

требности в отливках каждого предприятия, области и экономического района в целом, по рациональному развитию и оптимальному размещению литейного производства и повышению экономической эффективности промышленного производства в целом.

ЛИТЕРАТУРА

1. Ю. А. Нехендзп. О производстве, структуре и потреблении литья в СССР и за рубежом. Л., 1966. (Ленингр. Дом н.-техн. пропаганды).
2. В. М. Шестопал. Специализация и проектирование литейных цехов и заводов. М., «Машиностроение», 1969.
3. Я. А. Гольбин, В. А. Герман. Проблемы экономики литейного производства. Минск, «Наука и техника», 1971.
4. А. Н. Комин. Новые оптимальные цены на продукцию машиностроения. Экон. газета, 1972, № 20.

Поступила в редакцию
6 III 1972

ЗАМЕЧАНИЕ К АЛГОРИТМУ ДЖОНСОНА

Э. А. ЛОГИНОВ

(Москва)

В [1] дан изящный алгоритм для частной задачи календарного планирования: n деталей должны быть обработаны на двух станках, причем каждая *сначала* на станке A , *затем* на станке B (назовем последнее требование условием $(*)$). Зная длительность обработки каждой детали на каждом станке, требуется найти оптимальный порядок обработки деталей на станках, при котором общее время обработки минимально. Назовем общей задачу, в которой нет условия $(*)$, т. е. в которой часть деталей идет сначала на станок A , затем на B , а другая часть проходит станки в обратной последовательности: сначала B , потом A . Задачу же с условием $(*)$ так и будем называть задачей с $(*)$.

Наша первая цель — показать, что общая задача не труднее задачи с $(*)$, т. е. что алгоритм из [1] приводит к решению и общей задачи. Пусть алгоритмом [1] решены две задачи с $(*)$: одна отдельно для тех m из n деталей, которые вначале проходят станок A и уже затем B , и вторая для остальных $n - m$ деталей с обратным порядком обработки на станках. Обозначим через t_1 — минимальное общее время обработки группы I из m деталей; t_2 — аналогичное время для группы II . Оптимальную последовательность обработки в этих частных задачах считаем найденной. Именно, пусть детали группы I проходят обработку на обоих станках в одинаковой последовательности $I_1, I_2, \dots, I_m$ (в [1] показано, что порядок обработки деталей на станке A можно считать тем же, что и на станке B). Аналогично обработка группы II ведется в порядке $II_1, II_2, \dots, II_{n-m}$, так что на любом станке деталь II_{i+1} , $i = 1, 2, \dots, n - m - 1$, обрабатывается после детали II_i . Тогда оптимальный порядок поступления деталей на станки A и B в общей задаче выглядит так

$$I_1, I_2, \dots, I_m, II_1, II_2, \dots, II_{n-m} \text{ (станок } A),$$

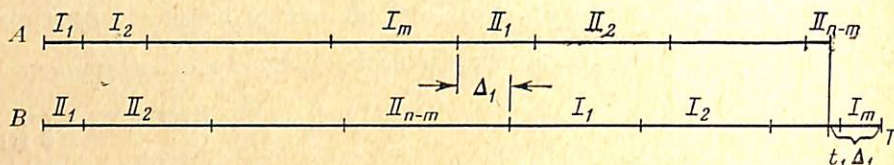
$$II_1, II_2, \dots, II_{n-m}, I_1, I_2, \dots, I_m \text{ (станок } B).$$

Докажем, что этот порядок оптимальный. Будем для определенности считать, что из всех n деталей последней обрабатывается деталь группы I , т. е. деталь I_m , а время окончания обработки детали I_m (и, значит, общее время обработки всех n деталей) обозначим через T . Так как в случае $T = t_1$, где t_1 — минимальное время обработки для группы I , оптимальность данного порядка очевидна, рассмотрим случай $T > t_1$. При $T > t_1$ все детали группы I (в общей задаче для n деталей) вступают в обработку на станке B позже, чем в задаче с $(*)$ для группы I . Действительно, если бы какая-либо деталь группы I в общей задаче вступала в обработку на станке B в тот же момент, что и в задаче с $(*)$ для группы I , то и все следующие за ней детали группы I обрабатывались бы в этих задачах одновременно, что дало бы $T = t_1$, ибо порядок обработки деталей группы I на станке A в обеих задачах один и тот же. Итак, при $T > t_1$ имеется запаздывание во времени вступления любой детали группы I в обработку на станке B в общей задаче (по сравнению с задачей для одной группы I). Это запаздывание Δ_i для детали I_i , $i = 1, 2, \dots, m$, обусловлено только тем, что в общей задаче при выбранном порядке обработки станок B непосредственно перед обработкой детали I_i занят деталью I_{i-1} (если $i \neq 1$) или деталью

Π_{n-m} (при $i = 1$), причем время простоя для детали I_i между обработкой ее на станках A и B не меньше запаздывания Δ_i . Следовательно, все детали группы I обрабатываются на станке B одна за другой непрерывно, например, как на рисунке.

Поскольку на станке B без перерыва обрабатываются и все детали группы II , то станок B в течение времени T полностью загружен, обрабатывая непрерывно детали групп II и I . Это означает, что в общей задаче T — минимально необходимое время обработки всей партии из n деталей на станках A и B , а указанная последовательность обработки оптимальна.

Остановимся на частной задаче для двух станков, решенной в [1]. В ней даны n деталей с длительностями обработки a_i (станок A) и b_i (станок B), $i = 1, 2, \dots, n$;



любая деталь поступает сначала на станок A , затем на B . Рассмотрим простое обоснование алгоритма из [1], основанное на такой лемме.

Лемма. Пусть $c_1 = \min(a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_n)$. Если длительность обработки всех деталей вместо a_i и b_i принять равными $a_i' = a_i - c$ и $b_i' = b_i - c$, $i = 1, 2, \dots, n$, где $-\infty < c \leq c_1$, то это не скажется на оптимальном порядке обработки.

Доказательство. Через $A_1^*, A_2^*, \dots, A_n^*$ и $B_1^*, B_2^*, \dots, B_n^*$ обозначим длительности обработки деталей при оптимальном порядке обработки их на станке A и B соответственно, через T^* — общее время обработки деталей при этом, через $A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_n$ и T — то же самое, но при произвольном порядке обработки. Пусть будет произвольный порядок обработки $i_1, i_2, \dots, i_n$. При фиксированном i_k вместо соответствующей i_k длительности A_{i_k} возьмем $A_{i_k}^* = A_{i_k} - c$, и если $k > 1$, $B_{i_{k-1}}^* = B_{i_{k-1}} - c$ вместо $B_{i_{k-1}}$. Тогда все детали i_s , $s > k$, следующие за i_k , вступят в обработку и на станке A , и на B на время c раньше прежнего, а время T перейдет в $T_1 = T - c$. Допустим, сделана замена A_s на $A_s^* = A_s - c$ и B_s на $B_s^* = B_s - c$, $s = 1, 2, \dots, n$, т. е. для всех деталей длительность обработки изменена одинаково на c . (Удобно говорить «уменьшена на c », помня, однако, что можно брать $c < 0$.) Это равносильно тому, что сначала мы заменяем A_1 на A_1^* , затем A_2 и B_1 — на A_2^* и B_1^* соответственно и т. д., наконец, B_n заменяем на B_n^* . Всякая такая замена уменьшает, как доказано, общее время T на величину c , а окончательное общее время после замены всех A_s и B_s на A_s^* и B_s^* , $s = 1, 2, \dots, n$, равно $T^* = T - (n+1)c$, где $-\infty < c \leq c_1$. Отсюда сразу следует справедливость леммы и, более того: минимальные общие времена обработки при новых и старых длительностях связаны равенством $T^{I*} = T^* - (n+1)c$. Лемма доказана.

Если взять $a_i^I = a_i - c_1$ и $b_i^I = b_i - c_1$, $i = 1, 2, \dots, n$, то в силу леммы для решения задачи достаточно найти оптимальный порядок обработки при этих длительностях a_i^I и b_i^I , одна из которых равна нулю. Для определенности пусть $b_n^I = 0$. Ясно, что деталь с $b_n^I = 0$ следует ставить на станок A последней. Действительно, при новых длительностях a_i^I и b_i^I через T_n^{I*} обозначим оптимальное время обработки всех n деталей, а через T_{n-1}^{I*} — всех, но без последней детали, т. е. без детали с

$b_n^I = 0$. Если $T_{n-1}^{I*} \geq \sum_{i=1}^n a_i^I$, то, оставив порядок обработки, дающий время T_{n-1}^{I*} ,

обрабатываем деталь с $b_n^I = 0$ последней. При этом общее время обработки всех n

деталей $T_n^I = T_{n-1}^{I*}$, т. е. оптимально. Если же $T_{n-1}^{I*} < \sum_{i=1}^n a_i^I$, то обработка n -й

детали последней дает общее время $T_n^I = \sum_{i=1}^n a_i^I$, т. е. также оптимальное время.

Аналогично, если $a_i^I = 0$, то деталь с $a_i^I = 0$ (не требующую обработки на станке A) следует на станке B обрабатывать первой. Таким образом, лемма позволяет свести исходную задачу к задаче с меньшим числом деталей. Поступая аналогично, находим

$c_2 = \min(a_i^I, b_i^I)$, $i = 1, 2, \dots, n-1$, и берем $a_i^{II} = a_i^I - c_2$ и $b_i^{II} = b_i^I - c_2$ и т. д. В результате определяется место в очереди детали с длительностью a_i или b_i , равной c_1 (начало очереди, если $a_i = c_1$, и конец, если $b_i = c_1$), затем детали с a_i или b_i , равной $c_1 + c_2$, и т. д., т. е. известный алгоритм из [1]. В [2], правда, задача усложнена дополнительным условием: если время начала работы станка A принять за нуль, то станок B может начать работу не раньше заданного момента Δ , $\Delta \geq 0$. Однако легко понять, что оптимальный порядок, найденный для $\Delta = 0$, остается оптимальным при $\Delta > 0$, т. е. учет Δ не портит этого порядка. В самом деле, пусть для $\Delta = 0$ (или, что то же, для $0 \leq \Delta \leq A_1^*$) найден оптимальный порядок. Не меняя порядка обработки, введем $\Delta > A_1^*$, т. е. пустим станок B в момент Δ . Если это не сдвинет время окончания обработки последней детали на станке B , то нечего и доказывать. Если сдвинет, это будет значить, что с момента Δ все детали на станке

B идут непрерывно, т. е. станок B освободится в момент $T = \Delta + \sum_{i=1}^n b_i$, а это, очевидно, минимальное время.

ЛИТЕРАТУРА

1. С. М. Джонсон. Оптимальные двух- и трехоперационные календарные планы производства с учетом подготовительно-заключительного времени. В сб. Календарное планирование. М., «Прогресс», 1966.
2. А. А. Корбут, Ю. Ю. Финкельштейн. Дискретное программирование. М., «Наука», 1969.

Поступила в редакцию
17 X 1969

ПСЕВДООБРАТНАЯ МАТРИЦА И ИНВАРИАНТНАЯ ФОРМА ДЛЯ МАТРИЦЫ БЛОЧНО-ДИАГОНАЛЬНОЙ СТРУКТУРЫ

М. С. ДУБСОН

(Москва)

Хорошо известно, что если матрица ограничений задачи линейного программирования $\min(c, x)$, $Ax = b$, $x \geq 0$ обладает специальной структурой, то ее можно учитывать в специальных алгоритмах линейного программирования, более эффективных, нежели стандартные процедуры. Однако заранее особенности структуры матрицы известны не всегда. В таких случаях нужны специальные методы для выявления особенностей структуры. В настоящей работе предлагается простой метод для выяснения, является ли данная матрица блочно-диагональной.

Введем обозначение: $R_p^{m \times n}$ — совокупность всех $m \times n$ матриц ранга p . Нам понадобятся также некоторые свойства псевдообратной матрицы A^+ , определяемой как единственное решение системы четырех матричных уравнений, называемых также уравнениями Пенроуза

$$ATA = A, \quad (1)$$

$$TAT = T, \quad (2)$$

$$(AT)' = AT, \quad (3)$$

$$(TA)' = TA. \quad (4)$$

Рассмотрим систему линейных уравнений

$$Ax = b, \quad A \in R_p^{m \times n}. \quad (5)$$

Лемма 1. Система линейных уравнений (5) совместна тогда и только тогда, когда $AA^+b = b$. В этом случае общее решение системы (5) имеет вид $A^+b + (I - A^+A)y^*$, где произвольный вектор $y \in R^n$.

* I — единичная матрица, порядок которой совпадает с порядком $A'A$, где штрих означает транспонирование.