

УДК 004.453

doi:10.31799/1684-8853-2023-2-15-26

EDN: WWXRRW

Научные статьи
Articles

Алгоритм метода ветвей и границ оптимизации расписаний выполнения пакетов заданий в конвейерных системах

К. В. Кротов^а, канд. техн. наук, доцент, orcid.org/0000-0002-9670-6141, krotov_k1@mail.ru

^аСевастопольский государственный университет, Университетская ул., 33, Севастополь, 299053, РФ

Введение: оптимизация расписаний выполнения пакетов заданий обеспечивает эффективную реализацию производственных и вычислительных процессов. Современные методы оптимизации расписаний характеризуются ограничениями на размерность задач либо невозможностью получить решения, приближающиеся к глобально оптимальному. **Цель:** разработать алгоритм метода ветвей и границ оптимизации расписаний выполнения пакетов заданий в конвейерных системах. **Результаты:** получена математическая модель многостадийных процессов, позволяющая идентифицировать моменты времени начала выполнения пакетов заданий в соответствующих позициях в последовательностях реализации действий с ними на приборах конвейерных систем. Представлен критерий оптимизации решений, соответствующий моменту времени окончания выполнения заданий, включенных в пакеты. Сформулирован способ разбиения множеств решений на их подмножества (ветвления вершин дерева), который предусматривает добавление по одному пакету разных типов в последовательности реализации действий с ними из множеств не включенных в них пакетов. Разработан способ построения решений по порядкам выполнения на приборах пакетов, входящих в сформированные подмножества, для которых вычисляются значения критерия, используемые при обновлении верхних оценок. Синтезирован способ определения значений нижних оценок критерия для множеств решений, соответствующих вершинам дерева, полученным в результате ветвления, а также алгоритм метода ветвей и границ. **Практическая значимость:** исследования, проведенные с использованием программной реализации алгоритма, показали, что он позволяет до 35 % сократить время на выполнение пакетов по сравнению с решениями без оптимизации. При малом количестве типов заданий алгоритм позволяет получить решение, соответствующее глобально оптимальному. При увеличении количества типов заданий точность приближения решений к глобально оптимальному составляет 0,9–0,95.

Ключевые слова – конвейерные системы, пакеты заданий, метод ветвей и границ, расписания выполнения пакетов заданий.

Для цитирования: Кротов К. В. Алгоритм метода ветвей и границ оптимизации расписаний выполнения пакетов заданий в конвейерных системах. *Информационно-управляющие системы*, 2023, № 2, с. 15–26. doi:10.31799/1684-8853-2023-2-15-26, EDN: WWXRRW

For citation: Krotov K. V. A branch and bound algorithm for the optimization of batch scheduling in pipeline systems. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2023, no. 2, pp. 15–26 (In Russian). doi:10.31799/1684-8853-2023-2-15-26, EDN: WWXRRW

Введение

Выполнение в конвейерных системах (КС) заданий разных типов, входящих в наборы, предусматривает формирование пакетов для реализации операций с ними на приборах систем. Решение задачи оптимизации расписаний выполнения пакетов заданий (ПЗ) в КС предполагает решение подзадач: 1) оптимизации составов пакетов; 2) оптимизации расписаний выполнения пакетов в КС. Автором предложены следующие способы решения указанных задач: 1) метод многоуровневой оптимизации расписаний выполнения ПЗ в КС [1]; 2) метод ветвей и границ (МВГ) для оптимизации решений по составам ПЗ при построении расписаний их выполнения в КС на основе эвристического правила [2]. В практических приложениях актуальным является решение задачи оптимизации расписаний выполнения ПЗ на приборах КС.

Современные методы оптимизации расписаний выполнения ПЗ могут быть объединены

в соответствии с типом обрабатываемых систем: 1) системы с одной машиной обработки заданий в составе пакетов (Batch Processing Machine – BPM); 2) системы потокового типа (Flow Shop), в состав которых входят машины последовательной обработки (Job Processing Machine – JPM) и BPM; 3) системы с последовательной обработкой единичных заданий на первой стадии выполнения операций с ними (системы Flow Shop либо Job Shop) и с BPM на второй стадии. В работах [3, 4] рассматриваются способы оптимизации составов пакетов для их выполнения на одной BPM. Синтезированы модели смешанного целочисленного линейного программирования (mixed-integer linear programming – MILP). Особенностью предложенной в [3] модели является учет зависимости времени настройки приборов от размеров партий. При распределении заданий по пакетам в [5] используется алгоритм муравьиной колонии, позволяющий группировать задания, имеющие приближенно одинаково-

вую длительность выполнения, и реализовать обмен заданиями между пакетами для снижения простоев приборов в ВРМ. В [6, 7] для определения составов ПЗ и расписаний их выполнения на приборах одной ВРМ применены эвристические правила, не гарантирующие приближения к оптимальному решению. В [8] рассматривается решение задачи построения расписаний выполнения ПЗ в системах второго типа. Для построения расписаний выполнения ПЗ использован метод обратного планирования, предусматривающий: 1) выбор пакета с наибольшим количеством заданий, определение момента времени начала действий с ним таким образом, чтобы момент времени окончания его выполнения совпадал с директивным сроком окончания для всех заданий; 2) при упорядочивании пакетов по невозрастанию количества заданий в них применение указанного подхода для каждого пакета. В [9] рассматривается система, содержащая несколько ВРМ. Способ определения составов пакетов основывается на условии, что включаемые в них задания имеют различные моменты времени поступления в систему. В соответствии с этими значениями реализуется упорядочивание заданий, на основе сформированной последовательности определяются составы пакетов с учетом пропускной способности ВРМ. Работа [10] посвящена синтезу модели МПР для оптимизации расписаний в системах с машинами в следующей последовательности: первая и третья — ЖРМ, вторая — ВРМ. В соответствии с порядком выполнения единичных заданий на ЖРМ определяются составы ПЗ и порядки их выполнения на ВРМ. Значительное количество переменных и ограничений обуславливают невозможность оптимизации решений за ограниченное время при значительной размерности задач. Особенностью системы в [11] является наличие ВРМ, приборы которых могут быть включены в многостадийные процессы обработки ПЗ по одному, обеспечивая тем самым их гибкость. Сформулирован алгоритм табу-поиска, в котором выделены уровни для: 1) перераспределения заданий между пакетами; 2) упорядочивания пакетов на машинах; 3) упорядочивания заданий в пакетах. Синтезированы операторы: 1) вставки, реализующий добавление заданий в пакеты для одной либо разных ВРМ; 2) замены, реализующий замену заданий в пакетах либо пакетов на разных машинах. Обработывающая система, рассматриваемая в [12], содержит ЖРМ на первой и третьей стадиях обработки партий деталей и ВРМ на второй стадии. На третьей стадии ЖРМ выполняет функции контроля деталей, получаемых на первой и второй стадиях, включаемых в контрольные партии для идентификации бракованных среди них. Авторами [12] получены

формулы для вычисления: 1) количества и составов производственных и контрольных партий; 2) рангов партий, в соответствии с которыми реализуется их упорядочивание на приборах. В [13] конвейер обработки ПЗ представлен в виде последовательности вершин графа потока управления, в котором каждой вершине соответствует определенное приложение. Разработанная авторами [13] программа реализует моделирование перемещения пакетов в конвейере с целью определять оптимальные параметры функционирования системы (количество пакетов, поступающих на вход приложений в единицу времени, размеры пакетов) и оптимальное распределение ресурсов между приложениями. Система, рассматриваемая в [14], содержит на первом этапе Job Shop, а на заключительной стадии обработки — ВРМ. По окончании обработки в Job Shop детали размещаются в буфере перед ВРМ. Задача распределения заданий по пакетам решается при различных моментах времени поступления их в буфер.

Современное состояние применения МВГ для оптимизации расписаний характеризуется работами [15–20], в которых решаются следующие задачи:

- оптимизация составов ПЗ и расписаний их выполнения на одном приборе с целью минимизировать затраты на производство изделий в составе партий, незавершенное производство [15];
- построение расписаний выполнения единичных заданий в системах Flow Shop [16];
- оптимизация расписаний выполнения единичных заданий на параллельных приборах [17];
- построение расписаний выполнения проектов при ограничениях на возобновляемые ресурсы и директивные сроки окончания [18];
- проектное планирование при разработке программ [19];
- построение расписаний выполнения заданий при условии нечетких длительностей реализации действий с ними на приборах [20].

Таким образом, применение МВГ для задач оптимизации расписаний выполнения ПЗ заданных составов в КС не реализовано, требуется развить МВГ для решения задачи оптимизации расписаний выполнения ПЗ в КС.

Математическая модель многостадийных процессов выполнения пакетов заданий в конвейерных системах

Синтез модели процессов выполнения ПЗ в КС обеспечивается введением обозначений для входных данных, параметров и решений, соответствующих порядкам реализации действий с ПЗ. Введены обозначения: N — количество типов заданий, выполняемых в КС; i — иден-

тификатор типа заданий, пакеты которых выполняются в КС ($i = \overline{1, N}$); l – идентификаторы приборов КС ($l = \overline{1, L}$); π^l – последовательности выполнения ПЗ на l -х приборах КС, соответствующие порядкам реализации действий с ними; $\mathbf{M} = (m_1, m_2, \dots, m_N)^T$ – вектор количества ПЗ i -х типов; $\mathbf{A} = \|a_{ih}\|_{N \times m_{\max}}$ – матрица составов ПЗ ($m_{\max} = \max_i(m_i)$); N_p – количество ПЗ, размещаемых в последовательностях π^l , порядок выполнения которых в КС оптимизируется; $\mathbf{T} = \|t_{li}\|_{L \times N}$ – матрица длительностей выполнения заданий на приборах КС; $\mathbf{T}^l = \|t_{ii'}^l\|_{N \times N}$ ($l = \overline{1, L}$) – матрицы длительностей переналадок l -х приборов с выполнения заданий i -х на выполнение заданий i' -х типов ($t_{ii}^l = 0$); j – номера позиций, которые занимают ПЗ в последовательностях π^l ($l = \overline{1, L}$); $\mathbf{R} = \|r_{ij}\|_{N \times N_p}$ – матрица количества заданий i -х типов в пакетах, занимающих в π^l j -е позиции; $\mathbf{T}^{pl} = \|t_{ij}^{pl}\|_{N \times N_p}$ – матрицы моментов времени начала выполнения ПЗ i -х типов, занимающих j -е позиции в последовательностях π^l .

Постановка задачи оптимизации расписаний выполнения ПЗ в КС формулируется следующим образом. Параметры процесса выполнения ПЗ в КС: 1) количество типов заданий N и приборов L ; 2) матрицы: \mathbf{T} длительностей выполнения заданий и \mathbf{T}^l ($l = \overline{1, L}$) длительностей переналадок приборов. Входными данными являются: вектор \mathbf{M} количества ПЗ и матрица \mathbf{A} составов ПЗ. Результатом являются: 1) матрица \mathbf{R} порядка выполнения ПЗ в КС; 2) матрицы \mathbf{T}^{pl} моментов времени начала выполнения ПЗ i -х типов, занимающих в π^l j -е позиции.

Математическая модель процессов выполнения ПЗ в КС обеспечивает определение значений элементов t_{ij}^{pl} матриц \mathbf{T}^{pl} на основе матриц \mathbf{T} , \mathbf{T}^l ($l = \overline{1, L}$) и матрицы \mathbf{R} порядка их выполнения. Определение значений t_{ij}^{pl} реализуется с использованием значений простоев l -х приборов в ожидании начала выполнения ПЗ i -х типов в j -х позициях в π^l ($l = \overline{1, L}$), обозначенных как Pr_{ij}^l . Выражения для вычисления значений Pr_{ij}^l получены в следующем виде:

$$-l = 1 : Pr_{ij}^1 = 0 \quad (i = \overline{1, N}; j = \overline{1, N_p}); \quad (1)$$

$$-l > 1 : Pr_{ij}^l = \max[0, (t_{ij}^{\pi^{(l-1)}} + r_{ij} \cdot t_{(l-1),i}) - (t_{i',j-1}^{pl} + r_{i',j-1} \cdot t_{i'}^l + t_{i',i}^l)], \quad (2)$$

где $j = \overline{2, N_p}$; i' – тип заданий в пакете, занимающем в последовательностях π^l $(j-1)$ -ю позицию, предшествующую j -й позиции пакета i -го типа.

Выражения для вычисления t_{ij}^{pl} ($l = \overline{1, L}$) получены в виде

$$-l = 1 : t_{i1}^{\pi^1} = 0 \quad (j = 1);$$

$$t_{ij}^{\pi^1} = t_{i',j-1}^{\pi^1} + r_{i',j-1} \cdot t_{i'}^1 + t_{i',i}^1 \quad (j = \overline{2, N_p}); \quad (3)$$

$$-l > 1 : t_{i1}^{pl} = \sum_{g=1}^{l-1} \sum_{s=1}^N t_{gs} \cdot r_{s1} \quad (j = 1);$$

$$t_{ij}^{pl} = t_{i',j-1}^{pl} + r_{i',j-1} \cdot t_{i'}^l + t_{i',i}^l + Pr_{ij}^l \quad (j = \overline{2, N_p}), \quad (4)$$

где i' интерпретируется по аналогии. В результате математическая модель процессов выполнения ПЗ в КС представлена выражениями (1)–(4).

Критерий оптимальности расписаний представляет собой момент времени окончания выполнения ПЗ на L -м приборе и определяется выражением вида $\max_{j=1, N_p} (t_{i,j}^{pl} + r_{i,j} \cdot t_{L,i})$. Задача оптимизации расписаний имеет вид

$$\min f(\mathbf{R}), \quad (5)$$

где $f(\mathbf{R}) = \max_{j=1, N_p} (t_{i,j}^{pl} + r_{i,j} \cdot t_{L,i})$.

Ограничения множества допустимых расписаний имеют вид:

1) ограничение на количество заданий i -го типа в пакетах в последовательностях π^l их выполнения на приборах КС: $\sum_{j=1}^{N_p} r_{ij} = \sum_{h=1}^{m_i} a_{ih}$;

2) ограничение на общее количество заданий N типов в пакетах в последовательностях π^l их выполнения на приборах КС: $\sum_{i=1}^N \sum_{j=1}^{N_p} r_{ij} = \sum_{i=1}^N \sum_{h=1}^{m_i} a_{ih}$.

Алгоритм метода ветвей и границ оптимизации расписаний выполнения ПЗ в КС

Метод ветвей и границ использован для оптимизации порядка выполнения ПЗ на приборах КС (последовательностей π^l ($l = \overline{1, L}$) реализации действий с ПЗ в КС). Для разработки алгоритма оптимизации расписаний выполнения ПЗ в КС, использующего МВГ, в рассмотрение введены обозначения: l_k – идентификатор вершины в дереве МВГ, формируемой на k -й итерации алгоритма (соответствующей размещению ПЗ i -го типа в j -й позиции в последовательностях π^l их выполнения в КС); G^{l_k} – множество решений, предусматривающих размещение одного h_i -го пакета i -го типа в j -й позиции в последовательностях π^l ($l = \overline{1, L}$), соответствующее l_k -й вершине; O^{l_k} – нижняя оценка значений критерия f , вы-

числяемая для l_k -й вершины, соответствующей G^{l_k} ; O^b — верхняя оценка значений критерия f , соответствующая решению-рекорду; $H_{i1}^{l_k}$ — множества номеров h_i ПЗ i -х типов ($i = 1, N$), которые сопоставлены l_k -й вершине дерева МВГ в целях реализации процедуры ветвления (ПЗ размещаются в $(j + 1)$ -й позиции в последовательностях π^l ($l = 1, L$) их выполнения в КС ($h_i \in H_{i1}^{l_k}$)); $H_{i2}^{l_k}$ — копии множеств $H_{i1}^{l_k}$ номеров h_i ПЗ i -х типов ($i = 1, N$), используемые при вычислении значения O^{l_k} и значения критерия f для решения, входящего в G^{l_k} ; $R^{l_k} = \left\| r_{ij}^{l_k} \right\|_{N \times N_p}$ — матрица количества заданий i -х типов в пакетах, занимающих j -е позиции в π^l , соответствующая l_k -й вершине дерева МВГ; $R_1^{l_k} = \left\| (r_{ij}^{l_k})_1 \right\|_{N \times N_p}$ — копия матрицы R^{l_k} , модифицируемая при вычислении O^{l_k} и значения критерия f для решения, входящего в G^{l_k} ; $T_{l_k}^{pl} = \left\| (t_{ij})_{l_k}^{pl} \right\|_{N \times N_p}$ — матрицы моментов времени начала выполнения ПЗ i -х типов, занимающих j -е позиции в π^l ($l = 1, L$), соответствующие l_k -й вершине дерева для вычисления O^{l_k} и значения критерия f ; I, I' — векторы типов заданий, в которых они упорядочены по убыванию средней длительности переналадок; N_I и $N_{I'}$ — количество элементов в I и I' ; j' — номер позиции, в которой в π^l размещаются оставшиеся не добавленными задания разных типов при вычислении O^{l_k} и значения критерия f ; Q_k — множество номеров l_k -х вершин k -го уровня дерева.

Разработка алгоритма МВГ включает синтез процедур: 1) разбиение множеств решений G^{l_k} на подмножества $G^{l_{k+1}}$ (ветвление l_k -х вершин дерева); 2) формирование решений по порядку выполнения ПЗ в π^l , входящих в множества G^{l_k} , для которых реализуется вычисление значений критерия f , используемых при определении значения оценки O^b и вычислении нижних оценок O^{l_k} для множеств G^{l_k} . Также требуется явно определить условия: 1) исключения бесперспективных множеств решений G^{l_k} , соответствующих l_k -м вершинам, из дальнейшего рассмотрения; 2) остановки алгоритма МВГ при получении оптимального решения.

Каждой вершине дерева соответствуют кортеж параметров $\langle k, l_k, i, h_i, j \rangle$, группа множеств $H_{i1}^{l_k}$ ($i = 1, N$), содержащих номера h_i пакетов i -х типов, которые будут включены в последовательности π^l на следующей итерации алгоритма (сопоставлены вершинам $(k + 1)$ -го уровня дерева МВГ). Реализации ветвления предшествует формирование множеств $H_{i1}^{l_k}$ ($i = 1, N$) номеров ПЗ i -х типов, которые размещаются в π^l (множества H_{i1}^0 , соответствующие корневой вершине G^0 , имеют вид $H_{i1}^0 = \{1, 2, \dots, m_i\}$). Процедура ветвле-

ния вершин дерева предполагает, что множество G^0 корневой вершины разбивается на подмножества решений G^{l_1} $\left(l_1 = \overline{1, L_1}, L_1 = \sum_{i=1}^N m_i \right)$, каждое

из которых предполагает размещение в $(j = 1)$ -й позиции в последовательностях π^l одного h_i -го пакета i -го типа ($i = 1, N$). Формируемым вершинам ставятся в соответствие кортежи $\langle 1, l_1, i, h_i, 1 \rangle$

$\left(l_1 = \overline{1, L_1}, L_1 = \sum_{i=1}^N m_i, h_i = \overline{1, m_i}, i = 1, N \right)$. После

формирования $(k = 1)$ -го уровня дерева МВГ и сопоставления каждой его вершине определенного h_i -го ПЗ i -го типа ($h_i = \overline{1, m_i}, i = 1, N$) реализуется модификация множеств $H_{i1}^{l_1}$ ($i = 1, N$).

Из множества $H_{i1}^{l_1}$ для заданий i -го типа, пакет которых сопоставлен с l_1 -й вершиной $(k = 1)$ -го уровня дерева МВГ, исключается идентификатор пакета h_i ($H_{i1}^{l_1} = H_{i1}^{l_1} \setminus \{h_i\}$). При реализации процедуры ветвления l_1 -й вершины формируются l_2 -е вершины на $(k = 2)$ -м уровне, каждой из которых соответствует множество G^{l_2} решений, предусматривающих размещение в $(j = 2)$ -й позиции в последовательностях π^l h_i -х пакетов i' -х типов ($i' \neq i$), идентификаторы которых принадлежат множествам $H_{i1}^{l_1}$ ($i = 1, N$), соответствующих l_1 -й вершине, подвергающейся ветвлению. Для l_2 -й вершины, полученной при ветвлении l_1 -й вершины, множества $H_{i1}^{l_2}$ ($i = 1, N$) имеют вид: 1) если $i = i'$, то $H_{i1}^{l_2} = H_{i1}^{l_1} \setminus \{h_i\}$; 2) если $i \neq i'$, то $H_{i1}^{l_2} = H_{i1}^{l_1}$. При ветвлении l_k -й вершины формируются (l_{k+1}) -е вершины, каждой из которых соответствует кортеж $\langle k + 1, l_{k+1}, i', h_i, j + 1 \rangle$ и множества $H_{i1}^{l_{k+1}}$. Каждой l_{k+1} -й вершине, получаемой в результате ветвления, сопоставлено множество решений $G^{l_{k+1}}$, предусматривающих размещение одного h_i -го пакета такого, что $h_i \in H_{i1}^{l_k}$ в текущей $(j + 1)$ -й позиции в последовательностях π^l их выполнения на приборах КС ($H_{i1}^{l_k}$ — множества идентификаторов ПЗ, не размещенных в последовательностях π^l , соответствующих l_k -й вершине).

Способ вычисления оценки O^{l_k} и значения критерия f для решения, входящего в множество G^{l_k} (используемого при возможном обновлении значения верхней оценки O^b), предусматривает формирование матрицы R^{l_k} порядка выполнения h_i -х пакетов, добавленных в π^l к текущей k -й итерации алгоритма. Идентификаторы i -х типов заданий, h_i -х пакетов, номера позиций j пакетов в последовательностях π^l , входящих в кортежи $\langle k, l_k, i, h_i, j \rangle$ для вершин, используются при инициализации значений элементов $r_{ij}^{l_k}$ матрицы R^{l_k} следующим образом: $r_{ij}^{l_k} = a_{ih}$ ($h = h_i, a_{ih}$ — элементы матрицы A , соответствующие

количеству заданий в пакетах, занимающих j -е позиции в последовательностях π^l). В итоге формируется матрица \mathbf{R}^{l_k} , соответствующая порядку выполнения ПЗ в π^l для рассматриваемой l_k -й вершины. Матрица \mathbf{R}^{l_k} передается при ветвлении l_k -й вершины дочерней (l_{k+1} -й) вершине в целях ее дополнения значением количества заданий в пакете, размещаемом в следующей ($j + 1$)-й позиции в π^l .

На основе матрицы \mathbf{R}^{l_k} формируется матрица $\mathbf{R}_1^{l_k}$, используемая при расчете O^{l_k} и значения критерия f для решения, входящего в G^{l_k} , используемого при обновлении оценки O^B . Алгоритм модификации матрицы $\mathbf{R}_1^{l_k}$ для вычисления O^{l_k} и значения критерия f (для обновления O^B) содержит шаги:

1. Присвоить счетчику N_{Γ} значение $N: N_{\Gamma} = N$. Присвоить j' значение позиции j из кортежа, соответствующего рассматриваемой l_k -й вершине: $j' = j$.

2. Если $N_{\Gamma} = 0$, то перейти на п. 4. Если $N_{\Gamma} \neq 0$, то определить номер s' элемента в векторе \mathbf{I}' , соответствующего типу заданий $i'_s: s' = \min s$. В соответствии с s' определить в \mathbf{I}' тип оставшихся заданий, которые размещаются в позиции j' в $\pi^l: i = i'_s$. Удалить i'_s из вектора \mathbf{I}' , модифицировать $N_{\Gamma}: N_{\Gamma} = N_{\Gamma} - 1$.

3. Модифицировать номер j' позиции в последовательностях π^l , в которой размещаются оставшиеся задания i -го типа: $j' = j' + 1$. Инициализировать элемент $(r_{ij'})_1^{l_k}$ матрицы $\mathbf{R}_1^{l_k}: (r_{ij'})_1^{l_k} = n^i - \sum_{s=1}^{j'} (r_{is})^{l_k}$. Перейти на п. 2.

4. Останов алгоритма.

В матрице $\mathbf{R}_1^{l_k}$ элементы столбцов с 1-го по j -й соответствуют ПЗ, добавленным в эти последовательности π^l при реализации процедуры ветвления l_k -х вершин. Элементы столбцов с индексами, большими j , формируются с учетом размещения в π^l оставшихся заданий i -х типов ($i = 1, N$), не размещенных в них до этого. Процедура вычисления оценки O^{l_k} предполагает реализацию этапов:

1. Расчет значений элементов матриц $\mathbf{T}_{l_k}^{pl}$ ($l = \overline{1, L}$) для ПЗ, размещенных в π^l в позициях с 1-ю по j -ю, где j – элемент кортежа $\langle k, l_k, i, h_i, j \rangle$ для l_k -й вершины, для которой вычисляется оценка O^{l_k} . Для этого предусматривается использование матрицы $\mathbf{R}_1^{l_k}$ и выражений (1)–(4).

2. Расчет значений элементов матрицы $\mathbf{T}_{l_k}^{pl}$ (для L -го прибора) для «остатков» заданий, размещенных в π^l в позициях, больших j -й. Для этого предусматривается использование матрицы $\mathbf{R}_1^{l_k}$, а также формулы вида

$$(t_{ij+1})^{pl} = (t_{i,j})^{pl} + r_{i,j} \cdot t_{L,i'} + t_{i,j}^L \quad (6)$$

На основе значений элементов матрицы $\mathbf{T}_{l_k}^{pl}$ нижняя оценка O^{l_k} для множества G^{l_k} определяется как момент времени окончания выполнения ПЗ на L -м приборе КС, а также оставшихся не размещенными в π^l заданий разных типов:

$$O^{l_k} = (t_{i,N_p})^{pl} + r_{i,N_p} \cdot t_{L,i}, \quad (7)$$

где N_p – позиция последнего ПЗ i -го типа в π^l на L -м приборе.

Способ формирования матрицы $\mathbf{R}_1^{l_k}$, способ расчета значений элементов матриц $\mathbf{T}_{l_k}^{pl}$ и $\mathbf{T}_{l_k}^{pl}$ соответствуют размещению в последовательностях π^l всех ПЗ для вершин, лежащих на пути из нулевой в рассматриваемую l_k -ю вершину, и размещению всех оставшихся заданий в последовательности π^l на L -м приборе.

Обновление значения верхней оценки O^B обеспечивается расчетом значения критерия f для сформированного порядка выполнения ПЗ в π^l ($l = \overline{1, L}$), которому соответствует матрица $\mathbf{R}_1^{l_k}$, синтезированная с использованием приведенного выше способа (для каждой l_k -й вершины). Вычисление значения f для решения, представленного матрицей $\mathbf{R}_1^{l_k}$, обеспечивается определением значений элементов $(t_{ij})_{l_k}^{pl}$ матриц $\mathbf{T}_{l_k}^{pl}$ ($l = \overline{1, L}$) в соответствии с выражениями (1)–(4). Значение критерия f для расписания выполнения ПЗ в КС, характеризуемого матрицей $\mathbf{R}_1^{l_k}$, определяется выражением

$$f(\mathbf{R}_1^{l_k}) = (t_{i,N_p})^{pl} + r_{i,N_p} \cdot t_{L,i}, \quad (8)$$

где i – тип заданий, пакет которых занимает последнюю N_p -ю позицию в последовательностях π^l ($l = \overline{1, L}$). При формировании l_k -х вершин на некотором k -м уровне дерева МВГ пересчет верхней оценки и обновление рекорда реализуется при условии $\min f(\mathbf{R}_1^{l_k}) < O^B$ (где $\min f(\mathbf{R}_1^{l_k})$ определяется среди значений критерия f , соответствующих решениям из множеств G^{l_k} для l_k -х вершин k -го уровня дерева).

Условие исключения дальнейшего ветвления l_k -х вершин дерева МВГ в целях сокращения мощности множества расписаний выполнения ПЗ в КС (отбрасывания бесперспективных направлений) имеет вид $O^{l_k} \geq O^B$, где O^{l_k} – нижняя оценка для множества решений G^{l_k} , соответствующего l_k -й вершине.

Условие останова алгоритма предполагает, что если для решения по порядкам выполнения ПЗ в КС, характеризуемого матрицей $\mathbf{R}_1^{l_k}$, получено значение критерия $f(\mathbf{R}_1^{l_k})$, для которого выполняется условие $f(\mathbf{R}_1^{l_k}) = O^{l_k} \leq O^B$ (l_k – индексы других вершин рассматриваемого k -го уровня

дерева МВГ, $l'_k \neq l_k$), то это решение является оптимальным, и дальнейший поиск прекращается.

Вычисление верхней оценки O^B и определение решения-рекорда для множества G^0 предусматривает, что сформированы: 1) множества номеров пакетов заданий i -х типов H_{i1}^0 ($H_{i1}^0 = \{1, 2, \dots, m_i\}$ ($i = 1, N$)); 2) векторы \mathbf{I} и \mathbf{I}' , элементами которых являются типы заданий, упорядоченные по неубыванию средней длительности переналадок приборов. Для сформированной матрицы \mathbf{R}_1^0 с использованием выражений (1)–(4) определяются значения элементов матриц \mathbf{T}_0^{nl} ($l = \overline{1, L}$). На основе значений элементов матрицы \mathbf{T}_0^{nl} идентифицируется значение критерия f для множества G^0 , которое выступает в качестве верхней оценки O^B , соответствующее ему решение является рекордом.

Алгоритм МВГ, позволяющий реализовать оптимизацию расписаний выполнения ПЗ на приборах КС, имеет следующий порядок шагов:

1. Для корневой вершины дерева МВГ (множества G^0) сформировать множества H_{i1}^0 ($i = \overline{1, N}$) в виде $H_{i1}^0 = \{1, 2, \dots, m_i\}$. Сформировать вектор типов заданий \mathbf{I} . Инициализировать вектор \mathbf{I}' : $\mathbf{I}' = \mathbf{I}$. Инициализировать элементы r_{ij}^0 матрицы \mathbf{R}^0 значениями 0. Инициализировать матрицу \mathbf{R}_1^0 : $\mathbf{R}_1^0 = \mathbf{R}^0$. Выполнить процедуру модификации матрицы \mathbf{R}_1^0 . На основе матрицы \mathbf{R}_1^0 с использованием выражений (1)–(4) определить значения элементов $(t_{ij})_0^{nl}$ матриц \mathbf{T}_0^{nl} . На основе матрицы \mathbf{T}_0^{nl} вычислить значение критерия f по формуле (8), соответствующее множеству G^0 . Инициализировать матрицу \mathbf{R}^B , соответствующую решению-рекорду: $\mathbf{R}^B = \mathbf{R}_1^0$, оценку O^B инициализировать значением критерия f .

2. Задать значение номера k итерации алгоритма равным 1 ($k = 1$). Инициализировать множество Q_1 : $Q_1 \neq \emptyset$. Выполнить формирование дочерних l_1 -х вершин дерева МВГ, используя для этого множества H_{i1}^0 ($i = \overline{1, N}$). Каждой сформированной вершине дерева МВГ поставить в соответствие кортеж вида $\langle 1, l_1, i, h_i, 1 \rangle$. Добавить идентификаторы сформированных l_1 -х вершин, полученных в результате ветвления, в множество Q_1 : $Q_1 = Q_1 \cup \{l_1\}$ $\left(l_1 = \overline{1, L}, L_1 = \sum_{i=1}^N m_i, |Q_1| = L_1 \right)$.

Для каждой дочерней l_1 -й вершины выполнить инициализацию множеств H_{i1}^l , используя множества H_{i1}^0 корневой вершины. Для каждой l_1 -й вершины выполнить модификацию множества H_{i1}^l , соответствующего типу заданий i из кортежа $\langle 1, l_1, i, h_i, 1 \rangle$: $H_{i1}^l = H_{i1}^l \setminus \{h_i\}$, где h_i – номер пакета, размещенного в $(j = 1)$ -й позиции в π^l ($l = \overline{1, L}$). Инициализировать векторы \mathbf{I} и \mathbf{I}' . Для каждой l_1 -й вершины на основе матрицы \mathbf{R}^0

сформировать матрицы \mathbf{R}^l , а затем их модифицировать путем инициализации элементов r_{i1}^l значением элемента a_{i,h_i} матрицы \mathbf{A} , индексы которого соответствуют значениям элементов кортежа для этой вершины. На основе матрицы \mathbf{R}^l сформировать матрицу \mathbf{R}_1^l . Для каждой l_1 -й вершины с использованием множеств H_{i1}^l ($i = \overline{1, N}$) путем реализации соответствующей процедуры модифицировать матрицу \mathbf{R}_1^l . Для каждой l_1 -й вершины на основе матрицы \mathbf{R}_1^l определить значение нижней оценки O^l для множества G^l с использованием рассмотренной процедуры. Для каждой l_1 -й вершины на основе сформированной матрицы \mathbf{R}_1^l с использованием выражений (1)–(4) определить значения элементов $(t_{ij})_l^{nl}$ матриц \mathbf{T}_1^{nl} ($l = \overline{1, L}$), а также значение $f(\mathbf{R}_1^l)$ для решения, входящего в множество G^l . Если для l_1 -х вершин выполняется условие $\min f(\mathbf{R}_1^l) < O^B$, то верхняя оценка O^B модифицируется ($O^B = \min f(\mathbf{R}_1^l)$).

3. Если для одной из l_k -х вершин k -го уровня дерева выполняется условие $f(\mathbf{R}_1^k) = O^k \leq O^B$ (O^k – нижние оценки l'_k -х вершин, принадлежащих k -му уровню дерева), то инициализировать матрицу \mathbf{R}^B : $\mathbf{R}^B = \mathbf{R}_1^k$. Перейти на п. 19. Если условие $f(\mathbf{R}_1^k) = O^k \leq O^B$ не выполняется ни для одной l_k -й вершины на k -м уровне дерева МВГ, то идентифицировать l_k -е вершины ($l_k \in Q_k$) такие, что $O^k \geq O^B$. Идентификаторы l_k -х вершин k -го уровня дерева МВГ, для которых $O^k \geq O^B$, исключить из множества Q_k : $Q_k = Q_k \setminus \{l_k\}$. Сформировать множество Q_k^B вершин – кандидатов на ветвление: $Q_k^B = Q_k$.

4. Если $Q_k^B = \emptyset$, то перейти на п. 18. Если $Q_k^B \neq \emptyset$, то среди l_k -х вершин ($l_k \in Q_k^B$) выбрать l_k^B -ю вершину, для которой $O^{l_k^B} = \min(O^{l_k} | l_k \in Q_k^B)$.

5. Модифицировать множество Q_k^B : $Q_k^B = Q_k^B \setminus \{l_k^B\}$. Если для l_k^B -й вершины $\bigcup_{i=1}^N H_{i1}^{l_k^B} = \emptyset$, то перейти на п. 4. Если $\bigcup_{i=1}^N H_{i1}^{l_k^B} \neq \emptyset$, то перейти на п. 6.

6. С использованием множеств $H_{i1}^{l_k^B}$ для l_k^B -й вершины сформировать дочерние вершины (количество сформированных дочерних вершин определяется как $\sum_{i=1}^N |H_{i1}^{l_k^B}|$). Модифицировать множество $H_{i1}^{l_k^B}$, соответствующее типу i заданий и номеру h_i пакета заданий в кортеже $\langle k, l_k^B, i, h_i, j \rangle$ для l_k^B -й вершины следующим образом: $H_{i1}^{l_k^B} = H_{i1}^{l_k^B} \setminus \{h_i\}$.

7. Инициализировать идентификатор (номер) l_{k+1} -й вершины на $(k + 1)$ -м уровне дерева МВГ

значением 1: $l_{k+1} = 1$. Инициализировать Q_{k+1} : $Q_{k+1} = \emptyset$.

8. Извлечь идентификатор l'_k -й вершины из множества Q_k в соответствии с условием $l'_k = \min(l_k | l_k \in Q_k)$. Исключить идентификатор l'_k -й вершины из множества Q_k : $Q_k = Q_k \setminus \{l'_k\}$. Если $l'_k = l_k^B$, то перейти на п. 10. Если $l'_k \neq l_k^B$, то присвоить рассматриваемой вершине идентификатор l_{k+1} вершины на $(k + 1)$ -м уровне и поставить ей в соответствие кортеж параметров вида $\langle k + 1, l_{k+1}, i, h_i, j \rangle$, в котором значения параметров i, h_i, j соответствуют значениям параметров из кортежа l'_k -й вершины. Поставить в соответствие l_{k+1} -й вершине множества $H_{i1}^{l_{k+1}}$ ($i = 1, N$) следующим образом: $H_{i1}^{l_{k+1}} = H_{i1}^{l_k}$ ($i = 1, N$). Инициализировать матрицу $R^{l_{k+1}}$ для l_{k+1} -й вершины: $R^{l_{k+1}} = R^{l_k}$. Дополнить множество Q_{k+1} номером l_{k+1} -й вершины: $Q_{k+1} = Q_{k+1} \cup \{l_{k+1}\}$. Модифицировать идентификатор l_{k+1} вершины на $(k + 1)$ -м уровне дерева МВГ: $l_{k+1} = l_{k+1} + 1$.

9. Если $l'_k < l_k^B$, то перейти на п. 8.

10. Исключить идентификатор l_k^B -й вершины из Q_k : $Q_k = Q_k \setminus \{l_k^B\}$. Инициализировать множества $H_{i2}^{l_k^B}$ ($i = 1, N$) для l_k^B -й вершины: $H_{i2}^{l_k^B} = H_{i1}^{l_k^B}$ ($i = 1, N$). Задать тип заданий i' , пакеты которых будут сопоставляться с вершинами, полученными в результате ветвления, равным 1. Модифицировать номер позиции j ПЗ из кортежа $\langle k, l_k^B, i, h_i, j \rangle$, соответствующего l_k^B -й вершине: $j = j + 1$.

11. Если для i' -го типа заданий выполняется условие $i' = i$ (где i — тип заданий в кортеже, соответствующем l_k^B -й вершине), то перейти на п. 14.

12. Если для множества $H_{i2}^{l_k^B}$, соответствующего рассматриваемому i' -му типу заданий, выполняется условие $H_{i2}^{l_k^B} = \emptyset$, то перейти на п. 14. Если $H_{i2}^{l_k^B} \neq \emptyset$, то извлечь из $H_{i2}^{l_k^B}$ номер пакета, размещаемого в j -й позиции в последовательностях π^l их выполнения на приборах КС, модифицировать множество $H_{i2}^{l_k^B}$: $h_{i'} = \min\{h_i | h_i \in H_{i2}^{l_k^B}\}$, $H_{i2}^{l_k^B} = H_{i2}^{l_k^B} \setminus \{h_{i'}\}$. С использованием значений: номера l_{k+1} -й вершины дерева МВГ, идентификатора типа заданий i' , идентификатора пакета заданий $h_{i'}$, номера j позиции пакета в последовательностях π^l — инициализировать параметры кортежа вершины, являющейся дочерней l_k^B -й вершине: $\langle k + 1, l_{k+1}, i', h_{i'}, j \rangle$. Инициализировать матрицу $R^{l_{k+1}}$, соответствующую добавленной вершине: $R^{l_{k+1}} = R^{l_k^B}$. Модифицировать матрицу $R^{l_{k+1}}$ путем инициализации элемента $r_{i'j}^{l_{k+1}} : r_{i'j}^{l_{k+1}} = a_{i'h_{i'}}$ ($a_{i'h_{i'}}$ — соответствующий элемент матрицы A). Дополнить множество Q_{k+1} :

$Q_{k+1} = Q_{k+1} \cup \{l_{k+1}\}$. Инициализировать множества $H_{i1}^{l_{k+1}}$ для l_{k+1} -й вершины: $H_{i1}^{l_{k+1}} = H_{i1}^{l_k^B}$.

13. Модифицировать номер вершины: $l_{k+1} = l_{k+1} + 1$. Перейти на п. 12.

14. Модифицировать идентификатор i' типа заданий, пакеты которых добавляются в последовательности π^l в рассматриваемую j -ю позицию: $i' = i' + 1$. Если $i' \leq N$, то модифицировать значение идентификатора l_{k+1} вершины на $(k + 1)$ -м уровне дерева МВГ: $l_{k+1} = l_{k+1} + 1$, перейти на п. 12. Если $i' > N$, то модифицировать идентификатор l_{k+1} -й вершины: $l_{k+1} = l_{k+1} + 1$. Перейти на п. 15.

15. Извлечь идентификатор l'_k -й вершины из множества Q_k в соответствии с условием $l'_k = \min(l_k | l_k \in Q_k)$. Исключить идентификатор l'_k -й вершины из множества Q_k : $Q_k = Q_k \setminus \{l'_k\}$. Присвоить вершине идентификатор l_{k+1} и поставить ей в соответствие кортеж параметров $\langle k + 1, l_{k+1}, i, h_i, j \rangle$, в котором значения параметров i, h_i, j соответствуют значениям кортежа параметров l'_k -й вершины. Поставить в соответствие рассматриваемой l_{k+1} -й вершине множества $H_{i1}^{l_{k+1}}$ ($i = 1, N$): $H_{i1}^{l_{k+1}} = H_{i1}^{l_k}$. Инициализировать матрицу $R^{l_{k+1}}$: $R^{l_{k+1}} = R^{l_k}$. Дополнить множество Q_{k+1} номером l_{k+1} -й вершины: $Q_{k+1} = Q_{k+1} \cup \{l_{k+1}\}$.

16. Если $Q_{k+1} \neq \emptyset$, то модифицировать значение идентификатора l_{k+1} вершины: $l_{k+1} = l_{k+1} + 1$, перейти на п. 15. Если $Q_{k+1} = \emptyset$, то перейти на п. 17.

17. Модифицировать значение идентификатора k текущего рассматриваемого уровня дерева МВГ: $k = k + 1$. Инициализировать векторы I и I' . Для каждой l_k -й вершины инициализировать матрицы $R_1^{l_k}$ на основе матриц $R^k : R_1^{l_k} = R^k$. Для каждой l_k -й вершины с использованием множеств $H_{i1}^{l_k}$ ($i = 1, N$) модифицировать матрицу $R_1^{l_k}$ и определить значение нижней оценки O^{l_k} для множества G^{l_k} . Для каждой l_k -й вершины на основе сформированной матрицы $R_1^{l_k}$ с использованием выражений (1)–(4) вычислить значения элементов матриц $T_{l_k}^{pl}$ ($l = 1, L$), а также значение $f(R_1^{l_k})$ для решения, входящего в множество G^{l_k} . Если для l_k -х вершин выполняется условие $\min f(R_1^{l_k}) < O^B$, то модифицировать значение O^B : $O^B = \min f(R_1^{l_k})$. Перейти на п. 3.

18. Для каждой l_k -й вершины k -го уровня ($l_k \in Q_k$) на основе сформированной матрицы R^{l_k} с использованием выражений (1)–(4) вычислить значения элементов $(t_{ij})_{l_k}^{pl}$ матриц $T_{l_k}^{pl}$ ($l = 1, L$), а также значение $f(R^{l_k})$ для решения, которому соответствует эта матрица. Среди решений по порядкам выполнения ПЗ на приборах КС, сопоставленных l_k -м вершинам ($l_k \in Q_k$), идентифицировать решение, для которого

$R^{l_k^*} = \arg \min f(R^{l_k})$, и соответствующую ему l_k^* -ю вершину. С использованием $R^{l_k^*}$ инициализировать решение-рекорд R^B : $R^B = R^{l_k^*}$.

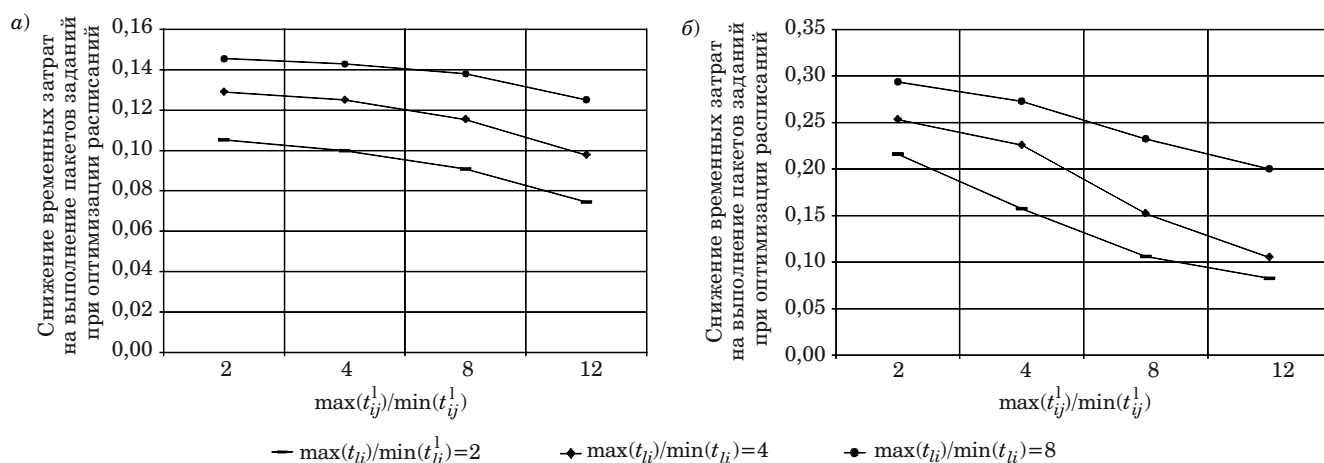
19. Останов алгоритма.

Исследование эффективности применения алгоритма МВГ при оптимизации расписаний выполнения ПЗ в КС

Для исследования эффективности применения МВГ при оптимизации расписаний выполнения ПЗ в КС использованы параметры: неоднородность длительностей выполнения заданий — $\max(t_{ij})/\min(t_{ij})$; неоднородность длительностей переналадок — $\max(t_{ij}^l)/\min(t_{ij}^l)$. Значения $\max(t_{ij}^l)/\min(t_{ij}^l) \in \{1, 2, 4, 8, 12\}$; значения $\max(t_{ij})/\min(t_{ij}) \in \{1, 2, 4, 8\}$. Значения $n \in \{2, 3, 5\}$, значения $m_i \in \{2, 4, 6\}$, $n^i = 6$ при $m_i = 2$, $n^i = 12$ при $m_i = 4$, $n^i = 24$ при $m_i = 6$. Значение $L = 3$. В качестве результатов фиксируются зависимости значений отношения $f_{с.в.в.п} = (f_{фикс} - f_{о.р.п})/f_{фикс}$ от входных параметров ($f_{с.в.в.п}$ — снижение времени выполнения пакетов при оптимизации расписаний по сравнению с фиксированными ПЗ, $f_{фикс}$ — значение критерия для фиксированных ПЗ, $f_{о.р.п}$ — значение критерия при оптимизации расписаний пакетов с использованием МВГ). Графики, отображающие зависимость значений $f_{с.в.в.п}$ от значений входных параметров задачи, представлены рис. 1–3. Вычислительная сложность алгоритма оценена как $O(n \cdot m_i^2)$, где n — количество типов заданий, m_i — количество пакетов каждого типа ($i = \overline{1, N}$). Оценка требуемой памяти для реализации алгоритма — $O(nm_i)$.

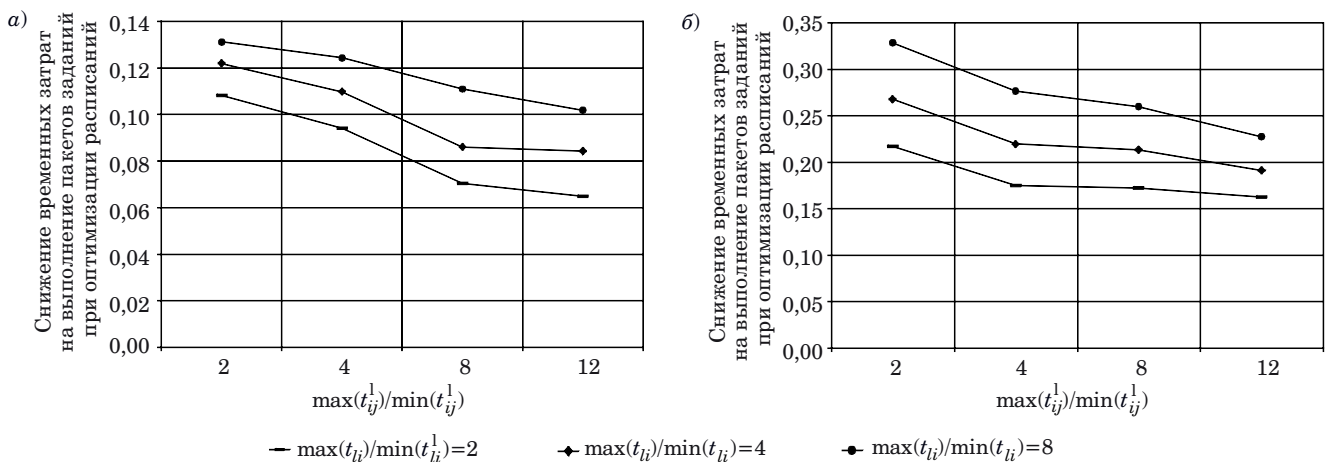
Интервалы времени получения результатов не превышают: при $n \in \{2, 3\}$ и $m_i = 2$ — 1 мин; при $n \in \{2, 3\}$ и $m_i = 4$ — 3 мин; при $n = 2$ и $m_i = 6$, $n = 5$ и $m_i = 2$ — 10 мин. Программа реализована с использованием языка С# в среде разработки Visual Studio. Анализ результатов показал, что использование МВГ позволяет снизить время выполнения ПЗ при оптимизации расписаний на 30–35 % по сравнению с не оптимизируемыми решениями. При $n = 2$ МВГ формирует оптимальное решение, соответствующее решению, полученному при полном переборе. При $n = 3$ точность приближения решения к глобально оптимальному составила 0,9–0,95.

Применение разработанного алгоритма метода МВГ оптимизации расписаний выполнения ПЗ в КС достаточно обширно. В частности, целесообразно его применение в системах оптимизации расписаний (планирования) многостадийных процессов обработки партий деталей в многономенклатурных механообрабатывающих производствах (планирование процессов реализуется с учетом динамически поступающих заказов на выпуск продукции), а также при оптимизации расписаний (планировании вычислительных процессов) обработки пакетов данных дистанционного зондирования Земли в составе web-ориентированных сервисов мониторинга ее поверхности в целях идентификации наличия на ней негативных природных явлений и техногенных воздействий на окружающую среду, их характеристик и условий распространения [21] (массивы значений отражательной способности земной поверхности, включаемые в пакеты, получены строго в одних и тех же каналах от одного и того же искусственного спутника Земли и соответствуют областям на ее поверхности



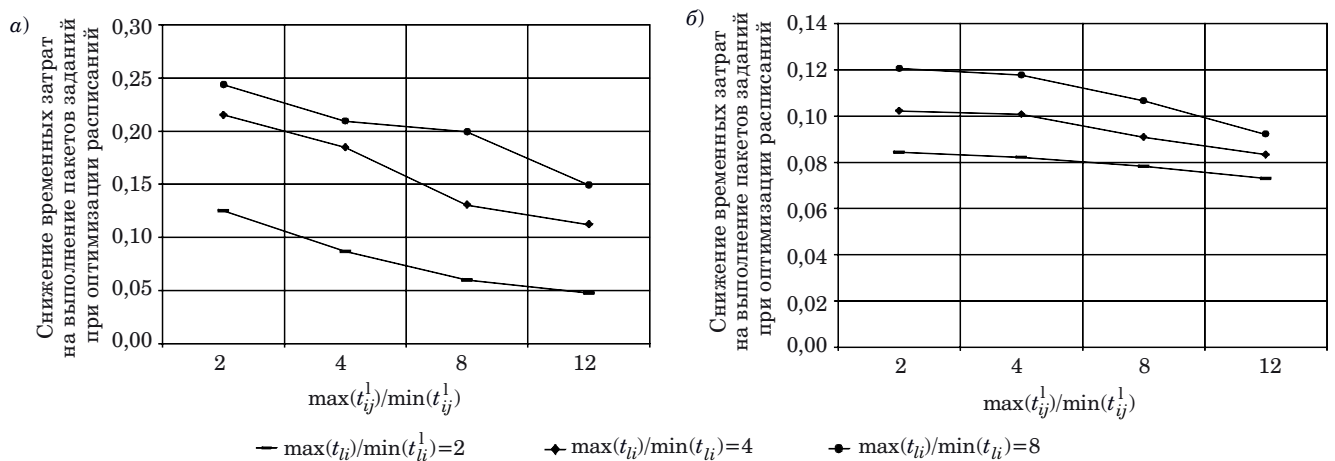
■ **Рис. 1.** Уменьшение времени выполнения пакетов заданий при оптимизации расписаний для: а — $n = 2$, $m_i = 2$ ($i = \overline{1, 2}$); б — $n = 2$, $m_i = 4$ ($i = \overline{1, 2}$)

■ **Fig. 1.** Decrease of the time of execution of the batches of tasks when optimizing schedules for: а — $n = 2$, $m_i = 2$ ($i = \overline{1, 2}$); б — $n = 2$, $m_i = 4$ ($i = \overline{1, 2}$)



■ **Рис. 2.** Уменьшение времени выполнения пакетов заданий при оптимизации расписаний для: а – $n = 2, m_i = 6$ ($i = \overline{1, 2}$); б – $n = 3, m_i = 2$ ($i = \overline{1, 3}$)

■ **Fig. 2.** Decrease of the time of execution of the batches of tasks when optimizing schedules for: а – $n = 2, m_i = 6$ ($i = \overline{1, 2}$); б – $n = 3, m_i = 2$ ($i = \overline{1, 3}$)



■ **Рис. 3.** Уменьшение времени выполнения пакетов заданий при оптимизации расписаний для: а – $n = 3, m_i = 4$ ($i = \overline{1, 3}$); б – $n = 5, m_i = 2$ ($i = \overline{1, 5}$)

■ **Fig. 3.** Decrease of the time of execution of the batches of tasks when optimizing schedules for: а – $n = 3, m_i = 4$ ($i = \overline{1, 3}$); б – $n = 5, m_i = 2$ ($i = \overline{1, 5}$)

одинакового размера). Составы партий деталей, выпускаемых на приборах КС, определяются заказами от потребителей, поступающими на предприятие. Оптимизация расписаний обработки партий деталей в КС позволяет значительно сократить время, затрачиваемое на их выпуск. Оптимизация расписаний выполнения ПЗ на обработку массивов данных дистанционного зондирования Земли позволяет значительно сократить время отклика на запросы пользователей по определению в заданных ими областях на земной поверхности наличия на ней негативных природных явлений и техногенных воздействий на окружающую среду, их характеристик и условий распространения [21].

Заключение

Современные методы построения расписаний выполнения ПЗ предусматривают, что обрабатываемые системы включают ВРМ. Эти методы характеризуются ограничением на размерность задач, а также невозможностью гарантированного получения решения, приближающегося к глобально оптимальному. Алгоритмы МВГ не ориентированы на оптимизацию расписаний многостадийных процессов выполнения ПЗ в КС. В связи с этим разработана математическая модель процессов выполнения ПЗ в КС, способы вычисления нижней и верхней оценок множеств решений, способ ветвления вершин в дереве

МВГ, а также непосредственно алгоритм МВГ для оптимизации расписаний выполнения ПЗ в КС. Использование разработанного алгоритма МВГ для оптимизации расписаний выполнения

ПЗ позволяет до 35 % сократить временные затраты на реализацию действий с ними по сравнению с не оптимизируемыми расписаниями.

Литература

1. **Кротов К. В.** Комплексный метод определения эффективных решений по составам партий данных и расписаниям их обработки в конвейерных системах. *Вычислительные технологии*, 2018, т. 23, № 3, с. 58–76.
2. **Кротов К. В.** Математическая модель и алгоритм метода ветвей и границ для оптимизации решений по составам пакетов в многостадийных системах. *Информатика и автоматизация*, 2022, т. 21, № 1, с. 5–40. doi:10.15622/ia.2022.21.1
3. **Shaoxiang Z., Naiming X., Qiao W.** Single batch machine scheduling with dual setup times for autoclave molding manufacturing. *Computers and Operations Research*, 2021, vol. 133, iss. C. doi:10.1016/j.cor.2021.105381
4. **Ogun B., Cigdem A.-U.** Mathematical models for a batch scheduling problem to minimize earliness and tardiness. *Journal of Industrial Engineering and Management*, 2018, vol. 11, iss. 3, pp. 390–405. doi:10.3926/jiem.2541
5. **Li X. L., Wang Y.** Scheduling batch processing machine using max–min ant system algorithm improved by a local search method. *Mathematical Problems in Engineering*, 2018, vol. 2018, Art. ID: 3124182. doi:10.1155/2018/3124182
6. **Jin M., Liu X., Luo W.** Single-machine parallel-batch scheduling with nonidentical job sizes and rejection. *Mathematics*, 2020, vol. 8, Art. ID: 258. doi:10.3390/math8020258
7. **Fan J.** Integrated scheduling problem on a single bounded batch machine with an unavailability constraint. *Discrete Dynamics in Nature and Society*, 2020, vol. 12, Art. ID: 8625849. doi:10.1155/2020/8625849
8. **Halim A. H., Hidayat N. P. A., Aribowo W.** Single item batch-scheduling flow shop with m batch-processing machines to minimize total actual flow time. *International Journal of Technology*, 2022, vol. 13, no. 4, pp. 816–826. doi:https://doi.org/10.14716/ijtech.v13i4.4869
9. **Hertrich Ch., Weiß Ch., Ackermann H., Heydrich S., Krumke S. O.** Scheduling a proportionate flow shop of batching machines. *Journal of Scheduling*, 2020, vol. 23, pp. 575–593. doi:10.1007/s10951-020-00667-2
10. **Mauluddin Yu.** Three-stage flow-shop scheduling model with batch processing machine and discrete processing machine. *The 3rd Annual Applied Science and Engineering Conf.*, 2018, vol. 197, Art. ID: 14002. doi:10.1051/mateconf/201819714002
11. **Shahvari O., Logendran R.** A comparison of two stage-based hybrid algorithms for a batch scheduling problem in hybrid flow shop with learning effect. *International Journal of Production Economics*, 2018, vol. 195, pp. 227–248.
12. **Suryadhini P. P., Sukoyo S., Suprayogi S., Halim A. H.** A batch scheduling model for a three-stage flow shop with job and batch processors considering a sampling inspection to minimize expected total actual flow time. *Journal of Industrial Engineering and Management*, 2021, vol. 14, no. (3), pp. 520–537. doi:10.3926/jiem.3438
13. **Lévai T., Németh F., Raghavan B., Rétvári G.** Batching: batch-scheduling data flow graphs with service-level objectives. *17th USENIX Symp. on Networked Systems Design and Implementation (NSDI'20)*, 2020, February 25–27, Santa Clara, CA, USA, pp. 633–649.
14. **Song L., Liu Ch., Shi H., Zhu J.** An improved immune genetic algorithm for solving the flexible job shop scheduling problem with batch processing. *Wireless Communications and Mobile Computing*, vol. 2022, Art. ID: 2856056. doi:10.1155/2022/2856056
15. **Indrapriyatna A. S., Triha H. A.** Modified branch and bound algorithm for batch scheduling in discrete manufacture system. *IOP Conf. Series: Materials Science and Engineering*, 2018, vol. 319, Art. ID: 012074. doi:10.1088/1757-899X/319/1/012074
16. **Gmys J., Mezmaz M., Melab N., Tuytens D.** A computationally efficient branch-and-bound algorithm for the permutation flow-shop scheduling problem. *European Journal of Operational Research*, 2020, vol. 284, iss. 3, pp. 814–833. doi:10.1016/j.ejor.2020.01.039
17. **Mazda Ch. N., Kurniawati D. A.** Branch and bound method to overcome delay delivery order in flow shop scheduling problem. *IOP Conf. Series: Materials Science and Engineering*, 2020, IOP Publishing. doi:10.1088/1757-899X/1003/1/012129
18. **Watermeyer K., Zimmermann J.** A branch-and-bound procedure for the resource-constrained project scheduling problem with partially renewable resources and general temporal constraints. *OR Spectrum*, 2020, vol. 42, pp. 427–460. doi:10.1007/s00291-020-00583-z
19. **Su Ch.-H., Wang J. Y.** A branch-and-bound algorithm for minimizing the total tardiness of multiple developers. *Mathematics*, 2022, vol. 10, Art. ID: 1200. doi:10.3390/math10071200
20. **Santi S., Selvakumari K.** Flow shop scheduling problem in neutrosophic environment using trapezoidal fuzzy numbers. *Eurasian Journal of Analytical Chemistry*, 2019, vol. 14(1), pp. 111–115.

21. Кротов К. В., Скатков А. В. Организация web-ориентированного сервиса мониторинга окружающей среды с использованием данных дистанционного зондирования Земли и конвейеризации обра-

ботки данных. *Труды учебных заведений связи*, 2021, т. 7, № 1, с. 105–121. <https://doi.org/10.31854/1813-324X-2021-7-1-105-121>

UDC 004.453

doi:10.31799/1684-8853-2023-2-15-26

EDN: WWXRRW

A branch and bound algorithm for the optimization of batch scheduling in pipeline systems

K. V. Krotov^a, PhD, Tech., Associate Professor, orcid.org/0000-0002-9670-6141, krotov_k1@mail.ru

^aSevastopol State University, 33, Universitetskaya St., Sevastopol, 299053, Russian Federation

Introduction: The optimization of batch scheduling ensures efficiency in production and computational processes. Modern methods of scheduling optimization are characterized by limitations on the dimensionality of tasks or the impossibility of obtaining solutions that approach the globally optimal ones. **Purpose:** To develop a branch and bound algorithm for the optimization of batch scheduling in pipeline systems. **Results:** We obtain a mathematical model of multistage processes that allows to identify the point in time when batch jobs start running in the appropriate positions in the sequences of the implementation of actions with them on the devices of pipeline systems. We present the criterion for optimizing solutions that corresponds to the point in time when the tasks included into the batch stop running. We formulate a method of partitioning sets of solutions into their subsets (branch tree vertices). This method involves adding a package of different types one at a time to the sequence of implementation of actions with them from the sets not included into them. We develop a method for constructing solutions according to the order of execution on the devices of the packages which are included in the formed subsets. For these subsets we calculate the values of the criterion used when updating the upper estimates. We synthesize a method for determining the values of the lower estimates of the criterion for sets of solutions corresponding to the vertices of the tree obtained as a result of branching. A branch and bound algorithm is also developed. **Practical relevance:** The studies that are conducted with the use of the software implementation of the algorithm have shown that it allows up to 35% decrease in the time for batch jobs execution as compared to the solutions without optimization. With a small number of task types, the algorithm allows one to get a solution which approximates the globally optimal one. With the increase in the number of task types, the accuracy of approximation of the solutions to the globally optimal ones is 0.9–0.95.

Keywords – pipeline systems, batch jobs, branch and bound method, batch scheduling.

For citation: Krotov K. V. A branch and bound algorithm for the optimization of batch scheduling in pipeline systems. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2023, no. 2, pp. 15–26 (In Russian). doi:10.31799/1684-8853-2023-2-15-26, EDN: WWXRRW

References

1. Krotov K. V. The complex method for determining effective solutions for data batch sizes and processing schedules in conveyor systems. *Computing Technologies*, 2018, vol. 23, no. 3, pp. 58–76 (In Russian).
2. Krotov K. V. Mathematical model and algorithm of branch and boundary method for optimizing solutions for package compositions in multi-stage systems. *Informatics and Automation*, 2022, vol. 21, no. 1, pp. 5–40 (In Russian). doi:10.15622/ia.2022.21.1
3. Shaoxiang Z., Naiming X., Qiao W. Single batch machine scheduling with dual setup times for autoclave molding manufacturing. *Computers and Operations Research*, 2021, vol. 133, iss. C. doi:10.1016/j.cor.2021.105381
4. Ogun B., Cigdem A.-U. Mathematical models for a batch scheduling problem to minimize earliness and tardiness. *Journal of Industrial Engineering and Management*, 2018, vol. 11, iss. 3, pp. 390–405. doi:10.3926/jiem.2541
5. Li X. L., Wang Y. Scheduling batch processing machine using max-min ant system algorithm improved by a local search method. *Mathematical Problems in Engineering*, 2018, vol. 2018, Art. ID: 3124182. doi:10.1155/2018/3124182
6. Jin M., Liu X., Luo W. Single-machine parallel-batch scheduling with non identical job sizes and rejection. *Mathematics*, 2020, vol. 8, Art. ID: 258. doi:10.3390/math8020258
7. Fan J. Integrated scheduling problem on a single bounded batch machine with an unavailability constraint. *Discrete Dynamics in Nature and Society*, 2020, vol. 12, Art. ID: 8625849. doi:10.1155/2020/8625849
8. Halim A. H., Hidayat N. P. A., Aribowo W. Single item batch-scheduling flow shop with m batch-processing machines to minimize total actual flow time. *International Journal of Technology*, 2022, vol. 13, no. 4, pp. 816–826. doi:<https://doi.org/10.14716/ijtech.v13i4.4869>
9. Hertrich Ch., Weiß Ch., Ackermann H., Heydrich S., Krumke S. O. Scheduling a proportionate flow shop of batching machines. *Journal of Scheduling*, 2020, vol. 23, pp. 575–593. doi:10.1007/s10951-020-00667-2
10. Mauluddin Yu. Three-stage flow-shop scheduling model with batch processing machine and discrete processing machine. *The 3rd Annual Applied Science and Engineering Conf.*, 2018, vol. 197, Art. ID: 14002. doi:10.1051/mateconf/201819714002
11. Shahvari O., Logendran R. A comparison of two stage-based hybrid algorithms for a batch scheduling problem in hybrid flow shop with learning effect. *International Journal of Production Economics*, 2018, vol. 195, pp. 227–248.
12. Suryadhini P. P., Sukoyo S., Suprayogi S., Halim A. H. A batch scheduling model for a three-stage flow shop with job and batch processors considering a sampling inspection to minimize expected total actual flow time. *Journal of Industrial Engineering and Management*, 2021, vol. 14, no. (3), pp. 520–537. doi:10.3926/jiem.3438
13. Lévai T., Németh F., Raghavan B., Rétvári G. Batchy: batch-scheduling data flow graphs with service-level objectives. *17th USENIX Symp. on Networked Systems Design and Implementation (NSDI'20)*, 2020, February 25–27, Santa Clara, CA, USA, pp. 633–649.
14. Song L., Liu Ch., Shi H., Zhu J. An improved immune genetic algorithm for solving the flexible job shop scheduling problem with batch processing. *Wireless Communications and Mobile Computing*, vol. 2022, Art. ID: 2856056. doi:10.1155/2022/2856056
15. Indrapriyatna A. S., Triha H. A. Modified branch and bound algorithm for batch scheduling in discrete manufacture system. *IOP Conf. Series: Materials Science and Engineering*, 2018, vol. 319, Art. ID: 012074. doi:10.1088/1757-899X/319/1/012074
16. Gmys J., Mez maz M., Melab N., Tuyttens D. A computationally efficient branch-and-bound algorithm for the permutation flow-shop scheduling problem. *European Journal of*

- Operational Research*, 2020, vol. 284, iss. 3, pp. 814–833. doi:10.1016/j.ejor.2020.01.039
17. Mazda Ch. N., Kurniawati D. A. Branch and bound method to overcome delay delivery order in flow shop scheduling problem. *IOP Conf. Series: Materials Science and Engineering*, 2020, IOP Publishing. doi:10.1088/1757-899X/1003/1/012129
 18. Watermeyer K., Zimmermann J. A branch-and-bound procedure for the resource-constrained project scheduling problem with partially renewable resources and general temporal constraints. *OR Spectrum*, 2020, vol. 42, pp. 427–460. doi:10.1007/s00291-020-00583-z
 19. Su Ch.-H., Wang J. Y. A branch-and-bound algorithm for minimizing the total tardiness of multiple developers. *Mathematics*, 2022, vol. 10, Art. ID: 1200. doi:10.3390/math10071200
 20. Santi S., Selvakumari K. Flow shop scheduling problem in neutrosophic environment using trapezoidal fuzzy numbers. *Eurasian Journal of Analytical Chemistry*, 2019, vol. 14(1), pp. 111–115.
 21. Krotov K. V., Skatkov A. V. Organization of web-based environmental monitoring service using earth remote sensing data and pipelining data processing. *Proc. of Telecommunication Universities*, 2021, vol. 7, no. 1, pp. 105–121 (In Russian). <https://doi.org/10.31854/1813-324X-2021-7-1-105-121>

УВАЖАЕМЫЕ АВТОРЫ!

Научные базы данных, включая Scopus и Web of Science, обрабатывают данные автоматически. С одной стороны, это ускоряет процесс обработки данных, с другой — различия в транслитерации ФИО, неточные данные о месте работы, области научного знания и т. д. приводят к тому, что в базах оказывается несколько авторских страниц для одного и того же человека. В результате для всех по отдельности считаются индексы цитирования, что снижает рейтинг ученого.

Для идентификации авторов в сетях Thomson Reuters проводит регистрацию с присвоением уникального индекса (ID) для каждого из авторов научных публикаций.

Процедура получения ID бесплатна и очень проста, есть возможность провести регистрацию на 12 языках, включая русский (чтобы выбрать язык, кликните на зеленое поле вверху справа на стартовой странице): <https://orcid.org>