МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ИМ. И. И. ПОЛЗУНОВА»

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Кафедра Прикладная математика

А.В. Сорокин

ИСПОЛЬЗОВАНИЕ АЛГОРИТМА ДЖОНСОНА ДЛЯ РЕШЕНИЯ ЗАДАЧИ УПОРЯДОЧЕНИЯ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к лабораторной работе по дисциплине
«Основы моделирования»

Барнаул 2021

УДК 519.8

Сорокин А.В. Использование алгоритма Джонсона для решения задачи упорядочения. Методические указания к лабораторной работе по дисциплине « Основы моделирования» / А.В. Сорокин; Алт. госуд. технич. ун-т им. И.И. Ползунова. - Барнаул, 2021. – 22 с.

В методических указаниях изложен материал по дисциплине «Основы моделирования», используемый для выполнения практических заданий. Материал содержит описание задачи упорядочения на основе производственной задачи – обработки определенного количества деталей на двух или трех обрабатывающих станках. Рассматривается вывод правил Джонсона, на основе графика Ганта, и построение алгоритма Джонсона с использованием правил Джонсона для задачи с произвольным количеством деталей и двумя станками. Показывается использование алгоритма Джонсона для решения задачи с произвольным количеством деталей и тремя станками. Материал снабжен большим количеством примеров по каждой из рассматриваемых задач упорядочения. Методические указания предназначены для студентов, обучающихся по направлению «Программная инженерия», «Информатика и вычислительная техника».

Содержание

1. Постановка задачи упорядочения

Задачи упорядочения в общем случае представляют собой задачи выбора порядка обслуживания, оптимизирующего какой-либо существенный показатель качества функционирования системы. К задачам упорядочения относятся, например, задачи выбора дисциплины обслуживания в системе массового обслуживания, задачи составления оптимальных расписаний, задачи календарного планирования производства и т.д.

В качестве примера простейшей задачи упорядочения рассмотрим так называемую задачу Беллмана-Джонсона $n \times 2$ [1].

Пусть на двух станках (A и B) необходимо обработать n разных деталей с номерами $i = 1, 2, \ldots, n$. Пусть даны нормы времени a_i и b_i обработки детали i на станках A и B соответственно и пусть задано, что маршрут обработки для всех деталей жесткий: *сначала деталь обрабатывается на станке A, затем на станке* B. При этом:

1) для каждой детали обработка на станке B может начинаться не раньше, чем окончится ее обработка на станке A;
2) на каждом станке одновременно может обрабатываться не более одной детали;
3) начавшаяся операция не прерывается до полного ее завершения.

Пусть конкретные значения величин a_i и b_i для случая $n = 5$ следующие (см.табл.1):

Будем запускать детали в производство в порядке их номеров и определим при помощи линейных диаграмм (графика Ганта) время T окончания полной обработки всех деталей (см.рис.1).

Таблица 1

i	a_i	b_i
1	4	1
2	30	4
3	6	30
4	4	5
5	2	3

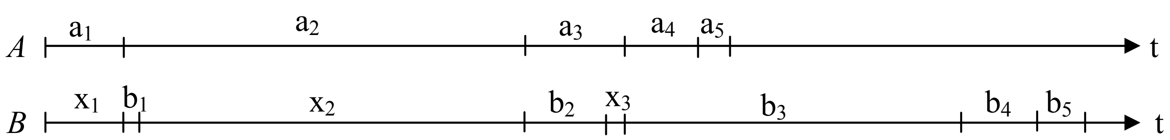


Рис.1 . График Ганта обработки пяти деталей на двух станках

Как видно из графика, пока станок A будет обрабатывать первую деталь, станок B будет простаивать, причем величина простоя $x_1 = a_1 = 4$. Так как $a_1 + a_2 > x_1 + b_1$, то и во время обработки второй детали на станке A станок B не будет загружен полностью. Время простоя $x_2 = a_1 + a_2 - x_1 - b_1 = 4 + 30 - 4 - 1 = 29$. Так как $a_1 + a_2 + a_3 > x_1 + b_1 + x_2 + b_2$, то будет иметь место еще один простой станка B, причем $x_3 = a_1 + a_2 + a_3 - x_1 - b_1 - x_2 - b_2 = 4 + 30 + 6 - 4 - 1 - 29 - 4 = 2$. Так как $a_1 + a_2 + a_3 + a_4 < x_1 + b_1 + x_2$

+ b_2 + x_3 + b_3, то очередного возможного простоя станка В не будет, т.е. $x_4 = 0$. Аналогично получаем, что и $x_5 = 0$. Тогда общее время простоев станка В будет равно $X = x_1 + x_2 + x_3 + x_4 + x_5 = 4 + 29 + 2 + 0 + 0 = 35$, а общее время полной обработки всех деталей $T = T_B + X = b_1 + b_2 + b_3 + b_4 + b_5 + X = 1 + 4 + 30 + 5 + 3 + 35 = 43 + 35 + = 78$.

Нетрудно заметить, что величина простоев X (и, следовательно, время T окончания полной обработки всех деталей) будет зависеть от последовательности, в которой детали обрабатываются. Например, если мы вместо последовательности 1-2-3-4-5 воспользуемся обратной последовательностью 5-4-3-2-1, то получим $x_1 = 2$, $x_2 = 1$, $x_3 = 1$, $x_4 = x_5 = 0$, откуда будет следовать, что X = 4 и T = 43 + 4 = 47. Как видим, получили существенный выигрыш: величина простоев станка В уменьшилась чуть ли не в 9 раз, а общее время обработки чуть ли не на 40 %. Отсюда <u>приходим к задаче упорядочения: выбрать такую последовательность обработки деталей, при которой время T окончания полной обработки всех деталей, или, что то же самое, общее время простоев было минимальным</u>.

Для n деталей существует $n!$ различных последовательностей. Если эти последовательности пронумеровать и обозначить текущее значение номера последовательности через s $(s = 1, 2, \ldots, n!)$, то мы сможем задачу Джонсона $n \times 2$ упорядочения обработки записать сокращенно как

- Задачу определения такого номера $s^{опт}$, при котором X будет минимальным, т.е.

$$X^{опт} = \min_s X(s).$$

При больших n число $n!$ возможных вариантов становится настолько огромным, что решить задачу выбора оптимальной последовательности обработки деталей методом простого перебора и сравнения вариантов становится чрезмерно трудоемким делом даже для компьютера. Отсюда приходим к необходимости разработки более эффективных алгоритмов упорядочения, позволяющих отыскать оптимальную последовательность $s^{опт}$ (или последовательность, для которой $X(s)$ достаточно близко к минимальному значению $X(s^{опт})$ за разумное время. Некоторые из таких алгоритмов, в том числе и алгоритм решения задачи Джонсона n × 2, будут рассмотрены ниже.

Мы рассмотрели очень простой пример задачи упорядочения. На практике встречаются значительно более сложные задачи.

Во-первых, число станков, как правило, бывает значительно больше двух, поэтому вместо задачи n × 2 приходится рассматривать более общую задачу $n \times m$;

Во-вторых, маршруты обработки деталей не всегда бывают жесткими, что приводит не только к дополнительным возможностям минимизации простоя, но и к значительному увеличению числа возможных вариантов обработки;

В-третьих, детали, как правило, обрабатываются не по одной, а партиями, которые в случае необходимости можно дробить и изменять как в сторону уменьшения, так и в сторону увеличения, а также обрабатывать на параллельном оборудовании, так как одинаковых станков может быть несколько;

В-четвертых, некоторые виды оборудования являются взаимозаменяемыми на тех или иных операциях, но могут отличаться своей производительностью и/или стоимостью обработки;

В-пятых, приходится учитывать затраты времени не только на обработку и технологическое **пролеживание**, но и на транспортировку, а также возможность ожидания уже готовой партией транспортного средства и возможность передачи на другое рабочее место неполной партии;

В-шестых, в некоторых случаях приходится учитывать возможность брака и повторной обработки деталей, возможность отказов в работе из-за неисправностей оборудования, невыхода рабочего, нехватки материалов и т.п.;

В-седьмых, иногда приходится учитывать зависимость нормативов продолжительности операций от смены, размеров партии, вида оборудования, квалификации (разряда) рабочего и т.д., а также стоимость переналадки оборудования и возможность перехода к ускоренному режиму обработки деталей;

В-восьмых, приходится учитывать директивные сроки, к которым должны быть изготовлены те или иные детали, а также неодновременность их получения на обработку;

В-девятых, при расчете объемов незавершенного производства приходится учитывать различную стоимость деталей и изделий;

В-десятых, приходится учитывать возможность петель в маршруте детали, неодинаковость и пересечения маршрутов различных деталей. Все это значительно усложняет задачи упорядочения, их математические модели и методы решения.

Перейдем к рассмотрению некоторых алгоритмов решения задач упорядочения.

2. Алгоритм Джонсона для решения задачи упорядочения n × 2

Аналитические методы решения задач упорядочения разработаны лишь для очень простых случаев. Джонсон в 1963 г. предложил алгоритм построения оптимальной последовательности обработки для случая n деталей и двух станков с жесткими и одинаковыми маршрутами обработки, когда в качестве критерия оптимальности выбирается минимум общей продолжительности обработки, а начальные и конечные директивные сроки и таковы, что накладываемые этими сроками ограничения можно не учитывать. При постановке задачи также неявно предполагается, что имеется достаточная площадь для хранения межоперационных заделов и что стоимость хранения этих заделов либо одинакова для всех видов деталей, либо она незначительна. Пример такой задачи мы уже рассмотрели в предыдущем параграфе. Здесь мы попытаемся построить ее математическую модель и алгоритм решения.

2.1. Математическая модель задачи $n \times 2$

Пусть дана некоторая последовательность обработки деталей, имеющая номер s. Для простоты будем предполагать, что детали в этой последовательности пронумерованы так, что их номера в последовательности составляют натуральный ряд чисел $1, 2, \ldots, n$.

Выведем формулу для величины простоев $X = \sum^{n} x_i$. Анализируя график Ганта на рис.1, имеем:

$x_1 = a_1$,

$x_2 = \max(a_1 + a_2 - x_1 - b_1; 0)$ (это следует из графика Гранта и условия, что если $a_1 + a_2 \le x_1 + b_1$, то простой $x_2 = 0$),

$x_3 = \max(a_1 + a_2 + a_3 - x_1 - b_1 - x_2 - b_2; 0)$,

$x_4 = \max(a_1 + a_2 + a_3 + a_4 - x_1 - b_1 - x_2 - b_2 - x_3 - b_3; 0)$,

…………………………………………………………..

$$x_n = \max\left(\sum_{i=1}^{n} a_i - \sum_{i=1}^{n-1} x_i - \sum_{i=1}^{n-1} b_i ; 0\right)$$

Суммируя последовательно простои, получим, что

$$x_1 + x_2 = \max(a_1 + a_2 - b_1; x_1),$$
$$x_1 + x_2 + x_3 = \max(a_1 + a_2 + a_3 - b_1 - b_2; x_1 + x_2),$$
……………………………………………………

$$X = \sum_{i=1}^{n} x_i = \max\left(\sum_{i=1}^{n} a_i - \sum_{i=1}^{n-1} b_i ; \sum_{i=1}^{n-1} x_i\right).$$

Подставим в последнюю формулу, вместо $\sum_{i=1}^{n-1} x_i$, выражение

$$\sum_{i=1}^{n-1} x_i = \max\left(\sum_{i=1}^{n-1} a_i - \sum_{i=1}^{n-2} b_i ; \sum_{i=1}^{n-2} x_i\right),$$

и учитывая свойство $\max(A, \max(B, C)) = \max(A, B, C)$, получим

$$X = \sum_{i=1}^{n} x_i = \max\left(\sum_{i=1}^{n} a_i - \sum_{i=1}^{n-1} b_i ; \sum_{i=1}^{n-1} x_i\right) = \max\left(\sum_{i=1}^{n} a_i - \sum_{i=1}^{n-1} b_i ; \sum_{i=1}^{n-1} a_i - \sum_{i=1}^{n-2} b_i ; \sum_{i=1}^{n-2} x_i\right).$$

Подставляя теперь в это выражение, вместо $\sum_{i=1}^{n-2} x_i$,

$$\sum_{i=1}^{n-2} x_i = \max\left(\sum_{i=1}^{n-2} a_i - \sum_{i=1}^{n-3} b_i ; \sum_{i=1}^{n-3} x_i\right),$$

а затем, вместо $\sum_{i=1}^{n-3} x_i$, соответствующее выражение и т.д., получим цепочку равенств

$$X = \sum_{i=1}^{n} x_i = \max\left(\sum_{i=1}^{n} a_i - \sum_{i=1}^{n-1} b_i ; \sum_{i=1}^{n-1} x_i\right) = \max\left(\sum_{i=1}^{n} a_i - \sum_{i=1}^{n-1} b_i ; \sum_{i=1}^{n-1} a_i - \sum_{i=1}^{n-2} b_i ; \sum_{i=1}^{n-2} x_i\right) =$$
$$= \max\left(\sum_{i=1}^{n} a_i - \sum_{i=1}^{n-1} b_i ; \sum_{i=1}^{n-1} a_i - \sum_{i=1}^{n-2} b_i ; \sum_{i=1}^{n-2} a_i - \sum_{i=1}^{n-3} b_i ; \sum_{i=1}^{n-3} x_i ;\right) = \ldots =$$
$$= \max\left(\sum_{i=1}^{n} a_i - \sum_{i=1}^{n-1} b_i ; \sum_{i=1}^{n-1} a_i - \sum_{i=1}^{n-2} b_i ; \ldots ; \sum_{i=1}^{2} a_i - \sum_{i=1}^{1} b_i ; \sum_{i=1}^{1} x_i ;\right) =$$
$$= \max\left(\sum_{i=1}^{n} a_i - \sum_{i=1}^{n-1} b_i ; \sum_{i=1}^{n-1} a_i - \sum_{i=1}^{n-2} b_i ; \ldots ; \sum_{i=1}^{2} a_i - \sum_{i=1}^{1} b_i ; a_1\right).$$

Отсюда, учитывая, что $a_1 = \sum_{i=1}^{1} a_i - \sum_{i=1}^{0} b_i$, получим общую формулу для простоя:

$$X = \max_{1 \le u \le n} \left(\sum_{i=1}^{u} a_i - \sum_{i=1}^{u-1} b_i \right),$$

или

$$X = \max_{1 \le u \le n} K(u),$$

где $K(u) = \sum_{i=1}^{u} a_i - \sum_{i=1}^{u-1} b_i$.

Учитывая, что X зависит от порядка следования деталей, т.е. от номера s последовательности, а также то, что

$$\min_S T(S) = \min_S \left(\sum_{i=1}^{n} b_i + X(S) \right) = \sum_{i=1}^{n} b_i + \min_S X(S),$$

получаем следующую *математическую модель задачи* $n \times 2$:

$$\min_s X(S) = \min_s \left(\max_{1 \le u \le n} K(u) \right) = \min_s \max_{1 \le u \le n} \left(\sum_{i=1}^{u} a_i - \sum_{i=1}^{u-1} b_i \right).$$

Для построения алгоритма получения оптимальной последовательности $s^{опт}$ выведем *правило Джонсона*, указывающее, каким образом путем элементарных перестановок деталей (т.е., путем обмена местами двух соседних деталей) переходить к последовательности, для которой $X(s)$ становится меньше (точнее – не больше), чем $X(s)$ для предыдущей последовательности.

2.2. Вывод правила Джонсона

Пусть мы имеем две последовательности s' и s''':

$$s' = 1, 2, \ldots, j-1, j, j+1, j+2, \ldots, n,$$
$$s'' = 1, 2, \ldots, j-1, j+1, j, j+2, \ldots, n,$$

отличающиеся друг от друга лишь порядком следования деталей с номерами j и $j+1$. Найдем для этих последовательностей $K'(s)$ и $K''(s)$, и попытаемся сравнить $X(s')$ и $X(s'')$).

$$K'(u) = \begin{cases} \sum_{i=1}^{u} a_i - \sum_{i=1}^{u-1} b_i & - \text{ для всех } u \in [1; j-1], \\ \sum_{i=1}^{u-1} a_i + a_j - \sum_{i=1}^{u-1} b_i & - \text{ для } \quad u = j, \\ \sum_{i=1}^{u-2} a_i + a_j + a_{j+1} - \sum_{i=1}^{u-2} b_i - b_j & - \text{ для } u = j+1, \\ \sum_{i=1}^{u} a_i - \sum_{i=1}^{u-1} b_i & - \text{ для всех } u \in [j+2, n]. \end{cases}$$

$$K''(u) = \begin{cases} \sum_{i=1}^{u} a_i - \sum_{i=1}^{u-1} b_i & - \text{ для всех } u \in [1; j-1], \\ \sum_{i=1}^{u-1} a_i + a_{j+1} - \sum_{i=1}^{u-1} b_i & - \text{ для } u = j \\ \sum_{i=1}^{u-2} a_i + a_{j+1} + a_j - \sum_{i=1}^{u-2} b_i - b_{j+1} & - \text{ для } u = j+1, \\ \sum_{i=1}^{u} a_i - \sum_{i=1}^{u-1} b_i & - \text{ для всех } u \in [j+2, n]. \end{cases}$$

Сравнивая выражения $K'(s)$ и $K''(s)$, видим, что для $u \in [1, j-1] \cup [j+2; n]$ имеем $K'(u) = K''(u)$. Поэтому, если

$$\max(K'(j); K'(j+1)) = \max(K''(j); K''(j+1)), \text{ то } X(S') = X''(S).$$

Если же например,

$$\max(K''(j); K''(j+1)) < \max(K'(j); K'(j+1)), \text{ то } X(S'') \le X'(S).$$

Появление, наряду со знаком "<" еще и знака "=" здесь объясняется тем, что $\max_u K(u)$ может достигаться не только при $u = j$ и $u = j + 1$, но и при прочих значениях u, а в таких случаях, как уже видели выше, $K'(u) = K''(u)$.

Итак, мы получили следующие результаты:

а) $[\max(K'(j); K'(j+1)) = \max(K''(j); K''(j+1)) \to X(S') = X(S'')]$;
б) $[\max(K'(j); K'(j+1)) < \max(K''(j); K''(j+1)) \to X(S') \le X(S'')]$;
в) $[\max(K'(j); K'(j+1)) > \max(K''(j); K''(j+1)) \to X(S') \ge X(S'')]$;

Соотношения между $\max(K'(j); K'(j+1))$ и $\max(K''(j); K''(j+1))$ не изменится, если из этих выражений вычтем одно и то же число $\sum_{i=1}^{j+1} a_i - \sum_{i=1}^{j-1} b_i$. Получим:

а) $[\max(K'(j); K'(j+1)) = \max(K''(j); K''(j+1))] \leftrightarrow$

$$\leftrightarrow [\max(K'(j) - (\sum_{i=1}^{j+1} a_i - \sum_{i=1}^{j-1} b_i); K'(j+1) - (\sum_{i=1}^{j+1} a_i - \sum_{i=1}^{j-1} b_i)) =$$

$$= [\max(K''(j) - (\sum_{i=1}^{j+1} a_i - \sum_{i=1}^{j-1} b_i); K''(j+1) - (\sum_{i=1}^{j+1} a_i - \sum_{i=1}^{j-1} b_i))] \leftrightarrow$$

$$\leftrightarrow [\max(-a_{j+1}; -b_j) = \max(-a_j; -b_{j+1})] \leftrightarrow [\min(a_{j+1}; b_j) = \min(a_j; b_{j+1})];$$

б) $[\max(K'(j); K'(j+1)) < \max(K''(j); K''(j+1))] \leftrightarrow$

$$\leftrightarrow [\max(-a_{j+1}; -b_j) < \max(-a_j; -b_{j+1})] \leftrightarrow [\min(a_{j+1}; b_j) > \min(a_j; b_{j+1})];$$

в) $[\max(K'(j); K'(j+1)) > \max(K''(j); K''(j+1))] \leftrightarrow [\min(a_{j+1}; b_j) < \min(a_j; b_{j+1})]$.

Следовательно:

а) $[\min(a_{j+1}; b_j) = \min(a_j; b_{j+1})] \to [X(S') = X(S'')]$,
б) $[\min(a_{j+1}; b_j) > \min(a_j; b_{j+1})] \to [X(S') \le X(S'')]$,
в) $[\min(a_{j+1}; b_j) < \min(a_j; b_{j+1})] \to [X(S') \ge X(S'')]$.

Получаем, что последовательность S' не менее выгодна, чем S'', если среди величин a_j, a_{j+1}, b_j, b_{j+1}, самыми минимальными будут или a_j или b_{j+1}, если же самыми минимальными будут или a_{j+1} или b_j, то не менее выгодной будет последовательность S''. Во всех остальных случаях последовательности S' и S'' будут эквивалентными.

Таким образом, если мы в какой-либо последовательности S переставим две рядом стоящие детали, например, детали с номерами j и j+1, то последовательность может улучшиться (т.е. может уменьшиться значение $X(S)$), если в результате перестановки впереди окажется та деталь, время обработки которой на первом станке будет самой малой из величин a_j, a_{j+1}, b_j, b_{j+1}, или окажется сзади та деталь, время обработки которой на втором станке будет самой малой из тех же величин a_j, a_{j+1}, b_j, b_{j+1}.

Действительно, если $\min(a_j, a_{j+1}, b_j, b_{j+1}) = a_{j+1}$, то более выгодной может оказаться последовательность S'', в которой деталь $j + 1$ стоит перед деталью j.

Если же $\min(a_j, a_{j+1}, b_j, b_{j+1}) = a_j$, то более выгодной может оказаться последовательность S', в которой деталь j стоит перед деталью $j + 1$.

Аналогично получаем, что при $\min(a_j, a_{j+1}, b_j, b_{j+1}) = b_{j+1}$ более выгодной может оказаться последовательность S', в которой деталь $j + 1$ стоит после детали j, а при $\min(a_j, a_{j+1}, b_j, b_{j+1}) = b_j$ более выгодной может оказаться последовательность S'', в которой деталь j стоит после детали $j + 1$.

Из сказанного выше получаем правило Джонсона для улучшения последовательности деталей:

а) если для двух соседних деталей j и $j + 1$ последовательности S' получаем, что $\min(a_j, a_{j+1}, b_j, b_{j+1}) = a_{j+1}$, то имеет смысл деталь с номером $j + 1$ поставить перед деталью с номером j, т.е. перейти к последовательности S'';

б) если $\min(a_j, a_{j+1}, b_j, b_{j+1}) = b_j$, то имеет смысл деталь с номером j поставить после детали с номером $j + 1$, т.е. перейти к последовательности S'';

в) во всех остальных случаях последовательность S' изменять не целесообразно, так как выигрыша не получим, а проигрыши получить можно.

Применяя правило Джонсона ко всем парам рядом стоящих деталей последовательности до тех пор, пока дальнейшие перестановки деталей, сулящие возможные уменьшения величин $X(S)$, перестанут быть возможными, мы получим оптимальную последовательность $S^{\text{опт}}$. Для этой последовательности будет характерно, что:

если $\min(a_1,a_2\ldots,a_n, b_1,b_2,\ldots,b_n) = a_k$ то деталь с номером k будет стоять на первом месте;

если же $\min(a_1,\ldots,a_n, b_1,\ldots,b_n) = b_l$, то деталь с номером l будет стоять на самом последнем месте.

Учитывая это обстоятельство, можно построить алгоритм Джонсона получения оптимальной последовательности для задачи $n \times 2$ более быстро, чем путем последовательных парных перестановок по правилу Джонсона.

2.3. Алгоритм Джонсона

1. Найти наименьший элемент в таблице значений $\{a_i, b_i\}$.

2. Если этот наименьший элемент принадлежит столбцу значений a_i, то деталь с соответствующим номером i поставить на первое свободное место последовательности (в первый раз это будет первое место, во второй – второе и т.д.); если же наименьший элемент принадлежит столбцу значений b_i, то деталь

с соответствующим номером i поставить на последнее свободное место последовательности (в первый раз это будет n-е место, во второй – $(n - 1)$ и т.д.); если одновременно найдется несколько одинаковых наименьших элементов, то среди них можно выбирать любой, принять за наименьший и поступать так, как описано в начале пункта 2 алгоритма.
3. Из таблицы значений $\{a_i, b_i\}$ вычеркнуть строку, соответствующую выбранному наименьшему элементу, и проверить, остались ли еще не вычеркнутые строки.
4. Если невычеркнутые строки еще остались, то рассматривать их как новую таблицу значений $\{a_i, b_i\}$ и перейти к пункту 1 алгоритма; если вычеркнуты все строки таблицы, то это означает, что алгоритм свою работу закончил.

Применим алгоритм Джонсона к таблице примера, рассмотренного в первом параграфе данной главы.

1. Просматривая таблицу значений $\{a_i, b_i\}$ находим, что минимальным является b_1=1.
2. Это означает, что деталь с номером $i^* = 1$ в оптимальной последовательности должна быть последней, т.е. иметь номер $i^{\text{нов.}} = n = 5$.
3. Вычеркиваем из таблицы первую строку.
4. Так как еще остались невычеркнутые строки, то возвращаемся к первому пункту алгоритма Джонсона.
5. Находим, что минимальным элементом остаточной таблицы является $a_5 = 2$.
6. Ставим деталь с номером $i^* = 5$ на первое место оптимальной последовательности, т.е. присваиваем номеру i^* значение 1.
7. Вычеркиваем из таблицы пятую строку.
8. Так как еще остались невычеркнутые строки, то опять возвращаемся к первому пункту алгоритма.
9. Находим, что минимальными являются $a_4 = b_2 = 4$.
10. Выбираем деталь, например, с номером $i^* = 4$ и помещаем ее в первое свободное место оптимальной последовательности, т.е. присваиваем номеру $i^{\text{нов}}$ значение 2.
11. Вычеркиваем четвертую строку.
12. Возвращаемся к началу.
13. Находим, что $(a_i, b_i) = b_2 = 4$.
14. Присваиваем детали с номером $i^* = 2$ номер $i^{\text{нов}} = n - 1 = 4$.
15. Вычеркиваем вторую строку.
16. Возвращаемся к таблице, содержащей еще одну строку.

Конечно, для оставшейся невычеркнутой детали осталось всего одно свободное место в оптимальной последовательности с номером $i^{\text{нов}} = 3$, поэтому решение уже получено – оптимальная последовательность в прежних номерах деталей будет иметь вид 5,4,3,2,1. Это же решение мы получили бы, если бы продолжили алгоритм Джонсона до конца (что сделал бы компьютер, работая по программе, реализующей алгоритм Джонсона). Действительно, проверив оставшуюся строку, компьютер обнаружил бы, что $\min(a_i, b_i) = a_3 = 6$, и поместила бы деталь с номером $i^* = 3$ в первое свободное место, т.е. присвоила бы ей номер 3. Вычеркнув третью строку таблицы, компьютер обнаружил бы, что невычеркнутых строк больше не осталось, и вышла бы на конец алгоритма – печать результата и останов работы программы.

У читателя может возникнуть сомнение, правильно ли поступаем, когда, обнаружив два (или больше) одинаковых минимальных элемента, принадлежащих одному и тому же столбцу (например, столбцу А), мы выбираем в качестве первого любой из этих элементов, не обращая внимания на значение соответствующего ему элемента в другом столбце (например, в столбце B).

На это сомнение можно ответить, что, согласно правилу Джонсона, в этом случае порядок следования (и, следовательно, порядок выбора деталей) безразличен, так как если $\min(a_{j+1}, b_j) = \min(a_j, b_{j+1}) = a_{j+1} = a_j$, то $X(S') = X(S'')$ независимо от того, какое соотношение между b_j и b_{j+1}. Аналогично получаем, что при $\min(a_{j+1}, b_j) = \min(a_j, b_{j+1}) = b_j = b_{j+1}$ будет иметь место $X(S') = X(S'')$ независимо от значений a_j и a_{j+1}.

Проиллюстрируем сказанное на примере задачи $n \times 2$, заданной при помощи таблицы 3.

Таблица 3

i	a_i	b_i
1	6	6
2	8	6
3	8	9
4	4	2
5	9	4

Таблица 4

i	a_i	b_i
3	8	9
1	6	6
2	8	6
5	9	4
4	4	2

Таблица 5

i	a_i	b_i
3	8	9
2	8	6
1	6	6
5	9	4
4	4	2

Таблица 6

i	a_i	b_i
1	6	6
3	8	9
2	8	6
5	9	4
4	4	2

Пользуясь алгоритмом Джонсона, мы можем получить оптимальные последовательности обработки, которым соответствуют табл. 4, 5 и 6.

Найдем $X = \max\limits_{1 \le u \le 5} K(u)$ для этих последовательностей

Для последовательности, соответствующей табл. 4:

$$X = \max(8;\ 5;\ 7;\ 10; 10) = 10.$$

Для последовательности, соответствующей табл. 5:

$$X = \max(8;\ 7;\ 7;\ 10; 10) = 10.$$

Для последовательности, соответствующей табл. 6:

$$X = \max(6;\ 8;\ 7;\ 10; 10) = 10.$$

Сумма простоев во всех трех последовательностях получилась одинаковой ($X = 10$). Различие лишь в распределении простоев: в первом случае $x_1 = 8$, $x_2 = 2$; во втором $x_1 = 8$, $x_2 = 2$, в третьем $x_1 = 6$, $x_2 = 2$, $x_4 = 2$.

Это может быть учтено при планировании заполнения простоев фоновой работой.

Алгоритм Джонсона для задачи $n \times 2$ может быть модифицирован и сведен к следующему [4]:

1. Находим все детали, для которых $a_i \le b_i$ и упорядочиваем их в порядке возрастания a_i.
2. Оставшиеся детали, для которых $a_i > b_i$, упорядочиваем в порядке убывания b_i.
3. Подписываем список деталей второй группы под списком деталей первой группы, т.е. обрабатываем сначала детали первой группы в порядке возрастания a_i, затем детали второй группы в порядке убывания b_i.

Упражнение 2.1. Покажите на примере задачи, описанной в табл.1, что модифицированный алгоритм Джонсона дает тот же результат, что и алгоритм, предложенный Джонсоном в работе [1].

Упражнение 2.2. Постройте для таблиц 4, 5 и 6 графики Ганта и убедитесь, что найденные при помощи $K(u)$ значения X и x_i соответствуют действительности. Укажите алгоритм вычисления x_i по найденным значениям $K(1)$, $K(2)$, $K(3)$, …$K(n)$.

Упражнение 2.3. Пользуясь формулой $X = \max\limits_{1<u<n} K(u)$, вычислите величины X для последовательностей 1-2-3-4-5 и 5-4-3-2-1 обработки деталей, описанных в табл.1.

Убедитесь, что в первом случае X = 35, а во втором X = 4. При помощи найденных значений $K(u)$ вычислите отдельные простои x_i станка B.

Упражнение 2.4. Найдите оптимальную последовательность обработки и соответствующие значения X и T для деталей, описанных в табл. 7, а также все оптимальные последовательности для табл. 8.

Таблица 7

i	a_i	b_i
1	3	7
2	5	8
3	2	9
4	4	1
5	8	11
6	7	6
7	4	10

Таблица 8

i	a_i	b_i
1	8	5
2	6	9
3	6	5
4	6	7
5	2	2
6	3	2

3. Алгоритм Джонсона для решения задачи упорядочения n× 3

Рассмотрим задачу выбора оптимальной последовательности s обработки n деталей на трех станках в случае, когда все детали имеют жесткий и одинаковый маршрут и выполняются все те же специальные условия, которые были оговорены при постановке задачи упорядочения $n \times 2$.

Обозначим станки через A, B, C, нормы времени обработки детали i соответственно через a_i, b_i и c_i, а простои станка B через x_i, станка C – через y_i. График Ганта в этом случае будет иметь следующий вид (рис.2):

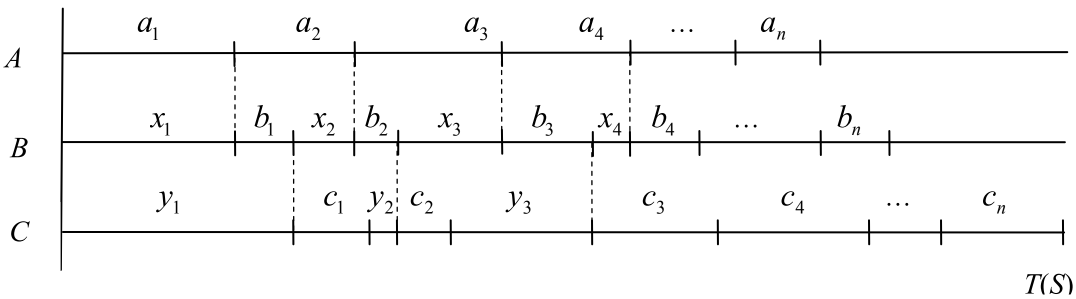


Рис.2. График Ганта для случая $n \times 3$

Согласно предыдущему параграфу для двух станков имеем:

$$X = \sum_{i=1}^{n} x_i = \max_{1 \le u \le n} \left(\sum_{i=1}^{u} a_i - \sum_{i=1}^{u-1} b_i \right)$$

Найдем $Y = \sum_{i=1}^{n} y_i$. Из рис. 2 имеем

$y_1 = x_1 + b_1$,

$y_2 = \max(x_1 + b_1 + x_2 + b_2 - y_1 - c_1;0)$,

$y_3 = \max(x_1 + b_1 + x_2 + b_2 + x_3 + b_3 - y_1 - c_1 - y_2 - c_2;0)$,

…………………………………………………………

…………………………………………………………

$$y_n = \max\left(\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} b_i - \sum_{i=1}^{n-1} y_i - \sum_{i=1}^{n-1} c_i ; 0\right).$$

Прибавляя к y_1 последовательно $y_2, y_3, \ldots$ и учитывая свойство максимума, получим суммы

$$y_1 + y_2 = \max(x_1 + b_1 + x_2 + b_2 - c_1; y_1),$$
$$y_1 + y_2 + y_3 = \max(x_1 + b_1 + x_2 + b_2 + x_3 + b_3 - c_1 - c_2; y_1 + y_2),$$

………………………………………………………………

………………………………………………………………

$$Y = \sum_{i=1}^{n} y_i = \max\left(\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} b_i - \sum_{i=1}^{n-1} c_i ; \sum_{i=1}^{n-1} y_i\right).$$

Ранее было показано, что

$$\sum_{i=1}^{n} x_i = \max_{1 \le u \le n}\left(\sum_{i=1}^{u} a_i - \sum_{i=1}^{u-1} b_i\right) = \max_{1 \le u \le n} K(u).$$

Учитывая это и вводя обозначение $H(v) = \sum_{i=1}^{v} b_i - \sum_{i=1}^{v-1} c_i$, получим

$$\sum_{i=1}^{n} y_i = \max\left(\max_{1 \le u \le n} K(u) + H(n); \sum_{i=1}^{n-1} y_i\right).$$

Но $\sum_{i=1}^{n-1} y_i = \max\left(\max_{1 \le u \le n-1} K(u) + H(n-1); \sum_{i=1}^{n-2} y_i\right)$ поэтому, развертывая рекуррентно формулу для $Y = \sum_{i=1}^{n} y_i$ путем подстановки формул для $\sum_{i=1}^{n-1} y_i$, $\sum_{i=1}^{n-2} y_i$ и т.д., получим:

$$Y = \sum_{i=1}^{n} y_i = \max(\max_{1 \le u \le n} K(u) + H(n); \max_{1 \le u \le n-1} K(u) + H(n-1); \ldots$$
$$\ldots; \max \max_{1 \le u \le 2} K(u) + H(2); \max_{1 \le u \le 1} K(u) + H(1)),$$

где $\max_{1 \le u \le 1} K(u) + H(1) = K(1) + H(1) = a_1 + b_1$.

Развернутую формулу для Y можно записать более компактно:

$$Y = \max_{1 \le v \le n}(\max_{1 \le u \le v} K(u) + H(v)) = \max_{1 \le u \le v \le n}(K(u) + H(v)).$$

Выбрав в качестве критерия оптимальности минимум времени окончания обработки T, получим следующую математическую модель задачи упорядочения $n \times 3$:

$$\min_s T(s) = \min_s\left(\sum_{i=1}^{n} c_i + Y(s)\right) = \sum_{i=1}^{n} c_i + \min_s Y(s).$$

Таким образом, мы приходим к задаче отыскания $S^{опт}$ – номера последовательности, обращающей Y(s) в минимум:

Для решения этой задачи Джонсоном был найден алгоритм в случае, когда выполняется условие:

$$(\min_i a_i \ge \max_j b_j) \vee (\min_k c_k \ge \max_j b_j)$$

Рассмотрим два следствия из этого условия.

1. Если $\min_i a_i \ge \max_j b_j$, то $a_p - b_r \ge 0$ при любых p и r. Следовательно, в этом случае $K(u)$ – монотонно возрастающая (точнее, неубывающая) функция от u:

$$K(1) \le K(2) \le K(3) \le \ldots \le K(j) \le K(j+1) \le \ldots \le K(n).$$

Действительно, при переходе от $K(j)$ к $K(j+1)$ мы прибавляем к $K(j)$ слагаемое $a_{j+1} - b_j \ge 0$, поэтому $K(j+1) \ge K(j)$ при любых j от 1 до $n-1$.

Используя условие неубывания функции $K(u)$, получаем, что функция $K(u)$ при условии $1 \le u \le v$ максимальное значение будет точно иметь в значении $K(v)$, поэтому

$$Y = \max_{1\le u\le v\le n}(K(u) + H(v)) = \max_{1\le v\le n}(K(v) + H(v)).$$

2. Если $\min_k c_k \ge \max_j b_j$, то $b_p - c_r \le 0$ при любых p и r. Отсюда следует, что в этом случае $H(v)$- монотонно убывающая (точнее, не возрастающая) функция:

$$H(1) \ge H(2) \ge H(3) \ge \ldots \ge H(j) \ge H(j+1) \ge \ldots \ge H(n).$$

Действительно, при переходе от $H(j)$ к $H(j+1)$ мы прибавляем к $H(j)$ слагаемое $b_{j+1} - c_j \le 0$, поэтому $H(j) \ge H(j+1)$ при любых j от 1 до $n-1$.

Используя условие невозрастания функции $H(v)$, получаем, что функция $H(v)$ при условии $1 \le u \le v$ максимальное значение будет точно иметь в значении $H(u)$, поэтому

$$Y = \max_{1\le u\le v\le n}(K(u) + H(v)) = \max_{1\le u\le n}(K(u) + H(u)).$$

Таким образом, как в случае $\min_i a_i \ge \max_j b_j$, так и в случае $\min_k c_k \ge \max_j b_j$ для Y получаем одну и ту же общую формулу простоев для третьего станка

$$Y = \max_{1\le u\le v\le n}(K(u) + H(v)).$$

В случае выполнения условия $(\min_i a_i \ge \max_j b_j) \vee (\min_k c_k \ge \max_j b_j)$ она превращается в формулу

$$Y = \max_{1\le u\le n}(K(u) + H(u)).$$

Таким образом, проблема поиска решения задачи $n \times 3$ в проблему поиска номера s последовательности, для которой

$$Y(s) = \min_s(\max_{1\le u\le n}(K(u) + H(u))).$$

Для нахождения правила построения такой оптимальной последовательности s обработки деталей рассмотрим две последовательности s' и s'':

$$s' = 1, 2, \ldots, j-1, j, j+1, j+2, \ldots, n,$$
$$s'' = 1, 2, \ldots, j-1, j+1, j, j+2, \ldots, n$$

отличающиеся друг от друга лишь перестановкой порядка обработки деталей j и $j+1$. Очевидно, что при $u \le j-1$ и при $u \ge j+2$ $K'(u) = K''(u)$ и $H'(u) = H''(u)$. Следовательно, $K'(u)$ и $K''(u)$, а также $H'(u)$ и $H''(u)$ могут отличаться лишь при $u = j$ и $u = j+1$. Учитывая сказанное и рассуждая аналогично, как и при построении правила Джонсона для задачи $n \times 2$, получаем:

а) $$[\max\,(K'(j) + H'(j); K'(j+1) + H'(j+1)) = \\ = \max\,(K''(j) + H''(j); K''(j+1) + H''(j+1))] \Rightarrow [Y(s') = Y(s'')];$$

б) $$[\max\,(K'(j) + H'(j); K'(j+1) + H'(j+1)) > \\ > \max\,(K''(j) + H''(j); K''(j+1) + H''(j+1))] \Rightarrow [Y(s') \ge Y(s'')];$$

в) $$[\max\,(K'(j) + H'(j); K'(j+1) + H'(j+1)) < \\ < \max\,(K''(j) + H''(j); K''(j+1) + H''(j+1))] \Rightarrow [Y(s') \le Y(s'')];$$

Соотношения (равенство и неравенство) между

$$\max\,(K'(j) + H'(j); K'(j+1) + H'(j+1)) \text{ и } \max\,(K''(j) + H''(j); K''(j+1) + H''(j+1))$$

в выражениях (а),(б),(в) не изменятся, если мы от каждой из сумм внутри $\max(\cdot)$ отнимем одно и то же число

$$\sum_{i=1}^{j+1} a_i - \sum_{i=1}^{j} b_i + \sum_{i=1}^{j+1} b_i - \sum_{i=1}^{j} c_i\ .$$

Тогда для каждого из выражений (а), (б), (в) получим соответственно следующие цепочки тождеств:

а) $$[\max\,(K'(j) + H'(j); K'(j+1) + H'(j+1)) = \\ = \max\,(K''(j) + H''(j); K''(j+1) + H''(j+1))] \Leftrightarrow; \\ \Leftrightarrow \max(-a_{j+1} - b_{j+1}; -b_j - c_j) = \max(-a_j - b_j; -b_{j+1} - c_{j+1}) \Leftrightarrow \\ \Leftrightarrow \min(a_{j+1} + b_{j+1}; b_j + c_j) = \min(a_j + b_j; b_{j+1} + c_{j+1});$$

б) $$[\max\,(K'(j) + H'(j); K'(j+1) + H'(j+1)) > \\ > \max\,(K''(j) + H''(j); K''(j+1) + H''(j+1))] \Leftrightarrow; \\ \Leftrightarrow \max(-a_{j+1} - b_{j+1}; -b_j - c_j) > \max(-a_j - b_j; -b_{j+1} - c_{j+1}) \Leftrightarrow \\ \Leftrightarrow \min(a_{j+1} + b_{j+1}; b_j + c_j) > \min(a_j + b_j; b_{j+1} + c_{j+1});$$

в) $$[\max\,(K'(j) + H'(j); K'(j+1) + H'(j+1)) < \\ < \max\,(K''(j) + H''(j); K''(j+1) + H''(j+1))] \Leftrightarrow; \\ \Leftrightarrow \max(-a_{j+1} - b_{j+1}; -b_j - c_j) < \max(-a_j - b_j; -b_{j+1} - c_{j+1}) \Leftrightarrow$$

$$\Leftrightarrow \min(a_{j+1}+b_{j+1}; b_j+c_j) < \min(a_j+b_j; b_{j+1}+c_{j+1});$$

Из них следует:

а) $[\min(a_{j+1}+b_{j+1}; b_j+c_j) = \min(a_j+b_j; b_{j+1}+c_{j+1})] \Rightarrow [Y(s') = Y(s'')]$;
б) $[\min(a_{j+1}+b_{j+1}; b_j+c_j) < \min(a_j+b_j; b_{j+1}+c_{j+1})] \Rightarrow [Y(s') \ge Y(s'')]$;
в) $[\min(a_{j+1}+b_{j+1}; b_j+c_j) > \min(a_j+b_j; b_{j+1}+c_{j+1})] \Rightarrow [Y(s') \le Y(s'')]$;

Обозначая $d_j = a_j + b_j$, $e_j = b_j + c_j$, из последних выражений получим условия:

а) $[\min(d_{j+1}; e_j) = \min(d_j; e_{j+1})] \Rightarrow [Y(s') = Y(s'')]$;

б) $[\min(d_{j+1}; e_j) < \min(d_j; e_{j+1})] \Rightarrow [Y(s') \ge Y(s'')]$;

в) $[\min(d_{j+1}; e_j) > \min(d_j; e_{j+1})] \Rightarrow [Y(s') \le Y(s'')]$.

Сравнивая эти условия с аналогичными условиями, полученными для задачи $n\times 2$, обнаруживаем, что они подобны (изоморфны) и функция

$$\widetilde{K}(u) = K(u) + H(u) = \sum_{i=1}^{u} a_i - \sum_{i=1}^{u-1} b_i + \sum_{i=1}^{u} b_i - \sum_{i=1}^{u-1} c_i = \left(\sum_{i=1}^{u} a_i + \sum_{i=1}^{u-1} b_i\right) - \left(\sum_{i=1}^{u} b_i + \sum_{i=1}^{u-1} c_i\right) = \sum_{i=1}^{u} d_i - \sum_{i=1}^{u-1} e_i$$

подобна функции $K(u) = \sum_{i=1}^{u} a_i - \sum_{i=1}^{u-1} b_i$. Это по сути является доказательством возможности сведения задачи n×3 к задаче n×2. Отсюда следует, что решение задачи $n\times 3$ при выполнении условия $(\min_i a_i \ge \max_j b_j) \vee (\min_k c_k \ge \max_j b_j)$ сводится к решению задачи $n\times 2$, если только интерпретировать:

- $a_i + b_i = d_i$ - как норму времени обработки детали i на некотором станке D_i,
- $b_i + c_i = e_i$ как норму времени обработки детали i на некотором станке E_i.

Отсюда получаем следующий алгоритм решения задачи упорядочения при $n\times 3$ в случае выполнения условия

$$(\min_i a_i \ge \max_j b_j) \vee (\min_k c_k \ge \max_j b_j).$$

Алгоритм

1. Производим построчное сложение элементов первого и второго столбцов таблицы: $d_i = a_i + b_i$.
2. Производим сложение второго и третьего столбцов таблицы: $e_i = b_i + c_i$.
3. К новой таблице $n\times 2$ применяем алгоритм Джонсона, используемый для задачи упорядочения $n\times 2$.

Пример. Рассмотрим пример решения задачи $n\times 3$. Пусть дана таблица 9:

Таблица 9

i	a_i	b_i	c_i
1	6	3	7
2	8	5	8
3	5	6	11
4	4	2	6
5	9	4	10

Для этой таблицы условие $(\min_i a_i \geq \max_j b_j) \vee (\min_k c_k \geq \max_j b_j)$ выполняется, так как $\min_k c_k = 6$, а $\max_j b_j = 6$. Следовательно, для нахождения оптимальной последовательности s, задачу упорядочения $n \times 3$ можно преобразовать к задаче $n \times 2$ и затем применить алгоритм Джонсона.

Строим таблицу i, d_i, e_i (см.табл.10).

i	d_i	e_i
1	9	10
2	13	13
3	11	17
4	6	8
5	13	14

Применяя к таблице i, d_i, e_i алгоритм Джонсона для случая $n \times 2$, получаем оптимальную последовательность обработки: 4-1-3-5-2.

Определим величины простоев Y и X для этой последовательности, представленной в таблице 11.

Таблица 11

i	a_i	b_i	c_i
4	4	2	6
1	6	3	7
3	5	6	11
5	9	4	10
2	8	5	8

$$Y = \max_{1 \leq u \leq 5}(K(u) + H(u)).$$

$K(1) = a_4 = 4$; $H(1) = b_4 = 2$;

$K(2) = 4 + 6 - 2 = 8$; $H(2) = 2 + 3 - 6 = -1$;

$K(3) = 4 + 6 + 5 - 2 - 3 = 15 - 5 = 10$; $H(3) = 11 - 13 = -2$;

$K(4) = 15 + 9 - 5 - 6 = 24 - 11 = 13$; $H(4) = 15 - 24 = -9$;

$K(5) = 32 - 15 = 17$; $H(5) = 20 - 34 = -14$;

$$Y = \max(4 + 2 ; 8 + (-1) ; 10 + (-2) ; 13 + (-9) ; 17 + (-14)) = \max(6 ; 7 ; 8 ; 4 ; 3) = 8 ;$$

$$X = \max_{1 \leq u \leq 5} K(u) = \max(4 ; 8 ; 10 ; 13 ; 17) = 17 .$$

Если же взять исходную последовательность обработки (см.табл.9), то получим:

$K(1) = 6$; $H(1) = 3$; $K(1) + H(1) = 9$;

$K(2) = 14 - 3 = 11$; $H(2) = 8 - 7 = 1$; $K(2) + H(2) = 12$;

$K(3) = 19 - 8 = 11$; $H(3) = 14 - 15 = -1$; $K(3) + H(3) = 10$;

$K(4) = 23 - 14 = 9$; $H(4) = 16 - 26 = -10$; $K(4) + H(4) = -1$;

$K(5) = 32 - 16 = 16$; $H(5) = 20 - 32 = -12$; $K(5) + H(5) = 4$.

$$Y = \max_{1 \le u \le 5}(K(u) + H(u)) = \max(9\,;12\,;10\,;-1\,;4) = 12\,;$$

$$X = \max_{1 \le u \le 5} K(u) = \max(6\,;11\,;11\,;9\,;16) = 16$$

Упражнение 3.1. Проверьте полученные результаты, т.е. найденные значения X и Y при помощи графиков Ганта.

Упражнение 3.2. Как изменилось бы решение задачи, если в качестве критерия оптимальности выбирался минимум общего простоя станков, т.е. $\min_s(X(s) + Y(s))$?

Упражнение 3.3. Найдите последовательность s, минимизирующую Y, а также соответствующие значения и X, Y и T для случая, соответствующего таблице, полученной из таблицы 9 путем перемены местами столбцов a_i и c_i.

Упражнение 3.4. Применим ли алгоритм Джонсона $n \times 3$ для таблицы, полученной из таблицы 9 путем перемены местами столбцов a_i и b_i? Почему? Как найти оптимальную последовательность обработки деталей в этом случае?

Для случая, когда условие

$$(\min_i a_i \ge \max_j b_j) \vee (\min_k c_k \ge \max_j b_j)$$

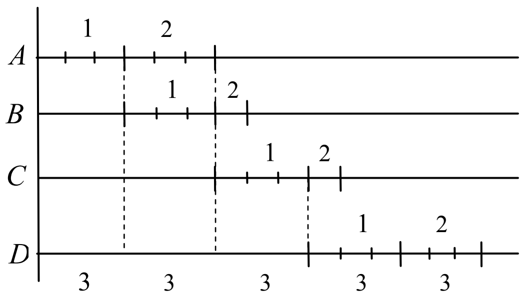
не выполняется, общего алгоритма для получения оптимальной последовательности еще не найдено. Не найден общий алгоритм и для решения более общей задачи $n \times m$. Однако и для общей задачи Джонсона $n \times m$ остается справедливым утверждение: определяя оптимальные последовательности (в смысле минимального времени обработки всех деталей), нужно минимизировать суммарное время простоев последнего станка.

При $m \ge 4$ возникают дополнительные сложности в связи с возможностью изменения последовательности обработки деталей. Джонсоном было доказано, что в оптимальной последовательности обработки деталей порядок их следования на первой и второй операциях должен быть одинаков, так же как должен быть одинаков порядок их следования на последней и предпоследней операциях (см. леммы 1 и 3 в [1]). Поэтому при $m \le 3$ количество вариантов последовательностей, среди которых следует искать оптимальные, равно $n!$. Что же касается порядка следования деталей на "внутренних" операциях при $m \ge 4$, то он в оптимальной последовательности может быть и различным. Так, например, при $m = 4$ порядок следования деталей на второй и третьей операциях в оптимальной последовательности не обязательно будет совпадать. Это можно показать на нижеследующем примере задачи 2×4 (см.табл. 12).

Таблица 12

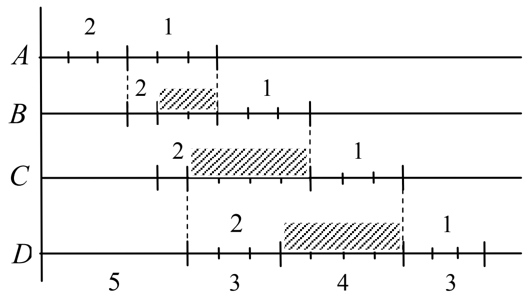
i	a_i	b_i	c_i	d_i
1	3	3	3	3
2	3	1	1	3

Если обрабатывать детали в последовательности 1-2 или 2-1, то, как видно из рис. 3«а» и 3«б», общее время обработки T = 15 ед. Если же на первой и второй операциях (на станках A и B) обрабатывать детали в последовательности 1-2, а на третьей и четвертой (на станках C и D) – в последовательности 2-1, то общее время обработки T = 14 ед. (см.рис. 3 «в»).



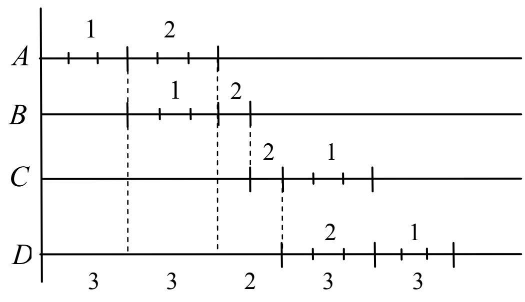
T=15

Рис. 3 «а»



T=15

Рис. 3 «б»



T=14

Рис. 3 «в»

4. Вопросы по данной теме

1. Какие показатели производственного процесса можно выделить при решении задачи упорядочения?
2. В чем состоит задача упорядочения, исследованная Джонсоном (задача Джонсона)?
3. Что является целевой функцией (оптимизируемым показателем) и оптимизирующей переменной в задаче Джонсона?
4. Какие ограничения имеются в задаче Джонсона и на что они влияют?
5. Откуда выводятся правила Джонсона?
6. Каким образом получается алгоритм Джонсона?
7. В чем суть алгоритма Джонсона?
8. Из каких величин состоит совокупная длительность производственного цикла (время окончания обработки последней детали на последнем станке)?
9. Как получить математическую модель задачи Джонсона для 2-х станков?
10. Как получить математическую модель задачи Джонсона для 3-х станков?
11. Какие имеются способы определения простоя 2-го станка в задаче Джонсона?
12. Какие имеются способы определения простоя 3-го станка в задаче Джонсона?
13. Какой способ нахождения решения для задачи Джонсона можно предложить помимо использования алгоритма Джонсона?
14. Сколько вариантов последовательностей запуска может быть в задаче Джонсона для 2-х и 3-х станков?
15. Каким образом решается задача Джонсона для 3-х станков?
16. На что влияет условие ($\min a_i \geq \max b_j$) или ($\min c_k \geq \max b_j$) в задаче Джонсона для 3-х станков?
17. За счет чего происходит сведение задачи упорядочения с тремя станками к задаче упорядочения с двумя станками?
18. Объясните, почему формула вычисления простоя на третьем станке в общем виде
$$Y = \max_{1 \leq u \leq v \leq n} (K(u) + H(v))$$
19. преобразуется к формуле
$$Y = \max_{1 \leq u \leq n} (K(u) + H(u)) \text{ ?}$$
20. Почему в общем случае трудоемкость (время) обработки детали не может быть равна нулю?

5. Список литературы

1. Шукис А.А. Задачи упорядочения: исследование операций. Учебное пособие. Барнаул: АПИ. 1982. – 68 с.
2. Астахова А.В. Методические указания "Моделирование производственных процессов на механообрабатывающих участках".- Барнаул: АПИ, 1988. – 22 с.
3. Дегтярев Ю.И. Исследование операций. М: Радио и связь, 1986. – 430 с.
4. Первозванский А.А. Математические модели в управлении производством. – М.: Наука, 1975. – 616 с.
5. Ржевский С.В. Исследование операций: Учебное пособие. – СПб: Изд-во «Лань», 2013, 480 с.
6. Есипов Б.А. Методы исследование операций: Учебное пособие. – СПб: Изд-во «Лань», 2013, 304 с.
7. Горлач Б.А. Исследование операций: Учебное пособие. – СПб: Изд-во «Лань», 2013, 448 с.