

УДК 519.8

DOI 10.17223/20710410/64/8

ЗАДАЧА МИНИМИЗАЦИИ ОБЩЕГО ВРЕМЕНИ ОБРАБОТКИ ИДЕНТИЧНЫХ ДЕТАЛЕЙ¹

А. А. Романова*, В. В. Сервах***, В. Ю. Тавченко*

* Омский государственный университет им. Ф. М. Достоевского, г. Омск, Россия

** Институт математики им. С. Л. Соболева СО РАН, г. Омск, Россия

E-mail: RomanovaAA@omsu.ru, svv_usa@rambler.ru, nikapolicheva@mail.ru

Рассматривается задача минимизации общего времени обработки идентичных деталей со сложным технологическим маршрутом, когда возможно неоднократное поступление деталей на некоторые машины. Исследуются вопросы вычислительной сложности данной задачи, доказана её NP-трудность в обычном смысле. При фиксированном числе деталей доказана псевдополиномиальная разрешимость задачи. Исследуется вопрос использования циклических расписаний при построении приближённых решений.

Ключевые слова: расписание, идентичные детали, NP-трудность, псевдополиномиальный алгоритм, теория сложности.

MAKESPAN MINIMIZATION IN REENTRANT FLOW SHOP PROBLEM WITH IDENTICAL JOBS

A. A. Romanova*, V. V. Servakh***, V. Yu. Tavchenko*

* Dostoevsky Omsk State University, Omsk, Russia

** Sobolev Institute of Mathematics, Omsk, Russia

We consider the reentrant flow shop problem $F|reentrant, p_{ij} = p_i|C_{\max}$ with identical jobs and makespan criterion. We prove its NP-hardness in the ordinary sense. The proof is performed by polynomial reduction of the problem $J3|n = 3|C_{\max}$ to the problem $F|reentrant, p_{ij} = p_i|C_{\max}$ with three identical jobs. Using the input data of the problem $J3|n = 3|C_{\max}$, we have constructed a special type of job for the problem $F|reentrant, p_{ij} = p_i|C_{\max}$. In the proof, we analyze all possible variants of critical paths in the constructed instance. We also propose a dynamic programming algorithm to find the optimal solution of the problem $F|reentrant, p_{ij} = p_i|C_{\max}$. Analysis of the time complexity of the algorithm showed that the problem with fixed number of jobs is pseudopolynomially solvable. Next, we study the use of cyclic schedules to construct approximate solutions. A cyclic schedule with a minimum cycle time can be found in polynomial time. We propose a polynomial algorithm for finding an approximate solution. This algorithm is based on the construction of a cyclic schedule with a minimum cycle time and its subsequent compaction at the beginning and at the end. We construct an upper bound of the makespan of cyclic schedules. This bound depends on the number of jobs processed in one cycle. The paper provides numerous examples characterizing cyclic schedules from both the positive and negative sides to solve the problem $F|reentrant, p_{ij} = p_i|C_{\max}$.

¹Работа выполнена в рамках госзадания ИМ СО РАН, проект FWNF-2022-0020.

Keywords: *schedule, identical jobs, NP-hardness, pseudopolynomial algorithm, theory of NP-completeness.*

Введение

Рассматривается задача обработки идентичных деталей со сложным технологическим маршрутом на современной роботизированной производственной линии. В общепринятой номинации [1] эта задача обозначается как $F|reentrant, p_{ij} = p_i|C_{\max}$, подразумевая, что все детали проходят одинаковый технологический маршрут и длительности соответствующих операций равны. Тем самым все детали идентичны, а термин «reentrant» означает, что при обработке деталь может поступать на некоторые машины неоднократно. Задача $F|reentrant, p_{ij} = p_i|C_{\max}$ является обобщением классической одномаршрутной задачи Flow-Shop и впервые была сформулирована в [2]. Большинство известных результатов можно найти в [3]. Данная задача является NP-трудной даже для случая двух машин. Фактически большинство теоретических вопросов, связанных с общей постановкой, так или иначе закрыты. Если же детали идентичны, то возникает множество математически интересных постановок. Ключевым является вопрос о вычислительной сложности задачи $F|reentrant, p_{ij} = p_i|C_{\max}$. В литературе рассматриваются различные частные, в том числе полиномиально разрешимые случаи. Однако, как отмечено в обзоре [4], вопрос о полиномиальной разрешимости задачи с идентичными деталями даже для единичных длительностей работ остаётся открытым.

Будем придерживаться следующей постановки. Имеется заказ на выпуск N идентичных деталей. Для их обработки имеется m различных машин $M_1, M_2, \dots, M_m$. Все детали в процессе обработки проходят одинаковый технологический маршрут, который состоит из n последовательно выполняемых операций $O_1, O_2, \dots, O_n$. Операция O_i выполняется на машине M_{m_i} в течение $p_i \in \mathbb{Z}^+$ единиц времени, $i = 1, \dots, n$. Прерывания операций запрещены. Машина не может выполнять более одной операции одновременно. В технологическом маршруте машины могут повторяться. Требуется минимизировать общее время выполнения заказа.

В классической одномаршрутной задаче число операций n совпадает с числом машин m , а технологический маршрут задаётся последовательностью $(M_1, M_2, \dots, M_m)$. Для идентичных деталей с таким маршрутом получаем обычный конвейер, и задача решается тривиально. Неоднократное использование машин в технологическом маршруте приводит к различным постановкам задач. В литературе рассматривались случаи V-маршрута $(M_1, M_2, \dots, M_m, M_{m-1}, \dots, M_2, M_1)$ [5], циклического $(M_1, M_2, \dots, M_m, M_1, M_2, \dots, M_m, \dots, M_1, M_2, \dots, M_m)$ [6, 7], замкнутого $(M_1, M_2, \dots, M_m, M_1)$ [8], с постоянным возвратом на первую машину $(M_1, M_2, M_1, M_3, M_1, \dots, M_1, M_m, M_1)$ [9, 10] и т. д. [11–13]. Особенно интересен последний случай, в котором машина M_1 рассматривается как транспортное средство, перемещающее деталь между операциями [14, 15].

Важным аспектом является взаимосвязь рассматриваемой задачи с задачей построения циклических расписаний. Циклические расписания строятся за полиномиальное время и обеспечивают ритмичность производства, равномерный выход продукции, более удобную логистику. Наиболее значимым является результат из [16], где показано, что с ростом количества деталей циклическое расписание с минимальным временем цикла обеспечивает асимптотически точное решение для критерия $C_{\max}$. Однако такой подход не решает проблем загрузки оборудования на стадии запуска и стадии завершения обработки партии деталей в производство. Поэтому, если производственная но-

Структура работы следующая. В п. 1 изложен основной результат: доказано, что задача является NP-трудной в обычном смысле. В п. 2 приведён алгоритм построения точного решения, который является псевдополиномиальным при фиксированном числе деталей. В п. 3 обсуждается вопрос об использовании циклических расписаний при минимизации $C_{\max}$, в том числе при наличии дополнительных ограничений на расписание.

Некоторые подходы к анализу сложности задачи предложены в [17]. Входные параметры задачи — это число деталей N , число операций n и их длительности p_i , $i = 1, 2, \dots, n$. Длина входа равна $O(\log_2 N + n \log_2 p_{\max})$, где $p_{\max}$ — максимальная длительность операций. Далее доказана NP-трудность задачи и сделан анализ её сложности в зависимости от входных параметров.

Доказательство. Рассмотрим NP-трудную разномаршрутную задачу $J3|n = 3|C_{\max}$ с тремя машинами M_1, M_2, M_3 и тремя деталями J_1, J_2, J_3 [18]. Сформулируем её как задачу распознавания. Задано число d . Существует ли допустимое расписание выполнения работ длины не более d ? Полиномиально сведём её к задаче $F|reentrant, p_{ij} = p_i|C_{\max}$ с тремя идентичными деталями.

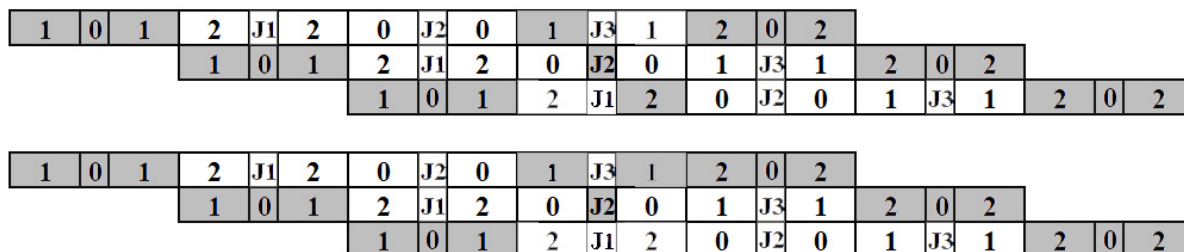
Пусть M_1 и M_2 — машины, на которых выполняются соответственно первая и последняя операция детали $J2$. Если эти операции выполняются на одной машине, то обозначим её M_1 . Добавим машину M_0 и три фиктивные машины M_3 , M_4 и M_5 , которые в дальнейшем будут заменены на технологические маршруты деталей $J1$, $J2$ и $J3$. Рассмотрим деталь со следующим технологическим маршрутом и соответствующими длительностями:

O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}	O_{11}	O_{12}	O_{13}	O_{14}	O_{15}
M_1	M_0	M_1	M_2	M_3	M_2	M_0	M_4	M_0	M_1	M_5	M_1	M_2	M_0	M_2
$4d$	d	$4d$	$4d$	d	$4d$	$4d$	d	$4d$	$4d$	d	$4d$	$4d$	d	$4d$

Пусть необходимо обработать три таких детали. Оптимальное расписание приведено на рис. 1. Его длина равна $63d$. Номера машин подписаны. Наиболее загруженными являются машины M_1 и M_2 , и в расписании они представлены симметрично относительно операции обращения времени. Операции каждой детали обрабатываются без простоев, и операция O_1 второй детали выполняется только после завершения операции O_3 первой детали. При любой другой последовательности операций время обработки будет не менее $66d$. Такая конструкция позволяет локализовать операции O_5 , O_8 и O_{11} в оптимальном расписании.

[illegible]

Рис. 1. Оптимальное расписание для трёх деталей

Рис. 5. Критический путь для двух вариантов $J2 - J2$

9 комбинаций первой и последней операций критического пути. К ним добавятся ещё 3 варианта во втором случае. С учётом этого общее количество различных вариантов концов критического пути равно 12. Рассмотрим некоторые из них подробнее.

Пусть критический путь для задачи с деталями $J1, J2, J3$ содержит первую и последнюю операции детали $J1$. Назовём его $J1 - J1$. Тогда выделенная на первой части рис. 4 последовательность операций имеет длину больше чем $63d$. Вариант $J3 - J3$ аналогичен и приведён на второй части рисунка.

Вариант $J2 - J2$. Первая операция критического пути выполняется на машине M_1 , последняя — на машине M_2 . Выделенная на рис. 5 последовательность операций имеет длину больше чем $63d$. Критический путь в случае, если последняя операция детали $J2$ выполняется на машине M_1 , приведён на второй части рисунка.

Все остальные комбинации также дают критический путь, длина которого больше чем $63d$. Действительно, в любом случае слева критический путь упирается в предшествующую операцию, выполняемую на машине 1 или 2, эти операции невозможно сдвинуть на более ранний срок. Справа также всё ограничивают операции, сдвиг которых увеличивает длину критического пути. ■

Отметим, что доказана только обычная NP-трудность рассматриваемой задачи. Ниже предложен алгоритм построения точного решения задачи, который при фиксированном числе деталей является псевдополиномиальным. Более того, на основе этого алгоритма может быть построена вполне полиномиальная аппроксимационная схема [19]. Таким образом, полученный результат является неулучшаемым для случая фиксированного числа работ. Однако вопрос о вычислительной сложности задачи в зависимости от числа работ остаётся открытым.

2. Точный алгоритм решения задачи

Опишем алгоритм построения точного решения рассматриваемой задачи, идея которого предложена в [20] для задачи календарного планирования с ограниченными ресурсами. Далее этот алгоритм адаптирован для задачи с идентичными деталями $F|reentrant, p_{ij} = p_i|C_{\max}$.

В основе алгоритма лежит геометрическая интерпретация задачи, в которой каждой детали сопоставляется ось в N -мерном пространстве и каждая точка параллелепипеда $[0, P]^N$ соответствует промежуточному состоянию обработки деталей, где $P = \sum_{i=1}^n p_i$ — суммарная длительность операций одной детали. Текущее состояние процесса обработки детали j будем задавать величиной x_j . Значение x_j определяет, сколько времени, без учёта простоев, деталь j уже обрабатывалась, и требуется ещё $P - x_j$ единиц времени для завершения обработки. Каждое расписание обработки деталей отображается ломаной линией между точками $\mathbf{0} = (0, 0, \dots, 0)$, когда ни одна деталь

не начала обработку, и $\mathbf{S} = (P, P, \dots, P)$, когда все детали обработаны. Любое звено ломаной проходит между точками $(x_1, x_2, \dots, x_N)$ и $(x_1 + \tau_1, x_2 + \tau_2, \dots, x_N + \tau_N)$, где $\tau_i = 0$ или $\tau_i = \tau$. Такое звено соответствует одновременной обработке в течение времени τ тех деталей, для которых $\tau_j = \tau$. Остальные детали в этот период не обрабатываются. Длина этого участка ломаной полагается равной τ , а длина расписания равна сумме длин всех звеньев ломаной.

Так как p_i — целые, существует оптимальное расписание, в котором все операции начинаются и заканчиваются в целочисленные моменты времени и для каждого целого t на полуинтервале $(t - 1, t]$ множество обрабатываемых деталей не изменяется. Поэтому достаточно рассмотреть только такие расписания, для которых ломаная меняет направление только в целочисленных точках. Целочисленный вектор $\mathbf{x} = (x_1, x_2, \dots, x_N)$ будем называть *состоянием* обработки деталей. Множество всех состояний обозначим через $X = \{\mathbf{x} = (x_1, x_2, \dots, x_N) : x_j \in \{0, 1, \dots, P\}\}$. В такой ситуации ломаную можно разбить на участки единичной длины. Каждый из них соединяет целочисленные точки и соответствует переходу из состояния $\mathbf{x}$ в состояние $\mathbf{x} + \delta$, где $\delta = (\delta_1, \delta_2, \dots, \delta_N)$, $\delta_j \in \{0, 1\}$, $j = 1, 2, \dots, N$. Процесс перехода соответствует одновременному выполнению в единичном временном интервале операций тех деталей, для которых $\delta_j = 1$.

Идея алгоритма основана на схеме динамического программирования и заключается в переборе всевозможных состояний, а для каждого состояния по рекуррентной формуле вычисляется лучшее допустимое частичное расписание.

Имеется два типа ограничений, которые необходимо учесть. Первое — это непрерывность выполнения операций. Для состояния $\mathbf{x}$ выполнение условия

$$\sum_{k=1}^{i-1} p_k < x_j < \sum_{k=1}^{i-1} p_k + p_i$$

означает, что операция i детали j была начата и ещё не завершилась. Тогда, в силу непрерывности выполнения операций, δ_j должно быть равно 1. Поэтому для такого $\mathbf{x}$ переход $\mathbf{x} \rightarrow \mathbf{x} + \delta$, у которого $\delta_j = 0$, недопустим.

Второе — это запрет на одновременное выполнение двух операций на одной машине. Состояние $\mathbf{x}$ и единичные компоненты перехода δ однозначно определяют набор операций, которые находятся в состоянии выполнения. Если хотя бы одна пара из этого набора выполняется на одной и той же машине, то переход δ недопустим.

Множество допустимых переходов, приводящих в состояние $\mathbf{x} \in X$, обозначим через $\Delta_{\mathbf{x}}$. Каждый переход соответствует выполнению некоторого набора операций в течение единичного интервала времени. Необходимо найти кратчайшую последовательность допустимых переходов из состояния $\mathbf{0}$ в состояние $\mathbf{S}$.

Пусть $L(\mathbf{x})$ — наименьшее число переходов из состояния $\mathbf{0}$ в состояние $\mathbf{x}$. Выпишем рекуррентное соотношение для поиска кратчайшего пути:

$$L(\mathbf{x}) = \min_{\delta \in \Delta_{\mathbf{x}}} \{L(\mathbf{x} - \delta) + 1\}, \quad \mathbf{x} \in X \setminus \{\mathbf{0}\}.$$

Алгоритм начинает работу с начального состояния $\mathbf{0}$ и в порядке лексикографического возрастания перебирает все отмеченные состояния $\mathbf{x} \in X$, вычисляя значения $L(\mathbf{x})$. При этом запоминает вектор $\delta(\mathbf{x}) \in \Delta_{\mathbf{x}}$, на котором достигается минимум $L(\mathbf{x})$. Величина $L(\mathbf{S})$ определяет оптимальное значение целевой функции задачи.

Для восстановления решения первоначально полагаем $\mathbf{x} = \mathbf{S}$ и рассматриваем переход $\delta(\mathbf{S})$. Во временном интервале $(L(\mathbf{S}) - 1, L(\mathbf{S})]$ выполняются операции тех деталей,

для которых $\delta_j(\mathbf{S}) = 1$. Затем переходим в состояние $\mathbf{x} = \mathbf{S} - \delta(\mathbf{S})$. Во временном интервале $(L(\mathbf{S}) - 2, L(\mathbf{S}) - 1]$ выполняются операции тех деталей, для которых $\delta_j(\mathbf{x}) = 1$. И так далее, пока не дойдём до состояния $\mathbf{x} = \mathbf{0}$. Зная множества операций, выполняемых в каждом из интервалов $(t - 1, t]$, $t = 1, 2, \dots, L(\mathbf{S})$, определяем моменты начала их выполнения.

Трудоёмкость алгоритма линейно зависит от количества состояний. Для состояния $\mathbf{x}$ нужно выбрать минимум из $\Delta_{\mathbf{x}}$ чисел. Верхняя оценка трудоёмкости составит $O(2^N P^N) = O((2P)^N)$. Для идентичных деталей эту оценку можно улучшить. Без ограничения общности детали запускаются в порядке возрастания номеров. Очевидно, что в оптимальном расписании такой же порядок сохранится при выполнении соответствующих операций всех деталей. Поэтому в любой момент имеет место $x_1 > x_2 > \dots > x_N$. Это позволяет существенно сократить число перебираемых состояний. В случае $N < P$ верхняя оценка может быть уменьшена до $O((2P)^N / N!)$.

Временная сложность алгоритма экспоненциально зависит от количества деталей N . Число машин M не влияет на трудоёмкость, так как в алгоритме для каждой пары операций делается простая проверка, обрабатываются эти операции на одной машине или нет. Если число деталей фиксировать, то предложенный алгоритм становится псевдополиномиальным. Таким образом, справедлива следующая

Теорема 2. Задача минимизации общего времени обработки партии идентичных деталей при условии, что количество обрабатываемых деталей ограничено константой, разрешима за псевдополиномиальное время.

Отметим, что этот вывод может быть также получен из [21] на основе псевдополиномиальной разрешимости задачи $J|N = \text{const} | C_{\max}$. При $N = 2$ задача полиномиально разрешима. Уже при $N = 3$ рассматриваемая задача становится NP-трудной в обычном смысле, что показано выше.

В реальных задачах NP-трудность в зависимости от длительностей не является критической, так как трудно представить ситуацию очень длинных технологических маршрутов при обработке детали. А вот количество деталей может меняться и составлять несколько сотен, а возможно, и тысячи [17]. Поэтому важно исследование вычислительной сложности задачи в зависимости от параметра N . Для задачи с идентичными деталями длина входа по этому параметру составляет $O(\log_2 N)$. То есть полиномиальность по этому параметру предполагает наличие полиномиального алгоритма от величины $O(\log_2 N)$. Как вариант, трудоёмкость алгоритма может не зависеть от N . В следующем пункте исследуем возможность использования при минимизации $C_{\max}$ циклических расписаний. Алгоритм построения расписаний с минимальным временем цикла является полиномиальным и его трудоёмкость не зависит от N .

3. Использование циклических расписаний при минимизации общего времени

Расписание называется циклическим, если выполнение соответствующих операций любых двух последовательно обрабатываемых деталей происходит через промежуток времени C , который называется длиной цикла. Минимально возможная длина цикла равна суммарной длительности операций одной детали на самой загруженной машине. Цикл называется полным, если внутри него выполняются все операции $\{1, 2, \dots, n\}$, возможно, разных деталей. Циклические расписания обеспечивают ритмичность производства, равномерный выход продукции, более удобную логистику. При минимальном времени цикла C и росте числа идентичных деталей эффективность производства возрастает и его производительность стремится к максимально возможной. Более того,

циклическое расписание с минимальным временем цикла может быть найдено за полиномиальное время. Естественным образом возникает идея использования малотрудоёмкого алгоритма построения циклических расписаний для решения NP -трудной задачи минимизации общего времени выполнения заказа. На основе таких расписаний в [16] построен асимптотически точный алгоритм для критерия $C_{\max}$. Недостатком этого алгоритма и циклических расписаний является то, что максимальная загрузка оборудования достигается только в полном цикле. При запуске партии деталей и её завершении формируются неполные циклы и происходит простой оборудования. Для сложного технологического маршрута обработки детали такие простои могут быть довольно значительными. При больших размерах партии (более нескольких тысяч) они не так чувствительны, так как их доля при росте N стремится к нулю. Но когда требуется обработать несколько десятков или сотен сложных деталей, то временные потери могут составлять существенную часть общего времени обработки. В этом случае асимптотика ещё не даёт эффекта, а описанный выше псевдополиномиальный алгоритм уже не работает из-за высокой трудоёмкости. Приведём небольшой пример:

$$\begin{pmatrix} O_1 & O_2 & O_3 & O_4 & O_5 & O_6 & \dots & O_{2k-1} & O_{2k} & O_{2k+1} \\ M_0 & M_1 & M_0 & M_2 & M_0 & M_3 & \dots & M_0 & M_k & M_0 \\ 1 & p_1 & 1 & p_2 & 1 & p_3 & \dots & 1 & p_k & 1 \end{pmatrix}.$$

Здесь $p = \sum_{i=1}^k p_i \leq 1$. Минимальное значение длины цикла определяется загрузкой машины M_0 и равно $C^* = k + 1$. Но полные циклы, в которых выполняется весь набор операций, начинаются только с $(k+1)$ -й детали. Выполнив первую и вторую операцию первой детали, мы не можем выполнять третью операцию на машине M_0 , хотя она свободна. Ждём k единиц времени и только тогда запускаем вторую деталь. Только когда для первых k деталей будут завершены операции O_1 , начнёт выполнение операция O_3 первой детали. Далее загрузка машины M_0 будет максимально возможной. Но в итоге самая дефицитная машина в начальной стадии процесса, как и в заключительной стадии, простаивает в общей сложности k^2 единиц времени. Таким образом, главным недостатком циклических расписаний являются потери, связанные с запуском и завершением всей партии деталей.

Естественно возникает желание уплотнить неполные циклы. В рассмотренном примере сжатие циклического расписания действительно даёт эффект и приводит к оптимальному значению $C_{\max}$. Далее описан алгоритм 1 уплотнения циклического расписания.

Алгоритм 1 является полиномиальным. Так как алгоритм построения расписания с минимальной длиной цикла также полиномиален, то получить таким образом оптимум с критерием $C_{\max}$ невозможно, это противоречило бы теореме 1. Более того, если рассмотреть всё множество расписаний с минимальным временем цикла и каждое из них уплотнить, то ни одно из этих расписаний может не дать оптимума. Пример очень простой:

$$\begin{pmatrix} O_1 & O_2 & O_3 & O_4 \\ M_1 & M_2 & M_3 & M_1 \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

Минимальная длина цикла $C = 2$. При $N \geq 4$ любое циклическое расписание (с учётом возможного сжатия) имеет длину $2N + 2$. А оптимум достигается на расписании рис. 6 и имеет длину $2N$. При $N = 4$ отклонение от оптимума — в 1,25 раза. Это означает, что

Алгоритм 1. Уплотнение циклического расписания

- 1: Строим циклическое расписание с минимальным временем цикла.
- 2: Выделяем первый цикл с полным набором операций.
- 3: Операции, входящие в этот и предшествующие циклы, упорядочиваем по невозрастанию времени окончания.
- 4: Просматриваем список с конца: каждую операцию сдвигаем как можно позднее, пока не нарушится отношение предшествования или две операции не поступят на одну машину. Получим плотное расписание, когда ни одна из операций не может быть сдвинута на более поздний срок.
- 5: Аналогично при завершении заказа. Берём последний цикл с полным набором операций. Операции этого и всех последующих циклов упорядочиваем по возрастанию моментов их начала.
- 6: Просматриваем список с начала, каждую операцию сдвигаем как можно раньше, до первого нарушения отношения предшествования или поступления двух операций на одну машину. Получим плотное расписание, когда ни одна операция не может быть сдвинута на более ранний срок.

для критерия минимизации общего времени обработки деталей в классе циклических расписаний нельзя гарантировать решение с оценкой точности лучше чем 1,25.

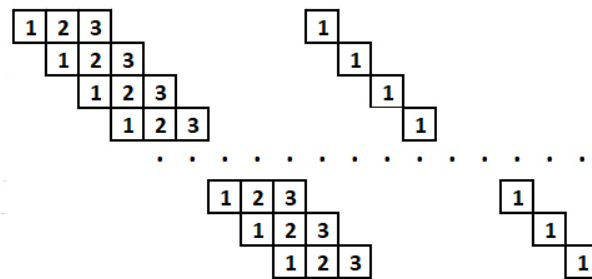


Рис. 6. Оптимальное расписание

Если исследовать вопрос о приближении, то значение $C_{\max}$ можно оценить сверху величиной $CN + C(n - 1)$, так как количество неполных циклов не превосходит $2C(n - 1)$, а полных циклов будет $(N - n + 1)$. Эта оценка может быть улучшена до значения $CN + C(k - 1)$, где k — количество обрабатываемых деталей в одном цикле. Логично было бы предположить, что чем меньше деталей задействовано в цикле, тем лучше должно быть расписание после уплотнения. Это хорошо согласуется с прежними результатами из [22], где доказана псевдополиномиальная разрешимость задачи минимизации числа деталей при условии оптимальной длины цикла. Однако удалось построить пример, когда это не так.

Пусть в предыдущем примере длительности операций равны 2, 3, 3 и 2 единицы соответственно. На первой части рис. 7 приведено циклическое расписание для $N = 5$ с минимально возможным числом деталей в цикле, равным трём. Циклы отделены вертикальными линиями. Два первых и два последних цикла неполные. На второй части рис. 7 отображено расписание после уплотнения. Его длина равна 25. На рис. 8 приведено расписание, когда в цикле минимальной длины обрабатываются четыре детали. После его уплотнения длина расписания равна 21. Тем самым число деталей в цикле

не является определяющим для поиска лучшего приближения в классе циклических расписаний.

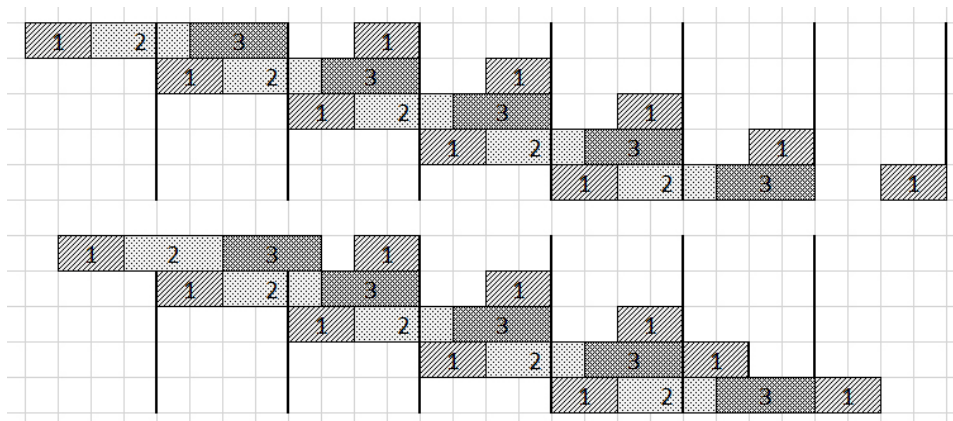


Рис. 7. Расписание для трёх деталей в цикле

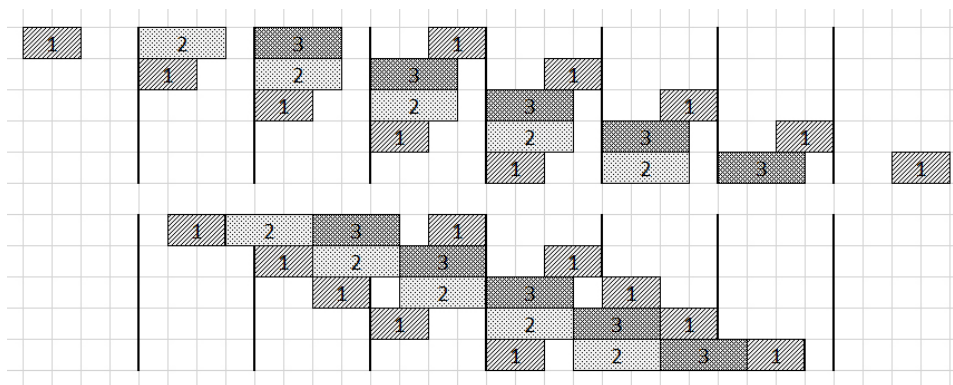


Рис. 8. Расписание для четырёх деталей в цикле

Пока непонятно, какое циклическое расписание приведёт к самой плотной упаковке. Но, как сказано ранее, в классе алгоритмов, включающих построение циклического расписания с дальнейшим уплотнением, достижение оптимума не гарантировано. Это косвенно говорит о том, что невозможно построить алгоритм минимизации $C_{\max}$, трудоёмкость которого не зависит от числа деталей.

Циклические расписания могут существенно помочь, когда количество деталей, одновременно находящихся в обработке, ограничено некоторой величиной [23]. Приведённый в начале пункта пример с $(2k + 1)$ деталями особенно важен в случае, когда число деталей, одновременно находящихся в обработке, ограничено некоторой величиной H . Для обеспечения наибольшей производительности линии необходимо запускать в производство сразу $k + 1$ деталей, и эта величина может существенно превосходить H . Пример циклического расписания для детали с шестью операциями при $H = 2$ приведён на рис. 9. Операции O_1, O_3, O_5 выполняются на первой машине, а O_2, O_4, O_6 — на второй. Минимально возможная длина цикла равна суммарной длительности операций одной детали на самой загруженной машине.

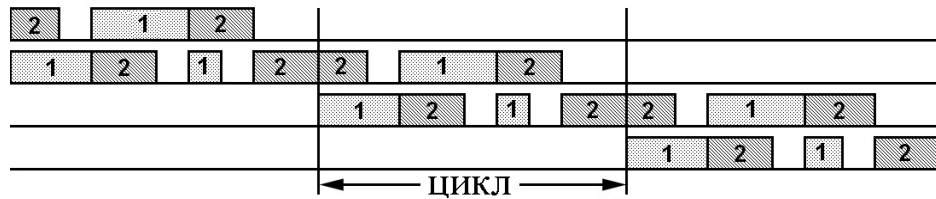


Рис. 9. Фрагмент циклического расписания

Теорема 3. При $H = 2$ циклическое расписание с минимальным временем цикла после уплотнения даёт оптимальное решение для критерия $C_{\max}$.

Доказательство основано на том, что при $H = 2$ операции, не вошедшие в основной цикл, удаётся уплотнить, и при этом достигается нижняя оценка оптимального времени обработки партии деталей. При $H = 2$ задача полиномиально разрешима.

Следствие 1. Задача $F|reentrant, p_{ij}=p_i, H=2|C_{\max}$ полиномиально разрешима.

При $H = 3$ операции, не вошедшие в полный цикл, не всегда удаётся уплотнить до состояния, когда будет достигнута нижняя оценка. Кроме того, вопрос о вычислительной сложности задачи построения оптимального циклического расписания для $H = 3$ до сих пор остаётся открытым.

Заключение

Исследуется вычислительная сложность задачи минимизации общего времени обработки идентичных деталей со сложным технологическим маршрутом, когда возможно неоднократное поступление деталей на некоторые машины. Доказано, что задача $F|reentrant, p_{ij} = p_i|C_{\max}$ является NP-трудной в обычном смысле. Предложен алгоритм построения точного решения задачи. При фиксированном числе деталей N алгоритм является псевдополиномиальным и на его основе может быть построена вполне полиномиальная аппроксимационная схема. Если зафиксировать технологический маршрут и длительности операций, то в задаче остаётся единственный параметр — число деталей N , и длина входа составит $O(\log_2 N)$. Такая ситуация более характерна для реального производства. В этом случае вопрос о вычислительной сложности остаётся открытым. Исследуется взаимосвязь данной задачи с задачей построения циклических расписаний с минимальным временем цикла, которая полиномиально разрешима. Построены оценки возможного отклонения решения, полученного в классе циклических расписаний, от оптимального значения для критерия $C_{\max}$.

ЛИТЕРАТУРА

1. Pinedo M. L. Scheduling. Theory, Algorithms, and Systems. Springer, 2008. 671 p.
2. Graves S. C., Meal H. M., Stefek D., and Zeghmi A. H. Scheduling of re-entrant flow shops // J. Oper. Management. 1983. V. 3. No. 4. P. 197–207.
3. Emmons H. and Vairaktarakis G. Flow Shop Scheduling: Theoretical Results, Algorithms, and Applications. Springer Science & Business Media, 2012. 334 p.
4. Shufan E., Grinshpoun T., Ikar E., and Ilani H. Reentrant flow shop with identical jobs and makespan criterion // J. Production Res. 2021. V. 61. No. 1. P. 183–197.
5. Lev V. and Adiri I. V-shop scheduling // Europ. J. Oper. Res. 1984. V. 18. No. 1. P. 51–56.
6. Pan J. C.-H. and Chen J.-S. Minimizing makespan in re-entrant permutation flow-shops // J. Oper. Res. Soc. 2003. V. 54. No. 6. P. 642–653.
7. Chen J.-S. A branch and bound procedure for the reentrant permutation flowshop scheduling problem // Intern. J. Adv. Manufacturing Technology. 2006. V. 29. P. 1186–1193.

8. Wang M. Y., Suresh P. S., and van de Velde S. L. Minimizing makespan in a class of reentrant shops // Oper. Res. 1997. V. 45. No. 5. P. 702–712.
9. Kubiak W., Lou Sh. X. C., and Wang Y. Mean flow time minimization in reentrant job shops with a hub // Oper. Res. 1996. V. 44. No. 5. P. 743–753.
10. Xie X., Tang L., and Li Y. Scheduling of a hub reentrant job shop to minimize makespan // Intern. J. Adv. Manufacturing Technology. 2011. V. 56. P. 743–753.
11. Middendorf M. and Timkovsky V. G. On scheduling cycle shops: Classification, complexity and approximation // J. Scheduling. 2002. V. 5(2). P. 135–169.
12. Timkovsky V. G. Cycle Shop Scheduling / Leung J. Y.-T. (ed.). Handbook of Scheduling. Ch. 7. Boca Raton; London; N.Y.; Washington, CRC Press, 2004.
13. Yu T.-S. and Pinedo M. L. Flow shops with reentry: Reversibility properties and makespan optimal schedules // Europ. J. Oper. Res. 2021. V. 282(2). P. 478–490.
14. Kats V. and Levner E. Minimizing the number of robots to meet a given cyclic schedule // Ann. Oper. Res. 1997. V. 69. P. 209–226.
15. Kats V. and Levner E. Cyclic scheduling in a robotic production line // J. Scheduling. 2002. V. 5(1). P. 23–41.
16. Boudoukh T., Penn M., and Weiss G. Scheduling jobshops with some identical or similar jobs // J. Scheduling. 2001. V. 4(4). P. 177–199.
17. Межецкая М. А., Сервах В. В. Задачи обработки деталей со сложным технологическим маршрутом // Современные проблемы науки и образования. 2013. № 1. <https://science-education.ru/ru/article/view?id=8407>.
18. Sotskov Y. N. and Shakhlevich N. V. NP-hardness of shop-scheduling problems with three jobs // Discrete Appl. Math. 1995. V. 59(3). P. 237–266.
19. Servakh V. V. and Shcherbinina T. A. A fully polynomial time approximation scheme for two project scheduling problems // IFAC Proc. Volumes. 2006. V. 39. Iss. 3. P. 131–135.
20. Сервах В. В. Эффективно разрешимый случай задачи календарного планирования с возобновляемыми ресурсами // Дискретн. анализ исслед. опер. Сер. 2. 2000. Т. 7. № 1. С. 75–82.
21. Middendorf M. and Timkovsky V. G. Transversal graphs for partially ordered sets: Sequencing, merging and scheduling problems // J. Combin. Optimization. 1999. V. 3. No. 4. P. 417–435.
22. Romanova A. A. and Servakh V. V. Optimization of identical jobs production on the base of cyclic schedules // J. Appl. Industr. Math. 2009. V. 3. No. 4. P. 496–504.
23. Боброва Е. А., Романова А. А., Сервах В. В. Сложность задачи построения циклических расписаний обработки однотипных деталей // Дискретн. анализ исслед. опер. 2013. Т. 20. № 4. С. 3–14.

REFERENCES

1. Pinedo M. L. Scheduling. Theory, Algorithms, and Systems. Springer, 2008. 671 p.
2. Graves S. C., Meal H. M., Stefek D., and Zeghmi A. H. Scheduling of re-entrant flow shops. J. Oper. Management, 1983, vol. 3, no. 4, pp. 197–207.
3. Emmons H. and Vairaktarakis G. Flow Shop Scheduling: Theoretical Results, Algorithms, and Applications. Springer Science & Business Media, 2012. 334 p.
4. Shufan E., Grinshpoun T., Ikar E., and Ilani H. Reentrant flow shop with identical jobs and makespan criterion. J. Production Res., 2021, vol. 61, no. 1, pp. 183–197.
5. Lev V. and Adiri I. V-shop scheduling. Europ. J. Oper. Res., 1984, vol. 18, no. 1, pp. 51–56.
6. Pan J. C.-H. and Chen J.-S. Minimizing makespan in re-entrant permutation flow-shops. J. Oper. Res. Soc., 2003, vol. 54, no. 6, pp. 642–653.

7. *Chen J.-S.* A branch and bound procedure for the reentrant permutation flowshop scheduling problem. *Intern. J. Adv. Manufacturing Technology*, 2006, vol. 29, pp. 1186–1193.
8. *Wang M. Y., Suresh P. S., and van de Velde S. L.* Minimizing makespan in a class of reentrant shops. *Oper. Res.*, 1997, vol. 45, no. 5, pp. 702–712.
9. *Kubiak W., Lou Sh. X. C., and Wang Y.* Mean flow time minimization in reentrant job shops with a hub. *Oper. Res.*, 1996, vol. 44, no. 5, pp. 743–753.
10. *Xie X., Tang L., and Li Y.* Scheduling of a hub reentrant job shop to minimize makespan. *Intern. J. Adv. Manufacturing Technology*, 2011, vol. 56, pp. 743–753.
11. *Middendorf M. and Timkovsky V. G.* On scheduling cycle shops: Classification, complexity and approximation. *J. Scheduling*, 2002, vol. 5(2), pp. 135–169.
12. *Timkovsky V. G.* Cycle Shop Scheduling. *Leung J. Y.-T. (ed.). Handbook of Scheduling. Ch. 7.* Boca Raton; London; N.Y.; Washington, CRC Press, 2004.
13. *Yu T.-S. and Pinedo M. L.* Flow shops with reentry: Reversibility properties and makespan optimal schedules. *Europ. J. Oper. Res.*, 2021, vol. 282(2), pp. 478–490.
14. *Kats V. and Levner E.* Minimizing the number of robots to meet a given cyclic schedule. *Ann. Oper. Res.*, 1997, vol. 69, pp. 209–226.
15. *Kats V. and Levner E.* Cyclic scheduling in a robotic production line. *J. Scheduling*, 2002, vol. 5(1), pp. 23–41.
16. *Boudoukh T., Penn M., and Weiss G.* Scheduling jobshops with some identical or similar jobs. *J. Scheduling*, 2001, vol. 4(4), pp. 177–199.
17. *Mezhetskaya M. A. and Servakh V. V.* Zadachi obrabotki detaley so slozhnym tekhnologicheskim marshrutom [The shop scheduling problems with complex technological route]. *Sovremennye Problemy Nauki i Obrazovaniya*, 2013, no. 1, <https://science-education.ru/ru/article/view?id=8407>. (in Russian)
18. *Sotskov Y. N. and Shakhlevich N. V.* NP-hardness of shop-scheduling problems with three jobs. *Discrete Appl. Math.*, 1995, vol. 59(3), pp. 237–266.
19. *Servakh V. V. and Shcherbinina T. A.* A fully polynomial time approximation scheme for two project scheduling problems. *IFAC Proc. Volumes*, 2006, vol. 39, iss. 3, pp. 131–135.
20. *Servakh V. V.* Effektivno razreshimyy sluchay zadachi kalendarnogo planirovaniya s vozobnovimymi resursami [An efficiently solvable case of the scheduling problem with renewable resources]. *Diskretnyy Analiz i Issledovaniye Operatsiy, Ser. 2*, 2000, vol. 7, no. 1, pp. 75–82. (in Russian)
21. *Middendorf M. and Timkovsky V. G.* Transversal graphs for partially ordered sets: Sequencing, merging and scheduling problems. *J. Combinat. Optimization*, 1999, vol. 3, no. 4, pp. 417–435.
22. *Romanova A. A. and Servakh V. V.* Optimization of identical jobs production on the base of cyclic schedules. *J. Appl. Industr. Math.*, 2009, vol. 3, no. 4, pp. 496–504.
23. *Bobrova E. A., Romanova A. A., and Servakh V. V.* Slozhnost zadachi postroyeniya tsiklicheskih raspisaniy obrabotki odnotipnykh detaley [Complexity of cyclic scheduling for identical jobs.] *Diskretnyy Analiz i Issledovaniye Operatsiy*, 2013, vol. 20, no. 4, pp. 3–14. (in Russian)