

Многопроцессорные расписания модульных работ с балансом по интервалам¹

Марк Ш. Левин

Институт проблем передачи информации, Российская академия наук, Москва, Россия
email: mslevin@act.org

Поступила в редколлегию 17.12.2018

Аннотация—В статье рассматривается задача составления многопроцессорного расписания составных (модульных) работ с балансом по временным интервалам. Приведен обзор литературы по близким задачам планирования типа “точно-вовремя” (just-in-time). Предложена новая модель планирования для исследуемой задачи. Описан иллюстративный пример планирования в строительстве с учетом согласования производства панелей и процессом сборки домов несколькими бригадами. Эвристическая схема решения включает этапы: (i) кластеризация исходного множества строительных изделий (панелей и др.) для получения восьми базовых групп, которые соответствуют производственным конвейерам; (ii) построение предварительного плана сборки домов (горизонт планирования - примерно два года); (iii) выделение временных интервалов с нарушением баланса по строительным изделиям; (iv) модификация плана сборки домов. Кратко указано о возможном приложении задачи в передаче информации.

КЛЮЧЕВЫЕ СЛОВА: многопроцессорные расписания, комбинаторная оптимизация, баланс по интервалам, балансная кластеризация, строительство, эвристики,

1. ВВЕДЕНИЕ

Базовый взгляд автора на задачи балансной кластеризации содержатся в работах [65, 66]. В данной статье описана специальная новая задача многопроцессорного планирования составных (модульных) работ с балансом по временным интервалам. При этом используется следующая иерархия: (1) множество базовых элементов (модулей); (2) множество составных работ (заданий) для многопроцессорного расписания, состоящие из указанных базовых элементов; (3) результирующее многопроцессорное расписание с учетом ограничение в виде пропорции по использованным базовым элементам на каждом временном интервале. Данное исследование основано на специальной задаче составления расписания с учетом баланса по временным интервалам, использованной для планирования сборки жилых домов. На Рис. 1 представлены основные типы задач планирования по методологии “точно-вовремя” (Just-In-Time - JIT). В конце статьи кратко указано возможное приложение в передаче данных. Статья базируется на предварительном материале [67].

¹ Исследование выполнено в ИППИ РАН за счет гранта Российского научного фонда (проект 14-50-00150).

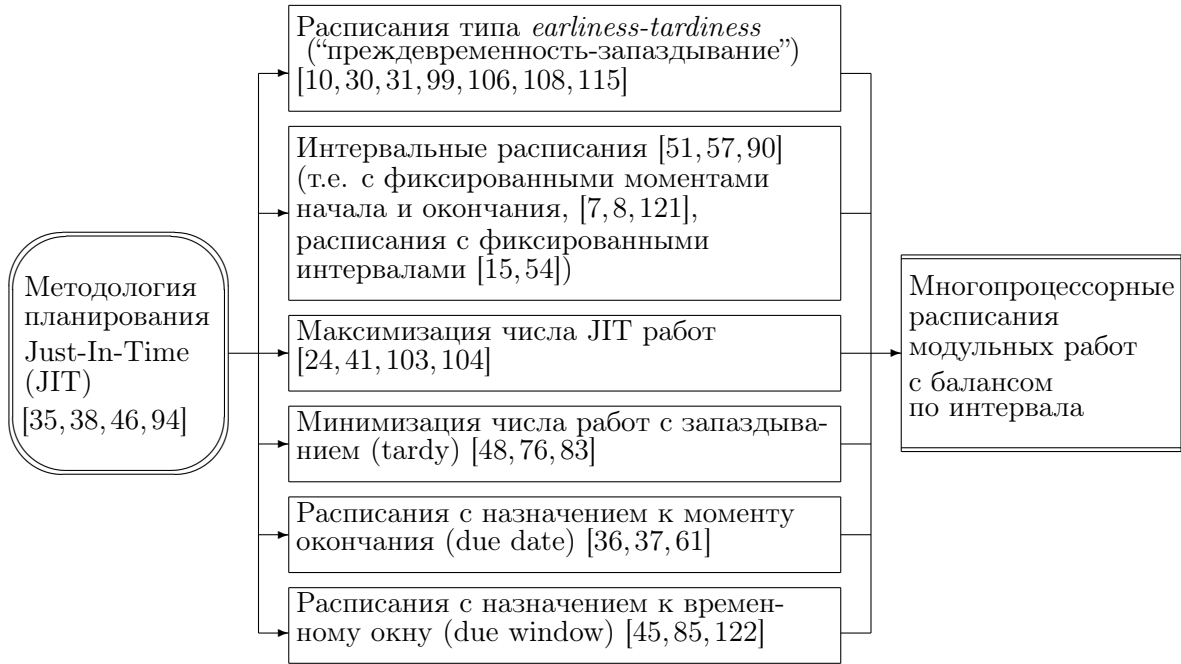


Рис. 1. Основные задачи планирования типа Just-In-Time (JIT)

2. О РАСПИСАНИЯХ ТИПА “ТОЧНО-ВОВРЕМЯ” (JUST-IN-TIME)

Системы планирования расписаний типа “точно-вовремя” (Just-In-Time JIT) или (Kanban системы) имеют широкое распространение в планировании производства и строительства [3, 46, 52, 93, 94]. Задачи планирования типа JIT представляют собой модели на основе принципа “преждевременность-запаздывание” (earliness-tardiness) [5, 6, 30, 31, 46, 61, 69, 93, 99, 100, 108, 115, 117] и модели интервальных расписаний (interval scheduling) [51, 54, 57, 90]. При этом предполагается, что имеются работы, выполняемые точно во время (т.е., к заданному моменту времени) и штрафуются работу, выполненный преждевременно и с запаздыванием. Далее приведен краткий обзор по исследованиям в области методологии планирования Just-In-Time (JIT): (а) общие вопросы расписания типа JIT (Таблица 1), (б) модели расписаний типа JIT (Таблица 2), (в) методы формирования расписания типа JIT (Таблица 3), (г) базовые приложения расписания типа JIT (Таблица 4).

Таблица 1. Основные вопросы в методологии планирования JIT

Ном.	Исследование	Источник
1.	JIT философия, основные обзоры (модели, методы, приложения)	[3, 46, 52, 93, 94]
2.	Литературные обзоры (системы JIT, системы JIT-KANBAN)	[35, 46, 101]
3.	Системы управления типа JIT в строительстве	[92]
4.	Управление производственными потоками в системах JIT	[73]
5.	Методология робастного проектирования в системах типа Kanban	[82]
6.	Уровневый подход к производственным JIT системам	[34]

Ниже описаны два упрощенных иллюстративных примера.

Рассматривается исходное множество работ (заданий) $A = \{a_1, \dots, a_i, \dots, a_n\}$, имеются данные:

- (а) время обработки (выполнения) работы $\theta(a_i); \forall a_i \in A$;
- (б) общий горизонт планирования $[0, T]$ для обработки работ множества A ;
- (в) временной интервал для обработки работы a_i (т.е., начальный момент и момент окончания): $\lambda^{a_i} = [t_1^{a_i}, t_2^{a_i}], t_1^{a_i} < t_2^{a_i}; t_1^{a_i}, t_2^{a_i} \in [0, T]$.

Таблица 2. Некоторые типовые задачи расписаний JIT

№.	Задача	Источник
1.	Обзор задач с максимизацией числа JIT работ	[104]
2.	Максимизация взвешенного числа JIT работ (параллельные процессоры)	[1, 41, 107]
3.	Максимизация числа JIT задачах типа flow-shop	[24, 40, 103]
4.	Обзор задач типа JIT с работами равного размера	[105, 114]
5.	Задачи с взвешенными запаздыванием и выполнением раньше заданного времени при критерии Minmax и одинаковыми временами выполнения работ	[33]
6.	Многокритериальные задачи типа JIT	[12, 42, 84, 110]
7.	Дву-критериальный подход с управляемыми временами обработки	[119, 123]
8.	Расписания с фиксированными интервалами выполнения и неопределенностью	[15]
9.	Задачи типа JIT с неопределенными параметрами	[14]
10.	Стохастические задачи с минимизацией числа запаздываний	[28, 29, 79, 111]
11.	Стохастические расписания типа JIT на одном процессоре	[102, 109]
12.	Стохастические расписания на параллельных процессорах	[16]
13.	Задачи типа JIT с смешанными моделями многими уровнями	[13, 59, 77]
14.	Задачи типа JIT в режиме реального времени (как взвешенные паросочетания)	[74, 93]
15.	Циклические производственные задачи с политикой доставки типа JIT	[89]
16.	Задачи типа JIT с управляемыми временами выполнения (параллельные процессоры)	[68]
17.	Задачи типа JIT с конкурирующими агентами	[25]

Таблица 3. Подходы к решению

Ном.	Подход к решению	Источник
1.	Общие подходы:	
1.1.	Подходы к упорядочению на основе смешанных моделей в JIT производственных системах (обзор)	[26]
1.2.	Декомпозиционные методы для JIT систем	[20, 55]
1.3.	Алгоритмы динамического программирования	[78, 108]
1.4.	Унифицированный подход на основе целевого программирования для JIT производственных систем	[69]
1.5.	Имитационное моделирование для JIT расписаний производственных систем	[43]
2.	Переборные методы:	
2.1.	Алгоритмы типа ветвей-и-границ (Branch-and-Bound)	[9, 50, 98, 106]
2.2.	Алгоритм типа Branch-and-Cut	[91]
2.3.	Алгоритм динамического программирования	[78]
3.	Подходы математического программирования на основе ограничений:	
3.1.	Программирование с ограничениями для JIT расписаний	[80]
3.2.	Программирование с ограничениями для стохастических расписаний (минимизация работ с запаздыванием)	[28]
4.	Некоторые полиномиальные методы:	
4.1.	Полиномиальные и полиномиальные приближенные алгоритмы (PTAS) для параллельных процессоров	[113]
4.2.	Полиномиально разрешимые задачи типа JIT с многими требованиями по времени выполнения (multiple due dates)	[112]
4.3.	Алгоритм квадратичной сложности для максимизации JIT работ (идентичные параллельные процессоры)	[19]
4.4.	Полиномиальные алгоритмы для минимизации функций штрафов в задачах типа JIT (параллельные машины с идентичными работами)	[27]
4.5.	Полиномиальные приближенные алгоритмы с гарантированными погрешностями (fully polynomial-time approximation schemes FPTAS)	[47, 49, 53]
5.	Эвристики и мета-эвристики:	
5.1.	Эвристики на параллельных процессорах	[60]
5.2.	Эволюционные (генетические) алгоритмы	[95, 117, 124]
5.3.	Методы Tabu search	[14, 15, 75]
5.4.	Мета-эвристики	[2, 6, 12, 23]

Таблица 4. Некоторые области приложений методологии планирования JIT

Ном.	Область приложения	Источник
1.	Планирование расписания в производственных системах	[46, 58, 78, 81]
2.	Оптимизация регистров при проектировании компьютеров	[97]
3.	Расписания в сетях связи (сенсоры, переключающие устройства)	[70, 74, 93, 120]
4.	Планирование в строительстве	[92]
5.	Назначение частот с системах связи	[11, 21, 39, 54]
6.	Проектирование интегральных схем (VLSI)	[17]
7.	Планирование расписаний на транспорте и в учебных учреждениях	[4, 12, 18, 56, 72]
8.	Идентификация в биоинформатике	[22]
9.	Планирование технического обслуживания	[56, 86]

Числовые примеры представлены следующим образом:

Пример 1: расписание для одного процессора (Рис. 2, Таблица 5), работы $A = \{a_1, \dots, a_7\}$, общий горизонт планирования $[0, 5.0]$.

Пример 2: расписание для трех процессоров (Рис. 3, Таблица 6), работы $A = \{a_1, \dots, a_{12}\}$, общий горизонт планирования $[0, 5.0]$.

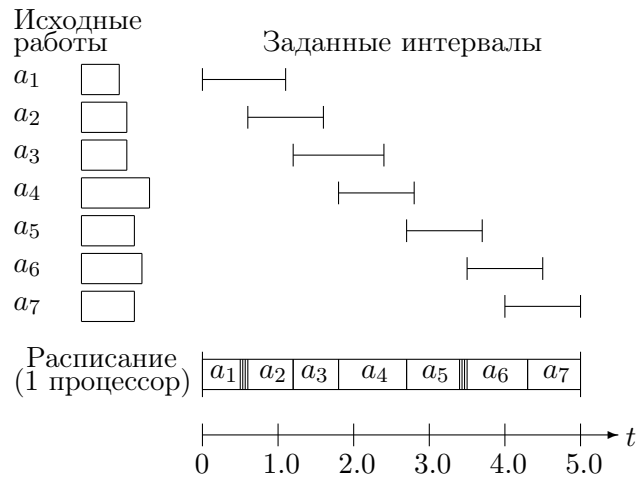


Рис. 2. Иллюстрация расписания (1 процессор)

Таблица 5. Исходные данные для примеров интервальных расписаний

Ном.	Работа a_i	Время выполнения $\theta(a_i)$	Интервал выполнения $\lambda^{a_i} = [t_1^{a_i}, t_2^{a_i}]$	Позиция в результирующем расписании
1.	a_1	0.5	$[0.0, 1.1]$	1
2.	a_2	0.6	$[0.6, 1.6]$	2
3.	a_3	0.6	$[1.2, 2.4]$	3
4.	a_4	0.9	$[1.8, 2.8]$	4
5.	a_5	0.7	$[2.7, 3.7]$	5
6.	a_6	0.8	$[3.5, 4.5]$	6
7.	a_7	0.7	$[4.0, 5.0]$	7

Следует указать основные целевые функции, которые обычно используются в методологии типа JIT:

1. минимизация общего времени выполнения исходного множества работ (makespan) [46],
2. максимизация числа работ, выполненных точно вовремя (JIT) [24, 104],
3. максимизация взвешенного числа работ, выполненных точно вовремя (JIT) [1, 41, 107],
4. минимизация числа работ, выполненных с запаздыванием [28],
5. минимизация взвешенных штрафов за выполнение работ ранее и позже указанного интервала [6, 44].

- 6. минимизация взвешенного числа работ, выполненных преждевременно и с запаздыванием [44],
- 7. минимизация прогнозируемого числа работ с запаздыванием [29],
- 8. минимизация прогнозируемой стоимости преждевременно выполненных работ и работ с запаздыванием [16].

Таблица 6. Данные для примеров интервального расписания

Ном.	Работа a_i	Время выполнения $\theta(a_i)$	Интервал выполнения $\lambda^{a_i} = [t_1^{a_i}, t_2^{a_i}]$	Номер процессора	Позиция в результирующем расписании
1.	a_1	1.2	[0.0, 1.5]	1	1
2.	a_2	1.3	[1.0, 2.5]	1	2
3.	a_3	1.2	[2.0, 4.0]	1	3
4.	a_4	1.1	[3.7, 5.0]	1	4
5.	a_5	0.7	[0.0, 2.0]	2	1
6.	a_6	0.6	[1.7, 2.7]	2	2
7.	a_7	0.7	[2.5, 4.0]	2	3
8.	a_8	1.0	[3.9, 5.0]	2	4
9.	a_9	1.2	[0.0, 1.5]	3	1
10.	a_{10}	1.3	[1.0, 2.5]	3	2
11.	a_{11}	1.2	[2.6, 4.0]	3	3
12.	a_{12}	1.2	[3.0, 5.0]	3	4

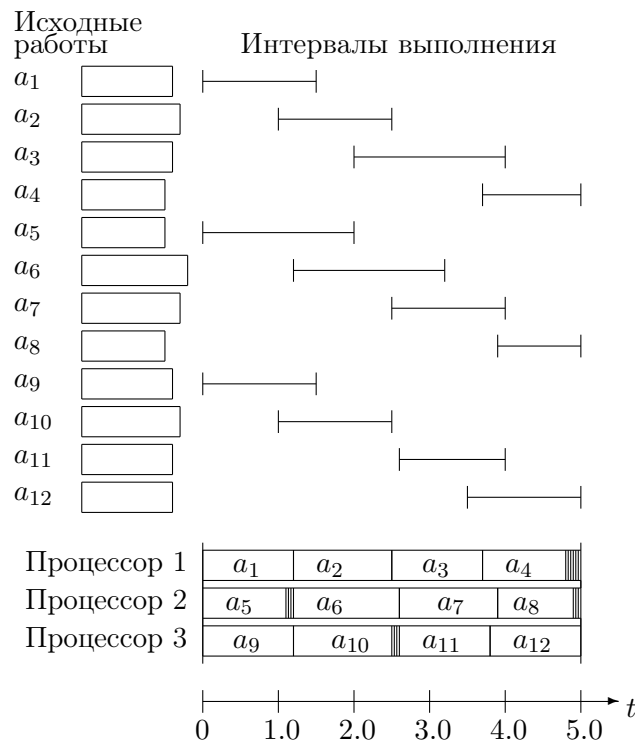


Рис. 3. 3-процессорное расписание

Базовая формальная постановка задачи рассматриваемого типа может иметь следующий вид. Пусть имеется множество работ (заданий) $A = \{a_1, \dots, a_i, \dots, a_n\}$, которое выполняется на m одинаковых процессорах (машинах) $P = \{P_1, \dots, P_j, \dots, P_m\}$. Каждый процессор P_j может выполнять только одну работу в один момент времени, каждая работа a_i может быть выполнена на любом процессоре. Каждая работа a_i имеет параметры: (а) время обработки (выполнения) $\theta(a_i)$, (б) временной интервал обработки (т.е., окно) $\lambda^{a_i} = [t_1^{a_i}, t_2^{a_i}]$, (с) время завершения

обработки работы a_i в расписании S обозначается $C(a_i)$, (d) начало интервала выполнения работы обозначается $u(a_i) = \max\{0, t_1^{a_i} - C(a_i)\}$, (e) окончание интервала выполнения работы обозначается $v(a_i) = \max\{0, C(a_i) - t_2^{a_i}\}$. Кроме того, задаются величины неотрицательного штрафа (стоимости) преждевременного начала выполнения работы $\alpha u(a_i)$ и опоздания завершения выполнения работы $\beta v(a_i)$. Модель имеет вид (целевая функция - как сумма штрафов)

Найти расписание S такое, что минимизируется $F^{sum}(S) = \sum_{i=1}^n [\alpha u(a_i) + \beta v(a_i)]$.

В задаче может минимизироваться максимальный штраф, например:

минимизировать $F^{max}(S) = \max_{1 \leq i \leq n} \{\alpha u(a_i), \beta v(a_i)\}$

Дополнительно целесообразно указать, что ряд моделей комбинаторной оптимизации близки к расписаниям типа JT, например: задачи k -расцветки, задача максимальной взвешенной клики, задачи назначения/размещения, задачи формирования расписаний в спорте и учебных учреждениях, задача максимального независимого множества, задача взвешенного сочетания [54, 74, 93].

Некоторые специальные упрощенные модели планирования типа JT могут решаться на основе полиномиальных алгоритмов [19, 45, 87, 112, 113, 116, 118] или полиномиальных приближенных алгоритмов с гарантированными погрешностями (PTAS/FPTAS) [47, 49, 53, 113]. В основном, модели этого типа являются NP-трудными [30, 31, 42, 45, 50, 90, 110]. Таким образом, следующие методы решения могут быть использованы: (i) переборные методы (методы типа ветвей и границ, динамическое программирование) [9, 50, 78], (ii) различные эвристики и мета-эвристики (локальная оптимизация, эволюционные алгоритмы, и др.) [60, 95]. Очевидно, аналогичная ситуация имеет место в случае многокритериальных задач типа JT [12, 110]).

3. РАСПИСАНИЕ ДЛЯ МОДУЛЬНЫХ РАБОТ С БАЛАНСОМ ПО ИНТЕРВАЛАМ

Рассматривается трех-стадийная система: стадия производства (производство множества базовых модулей), транспортная стадия, и стадия использования модулей - сборка. Задача заключается в построении расписания для сборки из составных работ (заданий) на третьей стадии с учетом ограничения(ий) баланса со стороны первой и второй стадий. Следует заметить, что основная цель заключается в удалении подсистемы буфера перед третьей стадией или ее использования в виде некой простейшей подсистемы (как в системах JT).

На Рис. 4 представлен иллюстративный пример. Имеется множество из пяти базовых элемента (детали, элементы строительства, элементарные работы/задания, модули): $\bar{\Delta} = \{\Delta_1, \Delta_2, \Delta_3, \Delta_4, \Delta_5\}$. Указанные элементы производятся на пяти производствах (например, организациях, конвейерах). Система направлена на выполнение (использование, обработку, сборку) четырех составных (модульных) работ: $a_1 = \langle \Delta_1 \rightarrow \Delta_2 \rightarrow \Delta_3 \rangle$, $a_2 = \langle \Delta_2 \rightarrow \Delta_5 \rangle$, $a_3 = \langle \Delta_1 \rightarrow \Delta_2 \rightarrow \Delta_4 \rightarrow \Delta_5 \rangle$, $a_4 = \langle \Delta_1 \rightarrow \Delta_2 \rightarrow \Delta_2 \rightarrow \Delta_3 \rightarrow \Delta_4 \rightarrow \Delta_5 \rangle$.

В примере, процесс обработки (сборки) выполняется на основе трех процессоров/бригад (Рис. 4). Общее расписание имеет вид $S = \{S_1, S_2, S_3\}$, где a_0 обозначает "пустой" элемент или временной период): (i) процессор (бригада) P_1 : $S_1 = \langle a_4 \rightarrow a_4 \rangle$, (ii) процессор (бригада) P_2 : $S_2 = \langle a_2 \rightarrow a_0 \rightarrow a_3 \rightarrow a_0 \rightarrow a_3 \rangle$, and (iii) процессор (бригада) P_3 : $S_3 = \langle a_3 \rightarrow a_1 \rightarrow a_3 \rightarrow a_0 \rangle$.

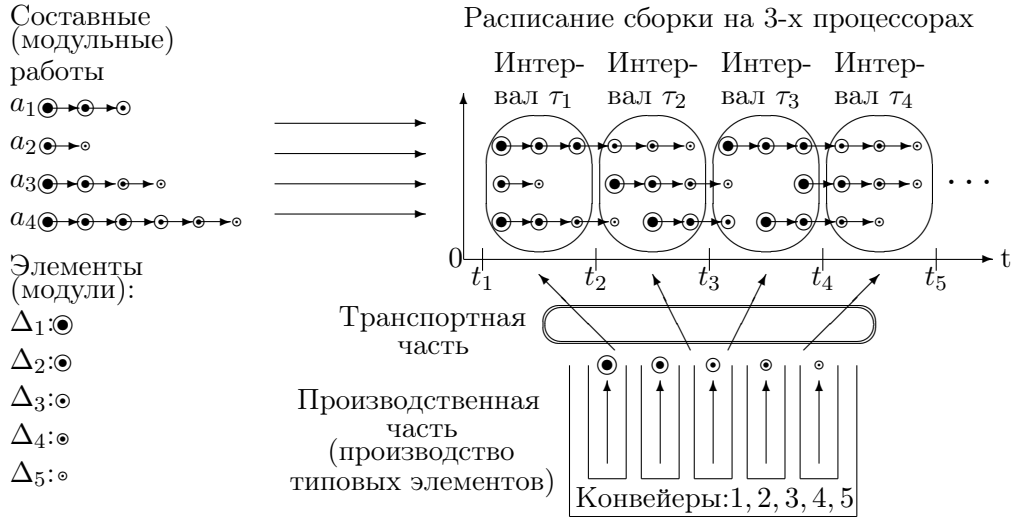


Рис. 4. Многопроцессорное расписание (баланс по элементной структуре)

Базовое требование к общему расписанию S направлено на построение расписания сборки с балансом по элементной структуре на каждом временном интервале, здесь: $\tau_1 = [t_1, t_2]$, $\tau_2 = [t_2, t_3]$, $\tau_3 = [t_3, t_4]$, $\tau_4 = [t_4, t_5]$. На каждом временном интервале используются следующие элементы (также дополнительный 6-й “постоя” элемент Δ_6 , этот тип соответствует “пустому” элементу/периоду a_0):

- (1) $X_{\tau_1} = \{\Delta_1, \Delta_1, \Delta_2, \Delta_2, \Delta_2, \Delta_2, \Delta_3, \Delta_5, \Delta_6\}$,
 - (2) $X_{\tau_2} = \{\Delta_1, \Delta_1, \Delta_2, \Delta_2, \Delta_3, \Delta_4, \Delta_4, \Delta_5, \Delta_5\}$,
 - (3) $X_{\tau_3} = \{\Delta_1, \Delta_1, \Delta_1, \Delta_2, \Delta_2, \Delta_2, \Delta_3, \Delta_5, \Delta_6\}$,
 - (4) $X_{\tau_4} = \{\Delta_2, \Delta_3, \Delta_4, \Delta_4, \Delta_4, \Delta_5, \Delta_5, \Delta_5, \Delta_6\}$.
- Очевидно, решение кластеризации имеет вид: $\tilde{X}(S) = \{X_{\tau_1}, X_{\tau_2}, X_{\tau_3}, X_{\tau_4}\}$.

Получаются следующие соответствующие оценки на основе мультимножеств (по числу базовых типов элементов [64]): $e(X_{\tau_1}) = (2, 4, 1, 0, 1, 1)$, $e(X_{\tau_2}) = (2, 2, 1, 2, 2, 0)$,

$e(X_{\tau_3}) = (3, 3, 1, 0, 1, 1)$, $e(X_{\tau_4}) = (0, 1, 1, 3, 3, 1)$.

Базовая оценка на основе мультимножеств должна соответствовать выходной структуре (т.е., производительности) производственной системы (с учетом транспортной системы), например: $e_0 = (2, 3, 2, 1, 1, 0)$. Близости между оценками $(\delta(e_0, e(X_{\tau_i})), \iota = \overline{1, 4})$ содержатся в Таблице 7.

Таблица 7. Близости между оценками кластеров и базовыми оценками $(\delta(e_0, e(X_{\tau_i}))$

Базовая оценка	Оценка кластера:	$e(X_{\tau_1}) = (2, 4, 1, 0, 1, 1)$	$e(X_{\tau_2}) = (2, 2, 1, 2, 2, 0)$	$e(X_{\tau_3}) = (3, 3, 1, 0, 1, 1)$	$e(X_{\tau_4}) = (0, 1, 1, 3, 3, 1)$
$e_0 = (2, 3, 2, 1, 1, 0)$		3	3	4	15

Пусть $P = \{P_1, \dots, P_\xi, \dots, P_m\}$ - множество процессоров/бригад. Задача составления расписания имеет вид:

Найти общее расписание S и соответствующее решение кластеризации

$$\tilde{X}(S) = \{X_{\tau_1}, X_{\tau_2}, X_{\tau_3}, \dots, X_{\tau_k}\}$$

такие, что (i) число временных интервалов (число k), т.е., длина на расписаний - makespan) является минимальным, (ii) близость между элементной структурой для каждого интервала

и базовой (заданной) структурой является ограниченной (т.е., $\max_{\iota=1,k} \delta(e_0, e(X_{\tau_\iota})) \leq \delta^0$, δ^0 - некоторое совместное ограничение для производственной и транспортной частей).

Общая формальная модель имеет вид (m процессоров/бригад, $L(S) = k$):

$$\min L(S) = \max_{\xi=1,m} L(S_\xi) \quad s.t. \quad \max_{\iota=1,k} \delta(e_0, e(X_{\tau_\iota})) \leq \delta^0.$$

В приведенном числовом примере $L(S) = 4$, $\widehat{B}(\widetilde{X}(S)) = 4$.

Иллюстрация для рассматриваемой иерархии представлена на Рис. 5.



Рис. 5. Иерархия исследуемых компонентов

Очевидно, что могут рассматриваться различные версии предложенной задачи (например, использование различных шкал для оценивания параметров, различные ограничения, различные целевые функции, разные уровни неопределенности).

Дополнительно следует указать следующее:

Замечание 1. Указанные типы моделей являются очень сложными (NP-сложными), за исключением некоторых простейших постановок. Очевидно, необходимы специальные исследования сложности для задач рассмотренного типа.

Замечание 2. Перспективным является использование схем решения на основе различных эвристик и метаэвристик.

4. ПЛАНИРОВАНИЕ В СТРОИТЕЛЬСТВЕ С БАЛАНСОМ ПО ИНТЕРВАЛАМ

4.1. Строительный цикл и балансировка

Общий цикл в крупно-блочном строительстве включает следующие этапы:

0. Предварительный этап: анализа требований к зданию и функциональности, геологические исследования и др.

1. Генерация архитектурных идей и проектирование: (1.1) генерация основных архитектурных предложений, (1.2) конструкторское проектирование (например, расчеты, подготовка проектной документации).

2. Производственный этап: производство строительных блоков, изделий, деталей и т.п.

3. Этап транспортирования и сборки: транспортирование компонентов зданий (блоков, деталей) к месту сборки, сборка здания.

4. Этап эксплуатации (включая техническое обслуживание).

5. Утилизация здания (разрушения здания, утилизация компонентов).

Современные промышленные технологии модульного строительства направлены на уменьшение сроков строительства и уменьшение стоимости процесса строительства [62, 88]. В 1980

году в Москве был предложен новый подход к крупнопанельному жилищному строительству на основе метода каталожных элементов КОПЭ (каталожный объемно-планировочные элементы) [71, 96]. Следует отметить, что этап производства строительных изделий является наиболее затратным и инерционным. Поэтому координация (согласование, балансировка) между стадией производства строительных изделий и стадией сборки домов часто является узким местом. Очевидно, что наилучшим решением является организация на основе подхода ЛПТ. В реальных условиях строительства допускается временной интервал между производством строительных блоков/деталей и их использованием на стадии сборки до 3-х дней. Но в качестве главного требования выступает необходимость согласования (балансировки) использования (т.е., сборки) типов строительных деталей с учетом типа производства.

Далее приведена задача балансировки [63]. Рассматривается 3-стадийная система (Рис. 6): (i) производство; (ii) транспортировка; (iii) сборка. В последующем примере, временной интервал равен месяцу (для упрощения).

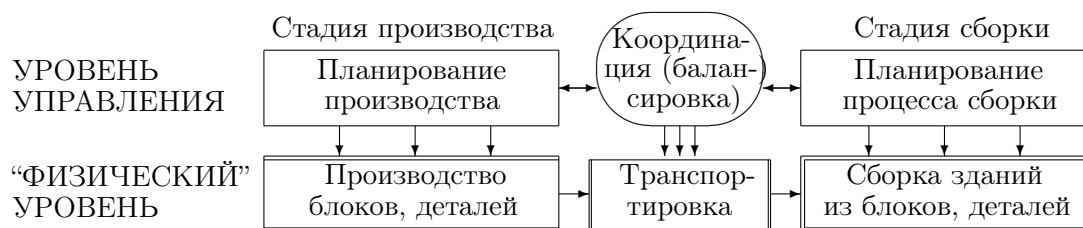


Рис. 6. Схема производства и строительства

4.2. Иллюстративная задача планирования

В данном разделе описан пример формирования плана строительства (метод КОПЭ, домостроительный комбинат ДСК-2, Москва, 1982–1983) [63]. Пример основан на программе (Fortran, разработана автором [63]). Подробные данные по расчетам содержатся в [63] (частичные данные приведены в [67]). В системе КОПЭ используются следующие основные архитектурные секции (как колонны):

- (а) каталожные структурные элементы (соответствуют квартире): g_1, g_2, g_5, g_9 ;
- (б) соединительные элементы (для соединения структурных элементов): w_1, w_2, w_3, w_6, w_7 .

Система строительства по методу КОПЭ базируется на каталоге деталей, включающем около 500 элементов. Для уменьшения размерности задачи детали были сгруппированы в 8 основных типов (с учетом производственных мощностей): (1) наружные стеновые панели d_1 , (2) внутренние стеновые панели для 1-го этажа d_2 , (3) внутренние стеновые панели d_3 , (4) перекрытия d_4 , (5) панели покрытия d_5 , (6) лестницы (лестничные марши) d_6 , (7) элементы инженерных коммуникаций d_7 , (8) элементы балконов d_8 .

Кроме того, был выделен один тип деталей (наружные стеновые панели d_1), производство которых является наиболее затратным и инерционным. Т.е., требования баланса должны выполняться, в первую очередь, по данному типу деталей.

В качестве типовых этажей рассматриваются следующие: (i) подвал r_1 , (2) первый этаж r_2 , (3) нижний этаж r_3 , (4) средний этаж r_4 , (5) верхний этаж r_5 , (6) последний этаж r_6 , (7) чердак r_7 , (8) engineering storey (mechanical floor) r_8 . Конфигурации рассматриваемых 2-х типов зданий по типовым этажам приведены в Таблице 8.

Таблица 8. Конфигурация типовых зданий на основе типовых этажей

№.	Тип здания	Тип этажа: r_1 r_2 r_3 r_4 r_5 r_6 r_7 r_8							
1.	18-этажное здание	0	1	0	11	5	1	1	1
2.	22-этажное здание	0	1	4	11	5	1	1	1

Иерархия конструктивных элементов в крупно-панельном строительстве представлены на Рис. 7.

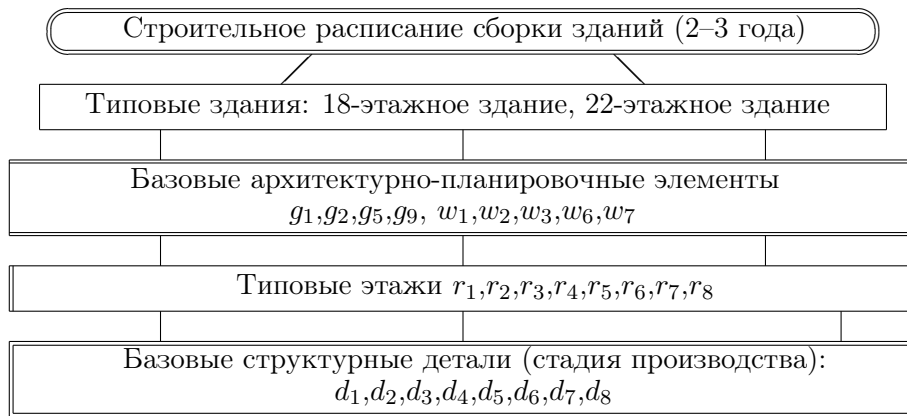


Рис. 7. Иерархия: структурные панели – здания

Предлагаемая эвристическая схема решения основана на модификации исходного расписания (Рис. 8).

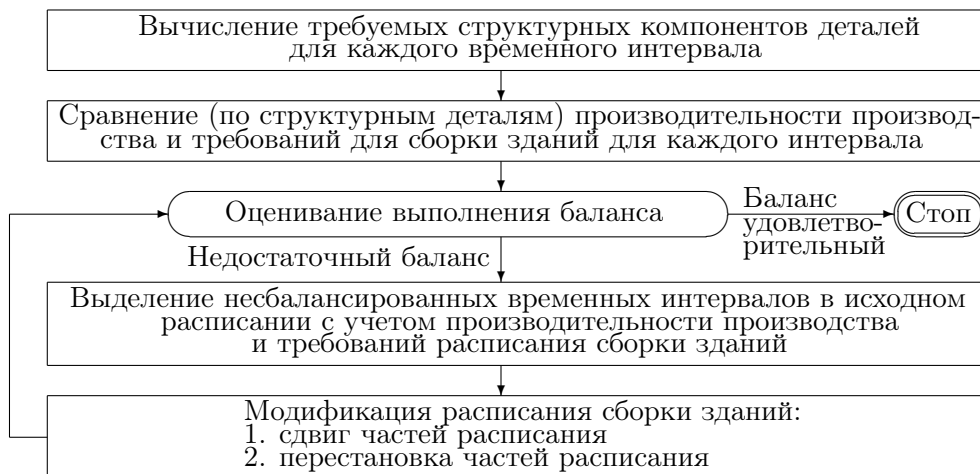


Рис. 8. Эвристическая схема решения для балансирования расписания

Таблица 9 содержит список рассматриваемых зданий.

Таблица 9. Список зданий

Ном.	Обозначение	Адрес	Число этажей	Общая площадь	Время сборки	Начало сборки
1.	a_1 (B1A)	Воронцово, кор. 1А	18	17.5	9.0	0.5
2.	a_2 (B2)	Воронцово, кор. 2	22	16.4	6.2	8.0
3.	a_3 (B5A)	Воронцово, кор. 5А	18	13.3	4.5	6.5
4.	a_4 (B6A)	Воронцово, кор. 6А	18	17.7	4.8	7.0
5.	a_5 (B6Б)	Воронцово, кор. 6Б	22	24.0	6.4	8.8
6.	a_6 (B2А)	Воронцово, кор. 2А	22	11.3	3.0	9.5
7.	a_7 (B2А)	Бибирево, кор. 2А	22	16.3	4.3	11.8
8.	a_8 (B2Б)	Бибирево, кор. 2Б	22	22.7	6.1	9.7
9.	a_9 (B4)	Бибирево, кор. 4	22	29.0	7.8	11.0

Конфигурация зданий на основе типовых секций приведена в Таблице 10. Предварительное расписание сборки зданий S^0 для восьми строительных бригад представлено на Рис. 9. Горизонт планирования включает 19 месяцев (январь 1982 – июль 1983).

Таблица 10. Конфигурация зданий на основе типовых секций

Здание	g_1	g_2	g_5	g_9	w_1	w_2	w_3	w_6	w_7
a_1	2	1	1	0	3	0	4	0	0
a_2	0	2	0	1	1	1	4	0	1
a_3	1	1	1	0	2	0	4	0	0
a_4	0	1	1	1	1	1	6	0	0
a_5	0	1	1	2	1	2	8	0	0
a_6	2	1	1	0	1	1	4	0	0
a_7	1	1	0	1	1	1	5	0	1
a_8	1	1	1	1	2	1	6	0	0
a_9	1	1	1	2	2	2	7	0	0

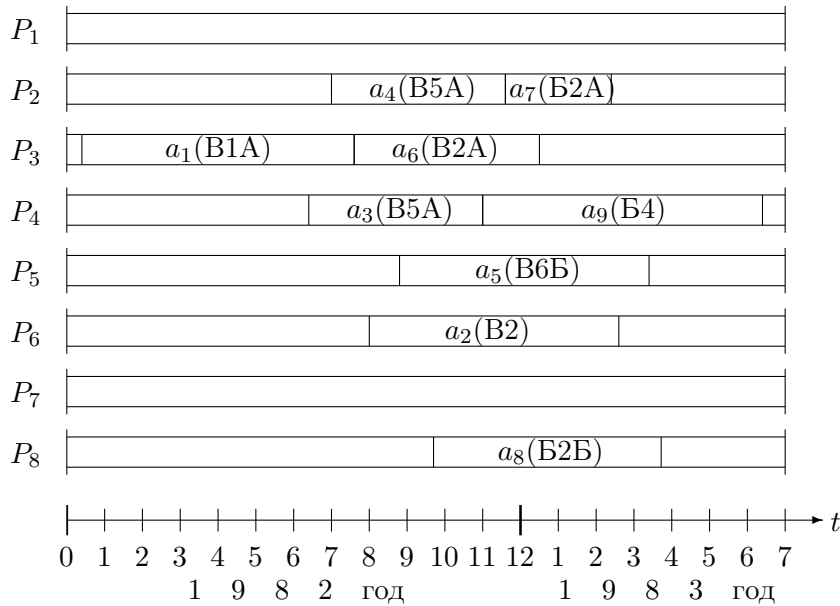


Рис. 9. Исходное расписание сборки зданий S^0

Таблица 11 содержит требования по структурным секциям для каждого интервала указанного расписания. Задача формирования расписания рассматривается следующим образом. Имеются следующие компоненты:

- (1) множество процессоров (бригады) $P = \{P_1, \dots, P_8\}$;
- (2) множество зданий $A = \{a_1, \dots, a_9\}$;
- (3) 19 временных интервалом (месяцев) $(\tau_1, \dots, \tau_\epsilon, \dots, \tau_{19})$;
- (4) расписание S (Рис. 9);
- (5) решение кластеризации для расписания S : $\tilde{X}(S) = \{X_{\tau_1}, \dots, X_{\tau_\epsilon}, \dots, X_{\tau_{19}}\}$, X_{τ_ϵ} соответствует множеству требуемых деталей для интервалов τ_ϵ ($\epsilon = \overline{1, 19}$);
- (6) оценки по требуемым деталям (панелям) для интервалов τ_ϵ ($\epsilon = \overline{1, 19}$): $\gamma_{d_1}(X_{\tau_\epsilon})$ - число требуемых деталей d_1 для временного интервала τ_ϵ , и т.д. (Таблица 12). В Таблице 12: $\mu(\gamma_{d_\kappa}) = \frac{\gamma_{d_\kappa}}{\sum_{\kappa=\overline{1,8}} \gamma_{d_\kappa}}$, $\kappa = \overline{1, 8}$;
- (7) производительность (по деталям d_1, \dots, d_8) производственной системы равна $\hat{\gamma}_0 = \{\gamma_{d_1}^0, \dots, \gamma_{d_\kappa}^0, \dots, \gamma_{d_8}^0\}$.

Таблица 11. Требования по структурным секциям по каждому интервалу (месяцу)

Ном.	Месяц	g_1	g_2	g_5	g_9	w_1	w_2	w_3	w_6	w_7
1.	1/1982	0.091	0.045	0.045	0.0	0.136	0.0	0.182	0.0	0.0
2.	2/1982	0.182	0.091	0.091	0.0	0.273	0.0	0.364	0.0	0.0
3.	3/1982	0.182	0.091	0.091	0.0	0.273	0.0	0.364	0.0	0.0
4.	4/1982	0.182	0.091	0.091	0.0	0.273	0.0	0.364	0.0	0.0
5.	5/1982	0.182	0.091	0.091	0.0	0.273	0.0	0.364	0.0	0.0
6.	6/1982	0.182	0.091	0.091	0.0	0.273	0.0	0.364	0.0	0.0
7.	7/1982	0.274	0.183	0.183	0.0	0.456	0.0	0.731	0.0	0.0
8.	8/1982	0.365	0.447	0.447	0.172	0.812	0.172	2.131	0.0	0.0
9.	9/1982	0.365	0.800	0.478	0.396	1.004	0.396	3.024	0.0	0.161
10.	10/1982	0.324	1.095	0.772	0.693	1.257	0.693	4.796	0.0	0.161
11.	11/1982	0.345	1.328	1.006	0.818	1.512	0.818	5.976	0.0	0.161
12.	12/1982	0.350	1.290	0.909	1.078	1.421	1.078	6.163	0.0	0.219
13.	1/1983	0.524	1.169	0.614	1.124	1.299	1.124	5.594	0.0	0.394
14.	2/1983	0.524	1.002	0.447	1.124	1.133	1.124	4.927	0.0	0.394
15.	3/1983	0.524	0.774	0.447	0.995	1.003	0.995	4.411	0.0	0.265
16.	4/1983	0.492	0.530	0.298	0.697	0.789	0.697	3.155	0.0	0.233
17.	5/1983	0.140	0.140	0.128	0.267	0.267	0.267	0.953	0.0	0.012
18.	6/1983	0.128	0.128	0.128	0.256	0.256	0.256	0.895	0.0	0.0
19.	7/1983	0.100	0.100	0.100	0.199	0.199	0.199	0.698	0.0	0.0

Таблица 12. Требования в расписании S^0 типовых структурных деталей по месяцам τ_ϵ

τ_ϵ	$\gamma_{d_1}; \mu(\gamma_{d_1})$	$\gamma_{d_2}; \mu(\gamma_{d_2})$	$\gamma_{d_3}; \mu(\gamma_{d_3})$	$\gamma_{d_4}; \mu(\gamma_{d_4})$	$\gamma_{d_5}; \mu(\gamma_{d_5})$	$\gamma_{d_6}; \mu(\gamma_{d_6})$	$\gamma_{d_7}; \mu(\gamma_{d_7})$	$\gamma_{d_8}; \mu(\gamma_{d_8})$
1	79; 24.7	122; 37.9	27; 8.6	70; 21.9	0; 0.0	8; 2.6	9; 2.9	4; 1.3
2	137; 21.7	0; 0.0	253; 40.0	166; 26.3	0; 0.0	16; 2.7	50; 8.0	8; 1.3
3	137; 21.7	0; 0.0	253; 40.0	166; 26.3	0; 0.0	16; 2.7	50; 8.0	8; 1.3
4	137; 21.7	0; 0.0	253; 40.0	166; 26.3	0; 0.0	16; 2.7	50; 8.0	8; 1.3
5	137; 21.7	0; 0.0	253; 40.0	166; 26.3	0; 0.0	16; 2.7	50; 8.0	8; 1.3
6	137; 21.7	0; 0.0	253; 40.0	166; 26.3	0; 0.0	16; 2.7	50; 8.0	8; 1.3
7	250; 22.3	92; 8.2	377; 33.7	279; 24.9	0; 0.0	29; 2.6	77; 6.9	14; 1.3
8	576; 22.5	93; 3.7	932; 36.5	659; 25.8	0; 0.0	66; 2.6	187; 7.3	41; 1.6
9	842; 23.0	181; 5.0	1300; 35.6	912; 24.9	0; 0.0	94; 2.6	251; 6.9	72; 2.0
10	1250; 23.0	222; 4.1	1866; 34.3	1277; 23.5	109; 2.0	129; 2.4	347; 6.4	101; 1.9
11	1468; 22.6	18; 0.3	2448; 37.7	1615; 24.9	84; 1.3	158; 2.4	461; 7.1	122; 1.9
12	1562; 22.9	231; 3.4	2385; 35.3	1654; 24.2	94; 1.4	164; 2.4	459; 6.7	139; 2.0
13	1446; 22.8	0; 0.0	2418; 38.1	1589; 25.8	59; 0.9	155; 2.5	452; 7.1	146; 2.3
14	1296; 22.7	0; 0.0	2187; 38.3	1452; 25.5	26; 0.5	142; 2.5	428; 7.5	138; 2.4
15	1156; 22.5	0; 0.0	1938; 37.7	1244; 24.2	80; 1.6	121; 2.4	373; 7.3	114; 2.2
16	965; 21.5	0; 0.0	1554; 34.6	810; 18.1	305; 6.8	83; 1.9	279; 6.2	76; 1.7
17	289; 23.3	0; 0.0	453; 36.6	305; 24.6	16; 1.3	29; 2.4	90; 7.3	26; 2.1
18	261; 22.6	0; 0.0	447; 38.6	305; 26.4	0; 0.0	29; 2.5	88; 7.6	26; 2.3
19	276; 21.0	0; 0.0	409; 31.2	164; 12.5	154; 11.8	17; 1.4	68; 5.2	13; 1.0

Постановку задачи планирования можно сформулировать так:

Найти расписание S и соответствующие $\tilde{X} = \{X_{\tau_1}, \dots, X_{\tau_\epsilon}, \dots, X_{\tau_{19}}\}$ так, что $(\gamma_{d_1}(X_{\tau_\epsilon}), \dots, \gamma_{d_8}(X_{\tau_\epsilon})) \preceq \hat{\gamma}_0 \quad \forall \tau_\epsilon = \overline{1, 19}$.

Очевидно, что дополнительно в модели можно учитывать и целевую функцию (или функции).

Процесс решений данной задачи можно рассматривать как модификацию (улучшение) исходного расписания (здесь это S^0). В качестве операций модификации исходного расписания

можно привести следующие: (i) сдвиг вправо времени сборки здания a_ζ , (ii) сдвиг влево времени сборки здания a_ζ , (iii) перестановка позиций двух зданий a_{ζ_1} and a_{ζ_2} в расписании: (а) случай 1: a_{ζ_1} и a_{ζ_2} в исходном расписании выполняются на одном процессоре, (б) случай 2: a_{ζ_1} и a_{ζ_2} в исходном расписании выполняются на разных процессорах.

Следует заметить, что более длинное здание (по числу подъездов) содержит пропорционально меньше деталей типа наружных стеновых панелей d_1 .

Таблица 13 содержит возможные операции модификации исходного расписания. Используемые оценки полезности и стоимости имеют только иллюстративный характер.

Таблица 13. Операции модификации расписания $S^0 \implies S'$

Ном. ι	Здание	Тип изменения	Бинарная переменная	Полезность $c_{\iota,j}$	Стоимость $b_{\iota,j}$
1.	a_6	V_1^1 : нет	$y_{1,1}$	0.0	0.0
		V_2^1 : сдвиг вправо, 3 дня	$y_{1,2}$	0.5	1.0
		V_3^1 : сдвиг вправо, 7 дней	$y_{1,3}$	1.5	2.0
		V_4^1 : сдвиг вправо, 14 дней	$y_{1,4}$	2.5	3.0
		V_5^1 : сдвиг вправо, 21 день	$y_{1,5}$	3.5	4.0
2.	a_7	V_1^2 : нет	$y_{2,1}$	0.0	0.0
		V_2^2 : сдвиг вправо, 3 дня	$y_{2,2}$	0.3	0.5
		V_3^2 : сдвиг вправо, 7 дней	$y_{2,3}$	1.0	0.8
		V_4^2 : сдвиг вправо, 14 дней	$y_{2,4}$	1.5	1.0
3.	a_8	V_1^3 : нет	$y_{3,1}$	0.0	0.0
		V_2^3 : сдвиг вправо, 7 дней	$y_{3,2}$	1.5	1.0
		V_3^3 : сдвиг вправо, 14 дней	$y_{3,3}$	2.5	1.5
		V_4^3 : сдвиг вправо, 21 день	$y_{3,4}$	3.5	2.0
4.	a_3, a_6	V_1^4 : нет	$y_{4,1}$	0.0	0.0
		V_2^4 : перестановка позиций	$y_{4,2}$	1.5	2.0

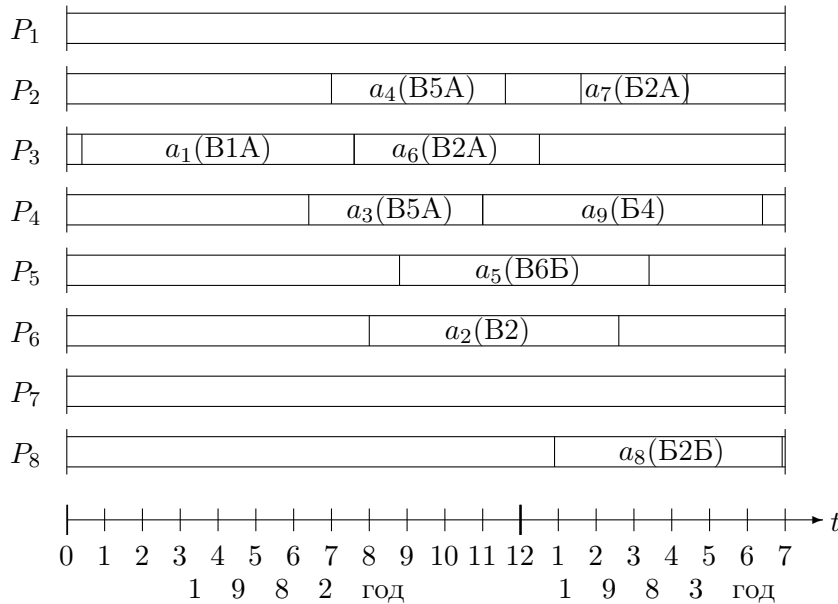


Рис. 10. Новое расписание сборки зданий

Здесь предполагаются независимые операции модификации исходного расписания. Иначе следует применять задачу на основе морфологической клики [64].

Таким образом,используется модель блочного рюкзака ($q_1 = 4, q_2 = 4, q_3 = 4, q_4 = 2$):

$$\max \sum_{\iota=1}^4 \sum_{j=1}^{q_{\iota}} y_{\iota,j} c_{\iota,j} \quad s.t. \quad \sum_{\iota=1}^4 \sum_{j=1}^{q_{\iota}} y_{\iota,j} b_{\iota,j} < b_2^{constr}, \quad \sum_{j=1}^{q_{\iota}} y_{\iota,j} \leq 1 \quad \forall \iota = \overline{1,4}, \quad \forall j = \overline{1, q_{\iota}} \quad \forall y_{\iota,j} \in \{0,1\}.$$

Используется простейшая эвристика (выбор и упаковка элементов в соответствии с значением $\frac{c_{\iota,j}}{b_{\iota,j}}$). Иллюстрация итогового (модифицированного) решения (т.е., улучшения S^0): $I(S^0) = V_1^1 \star V_4^2 \star V_4^3 \star V_1^4$. Бинарное решение имеет вид: $y_{1,1} = 0, y_{2,4} = 1, y_{3,4} = 1, y_{4,1} = 0$. Соответствующее расписание S' приведено на Рис. 10. На Рис. 11 представлена иллюстрация анализа баланса (по наружной стеновой панели d_1) для расписания S^0 и расписания S' .

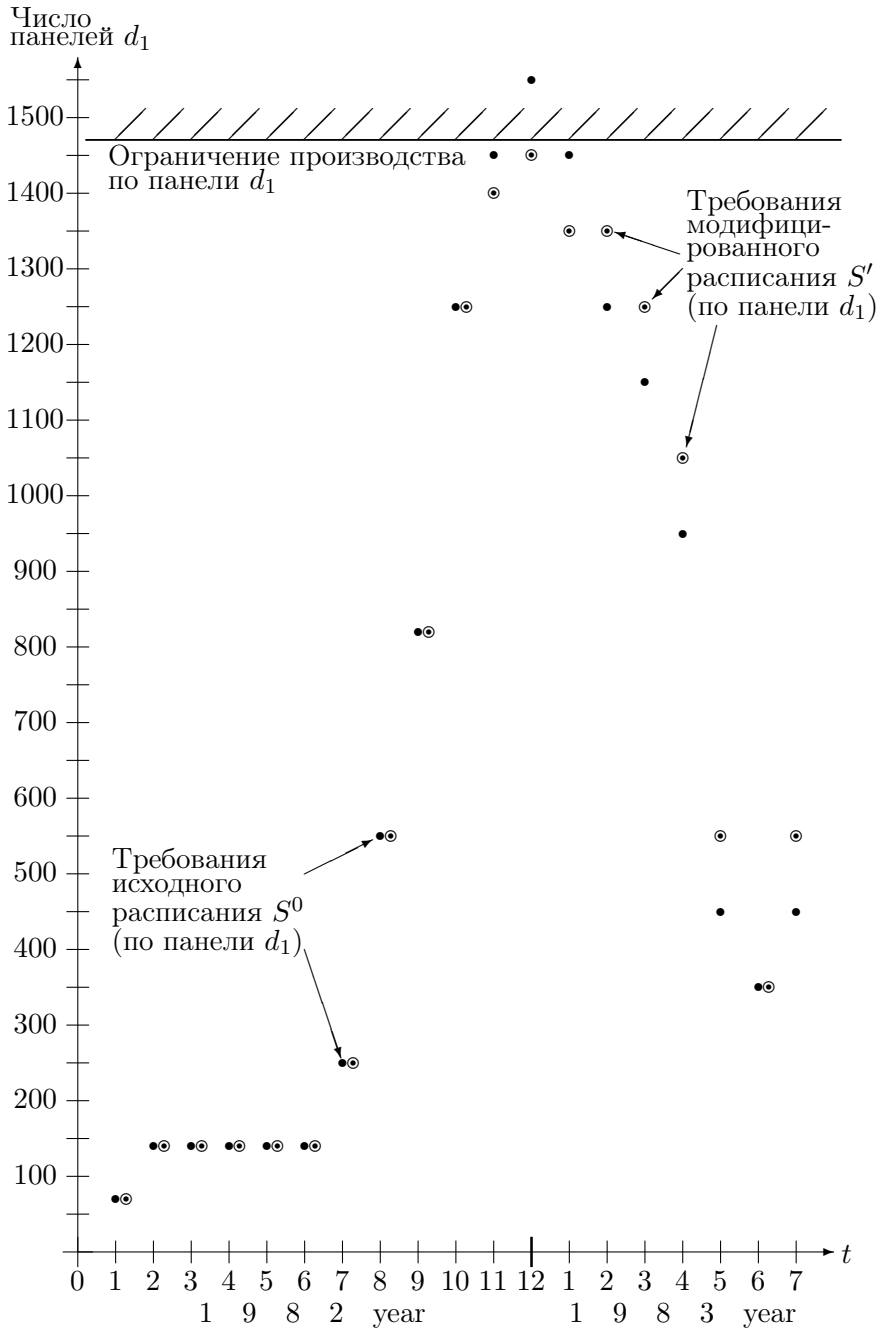


Рис. 11. Анализ баланса по структуре деталей (по детали d_1)

5. О ЗАДАЧЕ С БАЛАНСОМ ПРИ ПЕРЕДАЧЕ ИНФОРМАЦИИ

В последние годы передача информации на основе объектно-ориентированного подхода представляется перспективной. Похожий взгляд на процесс передачи данных можно рассмотреть следующим образом. Пусть имеются η исходных наборов сообщений на основе объектов:

$$A_1 = \langle a_{11}, a_{12}, a_{13}, \dots \rangle, \dots, A_i = \langle a_{i1}, a_{i2}, a_{i3}, \dots \rangle, \dots, A_\eta = \langle a_{\eta 1}, a_{\eta 2}, a_{\eta 3}, \dots \rangle.$$

Каждый набор сообщений A_i основан на множестве стандартных объектов (например: $b_{i1}, b_{i2}, b_{i3}, b_{i4}, \dots$) с предшествованием (как цепочка): $b_{i1} \leftarrow b_{i2} \leftarrow b_{i3} \leftarrow b_{i4} \leftarrow \dots$. Планирование заключается в формировании расписания для k подканалов так, чтобы выполнялось требование баланса по типовым объектам на каждом временном интервале (интервалы: $\tau_1, \tau_2, \tau_3, \dots$). Предполагается, что каждое сообщение может быть передано с некоторым сдвигом по времени. Иллюстративная схема передачи данных с учетом баланса по интервалам приведена на Рис. 12.

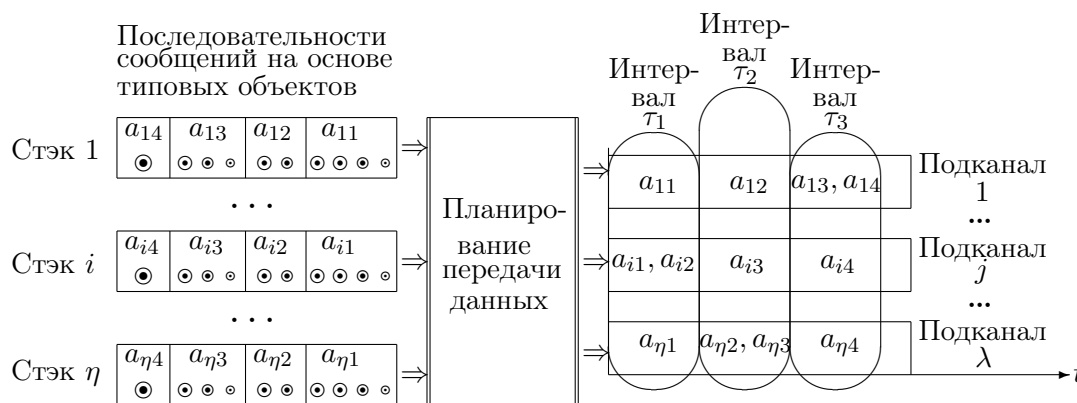


Рис. 12. Передача данных с учетом баланса по интервалам

6. ЗАКЛЮЧЕНИЕ

В работе предложена новая задача планирования многопроцессорных расписания модульных работ с учетом баланса по временным интервалам. Приведены модель задачи, эвристическая схема решения, иллюстративные примеры. Можно указать следующие направления будущих исследований: (1) исследование моделей с неопределенностью (например, с вероятностными оценками, с оценками в виде размытых множеств); (2) анализ применимости других подходов к решению (например, вероятностных алгоритмов); (3) исследование других приложений (например, логистика, сети связи); (4) использование описанного подхода в учебном процессе.

СПИСОК ЛИТЕРАТУРЫ

1. Adamu M.O., Abass O., Parallel machine scheduling to maximize the weighted number of just-in-time jobs. J. of Appl. Sci. and Technol., 2010, vol. 15, no. 1–2, pp. 27–34.
2. Adamu M.O., Adewumi A.O., Metaheuristics for scheduling on parallel machine to minimize weighted number of early and tardy jobs. Int. J. of Phys. Sci., 2012, vol. 7, no. 10, pp. 1641–1652.
3. Akturk M.S., Erhun F., An overview of design and operational issues of kanban systems. Int. J. of Prod. Res., 1999, vol. 37, no. 17, pp. 3859–3881.
4. Alvarez-Perez G.A., Gonzalez-Valarde J.L., Fowler J.W., Cross-docking - just in time scheduling: an alternative solution approach. J. of the ORS, 2009, vol. 60, no. 4, pp. 554–564.

5. Alvarez-Valdes R., Crespo E., Tamarit J.M., Villa F., Minimizing weighted earliness-tardiness on a single machine with a common due date using quadratic models. *TOP*, 2012, vol. 20, pp. 754–767.
6. Alvarez-Valdes R., Tamarit J.M., Villa F., Minimizing weighted earliness-tardiness on parallel machines using hybrid metaheuristics. *Comp. and Oper. Res.*, 2015, vol. 54, pp. 1–11.
7. Angelelli E., Filippi C., On the complexity of interval scheduling with a resource constraint. *Theor. Comp. Sci.*, 2011, vol. 412, no. 29, pp. 3650–3657.
8. Arkin E.M., Silverberg E.L., Scheduling jobs with fixed start and finish times. *Discr. Appl. Math.*, 1987, vol. 18, no. 1, pp. 1–8.
9. Baker K.R., Minimizing earliness and tardiness costs in stochastic scheduling. *EJOR*, 2014, vol. 236, no. 2, pp. 445–452.
10. Baker K.R., Scudder G.D., Sequencing with earliness and tardiness penalties: a review. *Oper. Res.*, 1990, vol. 38, no. 1, pp. 22–36.
11. Bar-Yehuda R., Poleyoy G., Rawitz D., Bandwidth allocation in cellular networks with multiple interferences. *Discr. Appl. Math.*, 2015, vol. 194, no. 23, pp. 23–36.
12. Bolori Arabani A.R., A multi-criteria cross-docking scheduling with just-in-time approach. *Int. J. Adv. Manuf. Technol.*, 2010, vol. 49, no. 5, pp. 741–756.
13. Boysen N., Bock S., Scheduling just-in-time part supply for mixed-model assembly lines. *EJOR*, 2011, vol. 211, no. 1, pp. 15–25.
14. Bozejko W., Rajba P., Wodecki M., Scheduling problem with uncertain parameters in just in time system. In: Rutkowski L., Korytkowski M., Scherer R., Tadeusiewicz R., Zadeh L.A., Zurada J.M. (eds), *Artif. Intell. and Soft Comput.*, LNCS 8468, Springer, pp. 456–467, 2014.
15. Branda M., Novotny J., Olstad A., Fixed interval scheduling under uncertainty - a tabu search algorithm for an extended robust coloring formulation. *Comp. and Ind. Eng.*, 2016, vol. 93, pp. 45–54.
16. Cai X., Zhou S., Stochastic scheduling on parallel machines subject to random breakdowns to minimize expected costs for earliness and tardy jobs. *Oper. Res.*, 1999, vol. 47, no. 3, pp. 422–437.
17. Carlisle M.C., Lloyd E.L., On the k -coloring of intervals. *Discr. Appl. Math.*, 1995, vol. 59, pp. 225–235.
18. Carter M.W., Tovey C.A., When is the classroom assignment problem hard? *Oper. Res.*, 1992, vol. 40, pp. 28–39.
19. Cepek O., Sung S.C., A quadratic time algorithm to maximize the number of just-in-time jobs on identical parallel machines. *Comp. and Oper. Res.*, 2005, vol. 32, no. 12, pp. 3265–3271.
20. Chen Z.-L., Powell W.B., A column generation based decomposition algorithm for a parallel machine just-in-time scheduling problem. *EJOR*, 1999, vol. 116, no. 1, pp. 220–232.
21. Chen B., Hassan R., Tzur M., Allocation of bandwidth and storage. *IIE Trans.*, 2002, vol. 34, pp. 501–507.
22. Chen Z.Z., Jiang T., Lin G.H., Rizzi R., Wen J.J., Xu D., Xu Y., More reliable protein NMR peak assignment via improved 2-interval scheduling. In: Di Battista G., Zwick U. (eds), *Algorithms - ESA 2003*, LNCS 2832, Springer, pp. 580–592, 2003.
23. Chetty S., Adewumi A.O., A study of the enhanced best performance algorithm for the just-in-time scheduling problem. *Discrete Dynamics in Nature and Society*, Vol. 2015, Art. ID 350308, 12 p.
24. Choi B.C., Yoon S.J., Maximizing the number of just-in-time jobs in flow-shop scheduling. *J. of Sched.*, 2007, vol. 10, no. 4–5, pp. 237–243.
25. Chung D.Y., Choi B.C., Just-in-time scheduling with competing agents. *Korean Oper. Res. Manag. Sci.*, 2012, vol. 37, pp. 19–28.
26. Dhamala T.N., Khadka S.R., A review on sequencing approaches for mixed-model just-in-time production systems. *Iranian J. of Optim.*, 2009, vol. 1, pp. 266–290.

27. Drobouchevich I.G., Sidney J.B., Minimization of earliness, tardiness and due date penalties on uniform parallel machines with identical jobs. *Comp. and Oper. Res.*, 2012, vol. 39, no. 9, pp. 1919–1926.
28. Elysi A., Salmasi N., Stochastic scheduling with minimizing the number of tardy jobs using change constrained programming. *Math. and Comp. Modell.*, 2013, vol. 57, no. 5–6, pp. 1154–1164.
29. Elysi A., Salmasi N., Stochastic flow-shop scheduling with minimizing the expected number of tardy jobs. *The Int. J. of Adv. Manuf. Technol.*, 2013, vol. 66, no. 1–4, pp. 337–346.
30. Garey M.R., Johnson D.S., *Computers and Intractability. The Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, San Francisco, 1979.
31. Garey M.R., Tarjan R.E., Wilfong G.T., One-processor scheduling with symmetric earliness and tardiness penalties. *Math. of Oper. Res.*, 1988, vol. 13, pp. 330–348.
32. Gerstl E., Mosheiov G., Due-window assignment problems with unit-time jobs. *Appl. Math. and Comput.*, 2013, vol. 220, pp. 487–495.
33. Gerstl E., Mosheiov G., Minmax weighted earliness-tardiness with identical processing times and two competing agents. *Comp. and Ind. Eng.*, 2017, vol. 107, pp. 171–177.
34. Giordano F., Schiraldi M.M., On Just-In-Time production leveling. In: *Operations Management, INTECH*, pp. 141–162, 2015.
35. Golhar D.Y., Stamm C.L., The just-in-time philosophy: a literature review. *The Int. J. of Prod. Res.*, 1991, vol. 29, no. 4, pp. 657–676.
36. Gordon V.S., Proth J.M., Chu C., A survey of the state-of-the-art of common due-date assignment and scheduling research. *EJOR*, 2002, vol. 139, no. 1, pp. 1–25.
37. Gordon V.S., Proth J.-M., Strusevich V.A., Scheduling with due date assignment. In: Leung J.Y. (ed.), *Handbook of Scheduling: Algorithms, Models and Performance Analysis*. Chapman & Hall/CRC Press, Boca Raton, FL, USA, Chapter 10, 2004.
38. Groenevelt H., The just-in-time systems. In: *Handbook in Operations Research and Management Science*, 1993, vol. 4, pp. 629–670.
39. Gupta U.I., Lee D.T., Leung J.Y.-T., An optimal solution the the channel-assignment problem. *IEEE Trans. on Comp.*, 1979, vol. 28, pp. 807–810.
40. Hermelin D., Shabtay D., Talmon N., On the parameterized tractability of the just-in-time flow-shop scheduling problem. *Electr. prepr.*, 23 p., Sep. 13, 2017. <http://arxiv.org/abs/1709.04169> [cs.DS]
41. Hiraishi K., Levner E., Vlach M., Scheduling of parallel identical machines to maximize the weighted number of just-in-time jobs. *Comp. and Oper. Res.*, 2002, vol. 29, no. 7, pp. 841–848.
42. Hoogeveen J.A., Multicriteria scheduling. *EJOR*, 2005, vol. 167, no. 3, pp. 592–623.
43. Huang P.Y., Rees L.P., Taylor III B.W., A simulation analysis of the Japanese just-in-time techniques (with kanbans) for a multiline, multistage production systems. *Dec. Sci.*, 1983, vol. 14, no. 3, pp. 326–344.
44. Janiak A., Januszkiewicz R., Scheduling of unit-time jobs distinct due windows on parallel processors. In: *Proc. 9th IFAC Workshop on Intell. Manuf. Syst.*, pp. 118–121, 2008.
45. Janiak A., Janiak W., Krysiak T., Kwiatkowski T., A survey on scheduling problems with due windows. *EJOR*, 2015, vol. 242, no. 2, pp. 347–357.
46. Josefowska J., *Just-in-Time Scheduling: Models and Algorithms for Computer and Manufacturing Systems*. Springer, 2010.
47. Kacem I., Fully polynomial time approximation scheme for the total weighted tardiness with a common due date. *Discr. Appl. Math.*, 2010, vol. 158, no. 9, pp. 1035–1040.
48. Kayvanfar V., Komaki G.H.M., Aalaei A., Zandieh M., Minimizing total tardiness and earliness on unrelated parallel machines with controllable processing times. *Comput. Oper. Res.*, 2014, vol. 41, pp. 31–43.

49. Kellerer H., Rustogi K., Strusevich V.A., A fast FPTAS for single machine scheduling problem of minimizing total weighted earliness and tardiness about large common due date. *Omega*, 2018 (in Press).
50. Keshavarz T., Savelsbergh M., Salmasi N., A branch-and-bound algorithm for the single machine sequence-dependent group scheduling problem with earliness and tardiness penalties. *App. Math. Modell.*, 2015, vol. 39, no. 20, pp. 6410–6424.
51. Kolen A.W.J., Lenstra J.K., Papadimitriou Ch.H., Spieksma F.C.R., Interval scheduling: a survey. *Naval Res. Logist.*, 2007, vol. 54, no. 5, pp. 530–543.
52. Kootanaee A.J., Babu K.N., Talari H.F., Just-in-time manufacturing system: from introduction to implement. *Int. J. of Economics, Business and Finance*, 2013, vol. 1, no. 2, pp. 7–25.
53. Kovalyov M.Y., Kubiak W., A fully polynomial approximation scheme for the weighted earliness-tardiness problem. *Oper. Res.*, 1999, vol. 47, no. 5, pp. 757–761.
54. Kovalyov M.Y., Ng C.T., Cheng T.C.E., Fixed interval scheduling: models, applications, computational complexity and algorithms. *EJOR*, 2007, vol. 178, no. 2, pp. 331–342.
55. Krieg G.N., Kuhn H., A decomposition method for multi-product kanban systems with setup times and lost sales. *IEE Transactions*, 2002, vol. 34, pp. 613–625.
56. Kroon K.G., Salomon M., van Wassenhove L.N., Exact and approximation algorithms for the tactical fixed interval scheduling problem. *Oper. Res.*, 1997, vol. 45, pp. 624–638.
57. Krumke S.O., Thielen C., Westphal S., Interval scheduling on related machines. *Comp. and Oper. Res.*, 2011, vol. 38, pp. 1836–1844.
58. Kubiak W., Sethi S., Optimal just-in-time schedules for flexible transfer lines. *Int. J. of Flexible Manuf. Syst.*, 1994, vol. 6, no. 2, pp. 137–154.
59. Kubiak W., Steiner G., Yeomans J.S., Optimal level schedules for mixed-model, multi level just-in-time assembly systems. *Annals of Oper. Res.*, 1997, vol. 69, pp. 241–259.
60. Laguna M., Velarde J.L.G., A search heuristic for just-in-time scheduling in parallel machines. *J. of Intell. Manuf.*, 1991, vol. 2, no. 4, pp. 253–260.
61. Lauff V., Werner F., Scheduling with common due date, earliness and tardiness penalties for multimachine problems: a survey. *Mathematical and Computer Modelling*, 2004, vol. 40, no. 5–6, pp. 637–655.
62. Lawson M., Ogden R., Goodier C., *Design in Modular Construction*. CRC Press, 2014.
63. Левин М.Ш. Техно-рабочий проект подсистемы формирования заводского производства на основе поточных графиков строительства. “Моспроект-1”, ГлавАПУ, Москва, 1982.
64. Levin M.Sh., *Modular System Design and Evaluation*. Springer, 2015.
65. Levin M.Sh., Towards balanced clustering - part 1 (preliminaries). *Elec. prepr.*, 21 p., Jun. 9, 2017. <http://arxiv.org/abs/1706.03065> [cs.DS]
66. Levin M.Sh., On balanced clustering (indices, models, examples). *J. of Commun. Technol. and Electronics*, 2017, vol. 62, no. 12, pp. 1506–1515.
67. Levin M.Sh., Time-interval balanced clustering in multi-processor scheduling of composite modular jobs (preliminary description) *Elec. prepr.*, 37 p., Nov. 11, 2018. <http://arxiv.org/abs/1811.04458> [cs.AI]
68. Leyvand Y., Shabtay D., Steiner G., Yedidsion L., Just-in-time scheduling with controllable processing times on parallel machines. *J. Comb. Optim.*, 2010, vol. 19, no. 3, pp. 347–368.
69. Li L., Fonesca D.J., Chen D.S., Earliness-tardiness production planning for just in time manufacturing: A unifying approach by goal programming. *EJOR*, 2006, vol. 175, no. 1, pp. 508–515.
70. Liu K., Abu-Ghazaleh N., Kang K.D., JiTS: Just-in-time scheduling for real-time sensor data dissemination. *Elec. prepr.*, 25 p., Aug. 16, 2006. <http://arxiv.org/abs/0608069> [cs.NI]

71. Максименко Б.А., Метод каталожных объемно-планировочных элементов как новый этап развития крупнопанельного домостроения. Передовой опыт в строительстве Москвы, 1982, вып. 2, С. 7–9.
72. Martello S., Toth P., A heuristic approach to the bus driver scheduling problem. *EJOR*, 1986, vol. 24, pp. 106–117.
73. Meissner S., Controlling just-in-sequence flow-production. *Logistics Research*, 2010, vol. 2, pp. 45–53.
74. McGarry M.P., Reisslein M., Colbourn C.J., Maier M., Auzada F., Scheutzow M., Just-in-time scheduling for multichannel EPONs. *IEEE Trans. of Lightwave Technology*, 2008, vol. 26, no. 10, pp. 1204–1216.
75. McMullen P.R., JIT sequencing for mixed-model assembly lines with setups using Tabu Search. *Production Planning and Control*, 1998, vol. 9, no. 5, pp. 504–510.
76. M'Hallah R., Bulfin R.L., Minimizing the weighted number of tardy jobs on parallel processors. *EJOR*, 2005, vol. 160, pp. 471–484.
77. Miltenburg J., Level schedules for mixed-model assembly lines in just-in-time production systems. *Manag. Sci.*, 1989, vol. 35, no. 2, pp. 192–207.
78. Miltenburg J., Steiner G., Yeomans S., A dynamic programming algorithm for scheduling mixed-model, just-in-time production systems. *Math. and Comp. Modell.*, 1990, vol. 13, no. 3, pp. 57–66.
79. Mittenthal J., Raghavachari M., Stochastic single machine scheduling with quadratic early-tardy penalties. *Oper. Res.*, 1993, vol. 41, pp. 786–796.
80. Monette J.N., Deville Y., Van Hentenryck P., Just-in-time scheduling with constrained programming. In: *Nineteenth Int. Conf. on Automated Planning and Scheduling ICAPS 2009*, pp. 241–248, 2009.
81. Monden Y., *Toyota Production System: An Integrated Approach to Just-In-Time*. 4th ed., CRC Press, 2011.
82. Mooeni F., Sanchez S.M., Vakharia A.J., A robust design methodology for Kanban system design. *Int. J. of Prod. Res.*, 1997, vol. 35, pp. 2821–2838.
83. Moore J.M., An n job, one machine sequencing algorithm for minimizing the number of late jobs. *Manag. Sci.*, 1968, vol. 15, pp. 102–109.
84. Mosheiov G., Oron D., A multi-criteria scheduling with due-window assignment problem. *Math. and Comp. Modell.*, 2008, vol. 48, no. 5, pp. 898–907.
85. Mosheiov G., Sarig A., Minmax scheduling problems with a common due-window. *Comput. Oper. Res.*, 2009, vol. 36, no. 6, pp. 1886–1892.
86. Mosheiov G., Sarig A., Scheduling a maintenance activity and due-window assignment on a single machine. *Comput. Oper. Res.*, 2009, vol. 36, no. 9, pp. 2541–2545.
87. Mosheiov G., Sarig A., Scheduling with a common due-window: polynomial solvable cases. *Inf. Sci.*, 2010, vol. 180, no. 8, pp. 1492–1505.
88. Mullens M.A., *Factory Design for Modular Homebuilding*. Constructability Press, 2011.
89. Nori V.S., Sarker B.R., Cyclic scheduling for a multi-product, single-facility production system operating under a just-in-time delivery policy. *J. of the ORS*, 1996, vol. 47, no. 7, pp. 930–935.
90. Papadimitriou Ch.H., Steiglitz K., *Combinatorial Optimization: Algorithms and Complexity*. Dover, 2013.
91. Pereira J., Vasquez O.C., The single machine weighted mean squared deviation problem. *EJOR*, 2017, vol. 261, no. 2, pp. 515–529.
92. Pheng L.S., Jayawickrama T.S., Just-in-Time Management of a Building Project in the Middle-East. In: *Rios-Mercado R., Rios-Solis Y.A. (eds), Just-in-Time Systems*. Springer, pp. 261–286, 2012.
93. Pinedo M.L., *Scheduling: Theory, Algorithms, and Systems*. 5th ed., Springer, New York, 2016.

94. Rios-Mercado R., Rios-Solis Y.A. (eds), *Just-in-Time Systems*. Springer, 2012.
95. Roach A., Nagi R., A hybrid GA-SA algorithm for just-in-time scheduling of multi-level assemblies. *Comp. and Oper. Res.*, 1996, vol. 30, no. 4, pp. 1047–1060.
96. Рочегов А., Вайсман Л. Новая система московского крупно-панельного домостроения. *Строительство и архитектура Москвы*, 1980, вып. 5, С. 4–7.
97. Van Rompaey K., Bolsens I., De Man H., Just in time scheduling. In: *Proc. IEEE 1992 Int. Conf. on Computer Design: VLSI in Computers and Processors ICCD'92*, pp. 295–300, 1992.
98. Ronconi D.P., Kawamura M.S., The single machine earliness and tardiness scheduling problem: lower bounds and a branch-and-bound algorithm. *Comput. and Appl. Math.*, 2010, vol. 29, no. 2, pp. 107–124.
99. Ronconi D.P., Birgin E.G., Mixed-Integer Programming Models for Flowshop Scheduling Problems Minimizing the Total Earliness and Tardiness. In: Rios-Mercado R., Rios-Solis Y.A. (eds), *Just-in-Time Systems*. Springer, pp. 91–105, 2012.
100. Rosa B.F., Souza M.J.F., de Souza S.R., de Franca Filho M.F., Ales Z., Machelon P.Y.P., Algorithms for job scheduling problems with distinct time windows and general earliness/tardiness penalties. *Comp. and Oper. Res.*, 2017, vol. 81, pp. 203–215.
101. Sendil Kumar C., Panneerselvam R., Literature review of JIT-KANBAN system. *Int. J. of Advanced Manuf. Technol.*, 2007, pp. 393–408.
102. Seo D.K., Klein C.M., Jang W., Single machine stochastic scheduling to minimize the expected number of tardy jobs using mathematical programming models. *Comp. and Ind. Eng.*, 2005, vol. 48, pp. 153–161.
103. Shabtay D., The just-in-time scheduling problem in flow shop scheduling systems. *EJOR*, 2012, vol. 216, no. 3, pp. 521–532.
104. Shabtay D., Steiner G., Scheduling to maximize the number of just-in-time jobs: a survey. In: Rios-Mercado R.Z., Rios-Solis Y.A. (eds), *Just-In-Time Systems*. Springer, pp. 3–20, 2012.
105. Soukhal A., Toung N.H., Just-in-Time Scheduling with Equal-Size Jobs. In: Rios-Mercado R., Rios-Solis Y.A. (eds), *Just-in-Time Systems*. Springer, pp. 107–146, 2012.
106. Sourd F., Kedad-Sidhoum S., A faster branch-and-bound algorithm for the earliness–tardiness scheduling problem. *J. of Scheduling*, 2008, vol. 11, no. 1, pp. 49–58.
107. Sung S.C., Vlach M., Maximizing weighted number of just-in-time jobs on unrelated parallel machines. *J. of Sched.*, 2005, vol. 8, no. 5, pp. 453–460.
108. Tanaka S., An exact algorithm for the single-machine earliness-tardiness scheduling problem. In: Rios-Mercado R., Rios-Solis Y.A. (eds), *Just-in-Time Systems*. Springer, pp. 21–40, 2012.
109. Tang H.Y., Zhao C.L., Cheng C.D., Single machine stochastic JIT scheduling problem subject to machine breakdowns. *Science in China Series A: Mathematics*, 2008, vol. 51, no. 2, pp. 273–292.
110. Tkindt V., Billaut J.-C., *Multicriteria Scheduling: Theory, Models and Algorithms*. 2nd ed., Springer, 2006.
111. Trietsch D., Baker K., Minimizing the number of tardy jobs with stochastically-ordered processing times. *J. of Scheduling*, 2008, vol. 11, no. 1, pp. 71–73.
112. Tuong N.H., Soukhal A., Some new polynomial cases in just-in-time scheduling problems with multiple due dates. In: *Proc. of the Sixth Int. Conf. on Research, Innovation and Vision for the Future in Computing and Communications Technologies (RIVFT08)*, pp. 36–41, 2008.
113. Tuong N.H., Soukhal A., Polynomial cases and PTAS for Just-in-Time scheduling on parallel machines around a common due-date. In: *11-th Int. Workshop on Project Management and Scheduling (PMS08)*. pp. 152–155, 2008.
114. Tuong N.H., Soukhal A., Due dates assignment and JIT scheduling with equal-size jobs. *EJOR*, 2010, vol. 205, no. 2, pp. 280–289.

115. Vallada E., Ruiz R., Scheduling Unrelated Parallel Machines with Sequence Dependent Setup Times and Weighted Earliness Tardiness Minimization. In: Rios-Mercado R., Rios-Solis Y.A. (eds), Just-in-Time Systems. Springer, pp. 67–90, 2012.
116. Valente J.M.S., Alves R.A.F.S., Efficient polynomial algorithms for special cases of weighted early/tardy scheduling with release dates and a common due date. *Pesquisa Operacional*, 2003, vol. 23, no. 3, pp. 443–456.
117. Valente J.M.S., Goncalves J.F., A genetic algorithm approach for the single machine scheduling problem with linear earliness and quadratic tardiness penalties. *Comp. and Oper. Res.*, 2009, vol. 36, no. 10, pp. 2707–2715.
118. Wang X.-Y., Wang M.-Z., Single machine common flow allowance scheduling with a rate-modifying activity. *Comp. and Ind. Eng.*, 2010, vol. 59, no. 4, pp. 898–902.
119. Wang D., Yin Y., Cheng T.C.E., A bicriterion approach to common flow allowances due window assignment and scheduling with controllable processing times. *Naval Res. Logist.*, 2017, vol. 64, no. 1, pp. 41–63.
120. Wei J.Y., McFarland R.I., Just-in-time signaling for WDM optical burst switching networks. *J. of Lightwave Technology*, 2000, vol. 18, no. 12, pp. 2019–2037.
121. Woeginger G.J., On-line scheduling of jobs with fixed start and end times. *Theor. Comp. Sci.*, 1994, vol. 130, pp. 5–16.
122. Yeung W.K., Oguz C., Cheng T.C.E., Single-machine scheduling with a common due window. *Comp. and Oper. Res.*, 2001, vol. 28, pp. 157–175.
123. Yin Y., Wang D.-J., Cheng T.C.E., Wu C.-C., Bi-criterion single machine scheduling and due-window assignment with common flow allowances and resource-dependent processing times. *J. of the ORS*, 2016, vol. 67, no. 9, pp. 1169–1183.
124. Yuce B., Fruggiero F., Packianather M.S., Pham D.T., Mastrocinque E., Lambiase A., Fera M., Hybrid Genetic Bees Algorithm applied to single machine scheduling with earliness and tardiness penalties. *Comp. and Ind. Eng.*, 2017, vol. 113, pp. 842–858.

Interval balanced multiprocessor scheduling of modular jobs

Levin M.Sh.

This article addresses multiprocessor scheduling of modular jobs while taking into account a balance requirements for each time interval. First, brief description of just-in-time planning approach is described. Further, the problem statement of the examined scheduling problem is suggested. A numerical example from planning in house-building illustrates the proposed problem and heuristic solving scheme. This scheme involves the following: (i) clustering of the initial set of building block/details to obtain eight typical group which correspond to manufacturing conveyers, (ii) generation of an initial plan for building assembly, (iii) detection of time intervals with disbalance by a basic building block/detail, (iv) modification of the plan (by multiple choice problem). A possible application of the proposed problem in information transmission is very briefly described as well.

KEYWORDS: multi-processor scheduling, combinatorial optimization, time-interval balancing, balanced clustering, home-building, heuristics