

УДК 519.854

СЛОЖНОСТЬ ЗАДАЧИ ПОСТРОЕНИЯ ЦИКЛИЧЕСКИХ РАСПИСАНИЙ ОБРАБОТКИ ОДНОТИПНЫХ ДЕТАЛЕЙ ^{*)}

Е. А. Боброва, А. А. Романова, В. В. Сервах

Аннотация. Исследована сложность построения оптимальных циклических расписаний в случае, когда число деталей, одновременно находящихся в обработке, не превосходит заданной величины H . Доказана NP-трудность задачи при фиксированном $H \geq 4$. При $H = 2$ предложен алгоритм полиномиальной трудоёмкости.

Ключевые слова: однотипные детали, циклическое расписание.

Введение

На производственной линии, состоящей из m машин, требуется обработать N однотипных деталей. Обработка каждой детали заключается в последовательном выполнении n операций $O_1 \rightarrow O_2 \rightarrow \dots \rightarrow O_n$. Операция O_i выполняется в течение p_i единиц времени на машине m_i ($i = 1, 2, \dots, n$). Машины в технологическом порядке могут повторяться. Одновременное выполнение двух и более операций на одной машине не допускается. Прерывания операций запрещены.

Расписание называется *циклическим*, если выполнение соответствующих операций любых двух последовательно обрабатываемых деталей происходит через промежуток времени, равный C и называемый *длиной цикла*. Расписание полностью определено, если заданы моменты начала выполнения каждой операций первой детали s_i и C . Тогда выполнение операции i детали j начинается в момент $s_i + (j - 1)C$.

Циклические расписания играют важную роль в организации производства [4, 7, 8, 10–14, 16]. Они обеспечивают ритмичность загрузки оборудования и исполнителей, позволяют эффективно планировать как поставки ресурсов, так и сбыт продукции. На основе циклических расписаний в [5] описаны асимптотически точные алгоритмы минимизации общего времени обработки однотипных деталей.

^{*)}Исследование выполнено при финансовой поддержке Российского фонда фундаментальных исследований (проекты 12-01-00184-а и 12-01-00122) и целевой программы СО РАН (интеграционный проект 7Б).

Задача минимизации циклического времени полиномиально разрешима [6], и длина цикла равна сумме длительностей операций на самой загруженной машине. Однако такое расписание имеет ряд недостатков. Прежде всего это связано с простоями между операциями, что требует дополнительной транспортировки и промежуточного хранения деталей. Часто число деталей, одновременно находящихся в обработке, ограничено технологически. Например, количество площадок H при сборке самолётов фиксировано, и пока площадка не освободится, невозможно начать сборку очередного самолёта. При химической обработке на деталь наносится многослойное покрытие, для чего она крепится на оснастке и последовательно опускается в соответствующие резервуары. Количество одновременно обрабатываемых деталей ограничено числом оснасток H . Аналогичные постановки возникают и в других отраслях промышленности.

Такую задачу обозначим через $J|id, H|C$, где J — тип технологического маршрута обработки деталей, в котором машины могут повторяться, id — идентичность деталей, H — верхняя граница максимального числа деталей, одновременно находящихся в обработке, а C — критерий (в данном случае необходимо минимизировать циклическое время). Задача $J|id, H|C$ NP-трудна в сильном смысле [8]. Отметим, что в этой работе H является частью входа. При фиксированном H в [2] предложен псевдополиномиальный алгоритм трудоёмкости $O((2e)^H H^{5/2} P^{2H-1})$, где P — суммарная длительность всех операций одной детали. Однако вопрос о существовании полиномиального алгоритма оставался открытым. Ниже показано, что при фиксированном $H \geq 4$ задача NP-трудна.

В случае $H = 2$ алгоритм из [2] имеет трудоёмкость $O(P^3)$. В [1] трудоёмкость алгоритма понижена до $O(Pn^2 \log_2 n)$ операций. Но оба алгоритма псевдополиномиальны. В настоящей работе построен алгоритм полиномиальной трудоёмкости для задачи $J|id, H = 2|C$. Вопрос о сложности задачи $J|id, H = 3|C$ остаётся открытым.

1. NP-трудность задачи при фиксированном $H \geq 4$

Теорема 1. Задача $J|id, H|C$ минимизации времени цикла при ограничении величиной H числа одновременно обрабатываемых деталей NP-трудна при фиксированном $H \geq 4$.

ДОКАЗАТЕЛЬСТВО. Рассмотрим NP-трудную задачу $J3|3 \text{ детали}|C_{\max}$ из [15] с тремя различными деталями J_1, J_2, J_3 и тремя машинами с номерами 1, 2, 3. Полиномиально сведем её к задаче $J|id, H = 4|C$.

Добавим машину с номером 0 и зададим технологический маршрут

однотипных деталей: сначала деталь обрабатывается на машине с номером 0, потом с номером 1, затем в соответствии с маршрутом детали J_1 и далее, как изображено на рис. 1.

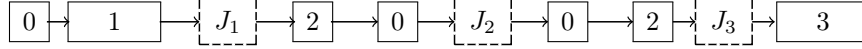


Рис. 1. Технологический маршрут детали

Зададим число A и длительности операций следующим образом:

m_i	0	1	1-3	2	0	1-3	0	2	1-3	3
p_i	A	$2A$	J_1	A	A	J_2	A	A	J_3	$2A$

Покажем, что в разномаршрутной задаче $J3|3$ детали $|C_{\max}$ существует допустимое расписание длины не более A тогда и только тогда, когда в сформированной задаче $J|id, H = 4|C$ существует расписание с временем цикла не более $3A$.

В одну сторону очевидно. Если в задаче $J3|3$ детали $|C_{\max}$ существует допустимое расписание длины не более A , то циклическое расписание, изображённое на рис. 2, удовлетворяет всем ограничениям и имеет длину цикла $3A$. На рисунке строки соответствуют деталям, цикл выделен тёмным цветом.

$k-4$	J_1	2	0	J_2	0	2	J_3	3				
$k-3$	0	1		J_1	2	0	J_2	0	2	J_3	3	
$k-2$				0	1		J_1	2	0	J_2	0	2
$k-1$							0	1		J_1	2	0
k							0	1		J_1	2	0

Рис. 2. Оптимальное циклическое расписание при $H = 4$

В другую сторону. Пусть найдено циклическое расписание с временем цикла не более $3A$. Без ограничения общности будем считать, что начало цикла совпадает с временем начала первой операции детали k . Заметим, что суммарная длительность выполнения операций одной детали на машине 0 равна $3A$. Поэтому длина цикла равна в точности $3A$, а начинается он в момент $3A(k-1)$. Более того, расположение операций на машине 0 фиксировано: сначала выполняется первая операция детали k , затем операция, следующая за фрагментом J_2 детали $k-2$, и, наконец, операция, предшествующая фрагменту J_2 детали $k-1$. В противном случае обработка детали растянется более чем на 4 цикла.

Рассмотрим расписание деталей $k-3$, $k-2$, $k-1$, k в промежутке времени $[3A(k-1), 3Ak]$. В силу заданной структуры операций на

этом временном интервале фрагмент J_1 принадлежит детали $k-1$, фрагмент J_2 — детали $k-2$, фрагмент J_3 — детали $k-3$, и каждый из этих фрагментов встречается в точности один раз.

Прежде всего отметим, что выполнение операций фрагмента J_2 на машинах 1–3 возможно только в промежутке $[3A(k-1), 3A(k-1) + A]$, так как они обязаны выполняться между операциями машины 0.

Далее, суммарная длительность первых двух операций детали $k-1$ равна $3A$. Следовательно, фрагмент J_1 начинается не ранее момента $3A(k-1)$. С другой стороны, он должен закончиться не позже момента $3A(k-1) + A$, так как следующая операция длительности A должна успеть выполняться до фиксированной операции на машине 0. Таким образом, операции фрагмента J_1 на машинах 1–3 должны выполняться в промежутке времени $[3A(k-1), 3A(k-1) + A]$.

Наконец, последняя операция детали $k-3$ длительностью $2A$ должна закончиться в рассматриваемом промежутке не позже времени $3Ak$, иначе выполнение детали растянется более чем на 4 цикла. Значит, операции фрагмента J_3 должны закончиться не позже времени $3A(k-1) + A$. С другой стороны, раньше времени $3A(k-1)$ они начаться не могут, иначе предшествующая операция на машине 2 не успеет выполняться после фиксированной операции на машине 0.

Таким образом, если время цикла равно $3A$, то выполнение фрагментов J_1, J_2, J_3 возможно только в промежутке $[3A(k-1), 3A(k-1) + A]$ для всех k , поэтому длина соответствующего расписания не превышает A .

Полиномиальность сводимости очевидна. Теорема 1 доказана.

2. Графическое представление задачи $J|id, H = 2|C$

Рассмотрим случай $H = 2$, когда одновременно может обрабатываться не более двух деталей. На рис. 3 приведён пример циклического расписания при $H = 2$ для задачи со следующими данными:

m_i	1	2	1	2	1	2
p_i	12	9	5	17	14	9

В течение одного цикла должны быть выполнены все операции (возможно, разных деталей), причём ровно по одному разу. Если в цикле выполняются операции i и j одной детали, то обязательно выполняются и операции $i+1, i+2, \dots, j-1$ этой детали. Тем самым цепочка операций разрезается на две части, содержащие последовательные операции только одной детали. Некоторая операция может быть разделена на две части. В силу непрерывности выполнения операций её части должны

примыкать к границам цикла. Внутри цикла требуется уложить операции как можно плотнее. Таким образом, для фиксированного разреза получаем задачу Job-Shop с цепочками операций $O_1 \rightarrow O_2 \rightarrow \dots \rightarrow O'_k$ и $O''_k \rightarrow O_{k+1} \rightarrow \dots \rightarrow O_n$ и дополнительными ограничениями.

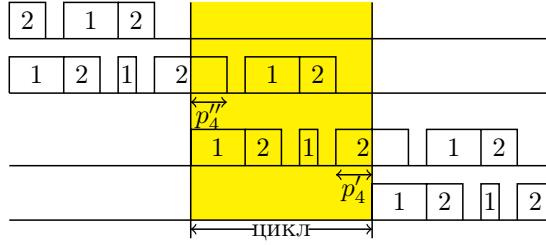


Рис. 3. Циклическое расписание при $H = 2$

Для решения задачи Job-Shop удобно использовать графическое представление задачи из [9]. В системе координат на осях откладываем длительности операций цепей $O_1 \rightarrow O_2 \rightarrow \dots \rightarrow O'_k$ и $O''_k \rightarrow O_{k+1} \rightarrow \dots \rightarrow O_n$ соответственно. Каждому допустимому расписанию отвечает путь от точки $(0, 0)$ до точки $(t, P - t)$, где t — длина первой цепи.

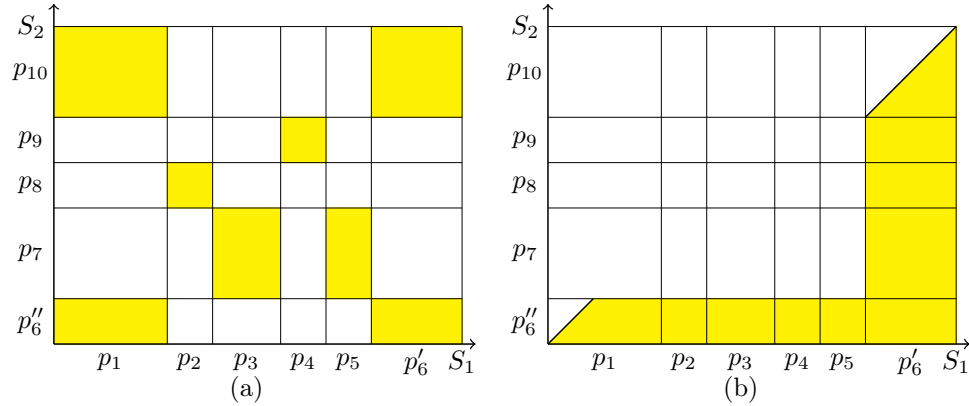


Рис. 4. Недопустимые области

На рис. 4 изображены недопустимые области, через которые путь не должен проходить (p'_k и p''_k — длительности частей операции O_k). Области первого типа на рис. 4(а) соответствуют одновременному выполнению двух операций на одной машине, а области второго типа на рис. 4(б) возникают из-за требования непрерывности операции, по которой сделан разрез на цепи.

В нашем случае число недопустимых областей первого типа не превосходит n^2 . Поэтому для фиксированного разреза минимальная длина цикла может быть найдена не более чем за $O(n^2 \log_2 n)$ операций. На рис. 5 приведён пример построения путей для одной из задач и вид соответствующего графа. Вершинами графа являются вершины запрещённых областей, которые огибает путь. Рёбра соединяют вершины, между которыми не лежат запрещённые области. Каждому ребру приписывается вес, равный расстоянию между инцидентными ему вершинами. Расстояние определяется следующим образом: пусть первая точка имеет координаты (x_1, y_1) , а вторая — координаты (x_2, y_2) , тогда расстояние между ними равно $\max(|x_1 - x_2|, |y_1 - y_2|)$. Расписание наименьшей длины соответствует кратчайшему пути между вершинами, отвечающему началу координат и точке $(t, P - t)$.

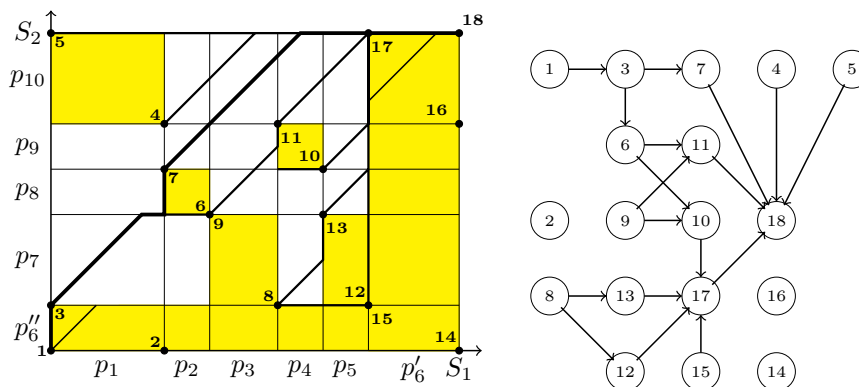


Рис. 5. Граф допустимых путей

3. Подход к решению задачи

Как показано выше, можно найти длину расписания, если известно, в каком месте последовательность операций делится на две части, одна из которых соответствует в цикле завершению обработки предыдущей детали, а другая — началу обработки новой. Теперь необходимо решить вопрос, в каком месте разрезать цепь, чтобы получить минимальную

длину цикла. Введём функцию $L : [0, P] \rightarrow R$, где $L(t)$ — длина оптимального расписания при разрезе последовательности операций на части длины t и $P - t$ соответственно. Необходимо найти такое t , при котором функция $L(t)$ принимает наименьшее значение.

Обозначим через $T_k = \sum_{i=1}^k p_i$ суммарную длину первых k операций детали. В [1] доказаны следующие свойства функции $L(t)$:

- (i) на интервалах (T_{k-1}, T_k) она непрерывна;
- (ii) в точках T_k она может быть разрывной, причём

$$L(T_k) \leq \lim_{t \rightarrow T_k - 0} L(t), \quad L(T_k) \leq \lim_{t \rightarrow T_k + 0} L(t)$$

и оба неравенства могут быть строгими;

(iii) функция $L(t)$ кусочно-линейна и угловой коэффициент любой линейной части графика равен либо 0, либо 1, либо -1 .

На рис. 6 приведён график функции $L(t)$ для примера с 10 операциями:

m_i	1	2	2	1	2	2	1	2	1	2
p_i	46	34	50	30	88	62	15	93	103	79

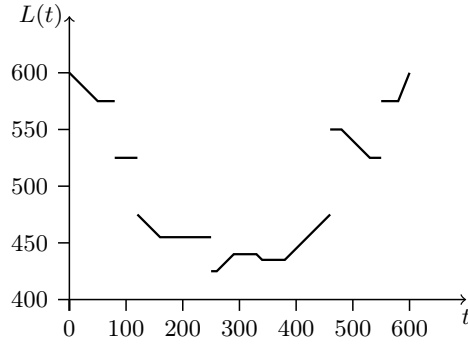


Рис. 6. Пример функции $L(t)$

По свойству (ii) $L(T_k)$ может не совпадать с левым и правым пределами. Поэтому при поиске минимума функции необходимо вычислить значения $L(T_k)$ для всех $k = 1, 2, \dots, n - 1$. Кроме этого требуется найти минимум функции $L(t)$ внутри каждого интервала (T_{k-1}, T_k) .

Расширим графическое представление. По каждой из осей отложим все операции по обработке детали и построим граф допустимых путей для выделенных вершин в квадрате $\{(x, y) \mid 0 \leq x \leq P, 0 \leq y \leq P\}$. Считаем выделенными вершинами левый верхний и правый нижний углы запрещённых областей. Веса дуг у построенного графа не зависят от параметра t . Для того чтобы из этого представления получить граф для конкретного значения t , нужно отбросить вершины, соответствующие координаты которых больше t по оси абсцисс и меньше t по оси ординат.

Если разрез лежит внутри интервала $t \in (T_{k-1}, T_k)$, то отбрасываем также точки, лежащие внутри запрещённых областей второго типа. Кроме того, нужно добавить в граф вершины, соответствующие точкам $(0, t)$ и (t, P) , и инцидентные с ними дуги. Веса добавленных дуг зависят от конкретного значения t .

Заметим, что при изменении t вид графа меняется дискретно. Как отмечено выше, отдельно надо рассмотреть его конфигурации в точках $t = T_k$. Внутри интервала (T_{k-1}, T_k) структура графа путей не меняется, пока луч, выходящий из точки $(0, t)$ под углом 45° к оси Ox , первой пересекает одну и ту же область. Меняться будут только длины добавляемых дуг. Значит, на некотором интервале можно рассматривать параметрический граф, причём зависят от параметра t только длины дуг, инцидентных $(0, t)$ и (t, P) . Множество вершин графа и длины остальных дуг неизменны. Кратчайшее расстояние на таком параметрическом сетевом графе может быть найдено методом динамического программирования за полиномиальное время. Число точек, в которых может происходить изменение конфигурации графа, не превосходит n^2 . Такой подход позволяет решить задачу за полиномиальное время.

4. Алгоритм решения задачи $J|id, H = 2|C$

На основе графического представления задачи с учётом особенностей поиска кратчайшего расстояния на параметрическом графе предлагается следующий алгоритм поиска минимума функции $L(t)$.

ШАГ 1. Строим общее графическое представление: по обеим осям откладываем операции, соответствующие обработке одной детали. Определяем множество запрещённых областей. Для каждой запрещённой области проецируем верхние левые и нижние правые углы на ось ординат под углом 45° , упорядочиваем полученные проекции точек по невозрастанию. Получаем последовательность проекций t_1, t_2, \dots, t_l .

ШАГ 2. Строим граф возможных путей на множестве точек, содержащем левые верхние и правые нижние точки недопустимых областей первого типа. Процедура трудоёмкости $K \log K$, где K — число недопустимых областей, подробно описана в [3].

ШАГ 3. Для $k = 1, 2, \dots, n - 1$ находим $L(T_k)$ и выбираем наименьшее значение: $L_{\min} = \min\{L(T_k)\}$, $t_{\min} = \operatorname{argmin}\{L(T_k) = L_{\min}\}$, $k = 1, 2, \dots, n - 1$. Для вычисления $L(T_k)$ исключаем вершины графа, соответствующие координаты которых больше T_k по оси абсцисс и меньше T_k по оси ординат, и добавляем начальную и конечную вершины,

отвечающие точкам $(0, T_k)$ и (T_k, P) . Вводим соответствующие дуги и вычисляем их веса. В построенном графе находим $L(T_k)$.

Заметим, что далее поиск минимума достаточно вести на интервале $(P - L_{\min}, L_{\min})$.

ШАГ 4. На каждом из непустых отрезков $[t_m, t_{m+1}] \cap (T_{k-1}, T_k)$, $k = 1, 2, \dots, n$, $m = 1, 2, \dots, l - 1$, вычисляем минимальное значение функции $L(t)$. Соответствующая процедура описана ниже. На данном шаге при исключении вершин исходного графа важно учитывать области второго типа. При $t_m = T_{k-1}$ или $t_{m+1} = T_k$ минимума на открытом интервале может не существовать. Тогда достаточно рассмотреть замыкание интервала, а для граничных точек значения $L(t)$ вычислены на предыдущем шаге, причём по свойству (ii) значение функции $L(t)$ в крайней точке не превосходит величины соответствующего одностороннего предела.

Если получено значение, меньшее L_{\min} , то обновляем рекорд и момент времени t_{\min} , на котором он достигается.

ШАГ 5. $L_{\min} = L(t_{\min})$ — наименьшее значение функции $L(t)$ на $[T_0, T_n]$.

Процедура вычисления минимума на интервале. Ищется минимум функции $L(t)$ на отрезке $t \in [t_m, t_{m+1}] \cap (T_{k-1}, T_k)$. Важно, что граф на этом отрезке не меняется, так как для этих значений t не меняется область, которую первой пересекает луч, выходящий из точки $(0, t)$ под углом 45° к осям. Расстояние от точки $(0, t)$ до левого верхнего угла этой области (x_1, y_1) будет равно $y_1 - t$. До её правого нижнего угла (x_2, y_2) при условии $t < y_2$ расстояние равно x_2 . Если $t > y_2$, то точка (x_2, y_2) будет удалена, так как лежит в недопустимой области второго типа. Случай равенства $t = y_2$ рассматривался на шаге 3 алгоритма. Таким образом, из вершины графа, отвечающей точке $(0, t)$, выходит либо одна, либо две дуги, причём вес только одной зависит от параметра t .

Теперь из каждой конечной вершины оставшегося графа необходимо провести дуги до вершины, отвечающей точке (t, P) . Пусть точка с координатами (x, y) отвечает некоторой конечной вершине графа. Вес дуги, проведённой из (x, y) в (t, P) , равен $\max\{t - x, P - y\}$, кроме случая, когда при обходе недопустимой области второго типа приходится обходить и области первого типа. Пусть j — наибольший номер операции, которая выполняется на одной машине с операцией O_k , т.е. $m_j = m_k$, и выполнены условия $y < x + P - t$, $T_j > P - t + T_{k-1}$. Тогда область, соответствующую одновременному выполнению операций j и k , требуется обойти, а длина пути от (x, y) до (t, P) равна $T_j - y + t - T_{k-1}$.

Аналогично строится путь, если луч, выходящий из точки $(0, t)$, не пересекает область первого типа. Тогда между вершинами, отвечающими точкам $(0, t)$ и (t, P) , проходит только одна дуга весом $\max\{t, P - t\}$ или $T_j - T_{k-1}$, если $m_j = m_k$, и выполнены условия $t < P/2$, $T_j > P - t + T_{k-1}$.

Граф построен. В нём от параметра t зависят вес только одной начальной дуги, причём линейно, и веса нескольких дуг, входящих в конечную вершину. Вид графа изображён на рис. 7. Функция весов $c_{xy}(t)$ на дугах, входящих в конечную вершину, имеет вид

$$c_{xy}(t) = \begin{cases} T_j - y + t - T_{k-1}, & \text{если } y < x + P - t, T_j > P - t + T_{k-1}, \\ \max\{t - x, P - y\} & \text{в противном случае.} \end{cases}$$

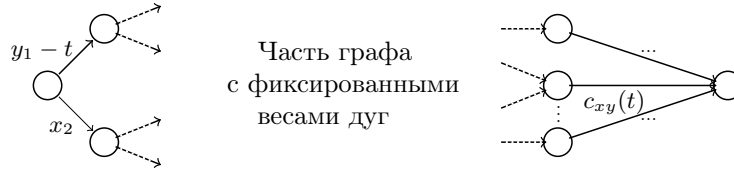


Рис. 7. Вид графа при разрезе внутри операции

Используем классический алгоритм поиска кратчайшего пути в сетевом графе. Длина пути до всех вершин за исключением последней равна либо константе α , либо функции вида $\beta - t$, либо $\min\{\alpha, \beta - t\}$. Для последней вершины получаем несложную кусочно-линейную функцию, минимум которой на отрезке $[t_m, t_{m+1}] \cap (T_{k-1}, T_k)$ ищется тривиально.

Оценим трудоёмкость алгоритма. Пусть r — число недопустимых областей в общем графическом представлении. Граф путей на нём строится за $O(r \log_2 r)$ операций. Формирование графов в точках $t = T_k$ и параметрических графов на интервалах $t \in [t_m, t_{m+1}] \cap (T_{k-1}, T_k)$, $k = 1, 2, \dots, n - 1$, а также поиск кратчайших путей реализуется за линейное от числа вершин время. Общее число точек и интервалов не превышает $2r + 2n \leq 4r$, значит, число операций при реализации алгоритма не превзойдёт $O(r \log_2 r + r^2)$ или $O(r^2)$ операций. Если оценивать через общие данные задачи, то, учитывая, что количество недопустимых областей r ограничено величиной n^2 , верхняя оценка трудоёмкости алгоритма составит $O(n^4)$ операций. Таким образом доказана

Теорема 2. Задача минимизации времени цикла при ограничении величиной $H = 2$ числа одновременно обрабатываемых деталей полиномиально разрешима с трудоёмкостью, не превышающей $O(n^4)$ операций.

Алгоритм реализован на языке C++.

ЛИТЕРАТУРА

1. **Межецкая М. А.** Задачи выпуска партии однотипных деталей с ограничением на число одновременно обрабатываемых деталей. — Омск: Изд-во ОмГУ, 2008. — 35 с. (Препринт)
2. **Романова А. А., Сервах В. В.** Оптимизация выпуска однотипных деталей на основе циклических расписаний // Дискрет. анализ и исслед. операций. — 2008. — Т. 15, № 5. — С. 47–60.
3. **Сервах В. В.** О задаче Акерса — Фридмана // Управляемые системы. — 1983. — Вып. 23. — С. 70–81.
4. **Тимковский В. Г.** Приближённое решение задачи составления расписания циклической системы // Экономика и мат. методы. — 1986. — Т. 1. — С. 171–174.
5. **Boudoukh T., Penn M., Weiss G.** Scheduling jobshops with some identical or similar jobs // J. Sched. — 2001. — Vol. 4. — P. 177–199.
6. **Brucker P.** Scheduling algorithms. — Leipzig: Springer-Verl., 2007. — 371 p.
7. **Hall N. G., Lee T. E., Posner M. E.** The complexity of cyclic shop scheduling problems // J. Sched. — 2002. — Vol. 5. — P. 307–327.
8. **Hanan C.** Study of a NP-hard cyclic scheduling problem: the recurrent job-shop // Eur. J. Oper. Res. — 1994. — Vol. 72. — P. 82–101.
9. **Hardgrave W. W., Nemhauser G. A.** Geometric model and graphical algorithm for a sequencing problem // Oper. Res. — 1963. — Vol. 11, N 6. — P. 889–900.
10. **Kamoun H., Sriskandarajah C.** The complexity of scheduling jobs in repetitive manufacturing systems // Eur. J. Oper. Res. — 1993. — Vol. 70. — P. 350–364.
11. **Levner E., Kats V., Pablo D., Cheng E.** Complexity of cyclic scheduling problems: a state-of-the-art survey // Comp. Ind. Eng. — 2010. — Vol. 59, N 2. — P. 352–361.
12. **McCormick S. T., Rao U. S.** Some complexity results in cyclic scheduling // Math. Comput. Modeling. — 1994. — Vol. 20. — P. 107–122.
13. **Rao U., Jackson P.** Identical jobs cyclic scheduling: subproblems, properties, complexity and solution approaches. — Ithaca, NY: Cornell Univ. Press, 1993. — 48 p.
14. **Roundy R.** Cyclic schedules for job shops with identical jobs // Math. Oper. Res. — 1992. — Vol. 17, N 4. — P. 842–865.
15. **Sotskov Y. N., Shakhlevich N. V.** NP-hardness of shop-scheduling problems with three jobs // Discrete Appl. Math. — 1995. — Vol. 59, N 3. — P. 237–266.

- 16. Timkovsky V. G.** Cyclic shop scheduling. Handbook of scheduling: algorithms, models, and performance analysis. — Boca Raton: Chapman & Hall/CRC, 2004. — P. 7.1–7.22.

Боброва Екатерина Александровна,

e-mail: eabobrova88@gmail.com

Романова Анна Анатольевна,

e-mail: anuta81@bk.ru

Сервах Владимир Вицентьевич,

e-mail: svv_usa@rambler.ru

Статья поступила

21 ноября 2012 г.

Переработанный вариант —

3 апреля 2013 г.