Введение в теорию расписаний

Теорию расписаний можно рассматривать как математический аппарат, предназначенный для решения задач календарного планирования производственных процессов. Производственные процесс состоят из совокупностей работ. Во многих случаях работы могут быть представлены в виде последовательностей операций, выполняемых на определенных машинах. Так например, обстоит дело с изготовлением деталей на станках. Мы получаем различные задачи теории расписаний в зависимости от числа рассматриваемых работ и их операционного состава; от количества и типов машин; от критерия оценки расписания и т.д.

Рассмотрим примеры задач.

<u>Задача Акерса-Фридмена</u>. Пусть имеется n деталей и m станков. Каждая деталь должна подвергнутся обработке на станках в соответствии с технологией её изготовления. Каждая детал обрабатывается на каждом из m станков ~~один раз~~ в определенном порядке. Для j-й детали известен порядок ее обработки на m станках

$$M_j = (\pi_j^1, \pi_j^2, \dots, \pi_j^m) \text{ - перестановка станков } 1, 2, \dots, m,$$

где π_j^i - номер станка, на котором происходит выполнение i-й операции над j-й деталью, $\pi_j^i \in \{1, 2, \dots, m\}$ и $\pi_j^k \neq \pi_j^s$ при $k \neq s$ (один станок производит ровно одну операцию над деталь

Известно сколько времени занимает обработка любой детали на любом станке

$$\tau_{ij} \geq 0 \text{ — длительность обработки } j\text{-й детали на } i\text{-м станке},$$

где $i \in \{1, 2, \dots, m\}$, $j \in \{1, 2, \dots, n\}$.

В задаче требуется найти моменты начала обработки каждой детали на каждом станке.

Введем обозначения:

t_{ij} - момент начала обработки j-й детали на i-м станке

$\bar{t}_{ij}$ - момент окончания обработки j-й детали на i-м станке.

Мы предполагаем, что процесс обработки детали на станке непрерывный, т.е мы запрещаем прерывание процесса обработки детали на станке. Тогда

$$\bar{t}_{ij} = t_{ij} + \tau_{ij}$$

(момент завершения обработки j-й детали на i-м станке равно момент начала обработки этой детали на этом станке плюс длительность обработки детали).

<u>Опр.</u> Матрица $T = (t_{ij})$ размера $m \times n$ наз-ся допустимым расписанием, если

1. $\forall i \in \{1, 2, \dots, m\}\ \forall j \in \{1, 2, \dots, n\}\ t_{ij} \geq 0$ (для любого ~~стро~~ ~~станка~~ строки i любого столбца j, элемент матрицы, стоящий на пересечении i-й строки и j-го столбца не меньше 0)

2. Операция $k \in \{1, 2, \dots, m-1\}$ над деталью $j \in \{1, 2, \dots, n\}$ не может завершиться позже начала следующей операции над этой деталью

$$\bar{t}_{\pi_j^k j} \leq t_{\pi_j^{k+1} j}$$

3. На одном и том же станке не могут обрабатываться одновременно две детали

$$[t_{ij}, \bar{t}_{ij}) \cap [t_{ik}, \bar{t}_{ik}) = \emptyset$$

Приведем критерии оптимальности расписаний.

Есть директивные сроки завершения обработки деталей на станках. Если мы их нарушаем то нас штрафуют. Таким образом задана матрица

D - матрица размера $m \times n$ директивных сроков завершения операций над деталями.

Эта матрица говорит в какие моменты должны завершиться обработки деталей на станках

D_{ij} – директивный срок завершения обработки j-й детали на i-м станке

Если момент завершения обработки j-й детали на i-м станке позже предписанного срока D_{ij}, т.е. $\bar{t}_{ij} > D_{ij}$, то мы получаем штраф

$$w_{ij} \cdot (\bar{t}_{ij} - D_{ij}),$$

где $w_{ij} \in \mathbb{R}$ и $w_{ij} > 0$. Если момент завершения обработки j-й детали на i-м станке не больше директивного срока D_{ij}, т.е. $\bar{t}_{ij} \leq D_{ij}$, то штрафа нет, т.е. штраф равен 0.

В общем случае штраф за обслуживание j-детали на i-м станке равен

$$\max\left(w_{ij} \cdot (\bar{t}_{ij} - D_{ij}), 0\right) = w_{ij} \cdot \max\left(\bar{t}_{ij} - t_{ij}, 0\right) =$$

Присутствие максимума усложняет запись. Вместо взятия максимума с 0, будем над разностью ставить +

$$= w_{ij} \cdot (\bar{t}_{ij} - t_{ij})^{+}.$$

Критерий оптимальности расписания.

(1) Критерий минимума суммарного штрафа за нарушения директивных сроков обработки деталей

$$\sum_{i,j} w_{ij} (\bar{t}_{ij} - D_{ij})^{+} \to \min_{t_{ij}}$$

(суммируем по всем станкам i и всем деталям j штрафы за нарушения директивн. сроков)

Смотрим при обработке каких деталей на каких станках мы не уложились в срок. Суммируем штрафы. Наша цель минимизировать общий штраф.

(2) Минимаксный критерий.

$$\max_{i,j} w_{ij} \cdot (\bar{t}_{ij} - D_{ij})^{+} \to \min_{t_{ij}}$$

(максимум по всем станкам i и всем деталям j штраф за ~~нарушение~~ обслуживание j-й детали на i-м станке).

В отличие от предыдущего критерия, в котором мы минимизируем суммарный штраф, здесь мы минимизируем максимальный среди всех штрафов.

Частным случаем этого критерия является

(3) Критерий минимума длины расписания.

$$\max_{i,j} \bar{t}_{ij} \to \min_{t_{ij}}$$

Здесь мы минимизируем момент окончания обработки всех деталей.

Пример. Пусть имеется $m = 3$ станка и $n = 2$ детали.

Заданы порядки обработки деталей на станках:

$M_1 = (1, 2, 3)$ (первая деталь сначала обрабатывается на первой машине, затем на второй и затем на третьей).

$M_2 = (3, 2, 1)$

Известны длительности обработки деталей на каждом станке. ~~Время~~ Длительности обработки деталей на станках хранятся в виде матрицы:

$$\tau = \begin{array}{c c} & \begin{array}{cc} 1 & 2 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{pmatrix} 1 & 1 \\ 2 & 1 \\ 1 & 3 \end{pmatrix} \end{array}$$

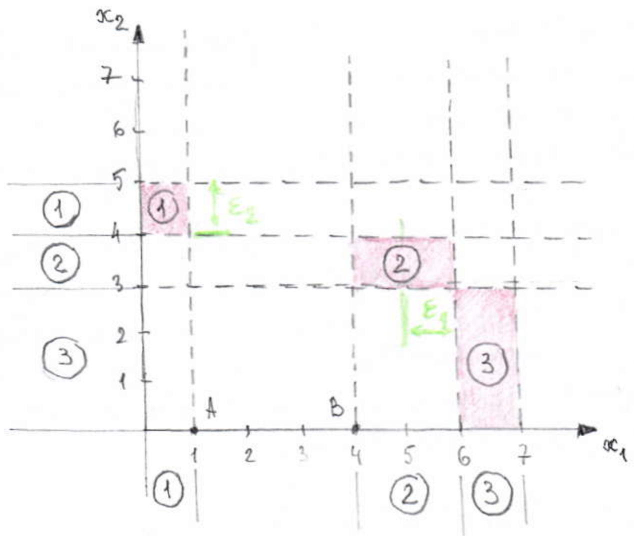
Строки матрицы соответствуют станкам, а столбцы деталям. Например, длительность обработки 1-й детали на ~~в~~ 2-м станке равна $\tau_{21} = 2$.

Требуется найти допустимое расписание, которое оптимально с точки зрения одного из трёх критериев оптимальности.

Расписание удобно представлять в виде матрицы, в которой на пересечении i-й строки и j-го столбца находится момент начала обработки j-й детали на i-м станке. Рассмотрим расписание

$$T = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cc} 1 & 2 \end{array} \\ \begin{pmatrix} 0 & 4 \\ 4 & 3 \\ 6 & 0 \end{pmatrix} \end{array}$$

Расписание для двух деталей удобно изображать графически. Фиксируем на плоскости прямоугольную декартову систему координат. На оси абсцисс отмечаем моменты начала и конца обработки первой детали на станках. На оси ординат отмечаем моменты начала и конца обработки второй детали на станках.



Рассмотрим первый станок. Первая деталь обрабатывается на нем в промежутке $[0,0+1)$, а вторая в промежутке $[4,4+1)$. Отмечаем эти промежутки на оси абсцисс и оси ординат и выделим прямоугольник - декартово произведение этих промежутков.

Рассмотрим второй станок. Первая деталь обрабатывается на нем в промежутке $[4,4+2)$, вторая деталь - в промежутке $[3,3+1)$. Изображаем декартово произведение этих промежутков.

Наконец, рассмотрим третий станок. На этом станке первая деталь обрабатывается в промежутке $[6,6+1)$; вторая деталь - в промежутке $[0,0+3)$. Изображаем декартово произведение этих временных промежутков.

Убедимся, что это расписание допустимо, т.е. что оно удовлетворяет трем условиям из определения.

1) Все моменты начала обработки деталей в расписании T неотрицательны.

2) Проверим, что каждая операция над деталью завершается до того момента как начнётся следующая операция над этой же деталью.

Сначала рассматриваем первую деталь. Видно, что точка завершения (т.A) обработки детали станком 1 находится до точки начала (т.B) обработки станком 2. Также момент завершения обработки детали станком 2 не позже начала обработки детали 3-м станком.

Рассматриваем вторую деталь. Вторая деталь сначала обрабатывается третьим станком, а затем вторым. Момент завершения обработки третьим станком не позже начала обработки вторым станком. Момент завершения обработки вторым станком не позже начала обработки 1-м станком.

3) Проверим, что никакой станок не обрабатывает одновременно две или более деталей.

Изучаем станок 1. Этот станок обрабатывает первую деталь в промежутке $[0,1)$, а вторую деталь в промежутке $[4,5)$. Эти промежутки не пересекаются $[0,1) \cap [4,5) = \emptyset$. Значит в любой момент времени 1-й станок обрабатывает не более одной детали.

Рассматриваем второй станок. Первая деталь обрабатывается на этом станке в промежутке $[4,6)$, а вторая - в промежутке $[3,4)$. Эти два промежутка не пересекаются.

Наконец, третий станок обрабатывает первую деталь в промежутке $[6,7)$ и вторую деталь - в промежутке $[0,3)$. Эти два промежутка также не пересекаются.

Итак, третье условие выполняется и расписание допустимо.

Как оценить качество представленного расписания относительно критериев оптимальности?
Для этого необходимо задать директивные сроки и ~~штрафы за нарушен.~~ штрафные коэффициенты ω за нарушения директивных сроков.

$$D = \begin{array}{c|c} & \begin{array}{cc} 1 & 2 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{pmatrix} 1 & 4 \\ 5 & 4 \\ 7 & 3 \end{pmatrix} \end{array} \text{ – матрица директивных сроков.}$$

Штрафные коэффициенты не будем определять. Для нас это сейчас не важно. Рассмотрим элемент матрицы D, стоящий на пересечении 2-й строки и 1-го столбца $d_{21} = 5$. Это значит, что обработка 1-й детали на 2-м станке должна завершиться не позднее 5 ед. времени. Если позднее, то нам необходимо заплатить штраф. Отметим директивные сроки на картинке. Видим, что директивные сроки нарушаются только при обработке первой детали вторым станком (штраф ε_1) и при обработке второй детали первым станком (штраф ε_2).

Суммарный штраф $\varepsilon_1+\varepsilon_2$. Относительно первого критерия оптимальности расписания оптимальное расписание – это допустимое расписание с минимальным суммарным штрафом.

Можно рассматривать не сумму штрафов, а максимум среди всех штрафов $\max(\varepsilon_1,\varepsilon_2)$. Относительно минимаксного критерия, оптимальное расписание – это допустимое расписание такое, что максимум штрафов минимально.

Можно рассматривать момент завершения обработки всех деталей. В нашем случае, обработка первой детали завершается в момент времени 7, а второй детали – в момент времени 5. Обработка первой детали завершается позже обработки второй. Обработка всех деталей завершится в момент времени 7. С точки зрения третьего критерия оптимальности, оптимальное расписание – это допустимое расписание такое, что момент завершения обработки всех деталей минимален.

Рассмотрим частные случаи задачи Акерса–Фридмена

§ Задачи с одним станком

Пусть число станков $m=1$, а деталей n.
Детали $1,2,\dots,n$.
Перестановка $\pi=(\pi_1,\pi_2,\dots,\pi_n)$ деталей – это некоторое расположение последовательно друг за другом деталей $1,2,\dots,n$.
Ранее мы использовали в обозначениях два нижних индекса: номер станка и номер детали. Поскольку у нас станок 1, то один из индексов не нужен.
Вместо обозначения τ_{1j} будем использовать τ_j:

τ_j – длительность обработки j-й детали $(j\in\{1,2,\dots,n\})$.

Вместо t_{1j} используем t_j

t_j – момент начала обработки j-й детали.

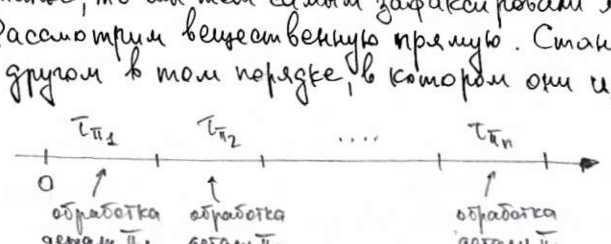
Вместо D_{1j} используем D_j

D_j – директивный срок завершения обслуживания j-й детали

Вместо ω_{1j} используем ω_j

ω_j – штрафующий коэфф. за нарушение директивного срока завершения обработки j-й детали.

Если мы зафиксируем некоторую перестановку $\pi=(\pi_1,\pi_2,\dots,\pi_n)$ n деталей $1,2,\dots,n$, задающую порядок обработки деталей на станке, то мы тем самым зафиксировали моменты начал начал обработки деталей на станке. Почему? Рассмотрим вещественную прямую. Станок обрабатывает детали непрерывно последовательно друг за другом в том порядке, в котором они идут в перестановке π.



~~Обработка первой детали~~ В начале на станок подаётся первая в перестановке π деталь. Обработка этой детали начинается в нулевой момент времени. Заканчивается обработка через τ_{π_1}.

$$t_{\pi_1} = 0 \qquad \bar{t}_{\pi_1} = t_{\pi_1} + \tau_{\pi_1}$$

Как только закончилась обработка первой детали из списка π, немедленно начинается обработка второй детали из списка и заканчивается через τ_{π_2}

$$t_{\pi_2} = t_{\pi_1} + \tau_{\pi_1} \qquad \bar{t}_{\pi_2} = t_{\pi_2} + \tau_{\pi_2}$$

... ...

В общем случае, обработка i-й детали в списке начинается в момент завершения обработки предыдущей детали и заканчивается через τ_{π_i}

$$t_{\pi_i} = t_{\pi_{i-1}} + \tau_{\pi_{i-1}} \qquad \bar{t}_{\pi_i} = t_{\pi_i} + \tau_{\pi_i}.$$

Время начала и завершения обработки последней детали

$$t_{\pi_n} = t_{\pi_{n-1}} + \tau_{\pi_{n-1}} \qquad \bar{t}_{\pi_n} = t_{\pi_n} + \tau_{\pi_n}.$$

Фактически, нам необходимо найти перестановку π деталей, которая минимизирует ту или иную целевую функцию.

Если $\pi = (\pi_1, \pi_2, \dots, \pi_n)$ - перестановка деталей, то будем обозначать

$t_i(\pi)$ - время начала обработки детали i на станке

$\bar{t}_i(\pi)$ - время завершения обработки детали i на станке

<u>Задача Макнотона с критерием минимума суммарного штрафа</u>. Требуется найти перестановку π деталей $1, 2, \dots, n$, которая минимизирует сумму штрафов за нарушение директивных сроков

$$F(\pi) = \sum_{j=1}^{n} \omega_j \cdot \left(\bar{t}_j(\pi) - D_j\right) \to \min_{\pi} \qquad (*)$$

Пусть P_{opt} - это множество оптимальных перестановок π, т.е. перестановок π, которые доставляют минимум целевой функции $F(\pi)$.

Рассмотрим подход к решению этой задачи, в основе которого лежит понятие решающего правила.

Пусть $h: \{1, 2, \dots, n\} \to \mathbb{R}$ - это произвольное отображение, которое каждой детале ставит в соответствие некоторое вещественное число. Введем обозначение

$$P_h = \{\pi = (\pi_1, \pi_2, \dots, \pi_n) : h(\pi_1) \le h(\pi_2) \le \dots \le h(\pi_n)\}$$

P_h - это множество перестановок деталей, которые упорядочивают значения функции h в порядке неубывания.

<u>Пример</u>. Пусть $n = 3$. Рассмотрим функцию $h: \{1,2,3\} \to \mathbb{R}$ $\quad h(1) = 3, h(2) = 2, h(1) = 1$.
Как необходимо переставить детали 1, 2 и 3 так, что значения функции h будут располагаться в неубывающем порядке. В перестановке сначала идут детали с маленькими значениями функции h, а дальше детали с большими значениями функции h

$$(3, 2, 1)$$

т.к. $h(3) \le h(2) \le h(1)$.

Такая перестановка единственная. Поэтому $P_h = \{(3,2,1)\}$.

<u>Опр</u>. Будем говорить, что функция $h: \{1,2,\dots,n\} \to \mathbb{R}$ определяет решающее правило, если $P_h \subseteq P_{opt}$ т.е. если любая перестановка, упорядывающая значения ф-ции h в неубывающем порядке, является оптимальной перестановкой в задаче $(*)$.

Решающее правило наз-ся строгим, если $P_h = P_{opt}$.

Рассмотрим перестановку деталей $\pi = (\pi_1, \pi_2, \dots, \pi_j, \pi_{j+1}, \dots, \pi_n)$. Пусть $\pi^{(j,j+1)}$ - это перестановка деталей, которая получается из перестановки π заменой местами соседних деталей π_j и π_{j+1}.

Рассмотрим два условия

(У1) Для любой перестановки π и любого индекса j имеет место импликация

$$h(\pi_j) \geq h(\pi_{j+1}) \rightarrow F(\pi) \geq F(\pi^{(j,j+1)})$$

(если значение функции h на j-м элементе перестановки π не меньше значения функции h на $(j+1)$-м элементе перестановки π, то значение функции F на π не меньше, чем значение F на перестановке π с переставленными в ней j-м и $(j+1)$-м элементами).

(У2) Для любой перестановки π и любого индекса j имеет место импликация

$$h(\pi_j) > h(\pi_{j+1}) \rightarrow F(\pi) > F(\pi^{(j,j+1)}).$$

Лемма. ① (У1) $\rightarrow P_h \subseteq P_{opt}$;

② (У2) $\rightarrow P_{opt} \subseteq P_h$.

Док-во. ② Пусть выполняется условие (У2). Пусть $\pi \in P_{opt}$. Хотим показать, что $\pi \in P_h$. От противного. Допустим $\pi \notin P_h$. Перестановка π не упорядочивает значения функции h, т.е. найдётся индекс j

$$h(\pi_j) > h(\pi_{j+1}).$$

По условию (У1) из этого вытекает

$$F(\pi) > F(\pi^{(j,j+1)}).$$

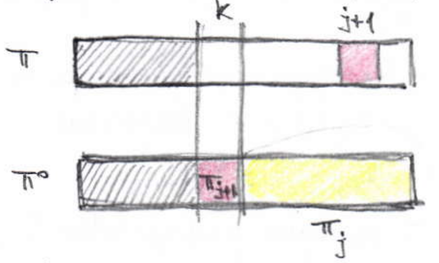
Существует перестановка, на которой значение F меньше, чем на оптимальной перестановке. Полученное противоречие доказывает $\pi \in P_h$.

① Пусть выполнено условие (У1). Пусть $\pi^\circ \in P_h$, т.е. перестановка π° упорядочивает значения функции h по неубыванию.

Покажем, что $\pi^\circ \in P_{opt}$. От противного, допустим $\pi^\circ \notin P_{opt}$. Рассмотрим произвольную перестановку π отличную от π°

$$\pi \neq \pi^\circ.$$

Нас будут интересовать совпадающие префиксы у перестановок π и π°.



Пусть k - минимальный индекс, в котором перестановки π и π° различаются. Начала перестановок π и π° до k-го элемента идентичны. Заметим, что идентичные начала перестановок π и π° могут быть пустыми.

$$\pi^\circ_i = \pi_i \quad \forall i \in \{1,2,\dots,k-1\} \quad \text{и} \quad \pi^\circ_k \neq \pi_k.$$

Посмотрим на k-й элемент перестановки π°. Этот элемент есть в перестановке π. Пусть этот элемент находится на позиции $j+1$ и $j \geq k$.

Обозначим индексы j и k так

$$j = j(\pi) \quad \text{и} \quad k = k(\pi),$$

чтобы подчеркнуть, что эти индексы зависят от перестановки π.

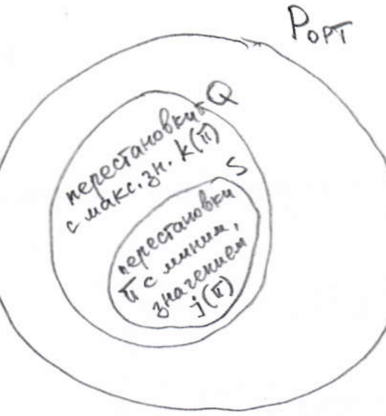
В множестве P_{opt} выделим перестановки с максимальным значением k. Получим мн-во G. Среди выделенных перестановок с макс. значением k выберем перестановки с минимальным значением j. Получим мн-во S.

Пусть $\pi \in S$. Посмотрим на j-й элемент перестановки, т.е. на элемент π_j. Где находится этот элемент в перестановке π°? Во второй части

$$\pi_j \in \{\pi^\circ_{k+1}, \pi^\circ_{k+2}, \dots, \pi^\circ_n\}.$$

Вспомним, что $\pi^0 \in P_h$, т.е перестановка π^0 упорядочивает значения функции h. Следовательно,

$$h(\pi_{j+1}) \geq h(\pi_j).$$



По перестановке π построим новую перестановку $\pi^{(j,j+1)}$.

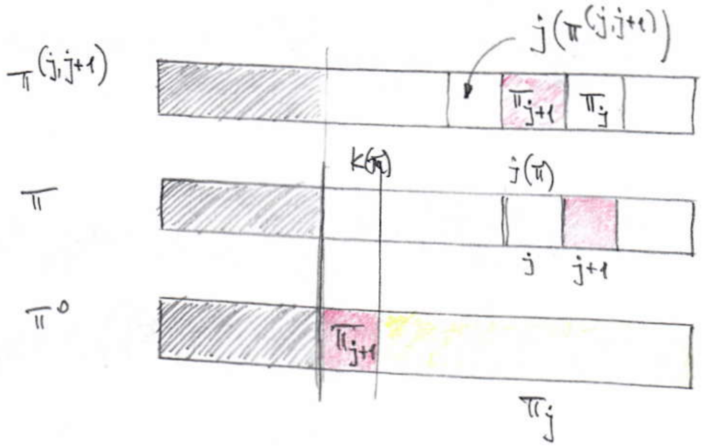
По условию (У1)

$$F(\pi) \geq F(\pi^{(j,j+1)}).$$

новая перестановка

Т.к π – это оптимальная перестановка, то $\pi^{(j,j+1)}$ тоже оптимальная перестановка.

$$\pi^{(j,j+1)} \in P_{OPT}.$$



В новой перестановке первые $k-1$ элементов идентичны первым $k-1$ элементам перестановки π^0. Значит

$$k(\pi^{(j,j+1)}) \geq k(\pi).$$

Если $k(\pi^{(j,j+1)}) > k(\pi)$, то π – это не перестановка из P_{OPT} с максимальным значением $k(\pi)$, т.е. $\pi \notin P_{OPT}$. Следовательно,

$$k(\pi^{(j,j+1)}) = k(\pi).$$

Стало быть, индекс k у новой перестановки $\pi^{(j,j+1)}$ максимален и $\pi^{(j,j+1)} \in Q$. Отметим $j(\pi)$ и $j(\pi^{(j,j+1)})$. Видим, что $j(\pi) > j(\pi^{(j,j+1)})$. Значит перестановка π не является перестановкой из S с минимальным значением индексом $j(\pi)$. Получили противоречие. ∎

Рассмотрим частные случаи задачи Макнотона.

<u>Случай 1</u>. Пусть все директивные сроки равны 0.

$$D_j = 0, \quad \forall j \in \{1, 2, \dots, n\}$$

Целевая функция (*) примет вид

$$\sum_{j=1}^{n} w_j \cdot (\bar{t}_j - D_j)^+ = \sum_{j=1}^{n} w_j \cdot \bar{t}_j = \sum_{j=1}^{n} w_j \cdot (t_j + \tau_j) = \sum_{j=1}^{n} w_j \cdot t_j + \underbrace{\sum_{j=1}^{n} w_j \cdot \tau_j}_{=\alpha},$$

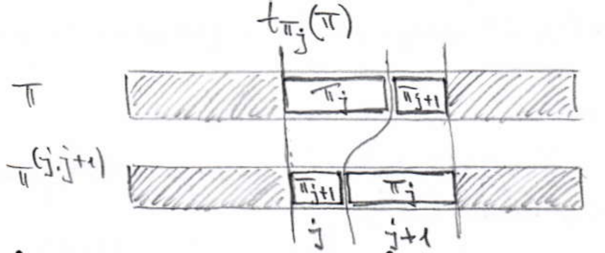
где α – это константа, которая не зависит от расписания. Поэтому мы можем исключить это слагаемое из рассмотрения. Итак, мы ищем перестановку π деталей такую, что

$$\sum_{j=1}^{n} w_j \cdot t_j(\pi) \to \min_{\pi} \qquad (**)$$

<u>Теорема 1</u>. Функция $h(j) = \frac{\tau_j}{w_j}$ определяет строгое решающее правило для задачи (**).

<u>Док-во</u>. Рассмотрим произвольную перестановку π деталей $1, 2, \dots, n$. В перестановке π поменяем местами детали π_j и π_{j+1}, стоящие на позициях j и $j+1$. Получим новую перестановку $\pi^{(j,j+1)}$. Эти две перестановки отличаются друг от друга только в двух

позициях j и $j+1$.



Расписания, соответствующие этим перестановкам, отличаются друг от друга лишь моментами начала работ π_j и π_{j+1}.

$$F(\pi) - F(\pi^{(j,j+1)}) = \sum_{k=1}^{n} w_k \cdot t_k(\pi) - \sum_{k=1}^{n} w_k \cdot t_k(\pi^{(j,j+1)}) = w_{\pi_j} \cdot t_{\pi_j}(\pi) + w_{\pi_{j+1}} \cdot t_{\pi_{j+1}}(\pi) -$$

$$- \underbrace{w_{\pi_j^{(j,j+1)}}}_{w_{\pi_{j+1}}} \cdot \underbrace{t_{\pi_j^{(j,j+1)}}(\pi^{(j,j+1)})}_{t_{\pi_j}(\pi)} - \underbrace{w_{\pi_{j+1}^{(j,j+1)}}}_{w_{\pi_j}} \cdot \underbrace{t_{\pi_{j+1}^{(j,j+1)}}(\pi^{(j,j+1)})}_{t_{\pi_j}(\pi) + \tau_{\pi_{j+1}}} =$$

$$= \underline{w_{\pi_j} \cdot t_{\pi_j}(\pi)} + w_{\pi_{j+1}} \cdot t_{\pi_{j+1}}(\pi) - w_{\pi_{j+1}} \cdot t_{\pi_j}(\pi) - \underline{w_{\pi_j}(t_{\pi_j}(\pi) + \tau_{\pi_{j+1}})} =$$

$$= -w_{\pi_j} \cdot (\cancel{t_{\pi_j}(\pi)} + \tau_{\pi_{j+1}} - \cancel{t_{\pi_j}(\pi)}) + w_{\pi_{j+1}} \cdot (\underbrace{t_{\pi_{j+1}}(\pi) - t_{\pi_j}(\pi)}_{\tau_{\pi_j}}) =$$

$$= w_{\pi_{j+1}} \cdot \tau_{\pi_j} - w_{\pi_j} \cdot \tau_{\pi_{j+1}} = w_{\pi_j} \cdot \underbrace{\frac{\tau_{\pi_j}}{w_{\pi_j}}}_{h(\pi_j)=} \cdot w_{\pi_{j+1}} - w_{\pi_{j+1}} \cdot \underbrace{\frac{\tau_{\pi_{j+1}}}{w_{\pi_{j+1}}}}_{=h(\pi_{j+1})} \cdot w_{\pi_j} =$$

$$= \underbrace{w_{\pi_j} \cdot w_{\pi_{j+1}}}_{\alpha} \cdot (h(\pi_j) - h(\pi_{j+1}))$$

Мы предполагаем, что $w_k > 0$, $\forall k \in \{1, 2, \dots, n\}$. Получаем

$$F(\pi) - F(\pi^{(j,j+1)}) = \underset{\overset{\vee}{0}}{\alpha} \cdot (h(\pi_j) - h(\pi_{j+1})).$$

Убедимся, что для функции h выполнено условие (У1)

$$h(\pi_j) \geq h(\pi_{j+1}) \rightarrow 0 \leq \alpha \cdot (h(\pi_j) - h(\pi_{j+1})) = F(\pi) - F(\pi^{(j,j+1)}) \rightarrow 0 \leq F(\pi) - F(\pi^{(j,j+1)})$$

$$\rightarrow F(\pi) \geq F(\pi^{(j,j+1)}).$$

Аналогичным образом, проверяется выполнимость условия (У2).

По лемме $P_h = P_{OPT}$ (мн-во перестановок, упорядывающих значения функции h совпадает с множеством оптимальных перестановок). ▨

Эта теорема лежит в основе алгоритма решения задачи (*) в случае 1, т.е. когда все директивные сроки $= 0$.

<u>Пример</u>. Пусть $n = 3$ – кол-во деталей. Длительности обработки деталей

$\tau_1 = 3$ (деталь 1 обрабатывается 3 ед. времени)
$\tau_2 = 2$
$\tau_3 = 4$

Значения <u>штрафующих коэффициентов</u> для каждой детали:

$w_1 = 2$
$w_2 = 1$
$w_3 = 4$

Для каждой детали j вычислим отношение τ_j/w_j:

$$\frac{\tau_1}{w_1}=\frac{3}{2} \qquad \frac{\tau_2}{w_2}=2 \qquad \frac{\tau_3}{w_3}=1.$$

Отсортируем эту последовательность в порядке неубывания: $\frac{\tau_3}{w_3}=1 \le \frac{\tau_1}{w_1}=\frac{3}{2} \le \frac{\tau_2}{w_2}=2$. Перестановка деталей $(3,1,2)$ упорядочивает значения функции $h(j)=\tau_j/w_j$. По теореме 1 эта перестановка является оптимальной перестановкой задачи (**). Сначала обрабатываем на станке 3-ю деталь, затем обрабатываем 1-ую деталь и, наконец, 2-ую деталь.

Случай 2. Пусть длительности обработки деталей на станке равны:

$$\tau_1=\tau_2=\ldots=\tau_n=\tau>0$$

и значения штрафующих коэффициентов равны.

$$w_1=w_2=\ldots=w_n=w>0.$$

Целевая функция (*) примет вид

$$F(\pi)=\sum_{k=1}^{n}\overset{w}{\overset{\|}{w_k}}\cdot\Big(\overset{=k\cdot\tau}{t_{\pi_k}(\pi)}-D_{\pi_k}\Big)^{+}=\sum_{k=1}^{n}w\cdot(k\cdot\tau-D_{\pi_k})^{+}.$$

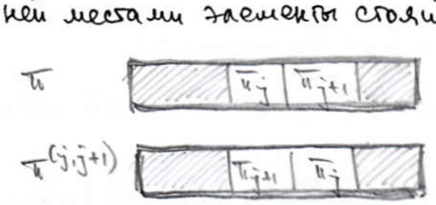
(берем сумму по всем индексам k элемент. перестановки π)

Эту целевую функцию мы хотим минимизировать. Заметим, что w и τ не зависят от перестановки π. Мы можем разделить сумму на w и на τ. Величины w и τ исчезнут. Можно рассматривать задачу нахождения перестановки π, которая минимизирует функционал

$$F(\pi)=\sum_{k=1}^{n}(k-D_{\pi_k})^{+}\to\min \qquad (***)$$

Теорема 2. Функция $h(j)=D_j$ определяет решающее правило для задачи (***).

Док-во. Достаточно проверить, что выполняется условие (У1). Рассмотрим произвольную перестановку π. Поменяем в ней местами элементы стоящие на j-м и $(j+1)$-м местах.



$$F(\pi)-F(\pi^{(j,j+1)})=\sum_{k=1}^{n}(k-D_{\pi_k})^{+}-\sum_{k=1}^{n}(k-D_{\pi_k^{(j,j+1)}})^{+}=(j-D_{\pi_j})^{+}+(j+1-D_{\pi_{j+1}})^{+}-$$

$$-\Big(j-\underbrace{D_{\pi_j^{(j,j+1)}}}_{D_{\pi_{j+1}}}\Big)-\Big(j+1-\underbrace{D_{\pi_{j+1}^{(j,j+1)}}}_{D_{\pi_j}}\Big)=(j-D_{\pi_j})^{+}+(j+1-D_{\pi_{j+1}})^{+}-$$

$$-(j-D_{\pi_{j+1}})^{+}-(j+1-D_{\pi_j})^{+}=\Big((j+1-D_{\pi_{j+1}})^{+}-(j-D_{\pi_{j+1}})^{+}\Big)-$$

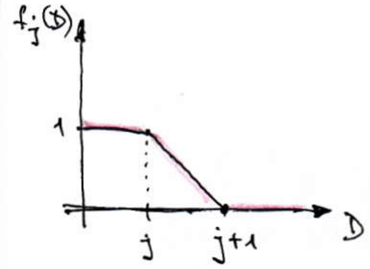
$$-\Big((j+1-D_{\pi_j})^{+}-(j-D_{\pi_j})^{+}\Big)=f_j(D_{\pi_{j+1}})-f_j(D_{\pi_j}),$$

где $f_j(D)=(j+1-D)^{+}-(j-D)^{+}$ – функция.

Утв. Функция $f_j(D)$ невозрастающая.

Док-во. Изучим функцию $f_j(D)$. Разобьем область её определения на три части: $[0,j)$, $[j,j+1)$, $[j+1,\infty)$.

① Если $D<j$, то $f_j(D)=(\underset{\overset{\vee}{0}}{j+1-D})^{+}-(\underset{\overset{\vee}{0}}{j-D})^{+}=(j+1-D)-(j-D)=1$.



② Если $j \le D < j+1$, то $f_j(D) = (j+1-D)^+ - \underbrace{(j-D)^+}_{0} = j+1-D$.

③ Если $D \ge j+1$, то $f_j(D) = \underbrace{(j+1-D)^+}_{0} - \underbrace{(j-D)^+}_{0} = 0$

Видим, что функция $f_j(D)$ невозрастающая.

Возвращаемся к доказательству теоремы 2. Имеем

$$F(\pi) - F(\pi^{(j,j+1)}) = f_j(D_{\pi_{j+1}}) - f_j(D_{\pi_j}).$$

Все готово для того, чтобы проверить условие (У1). Пусть $h(\pi_j) \ge h(\pi_{j+1})$. Тогда $D_{\pi_j} \ge D_{\pi_{j+1}}$. Т.к. $f_j(D)$ - невозрастающая функция, то

$$f_j(D_{\pi_j}) \le f_j(D_{\pi_{j+1}}) \qquad f_j(D_{\pi_{j+1}}) - f_j(D_{\pi_j}) \ge 0.$$

Следовательно, $F(\pi) - F(\pi^{(j,j+1)}) \ge 0 \qquad F(\pi) \ge F(\pi^{(j,j+1)})$.

Эта теорема лежит в основе алгоритма решения задачи (***).

Пример. Пусть $n=3$ деталей. Длительности обработки деталей равны 1

$$\tau_1 = \tau_2 = \tau_3 = 1.$$

Все штрафные коэффициенты равны 1: $w_1 = w_2 = w_3 = 1$. Директивные сроки

$$D_1 = 2, \ D_2 = 1, \ D_3 = 3.$$

Эта последовательность является последовательностью значений ф-ции h

$$h(1) = D_1 = 2 \qquad h(2) = D_2 = 1 \qquad h(3) = D_3 = 3.$$

Требуется найти перестановку деталей 1, 2 и 3, упорядочивающую значения функции h в порядке неубывания. Сортируем последовательность значений функции h по неубыванию

$$D_2 = 1, \ D_1 = 2 \text{ и } D_3 = 3.$$

Получаем перестановку деталей (2, 1, 3), которая является оптимальной перестановкой задачи (***).

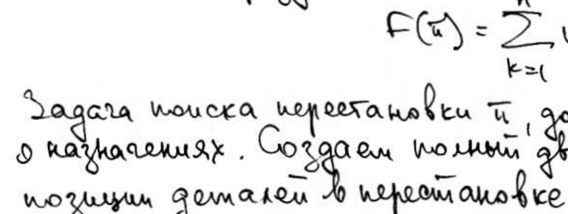
Случай 3. Пусть длительности обработки деталей равны $\tau_1 = \tau_2 = \ldots = \tau_n = \tau > 0$. Пусть $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ - это перестановка деталей $1, 2, \ldots, n$, определяющая порядок обработки деталей. Время окончания обработки детали π_j $t_{\pi_j} = \tau \cdot j$. Целевая функция примет вид

$$F(\pi) = \sum_{k=1}^{n} w_{\pi_k} \cdot (t_{\pi_k}(\pi) - D_{\pi_k})^+ = \sum_{k=1}^{n} w_{\pi_k} \cdot (\tau \cdot k - D_{\pi_k})^+.$$

Мы можем разделить на τ и получим целевую функцию

$$F(\pi) = \sum_{k=1}^{n} w_{\pi_k} \cdot (k - D_{\pi_k})^+ \to \min_{\pi} \quad (****)$$

Задача поиска перестановки π, доставляющей минимум этой целевой функции, сводится к задаче о назначениях. Создаем полный двудольный граф, в одной доле которого детали, а в другой - позиции деталей в перестановке. Между долями проведены все возможные ребра. Припишем ребрам веса, если мы помещаем в перестановку на k-ую позицию деталь j, то платим штраф



$$c_{kj} = w_j \cdot (k - D_j)^+.$$

Этот штраф и есть вес ребра ~~c_{kj}~~ $\{k, j\}$.

Совершенное паросочетание мин. веса определяет оптимальную перестановку π в (****)