СИБИРСКИЙ ЖУРНАЛ ИССЛЕДОВАНИЯ ОПЕРАЦИЙ
Январь – март 1994. Том 1, № 1, 20–42

УДК 519.854.2

ЭФФЕКТИВНОЕ ПОСТРОЕНИЕ РАСПИСАНИЙ В СИСТЕМАХ ОТКРЫТОГО ТИПА *)

С. В. Севастьянов

В классе задач «*open shop*» выделен новый подкласс полиномиально разрешимых задач. Показано, что для задачи с m машинами и n работами оптимальное расписание строится полиномиальным от m и n алгоритмом, если нагрузка $M_{i'}$ одной из машин (обозначим ее через i') превосходит нагрузку на остальных машинах на заданную величину «доминирования» $M_{i'} - M_i \geqslant (2m-4)K$, $i \neq i'$, где K— максимальная длительность одной операции. Ослабленное условие доминирования $M_{i'} - M_i \geqslant (m-1)K$, $i \neq i'$, в сочетании с условием $M = \max M_i \geqslant (5.45m - 7)K$ также позволяет эффективно строить оптимальное расписание системы. В обоих случаях длина оптимального расписания равна M. Описывается также приближенный алгоритм трудоемкости $O(nm^2)$, который для любой функции $\eta(m)$ и любого входа задачи open shop, удовлетворяющего условию $M \geqslant \eta(m)K$, позволяет находить расписание с абсолютной оценкой точности, зависящей от функции $\eta(m)$ и не зависящей от числа работ. В частности, при $M \geqslant (7m-6)K$ длина расписания не превосходит величины $M + (m-1)K/3$, что улучшает известную оценку длины любого жадного расписания.

§ 1. Постановка задачи и обзор результатов

Обслуживающую систему с заданным множеством машин и заданным множеством работ мы называем *системой открытого типа*, если каждая работа состоит из нескольких не пересекающихся во времени операций, на порядок выполнения которых не наложено никаких ограничений. Таким образом, после выполнения очередной операции какой-то работы разрешается приступить к любой из еще не выполненных операций этой работы. Иначе говоря, все машины, на которых должны выполняться оставшиеся операции данной работы, одинаково доступны (открыты) для этой работы. Мы не будем приводить здесь формального определения системы открытого типа в общем случае, поскольку мы рассматриваем конкретный случай таких систем, известный в теории расписаний под названием системы «open shop».

---

*) Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (код проекта 93-012-489) и частичной поддержке Международного научного фонда (ISF).

Дадим определение системы «open shop» (далее — *системы* OS). Имеется множество машин $\mathcal{M} = \{1, \dots, m\}$ и множество работ $\mathcal{N} = \{1, \dots, n\}$. Каждая работа $j \in \mathcal{N}$ состоит из m операций $\{o_1^j, \dots, o_m^j\} \doteq O^j$. Известно, что операция o_i^j выполняется на машине i за время t_{ji} ($j \in \mathcal{N}$, $i \in \mathcal{M}$). Операции могут быть выполнены в любом порядке, но при соблюдении следующих условий:

($*$) в каждый момент времени выполняется не более одной операции каждой работы;

($**$) в каждый момент времени выполняется не более одной операции на каждой машине;

($***$) каждая операция неразрывна.

Расписанием выполнения работ в обслуживающей системе называется матрица $S = (s_{ji})_{j \in \mathcal{N}, i \in \mathcal{M}}$ неотрицательных чисел, где s_{ji} задает момент начала операции o_i^j. Расписание, удовлетворяющее всем требованиям системы, называется *допустимым*.

Длиной расписания S называется функция

$$T(S) = \max_{j,i}(s_{ji} + t_{ji}),$$

задающая длину временного интервала (начинающегося в нулевой момент времени), в течение которого выполняются все операции системы.

Допустимое расписание минимальной длины называют также *оптимальным по быстродействию расписанием системы.*

В теории расписаний задачей «open shop» (далее — *задачей* OS) называется следующая

Задача. *Найти оптимальное по быстродействию расписание* S_{opt} *в системе* OS.

В [1] был построен эффективный алгоритм решения рассматриваемой задачи в случае $m = 2$, а также доказана ее NP-трудность при $m \geqslant 3$. В обзоре [2] указано, что при переменном числе машин задача является NP-трудной в сильном смысле. В [3–5] для класса задач OS были выделены подклассы эффективно разрешимых задач OS следующего типа.

Пусть

$F_i = \{o_i^j,\ j = 1, \dots, n\}$ — множество операций машины i,

$M_i = \sum_{j \in \mathcal{N}} t_{ji}$ — суммарная нагрузка машины i,

$M = \max\limits_i M_i$,

$K = \max\limits_{ij} t_{ji}$ — максимальная длительность одной операции.

В силу условия (**) очевидна следующая оценка снизу длины любого (в том числе оптимального) расписания:

$$T(S) \geqslant M. \tag{1}$$

Т. Фиала в [3] впервые показал, что если M достаточно велико, т. е.

$$M \geqslant \eta(m)K \tag{2}$$

для определенной функции η от числа машин, то

$$T(S_{\text{opt}}) = M \tag{3}$$

и оптимальное расписание находится эффективно. Поскольку M как сумма длительностей n операций имеет тенденцию к росту с ростом числа работ, а правая часть неравенства (2) не зависит от n, класс эффективно разрешимых задач, описываемый неравенством (2), довольно широк. Причем ясно, что он тем шире, чем меньше функция $\eta(m)$.

Таким образом, возникает следующая

Задача. *Найти наименьшую функцию $\eta(m) = \eta^*(m)$, для которой из неравенства* (2) *вытекает равенство* (3).

В [3–5] в качестве верхних оценок функции $\eta^*(m)$ рассматривались некоторые функции η_1, η_2, η_3, имеющие порядок $\Theta(m \log m)$, а в [4] — функция η_4 порядка $\Theta(m^2)$. (Мы говорим, что *функция $\eta(m)$ имеет порядок* $\Theta(\varphi(m))$, если существуют положительные константы λ_1 и λ_2 такие, что $\lambda_1\varphi(m) \leqslant \eta(m) \leqslant \lambda_2\varphi(m)$ при любом m.) При этом использование функции η_4 дает более точные верхние оценки функции η^* для малых значений m.

В настоящей работе выделяются новые подклассы эффективно разрешимых задач OS, основанные на понятии доминирования.

Пусть машина i^d такова, что $M_{i^d} = \max\limits_i M_i$; $M_{i^d} > M_i$ для всех i $(i \neq i^d)$. Машину i^d будем называть *доминантой*, остальные машины — *минорантами*, а величину $\Delta = M_{i^d} - \max\limits_{i \neq i^d} M_i$ — *величиной доминирования.*

В статье доказываются три теоремы. Теорема 1 утверждает, что если $\Delta \geqslant (2m-4)K$, то $T(S_{\text{opt}}) = M$; при этом оптимальное расписание находится эффективно. В теореме 2 выделяется подкласс эффективно разрешимых задач OS при более слабом условии доминирования $\Delta \geqslant (m-1)K$ в сочетании с определенным условием вида (2), где функция $\eta(m)$ линейная.

В [4] был приведен следующий простой, но весьма полезный результат, принадлежащий А. Рачмань: *любой «жадный»* (дословный перевод: *greedy*) *алгоритм позволяет строить такое расписание S для задачи* OS, *что*

$$T(S) \leqslant M + (m-1)K. \tag{4}$$

Отсюда ввиду (1) вытекает оценка абсолютной погрешности алгоритма

$$T(S) - T(S_{\text{opt}}) \leqslant (m-1)K,$$

не зависящая от числа работ.

В доказательстве теоремы 3 мы описываем приближенный полиномиальный алгоритм (также из класса жадных) с лучшей оценкой длины расписания, получаемой при условии выполнения определенного неравенства вида (2) с линейной функцией $\eta(m)$.

Сформулируем два результата из работ [5, 6], которые мы будем использовать в доказательствах теорем 2 и 3.

Следствие 1 теоремы [6, с. 67]. *Пусть* $\{x_1, \dots, x_n\} \subset \mathbb{R}^m$, $x = \frac{1}{n}\sum_{i=1}^n x_i$, $B = \operatorname{conv}\{x_1, \dots, x_n\}$. *Тогда для любой точки* $a \in \mathbb{R}^m$ *с трудоемкостью* $O(n^2m^2)$ *находится перестановка* $\pi = (\pi_1, \dots, \pi_n)$ *такая, что*

$$\sum_{i=1}^{k} x_{\pi_i} - (k-m+1)x \in (m-1)B + B_a, \quad k = 1, \dots, n,$$

где $B_a = \operatorname{conv}\{0, a - \frac{1}{m}B\}$.

Этот результат мы используем в следующей частной форме.

Лемма 1. *Пусть* $x_j = (x_j(1), \dots, x_j(m)) \in [0, K]^m$, $j = 1, \dots, n$; $x = \frac{1}{n}\sum_{j=1}^n x_j$. *Тогда с трудоемкостью* $O(n^2m^2)$ *находится перестановка* $\pi = (\pi_1, \dots, \pi_n)$ *такая, что для любых* $k = 1, \dots, n$; $i = 1, \dots, m$ *выполнено включение*

$$\sum_{j=1}^{k} x_{\pi_j}(i) - (k-m+1)x(i) \in [-K/m, (m-1)K].$$

Лемму 5 [5, с. 146] переформулируем следующим образом.

Лемма 2. *Пусть заданы натуральные числа* m, n, k, q $(q < k)$, *множество* Ψ, $|\Psi| = kn$, *семейство чисел* $\{x_\psi \in \mathbb{R}^+ \mid \psi \in \Psi\}$ *и два разбиения множества* Ψ *на подмножества:* $\Psi = \bigcup_{i=1}^m \Psi_i$, $\Psi = \bigcup_{j=1}^n \Psi^j$, *причем* $|\Psi^j| = k$, $j = 1, \dots, n$. *Тогда с трудоемкостью* $O(mnk)$ *может быть найдено подмножество* $\Psi' \subset \Psi$, *удовлетворяющее соотношениям*

$$|\Psi' \cap \Psi^j| = q, \qquad j = 1, \dots, n,$$

$$\left| \sum_{\psi \in \Psi' \cap \Psi_i} x_\psi - \frac{q}{k} X'_i \right| < K_i, \qquad i = 1, \dots, m,$$

где $K_i = \max\limits_{\psi\in\Psi_i} x_\psi$, $X_i = \sum\limits_{\psi\in\Psi_i} x_\psi$, $i = 1,\ldots,m$.

Структура статьи следующая. В § 2 приведено описание произвольного жадного алгоритма. Доказано, что для любых заданных приоритетных порядков работ на машинах возможна реализация жадного алгоритма за $O(nm\min\{n,m\})$ операций. В § 3 исследованы свойства, общие для всех жадных расписаний (т. е. получаемых жадными алгоритмами). Теоремы о доминировании и соответствующие алгоритмы приведены в § 4, а обоснование оценок приближенного алгоритма — в § 5. В конце статьи дан полный список используемых обозначений.

§ 2. Жадный алгоритм построения расписания задачи OS

В следующих параграфах будет рассматриваться применение жадных алгоритмов к решению задачи OS. Почти все известные автору алгоритмы решения задачи OS являются жадными и различаются лишь правилами построения приоритетных порядков. Таковыми же будут алгоритмы $\mathcal{A}_2$, $\mathcal{A}_3$, а также с небольшими отличиями и алгоритм $\mathcal{A}_1$, которые мы будем строить в доказательствах теорем 1–3.

Уточним понятие жадного алгоритма для задачи OS.

Пусть задан некоторый порядок машин $(1,\ldots,m)$, который будем называть «приоритетным». (Для удобства считаем, что машины занумерованы в соответствии с этим порядком.) Кроме того, для каждой машины $i = 1,\ldots,m$ задан «приоритетный» порядок работ $p^i = (p^i_1,\ldots,p^i_n)$.

Жадным алгоритмом решения задачи OS будем называть алгоритм, строящий расписание на следующих принципах.

1. Машина не простаивает, если есть работа, готовая к выполнению на ней.
2. При выборе очередной работы для освободившейся машины i из нескольких работ, готовых к выполнению на этой машине, предпочтение отдается наиболее приоритетной работе (т. е. стоя́щей раньше в приоритетном порядке p^i).
3. При выборе очередной операции работы предпочтение отдается ее операции (из множества еще не начатых) на наиболее приоритетной из свободных машин.

Согласно приведенному определению всякий жадный алгоритм состоит из двух этапов. На первом этапе определяются приоритетный порядок (нумерация) $(1,\ldots,m)$ машин и приоритетные порядки $\{p^i \mid i \in \mathcal{M}\}$ работ на машинах, а на втором этапе для заданных (найденных) приоритетных порядков в соответствии с принципами 1–3 конструируется жадное расписание. Первый этап специфичен для каждого алгоритма и не поддается единому описанию. Идеология второго этапа для

всех жадных алгоритмов идентична, однако возможны различные варианты его реализации, имеющие различные характеристики трудоемкости и требуемого объема памяти. При этом возможен учет специфики приоритетных порядков, полученных на первом этапе. Нетрудно предложить реализацию второго этапа алгоритма с трудоемкостью $O(nm^3)$, где для каждой из nm операций время ее старта в расписании будет находиться в худшем случае за $O(m^2)$ вычислительных операций. Наши дальнейшие усилия направлены на то, чтобы показать, что для любых исходных приоритетных порядков нахождение времени старта каждой операции можно реализовать с трудоемкостью $O(\min\{n, m\})$, что даст оценку трудоемкости второго этапа алгоритма $O(nm \min\{n, m\})$. (Приведенная в монографии [7, с. 22] оценка трудоемкости жадного алгоритма $O(nm)$ представляется ошибочной.) Объем требуемой для этого дополнительной памяти не превосходит по порядку объема входной информации.

Заметим, что хотя в системе OS понятия «работа» и «машина» взаимодвойственны, приведенное определение жадного алгоритма не симметрично относительно этих понятий, поскольку в нем приоритетные порядки работ могут быть различны на разных машинах, в то время как приоритетные порядки машин берутся одинаковыми для всех работ. Это связано с тем, что произвольная комбинация приоритетов работ на разных машинах и машин для разных работ может оказаться противоречивой.

ОПИСАНИЕ ВТОРОГО ЭТАПА ЖАДНОГО АЛГОРИТМА

На втором этапе жадного алгоритма строится жадное расписание для заданных приоритетных порядков работ на машинах.

Вначале опишем версию алгоритма в предположении $m \leqslant n$. Предполагаем, что машины занумерованы в соответствии с заданным на них приоритетным порядком. Работу, выполняемую на машине i в момент времени t, обозначим $J_i(t)$. Запись $J_i(t) = 0$ означает, что машина i простаивает. Множество $J(t) = \{J_i(t) \mid i \in \mathcal{M}\} \setminus \{0\}$ называем *фронтом работ в момент времени* t. На каждом шаге алгоритма будет известна *граница определенности* текущего (частичного) расписания S, т. е. такая величина $\hat{t} \geqslant 0$, что в интервале $[0, \hat{t})$ никаких дальнейших изменений в расписании S происходить не будет. Это означает, что никаким операциям не будут приписываться времена их начала s_{ji} в интервале $[0, \hat{t})$. (К этому добавим, что алгоритм не переопределяет однажды назначенные времена s_{ji}.) В процессе работы алгоритма граница $\hat{t}$ скачкообразно перемещается от нуля в положительном направлении, пока не достигнет момента $T(S)$ окончания последней операции.

Множество $\hat{J} = \{J_1, \dots, J_m\} = J(\hat{t})$ называем также *текущим фронтом работ*.

Алгоритм состоит из предварительного шага и nm последующих шагов, где в конце шага $k \geqslant 1$ будут известны k операций, заканчивающихся в расписании S до текущей границы $\hat{t}$. Кроме величины $\hat{t}$ и фронта $\hat{J}$ в конце каждого шага для машины $i \in \mathcal{M}$ будет известен список $U_i(\hat{t}) = (u_1^i, u_2^i, \ldots, u_{a_i}^i)$ работ (упорядоченных в соответствии с их приоритетным порядком), моменты начала операций которых на машине i еще не определены; a_i — текущая длина списка $U_i(\hat{t})$.

Для уменьшения трудоемкости алгоритма введем два дополнительных массива. Целый массив $\Phi[1..n]$ при $\Phi[j] = 0$ сообщает, что работа j не является работой текущего фронта $\hat{J}$. В противном случае $\Phi[j]$ сообщает номер машины, на которой выполняется работа $j \in \hat{J}$. Элементы битового массива $B[1..m, 1..m]$ принимают значения 0 или 1. Если $J_i = 0$ (т. е. машина i простаивает), то $B[i, i'] = 1$, если и только если работа фронта $J_{i'}$ входит в список U_i. (Таким образом, в случае $J_i = 0$ i-я строка матрицы B содержит ровно a_i единиц.) Если $J_i \neq 0$, то значения $B[i, i']$ могут быть произвольны.

Предварительный шаг. Полагаем $a_i := n$, $U_i = (u_1^i, \ldots, u_n^i) := (p_1^i, \ldots, p_n^i) = p^i$, $i = 1, \ldots, m$, $\Phi \equiv 0$, $B \equiv 1$; $\hat{t} := 0$. Для $i = 1, \ldots, m$ в списке U_i находим работу $j = u_k^i$ с наименьшим индексом k такую, что $\Phi[j] = 0$, и включаем ее во фронт, исключая из списка U_i: $J_i := j$, $s_{ji} := \hat{t}$, $U_i := U_i \setminus \{j\}$; $a_i := a_i - 1$, $\Phi[j] := i$, $B[i, i] := 0$. (Ясно, что порядок просмотра машин на этом шаге существен.)

Отметим свойство $\hat{t} \leqslant s_{J_i i} + t_{J_i i}$, которое на каждом шаге алгоритма будет выполняться для любой работы текущего фронта $J_i \in \hat{J}$.

Трудоемкость предварительного шага равна $O(n + m^2)$.

Шаг k $(k = 1, \ldots, nm)$ состоит из трех логических действий.

Действие 1. Среди непростаивающих машин $\{i \mid J_i \neq 0\}$ находим машину с наименьшим номером $i = i'$, для которого достигается $\min\limits_i(s_{J_i i} + t_{J_i i}) \doteq t'$. (На этой машине раньше других заканчивается работа фронта $j' = J_{i'}$.) Ясно, что $\hat{t} \leqslant t'$. Положим $\hat{t} := t'$. Поскольку освободились одновременно работа j' и машина i', нужно найти очередную операцию как для работы j' (действие 2), так и для машины i' (действие 3).

Действие 2. Полагаем $\Phi[j'] := 0$. Просматриваем столбец i' матрицы B и отыскиваем все машины i такие, что

$$J_i = 0, \quad a_i > 0, \quad B[i, i'] = 1. \tag{5}$$

Если таковых нет, переходим к действию 3. Пусть такие машины нашлись и первая из них — машина i''. Это означает, что только что закончившаяся работа j' входит в список $U_{i''}$ простаивающей машины i''. Назначаем работу j' на машину i'': $J_{i''} := j'$, $\Phi[j'] := i''$, $s_{j'i''} := \hat{t}$,

$U_{i''} := U_{i''} \setminus \{j'\}$ (так как машина i'' простаивала, имеем $a_{i''} \leqslant m-1$; следовательно, работу j' в списке $U_{i''}$ мы найдем за $O(\min\{n,m\})$ операций); $a_{i''} := a_{i''} - 1$. Для остальных машин i со свойством (5) полагаем $B[i,i'] := 0$; $B[i,i''] := 1$ (в i-й строке остается ровно a_i единиц).

Действие 3. Полагаем $J_{i'} := 0$, $B[i',i] := 0$, $i = 1,\ldots,m$. Если $a_{i'} > 0$, то последовательно просматриваем список $U_{i'}$: $j := u_1^{i'}, u_2^{i'}, \ldots$. Если $\Phi[j] \neq 0$, то $B[i',\Phi[j]] := 1$. Если $\Phi[j] = 0$, то прекращаем просмотр списка (нашлась работа j, не входящая во фронт $\hat{J}$) и полагаем

$$\Phi[j] := i', \quad J_{i'} := j, \quad s_{ji'} := \hat{t}, \quad U_{i'} := U_{i'} \setminus \{j\}, \quad a_{i'} := a_{i'} - 1.$$

Наконец, если весь список $U_{i'}$ просмотрен и работа j с $\Phi[j] = 0$ не найдена, то машина i' становится простаивающей, а i'-я строка в матрице B в точности перечисляет номера машин, из-за которых машина i' простаивает. В последнем случае имеем $a_{i'} \leqslant m-1$. Поэтому для просмотра списка $U_{i'}$ требуется $O(\min\{n,m\})$ элементарных вычислительных операций. Заметим, что в случае включения работы j во фронт не требуется производить какие-то изменения в массиве B. Это включение могло бы добавить единицу в i-ю строку массива B, если $j \in U_i$. Но для простаивающей машины i этого произойти не может (так как количество единиц в такой строке уже максимально — равно a_i), а для непростаивающей машины i значения $B[i,\cdot]$ несущественны.

Работа жадного алгоритма на втором этапе в случае $m \leqslant n$ описана.

Нетрудно убедиться, что на любом шаге k, $k = 1,\ldots,nm$, для выполнения каждого из трех действий требуется $O(m)$ вычислений, что с учетом $m \leqslant n$ дает общую оценку трудоемкости второго этапа, равную $O(nm\min\{n,m\})$.

Заметим, что если бы мы не ввели массивов Φ и B, а ограничились массивом J и списками $\{U_i \mid i \in \mathcal{M}\}$, то для назначения новой работы на машину i' (для чего отыскивается первая работа в списке $U_{i'}$, отличная от всех работ фронта) пришлось бы затратить $O(m^2)$ операций, сравнивая каждую очередную работу списка $U_{i'}$ со всеми работами фронта. Точно так же для отыскания простаивающей машины, имеющей в своем списке работу j', потребовалось бы выполнение $O(m^2)$ операций.

В случае $m > n$ битовый массив B размером $m \times m$, вообще говоря, уже не дает экономии памяти. Вместо него можно использовать списки $\{U^j \mid j \in \mathcal{N}\}$, где для каждой работы $j \in \mathcal{N}$ задан список U^j еще не пройденных ею машин, упорядоченный по убыванию их приоритетов. При этом упрощается логическая схема алгоритма, поскольку информация становится симметричной относительно работ и машин (массиву $J[1..m]$ соответствует массив $\Phi[1..n]$, а спискам $\{U_i \mid i \in \mathcal{M}\}$ — списки $\{U^j \mid j \in \mathcal{N}\}$) и отпадает необходимость в выполнении операций

над массивом B. В то же время мы лишаемся достигнутой за счет использования массива B (в случае $m \leqslant n$) экономии памяти.

Нахождение операции фронта, которая заканчивается раньше других (действие 1), в этом случае будем осуществлять путем просмотра не машин, а работ, затрачивая $O(n) = O(\min\{m, n\})$ операций. Действие 2 выполняем, просматривая список $U^{j'}$, а действие 3 — список $U_{i'}$. В обоих случаях затрачиваем $O(\min\{m, n\})$ операций. Таким образом, в случае $m > n$ оценка $O(mn\min\{m, n\})$ трудоемкости второго этапа остается справедливой.

Объем дополнительной памяти, используемой для работы алгоритма (списки $\{U_i\}, \{U^j\}$, массивы J, Φ, B), в обоих случаях по порядку не превосходит объема входной и выходной информации (куда входят массив чисел $\{t_{ji}\}$, расписание $\{s_{ji}\}$ и, возможно, массив приоритетов p).

Таким образом, нами доказана

Лемма 3. *Для любых приоритетных порядков, полученных на этапе 1 жадного алгоритма решения задачи OS, возможна реализация второго этапа алгоритма с трудоемкостью $O(nm\min\{n, m\})$ при объеме памяти, по порядку равном объему входной информации.*

§ 3. Свойства жадных расписаний

Пусть задано жадное расписание S, определенное вектором приоритетов $p = (p^1, \ldots, p^m)$. Для двух операций o_i^j, $o_i^{j'}$ машины i запись $o_i^j \succ_p o_i^{j'}$ будет означать бо́льший приоритет операции o_i^j. Интервал выполнения операции o' в расписании S обозначим $I(o')$. Для интервалов $I_1 = [a_1, b_1)$ и $I_2 = [a_2, b_2)$ отношение $I_1 \to I_2$ означает $b_1 \leqslant a_2$. Если для двух операций o_i^j и $o_i^{j'}$ машины i выполняется $(o_i^j \succ_p o_i^{j'})\&(I(o_i^{j'}) \to I(o_i^j))$, т. е. операция $o_i^{j'}$ выполняется раньше, чем более приоритетная операция o_i^j, то интервал $I(o_i^{j'})$ назовем *o_i^j-задержкой*. Операцию $o_{i'}^j$ работы j, предшествующую операции o_i^j, назовем *o_i^j-предоперацией*.

Если интервал $I(o_{i'}^j)$ некоторой o_i^j-предоперации начинается раньше некоторой o_i^j-задержки $I(o_i^{j'})$, то последнюю называем *покрываемой* в случае $I(o_i^{j'}) \subseteq I(o_{i'}^j)$ и *полупокрываемой* в случае $(I(o_i^{j'}) \not\subseteq I(o_{i'}^j))\&(I(o_i^{j'}) \cap I(o_{i'}^j) \neq \varnothing)$. (В последнем случае лишь начальный отрезок o_i^j-задержки содержится в $I(o_{i'}^j)$.)

Интервал простоя машины i, предшествующий интервалу $I(o_i^j)$, назовем *o_i^j-простоем.*

Сформулируем без доказательств несколько простых предложений о свойствах жадного расписания.

Предложение 1. *Каждый o_i^j-простой покрывается интервалами выполнения o_i^j-предопераций.*

Предложение 2. *Всякая o_i^j-задержка покрывается или полупокрывается единственной o_i^j-предоперацией.*

Предложение 3. *Для всякой o_i^j-предоперации существует не более одной полупокрываемой ею o_i^j-задержки.*

Для упрощения выкладок мы будем часто опускать множитель K, считая выполненным нормирующее условие $K = 1$ (что не влияет на общность получаемых результатов). Через $k'_{ji}(t)$ и $k''_{ji}(t)$ обозначим количество o_i^j-предопераций, начинающихся строго до момента t и заканчивающихся до момента t соответственно.

Из предложений 1–3 вытекает

Предложение 4. *Для любых операции o_i^j и момента $t \leqslant s_{ji}$ суммарная длина пересечений o_i^j-задержек и o_i^j-простоев с интервалом $[0,t)$ не превосходит суммарной длительности o_i^j-предопераций, выполняемых в интервале $[0,t)$, плюс $k''_{ji}(t)K$.*

Действительно, суммарной длительности o_i^j-предопераций соответствует суммарная длина покрываемых ими o_i^j-простоев и o_i^j-задержек, а также, возможно, одной o_i^j-задержки, полупокрываемой o_i^j-предоперацией $o_{i'}^j$ с $I(o_{i'}^j) \ni t$. Величина $k''_{ji}(t)K$ оценивает длину остальных полупокрываемых o_i^j-задержек.

В более грубой форме имеем

Предложение 5. *Для любых операции o_i^j и момента $t \leqslant s_{ji}$ суммарная длина пересечений o_i^j-задержек и o_i^j-простоев с интервалом $[0,t)$ не превосходит величины $(k'_{ji}(t) + k''_{ji}(t))K$.*

Пусть

$D_i(t)$ — длительность простоя машины i в интервале $[0,t]$;

$w_i(t)$ — время чистой работы машины i после момента t;

C_i — момент завершения машиной i последней операции;

$D_i \doteq D_i(C_i)$.

Тогда $T(S) = \max_i C_i = \max_i (D_i + M_i)$.

Из предложения 1 вытекают следующие два предложения.

Предложение 6. *При любом $i \in \mathcal{M}$ справедливо неравенство $D_i \leqslant (m-1)K$.*

Предложение 7. *При любом $i \in \mathcal{M}$ справедливо соотношение $C_i = M_i + D_i \leqslant M_i + (m-1)K$.*

Из предложения 7 следует оценка (4) длины любого жадного расписания.

Машину, заканчивающую работу последней, будем называть *критической* и обозначать i^*. Таким образом, $T(S) = C_{i^*}$. Через $\tilde{t}_i$ обозначим момент начала первого простоя на машине i.

Операции, выполняемые на машине i после момента $\tilde{t}_i$, будем называть *постоперациями*, а операции, являющиеся постоперациями на критической машине (равно как и работы, которым они принадлежат), — *критическими*. Согласно предложению 1 для всякой критической работы j одна из ее операций o_i^j (назовем ее *предкритической*) выполняется в интервале $I(o_i^j)$, включающем $\tilde{t}_{i^*}$.

Из предложения 7 следует

Предложение 8. *Если $\Delta \geqslant (m-1)K$, то машина-доминанта является критической.*

Поскольку в системе OS все операции на машине принадлежат разным работам, справедливо

Предложение 9. *Для любого жадного расписания и любой машины $i \in \mathcal{M}$ не более $m-1$ ее операций являются постоперациями.*

Из предложения 9 следует

Предложение 10. *При любом $i \in \mathcal{M}$ справедливо неравенство $\tilde{t}_i \geqslant M_i - (m-1)K$.*

Приоритетной укладкой S^p назовем такое (вообще говоря, недопустимое) расписание, в котором каждая машина $i \in \mathcal{M}$ начиная с нулевого момента времени работает без простоев, выполняя операции из множества F_i в соответствии с приоритетным порядком p^i.

Таким образом, время старта $\hat{s}_i(j)$ операции o_i^j в укладке S^p определяется формулой

$$\hat{s}_i(j) = \sum_{\{o_i^q \mid o_i^q \succ_p o_i^j\}} t_i^q.$$

В следующих трех предложениях речь идет о том, насколько время старта s_{ji} операции o_i^j в жадном расписании S, построенном для

заданных приоритетных порядков $\{p^1, \ldots, p^m\} = p$, может отличаться от времени старта этой операции в определенной для тех же порядков приоритетной укладке $S^p = \{\hat{s}_i(j)\}$.

Соотношение $s_{ji} < \hat{s}_i(j)$ возможно лишь тогда, когда операция o_i^j в расписании S выполняется раньше нескольких более приоритетных операций машины i. Как следует из § 2, выбор очередной операции на освободившуюся машину i делается не более чем из m первых операций списка U_i. Поэтому в очереди p^i операция o_i^j может опередить не больше, чем $m-1$ более приоритетных операций. Поскольку все остальные более приоритетные операции машины i к моменту назначения времени s_{ji} уже выполнены, справедливо

Предложение 11. *При любых* $j \in \mathcal{N}$, $i \in \mathcal{M}$ *справедливо неравенство* $s_{ji} \geqslant \hat{s}_i(j) - (m-1)K$.

Заметим, что при любом $t \leqslant s_{ji}$ интервал $[0, t)$, за вычетом o_i^j-задержек и o_i^j-простоев, заполнен интервалами выполнения более приоритетных, чем o_i^j, операций. Отсюда с учетом предложения 5 получаем

Предложение 12. *Для любых операции* o_i^j *и момента времени* $t \leqslant s_{ji}$ *справедлива оценка* $\hat{s}_i(j) \geqslant t - (k'_{ji}(t) + k''_{ji}(t))K$.

Из предложения 12 при $t = s_{ji}$ вытекает

Предложение 13. *При любых* $j \in \mathcal{N}$, $i \in \mathcal{M}$ *справедливо неравенство* $s_{ji} \leqslant \hat{s}_i(j) + (2m-2)K$.

Следующее предложение потребует небольшого доказательства.

Предложение 14. *Пусть при* $m \in \{2, 3\}$ *задано жадное расписание* S, *в котором машина* i' *имеет* $m-1$ *постопераций. Тогда одна из машин* $i \neq i'$ *до момента* $\tilde{t}_{i'}$ *работает без простоев.*

Доказательство. Пусть для определенности $i' = 1$ и $o_1^2, \ldots, o_1^m$ — постоперации машины 1. Предшествующий им простой на машине 1 в интервале $[\tilde{t}_1, \tilde{t}_1 + \varepsilon)$ означает, что в этом интервале операции работ $2, \ldots, m$ выполняются на машинах $2, \ldots, m$. Пусть такими операциями являются $o_2^2, \ldots, o_m^m$. Предположим, что в расписании S имеются интервалы простоя, предшествующие моменту $\tilde{t}_1$. Из них рассмотрим тот простой I', который оканчивается последним. (Пусть он принадлежит машине 2.) В силу предложения 1 и соотношения

$$I' \prec I(o_2^2) \prec I(o_1^2) \tag{6}$$

имеем

$$I' \subseteq I(o_3^2) \cup \ldots \cup I(o_m^2), \tag{7}$$

что невозможно при $m = 2$. Поэтому при $m = 2$ предложение 14 доказано. При $m = 3$ из (6) и (7) следует, что операция o_3^2 является первой операцией работы 2 в расписании S. Поэтому согласно предложению 1 на машине 3 не существует простоев, предшествующих $I(o_3^2)$. Поскольку их нет также в интервале между $I(o_3^2)$ и $\tilde{t}_1$ (по определению I'), до момента $\tilde{t}_1$ машина 3 не простаивает. Предложение 14 доказано.

Замечание. При построении алгоритмов $\mathcal{A}_1 - \mathcal{A}_3$ в § 4, 5 будут использоваться жадные алгоритмы. Мы предполагаем, что на втором этапе их реализация совпадает с изложенной в § 2. Поэтому для каждого из них достаточно описать первый этап — построение приоритетных порядков. (В алгоритме $\mathcal{A}_1$ эти порядки произвольны, поэтому первый этап также не нуждается в описании.)

§ 4. Точное решение задачи OS при условии доминирования

Лемма 4. *Если при $m \leqslant 3$ одна из машин доминирует над остальными на величину $\Delta \geqslant (m-1)K$, то $T(S_{\text{opt}}) = M$ и оптимальное расписание строится любым жадным алгоритмом.*

Доказательство. Пусть S — жадное расписание. Из предложения 7 получаем, что доминанта является критической машиной, т. е. $C_{i^d} = T(S)$. Предположим, что $T(S) > M$ и $\tilde{t} = \tilde{t}_{i^d}$ — начало простоя на машине i^d. Тогда машина i^d имеет либо одну, либо две постоперации. Если имеется $m-1$ таких операций, то по предложению 14 в интервале $[0, \tilde{t} + \varepsilon)$ при $\varepsilon > 0$ одна из машин-минорант i работает без простоев. Но тогда $M_i \geqslant \tilde{t} + \varepsilon > M - (m-1)$, что противоречит неравенству $\Delta \geqslant m - 1$.

Пусть $m = 3$ и имеется единственная критическая операция $o_{i^d}^j$. Так как предкритической операции o_i^j может предшествовать не более одной $o_{i^d}^j$-предоперации, ввиду предложения 1 имеем

$$1 \geqslant D_i(\tilde{t}) = \tilde{t} - (M_i - w_i(\tilde{t})) \geqslant M - 1 - M_i + w_i(\tilde{t}) \geqslant m - 2 + w_i(\tilde{t}) > 1.$$

Лемма 4 доказана.

Теорема 1. *Пусть в системе OS с $m \geqslant 3$ машинами и n работами одна из машин доминирует над остальными на величину $\Delta \geqslant (2m-4)K$. Тогда $T(S_{\text{opt}}) = M$ и оптимальное расписание находится с трудоемкостью $O(nm^2)$.*

Доказательство. При $m = 3$ утверждение теоремы следует из лемм 3 и 4. Поэтому достаточно рассмотреть случай $m \geqslant 4$. Алгоритм $\mathcal{A}_1$ построения оптимального расписания состоит из двух этапов.

На первом этапе согласно произвольному жадному алгоритму строим расписание S. Так как $\Delta \geqslant 2m-4 \geqslant m-1$, по предложению 7 доминанта i^d является критической машиной. Если при этом $D_{i^d}=0$, то $T(S)=M$. Следовательно, в этом случае теорема доказана. Пусть $D_{i^d}>0$ и $\tilde t=\tilde t_{i^d}$ — начало простоя критической машины. В этом случае второй этап алгоритма $\mathcal{A}_1$ перестраивает расписание S в оптимальное расписание длины M.

Пусть $o^j_{i^d}, o^j_{i'}$ — критическая и предкритическая операции работы j. С учетом предложений 1 и 10 имеем

$$m-2 \geqslant D_{i'}(\tilde t)=\tilde t-(M_{i'}-w_{i'}(\tilde t)) \geqslant M-(m-1)-M_{i'}+w_{i'}(\tilde t)$$
$$\geqslant \Delta-(m-1)+w_{i'}(\tilde t) \geqslant m-3+w_{i'}(\tilde t). \quad (8)$$

Поскольку крайние левая и правая части в этой цепочке соотношений отличаются менее чем на единицу, каждое неравенство критическое. Так, например, неравенство $\tilde t \geqslant M-(m-1)$ нельзя заменить неравенством $\tilde t \geqslant M-(m-2)$. Поэтому

$$\tilde t < M-(m-2) \quad (9)$$

и количество критических операций в точности равно $m-1$. Кроме того, неравенство $m-2 \geqslant D_{i'}(\tilde t)$ нельзя заменить неравенством $m-3 \geqslant D_{i'}(\tilde t)$. Учитывая предложение 1, заключаем, что для любой критической операции $o^j_{i^d}$ до момента $\tilde t$ выполняется ровно $m-2$ $o^j_{i^d}$-предопераций. Значит, после $\tilde t$ не выполняется других операций критических работ, кроме критических и предкритических. Это позволяет осуществить простую перестройку расписания S в нежадное расписание S', устраняющую простои на машине i^d.

Изменим расписание критических операций, выполнив их в произвольном порядке в интервале $[\tilde t, M)$. Пусть операция $o^{j'}_{i^d}$ выполняется первой из них в интервале $[\tilde t, \tilde t+t_{j'i^d})$. Тогда остальные $m-2$ критических операций $o^j_{i^d}$, $j \neq j'$, начинаются в расписании S' не раньше момента $M-(m-2)$. В то же время из (8) имеем

$$\tilde t+w_i(\tilde t) \leqslant M_i+m-2 \leqslant M-m+2.$$

Следовательно, предкритическая операция любой критической работы j, $j \neq j'$, не пересечется с критической операцией этой работы. Чтобы устранить единственное пересечение операции $o^{j'}_{i^d}$ с предкритической операцией $o^{j'}_{i'}$, интервал $I(o^{j'}_{i'})$ переместим в интервал $[M-t_{j'i'}, M)$.

Тогда, во-первых, операция $o_{i'}^{j'}$ не пересечется с другими операциями машины i', поскольку при $m \geqslant 4$ имеет место соотношение

$$C_{i'} \leqslant M_{i'} + m - 1 \leqslant M - m + 3 \leqslant M - 1 \leqslant M - t_{j'i'},$$

а во-вторых, она не пересечется с операцией $o_{i^d}^{j'}$, поскольку согласно (9)

$$\tilde{t} + t_{j'i^d} < M - (m-2) + 1 \leqslant M - 1.$$

Таким образом, расписание S' допустимо и выполняется в интервале $[0, M)$. Поэтому оно оптимально.

Для завершения доказательства теоремы остается заметить, что перестройка исходного жадного расписания S в оптимальное была выполнена за $O(m)$ операций. Поскольку в качестве S мы брали произвольное жадное расписание (с произвольными приоритетными порядками), с учетом леммы 3 получаем верхнюю оценку трудоемкости алгоритма $\mathcal{A}_1$, о которой говорилось в теореме 1. Теорема 1 доказана.

В связи с доказанным в теореме 1 утверждением возникает задача о нахождении наименьшей функции $\Delta^*(m)$, для которой доминирование с величиной $\Delta = \Delta^*(m)K$ влечет свойство (3) оптимального расписания. Из теоремы 1 имеем оценку $\Delta^*(m) \leqslant 2m - 4$. Следующий пример показывает, что

$$\Delta^*(m) \geqslant m - 1. \tag{10}$$

Действительно, если $M = M_1 = m - \varepsilon$, $M_2 = \cdots = M_m = 1$ и одна из работ состоит из m операций единичной длины, то $T(S) = m > M$, в то время как $\Delta = m - 1 - \varepsilon$.

Из леммы 4 следует, что оценка (10) точна при $m = 2, 3$, т. е. $\Delta^*(2) = 1$, $\Delta^*(3) = 2$. Мы предполагаем, что при любом m

$$\Delta^*(m) = m - 1,$$

т. е. доминирование с величиной $\Delta \geqslant (m-1)K$ влечет свойство (3). Пока удается подтвердить справедливость этого предположения при дополнительном условии. Именно, справедлива

Теорема 2. *Пусть в системе* OS *с m машинами и n работами выполняется условие*

$$M \geqslant (5.45m - 7)K \tag{11}$$

и одна из машин доминирует над остальными на величину

$$\Delta \geqslant (m-1)K. \tag{12}$$

Тогда $T(S_{\mathrm{opt}}) = M$ и оптимальное расписание находится с помощью жадного алгоритма трудоемкости $O(n^2m^2)$.

Доказательство. При $m \leqslant 3$ утверждение теоремы следует из леммы 4. Покажем, что при $m \geqslant 4$ описываемый ниже жадный алгоритм $\mathcal{A}_2$ позволяет строить допустимое расписание S длины M. Ввиду замечания в конце § 3 достаточно описать лишь первый этап алгоритма $\mathcal{A}_2$.

Для каждой работы $j \in \mathcal{N}$ определен вектор $x_j = (t_{j1}, \dots, t_{jm}) \in [0,1]^m$. Ясно, что $\sum_{j=1}^n x_j = (M_1, \dots, M_m)$. Опираясь на лемму 1, за $O(n^2m^2)$ операций найдем перестановку $\pi = (\pi_1, \dots, \pi_n)$ такую, что при любых $i = 1, \dots, m$ и $l = 1, \dots, n$ выполняется равенство

$$\sum_{j=1}^{l} t_{\pi_j i} = \frac{(l-m+1)}{n} M_i + \lambda_i^l, \tag{13}$$

где

$$\lambda_i^l \in [-1/m, m-1]. \tag{14}$$

Далее считаем, что работы перенумерованы согласно перестановке π. Таким образом, в новой нумерации имеем $\pi = (1, 2, \dots, n)$. Полагаем $p^{i^d} := (1, 2, \dots, n)$, $p^i = (p_1^i, \dots, p_n^i) := (n, n-1, \dots, 1)$ при $i \neq i^d$. Алгоритм $\mathcal{A}_2$ описан.

Из предложения 8 и соотношения (12) следует, что доминанта i^d является критической машиной в расписании S, т. е. $T(S) = C_{i^d}$. Таким образом, для доказательства свойства (3) оптимального расписания достаточно показать, что $D_{i^d} = 0$, т. е. доминанта не простаивает.

Предположим противное, и пусть

$\tilde{t} = \tilde{t}_{i^d}$ — начало простоя на машине i^d,

$o_{i^d}^j$ и $o_{i''}^j$ — критическая и предкритическая операции критической работы j.

Так как $k'_{ji^d}(\tilde{t}) \leqslant m-1$, $k''_{ji^d}(\tilde{t}) \leqslant m-2$, по предложению 12 имеем

$$\hat{s}_{i^d}(j) \geqslant \tilde{t} - 2m + 3 \doteq t^*. \tag{15}$$

Далее, в силу соотношений $k'_{ji''}(s_{ji''}) \leqslant m-2$, $k''_{ji''}(s_{ji''}) \leqslant m-2$, $s_{ji''} > \tilde{t} - 1$ и предложения 12 получаем

$$\hat{s}_{i''}(j) \geqslant s_{ji''} - 2m + 4 > \tilde{t} - 2m + 3 = t^*. \tag{16}$$

Пусть k' — максимальное число k такое, что при любом $i \neq i^d$

$$\hat{s}_i(p_k^i) = \sum_{j=n-k+2}^{n} t_{ji} \leqslant t^*. \tag{17}$$

По условию (16) работы $\{p_1^i, \dots, p_{k'}^i\} = \{n-k'+1, \dots, n\}$ не могут быть критическими. Следовательно, таковыми могут быть лишь работы $\{1, 2, \dots, n-k'\}$. По выбору k' при $k = k'+1$ для некоторого $i = i'$ условие (17) нарушается, т. е. имеет место неравенство $\sum_{j=n-k'+1}^{n} t_{ji'} > t^*$. Таким образом,

$$\sum_{j=1}^{n-k'} t_{ji'} < M_{i'} - t^*. \tag{18}$$

С другой стороны, согласно (15) каждая критическая операция $o_{i^d}^j$ начинается в приоритетной укладке не раньше t^* и суммарная длина таких операций равна $M - \tilde{t}$. Поэтому последняя из них заканчивается в S^p не раньше момента $t^* + M - \tilde{t}$, и тем более не раньше этого момента заканчивается в S^p последняя из операций множества $\{o_{i^d}^1, \dots, o_{i^d}^{n-k'}\}$, включающего все критические операции. Таким образом,

$$\sum_{j=1}^{n-k'} t_{ji^d} \geqslant t^* + M - \tilde{t} = M - 2m + 3. \tag{19}$$

Из (13) при $l = n - k'$, $i \in \{i', i^d\}$ и из (18) получаем

$$\sum_{j=1}^{l} t_{ji'} = \tfrac{l-m+1}{n} M_{i'} + \lambda_{i'}^l < M_{i'} - t^*,$$

$$\frac{l-m+1}{n} < 1 - \frac{t^* + \lambda_{i'}^l}{M_{i'}},$$

$$\sum_{j=1}^{l} t_{ji^d} = \frac{l-m+1}{n} M + \lambda_{i^d}^l < M + \lambda_{i^d}^l - \tfrac{M}{M_{i'}}(t^* + \lambda_{i'}^l).$$

С учетом (19) это дает

$$M - 2m + 3 < M + \lambda_{i^d}^l - \tfrac{M}{M_{i'}}(t^* + \lambda_{i'}^l),$$

$$M(t^* + \lambda_{i'}^l) < M_{i'}(\lambda_{i^d}^l + 2m - 3).$$

Отсюда с учетом формул (12), (14), (15) и предложения 10 получаем

$$M(\tilde{t} - 2m + 3 - \tfrac{1}{m}) < (M - (m-1))(3m-4),$$
$$M(M - 3m + 4 - \tfrac{1}{m}) < (M - (m-1))(3m-4),$$
$$M^2 - (6m - 8 + \tfrac{1}{m})M + (m-1)(3m-4) < 0.$$

Следовательно, при $m \geqslant 3$ приходим к необходимому условию

$$\begin{aligned} M &< 3m-4+\tfrac{1}{2m}+\sqrt{\left(3m-4+\tfrac{1}{2m}\right)^2-(3m-4)(m-1)} \\ &< 3m-4+\tfrac{1}{2m}+\left(\sqrt{6}m-\tfrac{17}{2\sqrt{6}}+\tfrac{74}{48\sqrt{6}m}\right) < 5.45m-7, \end{aligned}$$

которое не выполняется ввиду (11).

Таким образом, наше предположение о существовании простоя на доминанте оказалось неверным. Следовательно, $T(S) = M$ и построенное согласно алгоритму $\mathcal{A}_2$ расписание S оптимально. Теорема 2 доказана.

§ 5. Быстрый алгоритм построения приближенных расписаний

Теорема 3. *Задача* OS *с m машинами и n работами приближенно решается с помощью жадного алгоритма трудоемкости $O(nm^2)$, позволяющего для всякого входа задачи строить расписание S длины*

$$T(S) \leqslant M + \left(\left\lceil \frac{(3m^2-2m)K}{M+2mK} \right\rceil - 1\right) K. \tag{20}$$

Доказательство. Будем предполагать выполненным условие

$$M_i = M, \quad i \in \mathcal{M}. \tag{21}$$

Это не ограничивает общности, так как всегда можно добиться выполнения соотношения (21) увеличением длительностей отдельных операций на «недогруженных машинах», оставляя без изменений величины m, n, M и K (трудоемкость такой процедуры — $O(nm)$), найти расписание $S = \{s_{ji}\}$ для полученного входа задачи, а затем вернуться к исходным длительностям операций, сохраняя то же расписание. Очевидно, что это расписание останется допустимым, а его длина будет удовлетворять найденной верхней оценке (20).

Опишем жадный алгоритм $\mathcal{A}_3$ построения расписания S с оценкой (20). Ввиду замечания в конце § 3 достаточно описать лишь первый этап алгоритма $\mathcal{A}_3$. Положим

$$l \doteq \left\lceil \frac{3m^2-2m}{M+2m} \right\rceil - 1. \tag{22}$$

Если $M < m-2$, то длина $T(S)$ любого расписания S, полученного жадным алгоритмом, будет удовлетворять оценке (20). Поэтому далее

считаем выполненным условие $M \geqslant m - 2$, которое обеспечивает справедливость неравенства $m - l - 1 \geqslant 0$, необходимого для корректности выражений (23), (24).

Пусть

F_i — множество операций машины i,

O^j — множество операций работы j,

$\mathcal{O}$ — множество всех операций системы.

Тогда разбиения $\mathcal{O} = \bigcup_{i\in\mathcal{M}} F_i$ и $\mathcal{O} = \bigcup_{j\in\mathcal{N}} O^j$ множества $\mathcal{O}$ удовлетворяют условиям леммы 2 при $k = m$, $\Psi = \mathcal{O}$, $\Psi_i = F_i$, $\Psi^j = O^j$. Положив $x_\psi = t_{ji}$ для $\psi = o_i^j \in \mathcal{O}$, получим семейство чисел $\{x_\psi \in \mathbb{R}^+ \mid \psi \in \Psi\}$, удовлетворяющее соотношениям $X_i = \sum_{\psi\in\Psi_i} x_\psi = M_i$, $K_i = \max\limits_{\psi\in\Psi_i} x_\psi \leqslant K$. Таким образом, оказавшись в условиях леммы 2 и опираясь на ее утверждение, при $q = m - l - 1$ мы имеем возможность с помощью алгоритма трудоемкости $O(nm^2)$ найти подмножество операций $\mathcal{O}' \subset \mathcal{O}$, удовлетворяющее соотношениям

$$\left|\mathcal{O}' \cap O^j\right| = m - l - 1, \quad j \in \mathcal{N}, \tag{23}$$

$$\left|\sum_{o_i^j \in \mathcal{O}'\cap F_i} t_{ji} - \frac{m-l-1}{m} M\right| < 1, \quad i \in \mathcal{M}. \tag{24}$$

После этого для каждой машины $i \in \mathcal{M}$ определим приоритетный порядок p^i так, что в нем сначала идут в произвольном порядке работы $j \in \mathcal{N}$, для которых $F_i \cap \mathcal{O}' \cap O^j \neq \varnothing$, а затем — все остальные. Иначе говоря, на машине i в приоритетной укладке S^p выполняются сначала операции из $\mathcal{O}'$, а затем все остальные. Алгоритм $\mathcal{A}_3$ описан.

Из описания алгоритма $\mathcal{A}_3$ и леммы 3 видно, что трудоемкость алгоритма на каждом этапе равна $O(nm^2)$, а значит, такова же трудоемкость алгоритма в целом. Остается доказать, что работающий на определенных выше приоритетных порядках жадный алгоритм строит расписание S такое, что $T(S)$ удовлетворяет оценке (20). Пусть это не так, т. е. $D_{i^*} > l$, где i^* — критическая машина, и пусть $o_{i^*}^j$ — операция, выполняемая последней. По предложению 1 в покрытии $o_{i^*}^j$-простоев машины i^* участвует не менее $l+1$ $o_{i^*}^j$-предопераций. Такие предоперации назовем *покрывающими*.

Пусть

Ψ_1 — множество покрывающих операций, дополненное операцией $o_{i^*}^j$;

$o_{i'}^j$ — предкритическая $o_{i^*}^j$-предоперация.

Из определения предкритической операции следует, что $o^j_{i'} \in \Psi_1$ и среди операций множества Ψ_1 операция $o^j_{i'}$ выполняется первой, причем

$$\tilde{t} - 1 < s_{ji'} \leqslant \tilde{t} < s_{ji'} + t_{ji'}. \tag{25}$$

Имеем $k''_{ji^*}(s_{ji'}) \leqslant m - l - 2, k'_{ji^*}(s_{ji'}) \leqslant m - l - 2$. Следовательно, по предложению 12 все операции множества Ψ_1 начинаются в приоритетной укладке S^p позже момента

$$\dot{t} \doteq s_{ji'} - k'_{ji^*}(s_{ji'}) - k''_{ji^*}(s_{ji'}) \geqslant s_{ji'} - 2(m - l - 2). \tag{26}$$

В то же время согласно (23), (24) и правилу построения приоритетных порядков в алгоритме $\mathcal{A}_3$ заключаем, что $m - l - 1$ операций множества $\mathcal{O}' \cap O^j \doteq \Psi_2$ заканчиваются в S^p раньше момента $\frac{m-l-1}{m}M + 1 \doteq \bar{t}$. Так как сумма мощностей множеств Ψ_1 и Ψ_2 не меньше $(l+2)+(m-l-1) = m + 1 > m$ и все операции из Ψ_1 и Ψ_2 являются операциями работы j, имеем $\Psi_1 \cap \Psi_2 \neq \varnothing$.

Таким образом, существует «общая» операция $o^j_{i_0} \in \Psi_1 \cap \Psi_2$, начинающаяся в S^p позже $\dot{t}$ и заканчивающаяся раньше $\bar{t}$. Очевидно, что ее длительность t_{ji_0} меньше длины интервала $(\dot{t}, \bar{t})$. Отсюда с учетом (22), (25), (26) получаем

$$\begin{aligned} t_{ji_0} &< \bar{t} - \dot{t} = M - \frac{l+1}{m}M + 1 + 2(m - l - 2) - s_{ji'} \\ &\leqslant M - (m-1) - s_{ji'} = \tilde{t} + \Sigma_{cr} - (m-1) - s_{ji'} \leqslant \tilde{t} - s_{ji'} < 1. \end{aligned} \tag{27}$$

Здесь Σ_{cr} — сумма длительностей критических операций, число которых в силу предложения 9 не превосходит $m - 1$.

Так как разность между крайней левой и крайней правой частями соотношения (27) меньше единицы, оценка (26) на величину $\dot{t}$ точна. Следовательно, $|\Psi_1| = l + 2$ и $|\Psi_1 \cap \Psi_2| = 1$.

Покажем, что критическая операция $o^j_{i^*}$ не может быть «общей» операцией. Предположим противное. Обозначив через Σ'_{cr} сумму длительностей остальных критических операций, из (27) получим

$$t_{ji^*} < \tilde{t} + t_{ji^*} + \Sigma'_{cr} - (m-1) - s_{ji'} \leqslant \tilde{t} + t_{ji^*} - s_{ji'} - 1.$$

Поэтому $\tilde{t} - s_{ji'} > 1$, что противоречит (25).

В силу (25), (27) операция $o^j_{i'}$ также не может быть «общей». Наконец, пусть $o^j_{i_0}$ — покрывающая операция, отличная от $o^j_{i'}$. Так как в покрытии $o^j_{i^*}$-простоев участвует лишь часть интервала $I(o^j_{i'})$, не превосходящая по длине $1 - (\tilde{t} - s_{ji'})$, а длительность покрывающей операции $o^j_{i_0}$ меньше $\tilde{t} - s_{ji'}$, заключаем, что общая длительность всех $l + 1$

покрывающих операций, участвующих в покрытии $o^j_{i^*}$-простоев, меньше l, что противоречит допущению $D_{i^*} > l$. Таким образом, допущение $D_{i^*} > l$, оказалось неверным. Следовательно, оценка (20) справедлива. Теорема 3 доказана.

Отметим несколько частных случаев теоремы 3, которые представляют самостоятельный интерес. Для упрощения формулировок полагаем $K = 1$.

Следствие теоремы 3.

1. *Если для некоторого* $i \in \{1, \dots, m-1\}$ *выполняется неравенство*

$$M \geqslant m - 2 + 3i + (3i-2)i/(m-i),$$

то $T(S) \leqslant M + m - 1 - i$. *В частности, при* $i = 2$ *и* $i = 1$ *получаем следующие утверждения*:

(a) *если* $M \geqslant m + 4 + \frac{8}{m-2}$, *то* $T(S) \leqslant M + m - 3$;

(b) *если* $M \geqslant m + 1 + \frac{1}{m-1}$, *то* $T(S) \leqslant M + m - 2$.

В случае трех машин отсюда получаем

(c) *если* $M \geqslant 4\frac{1}{2}$, *то* $T(S) \leqslant M + 1$.

2. *Если* $M \geqslant 4m - 4$, *то* $T(S) \leqslant M + (m-1)/2$.
3. *Если* $M \geqslant 7m - 6$, *то* $T(S) \leqslant M + (m-1)/3$.
4. *Если* $M \geqslant \frac{3}{4}m^2 - \frac{5}{2}m$, *то* $T(S) \leqslant M + 3$.

Список обозначений

m — количество машин;
$\mathcal{M} = \{1, \dots, m\}$ — множество машин;
n — количество работ;
$\mathcal{N} = \{1, \dots, n\}$ — множество работ;
o^j_i — операция работы j на машине i;
$O^j = \{o^j_i \mid i \in \mathcal{M}\}$ — операции работы j;
F_i — множество операций машины i;
t_{ji} — длительность операции o^j_i;
$S = \{s_{ji}\}$ — расписание;
s_{ji} — время старта операции o^j_i в расписании S;
S_{opt} — оптимальное расписание;
$M_i = \sum_{j=1}^n t_{ji}$;
$M = \max\limits_i M_i$;
$K = \max\limits_{ji} t_{ji}$;
$T(S)$ — длина расписания S;

i^d — машина-доминанта;

Δ — величина доминирования;

$\mathbb{R}^m$ — m-мерное векторное пространство;

$\pi = (\pi_1, \dots, \pi_n)$ — перестановка чисел $\{1, 2, \dots, n\}$;

$p^i = (p_1^i, \dots, p_n^i)$ — приоритетный порядок работ на машине i;

$p = \{p^1, \dots, p^m\}$;

$\succ_p$ — отношение приоритетного порядка на множествах F_i;

$S^p = \{\hat{s}_i(j) \mid j \in \mathcal{N},\ i \in \mathcal{M}\}$ — приоритетная укладка;

$\hat{s}_i(j)$ — время старта операции o_i^j в S^p;

$J_i(t)$ — работа, выполняемая на машине i в момент t;

$J(t) = \{J_i(t) \mid i \in \mathcal{M}\} \setminus \{0\}$ — фронт работ в момент t;

$I(o_i^j) = [s_{ji}, s_{ji} + t_{ji})$ — интервал выполнения операции o_i^j в расписании S;

$\hat{t}$ — граница определенности расписания S;

$\hat{J} = \{J_1, \dots, J_m\} = J(\hat{t})$ — текущий фронт работ;

$U_i(t) = \bigl(u_1^i(t), \dots, u_{a_i}^i(t)\bigr)$ — список невыполненных работ на машине i к моменту t, упорядоченный согласно приоритетному порядку p^i;

a_i — длина текущего списка $U_i(\hat{t})$;

$\Phi[j]$ — номер машины, на которой выполняется работа j текущего фронта (если $j \notin \hat{J}$, то $\Phi[j] = 0$);

$B[i, i']$ — элемент битового массива B, принимающий для простаивающей машины i значение 1, если и только если $J_{i'} \in U_i$;

$D_i(t)$ — длительность простоя машины i в интервале $[0, t]$;

C_i — момент завершения машиной i последней операции в расписании S;

$D_i = D_i(C_i)$;

i^* — критическая машина;

$\tilde{t}_i$ — момент начала первого простоя машины i;

$w(\tilde{t}, i)$ — продолжительность чистой работы машины i после момента $\tilde{t}$;

Σ_{cr} — сумма длительностей критических операций;

$k'_{ji}(t)$ — количество o_i^j-предопераций, начинающихся строго до момента t;

$k''_{ji}(t)$ — количество o_i^j-предопераций, заканчивающихся до момента t.

ЛИТЕРАТУРА

1. **Gonzalez T., Sahni S.** Open shop scheduling to minimize finish time // J. Assoc. Comput. Math. 1976. V. 23, N 4. P. 665–679.
2. **Graham R. L., Lawler E. L., Lenstra J. K., Rinnooy Kan A. H. G.** Optimization and approximation in deterministic sequencing and scheduling: a survey // Ann. Discrete Math. Amsterdam: North Holland. 1979. V. 5. P. 287–326.
3. **Fiala T.** An algorithm for the open-shop problem // Math. Oper. Res. 1983. V. 8, N 1. P. 100–109.
4. **Bárány I., Fiala T.** Nearly optimum solution of multimachine scheduling problems // Szigma. — Mat.-Közgaz-dasagi Folyóirat. 1982. V. 15, N 3. P. 177–191.
5. **Севастьянов С. В.** Полиномиально разрешимый случай задачи open-shop с произвольным числом машин // Кибернетика и системный анализ. 1992. № 6. С. 135–154.
6. **Севастьянов С. В.** О компактном суммировании векторов // Дискретная математика. 1991. Т. 3, № 3. С. 66–72.
7. **Танаев В. С., Сотсков Ю. Н., Струсевич В. А.** Теория расписаний. Многостадийные системы. М.: Наука, 1989.

Адрес автора:

РОССИЯ,
630090, Новосибирск,
Университетский пр., 4,
Институт математики СО РАН

Статья поступила

4 октября 1993 г.