О компьютеризированных доказательствах в теории расписаний

Черных И.Д.

Семинар “Модели и алгоритмы
для задач составления расписаний”
1.04.2023

*Исследование выполнено при поддержке
гранта РНФ № 22-71-10015*

Нижняя оценка оптимума и свойства примеров

Пусть $F(x) \to \min$ — задача на минимум, $LB \leqslant F^*$ — нижняя оценка оптимума.
Нас интересует доказательство свойств примеров, связанных с LB:

- Полиномиально разрешимые случаи, для которых $F^* = LB$;
- Алгоритмы приближенного решения, опирающиеся на LB:

$$F(x_A) \leqslant \rho LB \leqslant \rho F^*;$$

- Локализация оптимумов.

Локализация оптимумов

Определение

Точным интервалом локализации оптимумов для класса примеров $\mathcal{I}$ некоторой оптимизационной задачи на минимум относительно некоторой нижней оценки LB называется наиболее тесный интервал вида

$$\mathtt{OL}_{LB}(\mathcal{I}) = [LB, \rho^* LB],$$

содержащий оптимумы всех примеров из $\mathcal{I}$.

Локализация оптимумов

Определение

Точным интервалом локализации оптимумов для класса примеров $\mathcal{I}$ некоторой оптимизационной задачи на максимум относительно некоторой верхней оценки UB называется наиболее тесный интервал вида

$$\mathtt{OL}_{UB}(\mathcal{I}) = [\rho^* UB, UB],$$

содержащий оптимумы всех примеров из $\mathcal{I}$.

Локализация оптимумов

Определение

Точным интервалом локализации оптимумов для класса примеров $\mathcal{I}$ некоторой оптимизационной задачи на минимум относительно некоторой нижней оценки LB называется наиболее тесный интервал вида

$$\mathtt{OL}_{LB}(\mathcal{I}) = [LB, \rho^* LB],$$

содержащий оптимумы всех примеров из $\mathcal{I}$.

$$\rho^* = \sup_{I \in \mathcal{I}} \alpha(I) = \sup_{I \in \mathcal{I}} \frac{\mathrm{OPT}(I)}{LB(I)}.$$

Локализация оптимумов

Определение

Точным интервалом локализации оптимумов для класса примеров $\mathcal{I}$ некоторой оптимизационной задачи на минимум относительно некоторой нижней оценки LB называется наиболее тесный интервал вида

$$\mathtt{OL}_{LB}(\mathcal{I}) = [LB, \rho^* LB],$$

содержащий оптимумы всех примеров из $\mathcal{I}$.

$$\rho^* = \sup_{I \in \mathcal{I}} \alpha(I) = \sup_{I \in \mathcal{I}} \frac{\mathrm{OPT}(I)}{LB(I)}.$$

Зачем?

1. Качество нижней оценки LB.
2. Верхняя граница для оценки точности приближенных алгоритмов, опирающихся на LB.
3. Потенциал для построения приближенных алгоритмов с оценкой точности, неулучшаемой относительно LB.

Зачем это нужно?

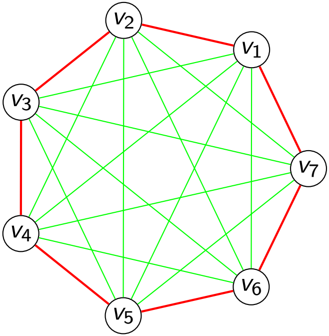
2-PSP-2w-max.

Постановка: в полном графе с двумя весовыми функциями w_1 и w_2 на рёбрах найти два непересекающихся гамильтоновых цикла H_1 и H_2 так, чтобы $w_1(H_1) + w_2(H_2) \to \max$.
Верхняя оценка: $UB = w_1^* + w_2^*$, где w_i^* — наибольший вес циклового покрытия относительно фунции w_i.

Зачем это нужно?

2-PSP-2w-max.

Постановка: в полном графе с двумя весовыми функциями w_1 и w_2 на рёбрах найти два непересекающихся гамильтоновых цикла H_1 и H_2 так, чтобы $w_1(H_1) + w_2(H_2) \to \max$.
Верхняя оценка: $UB = w_1^* + w_2^*$, где w_i^* — наибольший вес циклового покрытия относительно фунции w_i.



$$w_1 = w_2 = \begin{cases} 1 \text{ для красных рёбер}, \\ 0 \text{ для зелёных рёбер} \end{cases}$$

“Общая” постановка задачи

- Множество работ $\mathcal{J} = \{J_1, \ldots, J_n\}$,
- множество машин $\mathcal{M} = \{M_1, \ldots, M_m\}$,
- каждая машина M_i выполняет одну операцию O_{ji} для каждой работы J_j, даны длительности операций

$$P = \begin{pmatrix} p_{11} & p_{21} & p_{31} & \cdots & p_{n1} \\ p_{12} & p_{22} & p_{32} & \cdots & p_{n2} \\ \vdots & \vdots & \vdots & & \vdots \\ p_{1m} & p_{2m} & p_{3m} & \cdots & p_{nm} \end{pmatrix},$$

- операции одной работы или одной машины не могут пересекаться.

Многостадийные задачи теории расписаний

"Общая" постановка задачи

- Множество работ $\mathcal{J} = \{J_1, \dots, J_n\}$,
- множество машин $\mathcal{M} = \{M_1, \dots, M_m\}$,
- каждая машина M_i выполняет одну операцию O_{ji} для каждой работы J_j, даны длительности операций

$$P = \begin{pmatrix} p_{11} & p_{21} & p_{31} & \dots & p_{n1} \\ p_{12} & p_{22} & p_{32} & \dots & p_{n2} \\ \vdots & \vdots & \vdots & & \vdots \\ p_{1m} & p_{2m} & p_{3m} & \dots & p_{nm} \end{pmatrix},$$

- операции одной работы или одной машины не могут пересекаться.

Стандартная нижняя оценка длины расписания $C_{\max}$:

$$\bar{C} = \max_{i,j}\{\ell_i, d_j\} = \max_{i,j}\left\{\sum_{j=1}^{n} p_{ji}, \sum_{i=1}^{m} p_{ji}\right\}.$$

Примеры задач

Flow shop ($Fm||C_{max}$)

Операции каждой работы должны выполняться в порядке, совпадающем с порядком нумерации машин. Порядок выполнения операций на каждой машине не ограничен.

Перестановочный вариант Flow shop ($Fm|prmu|C_{max}$)

Операции каждой работы должны выполняться в порядке, совпадающем с порядком нумерации машин. Порядок выполнения операций на каждой машине не ограничен, но должен быть одинаков для всех машин.

Open shop ($Om||C_{max}$)

Операции могут выполняться в произвольном порядке, который обеспечивает допустимость (непересечение интервалов выполнения операций одной работы или одной машины).

Ранние расписания

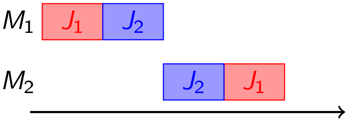
Определение

Расписание S называется ранним, если никакую операцию нельзя начать раньше, чем в S, без нарушения допустимости и порядка выполнения операций, индуцированного расписанием S.

Ранние расписания

Определение

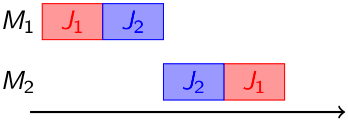
Расписание S называется **ранним**, если никакую операцию нельзя начать раньше, чем в S, без нарушения допустимости и порядка выполнения операций, индуцированного расписанием S.

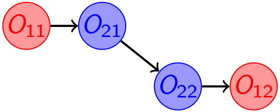


Ранние расписания

Определение

Расписание S называется ранним, если никакую операцию нельзя начать раньше, чем в S, без нарушения допустимости и порядка выполнения операций, индуцированного расписанием S.

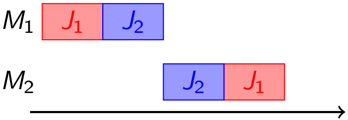


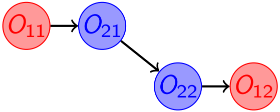


Ранние расписания

Определение

Расписание S называется **ранним**, если никакую операцию нельзя начать раньше, чем в S, без нарушения допустимости и порядка выполнения операций, индуцированного расписанием S.





Достаточно рассматривать только ранние расписания (среди них точно есть оптимальное). Для описания раннего расписания достаточно задать линейные порядки для операций каждой работы и каждой машины.

Схемы расписаний

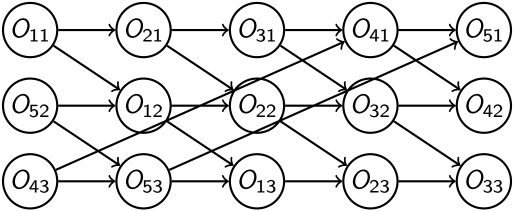
Определение

Схемой расписания для задачи с m машинами и n работами будем называть частичный порядок на множестве операций, в котором линейно упорядочены операции каждой работы и операции каждой машины. Такие схемы будем описывать с помощью графов транзитивной редукции порядка. Схема в совокупности с конкретным входом (матрицей длительностей) определяет единственное раннее расписание, длина такого расписания равна длине критического пути.

Схемы расписаний

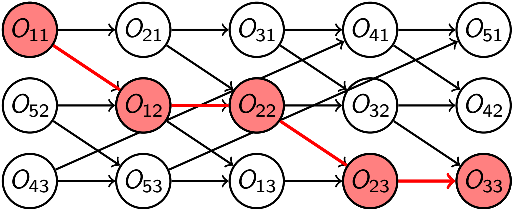
Определение

Схемой расписания для задачи с m машинами и n работами будем называть частичный порядок на множестве операций, в котором линейно упорядочены операции каждой работы и операции каждой машины. Такие схемы будем описывать с помощью графов транзитивной редукции порядка. Схема в совокупности с конкретным входом (матрицей длительностей) определяет единственное раннее расписание, длина такого расписания равна длине критического пути.



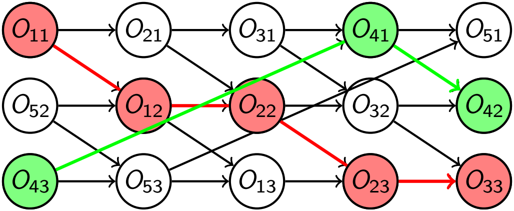
Определение

Схемой расписания для задачи с m машинами и n работами будем называть частичный порядок на множестве операций, в котором линейно упорядочены операции каждой работы и операции каждой машины. Такие схемы будем описывать с помощью графов транзитивной редукции порядка. Схема в совокупности с конкретным входом (матрицей длительностей) определяет единственное раннее расписание, длина такого расписания равна длине критического пути.



Определение

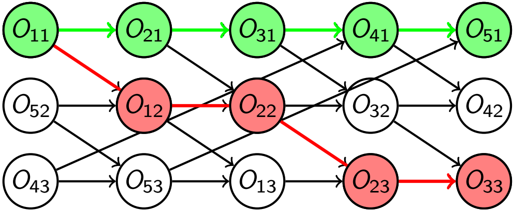
Схемой расписания для задачи с m машинами и n работами будем называть частичный порядок на множестве операций, в котором линейно упорядочены операции каждой работы и операции каждой машины. Такие схемы будем описывать с помощью графов транзитивной редукции порядка. Схема в совокупности с конкретным входом (матрицей длительностей) определяет единственное раннее расписание, длина такого расписания равна длине критического пути.



Схемы расписаний

Определение

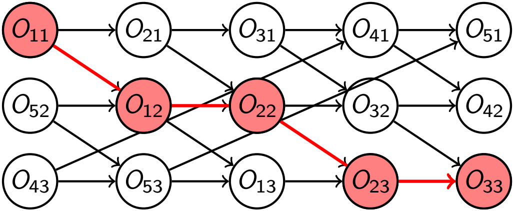
Схемой расписания для задачи с m машинами и n работами будем называть частичный порядок на множестве операций, в котором линейно упорядочены операции каждой работы и операции каждой машины. Такие схемы будем описывать с помощью графов транзитивной редукции порядка. Схема в совокупности с конкретным входом (матрицей длительностей) определяет единственное раннее расписание, длина такого расписания равна длине критического пути.



Схемы расписаний

Определение

Схемой расписания для задачи с m машинами и n работами будем называть частичный порядок на множестве операций, в котором линейно упорядочены операции каждой работы и операции каждой машины. Такие схемы будем описывать с помощью графов транзитивной редукции порядка. Схема в совокупности с конкретным входом (матрицей длительностей) определяет единственное раннее расписание, длина такого расписания равна длине критического пути.



Перестановочные расписания для $Fm||C_{\max}$

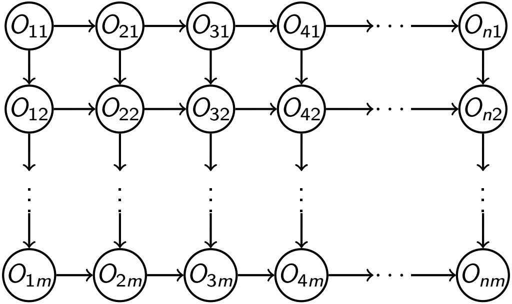
Определение

Раннее расписание для $F||C_{\max}$, в котором все машины выполняют работы в одном и том же порядке π, называется перестановочным и обозначается S_π.

Перестановочные расписания для $Fm||C_{\max}$

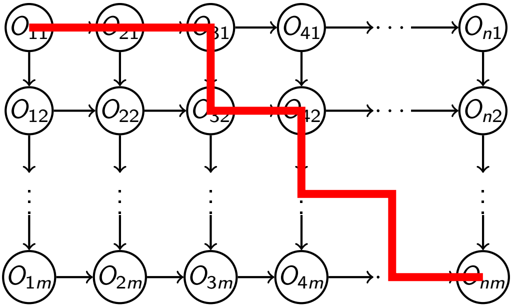
Определение

Раннее расписание для $F||C_{\max}$, в котором все машины выполняют работы в одном и том же порядке π, называется перестановочным и обозначается S_π.



Определение

Раннее расписание для $F||C_{max}$, в котором все машины выполняют работы в одном и том же порядке π, называется перестановочным и обозначается S_π.



Алгоритм Авербаха-Бермана

Алгоритм

1. Положить $\pi = (1, 2, \ldots, n)$, $\pi' = (n, n-1, \ldots, 1)$.
2. Построить расписания S_π и $S_{\pi'}$.
3. Выдать лучшее из них.

Алгоритм Авербаха-Бермана

Алгоритм

1. Положить $\pi = (1, 2, \dots, n)$, $\pi' = (n, n-1, \dots, 1)$.
2. Построить расписания S_π и $S_{\pi'}$.
3. Выдать лучшее из них.

Теорема

Пусть S_{AB} — расписание, построенное алгоритмом Авербаха-Бермана. Тогда

$$C_{\max}(S_{AB}) \leqslant \frac{m+1}{2}\bar{C}.$$

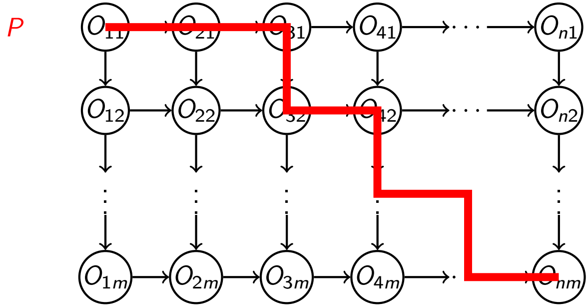
Доказательство

Расписание S_π:

$$
\begin{tikzcd}
O_{11} \arrow[r] \arrow[d] & O_{21} \arrow[r] \arrow[d] & O_{31} \arrow[r] \arrow[d] & O_{41} \arrow[r] \arrow[d] & \cdots \arrow[r] & O_{n1} \arrow[d] \\
O_{12} \arrow[r] \arrow[d] & O_{22} \arrow[r] \arrow[d] & O_{32} \arrow[r] \arrow[d] & O_{42} \arrow[r] \arrow[d] & \cdots \arrow[r] & O_{n2} \arrow[d] \\
\vdots \arrow[d] & \vdots \arrow[d] & \vdots \arrow[d] & \vdots \arrow[d] & & \vdots \arrow[d] \\
O_{1m} \arrow[r] & O_{2m} \arrow[r] & O_{3m} \arrow[r] & O_{4m} \arrow[r] & \cdots \arrow[r] & O_{nm}
\end{tikzcd}
$$

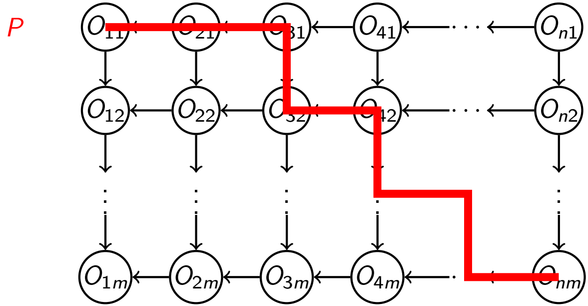
Доказательство

Расписание S_π:



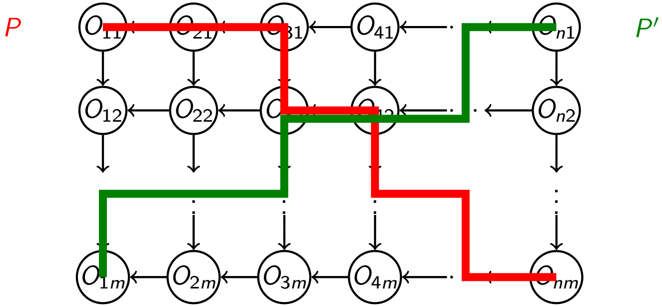
Доказательство

Расписание $S_{\pi'}$:



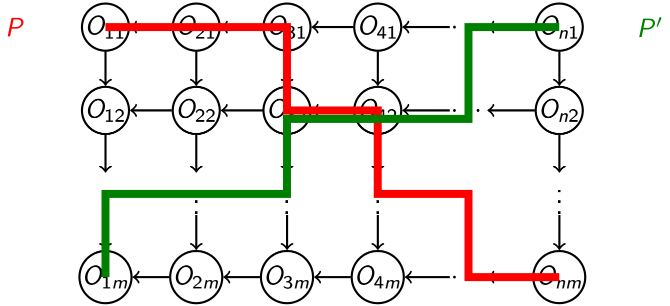
Доказательство

Расписание $S_{\pi'}$:



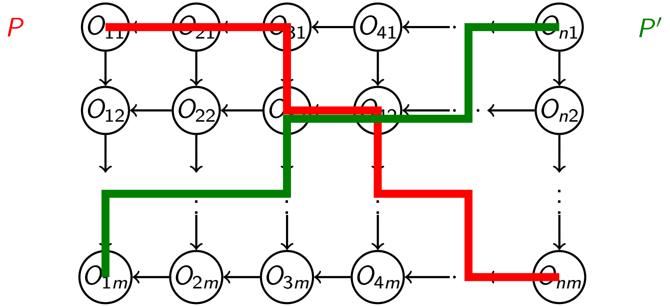
Доказательство

Расписание $S_{\pi'}$:



$$C_{\max}(S_\pi)+C_{\max}(S_{\pi'}) = \sum_{O_{ji}\in P} p_{ji} + \sum_{O_{ji}\in P'} p_{ji} = \sum_{O_{ji}\in P\cup P'} p_{ji} + \sum_{O_{ji}\in P\cap P'} p_{ji}$$

Расписание $S_{\pi'}$:



$$C_{\max}(S_\pi)+C_{\max}(S_{\pi'}) = \sum_{O_{ji}\in P} p_{ji} + \sum_{O_{ji}\in P'} p_{ji} = \sum_{O_{ji}\in P\cup P'} p_{ji} + \sum_{O_{ji}\in P\cap P'} p_{ji} \leqslant$$

$$\sum_{i,j} p_{ji} + \bar{C} = \sum_{i=1}^{m} \ell_i + \bar{C} \leqslant (m+1)\bar{C}.$$

Локализация оптимумов для $F2||C_{\max}$ и $F3||C_{\max}$

Пусть $\mathcal{I}_m$ — класс примеров задачи $Fm||C_{\max}$. Вопрос: для какого минимального значения $\rho^F(m)$ гарантируется

$$\forall I \in \mathcal{I}_m \; C^*_{\max}(I) \in [\bar{C}, \rho^F(m)\bar{C}]?$$

Локализация оптимумов для $F2||C_{\max}$ и $F3||C_{\max}$

Пусть $\mathcal{I}_m$ — класс примеров задачи $Fm||C_{\max}$. Вопрос: для какого минимального значения $\rho^F(m)$ гарантируется

$$\forall I \in \mathcal{I}_m \; C^*_{\max}(I) \in [\bar{C}, \rho^F(m)\bar{C}]?$$

Из теоремы об алгоритме А.-Б. имеем $\rho^F(m) \leqslant \dfrac{m+1}{2}$.

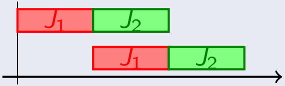
Пусть $\mathcal{I}_m$ — класс примеров задачи $Fm||C_{\max}$. Вопрос: для какого минимального значения $\rho^F(m)$ гарантируется

$$\forall I \in \mathcal{I}_m \; C^*_{\max}(I) \in [\bar{C}, \rho^F(m)\bar{C}]?$$

Из теоремы об алгоритме А.-Б. имеем $\rho^F(m) \leqslant \dfrac{m+1}{2}$.

$\rho^F(2) = 3/2$

$$P = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$



Локализация оптимумов для $F2||C_{\max}$ и $F3||C_{\max}$

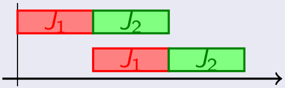
Пусть $\mathcal{I}_m$ — класс примеров задачи $Fm||C_{\max}$. Вопрос: для какого минимального значения $\rho^F(m)$ гарантируется

$$\forall I \in \mathcal{I}_m \; C^*_{\max}(I) \in [\bar{C}, \rho^F(m)\bar{C}]?$$

Из теоремы об алгоритме А.-Б. имеем $\rho^F(m) \leqslant \dfrac{m+1}{2}$.

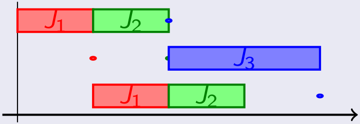
$\rho^F(2) = 3/2$

$$P = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$



$\rho^F(3) = 2$

$$P = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 2 \\ 1 & 1 & 0 \end{pmatrix}$$



Локализация оптимумов для $F2||C_{\max}$ и $F3||C_{\max}$

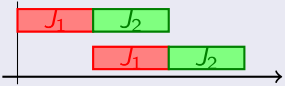
Пусть $\mathcal{I}_m$ — класс примеров задачи $Fm||C_{\max}$. Вопрос: для какого минимального значения $\rho^F(m)$ гарантируется

$$\forall I \in \mathcal{I}_m \;\; C^*_{\max}(I) \in [\bar{C}, \rho^F(m)\bar{C}]?$$

Из теоремы об алгоритме А.-Б. имеем $\rho^F(m) \leqslant \dfrac{m+1}{2}$.

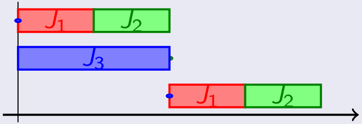
$\rho^F(2) = 3/2$

$$P = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$



$\rho^F(3) = 2$

$$P = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 2 \\ 1 & 1 & 0 \end{pmatrix}$$



Не всё так просто

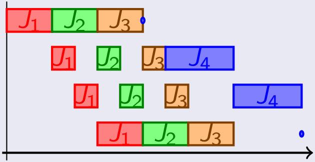
Рассмотрим Flow shop с $m = 4$. Тут следует различать обычную постановку и перестановочную!

Не всё так просто

Рассмотрим Flow shop с $m = 4$. Тут следует различать обычную постановку и перестановочную!

$\rho^{F|prmu}(4) \geqslant 13/6$

$$P = \begin{pmatrix} 2 & 2 & 2 & 0 \\ 1 & 1 & 1 & 3 \\ 1 & 1 & 1 & 3 \\ 2 & 2 & 2 & 0 \end{pmatrix}$$

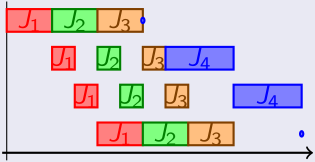


Не всё так просто

Рассмотрим Flow shop с $m = 4$. Тут следует различать обычную постановку и перестановочную!

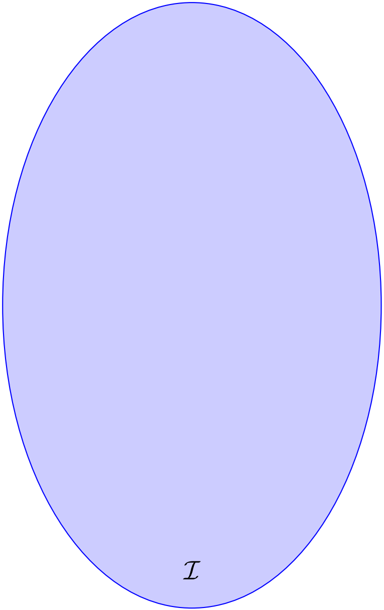
$\rho^{F|prmu}(4) \geqslant 13/6$

$$P = \begin{pmatrix} 2 & 2 & 2 & 0 \\ 1 & 1 & 1 & 3 \\ 1 & 1 & 1 & 3 \\ 2 & 2 & 2 & 0 \end{pmatrix}$$



$\rho^{F}(4) \geqslant 67/32$

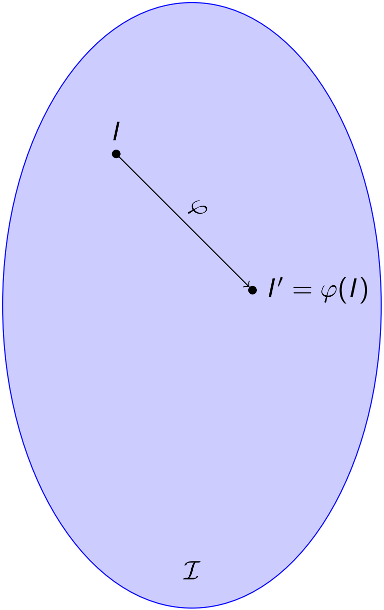
$$P = \begin{pmatrix} 9 & 13 & 10 & 0 \\ 3 & 3 & 3 & 23 \\ 9 & 8 & 9 & 6 \\ 11 & 8 & 10 & 3 \end{pmatrix}$$



$\mathcal{I}$ — множество **примеров**, для которых мы хотим доказать существование решения, для которого выполняется свойство $\mathcal{P}$:

$$\forall I \in \mathcal{I} \exists S(I) \mid \text{ выполняется } \mathcal{P}(S(I)).$$

Общая идея исследования



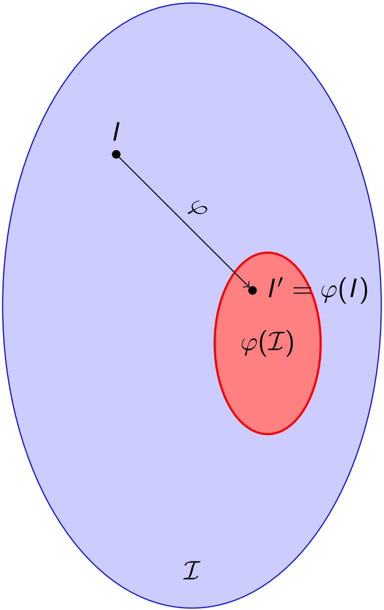
$\mathcal{I}$ — множество **примеров**, для которых мы хотим доказать существование решения, для которого выполняется свойство $\mathcal{P}$:

$\forall I \in \mathcal{I} \exists S(I) \mid$ выполняется $\mathcal{P}(S(I))$.

Преобразование $\varphi : \mathcal{I} \to \mathcal{I}$:

$$\forall I \in \mathcal{I}\ \mathcal{P}(\varphi(I)) \Rightarrow \mathcal{P}(I).$$

Общая идея исследования



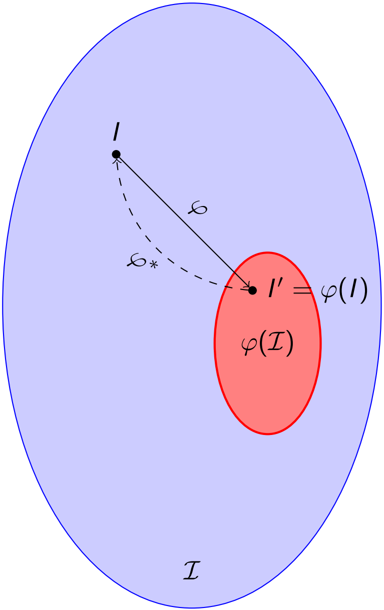
$\mathcal{I}$ — множество **примеров**, для которых мы хотим доказать существование решения, для которого выполняется свойство $\mathcal{P}$:

$\forall I \in \mathcal{I} \exists S(I) \mid$ выполняется $\mathcal{P}(S(I))$.

Преобразование $\varphi : \mathcal{I} \to \mathcal{I}$:

$$\forall I \in \mathcal{I}\ \mathcal{P}(\varphi(I)) \Rightarrow \mathcal{P}(I).$$

Общая идея исследования



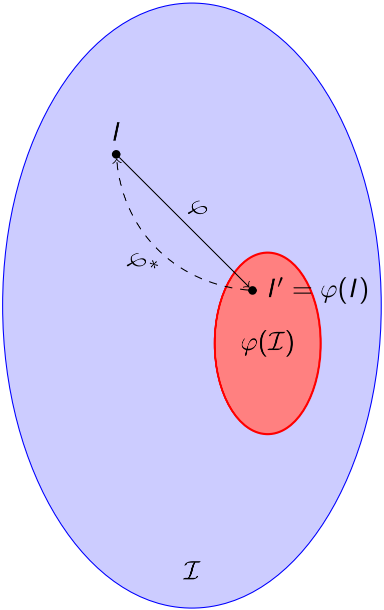
$\mathcal{I}$ — множество **примеров**, для которых мы хотим доказать существование решения, для которого выполняется свойство $\mathcal{P}$:

$$\forall I \in \mathcal{I} \exists S(I) \mid \text{ выполняется } \mathcal{P}(S(I)).$$

Обратное преобразование решения $\varphi^* : S(I') \to S(I)$:

$$\forall I \in \mathcal{I}\, \mathcal{P}(S(\varphi(I))) \Rightarrow \mathcal{P}(\varphi^*(S(\varphi(I)))).$$

Общая идея исследования



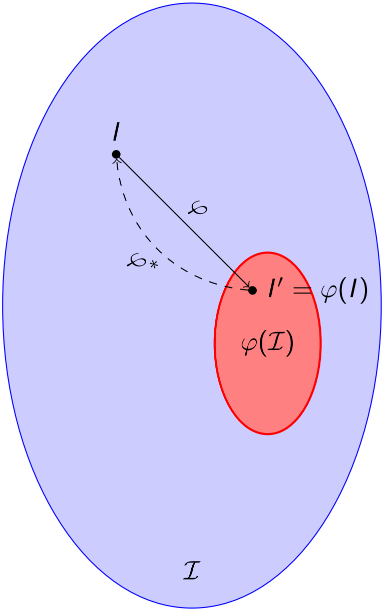
$\mathcal{I}$ — множество **примеров**, для которых мы хотим доказать существование решения, для которого выполняется свойство $\mathcal{P}$:

$\forall I \in \mathcal{I} \exists S(I) \mid$ выполняется $\mathcal{P}(S(I))$.

Обратное преобразование решения $\varphi^* : S(I') \to S(I)$:

$\forall I \in \mathcal{I}\, \mathcal{P}(S(\varphi(I))) \Rightarrow \mathcal{P}(\varphi^*(S(\varphi(I))))$.

I



$\mathcal{I}$ — множество **примеров**, для которых мы хотим доказать существование решения, для которого выполняется свойство $\mathcal{P}$:

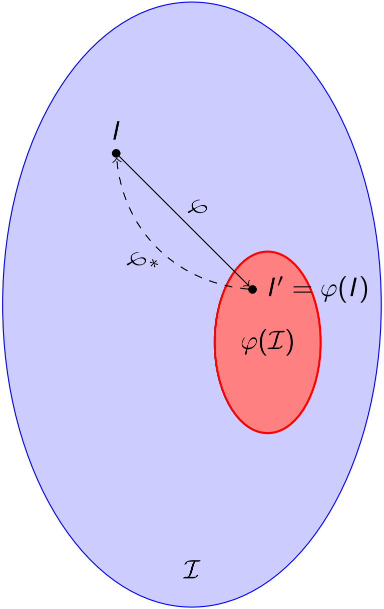
$$\forall I \in \mathcal{I} \exists S(I) \mid \text{ выполняется } \mathcal{P}(S(I)).$$

Обратное преобразование решения $\varphi^* : S(I') \to S(I)$:

$$\forall I \in \mathcal{I}\, \mathcal{P}(S(\varphi(I))) \Rightarrow \mathcal{P}(\varphi^*(S(\varphi(I)))).$$

$$I \xrightarrow{\varphi} I'$$

Общая идея исследования



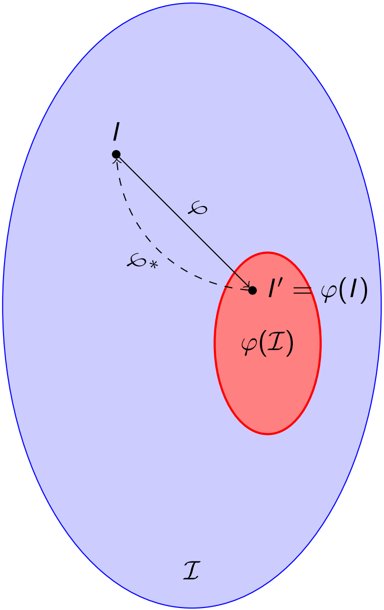
$\mathcal{I}$ — множество **примеров**, для которых мы хотим доказать существование решения, для которого выполняется свойство $\mathcal{P}$:

$\forall I \in \mathcal{I} \exists S(I) \mid$ выполняется $\mathcal{P}(S(I))$.

Обратное преобразование решения $\varphi^* : S(I') \to S(I)$:

$\forall I \in \mathcal{I}\, \mathcal{P}(S(\varphi(I))) \Rightarrow \mathcal{P}(\varphi^*(S(\varphi(I))))$.

$$\begin{tikzcd} I \arrow[r, "\varphi"] & I' \arrow[d, "\mathcal{A}"] \\ & S' \end{tikzcd}$$



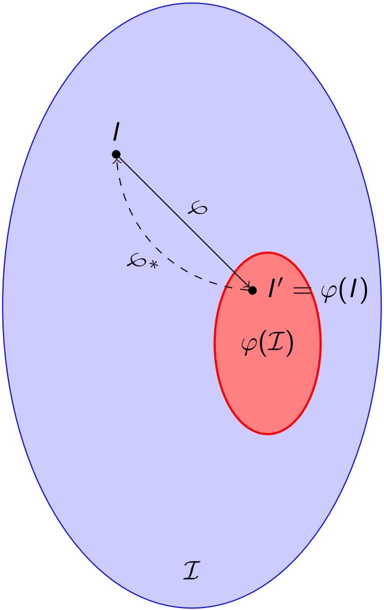
$\mathcal{I}$ — множество **примеров**, для которых мы хотим доказать существование решения, для которого выполняется свойство $\mathcal{P}$:

$\forall I \in \mathcal{I} \exists S(I) \mid$ выполняется $\mathcal{P}(S(I))$.

Обратное преобразование решения $\varphi^* : S(I') \to S(I)$:

$\forall I \in \mathcal{I}\, \mathcal{P}(S(\varphi(I))) \Rightarrow \mathcal{P}(\varphi^*(S(\varphi(I))))$.

$$\begin{tikzcd}
I \arrow[r, "\varphi"] & I' \arrow[d, "\mathcal{A}"] \\
S & S' \arrow[l, "\varphi^*"']
\end{tikzcd}$$



$\mathcal{I}$ — множество **примеров**, для которых мы хотим доказать существование решения, для которого выполняется свойство $\mathcal{P}$:

$$\forall I \in \mathcal{I} \exists S(I) \mid \text{ выполняется } \mathcal{P}(S(I)).$$

Обратное преобразование решения $\varphi^* : S(I') \to S(I)$:

$$\forall I \in \mathcal{I}\; \mathcal{P}(S(\varphi(I))) \Rightarrow \mathcal{P}(\varphi^*(S(\varphi(I)))).$$

$$\begin{tikzcd} I \arrow[r, "\varphi"] \arrow[d, rightsquigarrow] & I' \arrow[d, "\mathcal{A}"] \\ S & S' \arrow[l, "\varphi^*"'] \end{tikzcd}$$

Обратимые преобразования для локализации оптимумов

$$
\begin{tikzcd}
I \arrow[r, "\varphi"] \arrow[d, rightsquigarrow] & I' \arrow[d, "\mathcal{A}"] \\
S & S' \arrow[l, "\varphi^*"']
\end{tikzcd}
$$

Обратимые преобразования для локализации оптимумов

$$\begin{tikzcd} I \arrow[r, "\varphi"] \arrow[d, squiggly] & I' \arrow[d, "\mathcal{A}"] \\ S & S' \arrow[l, "\varphi^*"'] \end{tikzcd}$$

Пусть $\varphi : \mathcal{I} \to \mathcal{I}$ сохраняет нижнюю оценку LB,
а φ^* сохраняет значение целевой функции.

$$
\begin{tikzcd}
I \arrow[r, "\varphi"] \arrow[d, squiggly] & I' \arrow[d, "\mathcal{A}"] \\
S & S' \arrow[l, "\varphi^*"']
\end{tikzcd}
$$

Пусть $\varphi : \mathcal{I} \to \mathcal{I}$ сохраняет нижнюю оценку LB,
а φ^* сохраняет значение целевой функции.
Тогда

$$LB(I) = LB(\varphi(I)) \tag{1}$$

$$OPT(I) \leqslant OPT(\varphi(I)) \tag{2}$$

$$\frac{OPT(I)}{LB(I)} \leqslant \frac{OPT(\varphi(I))}{LB(\varphi(I))}$$

Следовательно, $\mathtt{OL}_{LB}(\mathcal{I}) = \mathtt{OL}_{LB}(\varphi(\mathcal{I}))$.

Склеивание работ

Определение

Склеиванием подмножества работ K в примере I называется такое следующее преобразование примера $I \to I'$:

$$\mathcal{J}(I') = \mathcal{J}(I) \setminus K \cup \{J_K\},$$

$$\forall i = 1, \ldots, m \quad p_K i = \sum_{J_j \in K} p_{ji}.$$

Склеивание работ

Определение

Склеиванием подмножества работ K в примере I называется такое следующее преобразование примера $I \to I'$:

$$\mathcal{J}(I') = \mathcal{J}(I) \setminus K \cup \{J_K\},$$

$$\forall i = 1, \ldots, m \quad p_K i = \sum_{J_j \in K} p_{ji}.$$

$$P = \begin{pmatrix} p_{11} & p_{21} & \cdots & p_{j-1,1} & p_{j1} & \cdots & p_{r1} & p_{r+1,1} & \cdots & p_{n1} \\ p_{12} & p_{22} & \cdots & p_{j-1,2} & p_{j2} & \cdots & p_{r2} & p_{r+1,2} & \cdots & p_{n2} \\ p_{13} & p_{23} & \cdots & p_{j-1,3} & p_{j3} & \cdots & p_{r3} & p_{r+1,3} & \cdots & p_{n3} \\ \vdots & \vdots & & & & & & & & \vdots \\ p_{1m} & p_{2m} & \cdots & p_{j-1,m} & p_{jm} & \cdots & p_{rm} & p_{r+1,m} & \cdots & p_{nm} \end{pmatrix}$$

Склеивание работ

Определение

Склеиванием подмножества работ K в примере I называется такое следующее преобразование примера $I \to I'$:

$$\mathcal{J}(I') = \mathcal{J}(I) \setminus K \cup \{J_K\},$$

$$\forall i = 1, \dots, m \quad p_K i = \sum_{J_j \in K} p_{ji}.$$

$$P = \begin{pmatrix} p_{11} & p_{21} & \cdots & p_{j-1,1} & p_{j1} & \cdots & p_{r1} & p_{r+1,1} & \cdots & p_{n1} \\ p_{12} & p_{22} & \cdots & p_{j-1,2} & p_{j2} & \cdots & p_{r2} & p_{r+1,2} & \cdots & p_{n2} \\ p_{13} & p_{23} & \cdots & p_{j-1,3} & p_{j3} & \cdots & p_{r3} & p_{r+1,3} & \cdots & p_{n3} \\ \vdots & \vdots & & & & & & & & \vdots \\ p_{1m} & p_{2m} & \cdots & p_{j-1,m} & p_{jm} & \cdots & p_{rm} & p_{r+1,m} & \cdots & p_{nm} \end{pmatrix}$$

Склеивание работ

Определение

Склеиванием подмножества работ K в примере I называется такое следующее преобразование примера $I \to I'$:

$$\mathcal{J}(I') = \mathcal{J}(I) \setminus K \cup \{J_K\},$$

$$\forall i = 1, \ldots, m \quad p_K i = \sum_{J_j \in K} p_{ji}.$$

$$P = \begin{pmatrix} p_{11} & p_{21} & \cdots & p_{j-1,1} & \boxed{\begin{matrix} \cdots \\ \cdots \\ \cdots \\ \\ \cdots \end{matrix}} & p_{r+1,1} & \cdots & p_{n1} \\ p_{12} & p_{22} & \cdots & p_{j-1,2} & & p_{r+1,2} & \cdots & p_{n2} \\ p_{13} & p_{23} & \cdots & p_{j-1,3} & & p_{r+1,3} & \cdots & p_{n3} \\ \vdots & \vdots & & & & & & \vdots \\ p_{1m} & p_{2m} & \cdots & p_{j-1,m} & & p_{r+1,m} & \cdots & p_{nm} \end{pmatrix}$$

Склеивание работ

Определение

Склеиванием подмножества работ K в примере I называется такое следующее преобразование примера $I \to I'$:

$$\mathcal{J}(I') = \mathcal{J}(I) \setminus K \cup \{J_K\},$$

$$\forall i = 1, \ldots, m \quad p_K i = \sum_{J_j \in K} p_{ji}.$$

$$P = \begin{pmatrix} p_{11} & p_{21} & \cdots & p_{j-1,1} & \sum_{k=j}^{r} p_{k1} & p_{r+1,1} & \cdots & p_{n1} \\ p_{12} & p_{22} & \cdots & p_{j-1,2} & \sum_{k=j}^{r} p_{k2} & p_{r+1,2} & \cdots & p_{n2} \\ p_{13} & p_{23} & \cdots & p_{j-1,3} & \sum_{k=j}^{r} p_{k3} & p_{r+1,3} & \cdots & p_{n3} \\ \vdots & \vdots & & & & & & \vdots \\ p_{1m} & p_{2m} & \cdots & p_{j-1,m} & \sum_{k=j}^{r} p_{km} & p_{r+1,m} & \cdots & p_{nm} \end{pmatrix}$$

Склеивание работ

Определение

Склеиванием подмножества работ K в примере I называется такое следующее преобразование примера $I \to I'$:

$$\mathcal{J}(I') = \mathcal{J}(I) \setminus K \cup \{J_K\},$$

$$\forall i = 1, \dots, m \quad p_K i = \sum_{J_j \in K} p_{ji}.$$

$$P = \begin{pmatrix} p_{11} & p_{21} & \cdots & p_{j-1,1} & \sum_{k=j}^{r} p_{k1} & p_{r+1,1} & \cdots & p_{n1} \\ p_{12} & p_{22} & \cdots & p_{j-1,2} & \sum_{k=j}^{r} p_{k2} & p_{r+1,2} & \cdots & p_{n2} \\ p_{13} & p_{23} & \cdots & p_{j-1,3} & \sum_{k=j}^{r} p_{k3} & p_{r+1,3} & \cdots & p_{n3} \\ \vdots & \vdots & & & & & & \vdots \\ p_{1m} & p_{2m} & \cdots & p_{j-1,m} & \sum_{k=j}^{r} p_{km} & p_{r+1,m} & \cdots & p_{nm} \end{pmatrix}$$

$$\bar{C}(I') = \bar{C}(I) \iff \sum_{J_j \in K} d_j \leqslant \bar{C}.$$

Теорема о склеивании

Теорема

Любой пример I задачи с m машинами можно с помощью серии склеиваний работ преобразовать в I' так, что $\bar{C}(I') = \bar{C}(I)$ и пример I' содержит не более $2m - 1$ работы.

Теорема о склеивании

Теорема

Любой пример I задачи с m машинами можно с помощью серии склеиваний работ преобразовать в I' так, что $\bar{C}(I') = \bar{C}(I)$ и пример I' содержит не более $2m-1$ работы.

Доказательство

Достаточно доказать, что можно сгруппировать величины $d_1, \ldots, d_n$ в не более чем $2m-1$ группу так, что в каждой группе сумма элементов не превосходит $\bar{C}$.

$$\sum_{j=1}^{n} d_j = \sum_{i=1} \ell_i \leqslant m\bar{C}.$$

Величину будем называть большой, если её длина строго больше $\bar{C}/2$.

Доказательство (продолжение)

Доказательство

$$\sum_{j=1}^{n} d_j = \sum_{i=1} \ell_i \leqslant m\bar{C}.$$

Величину будем называть большой, если её длина строго больше $\bar{C}/2$.

Доказательство (продолжение)

Доказательство

$$\sum_{j=1}^{n} d_j = \sum_{i=1} \ell_i \leqslant m\bar{C}.$$

Величину будем называть большой, если её длина строго больше $\bar{C}/2$.

$$d_1 \qquad d_2 \qquad \cdots \qquad d_s \qquad d_{s+1} \qquad \cdots \qquad d_n$$

Доказательство (продолжение)

Доказательство

$$\sum_{j=1}^{n} d_j = \sum_{i=1} \ell_i \leqslant m\bar{C}.$$

Величину будем называть большой, если её длина строго больше $\bar{C}/2$.

$$\boxed{\boxed{d_1 \quad d_2 \quad \cdots \quad d_s} \quad d_{s+1}} \quad \cdots \quad d_n$$

Итерация: Пусть $\sum_{j=1}^{s} d_j \leqslant \bar{C}$, а $\sum_{j=1}^{s+1} d_j > \bar{C}$.

Доказательство (продолжение)

Доказательство

$$\sum_{j=1}^{n} d_j = \sum_{i=1} \ell_i \leqslant m\bar{C}.$$

Величину будем называть большой, если её длина строго больше $\bar{C}/2$.

$$\boxed{\boxed{d_1 \quad d_2 \quad \cdots \quad d_s} \quad d_{s+1}} \quad \cdots \quad d_n$$

Итерация: Пусть $\sum_{j=1}^{s} d_j \leqslant \bar{C}$, а $\sum_{j=1}^{s+1} d_j > \bar{C}$. Группируем $D_1 = \sum_{j=1}^{s} d_j$.

Доказательство (продолжение)

Доказательство

$$\sum_{j=1}^{n} d_j = \sum_{i=1} \ell_i \leqslant m\bar{C}.$$

Величину будем называть большой, если её длина строго больше $\bar{C}/2$.

$$\boxed{\boxed{d_1 \quad d_2 \quad \cdots \quad d_s} \quad d_{s+1}} \quad \cdots \quad d_n$$

Итерация: Пусть $\sum_{j=1}^{s} d_j \leqslant \bar{C}$, а $\sum_{j=1}^{s+1} d_j > \bar{C}$. Группируем $D_1 = \sum_{j=1}^{s} d_j$.
Поскольку $D_1 + d_{s+1} > \bar{C}$, по крайней мере одна из величин D_1 и d_{s+1} большая. Отложим её в множество B (изначально пустое). Заканчиваем, если величины закончились, либо в B набралось $2m - 2$ элемента.

Доказательство (продолжение)

Доказательство

$$\sum_{j=1}^{n} d_j = \sum_{i=1} \ell_i \leqslant m\bar{C}.$$

Величину будем называть большой, если её длина строго больше $\bar{C}/2$.

$$\boxed{\boxed{d_1 \quad d_2 \quad \cdots \quad d_s} \quad d_{s+1}} \quad \cdots \quad d_n$$

Итерация: Пусть $\sum\limits_{j=1}^{s} d_j \leqslant \bar{C}$, а $\sum\limits_{j=1}^{s+1} d_j > \bar{C}$. Группируем $D_1 = \sum\limits_{j=1}^{s} d_j$.

Поскольку $D_1 + d_{s+1} > \bar{C}$, по крайней мере одна из величин D_1 и d_{s+1} большая. Отложим её в множество B (изначально пустое).

Заканчиваем, если величины закончились, либо в B набралось $2m-2$ элемента.

В первом случае осталось меньше $2m-2$ величин.

Доказательство (продолжение)

Доказательство

$$\sum_{j=1}^{n} d_j = \sum_{i=1} \ell_i \leqslant m\bar{C}.$$

Величину будем называть большой, если её длина строго больше $\bar{C}/2$.

$$\boxed{\boxed{d_1 \quad d_2 \quad \cdots \quad d_s} \quad d_{s+1}} \quad \cdots \quad d_n$$

Итерация: Пусть $\sum_{j=1}^{s} d_j \leqslant \bar{C}$, а $\sum_{j=1}^{s+1} d_j > \bar{C}$. Группируем $D_1 = \sum_{j=1}^{s} d_j$.

Поскольку $D_1 + d_{s+1} > \bar{C}$, по крайней мере одна из величин D_1 и d_{s+1} большая. Отложим её в множество B (изначально пустое).

Заканчиваем, если величины закончились, либо в B набралось $2m - 2$ элемента.

Во втором, сумма отложенных в B превышает $(m-1)\bar{C}$.

Следовательно, все остальные можно сгруппировать в одну группу.

Пример использования: задача $O2||C_{max}$

Теорема Гонзалеза-Сани (1976)

Для задачи $O2||C_{max}$ длина оптимального расписания всегда совпадает с $\bar{C}$. Такое расписание можно построить за линейное время.

Следствие

$$\rho^O(2) = 1.$$

Пример использования: задача $O2||C_{\max}$

Теорема Гонзалеза-Сани (1976)

Для задачи $O2||C_{\max}$ длина оптимального расписания всегда совпадает с $\bar{C}$. Такое расписание можно построить за линейное время.

Следствие

$$\rho^{O}(2) = 1.$$

- Проведем склеивание работ входа I так, как описано в доказательстве теоремы о склеивании. Получим вход I' с не более чем тремя работами, не изменив $\bar{C}$.

Пример использования: задача $O2||C_{\max}$

Теорема Гонзалеза-Сани (1976)

Для задачи $O2||C_{\max}$ длина оптимального расписания всегда совпадает с $\bar{C}$. Такое расписание можно построить за линейное время.

Следствие

$$\rho^O(2) = 1.$$

- Проведем склеивание работ входа I так, как описано в доказательстве теоремы о склеивании. Получим вход I' с не более чем тремя работами, не изменив $\bar{C}$.
- Обозначим операции первой (второй) машины через a_j (b_j), $j = 1, 2, 3$.

Пример использования: задача $O2||C_{\max}$

Теорема Гонзалеза-Сани (1976)

Для задачи $O2||C_{\max}$ длина оптимального расписания всегда совпадает с $\bar{C}$. Такое расписание можно построить за линейное время.

Следствие

$$\rho^O(2) = 1.$$

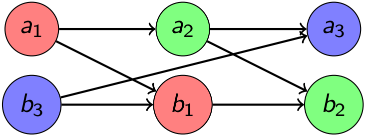
- Проведем склеивание работ входа I так, как описано в доказательстве теоремы о склеивании. Получим вход I' с не более чем тремя работами, не изменив $\bar{C}$.
- Обозначим операции первой (второй) машины через a_j (b_j), $j = 1, 2, 3$.
- Возможны два случая: все неравенства вида $a_j \geqslant b_j$ одновременно выполняются (или одновременно нарушаются) или из этих неравенств выполняется ровно 1 (или 2).

Пример использования: задача $O2||C_{\max}$ (продолжение)

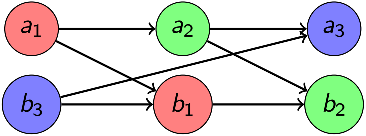
- **Случай I:** $\forall j = 1,2,3\ a_j \geqslant b_j$. Без ограничения общности $a_3 = \max\{a_j\}$.

Пример использования: задача $O2||C_{\max}$ (продолжение)

- **Случай I:** $\forall j = 1, 2, 3\ a_j \geqslant b_j$. Без ограничения общности $a_3 = \max\{a_j\}$.
- Построим расписание в соответствии со следующим порядком выполнения операций:



- **Случай I:** $\forall j = 1, 2, 3$ $a_j \geqslant b_j$. Без ограничения общности $a_3 = \max\{a_j\}$.
- Построим расписание в соответствии со следующим порядком выполнения операций:



- Поскольку $a_3 \geqslant a_2 \geqslant b_2$ и $a_2 + a_3 \geqslant a_2 + a_1 \geqslant b_1 + b_2$, длина такого расписания не превосходит $\bar{C}$.

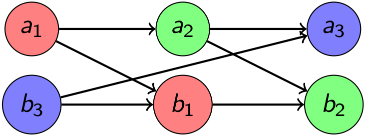
Пример использования: задача $O2||C_{max}$ (продолжение)

- **Случай II:** $a_1 \leqslant b_1$, $a_2 \geqslant b_2$, $a_3 \geqslant b_3$. Без ограничения общности $b_3 \geqslant b_2$.

Пример использования: задача $O2||C_{max}$ (продолжение)

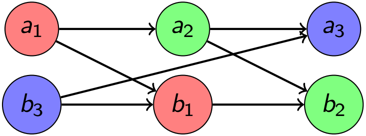
- **Случай II:** $a_1 \leqslant b_1$, $a_2 \geqslant b_2$, $a_3 \geqslant b_3$. Без ограничения общности $b_3 \geqslant b_2$.
- **Случай IIa:** $b_3 \geqslant a_1$.

- **Случай II:** $a_1 \leqslant b_1$, $a_2 \geqslant b_2$, $a_3 \geqslant b_3$. Без ограничения общности $b_3 \geqslant b_2$.
- **Случай IIa:** $b_3 \geqslant a_1$.
- Построим расписание в соответствии со следующим порядком выполнения операций:



Пример использования: задача $O2||C_{\max}$ (продолжение)

- **Случай II:** $a_1 \leqslant b_1$, $a_2 \geqslant b_2$, $a_3 \geqslant b_3$. Без ограничения общности $b_3 \geqslant b_2$.
- **Случай IIa:** $b_3 \geqslant a_1$.
- Построим расписание в соответствии со следующим порядком выполнения операций:

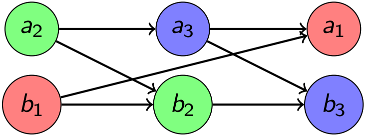


- Поскольку $a_1 \leqslant b_3$ и $a_3 \geqslant b_3 \geqslant b_2$, длина такого расписания не превосходит $\bar{C}$.

Пример использования: задача $O2||C_{max}$ (продолжение)

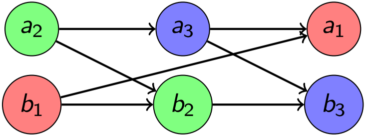
- **Случай II:** $a_1 \leqslant b_1$, $a_2 \geqslant b_2$, $a_3 \geqslant b_3$. Без ограничения общности $b_3 \geqslant b_2$.
- **Случай IIb:** $b_3 \leqslant a_1$.

- **Случай II:** $a_1 \leqslant b_1$, $a_2 \geqslant b_2$, $a_3 \geqslant b_3$. Без ограничения общности $b_3 \geqslant b_2$.
- **Случай IIb:** $b_3 \leqslant a_1$.
- Построим расписание в соответствии со следующим порядком выполнения операций:



Пример использования: задача $O2||C_{\max}$ (продолжение)

- **Случай II:** $a_1 \leqslant b_1$, $a_2 \geqslant b_2$, $a_3 \geqslant b_3$. Без ограничения общности $b_3 \geqslant b_2$.
- **Случай IIb:** $b_3 \leqslant a_1$.
- Построим расписание в соответствии со следующим порядком выполнения операций:



- Поскольку $a_1 \geqslant b_3$ и $a_3 \geqslant b_3 \geqslant b_2$, длина такого расписания не превосходит $\bar{C}$.

Алгоритм де Верра́ (1989)

Алгоритм

Алгоритм де Верра́ (1989)

Алгоритм

1. Если существует $k | a_k + b_k = \bar{C}$, то можно построить тривиальное расписание длины $\bar{C}$:

Алгоритм де Веррá (1989)

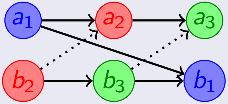
Алгоритм

1. Если существует $k|a_k + b_k = \bar{C}$, то можно построить тривиальное расписание длины $\bar{C}$;
2. Иначе: при необходимости, переименовать работы/машины так, чтобы выполнялись условия $a_1 \geqslant b_2$ и $b_1 \geqslant a_3$.

Алгоритм де Верра́ (1989)

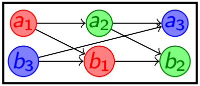
Алгоритм

1. Если существует $k | a_k + b_k = \bar{C}$, то можно построить тривиальное расписание длины $\bar{C}$;
2. Иначе: при необходимости, переименовать работы/машины так, чтобы выполнялись условия $a_1 \geqslant b_2$ и $b_1 \geqslant a_3$.
3. Построить расписание:



Вернемся к "неэффективному" варианту

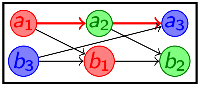
$$a_3 = \max_j \{a_j, b_j\}$$



S_1

Вернемся к “неэффективному” варианту

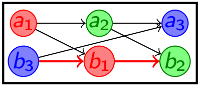
$$a_3 = \max_j\{a_j, b_j\}$$



S_1

Вернемся к "неэффективному" варианту

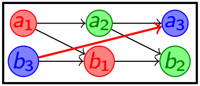
$$a_3 = \max_j \{a_j, b_j\}$$



S_1

Вернемся к "неэффективному" варианту

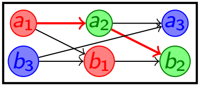
$$a_3 = \max_j \{a_j, b_j\}$$



S_1

Вернемся к “неэффективному” варианту

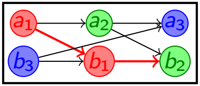
$$a_3 = \max_j \{a_j, b_j\}$$



S_1

Вернемся к “неэффективному” варианту

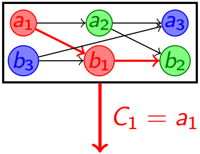
$$a_3 = \max_j\{a_j, b_j\}$$



S_1

Вернемся к “неэффективному” варианту

$$a_3 = \max_j \{a_j, b_j\}$$

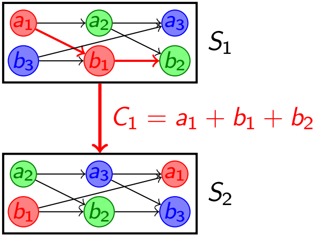


S_1

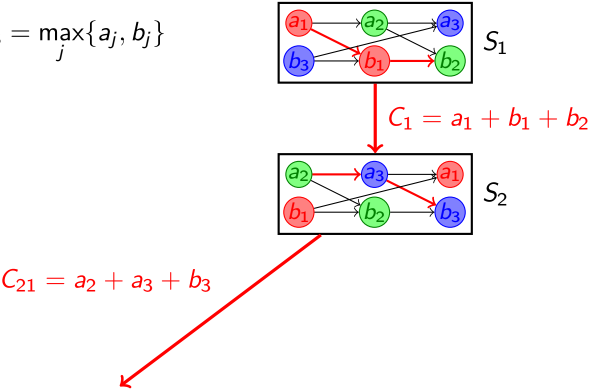
$C_1 = a_1 + b_1 + b_2$

Вернемся к "неэффективному" варианту

$$a_3 = \max_j\{a_j, b_j\}$$

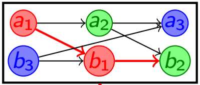


$a_3 = \max_j \{a_j, b_j\}$



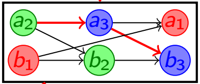
Вернемся к "неэффективному" варианту

$$a_3 = \max_j \{a_j, b_j\}$$



S_1

$C_1 = a_1 + b_1 + b_2$

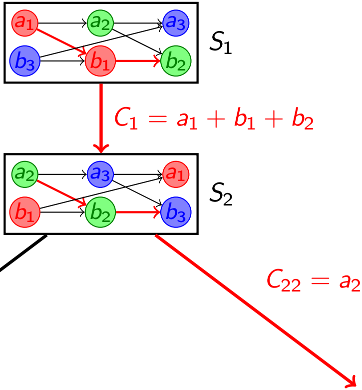


S_2

$C_{21} = a_2 + a_3 + b_3$

$$\boxed{C_1 + C_{21} = \ell_1 + \ell_2 \leqslant 2\bar{C}}$$

$$a_3 = \max_j \{a_j, b_j\}$$



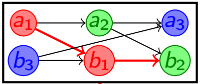
$$C_1 = a_1 + b_1 + b_2$$

$$C_{21} = a_2 + a_3 + b_3$$

$$C_{22} = a_2 + b_2 + b_3$$

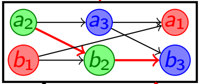
$$\boxed{C_1 + C_{21} = \ell_1 + \ell_2 \leqslant 2\bar{C}}$$

$a_3 = \max\limits_j \{a_j, b_j\}$



S_1

$C_1 = a_1 + b_1 + b_2$



S_2

$C_{21} = a_2 + a_3 + b_3$

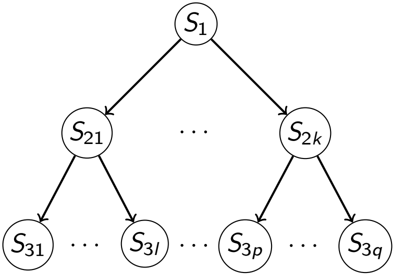
$C_{22} = a_2 + b_2 + b_3$

$\boxed{C_1 + C_{21} = \ell_1 + \ell_2 \leqslant 2\bar{C}}$

$\boxed{C_1 + C_{22} \leqslant \ell_1 + \ell_2 \leqslant 2\bar{C}}$

Дерево доказательства

- Вершины (кроме стоков) — *схемы расписаний*.
- Дуги соответствуют вариантам критических путей в схемах, из которых выходят.
- В стоках (висячих вершинах) — доказательства, что одно из построенных расписаний в пути от корня к стоку нам подходит.



Как автоматизировать процесс?

1. Достаточно рассматривать примеры с не более чем $2m - 1$ работой.
2. Достаточно рассматривать примеры, для которых $\bar{C} = 1$.
3. Нужно решить две проблемы:
   - Как выбирать следующую схему?
   - Как доказывать, что дальше ветвить не нужно?

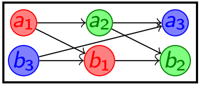
Как автоматизировать процесс?

1. Достаточно рассматривать примеры с не более чем $2m - 1$ работой.
2. Достаточно рассматривать примеры, для которых $\bar{C} = 1$.
3. Нужно решить две проблемы:
   - Как выбирать следующую схему?
   - Как доказывать, что дальше ветвить не нужно?

Основные идеи

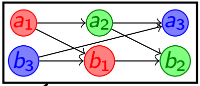
1. Рассматривать ветвление как "разбиение" множества примеров. Тогда каждая вершина соответствует подмножеству примеров.
2. В рассматриваемом подмножестве найти критический пример (наихудший по длине расписания для рассматриваемого подслучая).
3. Если длина расписания для критического примера нас устраивает, то мы получили доказательство для этого подслучая.
4. Иначе выбрать следующую схему так, чтобы она наилучшим образом подходила для найденного критического примера.

Ещё раз на примере



S_1

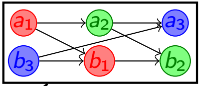
Ещё раз на примере



S_1

$$C_1 = a_1 + b_1 + b_2$$

Ещё раз на примере

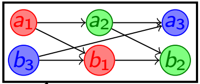


S_1

$$C_1 = a_1 + b_1 + b_2$$

$$a_1 = b_2 = 1$$

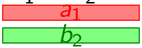
Ещё раз на примере



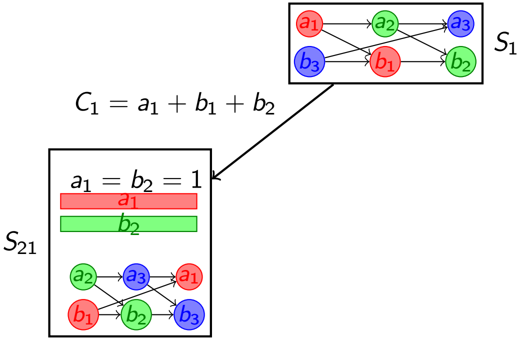
S_1

$C_1 = a_1 + b_1 + b_2$

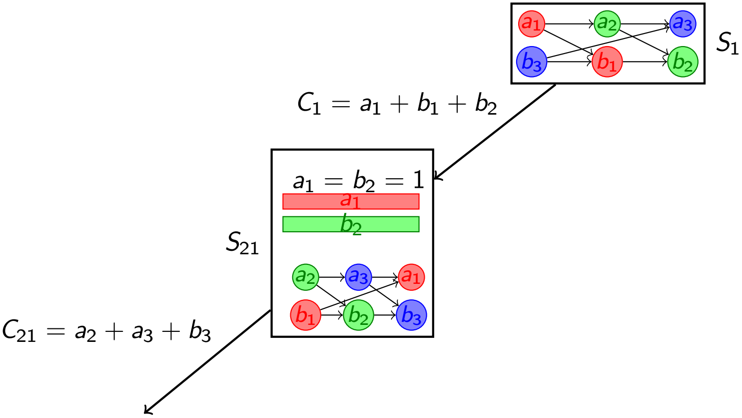
$a_1 = b_2 = 1$

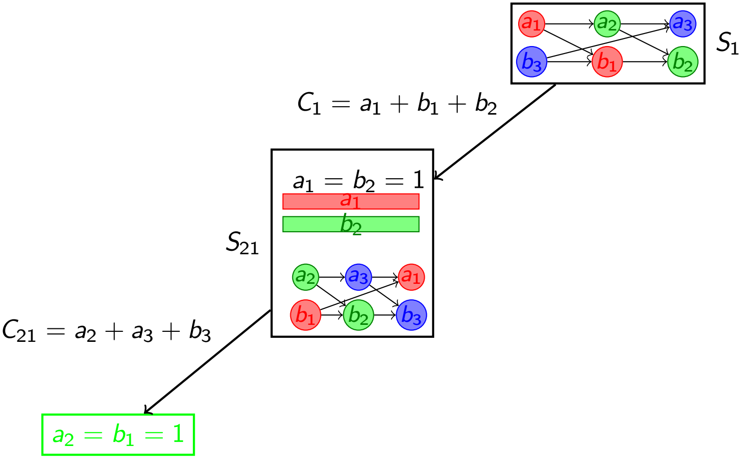


Ещё раз на примере

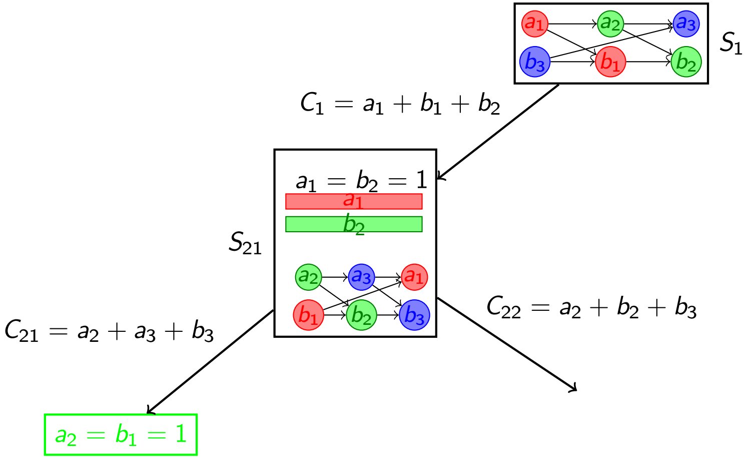


Ещё раз на примере

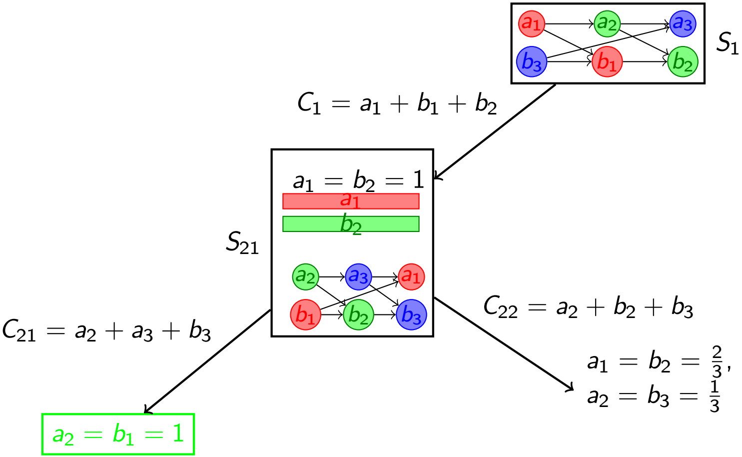



a_1 a_2 a_3 b_3 b_1 b_2
S_1
$C_1 = a_1 + b_1 + b_2$
$a_1 = b_2 = 1$
a_1
b_2
S_{21}
a_2 a_3 a_1 b_1 b_2 b_3
$C_{21} = a_2 + a_3 + b_3$
$a_2 = b_1 = 1$

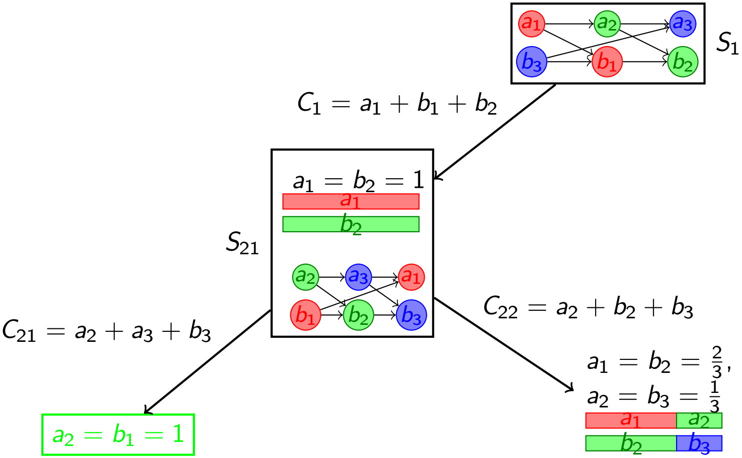
Ещё раз на примере



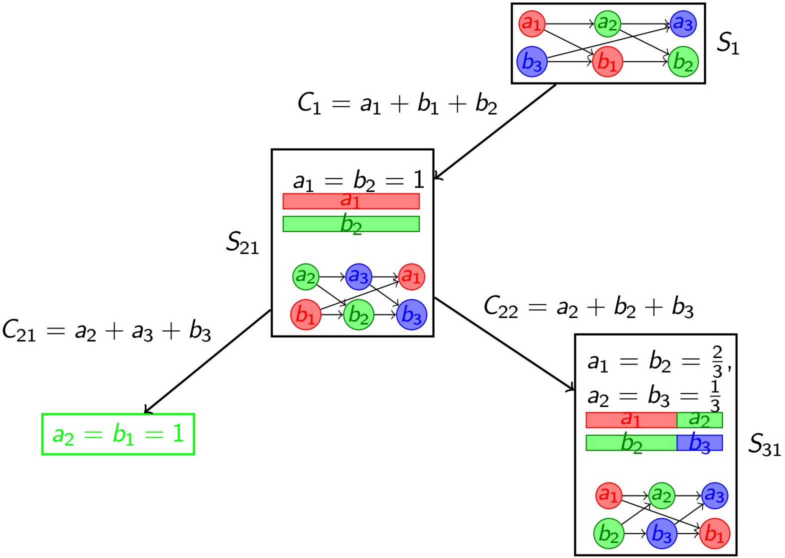
Ещё раз на примере



Ещё раз на примере



Ещё раз на примере



Нахождение критического примера

Задача линейного программирования

- Неотрицательные переменные: длительности операций (a_j, b_j) и вспомогательная ρ.
- Целевая функция: $\rho \to \max$.
- Ограничения:

$$
\begin{cases}
\ell_1 = a_1 + a_2 + a_3 \leqslant 1, \\
\ell_2 = b_1 + b_2 + b_3 \leqslant 1, \\
d_1 = a_1 + b_1 \leqslant 1, \\
d_2 = a_2 + b_2 \leqslant 1, \\
d_3 = a_3 + b_3 \leqslant 1, \\
C_1 = \sum\limits_{O \in P_1} p(O) \geqslant \rho, \\
\dots \\
C_k = \sum\limits_{O \in P_k} p(O) \geqslant \rho, \\
a_j, b_j, \rho \geqslant 0.
\end{cases}
$$

Дальнейшая автоматизация

Выбор следующей (улучшающей) схемы

Возможные варианты:

1. По критическому примеру строить оптимальное расписание, по расписанию восстанавливать последовательность операций (схему).
2. Хранить некоторый набор (pool) схем, выбирать из них наилучшую для найденного критического примера.

Оптимизация процесса

- Использование типов схем (шаблонов), из которых различные схемы генерируются применением перенумерации машин/работ.
- Уточнение примера, не ограничивающее общность (например, $a_3 = \max\{a_j, b_j\}$).
- Разбиение на подслучаи (по числу работ, по суммарной нагрузке,...)

Верификация доказательства

- Доказательство конструктивное: по дереву доказательства понятно, какие именно нужно строить расписания для конкретного примера, в зависимости от того, какие пути реализовался как критическе в уже построенных.
- Для висячих вершин (стоков) требуется теоретическое подтверждение (доказательство), что одно из построенных расписаний на пути к этой вершине имеет длину, не превосходящую нужной нам.
- Это можно было бы показать с помощью решения ЗЛП — только для висячих вершин.
- Можно проще: если для каждой висячей вершины хранить оптимальное решение двойственной ЗЛП! Достаточно будет только проверить, что это решение допустимое, и значение целевой функции не превосходит нужной границы.

Результаты

Задача	Интервал локализации
$O3\|\|C_{\max}$	$[\bar{C}, 4/3\bar{C}]$
$O3\|\nu = 2\|C_{\max}$	$[\bar{C}, 5/4\bar{C}]$
$F3\|pmtn\|C_{\max}$	$[\bar{C}, 9/5\bar{C}]$
$F4\|prmu, n \leq 4\|C_{\max}$	$[\bar{C}, 13/6\bar{C}]$

Результаты

Задача	Интервал локализации
$O3\|\|C_{\max}$	$[\bar{C}, 4/3\bar{C}]$
$O3\|\nu = 2\|C_{\max}$	$[\bar{C}, 5/4\bar{C}]$
$F3\|pmtn\|C_{\max}$	$[\bar{C}, 9/5\bar{C}]$
$F4\|prmu, n \leq 4\|C_{\max}$	$[\bar{C}, 13/6\bar{C}]$

- Доказано, что для $O||C_{\max}$ $C^*_{\max} \leqslant \Delta(I)/2$, где $\Delta(I) \doteq \sum \ell_i$ — суммарная нагрузка входа.

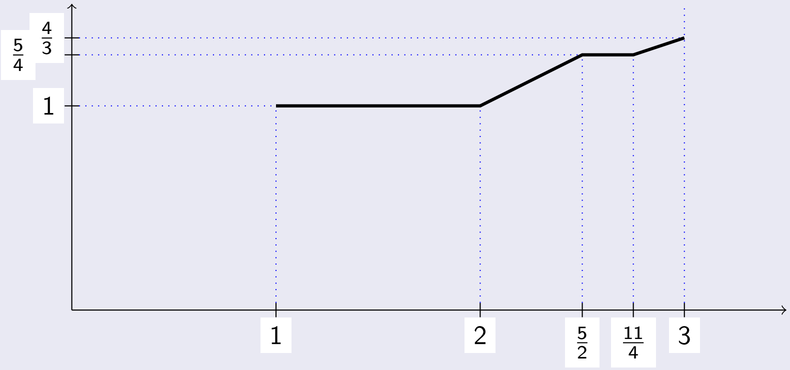
Результаты

Задача	Интервал локализации
$O3\|\|C_{\max}$	$[\bar{C}, 4/3\bar{C}]$
$O3\|\nu = 2\|C_{\max}$	$[\bar{C}, 5/4\bar{C}]$
$F3\|pmtn\|C_{\max}$	$[\bar{C}, 9/5\bar{C}]$
$F4\|prmu, n \leq 4\|C_{\max}$	$[\bar{C}, 13/6\bar{C}]$

- Доказано, что для $O||C_{\max}$ $C^*_{\max} \leqslant \Delta(I)/2$, где $\Delta(I) \doteq \sum \ell_i$ — суммарная нагрузка входа.
- Получен точный вид функции зависимости верхней границы оптимумов от суммарной нагрузки для $O3||C_{\max}$

Результаты

- Получен точный вид функции зависимости верхней границы оптимумов от суммарной нагрузки для $O3||C_{\max}$



Некоторые открытые вопросы

- Локализация оптимумов для $O4||C_{\max}$ (неизвестно, существует ли пример с длиной больше $\frac{4}{3}\bar{C}$)

Некоторые открытые вопросы

- Локализация оптимумов для $O4||C_{\max}$ (неизвестно, существует ли пример с длиной больше $\frac{4}{3}\bar{C}$)
- Локализация оптимумов для $F4|prmu|C_{\max}$ (гипотеза: $[\bar{C}, \frac{13}{6}\bar{C}]$)

Некоторые открытые вопросы

- Локализация оптимумов для $O4||C_{\max}$ (неизвестно, существует ли пример с длиной больше $\frac{4}{3}\bar{C}$)
- Локализация оптимумов для $F4|prmu|C_{\max}$ (гипотеза: $[\bar{C}, \frac{13}{6}\bar{C}]$)
- Локализация оптимумов для $F4||C_{\max}$ (известно, что верхняя граница интервала не меньше $\frac{67}{32}$)

Некоторые открытые вопросы

- Локализация оптимумов для $O4||C_{\max}$ (неизвестно, существует ли пример с длиной больше $\frac{4}{3}\bar{C}$)
- Локализация оптимумов для $F4|prmu|C_{\max}$ (гипотеза: $[\bar{C}, \frac{13}{6}\bar{C}]$)
- Локализация оптимумов для $F4||C_{\max}$ (известно, что верхняя граница интервала не меньше $\frac{67}{32}$)
- Гипотеза Свириденко: для $O||C_{\max}$ оптимум не превосходит $\bar{C} + p_{\max}$. Подтвердить для малых значений m.

Ну вот и все!

СПАСИБО ЗА ВНИМАНИЕ!

СПАСИБО ЗА ВНИМАНИЕ!

Вопросы?