

УДК 519.854.2

ОБ ОПТИМАЛЬНЫХ ЦИКЛАХ ДЛЯ РЕГУЛЯРНОЙ  
СБАЛАНСИРОВАННОЙ РОБОТИЗИРОВАННОЙ ЯЧЕЙКИ  
БЕЗ ЗАДЕРЖЕК \*)

*С. В. Павлов*

**Аннотация.** Рассматривается задача построения оптимальных циклических расписаний для роботизированной ячейки, обслуживаемой одним роботом, со строгой стратегией разгрузки в машинной среде flow shop и с критерием максимум производительности ячейки. Идентичные детали проходят обработку на каждой машине роботизированной ячейки одинаковое количество времени. На основе анализа возможных циклов (вариация циклических расписаний) находится оптимальное решение задачи с 5 машинами. Найденное оптимальное решение согласуется с известными гипотезами о структуре оптимальных расписаний для данной задачи.

**Ключевые слова:** роботизированная ячейка, flow shop, циклическое расписание, гипотеза Агнетиса.

**Введение**

В работе рассматриваются циклические расписания для роботизированной ячейки, обслуживаемой одним роботом. Идентичные детали проходят машины в среде flow shop таким образом, что их обработка выполняется без задержек, т. е. сразу после окончания обработки на какой-то машине деталь должна освободить машину и начать перемещение на следующую машину. Перемещение деталей между машинами осуществляется (за фиксированное время) роботом, который может переносить только одну деталь.

Как утверждается в [7], первые постановки задач о роботизированных ячейках относятся к 60-м годам, когда группой белорусских математиков был опубликован ряд работ по теме циклических процессов в сборочных линиях с использованием устройств для перемещения. В этих

---

\*) Исследование выполнено при финансовой поддержке Российского фонда фундаментальных исследований (проект 12-01-00184-а) и ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009–2013 гг. (гос. контракт № 14.740.11.0362).

исследованиях сформулирована задача о циклическом роботизированном конвейерном производстве, а также представлен метод запрещённых интервалов для решения этой задачи.

За рубежом постановки задач о роботизированных ячейках получили популярность позднее в связи с развитием производств, которые не требовали участия человека. Обширные обзоры, а также классификации задач могут быть найдены в [4, 5].

Постановки задач о роботизированных ячейках можно разделить на три группы в зависимости от того, какая стратегия разгрузки машин рассматривается: *неограниченная*, когда деталь может находиться на машине неограниченно долго после завершения операции, *интервальная*, когда допустимое время нахождения детали на машине лежит в некотором интервале, и *строгая*, когда деталь необходимо снять с машины строго в момент завершения обработки. В данной работе рассматривается последний вариант, в англоязычной литературе называемый *no-wait*.

В процессе выполнения циклического расписания робот движется по замкнутому маршруту, выполняя действия по переносу деталей. Последовательность действий робота в рамках одного периода расписания называется *циклом*. У циклов есть *степень*, которая определяется как количество деталей, которые начали свою обработку за время выполнения цикла. Циклы степени  $k$  называют *k-циклами*.

Известно, что для некоторых ограниченных классов расписаний за полиномиальное время можно получить оптимальное решение задачи. Например, можно найти наилучший 1-цикл для широкого класса задач о роботизированной ячейке [6, 8]. Кроме того, в [3] представлен алгоритм, позволяющий определить наилучший  $k$ -цикл для фиксированного  $k$  для задачи о роботизированной ячейке со строгой стратегией разгрузки. Тем не менее открытым остаётся вопрос о верхней границе степени оптимального цикла даже для простых постановок задачи [2], например, для рассматриваемой в настоящей работе.

В [1] высказана гипотеза (впоследствии названная «гипотезой Агнетиса») о том, что степень оптимального цикла в роботизированной ячейке со строгой стратегией разгрузки не превосходит  $m - 1$ , где  $m$  — число машин, и приведено доказательство для  $m = 2$  и 3. Позднее в [9] представлена гипотеза о структуре оптимальных расписаний в случае регулярной (расстояние между соседними машинами одинаковое) сбалансированной (время выполнения всех операций одинаковое) роботизированной ячейки со строгой стратегией разгрузки. Эта гипотеза согласуется с гипотезой Агнетиса и более детально представляет точное решение рас-

смаатриваемой задачи.

Как показывает анализ, проведённый в упомянутых выше статьях, для рассматриваемой постановки задачи вид оптимального решения существенно зависит от времени выполнения операции  $p$ , времени перемещения робота между соседними машинами  $\delta$  и числа машин  $m$ . В частности, важную роль играют интервалы значений параметра  $p$ , кратные  $4\delta$ , поскольку при пересечении параметром  $p$  границ этих интервалов изменяется множество допустимых решений. Случай  $p < 4\delta$  исследован. В этом случае оптимальным решением является «восходящий» 1-цикл, в котором просто последовательно выполняются операции одной работы. Следующий интервал значений параметра  $[4\delta, 8\delta)$  представляет собой первый не до конца исследованный случай. Гипотеза о том, что в этом случае оптимальным является 2-цикл, доказана ранее для всех значений  $m$ , за исключением  $m = 5$ . Именно случай  $m = 5$  рассматривается в настоящей статье. Упомянутая выше гипотеза подтверждается и в этом случае, таким образом, вопрос об оптимальном решении задачи о регулярной сбалансированной ячейке с запретом на задержки между операциями одной работы полностью закрыт для случая  $p \in [4\delta, 8\delta)$ .

В разд. 1 сформулирована математическая постановка задачи и даны определения. В разд. 2 доказан основной результат работы. В разд. 3 кратко сформулированы вопросы, представляющие интерес для дальнейшего исследования.

## 1. Постановка задачи

*Роботизированная ячейка* представляет собой  $m + 2$  машины  $\{M_0, \dots, M_{m+1}\}$ , на которых выполняются идентичные работы, и одного робота. Под *работой* понимается обработка некоторой детали. Каждая работа состоит из  $m$  операций  $\{O_j\}$ , выполняемых в порядке  $O_1 \rightarrow O_2 \rightarrow \dots \rightarrow O_m$  на машинах  $M_1, \dots, M_m$  соответственно. Машины  $M_0$  и  $M_{m+1}$  являются выделенными и представляют собой *вход* и *выход* роботизированной ячейки соответственно. На этих машинах не выполняются операции, и их можно рассматривать как склады деталей, ещё не поступивших в обработку и завершивших свою обработку соответственно.

**Определение 1.** *Сбалансированной* называется роботизированная ячейка, в которой время выполнения каждой операции равно  $p \geq 0$ , т. е. время выполнения всех операций одинаковое.

Время перемещения робота между машинами будем считать независимым от направления передвижения и того, переносит ли робот деталь. Время, за которое робот перемещается между машинами  $M_i$  и  $M_j$ ,

$i, j \in \{0, \dots, m+1\}$ , обозначим через  $\delta_i^j$ .

**Определение 2.** *Регулярной* называется роботизированная ячейка такая, что  $\delta_i^j = |i - j|\delta$ , где  $\delta \geq 0$  — константа, равная времени перемещения робота между двумя соседними машинами.

В случае, когда  $\delta = 0$ , решение задачи известно [4]. Поэтому будем полагать параметры  $p$  и  $\delta$  нормированными таким образом, что  $\delta = 1$ .

Роботизированная ячейка считается регулярной и сбалансированной. Также будем считать, что робот разгружает и загружает машины за пренебрежимо малое время. Таким образом, индивидуальная задача задаётся двумя параметрами: числом машин  $m$  и временем выполнения операции  $p$ .

Для того чтобы формально задать целевую функцию, нам потребуется ввести несколько определений.

**Определение 3.** *Состоянием* роботизированной ячейки называется кортеж  $\Omega \in \{\mathbb{R}^+ \cup \{\nabla\}\}^m \times [0, (m+1)] \times \{0, 1\}$ .

Рассмотрим состояние  $\Omega = \langle \omega_1, \dots, \omega_{m+2} \rangle$ . Первые  $m$  значений  $\omega_1, \dots, \omega_m$  определяют, есть ли на машине  $M_i$  ( $i = 1, \dots, m$ ) деталь. Если деталь есть, то  $\omega_i$  равно времени, которое деталь должна провести на машине для завершения операции. Если детали на машине  $M_i$  нет, то  $\omega_i = \nabla$ ,  $\nabla \notin \mathbb{R}$ . Компонента  $\omega_{m+1}$  равна времени, которое потребуется роботу, чтобы достичь машины  $M_0$  из текущего положения. Равенство  $\omega_{m+2} = 0$  означает, что робот свободен, а  $\omega_{m+2} = 1$  — что робот «держит деталь».

Например, состояние  $\Omega_{\nabla} = \langle \nabla, \nabla, \nabla, \nabla, \nabla, 0, 0 \rangle$  означает, что все машины пусты, а робот находится на машине  $M_0$ . Это состояние будем называть также *состоянием пустой ячейки*.

**Определение 4.** Под *действием*  $A_i$ ,  $i \in \{0, 1, \dots, m\}$ , понимается следующая последовательность элементарных действий, выполняемых роботом: снятие детали с машины  $M_i$ , перенос детали от машины  $M_i$  до машины  $M_{i+1}$  и загрузка детали на машину  $M_{i+1}$ .

Для удобной записи последовательности действий будем использовать знак  $\prod$  наравне с перечислением действий, а для операции конкатенации последовательностей — символ  $*$ . Будем обозначать  $j$ -е по счёту действие в последовательности  $\mathcal{A}$  через  $A^j$ . Количество действий  $K$  в конечной последовательности  $\mathcal{A} = A^1 * \dots * A^K$  будем обозначать через  $|\mathcal{A}|$ . Определим вспомогательную функцию  $\mu(j)$  такую, что  $A^j = A_{\mu(j)}$  (т.е.  $\mu(j)$  определяет номер машины, на которой начинается  $j$ -е действие),  $j \in \{1, \dots, |\mathcal{A}|\}$ .

**Определение 5.** Будем называть  $k$ -циклом такую последовательность действий  $\mathcal{A}$ , которая удовлетворяет следующим условиям.

(i) Между двумя последовательными (в циклическом смысле) действиями  $A_i$  ( $i = 0, \dots, m-1$ ) в  $\mathcal{A}$  содержится в точности одно вхождение  $A_{i+1}$ .

(ii) В  $\mathcal{A}$  не существует интервалов вида  $A_i * \dots * A_j * \dots * A_{j+1} * \dots * A_{i+1}$ , в которых действие  $A_{i+1}$  — ближайшее справа (в циклическом смысле) к действию  $A_i$ .

Из условия (i) вытекает симметричное ему: между двумя последовательными (в циклическом смысле) действиями  $A_i$  ( $i = 1, \dots, m$ ) в  $\mathcal{A}$  содержится в точности одно вхождение  $A_{i-1}$ . Из (i) также следует, что число вхождений действия  $A_i$  в последовательность  $\mathcal{A}$  равно  $k$  для всех  $i = 0, \dots, m$ . Число  $k$  называют *степенью* цикла.

Заметим, что если  $\mathcal{A}$  —  $k$ -цикл, то  $|\mathcal{A}| = k(m+1)$ .

Для заданного  $k$ -цикла  $\mathcal{A}$  пусть

$$J_b = \{j \in \{1, \dots, k(m+1)\} \mid \nexists j' < j, \mu(j') + 1 = \mu(j)\}$$

— множество номеров действий  $A^j = A_i$  ( $i \neq 0$ ) в  $\mathcal{A}$ , которым не предшествует  $A_{i-1}$ , и пусть

$$J_e = \{j \in \{1, \dots, k(m+1)\} \mid \nexists j' > j : \mu(j) + 1 = \mu(j')\}$$

— множество номеров действий  $A^j = A_i$  ( $i \neq m$ ) в  $\mathcal{A}$ , за которыми не следует  $A_{i+1}$ . Положим  $M' = \{\mu(j) \mid j \in J_b\}$ ,  $M'' = \{\mu(j) + 1 \mid j \in J_e\}$ . Из определения 5 следует, что  $M' = M''$ .

**Определение 6.** Циклическое расписание есть пара  $S = \{\mathcal{A}, I\}$ , где  $\mathcal{A}$  —  $k$ -цикл,  $I = \{t_1, t_2, \dots, t_{k(m+1)}\}$  — моменты времени начала выполнения действий из цикла (т. е.  $A^j$  начинается в момент времени  $t_j$ ). Без ограничения общности можно считать, что  $t_1 = 0$ , а  $\mathcal{A}$  начинается с  $A_0$ , что и будет подразумеваться в дальнейшем.

**Определение 7.** Циклическое расписание *допустимо*, если выполняются условия

(a)  $t_{j_2} - t_{j_1} = p + 1$  для всех  $j_1, j_2 \in \{1, \dots, k(m+1)\}$  таких, что  $\mu(j_1) = i$ ,  $j_2 = \min\{j > j_1 \mid \mu(j) = i + 1\}$ ;

(b)  $t_{j+1} - t_j - 1 \geq |\mu(j) + 1 - \mu(j+1)|$  для всех  $j \in \{1, \dots, k(m+1) - 1\}$ ;

(c) существует константа  $T$  (называемая *периодом циклического расписания*) такая, что для всех  $i \in M'$  справедливо равенство

$$T = t_{j_1} + p + 1 - t_{j_2},$$

где  $j_1 \in J_e, j_2 \in J_b$  и  $\mu(j_1) + 1 = \mu(j_2) = i$ ;

(d) выполняется неравенство  $T - t_{k(m+1)} \geq \mu(k(m+1)) + 2$ .

Заметим, что условия на существование константы  $T$  задают требования по-wait для тех пар действий, которые не содержатся целиком в одном цикле.

**Утверждение 1.** Если циклическое расписание допустимо, то состояния ячейки в момент времени 0 и  $T$  совпадают.

ДОКАЗАТЕЛЬСТВО. Обозначим через  $\Omega^0$  и  $\Omega^T$  состояния ячейки в моменты времени 0 и  $T$  соответственно. Сначала покажем, что  $\omega_i^0 = \omega_i^T$  для всех  $i \in \{1, \dots, m\}$ .

Действительно,  $\omega_i^0 = \omega_i^T = \nabla$  для  $i \notin M'$  по определению множества  $M'$ . Для  $i \in M'$  существуют  $j_1 \in J_e, j_2 \in J_b$  и  $\mu(j_1) + 1 = \mu(j_2) = i$ , из чего следует, что  $\omega_i^T = p + 1 - (T - t_{j_1}) = t_{j_2} = \omega_i^0$ .

Нетрудно заметить, что  $\omega_{m+1}^0 = \omega_{m+2}^0 = 0$ , поскольку цикл начинается с действия  $A_0$ . Так как

$$T - t_{k(m+1)} - 1 \geq \mu(k(m+1)) + 1,$$

робот на момент времени  $T$  свободен и у него достаточно времени, чтобы после завершения последнего действия в цикле к моменту времени  $T$  достичь машины  $M_0$ . Поэтому можно считать, что  $\omega_{m+1}^T = \omega_{m+2}^T = 0$ . Утверждение 1 доказано.

Из утверждения 1 следует, что допустимое циклическое расписание можно повторять. В дальнейшем под *расписанием* будем понимать допустимое циклическое расписание.

**Определение 8.** *Производительностью* расписания называется отношение  $k/T$ , где  $k$  — степень цикла, а  $T$  — период.

Целевой функцией рассматриваемой задачи является производительность расписания, которую следует максимизировать.

Известно, что, имея некоторый  $k$ -цикл  $\mathcal{A}$  и заданное время операции  $p$ , за полиномиальное время можно либо построить допустимое циклическое расписание, выполняющее этот цикл и имеющее минимальный период, либо показать, что не существует допустимого расписания. Таким образом, в дальнейшем будем говорить об оптимальности циклов. Например, для  $m = 3$  можно определить 3-цикл  $A_0A_1A_0A_2A_1A_0A_3A_2A_1A_3A_2A_3$ , для которого можно построить допустимое циклическое расписание только при значении  $p \geq 8$ . Обзор известных результатов в области построения расписаний по циклу можно найти в [4].

Определим следующие  $k$ -циклы:

$$C_a(k) = \prod_{i=1}^k \left[ A_0 \prod_{j=1}^i A_{i-j+1} \right] * \prod_{i=k+1}^m \left[ \prod_{j=1}^k A_{i-j+1} \right] * \prod_{i=1}^{k-1} \left[ \prod_{j=1}^{k-i} A_{m-j+1} \right],$$

$$C_b(k) = \prod_{i=0}^{2(k-1)-m} \left[ \prod_{j=0}^i A_{i-j} * \prod_{j=0}^{2(k-1)-m-i} A_{m-j} \right] * \prod_{i=2k-m-1}^{k-1} \left[ \prod_{j=0}^i A_{i-j} \right] * \prod_{i=k}^m \left[ \prod_{j=0}^{k-1} A_{i-j} \right] * \prod_{i=k-2}^{2k-m-1} \left[ \prod_{j=0}^i A_{m-j} \right],$$

$$C_c(k) = \prod_{i=0}^{2k-m-1} \left[ \prod_{j=0}^i A_{i-j} * \prod_{j=0}^{2k-m-1-i} A_{m-j} \right] * \prod_{i=2k-m}^{k-1} \left[ \prod_{j=0}^i A_{i-j} \right] * \prod_{i=k}^m \left[ \prod_{j=0}^{k-1} A_{i-j} \right] * \prod_{i=k-2}^{2k-m} \left[ \prod_{j=0}^i A_{m-j} \right],$$

$$C_d = A_0 \prod_{i=0}^{m-1} A_{m-i}, \quad C_e = \prod_{i=0}^{m/2} A_{2i} * \prod_{i=0}^{m/2-1} A_{2i+1}.$$

Представим длительность операции  $p$  в виде

$$p = 4(k-1) + \Delta, \quad (1)$$

где  $k$  — некоторое натуральное число, а  $0 \leq \Delta < 4$ . Пусть

$$\theta = \frac{m^2}{(2p+4)(3mp-p+4m)}, \quad \rho = \frac{m(2pm+4m-1)}{3mp-2p+4m}.$$

В [9] для рассматриваемой в настоящей работе постановки задачи без уточнения параметров  $\theta$  и  $\rho$  сформулирована

**Гипотеза.** Оптимальным циклом для рассматриваемой задачи является один из пяти циклов, представленных выше (т. е. либо один из трёх  $k$ -циклов  $C_a(k)$ ,  $C_b(k)$ ,  $C_c(k)$ , где  $k$  определяется из соотношения (1), либо один из двух 1-циклов  $C_d$ ,  $C_e$ ). При этом выбор оптимального цикла зависит от соотношения между  $k$  и  $m$  и значения  $\Delta$ , определяемого из (1), и задаётся табл. 1.

Производительности циклов составляют [9]

$$C_a(k) \rightarrow k/[(m+k-1)p+2(m+2k-1)],$$

$$C_b(k) \rightarrow k/[(2m-k)p+4m], \quad C_c(k) \rightarrow k/[(2m-k-1)p+(4m-2)],$$

$$C_e \rightarrow 1 / \left( 2 \frac{m-1}{m} p + 4 \right), \quad C_d \rightarrow 1/(p+4).$$

Т а б л и ц а 1

$m$ — нечётное				
	$k \leq \frac{m-1}{2}$	$k = \frac{m+1}{2}$	$\frac{m+1}{2} < k \leq m-1$	$m \leq k$
$\Delta < 2$	$C_a(k)$	$C_a(k)$	$C_b(k)$	$C_d$
$\Delta \geq 2$	$C_a(k)$	$C_a(k)$	$C_b(k)$	$C_d$
$m$ — чётное				
	$k \leq \frac{m}{2}$	$\frac{m}{2} < k \leq \theta$	$\theta \leq k \leq m-1$	$m \leq k$
$\Delta < 2$	$C_a(k)$	$C_e$	$C_b(k)$	$C_d$
	$k \leq \frac{m}{2}$	$\frac{m}{2} < k \leq \rho$	$\rho \leq k \leq m-1$	$m \leq k$
$\Delta \geq 2$	$C_a(k)$	$C_e$	$C_c(k)$	$C_d$

Значения параметров  $\theta$ ,  $\rho$  вытекают из значений производительностей циклов  $C_b(k)$ ,  $C_c(k)$ ,  $C_e$ . Эта гипотеза доказана для  $m \leq 4$ , а также для произвольного числа машин  $m$  в случаях, когда (s1)  $p \in [4(k-1), 4k)$ ,  $k \leq \frac{m+2}{4}$ ; (s2)  $p < 4$  и (s3)  $p \geq 4(m-1)$ .

Из известных результатов следует, что при  $p \in [4, 8)$  гипотеза доказана для  $m \geq 6$  и  $m \leq 4$ . Открытым оставался случай  $m = 5$ , который рассмотрен в следующем разделе.

## 2. Оптимальное решение для $m = 5$ и $p \in [4, 8)$

Рассмотрим случай  $m = 5$  и  $p \in [4, 8)$ . Представим  $p$  в виде  $p = 4(2-1) + \Delta$ . Величина  $k$  равна 2, следовательно,  $k = \frac{m-1}{2}$ , и в соответствии с вышеприведённой гипотезой можно сформулировать

**Утверждение 2.** Для регулярной сбалансированной роботизированной ячейки с одним роботом в среде flow shop и запретом на задержки между операциями одной работы, если  $m = 5$ , а  $p \in [4, 8)$ , оптимальным является цикл  $C_a(2) = A_0A_1A_0A_2A_1A_3A_2A_4A_3A_5A_4A_5$ .



На рис. 1 изображено не расписание, а условная схема работы цикла. Стрелками от машины  $M_i$  до  $M_{i+1}$  изображены действия  $A_i$ , а линиями из точек — перемещения робота, когда он не выполняет действий. Показаны лишь те интервалы времени, когда робот перемещается, поэтому, например, первое действие  $A_1$  изображено как происходящее сразу после действия  $A_0$ , хотя между окончанием действия  $A_0$  и началом действия  $A_1$  должен стоять интервал времени длиной  $p$ . Карта Ганта для расписания будет зависеть от величины  $p$ . Заметим также, что цикл начинается и заканчивается в состоянии, когда машины не выполняют работ, а робот находится на машине  $M_0$ .

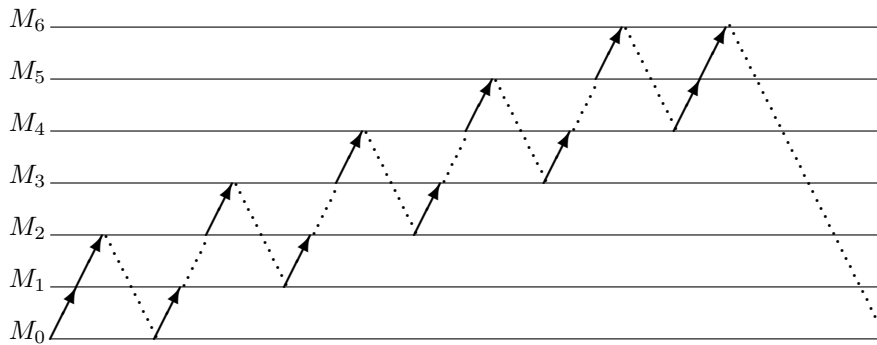


Рис. 1. Карта Ганта для цикла  $C_a(2)$

**ДОКАЗАТЕЛЬСТВО УТВЕРЖДЕНИЯ 2.** Сначала покажем, что вне зависимости от цикла ячейка будет в процессе выполнения любого допустимого периодического расписания проходить через состояние  $\Omega_{\nabla}$ , т. е. опустошаться. После этого можно рассмотреть только циклы, начинающиеся с пустой ячейки, и выбрать из них оптимальный. Приведём леммы, которые потребуются при дальнейшем доказательстве и верны в рассматриваемой постановке задачи.

**Лемма 1.** Если в некотором  $k$ -цикле  $\mathcal{A}$  встречается последовательность действий вида  $A_{j-1} * \dots * A_i A_j$ , где  $j \neq (i + 1)$ , то при значении  $p < 2|i - j|$  не существует допустимого циклического расписания, реализующего  $\mathcal{A}$ .

**ДОКАЗАТЕЛЬСТВО.** Рассмотрим случай, когда  $i > j$ . После завершения действия  $A_{j-1}$  роботу требуется не менее  $i - j + 1$  единиц времени, чтобы достичь машины  $M_i$  и выполнить действие  $A_i$ , а после этого — не менее  $i + 1 - j$  единиц времени, чтобы достичь машины  $M_j$ . В силу условия выполнения операций без задержек имеем  $p \geq 2(i + 1 - j)$ .

Рассмотрим случай, когда  $i < j$ . После завершения действия  $A_{j-1}$  роботу требуется не менее  $j - i$  единиц времени, чтобы достичь машины  $M_i$ , а после этого — не менее  $j - i$  единиц времени, чтобы выполнить действие  $A_i$  и достичь машины  $M_j$ . В силу условия выполнения операций без задержек имеем  $p \geq 2(j - i)$ . Лемма 1 доказана.

**Лемма 2** (приведена без доказательства в [9]). *Если  $p \in [4(k-1), 4k)$ , то в роботизированной ячейке может находиться в обработке одновременно не более  $k$  деталей.*

**ДОКАЗАТЕЛЬСТВО.** Допустим, что в роботизированной ячейке в процессе выполнения некоторого допустимого цикла  $C$  начинает обрабатываться деталь, в то время как на других машинах уже находится  $k - 1$  деталей. Рассмотрим в цикле  $C$  последовательность действий  $\mathcal{A} = A_0 A_{i_1} * \dots * A_{i_{k-1}} A_1$ , которая начинается с вхождения  $k$ -й детали, далее следуют  $k - 1$  действий по переносу уже обрабатываемых деталей, и наконец, снова совершается действие по переносу  $k$ -й детали. В силу условия `no-wait` определим нижнюю границу длительности операции:

$$p \geq \Sigma(\mathcal{A}) = \sum_{j=0}^{k-1} (1 + |i_j + 1 - i_{j+1}|),$$

где  $i_0 = 0, i_k = 1$ . Определим  $\Sigma_k = \min_{|\mathcal{A}|=(k+1)} \Sigma(\mathcal{A})$ .

Покажем, что для любого  $k \geq 1$  верно  $\Sigma_k \geq 4(k - 1)$ . При  $k = 1$  утверждение очевидно. Пусть для некоторого  $k$  утверждение доказано, положим  $k' = k + 1$ . Рассмотрим последовательность  $\mathcal{A} = A_0 A_{i_1} * \dots * A_{i_{k'-1}} A_1$  и  $j'$  такое, что  $i_{j'} = \max_{0 < j < k'} \{i_j \mid A_{i_j} \in \mathcal{A}\}$ . Обозначим  $i_{j'-1} = a$ ,  $i_{j'} = b$ ,  $i_{j'+1} = c$ . Из условия, что рассматриваемая последовательность действий является частью допустимого цикла, следует, что  $a + 1 < b$  и  $a + 1 \neq c$ , а также  $a \neq c$ .

Рассмотрим последовательность  $\mathcal{B} = \mathcal{A} \setminus A_{i_{j'}}$ , т. е. выкинем из  $\mathcal{A}$  действие  $A_{i_{j'}}$ . Тогда  $\mathcal{B}$  имеет длину  $|\mathcal{B}| = k' - 1$  и удовлетворяет условиям индукции, следовательно,

$$\Sigma(\mathcal{B}) \geq 4(k' - 2).$$

Рассмотрим в сумме  $\Sigma(\mathcal{A})$  пару слагаемых  $|a + 1 - b| + |b + 1 - c| + 2 = 2b - a - c + 2$ . Если  $a < c - 1$ , то

$$2b - a - c + 2 = |a + 1 - c| + 2b - 2c + 3 \geq |a + 1 - c| + 5.$$

Если  $a > c$ , то  $2b - a - c + 2 = |a + 1 - c| + 2b - 2a + 1 \geq |a + 1 - c| + 5$ . В обоих случаях  $\Sigma(\mathcal{A}) \geq \Sigma(\mathcal{B}) + 4$ . Отсюда следует, что  $\Sigma_{k'} \geq \Sigma_k + 4$ . Лемма 2 доказана.

Заметим, что по лемме 1 любой допустимый цикл не может содержать последовательности действий  $A_0A_4$  или  $A_0A_5$ . Тем самым достаточно рассмотреть случаи, когда цикл начинается с действий  $A_0A_1$ ,  $A_0A_2$  или  $A_0A_3$ . Для этих трёх случаев покажем, что в ходе выполнения цикла ячейка неизбежно достигнет состояния  $\Omega_{\nabla}$ , когда ни на одной из машин детали не обрабатываются, а робот находится на машине  $M_0$ .

СЛУЧАЙ 1. Если цикл начинается с действий  $A_0A_1$ , то в момент начала действия  $A_0$  ячейка в состоянии  $\Omega_{\nabla}$ . Действительно, предположим напротив, что в момент начала действия  $A_0$  в ячейке есть деталь, обрабатываемая на машине  $M_i$ ,  $i \in \{2, \dots, 5\}$ . Тогда с момента окончания действия  $A_0$  до начала действия  $A_1$  проходит время  $p$ , и деталь заведомо успеет завершить обработку и должна быть разгружена с машины, чего не происходит, и нарушается условие строгой стратегии разгрузки.

СЛУЧАЙ 2. Если цикл начинается с действий  $A_0A_2$ , то после выполнения действия  $A_0$  в ячейке оказывается две детали. По лемме 2 следует, что после начала цикла  $A_0A_2$  следуют действия  $A_1A_3A_2A_4A_3A_5$ . После завершения действия  $A_5$  одна из деталей покидает ячейку. По лемме 1 продолжения вида  $A_0A_4$  быть не может, поэтому после завершения действия  $A_5$  следует продолжение  $A_4A_5$ , и роботу остаётся только переместиться к машине  $M_0$ , что приводит ячейку в состояние  $\Omega_{\nabla}$ . Таким образом, полная последовательность действий до опустошения ячейки равна  $A_0A_2A_1A_3A_2A_4A_3A_5A_4A_5$ .

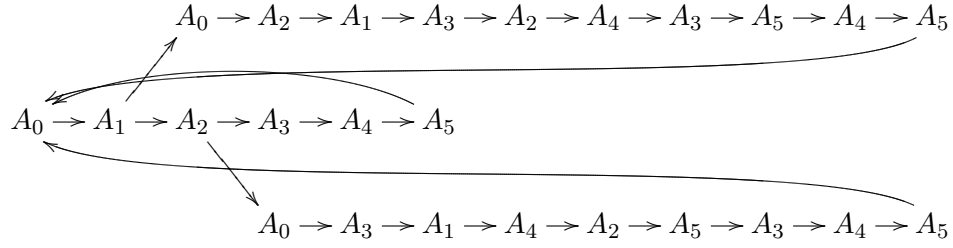
СЛУЧАЙ 3, когда цикл начинается с действий  $A_0A_3$ , разбирается аналогично предыдущему, но у него полная последовательность действий до опустошения ячейки равна  $A_0A_3A_1A_4A_2A_5A_3A_4A_5$ .

Таким образом, вне зависимости от начала цикла ячейка всё равно проходит через состояние  $\Omega_{\nabla}$ . Это позволяет в поисках оптимального решения рассмотреть только циклы, которые начинаются с этого состояния.

Структура орграфа, изображённого на рис. 2, непосредственно вытекает из предыдущих рассуждений. Таким образом, есть три варианта, какой может быть последовательность действий, начинающаяся и заканчивающаяся в состоянии  $\Omega_{\nabla}$ :

$$(d1) C_1 = A_0A_1A_2A_3A_4A_5,$$

$$(d2) C_2 = A_0A_1A_0A_2A_1A_3A_2A_4A_3A_5A_4A_5,$$



$$(d3) C_3 = A_0 A_1 A_2 A_0 A_3 A_1 A_4 A_2 A_5 A_3 A_4 A_5.$$

Рис. 2. Возможные варианты циклов

Отсюда следует, что любой допустимый цикл является некоторой комбинацией  $C_1$ ,  $C_2$  и  $C_3$  (например, это может быть  $C_1 * C_2 * C_3$ , или  $C_2 * C_2 * C_3$  или просто  $C_1$ ). Но известно [9], что поскольку каждый из  $C_i$ ,  $i \in \{1, 2, 3\}$ , заканчивается и начинается в состоянии  $\Omega_{\nabla}$ , если рассмотреть произвольный допустимый цикл  $C^*$ , то найдётся такая составляющая  $C_i$  из цикла  $C^*$ , что производительность  $C_i$  будет не меньше производительности  $C^*$ . Таким образом, для нахождения оптимального цикла достаточно сравнить между собой  $C_1, C_2, C_3$  и выбрать из них наилучший по производительности цикл.

Заметим, что цикл  $C_2$  совпадает с циклом  $C_a(2)$ , оптимальность которого доказывается. Цикл  $C_1$  называется *восходящим*, и его производительность равна  $1/(5p + 12)$ . Для цикла  $C_2$  производительность равна  $2/(6p + 16)$ . Для цикла  $C_3$  минимальный период расписания можно оценить следующим образом: с начала цикла до момента начала действия  $A_0$  второй работы должно пройти не менее  $2p + 6$  единиц времени. В силу условия выполнения операций без задержек время от начала действия  $A_0$  второй работы до завершения действия  $A_5$  второй работы равно  $5p + 6$ , а после завершения  $A_5$  роботу требуется 6 единиц времени, чтобы вернуться на машину  $M_0$ . Таким образом, расписание, реализующее цикл  $C_3$ , имеет период не менее чем  $7p + 18$ , а следовательно, производительность  $C_3$  не более чем  $2/(7p + 18)$ . Сравнивая производительности, получаем  $\frac{2}{6p+16} > \frac{2}{7p+18} > \frac{1}{5p+12}$ . Тем самым оптимальным является цикл  $C_2$ . Утверждение 2 доказано.

### 3. Открытые вопросы

Интересными для дальнейшего исследования представляются следующие вопросы: доказательство или опровержение гипотезы 1 в случаях,

когда  $m \geq 5$  и  $8 \leq p < 4(m - 1)$ ; постановка задачи для несбалансированной ячейки; постановка задачи для нерегулярной ячейки; постановка задачи с ненулевыми временами разгрузки и загрузки деталей на машину; постановка задачи для «кругового» расположения машин, когда предполагается, что машины расположены по кругу,  $M_0$  и  $M_{m+1}$  совпадают, а робот при перемещении выбирает наименьшую из двух дуг, соединяющих машины; нахождение верхней границы степени оптимального цикла.

### ЛИТЕРАТУРА

1. **Agnētis A.** Scheduling no-wait robotic cells with two and three machines // Eur. J. Oper. Res. — 2000. — Vol. 123, N 2. — P. 303–314.
2. **Brauner N.** Identical part production in cyclic robotic cells: concepts, overview and open questions // Discrete Appl. Math. — 2008. — Vol. 156, N 13. — P. 2480–2492.
3. **Che A., Chu C.** Multi-degree cyclic scheduling of a no-wait robotic cell with multiple robots // Eur. J. Oper. Res. — 2009. — Vol. 199, N 1. — P. 77–88.
4. **Crama Y. et al.** Cyclic scheduling in robotic flowshops // Ann. Oper. Res. — 2000. — Vol. 96. — P. 97–124.
5. **Dawande M., Geismar H. N., Sethi S. P., Sriskandarajah C.** Sequencing and scheduling in robotic cells: recent developments // J. Sched. — 2005. — Vol. 8. — P. 387–462.
6. **Kats V., Levner E.** A strongly polynomial algorithm for no-wait cyclic robotic flowshop scheduling // Oper. Res. Lett. — 1997. — Vol. 21, N 4. — P. 171–179.
7. **Levner E., Kats V., Alcaide D., Cheng T. C. E.** Complexity of cyclic scheduling problems: a state-of-the art survey // Comput. Ind. Eng. — 2010. — Vol. 59, N 2. — P. 352–361.
8. **Levner E., Kats V., Levit V. E.** An improved algorithm for cyclic flowshop scheduling in a robotic cell // Eur. J. Oper. Res. — 1997. — Vol. 97, N 3. — P. 500–508.
9. **Mangione F., Brauner N., Penz B.** Cyclic production for the robotic balanced no-wait flow shop // Int. Conf. Ind. Eng. Production Management (Porto, Portugal, 2003). Vol. 2. — Porto: IERM, 2003. — P. 539–547.

Павлов Сергей Вадимович,  
e-mail: kiberdweller@gmail.com

Статья поступила  
20 декабря 2011 г.

Переработанный вариант —  
4 октября 2012 г.