

ИНФОРМАТИКА,  
ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА  
И УПРАВЛЕНИЕ

INFORMATICS,  
COMPPUTER ENGINEERING  
AND CONTROL

УДК 519.854.3

DOI: 10.17212/1814-1196-2018-3-87-106

## Эффективный параметрический алгоритм оптимизации расписаний параллельных систем с заданным расписанием начала обслуживания\*

Ю.А. МЕЗЕНЦЕВ<sup>а</sup>, И.В. ЭСТРАЙХ<sup>б</sup>

630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет

<sup>а</sup> [mesyan@yandex.ru](mailto:mesyan@yandex.ru), [mezencev@corp.nstu.ru](mailto:mezencev@corp.nstu.ru)    <sup>б</sup> [estrajx@corp.nstu.ru](mailto:estrajx@corp.nstu.ru)

В статье предложен параметрический полиномиально трудоемкий алгоритм оптимизации расписаний параллельной системы с задержками на входе, который дополнен улучшающим обменным алгоритмом. Критерием эффективности в рассматриваемой постановке является максимальное быстродействие системы ( $C_{\max}$ ), что обеспечивает равномерную загрузку приборов. Наличие этого критерия, а также задержек начала обслуживания поступающих заявок делает задачу труднорешаемой с априорной оценкой любого аппроксимационного алгоритма, равной двум. Предложенный параметрический алгоритм основан на схеме динамического программирования модифицированной адаптивным сужением области поиска, что обеспечивает его вычислительную эффективность. Сформулированные задачи синтеза и предложенный алгоритм оптимизации расписаний параллельных обслуживающих систем ориентированы на практическое применение в календарном планировании и оперативном регулировании производственных процессов с дискретным характером. Оценена трудоемкость синтеза точных и приближенных к оптимальным по быстродействию расписаниям. Приведен иллюстративный пример использования представленных формальных постановок и алгоритма для синтеза оптимальных расписаний параллельной системы рассматриваемого типа. Проведенные вычислительные эксперименты на специально сгенерированных реализациях задачи больших размерностей показали хорошие практические результаты. Приведена итоговая сравнительная статистика точности и времени счета посредством программных реализаций параметрического и обменного алгоритмов с результатами альтернативных инструментов, показывающая неоспоримое преимущество представленных в статье алгоритмов. Экспериментально доказана вычислительная эффективность связи алгоритма, основанного на процедуре динамического программирования с отсевом части вариантов на каждом шаге и обменного алгоритма, что позволяет обеспечить приемлемое быстродействие и близость к оптимумам. Апостериорная оценка близости к оптимумам получаемых решений на доступных для оценки тестах не превысила 5 %.

**Ключевые слова:** оптимальное расписание, параллельные производственные системы, несвязанные параллельные системы, задержки начала обслуживания, критерий быстродействия, дискретная оптимизация, эффективный параметрический алгоритм, обменный улучшающий алгоритм

---

\* Статья получена 30 марта 2018 г.

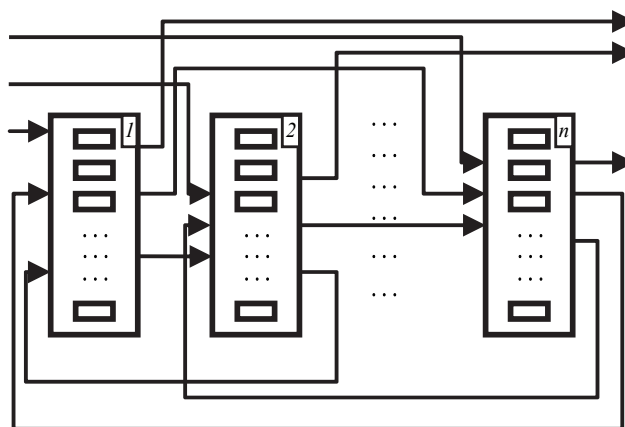
## ВВЕДЕНИЕ

Наиболее общей и актуальной задачей теории расписаний применительно к производственным системам является задача синтеза расписаний параллельно-последовательных обслуживающих систем. Общую постановку такой задачи, связанных с ней подзадач и описание некоторых алгоритмов решения можно найти в работах [1–13]. Целью настоящей статьи является описание логического завершения работы над важнейшими компонентами инструментария синтеза оптимальных по быстродействию расписаний общей задачи [4]. Таковыми являются эффективные алгоритмы оптимизации расписаний параллельной системы с задержками начала обслуживания. Для пояснения связи между одностадийной параллельной системой с заданным расписанием поступления заявок и многостадийной параллельно-последовательной системой рассмотрим графическое изображение примера последней (рисунок) [5].

Типичная параллельно-последовательная система состоит из совокупности блоков, или подсистем, содержащих взаимозаменяемые в общем случае неидентичные приборы. На производстве это группы взаимозаменяемого оборудования. На рисунке они пронумерованы от 1 до  $n$ . Будем считать заданными технологические последовательности обслуживания без прерываний всех заявок с точностью до группы оборудования (блока, подсистемы). Технологические маршруты фиксированы, но различны для различных заявок. Известным (детерминированным) также считается время обслуживания каждой заявки каждым прибором из каждого блока и расписание на входе системы.

Задача в данном случае заключается в том, чтобы определить расписание работы системы (всех блоков и приборов), оптимальное по какому-либо критерию качества.

К настоящему времени общая задача синтеза оптимальных расписаний параллельно-последовательных ОС весьма далека от приемлемого решения при сколько-нибудь реальной размерности реализаций. Один из возможных подходов к решению задачи синтеза оптимальных расписаний параллельно-последовательных систем заключается в декомпозиции общей задачи и выделении локальных подзадач по числу блоков параллельных приборов и координирующей задачи [4]. Каждая локальная подзадача на некотором этапе определяет расписание параллельной подсистемы соответствующего блока.



Параллельно-последовательные обслуживающие системы

Parallel-sequential service systems

Можно показать, что если в каждый момент времени известны расписания всех блоков параллельных подсистем, то последовательными приближениями (решая по этапам координирующую задачу) можно определить оптимальное расписание всей параллельно-последовательной системы. Поэтому синтез оптимальных расписаний параллельных систем, по сути, является ключевой подзадачей общей задачи теории расписаний для параллельно-последовательных систем.

## 1. ПОСТАНОВКИ ЗАДАЧИ ОПТИМИЗАЦИИ РАСПИСАНИЙ ПАРАЛЛЕЛЬНОЙ СИСТЕМЫ С ЗАДЕРЖКАМИ НАЧАЛА ОБСЛУЖИВАНИЯ ЗАЯВОК

Пусть имеется ряд заявок, которые необходимо распределить между параллельными приборами при известной (различной) производительности таким образом, чтобы минимизировать суммарное время обслуживания всех заявок (минимизировать время окончания работы всей ОС, минимизировать суммарные затраты либо максимизировать некоторый показатель качества обслуживания заявок). Прерывания обслуживания запрещены. Пусть также известно расписание поступления заявок в параллельную обслуживающую систему. В этом случае необходимо учитывать величины задержек поступления заявок. Обозначим задержку поступления  $j$ -й заявки в ОС через  $\tau_j^0$  и упорядочим заявки по возрастанию  $\tau_j^0$  (тогда  $T^0 = \|\tau_j^0\|$  – расписание на входе системы). Здесь и далее  $\|\cdot\|$  обозначает вектор, матрицу или тензор, соответствующей контексту размерности. Через  $x_{i,j}$  обозначены булевы переменные назначения заявки  $j$  на прибор  $i$ , через  $t_{i,j}$  – время обслуживания заявки  $j$  прибором  $i$  ( $T = \|t_{i,j}\|$ ), а также пусть  $T^1 = \|\tau_j^1\|$  – расписание на выходе системы.

Один из вариантов общей математической модели, позволяющей синтезировать расписания параллельной системы с задержками на входе, представляется следующим образом [5]:

$$\sum_{i=1}^I x_{i,j} = 1, \quad j = \overline{1, J}; \quad (1)$$

$$\underline{b}_i \leq \sum_{j=1}^J x_{i,j} \leq \overline{b}_i, \quad i = \overline{1, I}; \quad (2)$$

$$x_{i,j} = \begin{cases} 1, & \text{если заявка } j \text{ назначается на прибор } i, \\ 0 & \text{в противном случае;} \end{cases} \quad (3)$$

$$y_{i,j} \geq 0, \quad i = \overline{1, I}, \quad j = \overline{1, J}; \quad (4)$$

$$\tau_{i,j} = \tau_j^0 - \sum_{l=1}^{j-1} (\tau_{i,l} + t_{i,l})x_{i,l}, \quad i = \overline{1, I}, \quad j = \overline{1, J}; \quad (5)$$

$$\hat{\tau}_{i,j} = \tau_{i,j} + y_{i,j} \geq 0, \quad i = \overline{1, I}, \quad j = \overline{1, J}; \quad (6)$$

$$\sum_{j=1}^J \hat{\tau}_{i,j}x_{i,j} + \sum_{j=1}^J t_{i,j}x_{i,j} \leq \lambda, \quad i = \overline{1, I}, \quad (7)$$

где  $x_{i,j}$  – переменные-назначения, подлежащие определению;  $\hat{\tau}_{i,j}$  имеют смысл фактических задержек начала выполнения  $j$ -й заявки  $i$ -м прибором после завершения обслуживания им предшествующей заявки. Компенсирующие переменные  $y_{i,j}$  вводятся для того, чтобы избежать появления отрицательных задержек  $\tau_{i,j}$ . Действительно, наличие неравенств (4) и (6) гарантирует выполнение условий:

$$\hat{\tau}_{i,j} = \begin{cases} \tau_{i,j}, & \text{если } \tau_{i,j} \geq 0, \\ 0 & \text{в противном случае.} \end{cases}$$

Условия (1)–(3) характерны для задачи о назначениях, (1) – обеспечивают назначения любой заявки только на один прибор; (2) – назначения не менее  $\underline{b}_i$  и не более  $\overline{b}_i$  заявок на любой прибор  $i$ ; (5) и (6) вычисляют фактические задержки обслуживания  $j$ -й заявки  $i$ -м прибором после окончания обслуживания им предшествующей заявки.

Если для модели (1)–(7) задан критерий качества (или эффективности) расписания, например

$$\lambda \rightarrow \min, \quad (8)$$

то (1)–(8) определяют задачу синтеза оптимальных по быстродействию расписаний параллельной системы с задержками на входе, (7) и (8) реализуют минимаксный критерий, имеющий смысл критерия максимального быстродействия (иногда называемого также критерием равномерной загрузки) вида

$$\max_i \left\{ \sum_{j=1}^J (t_{i,j} + \hat{\tau}_{i,j})x_{i,j} \right\} \rightarrow \min.$$

Расписание, синтезируемое посредством решения задачи (1)–(8), полностью определяется оптимальными назначениями  $x_{i,j}^*$ ,  $i = \overline{1, I}$ ,  $j = \overline{1, J}$  и фактическими задержками при таких назначениях  $\hat{\tau}_{i,j}^*$ ,  $i = \overline{1, I}$ ,  $j = \overline{1, J}$ .

$$\text{Расписание на выходе системы: } \tau_j^1 = \sum_{i=1}^I (\hat{\tau}_{i,j} + t_{i,j})x_{i,j}, \quad j = \overline{1, J}.$$

$$\text{То же при оптимальном решении: } \tau_j^{1*} = \sum_{i=1}^I (\hat{\tau}_{i,j}^* + t_{i,j})x_{i,j}^*, \quad j = \overline{1, J}.$$

Ориентируясь на системы данного вида, отметим основные трудности, возникающие при решении задачи (1)–(8). Прежде всего это наличие булевых переменных, рекурсивных функций и ранцевых ограничений, определяющих принадлежность представленной задачи к классу NP-полных.

## 2. ПОДХОДЫ К ОПТИМИЗАЦИИ РАСПИСАНИЙ ПАРАЛЛЕЛЬНЫХ СИСТЕМ С ЗАДЕРЖКАМИ НАЧАЛА ОБСЛУЖИВАНИЯ

Один из подходов к решению задачи (1)–(8) заключается в приведении к ее нормальному виду задачи математического программирования с последующим прямым решением редуцированной задачи. Такая редукция обусловлена тем, что выражения (5)–(8) содержат рекурсивные функции фактических задержек  $\tau_{i,j}$ . В явном виде  $\tau_{i,j}$ , как рекурсивные функции относительно переменных  $x_{i,j}$ , определяет (5). Раскрытие рекурсий и дальнейшая линейаризация задачи (1)–(8), приводит к задаче смешанного линейного программирования следующего вида [5]:

$$\sum_{i=1}^I x_{i,j} = 1, \quad j = \overline{1, J}; \quad (9)$$

$$\underline{b}_i \leq \sum_{j=1}^J x_{i,j} \leq \overline{b}_i, \quad i = \overline{1, I}, \quad (10)$$

$$x_{i,j} = \begin{cases} 1, & \text{если заявка } j \text{ закрепляется за прибором } i, \\ 0 & \text{в противном случае;} \end{cases} \quad i = \overline{1, I}, \quad j = \overline{1, J} \quad (11)$$

$$u_{i,j,k} = \begin{cases} 1 & \text{при истинности } x_{i,j} x_{i,k} \bigcap_{l=k+1}^{j-1} \underline{x}_{i,l}, \\ 0 & \text{в противном случае,} \end{cases} \quad i = \overline{1, I}, \quad j = \overline{1, J}, \quad k = \overline{1, J} \quad (12)$$

$$-K + 2 \leq x_{i,j} + x_{i,k} - \sum_{l=k+1}^{j-1} x_{i,l} - K u_{i,j,k} \leq 1, \quad (13)$$

$$i = \overline{1, I}, \quad j = \overline{1, J}, \quad k = \overline{1, J}, \quad K = j - k + 1,$$

$$\widehat{\tau}_{i,j} = y_{i,j} + \tau_j^0 - \sum_{k=1}^{j-1} (\tau_k^0 + t_{i,k}) u_{i,j,k} \geq 0,$$

или

$$-y_{i,j} + \sum_{k=1}^{j-1} (\tau_k^0 + t_{i,k}) u_{i,j,k} \leq \tau_j^0, \quad i = \overline{1, I}, \quad j = \overline{1, J}, \quad (14)$$

$$\sum_{j=1}^J (\tau_j^0 + t_{i,j}) x_{i,j} + \sum_{j=1}^J y_{i,j} - \sum_{j=1}^J \sum_{k=1}^{j-1} (\tau_k^0 + t_{i,k}) u_{i,j,k} \leq \lambda, \quad i = \overline{1, I}, \quad (15)$$

$$y_{i,j} \geq 0, \quad i = \overline{1, I}, \quad j = \overline{1, J}, \quad (16)$$

$$\lambda \rightarrow \min, \quad (17)$$

где  $K = j - k + 1$ .

Несмотря на формальную разрешимость редуцированной относительно (1)–(8) задачи (9)–(17), данный подход можно считать лишь начальным этапом общего решения. Очевидно, что число булевых переменных в редуцированной задаче, как и число ограничений, увеличивается примерно в  $J/2$  раза по сравнению с (1)–(8). Тем самым формальная сложность (1)–(8) редуцируется в вычислительную сложность (9)–(17).

Другой возможный подход заключается в упрощении условий исходной задачи (1)–(8), а именно: вместо условий, содержащих рекурсивные функции (4)–(7), вводятся два минимаксных критерия, реализуемых следующим образом:

$$\sum_{j=1}^J \tau_j^0 x_{i,j} \leq \beta, \quad i = \overline{1, I}, \quad (18)$$

$$\sum_{j=1}^J t_{i,j} x_{i,j} \leq \lambda, \quad i = \overline{1, I}, \quad (19)$$

$$\lambda \rightarrow \min, \quad (20)$$

$$\beta \rightarrow \min. \quad (21)$$

Из контекста понятно, что второй подход к распределению заявок между приборами предполагает компромиссное решение по «чистому» быстродействию ОС без учета задержек и по равномерности распределения задержек между приборами. Далее определяется Парето-оптимальное решение (1)–(3) и (18)–(21), все компоненты которого, как показано в [2], допустимы для (1)–(8) и (9)–(17)) и из которого выделяется расписание, наилучшее для (1)–(8). Благодаря этому удастся исключить из рассмотрения большую часть вариантов и обеспечить приемлемые трудоемкость и точность для ряда практических реализаций задачи [4].

Численные эксперименты с обеими моделями подтверждают эффективность второго подхода, так как при использовании (1)–(3) и (18)–(21) для большинства примеров наблюдается падение быстродействия синтезируемых расписаний не более чем на 5 % от оптимального уровня, что для многих реальных объектов вполне приемлемо.

Развитие данного подхода привело к использованию декомпозиции задач (9)–(17), (1)–(3) и (18)–(21) и применению общей схемы динамического программирования (ДП) с отсевом части вариантов [4]. Это позволяет гаран-

тировано улучшить оценки точности решения, что, однако, связано с потерей скорости поиска решения.

Наконец, применение метода ДП непосредственно к исходной постановке (1)–(8) позволяет создать еще один алгоритм, обеспечивающий компромисс между трудоемкостью вычислений и точностью решения. Рассмотрим этот подход более подробно.

### 3. ДИНАМИЧЕСКОЕ ПРОГРАММИРОВАНИЕ С ОТСЕВОМ ВАРИАНТОВ В ОПТИМИЗАЦИИ РАСПИСАНИЙ ПАРАЛЛЕЛЬНОЙ СИСТЕМЫ С ЗАДЕРЖКАМИ ПОСТУПЛЕНИЯ ЗАЯВОК

Из-за наличия рекурсий ДП едва ли не единственный вычислительный метод, применимый непосредственно для решения задачи (1)–(8). Однако его прямое применение неэффективно из-за NP – полноты задачи. При попытке точного решения (1)–(8) ДП приводит к полному перебору всех допустимых вариантов. Нетрудно подсчитать число таких вариантов  $N$ . Например, если  $s$  – номер этапа и в (2) положить  $\underline{b}_i = 0$  и  $\bar{b}_i = J$ , то  $N = (J^{I+1} - J)/2$ . По этой причине метод ДП в задаче (1)–(8) имеет трудоемкость, превышающую экспоненциальную, и в «чистом» виде на реальных размерностях неприменим.

Для построения эффективного приближенного алгоритма воспользуемся общей схемой ДП, производя отсев локально наилучших вариантов на ряде шагов (этапов) динамического программирования.

Как и выше, будем считать упорядоченными все заявки  $j = \overline{1, J}$  по величинам исходных задержек (входному расписанию  $\|\tau_j^0\|$ ). Тогда в соответствии с процедурой ДП определим номера этапов  $s = \overline{1, J}$ . Обозначим через  $f_{i,s}(\tau_s^0, t_{i,s}, x_{i,s})$ ,  $i = \overline{1, I}$ , время завершения обслуживания прибором  $i$  заявки  $s$  на этапе  $s$ , а через  $\varphi_s(\tau_j^0, t_{i,j}, x_{i,j})$ ,  $i = \overline{1, I}$ ,  $j = \overline{1, s}$ , – условно минимальное время завершения обслуживания всех принятых заявок на этапах с первого по  $s$ -й.

$$f_{i,s}(\tau_s^0, t_{i,s}, x_{i,s}) = \max \left\{ 0, \left[ \tau_s^0 x_{i,s} - \varphi_{i,s-1}(\tau_{s-1}^0, t_{i,j}, x_{i,j}) \right] \right\} + t_{i,s} x_{i,s}, \quad (22)$$

$$i = \overline{1, I}, \quad j = \overline{1, s-1}.$$

Рекуррентное соотношение Беллмана для этой задачи:

$$\varphi_{i,s}(\tau_j^0, t_{i,j}, x_{i,j}) = \left\{ f_{i,s}(\tau_s^0, t_{i,s}, x_{i,s}) + \varphi_{i,s-1}(\tau_j^0, t_{i,j}, x_{i,j}) \right\}, \quad j = \overline{1, s-1}, \quad (23)$$

$$\varphi_s(\tau_j^0, t_{i,j}, x_{i,j}) = \max_i \left\{ \varphi_{i,s}(\tau_s^0, t_{i,j}, x_{i,j}) \right\}, \quad j = \overline{1, s}, \quad i = \overline{1, I}. \quad (24)$$

Для достижения максимального быстродействия по критерию (7)–(8) на последнем шаге следует выбрать минимальное значение  $\varphi_{\bar{I}}(\bar{\tau}_{\bar{I},j}, t_{i,j}, x_{i,j})$ , т. е. найти  $\lambda = \min\{\varphi_{\bar{I}}(\bar{\tau}_{\bar{I},j}, t_{i,j}, x_{i,j})\}$ .

Отсев части промежуточных расписаний на этапах ДП возможен в различных вариациях. Если отбрасывать все варианты расписания на шаге  $s$ , кроме локально наилучшего, то такой способ построения расписания совпадет с жадным алгоритмом. Если оставлять все промежуточные расписания, это приведет к полному перебору вариантов. При этом на первом этапе будем иметь  $J$ , на втором этапе –  $J^2$ , на этапе  $k$  –  $J^k$  вариантов промежуточного расписания. Если же пытаться найти компромисс между точностью и быстродействием, то с учетом стремления построить эффективный алгоритм число промежуточных расписаний должно полиномиально зависеть от количества булевых переменных задачи (1)–(8).

Рассмотрим одну из таких компромиссных вариаций. Определим максимально возможное число вариантов  $K$ , оставляемое на этапе  $k$  для дальнейшего анализа. Для удобства изложения примем в качестве  $K$  некоторую константу. Например, положим  $K = 1024$  и определим максимальное число  $K' = J^k \leq K$ . Поскольку на каждом этапе ДП число вариантов промежуточных расписаний увеличивается в  $J$  раз, проведем отсев  $J^{k-1}$  доли локально наихудших из генерируемых вариантов на всяком этапе, начиная с  $k+1$ -го. Вполне очевидно значение  $k: k = \left\lfloor \frac{\ln(K)}{\ln(J)} \right\rfloor$ , где  $\lfloor \cdot \rfloor$  – целая часть числа. Тогда общее число промежуточных расписаний  $N'$  составит

$$N' = J + J^2 + \dots + J^{k-1} + J^k + \dots + J^k = (J^{k-1} - J)/2 + (I - k + 1)J^k. \quad (25)$$

Поскольку  $k$  – некоторая задаваемая константа (параметр алгоритма), то (25) отображает полиномиальную зависимость трудоемкости ДП с отсевом вариантов от размерности задачи (1)–(8). Для наглядности сравним  $N$  с  $N'$ , положив значение параметра  $k = 3$ , а также  $I = 1000$ ,  $J = 100$ . Тогда  $N = (100^{1001} - 100)/2$ ,  $N' = (100^2 - 100)/2 + (1000 - 2)100^3 = 998\,004\,950$ . Данные обстоятельства опосредуют вполне заурядную трудоемкость параметрического алгоритма, которая кратна  $N'$ , и фактически бесконечную трудоемкость ДП, кратную  $N$ . Опишем параметрический алгоритм ДП с отсевом вариантов.

#### АЛГОРИТМ $A_p$

1. Ввод исходных данных  $(\tau_i^0, t_{i,j})$ ,  $j = \overline{1, J}$ ,  $i = \overline{1, I}$ , и параметров  $k$ ,  $K'$ .

Положим  $\varphi_{i,0}(\tau_i^0, t_{i,j}, x_{i,j}) \equiv 0$ , определим начальный номер шага  $\eta := 0$ .

2.  $\eta := \eta + 1$ .



3. Проверка номера этапа. Если  $\eta > I$ , то переход к этапу 7, если нет – следующий пункт.

4. На этапе  $\eta$  в соответствии с (22)–(24) генерируем все допустимые варианты расписания, вычисляем  $f_{\eta,j}(\bar{\tau}_{\eta,j}, t_{\eta,j}, x_{\eta,j})$  и длины расписаний  $\Phi_{\eta,j}(\bar{\tau}_{\eta,j}, t_{i,j}, x_{i,j})$ .

5. Проверка  $N^\eta$  – числа вариантов  $\Phi_{\eta,j}(\bar{\tau}_{\eta,j}, t_{i,j}, x_{i,j})$  на этапе  $\eta$ . Если  $\eta < k$ , т. е.  $N^\eta \leq N'$ , то переход к п. 2. В противном случае следующий пункт.

6. Отсев  $J^{k-1}$  из всех порожденных в п. 4 вариантов с наибольшими значениями длины расписания  $\Phi_{\eta,j}(\bar{\tau}_{\eta,j}, t_{i,j}, x_{i,j})$ . Переход к п. 2.

7. Выбор вариантов кратчайших расписаний. Составление расписаний обратным ходом ДП.

Проиллюстрируем работу алгоритма  $A_P$  небольшим числовым примером синтеза расписаний параллельной системы.

#### 4. ЧИСЛОВОЙ ПРИМЕР СИНТЕЗА РАСПИСАНИЙ ПАРАЛЛЕЛЬНОЙ СИСТЕМЫ

В параллельную обслуживающую систему, состоящую из двух неидентичных приборов, одновременно поступает семь заявок. Данные о нормах времени обслуживания и задержках поступления заявок приведены в табл. 1.

Таблица 1

Table 1

##### Исходные данные

##### Initial data

Номер заявки ( $j$ )	Время обслуживания ( $t_{i,j}$ )		Задержка поступления заявки ( $\tau_j^0$ )
	(номер прибора ( $i$ ))		
	1	2	
1	2	4	0
2	3	2	0
3	5	4	2
4	2	4	3
5	4	2	4
6	3	3	5
7	4	3	6

Применение алгоритма  $A_P$  динамического программирования с отсевом вариантов для решения данного примера отображено ниже в последовательности табл. 2–8, каждая из которых соответствует этапу ДП. Положим  $k = 3$ ,  $K' = 8$ .

Таблица 2

Table 2

## Результаты первого этапа

## Results of the first stage

Этап 1	$x_{1,1}$	$x_{2,1}$	$f_{i,1}$	$\phi_{i,1} = f_{i,1}, \phi_1 = \max_i \{\phi_{i,1}\}$
Заявка 1	1	0	(2, 0)	$\phi_1 = \max \{2, 0\} = 2$
	0	1	(0, 4)	$\phi_1 = \max \{0, 4\} = 4$

Таблица 3

Table 3

## Результаты второго этапа

## Results of the second stage

Этап 2	$x_{1,1}$	$x_{2,1}$	$x_{1,2}$	$x_{2,2}$	$f_{i,2}$	$\phi_{i,2} = \{f_{i,2} + \phi_{i,1}\}, \phi_2 = \max_i \{\phi_{i,2}\}$
Заявка 1 + + Заявка 2	1	0	1	0	(3, 0)	$\phi_2 = \max \{2+3; 0+0\}=5$
	1	0	0	1	(0, 2)	$\phi_2 = \max \{2+0; 0+2\}=2$
	0	1	1	0	(3, 0)	$\phi_2 = \max \{0+3; 4+0\}=4$
	0	1	0	1	(0, 2)	$\phi_2 = \max \{0+0; 4+2\}=6$

Таблица 4

Table 4

## Результаты третьего этапа и отсева вариантов

## Results of the third stage and sifting of options

Этап 3	$x_{1,1}$	$x_{2,1}$	$x_{1,2}$	$x_{2,2}$	$x_{1,3}$	$x_{2,3}$	$f_{i,3}$	$\phi_{i,3} = \{f_{i,3} + \phi_{i,2}\},$ $\phi_3 = \max_i \{\phi_{i,3}\}$
Заявка 1 + + Заявка 2 + + Заявка 3	1	0	1	0	1	0	(5, 0)	$\phi_3 = \max \{2+3+5; 0+0+0\}=10$
	1	0	0	1	0	1	(0, 4)	$\phi_3 = \max \{2+0+0; 0+2+4\}=6$
	1	0	1	0	0	1	(0,4+2)	$\phi_3 = \max \{3+2+0; 0+0+4+2\}=6$
	1	0	0	1	1	0	(5, 0)	$\phi_3 = \max \{2+0+5; 0+2+0\}=7$
	0	1	1	0	1	0	(5, 0)	$\phi_3 = \max \{0+3+5; 4+0+0\}=8$
	0	1	0	1	0	1	(0, 4)	$\phi_3 = \max \{0+0+0; 2+4+4\}=10$
	0	1	1	0	0	1	(0, 4)	$\phi_3 = \max \{0+3+0; 4+0+4\}=8$
	0	1	0	1	1	0	(5+2,0)	$\phi_3 = \max \{0+0+5+2; 2+4+0\}=7$

Заливкой в табл. 4 выделены строки (и варианты назначений), имеющие локально наилучшие промежуточные оценки длины расписания для третьего этапа. Все прочие строки (и варианты назначений) из дальнейшего рассмотр-

рения исключены. Тем самым вдвое сокращено число допустимых вариантов на данном этапе и всех последующих этапах.

Заливкой в табл. 5 и 6 выделены строки (и варианты назначений), имеющие локально наилучшие промежуточные оценки длины расписания для соответствующих этапов.

Таблица 5

Table 5

**Результаты четвертого этапа и отсев вариантов**

**Results of the fourth stage and sifting of options**

Этап 4	$x_{1,2}$	$x_{2,2}$	$x_{1,3}$	$x_{2,3}$	$x_{1,4}$	$x_{2,4}$	$f_{i,4}$	$\phi_{i,4} = \{f_{i,4} + \phi_{i,3}\},$ $\phi_4 = \max_i \{\phi_{i,4}\}$
Заявка 1 + + Заявка 2 + + Заявка 3 + + Заявка 4	0	1	0	1	0	1	(0,4)	$\max\{2+0+0+0;$ $0+2+4+4\}=10$
	1	0	0	1	0	1	(0,4)	$\max\{2+3+0+0;$ $0+0+6+2+4\}=12$
	0	1	1	0	0	1	(0,1+4)	$\max\{2+0+5+0;$ $0+2+0+5\}=7$
	0	1	1	0	0	1	(0,4)	$\max\{0+0+7+0;$ $2+4+0+4\}=10$
	0	1	0	1	1	0	(1+2,0)	$\max\{2+0+0+3;$ $0+2+4+0\}=6$
	1	0	0	1	1	0	(2,0)	$\max\{2+3+0+2;$ $0+0+6+0\}=7$
	0	1	1	0	1	0	(2,0)	$\max\{2+0+5+2;$ $0+2+0+0\}=9$
	0	1	1	0	1	0	(2,0)	$\max\{0+0+5+2+2;$ $2+4+0+0\}=9$

Таблица 6

Table 6

**Результаты пятого этапа и отсев вариантов**

**Results of the fifth stage and sifting of options**

Этап 5	$x_{1,3}$	$x_{2,3}$	$x_{1,4}$	$x_{2,4}$	$x_{1,5}$	$x_{2,5}$	$f_{i,5}$	$\phi_{i,5} = \{f_{i,5} + \phi_{i,4}\},$ $\phi_5 = \max_i \{\phi_{i,5}\}$
Заявка 1 + + Заявка 2 + + Заявка 3 + + Заявка 4 + + Заявка 5	1	0	0	1	0	1	(0, 2)	$\max\{2+0+5+0+0;$ $0+2+0+5+2\}=9$
	0	1	1	0	0	1	(0, 2)	$\max\{2+0+0+3+0;$ $0+2+4+0+2\}=8$
	0	1	1	0	0	1	(0, 2)	$\max\{2+3+0+2+0;$ $0+0+6+0+2\}=8$
	1	0	1	0	0	1	(0,2+2)	$\max\{2+0+5+2+0;$ $0+2+0+0+4\}=9$
	1	0	0	1	1	0	(4, 0)	$\max\{2+0+5+0+4;$ $0+2+0+5+0\}=11$

Окончание табл. 6

End tab. 6

Этап 5	$x_{1,3}$	$x_{2,3}$	$x_{1,4}$	$x_{2,4}$	$x_{1,5}$	$x_{2,5}$	$f_{i,5}$	$\phi_{i,5} = \{f_{i,5} + \phi_{i,4}\},$ $\phi_5 = \max_i \{\phi_{i,5}\}$
	0	1	1	0	1	0	(4, 0)	$\max\{2+0+0+3+4;$ $0+2+4+0+0\}=9$
	0	1	1	0	1	0	(4, 0)	$\max\{2+3+0+2+4;$ $0+0+6+0+0\}=11$
	1	0	1	0	1	0	(4, 0)	$\max\{2+0+5+2+4;$ $0+2+0+0+0\}=13$

Все прочие варианты из дальнейшего рассмотрения исключены. Тем самым вдвое сокращено число допустимых вариантов на данных и всех последующих этапах.

Темной заливкой в табл. 7 выделены строки (и варианты назначений), имеющие локально наилучшие промежуточные оценки длины расписания для этапа 6.

Таблица 7

Table 7

### Результаты шестого этапа и отсева вариантов

#### Results of the sixth stage and sifting of options

Этап 6	$x_{1,4}$	$x_{2,4}$	$x_{1,5}$	$x_{2,5}$	$x_{1,6}$	$x_{2,6}$	$f_{i,6}$	$\phi_{i,6} = \{f_{i,6} + \phi_{i,5}\},$ $\phi_6 = \max_i \{\phi_{i,6}\}$
	0	1	0	1	0	1	(0, 3)	$\max\{2+0+5+0+0+0;$ $0+2+0+5+2+3\}=12$
	1	0	0	1	0	1	(0, 3)	$\max\{2+0+0+3+0+0;$ $0+2+4+0+2+3\}=11$
	1	0	0	1	0	1	(0, 3)	$\max\{2+3+0+2+0+0;$ $0+0+6+0+2+3\}=11^*$
	1	0	1	0	0	1	(0, 3)	$\max\{2+0+0+3+4+0;$ $0+2+4+0+0+3\}=9$
	0	1	0	1	1	0	(3, 0)	$\max\{2+0+5+0+0+3;$ $0+2+0+5+2+0\}=10$
	1	0	0	1	1	0	(3, 0)	$\max\{2+0+0+3+0+3;$ $0+2+4+0+2+0\}=8$
	1	0	0	1	1	0	(3, 0)	$\max\{2+3+0+2+0+3;$ $0+0+6+0+2+0\}=10$
	1	0	1	0	1	0	(3, 0)	$\max\{2+0+0+3+4+3;$ $0+2+4+0+0+0\}=12$

Вдвое сокращено число допустимых вариантов на этапе 6.

Таблица 8

Table 8

## Результаты и выбор наилучших вариантов

## Results and selection of the best options

Этап 7	$x_{1,5}$	$x_{2,5}$	$x_{1,6}$	$x_{2,6}$	$x_{1,7}$	$x_{2,7}$	$f_{i,7}$	$\phi_{i,7} = \{f_{i,7} + \phi_{i,6}\},$ $\phi_7 = \max_i \{\phi_{i,7}\}$
Заявка 1 + + Заявка 2 + + Заявка 3 + + Заявка 4 + + Заявка 5 + + Заявка 6 + + Заявка 7	1	0	0	1	0	1	(0, 3)	$\max \{2+0+0+3+4+0+0;$ $0+2+4+0+0+3+3\}=12$
	0	1	1	0	0	1	(0, 3)	$\max \{2+0+5+0+0+3+0;$ $0+2+0+5+2+0+3\}=12$
	0	1	1	0	0	1	(0, 3)	$\max \{2+0+0+3+0+3+0;$ $0+2+4+0+2+0+3\}=11$
	0	1	1	0	0	1	(0, 3)	$\max \{2+3+0+2+0+3+0;$ $0+0+6+0+2+0+0+3\}=11$
	0	1	0	1	1	0	(4, 0)	$\max \{2+3+0+2+0+0+4;$ $0+0+6+0+2+3+0\}=11 *$
	1	0	0	1	1	0	(4, 0)	$\max \{2+0+0+3+4+0+4;$ $0+2+4+0+0+3+0\}=13$
	0	1	1	0	1	0	(4, 0)	$\max \{2+0+5+0+0+3+4;$ $0+2+0+5+2+0+0\}=14$
	0	1	1	0	1	0	(4, 0)	$\max \{2+0+0+3+0+3+4;$ $0+2+4+0+2+0+0\}=12$
0	1	1	0	1	0	(4, 0)	$\max \{2+3+0+2+0+3+4;$ $0+0+6+0+2+0+0\}=14$	

Светлой заливкой выделен вариант, исключенный алгоритмом  $A_P$  из рассмотрения, но тем не менее дающий впоследствии одно из оптимальных решений. Можно показать, что все полученные варианты расписания оптимальны по быстродействию. Это подтверждается прямым решением задачи (9)–(17) [4].

Темной заливкой на последнем этапе выделены два варианта, оптимальных по быстродействию расписаний, светлой заливкой – вариант оптимального расписания, исключенный из рассмотрения процедурой отсева части вариантов на предшествующем шаге. В табл. 9 и 10 детализированы полученные оптимальные расписания процедурой обратного хода ДП. Таблицы 11 и 12 содержат исключенные алгоритмом  $A_P$  из рассмотрения варианты оптимального решения [4].

Таблица 9

Table 9

## Наилучшее расписание 1

## The best schedule 1

Номер заявки (j)	Назначения		Расписание на выходе системы
	номер прибора (i)		
	1	2	
1	1	0	2
2	0	1	2
3	0	1	6
4	1	0	5
5	0	1	8
6	1	0	8
7	0	1	11

Таблица 10

Таблица 10

## Наилучшее расписание 2

## The best schedule 1

Номер заявки (j)	Назначения		Расписание на выходе системы
	номер прибора (i)		
	1	2	
1	1	0	2
2	1	0	5
3	0	1	6
4	1	0	7
5	0	1	8
6	1	0	10
7	0	1	11

Оценки трудоемкости алгоритма  $A_P$  и ДП для рассмотренного примера:

$$N = (2^8 - 2) / 2 = 127, \quad N' = (2^2 - 2) / 2 + (7 - 3 + 1)2^3 = 41.$$

Таблица 11

Table 11

Таблица 12

Таблица 12

## Наилучшее расписание 3

## The best schedule 3

Номер заявки (j)	Назначения		Расписание на выходе системы
	номер прибора (i)		
	1	2	
1	1	0	2
2	0	1	2
3	0	1	6
4	1	0	5
5	0	1	8
6	0	1	11
7	1	0	10

## Наилучшее расписание 4

## The best schedule 4

Номер заявки (j)	Назначения*		Расписание на выходе системы
	номер прибора (i)		
	1	2	
1	1	0	2
2	1	0	5
3	0	1	6
4	1	0	7
5	0	1	8
6	0	1	11
7	1	0	11

## 5. УЛУЧШАЮЩИЙ ОБМЕННЫЙ АЛГОРИТМ

Для улучшения решения, полученного алгоритмом  $A_P$ , можно применить процедуру, основанную на обмене заявками между приборами. Опишем этот алгоритм.

На первом этапе выбирается прибор с наибольшим суммарным временем обслуживания заявок и производится попытка передачи одной из заявок другому прибору. Если при этом происходит уменьшение значения целевой функции, процесс повторяется, в противном случае рассматривается следующий прибор.

На втором этапе выбирается прибор с наибольшим суммарным временем обслуживания заявок. Последовательно рассматриваются заявки данного прибора. В каждом случае осуществляется поиск заявки, обслуживаемой этим прибором за меньшее время, но назначенной другому прибору. Производится обмен заявками между приборами. Если при этом происходит уменьшение значения целевой функции, второй этап повторяется, в противном случае изменения отменяются и осуществляется поиск другой подходящей для обмена заявки. Общий обменный алгоритм, как и алгоритм, реализуемый на втором этапе, обозначим как  $A_C$ . Более подробно  $A_C$  на втором этапе можно описать следующим образом.

#### АЛГОРИТМ $A_C$

1. Выбирается прибор  $m$  с наибольшим суммарным временем обслуживания заявок  $\lambda_m = \lambda$ .
2. Положим  $i := 1$ .
3. Если  $i := m$ , то перейти к п. 14.
4. Положим  $l := 1$ .
5. Если  $x_{m,l} = 0$ , то перейти к п. 12.
6. Положим  $j := 1$ .
7. Если  $j = l$ , или  $x_{i,j} = 0$ , или  $t_{m,j} \geq t_{m,l}$ , или  $t_{i,l} - t_{i,j} \geq \lambda_m - \lambda_i$ , то перейти к п. 10.
8. Положим  $x_{i,j} = 0$ ,  $x_{m,j} = 1$ ,  $x_{i,l} = 1$ ,  $x_{m,l} = 0$ .
9. Вычислим значение целевой функции. Если оно уменьшилось, то перейти к п. 1, иначе положить  $x_{i,j} = 1$ ,  $x_{m,j} = 0$ ,  $x_{i,l} = 0$ ,  $x_{m,l} = 1$ .
10. Положим  $j := j + 1$ .
11. Если  $j \leq J$ , то перейти к п. 7.
12. Положим  $l := l + 1$ .
13. Если  $l \leq J$ , то перейти к п. 5.
14. Положим  $i := i + 1$ .
15. Если  $i \leq I$ , то перейти к п. 3, иначе завершить алгоритм.

#### 6. НЕКОТОРЫЕ РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ ПАРАМЕТРИЧЕСКОГО АЛГОРИТМА

Исходные данные тестовых примеров для задач (9)–(17), (1)–(3), (18)–(21) и (1)–(8), были сгенерированы датчиками случайных чисел. Размерность всех тестов одинакова (5 приборов, 100 заявок).

Таблица 13 содержит результаты тестирования реализаций с использованием средств решения бикритериальной релаксации задачи (1)–(8) и представленных выше алгоритмов  $A_P$  и  $A_C$ .

В табл. 13  $t_{da}$ ,  $\lambda_{da}$  – соответственно время решения (в формате часы : минуты : секунды) и значение критерия эффективности в результате применения базового алгоритма, основанного на бикритериальной релаксации с использованием инструментов IBM ILOG CPLEX;  $t_{dp}$ ,  $\lambda_{dp}$ ,  $P_{dp}$  и  $\Delta_{dp}$  – время решения (в секундах), значение критерия, относительное и абсолютное ухудшение (улучшения при отрицательном значении) критерия, достигнутые алгоритмом  $A_P$  в сравнении с базовым алгоритмом;  $\lambda_{dp}^c$ ,  $P_{dp}^c$  и  $\Delta_{dp}^c$  – те же значения, полученные с применением алгоритма  $A_C$ .

В целом очевиден абсолютный выигрыш в быстродействии, объясняемый эффективностью алгоритма  $A_P$  и его комбинации с  $A_C$ , а также практически полное превосходство в близости получаемых решений к оптимальным в сравнении с базовым алгоритмом.

Таблица 13

Table 13

### Сравнительные характеристики алгоритмов

#### Comparative characteristics of algorithms

№ п/п	$t_{da}$	$\lambda_{da}$	$t_{dp}$	$\lambda_{dp}$	$P_{dp}$	$\Delta_{dp}$	$\lambda_{dp}^c$	$P_{dp}^c$	$\Delta_{dp}^c$	$t_{dp}$	$\lambda_{dp}$	$P_{dp}$	$\Delta_{dp}$	$\lambda_{dp}^c$	$P_{dp}^c$	$\Delta_{dp}^c$
	(ч:м:с)	$K^* = 10$					$A_C$			$K^* = 11$				$A_C$		
1	0:07:18	389	17	368	-5,40	-21	364	-6,43	-25	435	363	-6,68	-26	359	-7,71	-30
2	0:15:29	335	18	337	0,60	2	335	0,00	0	436	331	-1,19	-4	329	-1,79	-6
3	0:09:21	401	17	382	-4,74	-19	382	-4,74	-19	438	375	-6,48	-26	375	-6,48	-26
4	0:01:42	356	18	370	3,93	14	366	2,81	10	437	365	2,53	9	361	1,40	5
5	0:12:11	334	17	337	0,90	3	333	-0,30	-1	435	337	0,90	3	337	0,90	3
6	4:43:14	363	17	377	3,86	14	369	1,65	6	434	375	3,31	12	361	-0,55	-2
7	0:11:56	403	18	404	0,25	1	404	0,25	1	434	404	0,25	1	401	-0,50	-2
8	1:34:02	395	17	398	0,76	3	397	0,51	2	433	386	-2,28	-9	386	-2,28	-9
9	0:03:33	364	17	372	2,20	8	372	2,20	8	434	371	1,92	7	371	1,92	7
10	0:15:37	395	17	392	-0,76	-3	366	-7,34	-29	434	389	-1,52	-6	376	-4,81	-19
Среднее					0,16	0,2		-1,14	-4,7			-0,93	-3,9		-1,99	-7,9

### ЗАКЛЮЧЕНИЕ

Основным результатом представленной работы является эффективный параметрический алгоритм оптимизации расписаний параллельной системы с задержками поступления заявок, основанный на схеме динамического программирования и отсева локально наихудших вариантов. Практическая ценность результатов работы заключается не только в обеспечении возможности решения одной из подзадач общей задачи теории расписаний фактически лю-



бой размерности, но и в самостоятельной ценности разработанного инструментария для нужд практики. В частности, актуальны его применения в календарном планировании производственных процессов с дискретным характером. Придадим исходным данным приведенного выше примера следующий содержательный смысл. В качестве заявок рассмотрим места размещения кустов скважин нефтегазоконденсатного месторождения, в качестве приборов – кустовые буровые установки,  $t_{i,j}$  – время (в месяцах) строительства и бурения куста  $j$  буровой  $i$ ,  $\tau_j^0$  – определенное технологической последовательностью и регламентами время (в месяцах) задержки начала бурения куста  $j$ . Тогда полученные решения (табл. 9–12) являются оптимальными по быстродействию расписаниями строительства и бурения семи кустов скважин двумя буровыми установками.

Важным практическим результатом работы явилась возможность достижения компромисса между точностью и быстродействием параметрического алгоритма путем варьирования его входных параметров. Кроме этого, оценена трудоемкость синтеза с его помощью точных и приближенных к оптимальным по быстродействию расписаниям.

Перспективы развития темы связаны с оценками точности алгоритмов решения рассмотренных в работе задач, в применении метода бинарных отсечений [14, 15] и методов решения оценочных задач к общей задаче оптимизации расписаний параллельно-последовательных производственных систем.

## СПИСОК ЛИТЕРАТУРЫ

1. Brucker P. Scheduling algorithms. – 5<sup>th</sup> ed. – Berlin; New York: Springer, 2007. – 372 p.
2. Pinedo M. Scheduling theory, algorithms, and systems – 3<sup>rd</sup> ed. – New York; London: Springer, 2008. – 672 p.
3. Multiprocessor scheduling: theory and applications / ed. by E. Levner. – Vienna, Austria: I-Tech Education and Publishing, 2007. – 436 p.
4. Мезенцев Ю.А. Прикладные задачи и алгоритмы оптимизации расписаний параллельных обслуживающих систем // Научный вестник НГТУ. – 2016. – № 1 (62). – С. 49–73.
5. Мезенцев Ю.А. Оптимизация расписаний параллельных динамических систем в календарном планировании // Информационные технологии. – 2008. – № 2. – С. 16–23.
6. Santos F.C., Vilarinho P.M. The problem of scheduling in parallel machines: a case study // Proceedings of the World Congress on Engineering WCE 2010. – London, 2010. – Vol. 3. – P. 2443–2448.
7. Kaabi J., Harrath Y. A survey of parallel machine scheduling under availability constraints // International Journal of Computer and Information Technology. – 2014. – Vol. 3, iss. 2. – P. 238–245.
8. Lin Y.K., Hsieh H.T., Hsieh F.Y. Unrelated parallel machine scheduling problem using an ant colony optimization approach // World Academy of Science, Engineering and Technology. – 2012. – Vol. 6. – P. 1798–1803.
9. Lin Y.K. Particle swarm optimization algorithm for unrelated parallel machine scheduling with release dates // Mathematical Problems in Engineering. – 2013. – Vol. 2013. – Art. 409486. – P. 1–9.
10. Lee W.C., Wu C.C., Chen P. A simulated annealing approach to makespan minimization on identical parallel machines // International Journal of Advanced Manufacturing Technology. – 2006. – Vol. 31, N 3–4. – P. 328–334.

11. *Gharbi A., Haouari M.* Minimizing makespan on parallel machines subject to release dates and delivery times // *Journal of Scheduling*. – 2002. – Vol. 5, N 4. – P. 329–355.
12. *Vazirani V.* Approximation algorithms. – Berlin; New York: Springer, 2001. – 375 p.
13. *Kashan A.H., Karimi B.* A discrete particle swarm optimization algorithm for scheduling parallel machines // *Computers and Industrial Engineering*. – 2009. – Vol. 56, N 1. – P. 216–223.
14. *Мезенцев Ю.А.* Метод бинарных отсечений и ветвлений целочисленного программирования // Доклады Академии наук высшей школы РФ. – 2011. – № 1 (16). – С. 12–25.
15. *Mezentsev Yu.A.* Binary cut-and-branch method for solving mixed integer programming problems // 2017 Constructive Nonsmooth Analysis and Related Topics (dedicated to the memory of V.F. Demyanov) (CNSA): Proceedings. – St. Petersburg, 2017. – P. 210–212.

*Мезенцев Юрий Анатольевич*, доктор технических наук, профессор кафедры автоматизированных систем управления Новосибирского государственного технического университета. Основное направление научных исследований – задачи и методы дискретного программирования. Имеет более 65 публикаций, в том числе четыре монографии. E-mail: mezentsev@corp.nstu.ru

*Эстрайх Игорь Викторович*, старший преподаватель кафедры автоматизированных систем управления Новосибирского государственного технического университета. Основное направление научных исследований – задачи и методы дискретного программирования. Имеет более 10 публикаций, в том числе одну монографию. E-mail: estrajx@corp.nstu.ru

DOI: 10.17212/1814-1196-2018-3-87-106

### ***An efficient parametric algorithm for optimal scheduling of unrelated parallel machines with release dates\****

*Yu.A. MEZENTSEV<sup>a</sup>, I.V. ESTRAYKH<sup>b</sup>*

*Novosibirsk State Technical University, 20, K. Marx Prospekt, Novosibirsk, 630073, Russian Federation*

<sup>a</sup> *mesyan@yandex.ru, mezentsev@corp.nstu.ru*    <sup>b</sup> *estrax@corp.nstu.ru*

#### **Abstract**

In this paper we propose a parametric polynomial-time algorithm for optimizing the schedules of a parallel system with input delays which is supplemented by an improving exchange algorithm. The criterion of efficiency in the statement concerned is a maximum speed of the system (Cmax), which ensures uniform loading of devices. The availability of this criterion, as well as delays in starting the service of incoming requests, makes the problem difficult to solve with an a priori estimate of any approximation algorithm equal to 2. The proposed parametric algorithm is based on the scheme of dynamic programming modified with adaptive narrowing of search domain ensuring its computational effectiveness. The formulated problems of synthesis and the proposed algorithm for optimizing the schedules of parallel service systems are focused on practical application of the production processes with a discrete character in scheduling and operational control. We discuss the complexity of exact schedule synthesis and compare it with approximate close to optimal solutions. An illustrative example of using formal statements and the algorithm for synthesizing optimal schedules of the parallel system of the

---

\* Received 30 March 2018.

type involved is given. The computational experiments performed on specially generated realizations of the problem with large dimensions show good practical results. The final comparative statistics of accuracy and counting time is provided by software implementations of parametric and exchange algorithms with the results of alternative tools showing an undeniable advantage of the algorithms presented in the paper. The computational efficiency of the algorithm based on the procedure of dynamic programming with the elimination of a part of the variants at each step followed by the exchange algorithm is experimentally proved, which makes it possible to provide an acceptable speed and closeness to optimums. The a posteriori estimate of the closeness to optimums of the solutions obtained on the tests available for evaluation did not exceed 5 %.

**Keywords:** optimal scheduling, parallel production systems, unrelated parallel machines, release date, criterion of processing speed, discrete optimization, effective parametric algorithm, improving exchange algorithm

## REFERENCES

1. Brucker P. *Scheduling algorithms*. 5<sup>th</sup> ed. Berlin, New York, Springer, 2007. 372 p.
2. Pinedo M. *Scheduling theory, algorithms, and systems*. 3<sup>rd</sup> ed. New York, London, Springer, 2008. 672 p.
3. Levner E., ed. *Multiprocessor scheduling: theory and applications*. Vienna, Austria, I-Tech Education and Publishing, 2007. 436 p.
4. Mezentsev Yu.A. Prikladnye zadachi i algoritmy optimizatsii raspisaniy paralel'nykh obsluzhivayushchikh sistem [Applications and algorithms for optimal scheduling of unrelated parallel service systems]. *Nauchnyi vestnik Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Science bulletin of the Novosibirsk state technical university*, 2016, no. 1 (62), pp. 49–73.
5. Mezentsev Yu.A. Optimizatsiya raspisaniy paralel'nykh dinamicheskikh sistem v kalendar-nom planirovanii [Optimization of schedules for dynamic parallel service systems]. *Informatsionnye tekhnologii – Information Technology*, 2008, no. 2, pp. 16–23.
6. Santos F.C., Vilarinho P.M. The problem of scheduling in parallel machines: a case study. *Proceedings of the World Congress on Engineering WCE 2010*, London, 2010, vol. 3, pp. 2443–2448.
7. Kaabi J., Harrath Y. A survey of parallel machine scheduling under availability constraints. *International Journal of Computer and Information Technology*, 2014, vol. 3, iss. 2, pp. 238–245.
8. Lin Y.K., Hsieh H.T., Hsieh F.Y. Unrelated parallel machine scheduling problem using an ant colony optimization approach. *World Academy of Science, Engineering and Technology*, 2012, vol. 6, pp. 1798–1803.
9. Lin Y.K. Particle swarm optimization algorithm for unrelated parallel machine scheduling with release dates. *Mathematical Problems in Engineering*, 2013, vol. 2013, art. 409486, pp. 1–9.
10. Lee W.C., Wu C.C., Chen P. A simulated annealing approach to makespan minimization on identical parallel machines. *International Journal of Advanced Manufacturing Technology*, 2006, vol. 31, no. 3–4, pp. 328–334.
11. Gharbi A., Haouari M. Minimizing makespan on parallel machines subject to release dates and delivery times. *Journal of Scheduling*, 2002, vol. 5, no. 4, pp. 329–355.
12. Vazirani V. *Approximation algorithms*. Berlin, New York, Springer, 2001. 375 p.
13. Kashan A.H., Karimi B. A discrete particle swarm optimization algorithm for scheduling parallel machines. *Computers and Industrial Engineering*, 2009, vol. 56, no. 1, pp. 216–223.
14. Mezentsev Yu.A. Metod binarnykh otsechenii i vetvlenii tselochislennogo programmirovaniya [Branch and binary cuts approach of integer programming]. *Doklady Akademii nauk vysshei shkoly Rossiiskoi Federatsii – Proceedings of the Russian higher school Academy of sciences*, 2011, no. 1 (16), pp. 12–25.

15. Mezentsev Yu.A. Binary cut-and-branch method for solving mixed integer programming problems. *2017 Constructive Nonsmooth Analysis and Related Topics (dedicated to the memory of V. F. Demyanov) (CNSA): Proceedings*, St. Petersburg, 2017, pp. 210–212.

Для цитирования:

Мезенцев Ю.А., Эстрайх И.В. Эффективный параметрический алгоритм оптимизации расписаний параллельных систем с заданным расписанием начала обслуживания // Научный вестник НГТУ. – 2018. – № 3 (72). – С. 87–106. – doi: 10.17212/1814-1196-2018-3-87-106.

For citation:

Mezentsev Yu., Estraykh I. Effektivnyi parametriceskii algoritm optimizatsii raspisanii paralel'nykh sistem s zadannym raspisaniem nachala obsluzhivaniya [An efficient parametric algorithm for optimal scheduling of unrelated parallel machines with release dates]. *Nauchnyi vestnik Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Science bulletin of the Novosibirsk state technical university*, 2018, no. 3 (72), pp. 87–106. doi: 10.17212/1814-1196-2018-3-87-106.