ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Июль—сентябрь 2006. Серия 1. Том 13, № 3. 83–102

УДК 519.8

О НЕКОТОРЫХ СВОЙСТВАХ ОПТИМАЛЬНЫХ РАСПИСАНИЙ В ЗАДАЧЕ ДЖОНСОНА С ПРЕРЫВАНИЯМИ*)

С. В. Севастьянов, Д. А. Чемисова, И. Д. Черных

Исследуются свойства оптимальных расписаний в NP-трудной задаче Джонсона с разрешением прерываний операций. В частности, доказано, что длина оптимального расписания всегда совпадает с суммарной длиной некоторого подмножества операций. С использованием найденных свойств показано, что оптимальное расписание любого примера может быть построено жадным алгоритмом (при подходящем задании приоритетов операций). Впервые описан алгоритм точного решения указанной задачи. Показано, что число прерываний в любом жадном расписании (а следовательно, и в оптимальном) не превосходит числа операций, что существенно лучше известных оценок числа прерываний в оптимальном расписании.

Введение

Под *задачей Джонсона* понимается задача Flow Shop на минимум длины расписания без прерываний операций. В этой статье рассматривается задача *с разрешением прерываний* операций, которая названа задачей $\mathcal{D}$. Согласно принятой классификации задач [3] она обозначается как $\mathrm{F}|\mathrm{pmtn}|\mathrm{C}_{\max}$.

Задача $\mathcal{D}$. Имеется множество машин $\mathcal{M} = \{M_1, \ldots, M_m\}$ и множество работ $\mathcal{J} = \{J_1, \ldots, J_n\}$. Каждая работа $J_j \in \mathcal{J}$ состоит из m операций O_{ij} $(i = 1, \ldots, m)$, которые должны выполняться в предписанном порядке $O_{1j} \prec \ldots \prec O_{mj}$; здесь $O' \prec O''$ означает, что операция O'' может начаться не раньше, чем закончится операция O'. Операция O_{ij} выполняется на машине M_i за время p_{ij}, причём никакие две операции, выполняемые на одной машине, не могут выполняться одновременно в течение ненулевого отрезка времени. Разрешаются прерывания любой

*) Исследование выполнено при финансовой поддержке Российского фонда фундаментальных исследований (проект 05–01–00960).

начатой операции; при этом сумма длин интервалов выполнения операции должна равняться её длительности. Требуется построить расписание S, минимизирующее время $C_{\max}(S)$ завершения всех работ.

Хотя эта задача известна давно (её первые исследования были выполнены Т. Гонзалезом и С. Сани в [4] более четверти века назад), она остаётся сравнительно малоизученной. Действительно, до сих пор в литературе не встречалось описания какого-либо алгоритма её точного решения (произвольной вычислительной сложности), что косвенно подтверждает не только вычислительную сложность её точного решения, но и конструктивную, связанную с проблемой описания конечного подмножества допустимых решений, содержащего хотя бы одно оптимальное решение.

В статье проводится исследование структурных свойств оптимальных расписаний задачи $\mathcal{D}$. Найденные свойства позволяют сделать вывод, что оптимальное расписание может быть построено с использованием жадного алгоритма при подходящем задании приоритетов операций. Таким образом, впервые описан алгоритм, гарантирующий точное решение задачи $\mathcal{D}$ за конечное время.

В [4] доказано, что задача $\mathrm{F3}|\mathrm{pmtn}|C_{\max}$ NP-трудна в сильном смысле. Поэтому актуально построение эффективных алгоритмов приближённого решения задачи $\mathcal{D}$ с гарантированными оценками точности. Однако в этой статье мы не будем касаться этого направления.

Помимо минимизации длины расписания рассмотрена минимизация числа прерываний в оптимальном расписании (рассматривая этот параметр как вторую целевую функцию). В [1, с. 39] было показано, что оптимальное расписание можно искать в классе расписаний, в которых число прерываний не превосходит $mn(n-1)$. В настоящей статье существенно улучшена эта оценка и показано, что при подходящем задании приоритетов в жадном алгоритме получается оптимальное расписание не более чем с $(m-2)(n-1)$ прерываниями.

Наконец, из полученных нами структурных свойств оптимальных расписаний вытекает новое интересное свойство, согласно которому длина оптимального расписания любого примера совпадает с суммарной длиной операций из некоторого подмножества. Это свойство довольно естественно для задач без разрешения прерываний операций, однако для задачи с разрешением прерываний оно несколько неожиданно. Достаточно сказать, что уже для простейших примеров задачи Dag Shop с прерываниями (и целочисленными длительностями операций) это свойство не только не выполняется, но не гарантируется целочисленность оптимума.

Статья организована следующим образом. В разделе 1 вводятся ис-

ходные понятия и обозначения. В разделе 2 исследуются свойства оптимальных расписаний задачи $\mathcal{D}$. В разделе 3 описывается жадный алгоритм построения допустимых расписаний для задачи $\mathcal{D}$ и исследуются свойства получаемых расписаний. В частности, показывается, что одно из получаемых расписаний является оптимальным. Приводится оценка числа прерываний в таком расписании. В разделе 4 проводится анализ свойств критических путей в жадном расписании. В частности, устанавливается существование полного критического пути, откуда вытекает свойство «целостности» оптимума любого примера задачи $\mathcal{D}$, т. е. длина оптимального расписания совпадает с суммарной длительностью операций некоторого подмножества.

1. Исходные понятия и обозначения

Через $\mathcal{P}_n$ будем обозначать множество перестановок индексов $\{1,\ldots, n\}$, а через $\mathcal{I}_{m,n}$ — множество всех примеров задачи $\mathcal{D}$ с m машинами и n работами.

Определение 1. Часть операции, выполняемую в заданном расписании без прерываний, будем называть *фрагментом* (операции). Произвольный фрагмент длины ε будем называть ε-фрагментом, 0-фрагмент будем называть *нулевым*, а максимальный по включению фрагмент — *максимальным фрагментом*.

Расписание для операции O_{ij} задается парой $(k_{ij}, \mathcal{T}_{ij})$, где k_{ij} — число максимальных фрагментов, на которые разбивается операция, а $\mathcal{T}_{ij} = \{[s_{ij}^k, c_{ij}^k] \,|\, k = 1,\ldots,k_{ij}\}$ — семейство интервалов времени, выделяемых для выполнения этих фрагментов $(0 \leqslant s_{ij}^1 \leqslant c_{ij}^1 \leqslant \cdots \leqslant s_{ij}^{k_{ij}} \leqslant c_{ij}^{k_{ij}})$. Ясно, что в допустимом расписании суммарная длина фрагментов операции должна равняться её длине: $\sum\limits_{k=1}^{k_{ij}} (c_{ij}^k - s_{ij}^k) = p_{ij}$. Через $|O'|, s_S(O')$ и $c_S(O')$ будем также обозначать длину фрагмента O' и моменты его начала и окончания в заданном расписании S; $s_S(O_{ij}) = s_{ij}^1$ и $c_S(O_{ij}) = c_{ij}^{k_{ij}}$ обозначают моменты начала и окончания операции O_{ij}. *Полное расписание* S определяется как совокупность расписаний по всем операциям: $S \doteq \{(k_{ij}, \mathcal{T}_{ij}) \,|\, 1 \leqslant i \leqslant m,\ 1 \leqslant j \leqslant n\}$. Через $C_{\max}(S) = \max\limits_{i,j}\ c_S(O_{ij})$ обозначаем *длину расписания* S.

В случае, когда прерывания запрещены, для задания расписания достаточно задать лишь моменты начала всех операций:
$$S \doteq \{s(O_{ij}) \,|\, 1 \leqslant i \leqslant m,\ 1 \leqslant j \leqslant n\}.$$

Определение 2. Точку t будем называть *точкой выполнения* опе-

рации O_{ij} в расписании S, если она принадлежит одному из интервалов семейства $\mathcal{T}_{ij}$, и называем *внутренней точкой* операции O_{ij}, если она является внутренней точкой одного из интервалов семейства $\mathcal{T}_{ij}$.

Ясно, что в одной точке t могут одновременно выполняться две, а с учетом существования нулевых операций — и более двух операций на одной машине. Однако никакая внутренняя точка операции O_{ij} не может являться точкой выполнения другой операции машины M_i.

Сформулируем задачу (будем называть её *базовой*) на составление расписания выполнения *заданий* из заданного множества $\mathcal{J}$.

Базовая задача. Имеется конечное множество $\mathcal{J}$ *заданий*, которые требуется выполнить без прерываний, и задан взвешенный орграф $G_p = (\mathcal{J}, U)$; каждой вершине $j \in \mathcal{J}$ приписан вес p_j. Расписание $S = \{s_S(j) \geqslant 0 \mid j \in \mathcal{J}\}$ выполнения заданий из множества $\mathcal{J}$ называется *допустимым* (относительно графа G_p), если для любой дуги $(i,j) \in U$ выполняется неравенство $s_S(j) \geqslant s_S(i) + p_i$. Величина $C_{\max}(S) \doteq \max\limits_{i \in \mathcal{J}}(s_S(i) + p_i)$ называется *длиной* расписания S. Требуется построить допустимое расписание минимальной длины.

Определение 3. Допустимое расписание S' для базовой задачи с заданным графом $G_p = (\mathcal{J}, U)$ называется *ранним* относительно графа G_p, если для любого расписания S, допустимого относительно G_p, и для любого задания $j \in \mathcal{J}$ выполняется соотношение $s_{S'}(j) \leqslant s_S(j)$.

Определение 4. Пусть $G_p = (\mathcal{J}, U)$ — орграф, вершинам которого приписаны веса. *Путём* $C = (j_1 \to j_2 \to \ldots \to j_k)$ в графе G_p называется такая последовательность вершин $j_i \in \mathcal{J}$, что $(j_i, j_{i+1}) \in U$ при каждом $i = 1, \ldots, k-1$. *Длиной* пути $C \subset G_p$ называется сумма весов вершин пути, обозначаемая через $|C|$. *Критическим путём* в графе G_p называется путь максимальной длины.

Из календарного планирования известен следующий результат.

Лемма 1. *Для любого взвешенного графа G_p без контуров положительной длины существует единственное раннее относительно G_p расписание. Оно оптимально для базовой задачи с графом G_p, а его длина совпадает с длиной критического пути в G_p.*

Определение 5. Отношение “$\prec$”, определенное на множестве пар элементов из V, называется отношением (частичного) *строгого порядка*, если оно иррефлексивно и транзитивно. Следовательно, для любой пары $v', v'' \in V$ выполняется не более одного из двух отношений: $v' \prec v''$ и $v'' \prec v'$ (т. е. отношение антисимметрично).

Будем говорить, что граф $G_p = (V, U)$ задает *допустимую схему расписания* для заданного примера $I = (p_{ij}) \in \mathcal{I}_{m,n}$ задачи $\mathcal{D}$, если:

множество вершин V является объединением непересекающихся подмножеств V_{ij} $(i = 1, \ldots, m;\ j = 1, \ldots, n)$;

вершинам $v \in V$ приписаны неотрицательные веса $p(v)$ такие, что $\sum\limits_{v \in V_{ij}} p(v) = p_{ij}$;

множество U состоит из дуг (v', v''), определённых для следующих пар вершин:

a) $v' \in V_{i,j},\ v'' \in V_{i+1,j}$ $(i = 1, \ldots, m-1;\ j = 1, \ldots, n)$;

b) на множестве пар вершин из $V_i \doteq \bigcup\limits_{j=1}^{n} V_{ij}$ $(i = 1, \ldots, m)$ определено отношение “$\prec$” линейного строгого порядка; для $v', v'' \in V_i$ множество U включает дугу (v', v'') тогда и только тогда, когда $v' \prec v''$.

Множество $\mathcal{G}(I)$ состоит из всех взвешенных орграфов $G_p = (V, U)$, задающих допустимые схемы расписаний примера I. Для заданного графа G_p элементы каждого множества V_{ij} ассоциируются с максимальными фрагментами операции O_{ij} в некотором расписании, а веса $p(v)$ вершин $v \in G_p$ — с длинами этих фрагментов. Таким образом, расписание S для базовой задачи с графом G_p может рассматриваться как расписание для примера I задачи $\mathcal{D}$. Справедливо

Утверждение 1. *Для любого входа $I \in \mathcal{I}_{m,n}$ и любого графа $G_p \in \mathcal{G}(I)$ любое расписание S, допустимое для базовой задачи с графом G_p, является допустимым и для примера I задачи $\mathcal{D}$.*

Пусть S — допустимое расписание для заданного примера I задачи $\mathcal{D}$. Оно задаёт разбиение операций на максимальные фрагменты и для каждой машины задаёт порядок выполняемых на ней фрагментов. Пусть O', O'' — различные максимальные фрагменты, выполняемые на одной машине. Будем говорить, что фрагмент O' *строго предшествует* фрагменту O'' в расписании S (обозначается $O' \prec_S O''$), если O' начинается раньше, чем заканчивается O''. Нетрудно видеть, что отношение “$\prec_S$” является отношением “почти линейного” строгого порядка на множестве фрагментов операций, выполняемых на одной машине: оно транзитивно и определено для любой пары различных фрагментов O', O'' — за исключением пар нулевых фрагментов O', O'', выполняемых одновременно. Чтобы получить из $\prec_S$ отношение линейного порядка $\prec$ на множестве максимальных фрагментов машины M_i, на каждом (максимальном по включению) подмножестве её нулевых максимальных фрагментов, выполняемых одновременно, достаточно выбрать какой-либо вариант ли-

нейного порядка. Определим взвешенный граф $G_p(S) = (V(S), U(S))$, вершинами которого являются все максимальные фрагменты операций в расписании S. Вершине $v \in V(S)$ приписывается вес, равный длине данного фрагмента. Для пары фрагментов $O', O'' \in V(S)$ определяем дугу $(O', O'') \in U(S)$ в двух случаях:

a) если $O' \in O_{i,j}$ и $O'' \in O_{i+1,j}$ для некоторых i, j;

b) если O' и O'' выполняются на одной машине и $O' \prec O''$.

Нетрудно видеть, что так определённый граф $G_p(S)$ принадлежит $\mathcal{G}(I)$ (т. е. задаёт допустимую схему расписания исходного примера I), и расписание S является допустимым решением базовой задачи с графом $G_p(S)$. Из этого замечания и утверждения 1 следует, что множества допустимых расписаний примера I исходной задачи и базовой задачи (по отношению ко всей совокупности графов $G_p \in \mathcal{G}(I)$) совпадают. Поэтому решение исходной задачи сводится к решению совокупности базовых задач относительно семейства графов $G_p \in \mathcal{G}(I)$ и выбору из найденных оптимальных (для базовых задач) решений наилучшего по критерию минимум $C_{\max}$.

Поскольку любой граф $G_p \in \mathcal{G}(I)$ является ациклическим, из леммы 1 следует, что решением базовой задачи для любого $G_p \in \mathcal{G}(I)$ является расписание, раннее относительно G_p, и что такое расписание существует и единственно. Тем самым справедливо

Утверждение 2. *Для отыскания оптимального решения задачи $\mathcal{D}$ с заданным входом $I \in \mathcal{I}_{m,n}$ для каждого $G_p \in \mathcal{G}(I)$ достаточно построить раннее расписание базовой задачи с графом G_p и из множества полученных расписаний выбрать кратчайшее.*

Однако утверждение 2 ещё не даёт конструктивного способа решения задачи $\mathcal{D}$, поскольку множество $\mathcal{G}(I)$ бесконечно (для заданного примера I допускается неограниченное число вариантов разбиения его операций на фрагменты). Для преодоления этой бесконечности нам потребуется дальнейший анализ свойств оптимальных расписаний.

2. Анализ свойств оптимальных расписаний в задаче F|pmtn|$C_{\max}$

Естественно, что для исследования какого-либо объекта (в нашем случае — оптимального расписания) прежде всего необходимо убедиться в его существовании. Существование оптимального расписания в задаче $\mathcal{D}$ будем выводить из полученного ранее результата (см. теорему 1).

Определение 6. Функция $F(x_1, \ldots, x_n)$ называется *регулярной*, если она не убывает по каждой переменной и непрерывна слева.

В [2] была сформулирована общая задача GP с ξ операциями $\{O_1, \ldots, O_\xi\}$ с допущением прерываний операций и k регулярными целевыми функциями $F_1(C), \ldots, F_k(C)$, зависящими от вектора $C(S) = (c(O_1), \ldots, c(O_\xi))$ моментов завершения операций в заданном расписании S. Доказана

Теорема 1[2]. *Для любого примера задачи GP с непустым множеством допустимых решений существует допустимое расписание S, лексикографически минимизирующее вектор-функцию $(F_1(C(S)), \ldots, F_k(C(S)))$.*

Так как задача $\mathcal{D}$ является частным случаем задачи GP и целевая функция $C_{\max}(C(S))$ регулярна, из теоремы 1 следует существование оптимального расписания задачи $\mathcal{D}$.

Определение 7. Будем говорить, что в расписании S работы J_{j_1} и J_{j_2} $(j_1 \neq j_2)$ выполняются на машине M_i с *перекрёстом*, если найдутся фрагменты O'_{ij_1}, O''_{ij_1} операции O_{ij_1} и фрагменты O'_{ij_2}, O''_{ij_2} операции O_{ij_2}, выполняемые в порядке $O'_{ij_1} \prec_S O'_{ij_2} \prec_S O''_{ij_1} \prec_S O''_{ij_2}$.

Определение 8. *Точками переключения* в расписании S называем моменты времени, в которые происходит изменение выполняемой работы хотя бы на одной из машин (в частности, простой машины считается выполнением "пустой" работы). Таким образом, точками переключения являются моменты прерывания, возобновления, начала и завершения операций.

Определение 9. Момент t будем называть моментом *w-перехода* работы J_j с машины M_i на машину M_{i+1} в расписании S, если $t = c_S(O_{ij}) = s_S(O_{i+1,j})$. При этом работу J_j и операции $O_{ij}, O_{i+1,j}$ будем называть *переходными*.

Определение 10. Машины M_1 и M_m будем называть *крайними*, а $M_2, \ldots, M_{m-1}$ — *средними*.

Определение 11. Пусть S — частичное расписание. Если момент $c_S(O_{ij})$ завершения операции O_{ij} в расписании S неопределён, полагаем $c_S(O_{ij}) = \infty$. Определим момент $r_S(O_{ij})$ *начала готовности операции O_{ij} к выполнению* в расписании S: $r_S(O_{ij}) = 0$ при $i = 1$; $r_S(O_{ij}) = c_S(O_{i-1,j})$ при $i > 1$. Операцию O_{ij} будем называть *готовой к выполнению* в момент времени $t \geqslant 0$, если $r_S(O_{ij}) \leqslant t \leqslant c_S(O_{ij})$.

Определение 12. Расписание S для примера I называется *ранним*, если оно раннее относительно графа $G_p(S)$.

Определение 13. Допустимое расписание S задачи $\mathcal{D}$ называется *плотным*, если не существуют непустой интервал $(t_1, t_2) \subseteq (0, C_{\max}(S))$ и машина M_i такие, что во всём интервале (t_1, t_2) машина M_i простаивает и для каждого $t \in (t_1, t_2)$ множество её операций, готовых к выполнению в момент t, непусто.

Замечание 1. Всякое плотное расписание S является ранним. Обратное неверно: на рис. 1 изображено расписание S выполнения двух работ на двух машинах, раннее относительно графа $G_p(S)$, но неплотное.

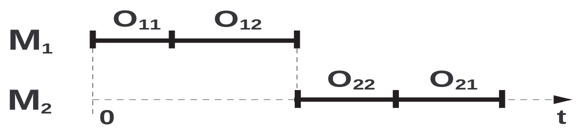

Рис. 1. Ранее расписание, не являющееся плотным

Определение 14. *Несущественными фрагментами* в заданном расписании S называем все нулевые максимальные фрагменты за исключением первого фрагмента каждой нулевой операции.

Замечание 2. Из любого допустимого расписания можно удалить все несущественные фрагменты без ущерба для качества расписания, т. е.: а) с сохранением допустимости расписания; б) без увеличения его длины; в) с уменьшением числа прерываний операций. Поэтому мы будем предполагать, что *все рассматриваемые далее расписания не содержат несущественных фрагментов*. В то же время мы обязаны учитывать нулевые операции: хотя они не оказывают влияния на длину расписания, они могут играть существенную роль при подсчёте числа прерываний операций.

Лемма 2. *Для любого примера $I \in \mathcal{I}_{m,n}$ задачи $\mathcal{D}$ существует оптимальное расписание S, обладающее следующими свойствами*:

(1*) *Расписание S плотное.*

(2*) *Всякая операция O_{ij}, фрагмент которой выполняется на машине M_i между двумя фрагментами O' и O'' другой операции $O_{ij'}$, становится готовой к выполнению не ранее чем с момента завершения фрагмента $O' : r_S(O_{ij}) \geqslant c_S(O')$. Как следствие, в S нет перекрёстов.*

(3*) *Каждое прерывание на средней машине M_i происходит в момент w-перехода некоторой работы J_j с машины M_{i-1} на машину M_i.*

(4*) *На машинах M_1 и M_m все операции выполняются без прерываний.*

(5*) *На M_m работы выполняются в порядке их окончания на машине M_{m-1}.*

(6*) *На M_1 работы выполняются в порядке их начала на машине M_2.*

(7*) *Расписание лексикографически минимизирует вектор-функцию $(F_1(C), F_2(C))$, где $F_1(C) = C_{\max}$, $F_2(C) = \sum\limits_{i=2}^{m-1} \sum\limits_{j=1}^{n} c(O_{ij})$ — сумма времён окончания операций на средних машинах, а C — вектор моментов окончания операций.*

(8*) *Каждая нулевая операция O_{ij} выполняется на средней машине M_i в момент w-перехода работы J_j с M_{i-1} на M_i.*

Будем ссылаться на совокупность свойств (1*)–(5*) как на ($*$), а на совокупность (1*)–(8*) как на ($**$).

Доказательство. Рассмотрим расписание S, лексикографически минимизирующее вектор-функцию $F(C) = (F_1(C), F_2(C), F_3(C))$, где $F_3(C) = \sum\limits_{i,j} c\,(O_{ij})$ — сумма моментов окончания всех операций. Поскольку все три функции регулярны, такое расписание S существует согласно теореме 1. Ясно, что оно оптимально, и что выполняется свойство (7*). Свойства (1*) и (8*) следуют из минимальности вектор-функции. Действительно, если бы какая-то нулевая операция O_{ij} на средней машине M_i выполнялась позже момента t окончания операции $O_{i-1,j}$, то перенеся её выполнение на момент t, можно уменьшить как $F_2(C)$, так и $F_3(C)$, не увеличивая $F_1(C)$. Аналогично, если бы нарушилась плотность расписания S в непустом интервале времени (t_1, t_2) (некоторая машина M_i простаивает, но имеется готовая для выполнения ненулевая операция O_{ij}), мы могли бы перенести в этот интервал ненулевой завершающий фрагмент операции O_{ij}, уменьшив при этом время её завершения и компоненту $F_3(C)$ и не уменьшив $F_1(C)$ и $F_2(C)$, что противоречит выбору расписания S.

Пусть не выполняется свойство (2*), т. е. фрагмент O'_{ij} операции O_{ij} выполняется между фрагментами O' и O'' операции $O_{ij'}$ и $r_S(O_{ij}) < c_S(O')$. Если предположить, что O_{ij} — нулевая операция, то перенеся её выполнение на момент $r_S(O_{ij})$, мы смогли бы уменьшить $F_3(C)$, не нарушая допустимости расписания и не увеличивая $F_1(C)$ и $F_2(C)$, что противоречит выбору расписания S. Пусть далее O_{ij} — ненулевая операция, и следовательно, по замечанию 2 её фрагмент O'_{ij} также ненулевой. Без ограничения общности можно считать, что $c_S(O_{ij}) < c_S(O_{ij'})$ и O'_{ij} — последний фрагмент операции O_{ij} (в противном случае, т. е. при $c_S(O_{ij'}) < c_S(O_{ij})$, приходим к искомой ситуации, поменяв ролями операции O_{ij} и O'_{ij}). Пусть $\varepsilon \doteq \min\{|O'|, |O'_{ij}|, c_S(O') - r_S(O_{ij})\}$. Тогда $\varepsilon > 0$.

Поменяв местами последние ε-подфрагменты фрагментов O'_{ij} и O', мы уменьшим время окончания операции O_{ij} как минимум на ε (с сохранением допустимости расписания), что приводит к лексикографическому уменьшению вектора F и противоречит выбору расписания S. Свойство (2*) доказано.

Докажем свойство (3*). Пусть в некоторый момент t на средней машине M_i произошло прерывание операции $O_{ij'}$. Так как по свойству (1*) сразу после момента t машина M_i не может простаивать, то в момент t на M_i начинает выполняться фрагмент O' некоторой операции $O_{ij''}$. Если $O_{ij''}$ — нулевая операция, то по свойству (8*) момент t является моментом w-перехода работы $J_{j''}$ с M_{i-1} на M_i, и свойство (3*) для данного случая выполнено. Если же O' — ненулевой фрагмент операции $O_{ij''}$, то из свойства (2*) следует, что $r_S(O_{ij''}) \geqslant t$, откуда следует (3*).

Теперь предположим, что расписание S не обладает какими-либо из свойств (4*)–(6*). Построим расписание S', которое на средних машинах совпадает с S, а на крайних машинах определяется согласно свойствам (1*) и (4*)–(6*). Ясно, что такое расписание существует и определяется "почти однозначно" (неоднозначность в определении порядка выполнения работ на M_1 и M_m может возникать при наличии нулевых операций на машинах M_2 и M_{m-1}). Докажем, что S' допустимо.

Предположим, что это не так, и пусть $\pi_1 = (\pi_1^1, \ldots, \pi_1^n)$ — порядок выполнения работ на машине M_1. Нарушение допустимости может произойти лишь в том случае, когда $c_{S'}(O_{1j}) > s_{S'}(O_{2j})$ для некоторой работы J_j $(j = \pi_1^k)$. Но поскольку расписание S допустимо, каждая из операций $O_{1\pi_1^\alpha}$ $(\alpha = 1, \ldots, k)$ завершается в S не позже момента $s_S(O_{2\pi_1^\alpha}) \leqslant s_S(O_{2\pi_1^k}) = s_{S'}(O_{2j})$. Отсюда получаем, что

$$\sum_{\alpha=1}^{k} |O_{1\pi_1^\alpha}| = c_{S'}(O_{1j}) \leqslant s_{S'}(O_{2j}).$$

Противоречие.

Покажем, что расписание S' (как и S) имеет минимальную длину. Ясно, что $C_{\max}(S)$ совпадает с оптимумом задачи $1|r_j,\mathrm{pmtn}|\ \mathrm{C}_{\max}$, на входе которой заданы длительности работ $p_j = |O_{mj}|,\ j = 1, \ldots, n$ и моменты поступления работ $r_j = c_S(O_{m-1,j})$. Как известно, оптимальным решением данной задачи является плотное расписание, в котором все работы выполняются без прерываний в порядке неубывания r_j. Но именно такое расписание мы имеем на машине M_m по построению S'.

Таким образом, в S' сохраняется свойство (7*). Покажем, что сохраняются свойства (1*)–(3*) и (8*), которыми обладало расписание S.

Плотность расписания S' (свойство (1*)), а также свойство (2*) на крайних машинах выполняются по построению S'. На машине M_2, в принципе, могло нарушиться любое из свойств (1*)–(3*), (8*) в том случае, если бы какая-то операция O_{1j} завершилась в S' раньше, чем в S. Однако если бы такое нарушение действительно произошло, это позволило бы выполнить преобразование расписания S' (как это было описано выше) с сохранением его допустимости и с уменьшением вектора F, что противоречит выбору расписания S. Таким образом, свойства (1*) – (3*) и (8*) сохраняются в расписании S'. Лемма 2 доказана.

3. Жадный алгоритм $\mathcal{A}_g$ и его свойства

Описываемый в этом разделе алгоритм $\mathcal{A}_g$ относится к семейству алгоритмов жадного типа. Свойства алгоритма $\mathcal{A}_g$ мы будем использовать в алгоритме точного решения задачи $\mathcal{D}$. Сначала сформулируем общие принципы работы жадного алгоритма при построении допустимого расписания в задаче $\mathcal{D}$, после чего уточним специфику применения такого алгоритма в рассматриваем нами случае.

Общие принципы работы жадного алгоритма при построении расписания в задаче $\mathcal{D}$

1. *Временна́я «прогонка» расписания.* Искомое *жадное* расписание S_g «восстанавливается» последовательно во времени: с возрастанием t оно становится полностью определённым в интервале $[0, t]$. Процесс восстановления расписания заканчивается при $t = C_{\max}(S_g)$. При этом момент t изменяется не непрерывно, а дискретно, перебирая все точки переключения расписания τ_i в порядке возрастания: $\tau_{i+1} > \tau_k$.

2. *Необратимость алгоритма.* Частичное расписание, определенное на k-м шаге алгоритма в интервале $[0, \tau_k]$, не изменяется на последующих шагах алгоритма.

3. *Плотность расписания.* Формально определение точки переключения τ_k (см. определение 8) включает в себя моменты перехода машины:

1) от выполнения одной операции к выполнению другой (при этом первая операция либо прерывается в момент τ_k, либо завершается);

2) после прерывания выполняемой операции к простою;

3) после завершения операции к простою;

4) от простоя к выполнению прерванной ранее операции;

5) от простоя к началу новой операции.

Однако для рассматриваемой задачи Flow Shop исключается появление некоторых из описанных выше ситуаций в *жадном расписании*, построенном при помощи жадного алгоритма. В частности, полностью

исключаются ситуации 2 и 4, а ситуация 5 (с началом некоторой операции O_{ij}) возможна лишь в момент окончания операции $O_{i-1,j}$. Все перечисленные ограничения проистекают из требования *плотности* строящегося расписания S_g (см. определение 13).

4. *Приоритетность.* В каждый момент времени t на машине M_i выполняется наиболее приоритетная операция из множества $\mathcal{O}_i(S_g, t)$ операций, готовых к выполнению на этой машине в момент t (см. определение 11). При этом *отношение приоритетности* на множестве всех операций каждой машины M_i задаётся априори (т. е. до построения расписания) и остаётся неизменным в течение всего времени работы алгоритма. В случае равенства приоритетов операции упорядочиваются по неубыванию моментов $r_S(O_{ij})$ их готовности к выполнению. Отсюда следует, что номер выполняемой на машине M_i операции может изменяться лишь в моменты изменения множества $\mathcal{O}_i(S_g, t)$, а точнее, лишь в двух случаях: а) когда завершается какая-то операция машины M_i; б) когда в множестве $\mathcal{O}_i(S_g, t)$ появляется новая операция с наибольшим приоритетом (что совпадает по времени с моментом завершения какой-то операции на машине M_{i-1}). Таким образом, множество всех точек переключения в жадном расписании совпадает с множеством моментов завершения операций в этом расписании (откуда, в частности, следует, что число точек переключения не превосходит числа операций). Из свойства приоритетности также вытекает следующее свойство.

5. *Причинная обусловленность прерываний.* Прерывание операции O_{ij} в жадном расписании возможно лишь в ситуации 1 и лишь в момент w-перехода какой-либо работы $J_{j'}$ $(j' \neq j)$ с машины M_{i-1} на машину M_i.

Специфика жадного алгоритма $\mathcal{A}_g(\pi^1, \ldots, \pi^{m-1}; I)$ состоит в том, что для заданного примера $I \in \mathcal{I}_{m,n}$ с помощью перестановок $\pi^1, \ldots, \pi^{m-1}$ из $\mathcal{P}_n$ задаются строгие приоритетные порядки выполнения операций на машинах $M_1, \ldots, M_{m-1}$: действительно, каждая перестановка π^i определяет перестановку номеров работ $\{1, \ldots, n\}$, а поскольку каждая работа J_j в модели Flow Shop представлена на каждой машине M_i единственной операцией O_{ij}, тем самым определяется приоритетный порядок на множестве операций машины M_i. Порядок является *строгим*: из любых двух операций одна всегда приоритетнее другой. В то же время всем операциям машины M_m приписываются равные априорные приоритеты. Как следствие, работы на машине M_m выполняются в порядке окончания их операций на машине M_{m-1} (что соответствует свойству (5*) леммы 2).

Лемма 3. *Для любого входа $I \in \mathcal{I}_{m,n}$ и любых перестановок $\pi^1, \ldots,$*

π^{m-1} *из* $\mathcal{P}_n$ *алгоритм* $\mathcal{A}_g(\pi^1, \ldots, \pi^{m-1}; I)$ *находит расписание* S_g, *удовлетворяющее свойствам* $(*)$ *леммы* 2.

Доказательство. Как было отмечено выше, свойство (5*) следует из специфики задания приоритета на машине M_m. Из принципа 3 следует *плотность* расписания S_g (свойство (1*)), а из принципа 5 — свойство (3*).

Докажем свойство (2*). Если операция O_{ij_1} была прервана в момент t, то по принципу 4 до момента её возобновления на M_i могут выполняться лишь более приоритетные чем O_{ij_1} операции. Если бы какая-то из операций O_{ij_2} имела момент готовности $r_{S_g}(O_{ij_2}) < t$, то это означало бы, что в интервале времени $(t - \varepsilon, t)$ выполняется не самая приоритетная операция O_{ij_1}, что противоречило бы принципу 4. Противоречие доказывает свойство (2*).

Далее, поскольку все работы готовы к выполнению на M_1 с момента 0, машина M_1 от начала до конца работает без простоев и выполняет работы согласно их априорному приоритетному порядку π^1 без прерываний. Кроме того, каждая работа, прибывающая на M_m, оказывается наименее приоритетной в списке готовых к выполнению работ (поскольку прибыла позже других), и следовательно, не может прервать текущую работу, выполняемую на M_m. Отсюда следует свойство (4*). Лемма 3 доказана.

Теорема 2. *Для любого входа* $I \in \mathcal{I}_{m,n}$ *существуют перестановки* $\pi^1, \ldots, \pi^{m-1}$ *из* $\mathcal{P}_n$ *такие, что получаемое согласно алгоритма* $\mathcal{A}_g(\pi^1, \ldots, \pi^{m-1}; I)$ *расписание* S_g *является оптимальным и удовлетворяет свойствам* $(**)$.

Доказательство. Пусть задан вход $I \in \mathcal{I}_{m,n}$. Рассмотрим оптимальное расписание S, удовлетворяющее свойствам $(**)$ (оно существует по лемме 2). На множестве операций машины M_k определим отношение "$\lhd_S$" следующим образом: $O_{kj'} \lhd_S O_{kj''}$, если $j' \neq j''$ и выполняется одно из соотношений

$$c_S(O_{kj'}) \leqslant s_S(O_{kj''}), \tag{1}$$

$$s_S(O_{kj''}) < s_S(O_{kj'}) \leqslant c_S(O_{kj'}) < c_S(O_{kj''}). \tag{2}$$

Так как по свойству (2*) в расписании S нет перекрёстов и несущественных фрагментов (замечание 2), то для любых двух операций $O_{kj'}, O_{kj''}$ имеет место одно из отношений: $O_{kj'} \lhd_S O_{kj''}$, либо $O_{kj''} \lhd_S O_{kj'}$, а для пар нулевых операций, выполняемых одновременно, имеют место оба отношения (поскольку для каждой пары выполняется (1)). Нетрудно убедиться, что $\lhd_S$ является отношением линейного нестрогого порядка на

множестве операций машины M_k, что позволяет упорядочить все операции машины M_k в цепочку: $O_{k\pi_1^k} \lhd_S O_{k\pi_2^k} \lhd_S \ldots \lhd_S O_{k\pi_n^k}$ для некоторой перестановки $\pi^k = (\pi_1^k, \ldots, \pi_n^k)$. (При наличии нескольких нулевых операций машины M_k, выполняемых одновременно в расписании S, перестановка π^k определяется неоднозначно. Однако для наших целей в качестве π^k годится любая из перестановок, определяемых отношением $\lhd_S$.) Используя определённые таким образом перестановки $\pi^1, \ldots, \pi^{m-1}$ в алгоритме $\mathcal{A}_g(\pi^1, \ldots, \pi^{m-1}; I)$, построим жадное расписание S_g и докажем его оптимальность.

Сначала индукцией по $k = 1, \ldots, m-1$ докажем, что на машинах M_k расписание S_g совпадает с S. Как было отмечено в доказательстве леммы 3, операции машины M_1 выполняются в расписании S_g без прерываний и без простоя согласно их приоритетам, определённым перестановкой π^1. Поскольку π^1 соответствует порядку выполнения работ на машине M_1 в расписании S, совпадение расписаний S и S_g на M_1 доказано, что обеспечивает базис индукции.

Пусть $1 < k < m$, и пусть совпадение расписаний S и S_g на машинах M_i $(i < k)$ доказано. Докажем, что S и S_g совпадают на машине M_k. Пусть это не так, и пусть t' — наименьший момент времени, начиная с которого расписания не совпадают. Это несовпадение может реализовываться одним из двух способов: (a) какая-то нулевая операция $O_{kj''}$, выполняемая в момент t' в одном из расписаний, выполняется в другом расписании в момент $t'' > t'$; (b) в непустом интервале (t', t'') в одном из расписаний выполняется ненулевая операция $O_{kj'}$, а в другом — другая операция $O_{kj''}$ (либо машина простаивает).

Пусть имеет место случай (a). Так как на M_{k-1} расписания S и S_g совпадают, а по свойству (8*) в расписании S нулевая операция $O_{kj''}$ выполняется в наиболее ранний возможный момент (в момент окончания операции $O_{k-1,j''}$), то $t' = c_S(O_{k-1,j''}) = c_{S_g}(O_{k-1,j''}) = c_S(O_{kj''}) < t'' = c_{S_g}(O_{kj''})$. Поскольку S_g — жадное расписание, в интервале (t', t'') машина M_k не простаивает, причем в этом интервале в расписании S_g на машине M_k могут выполняться лишь операции, более приоритетные чем $O_{kj''}$. Поскольку нулевая операция $O_{kj''}$ не может быть связана с более приоритетной операцией $O_{kj'}$ соотношением (2), по определению отношения $\lhd_S$ должно выполняться соотношение (1), т. е. в интервале (t', t'') на машине M_k в расписании S_g могут выполняться лишь те операции, которые заканчиваются в расписании S не позже момента t'. Однако все ненулевые операции на M_k, заканчивающиеся в S не позже момента t', и в расписании S_g заканчиваются не позже t' (поскольку по предположе-

нию расписания совпадают в интервале $[0, t')$). Нулевые же операции не могут заполнить собой ненулевой отрезок (t', t''). Следовательно, случай (a) невозможен.

Предположим, что имеет место случай (b). Легко понять, что ни в одном из расписаний S_g и S машина M_k не может простаивать в интервале (t', t'') в силу их плотности. Она не может простаивать в S_g, так как в течение всего интервала (t', t'') множество операций, готовых к выполнению на M_k, непусто (в нём содержится по крайней мере та ненулевая операция, которая в этом интервале выполняется в расписании S). По той же причине M_k не может простаивать и в расписании S.

Теперь предположим, что в интервале (t', t'') на машине M_k в расписаниях S_g и S выполняются соответственно фрагменты $O'_{kj'}$ и $O'_{kj''}$ ненулевых операций разных работ. Так как обе операции, очевидно, готовы к выполнению в момент t' в обоих расписаниях, а жадный алгоритм из множества готовых операций выбирает наиболее приоритетную (по принципу 4), то $O_{kj'} \lhd_S O_{kj''}$. Значит, выполняется одно из соотношений (1), (2). Предположим, что выполняется (1). Тогда $c_S(O_{kj'}) \leqslant s_S(O_{kj''}) \leqslant t'$. Отсюда следует, что операция $O_{kj'}$ завершается в расписании S до момента t'. Но так как в интервале $[0, t')$ расписания S и S_g совпадают, то и в расписании S_g операция $O_{kj'}$ завершается до момента t' и её выполнение в непустом интервале (t', t'') невозможно.

Предположим, что выполняется (2). Так как часть операции $O_{kj'}$ (в объёме, не меньшем $|O'_{kj'}| = t'' - t'$) выполняется в расписании S после фрагмента $O'_{kj''}$ (после чего следует ненулевой завершающий фрагмент операции $O_{kj''}$), то получаем ситуацию, описываемую в свойстве (2*): фрагмент $O'_{kj'}$ операции $O_{kj'}$ выполняется между двумя фрагментами операции $O_{kj''}$. Отсюда с учётом свойства (2*), которому удовлетворяет расписание S, следует, что операция $O_{kj'}$ становится готовой к выполнению не раньше момента t''. В то же время известно, что момент её готовности наступает не позже t'. Противоречие доказывает, что случай (b) также невозможен. Совпадение расписаний S и S_g на машинах M_i $(i < m)$ доказано.

Покажем, что расписание S_g (как и расписание S) имеет минимальную длину. Заметим, что оптимум исходной задачи, равный $C_{\max}(S)$, совпадает с оптимумом задачи $X \doteq \langle 1|r_j,\mathrm{pmtn}|\mathrm{C_{max}}\rangle$, на входе которой заданы длительности работ $p_j = |O_{mj}|$ $(j = 1, \ldots, n)$ и моменты поступления работ $r_j = c_S(O_{m-1,j})$; при этом оптимальным решением задачи X является плотное расписание, в котором все работы выполняются без прерываний в порядке неубывания r_j. С другой стороны, в расписании

S_g на машине M_m работы также выполняются в порядке окончания их операций на машине M_{m-1}. Поскольку выше было доказано совпадение расписаний S и S_g на машине M_{m-1}, в расписании S_g работы $\{J_j\}$ выполняются на машине M_m также в порядке неубывания величин r_j. Так как работы в расписании S_g выполняются плотно, это даёт оптимальное решение задачи X, а вместе с ней и исходной задачи. Теорема 2 доказана.

Теорема 2 позволяет строить алгоритм точного решения задачи $\mathcal{D}$.

Следствие 1. *Для нахождения оптимального расписания в задаче $\mathcal{D}$ для заданного примера $I \in \mathcal{I}_{m,n}$ достаточно перебрать все наборы из $m-1$ перестановок $\pi^1, \dots, \pi^{m-1} \in \mathcal{P}_n$, для каждого набора $(\pi^1, \dots, \pi^{m-1})$ с помощью алгоритма $\mathcal{A}_g(\pi^1, \dots, \pi^{m-1}; I)$ построить жадное расписание S_g и из множества полученных расписаний выбрать кратчайшее.*

Теорема 3. *Для любого примера $I \in \mathcal{I}_{m,n}$ задачи $\mathcal{D}$ с m машинами и n работами существует оптимальное расписание, имеющее не более $(m-2)(n-1)$ прерываний.*

Доказательство. Для заданного примера I рассмотрим оптимальное расписание S, полученное жадным алгоритмом $\mathcal{A}_g(\pi^1, \dots, \pi^{m-1}; I)$ при подходящем наборе приоритетных перестановок $(\pi^1, \dots, \pi^{m-1})$ (такой набор существует согласно теореме 2). Согласно принципу 5 жадных алгоритмов прерывание какой-либо операции O_{kj} на машине M_k может возникнуть только в момент w-перехода с M_{k-1} на M_k работы $J_{j'}$ $(j' \neq j)$ для выполнения на M_k операции $O_{kj'}$. Таким образом, всякая операция $O_{kj'}$ может являться причиной не более одного прерывания, и то лишь в том случае, когда $O_{kj'}$ — не первая и не последняя операция своей работы. (Последним операциям некуда торопиться и незачем кого-либо прерывать. Их достаточно выполнять в «фоновом режиме», когда на машине нет более срочных задач.) Кроме того, операция, выполняемая на какой-то машине первой, также не может прервать никакую другую операцию. Таким образом, “причинами прерываний” в жадном расписании S для примера I задачи $\mathcal{D}$ могут быть лишь непервые операции средних машин. Отсюда вытекает требуемая оценка. Теоерема 3 доказана.

4. Анализ свойств критических путей в жадном расписании

Пусть S — расписание, построенное жадным алгоритмом $\mathcal{A}_g(\pi^1, \dots, \pi^{m-1}; I)$ для заданного примера I задачи $\mathcal{D}$; $G_p(S)$ — граф предшествования фрагментов, построенный по расписанию S; $C = (O^1 \to O^2 \to \dots \to O^\alpha)$ — критический путь в графе $G_p(S)$. По лемме 3 и замечанию 1 расписание S является ранним, а по лемме 1 длина расписания S совпадает с длиной пути C. В дальнейшем критический путь в графе $G_p(S)$

будем называть *критическим путём в расписании* S.

Определение 15. Пусть S — допустимое расписание в задаче Джонсона, C — путь в графе $G_p(S)$. *Составом* пути C будем называть множество $\langle C\rangle$ операций, фрагменты которых входят в C. Путь C будем называть *полным*, если в нём содержатся все фрагменты всех операций $O_i \in \langle C\rangle$.

Заметим, что если путь C полный, то его длина $|C|$ однозначно определяется по его составу. Таким образом, если в $G_p(S)$ существует полный критический путь, то по лемме 1 длина расписания S однозначно определяется по составу такого пути.

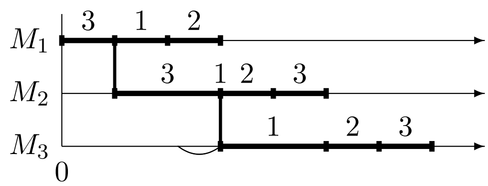

Рис. 2. Плотное расписание, не имеющее полного критического пути

Как показывает пример на рис. 2, где цифрами обозначены номера работ, не в любом плотном расписании существует полный критический путь. Этот пример удовлетворяет всем свойствам $(*)$ за исключением (2^*): операция O_{21} выполняется между двумя фрагментами операции O_{23}, но становится готовой к выполнению раньше момента завершения первого фрагмента операции O_{23}. Вместе с тем справедлива

Лемма 4. *В любом допустимом расписании задачи $\mathcal{D}$, обладающем свойствами $(*)$, существует полный критический путь.*

Доказательство. Пусть S — допустимое расписание примера I, обладающее свойствами $(*)$. Определим неубывающую последовательность моментов времени $0 = t_0 \leqslant t_1 \leqslant \ldots \leqslant t_m$ обратной индукцией по номеру момента. Полагаем $t_m = C_{\max}(S)$. Пусть моменты $t_k, \ldots, t_m$ определены. В качестве t_{k-1} возьмём ближайший к t_k момент $t \leqslant t_k$, который является либо моментом окончания интервала простоя на машине M_k (см. рис. 3 *а*; выпуклой дугой обозначается интервал простоя), либо моментом прерывания какой-то операции на машине M_k, завершающейся после момента t_k (рис. 3 *б*). Ясно, что на машинах M_1 и M_m вторая ситуация невозможна, так как S удовлетворяет свойству (4^*). Отсюда в силу плотности расписания на M_1 следует, что $t_0 = 0$. Таким образом, совокупность интервалов $\{[t_{k-1}, t_k] \,|\, k = 1, \ldots, m\}$ покрывает всю длину расписания S, и для каждого $k = 1, \ldots, m$ в интервале $[t_{k-1}, t_k]$ машина M_k работает без простоев.

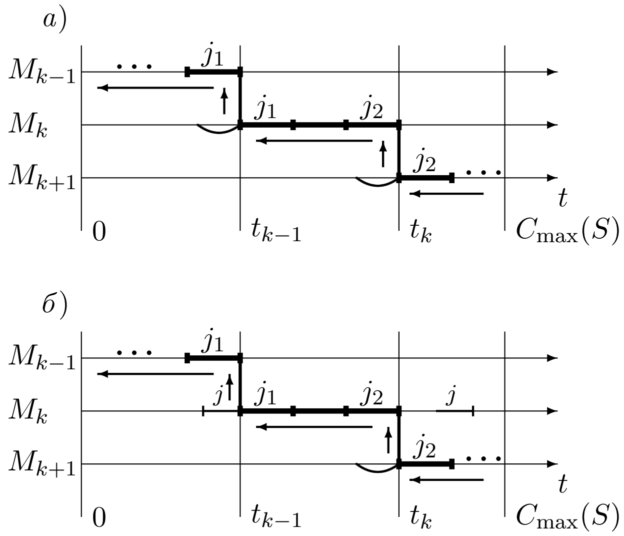

Рис. 3. Возможные варианты звена полного критического пути на машине M_k

Из свойств (1*) и (3*) следует, что каждый момент t_k ($k = 1, \ldots, m - 1$) является моментом w-перехода хотя бы одной работы с M_k на M_{k+1}. Для машин M_k, M_{k+1} выберем одну такую переходящую работу (номер которой будем обозначать через j_k) индукцией по убыванию k. В качестве j_{m-1} берём номер любой работы из множества работ, переходящих с M_{m-1} на M_m. Пусть для момента t_k ($1 < k < m$) номер j_k определён. Если $t_{k-1} < t_k$, то в качестве j_{k-1} берём номер произвольной работы из множества работ, переходящих с M_{k-1} на M_k в момент t_{k-1}. Если же $t_{k-1} = t_k$, то полагаем $j_{k-1} = j_k$.

Докажем, что в последнем случае работа J_{j_k} является переходной в момент t_k с машины M_{k-1} на M_k. Сначала покажем, что операция O_{kj_k} нулевая. Момент t_{k-1} по своему выбору является либо моментом окончания интервала простоя на машине M_k, либо моментом прерывания некоторой операции $O_{kj'}$ (при этом $j' \neq j_k$, так как O_{kj_k} в момент t_{k-1} завершается, а не прерывается). В обоих случаях из предположения $|O_{kj_k}| \neq 0$ следует существование в расписании S нулевого максимального фрагмента операции O_{kj_k}, выполняемого в момент t_k, что противоречит замечанию 2.

Итак, O_{kj_k} — нулевая операция, выполняемая в момент t_k. Если t_k — момент окончания простоя на M_k, то равенство $r_S(O_{kj_k}) = t_k$ вытекает из плотности расписания S. Если же t_k — момент прерывания какой-то операции $O_{kj'}$ ($j' \neq j_k$), то по свойству (2*) получаем $r_S(O_{kj_k}) = t_k$.

Таким образом, момент t_k является моментом w-перехода работы J_{j_k} с машины M_{k-1} на машину M_k.

В целях унификации наших выкладок к исходному семейству работ добавим две фиктивные работы (с номерами 0 и $n+1$), имеющие нулевые операции на всех машинах. Будем считать, что в расписании S операции работы 0 на всех машинах выполняются первыми, а операции работы $(n+1)$ — последними. Положим $j_0 = 0$; $j_m = n+1$.

Пусть O_{ij}^f и O_{ij}^l обозначают первый и последний максимальные фрагменты операции O_{ij} в расписании S (возможно, они совпадают). По заданной последовательности моментов $t_0 \leqslant t_1 \leqslant \ldots \leqslant t_m$ определим последовательность фрагментов $O^1, \ldots, O^\alpha$ в расписании S следующим образом: $C(S) \doteq C_1 \oplus C_2 \oplus \ldots \oplus C_m$, где C_k — последовательность вершин максимального по включению пути в графе $G_p(S)$, ведущего из вершины $O_{kj_{k-1}}^f$ в вершину $O_{kj_k}^l$. Нетрудно убедиться, что каждый из путей C_k $(k = 1, \ldots, m)$ существует, поскольку фрагменты $O_{kj_{k-1}}^f$ и $O_{kj_k}^l$ либо совпадают (в случае $t_{k-1} = t_k$), либо (при $t_{k-1} < t_k$) между ними имеет место отношение $O_{kj_{k-1}}^f \prec_S O_{kj_k}^l$. Кроме того, конечная вершина $O_{kj_k}^l$ пути C_k и начальная вершина O_{k+1,j_k}^f пути C_{k+1} принадлежат одной работе, поэтому дуга $(O_{kj_k}^l, O_{k+1,j_k}^f) \in U(S)$. Таким образом, $C(S)$ является путём в графе $G_p(S)$.

Так как в интервале $[t_{k-1}, t_k]$ машина M_k работает без простоев (по выбору моментов t_k), а C_k — максимальный по включению путь из фрагментов операций машины M_k, выполняемых между $O_{kj_{k-1}}^f$ и $O_{kj_k}^l$, то суммарная длина пути C_k равна $t_k - t_{k-1}$. Отсюда следует, что длина всего пути $C(S)$ равна $C_{\max}(S)$, т. е. путь $C(S)$ является критическим.

Докажем полноту этого пути, т. е., что никакая операция не представлена в пути $C(S)$ лишь частью своих фрагментов. Достаточно доказать это для операций средних машин, поскольку на M_1 и M_m операции выполняются без прерываний (по свойству (4*)). Пусть $C(S)$ содержит собственный фрагмент $O' \in C_k$ операции O_{kj}, выполняемый в интервале $[t_{k-1}, t_k]$. Тогда по замечанию 2 O' является ненулевым фрагментом. По правилу выбора момента t_{k-1} никакой фрагмент O'' операции O_{kj} не может заканчиваться строго позже момента t_k. Предположим, что ненулевой фрагмент O'' операции O_{kj} завершается до момента t_{k-1}. В силу плотности расписания S, на машине M_k не может быть простоя между фрагментами O'' и O', и в частности, непосредственно перед моментом t_{k-1}. Отсюда следует, что момент t_{k-1} был определён как момент прерывания некоторой операции $O_{kj'}$, которая возобновляет своё выполнение

после момента t_k (причём, $j' \neq j$, поскольку, как мы выяснили, операция O_{kj} не может иметь фрагментов, выполняемых после t_k). Но в этом случае фрагмент O' операции O_{kj} выполняется между двумя фрагментами операции $O_{kj'}$, и по свойству (2*) операция O_{kj} имеет момент готовности $r_S(O_{kj}) \geqslant t_{k-1}$, что противоречит предположению о выполнении её ненулевого фрагмента O'' до момента t_{k-1}. Лемма 4 доказана.

Из лемм 3, 4 и теоремы 2 вытекают

Следствие 2. *Для любого примера I задачи $\mathcal{D}$ и любого набора перестановок $(\pi^1, \ldots, \pi^{m-1})$ расписание S, получаемое жадным алгоритмом $\mathcal{A}_g(\pi^1, \ldots, \pi^{m-1}; I)$, содержит полный критический путь.*

Теорема 4. *Длина кратчайшего расписания любого примера задачи $\mathcal{D}$ совпадает с суммарной длиной операций из некоторого подмножества.*

ЛИТЕРАТУРА

1. **Танаев В. С., Сотсков Ю. Н., Струсевич В. А.** Теория расписаний. Многостадийные системы. М.: Наука, 1989.

2. **Baptiste Ph., Carlier J., Kononov A., Queyranne M., Sevastyanov S., Sviridenko M.** Structural properties of preemptive schedules // J. of Scheduling (submitted).

3. **Chen B., Potts C. N., Woeginger G. J.** A review of machine scheduling: complexity, algorithms and approximability // Handbook of combinatorial optimization. V. 3. Boston: Kluwer Acad. Publ., 1998. P. 21–169.

4. **Gonzalez T., Sahni S.** Flowshop and jobshop schedules: complexity and approximation // Oper. Res. 1978. V. 26, N 1. P. 36–52.

Адрес авторов:

Институт математики
им. С. Л. Соболева СО РАН,
пр. Академика Коптюга, 4,
630090 Новосибирск,
Россия.

Статья поступила
24 марта 2006 г.