета временной сложности проверялась на программной модели. Результаты статистического исследования представлены на рис. 7. Ввиду того, что АНП является адаптивным алгоритмом, а поиск максимального паросочетания выполнялся в разреженной матрице с коэффициентом заполнения меньше 30%, то статистическая временная сложность АНП меньше  $O(n^{1.5} \log n)$  и близка к линейной.

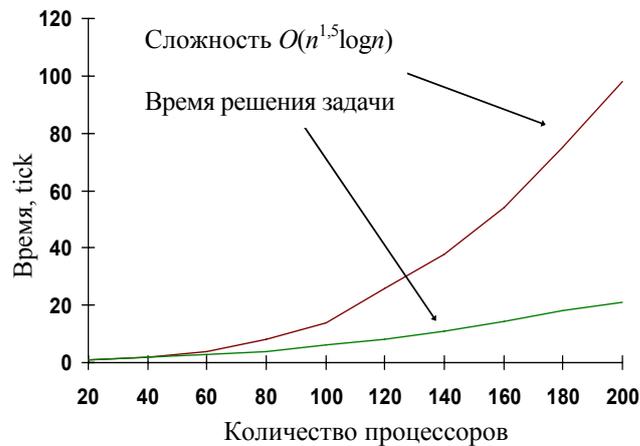


Рис. 7. Сравнение времен решения задачи планирования

## ВЫВОДЫ

Анализ наиболее известных подходов решения задач назначения (метод отжига и решение задачи с помощью оценочных функций) показывает, что они не могут быть приняты в качестве базовых для динамического планировщика, так как в первом случае временные параметры процесса поиска решения не удовлетворяют ограничениям по времени планирования, а во втором случае при приемлемом времени планирования качество планирования ухудшается. Наиболее приемлемым является метод направленного поиска, позволяющий осуществить поиск решения наилучшим способом, зависящим от исходной информации.

Венгерский алгоритм поиска максимального паросочетания во взвешенном двудольном графе удовлетворяет предъявленным требованиям, так как обеспечивает направленный поиск решения, соответствует отношению работа–ресурс в предлагаемой модели системы пространственного планирования. Классический венгерский алгоритм в каждой итерации выполняет поиск максимального паросочетания для невзвешенного двудольного графа с использованием алгоритма Карпа–Хопкрофта и имеет временную сложность  $O(n^3)$ . Она определяется алгоритмом Карпа–Хопкрофта с временной сложностью  $O(n^{2.5})$ .

Замена алгоритма Карпа–Хопкрофта другим алгоритмом с меньшей временной сложностью позволит снизить общую временную сложность венгерского алгоритма. Элементы теории построения адаптивного поиска максимального паросочетания являются основой этого решения, так как поиск максимального паросочетания для невзвешенного двудольного графа выполняется при степени заполнения МС не более 30%, что позволяет применить предложенные алгоритмы выполнения обязательных назначений и принцип исключающего планирования.

Использование принципа исключающего планирования способствует в значительной степени повысить эффективность системы динамического планирования с сохранением качества решения в заданных пределах.

Предложенные в работе процедуры подготовки исходных данных для планирования в реальном времени позволяют использовать модифицированный венгерский алгоритм в качестве базового для пространственного планировщика нижнего уровня систем реального времени.

## ЛИТЕРАТУРА

1. *Douglas T.* Distributed computing in practice: The Condor experience / Т. Douglas, Т. Tannenbaum, М. Livny // *Concurrency and Computation: Practice and Experience*. — 2005. — N 2. — P. 323–356.
2. *Tannenbaum Andrew S.* Distributed Systems: Principles and Paradigms (2nd Edition) / Andrew S. Tanenbaum, Maarten van Steen. — Upper Saddle River, NJ, USA: Prentice-Hall, Inc. — 2006.
3. *Метод опережающего планирования для ГРИД* / В.Н. Коваленко, Е.И. Коваленко, Д.А. Корягин, Э.З. Любимский // Препринт ИПМ. — 2005. — № 112. — [http://www.keldysh.ru/papers/2005/prep112/prep2005\\_112.html](http://www.keldysh.ru/papers/2005/prep112/prep2005_112.html).
4. *Platform LSF 7 Update 6. An Overview of New Features for Platform LSF Administrators.* Официальный сайт компании Platform Computing Corporation — 2009. — [http://www.platform.com/workload-management/whatsnew\\_lsf7u6.pdf](http://www.platform.com/workload-management/whatsnew_lsf7u6.pdf).

5. *Microsoft Windows Compute Cluster Server 2003*. Руководство пользователя — 2006. — [https://msdb.ru/Downloads/WindowsServer2003/CCS/CCS2003Guide\\_Rus.pdf](https://msdb.ru/Downloads/WindowsServer2003/CCS/CCS2003Guide_Rus.pdf).
6. *TORQUE Resource Manager Guide*. Официальный сайт компании Cluster Resources Inc. — 2009. — <http://www.clusterresources.com/products/torque-resource-manager.php>.
7. *PBS Works*. Официальный сайт компании Altair Engineering, Inc. — 2006. — <http://www.pbsworks.com/>.
8. *Ding X.* BWS: balanced work stealing for time-sharing multicores / X. Ding, K. Wang, P.B. Gibbons, X. Zhang // Proceedings of the 7-th ACM European Conference on Computer Systems. — EuroSys '12. — New York, NY, USA: ACM. — 2012. — P. 365–378.
9. *What is Condor?* Официальный сайт продукта Condor. — 2006. — <http://www.cs.wisc.edu/condor/description.html>.
10. *IBM Tivoli Workload Scheduler LoadLeveler*. Официальный сайт компании «Интерфейс» — 2007. — <http://www.interface.ru/home.asp?artId=6283>.
11. *Maui Scheduler Administrator's Guide*. Официальный сайт компании Cluster Resources Inc. — 2008. — <http://www.clusterresources.com/products/maui/docs/index.shtml>.
12. *Moab Workload Manager*. Официальный сайт компании Cluster Resources Inc. — 2008. — <http://www.clusterresources.com/products/moab-cluster-suite/workload-manager.php>.
13. *Симоненко А.В.* Выбор стратегии пространственного планирования в параллельных вычислительных системах / А.В. Симоненко // Информатика, управління та обчислювальна техніка. — К., 2001. — **35**. — С. 104–108.
14. *Симоненко А.В.* Система пространственного распределения заданий в распределённых вычислительных системах / А.В. Симоненко, С.В. Пих, Н.В. Слущкий, В.В. Воробйов // Информатика, управління та обчислювальна техніка. — 2012. — **56**. — С. 170.
15. *Kaufmann A.* Introduction a la combinatorique en vue des applications / A. Kaufmann. — Dunod, Paris. — 1968.
16. *Пападимитриу К.* Комбинаторная оптимизация. Алгоритмы и сложность / К. Пападимитриу, К. Стайглиц. — М.: Мир. — 1985.
17. *Berge C.* Theorie des graphes et ses application / C. Berge. — Dunod, Paris. — 1958.
18. *Casavant T.L.* A taxonomy of scheduling in general-purpose distributed computing systems // IEEE Trans. Softw.Eng.14 ,(1988) / T.L. Casavant, J.G. Kuhl .— P. 141–154.
19. *Blazevicz J.* Scheduling independent multiprocessor tasks on a uniform k-processor system / J. Blazevicz, M. Drozdowski, G. Schmidt, D. De Werra // Parallel Computer. — 1994. — **20**. — P. 15–28.
20. *Elsadek A.A.* Heuristic model for task allocation in a heterogeneous distributed systems / A.A. Elsadek, B.E. Wells // proc. of PDPTA'96, California USA. — 1996. — **2**. — P. 659–671.
21. *Freund R.F.* Generational Scheduling for Heterogeneous Computing Systems / R.F. Freund, B.R. Carter, D. Watson et al. // proc. of PDPTA'96, California USA, —1996. — **2**. — P. 769–778.
22. *Hopcroft J.E.* An  $n^{2.5}$  Algorithm for Maximum Matchings in Bipartite Graphs // SIAM J. Comput. 2(4) / J.E. Hopcroft, R.M. Karp. — 1973. — P. 225–231.
23. *Tan M., Antonio J.K, et. al.* Scheduling and data relocation for sequentially executed subtasks in a heterogeneous computing system / M. Tan, J.K. Antonio et. al. // HCW'95. — 1995. — P. 109–120.

Поступила 07.12.2015