# Линейное программирование

Процедура отделимости

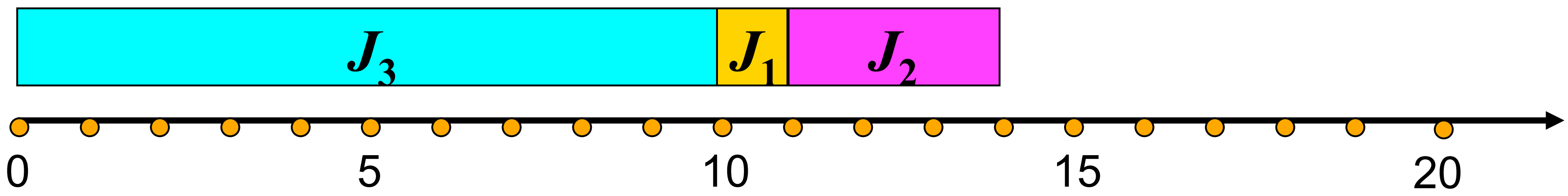
# $1|r_j|\Sigma C_j$

- Одна машина.
- $\boldsymbol{J} = \{1,\dots, n\}$ – работы.
- $p_j \geq 0$ – длительность работы $j$.
- $r_j \geq 0$ – время поступления работы $j$.
- $C_j(\sigma)$ – момент завершения работы в $\sigma$.
- Прерывания запрещены.
- Машина обслуживает не более одной работы одновременно.

# Пример

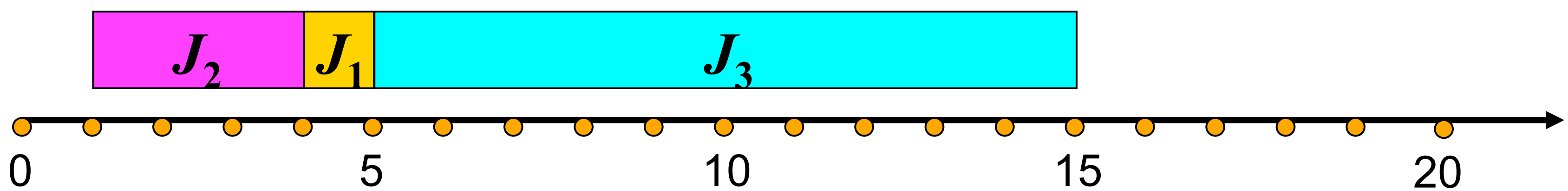
$\sigma_1$

$$C_1(\sigma_1) + C_2(\sigma_1) + C_3(\sigma_1) = 35$$




$\sigma_2$

$$C_1(\sigma_2) + C_2(\sigma_2) + C_3(\sigma_2) = 24$$

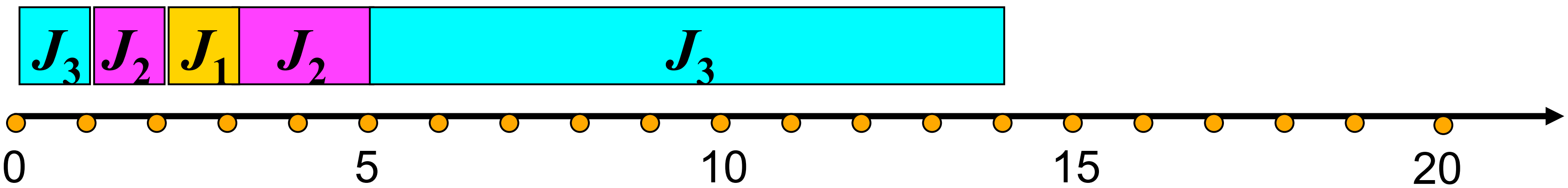



| | | |
|---|---|---|
| $J_1$ | $p_1 = 1$ | $r_1 = 2$ |
| $J_2$ | $p_2 = 3$ | $r_2 = 1$ |
| $J_3$ | $p_3 = 10$ | $r_3 = 0$ |

# Решение с прерываниями

$$C_1(\sigma^{pr}) + C_2(\sigma^{pr}) + C_3(\sigma^{pr}) = 22$$

$\sigma^{pr}$




| | | |
|---|---|---|
| $J_1$ | $p_1 = 1$ | $r_1 = 2$ |
| $J_2$ | $p_2 = 3$ | $r_2 = 1$ |
| $J_3$ | $p_3 = 10$ | $r_3 = 0$ |

# Нижняя оценка

- Пусть $C_j(\sigma^{\mathrm{pr}})$ — время завершения работы $j$ в оптимальном расписании с прерываниями.
- Пусть OPT — сумма моментов завершения работ в оптимальном расписании без прерываний.
- Тогда

$$\sum_{j=1}^{n} C_j\left(\sigma^{\mathrm{pr}}\right) \le \mathrm{OPT}.$$

# Алгоритм «Перестройка расписания с прерываниями»

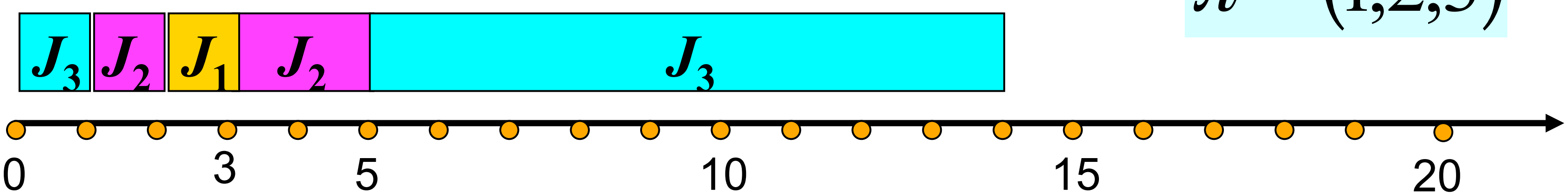
1. Найти оптимальное расписание $\sigma^{pr}$ с прерываниями.
2. Упорядочить работы по моментам их завершения в расписании $\sigma^{pr}$. Обозначим полученную перестановку через $\pi$.
3. Построить раннее расписание $\sigma$ относительно перестановки $\pi$.
4. **Output** ($\sigma$)

# Перестройка расписания с прерываниями

$$C_1(\sigma^{pr}) + C_2(\sigma^{pr}) + C_3(\sigma^{pr}) = 22$$

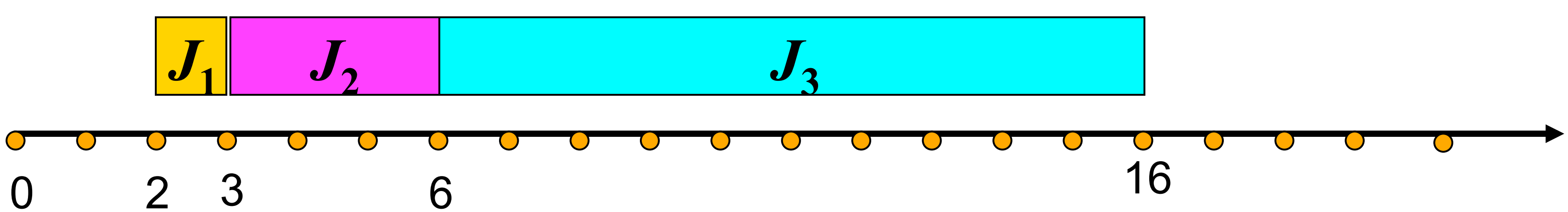
$\sigma^{pr}$

$$\pi = (1,2,3)$$




$$C_1(\sigma) + C_2(\sigma) + C_3(\sigma) = 25$$

$\sigma$



# Оценка на сдвижку работ

**Лемма 11.1**

Занумеруем работы в порядке их появления в расписании $\sigma^{pr}$, т.е. $C_1(\sigma^{pr}) \leq C_2(\sigma^{pr}) \leq \ldots \leq C_n(\sigma^{pr})$. Для каждой работы $j = 1,\ldots,n,$ $C_j(\sigma) \leq 2C_j(\sigma^{pr})$.

# Доказательство

- Оценим момент окончания работы $j$ в расписании $\sigma^{\mathrm{pr}}$. Поскольку работы $1,\dots,j-1$ заканчиваются в $\sigma^{\mathrm{pr}}$ до завершения работы $j$, то

$$C_j\left(\sigma^{\mathrm{pr}}\right) \geq \max_{k=1,\dots,j} r_k, \quad C_j\left(\sigma^{\mathrm{pr}}\right) \geq \sum_{k=1}^{j} p_k.$$

- Так как σ — раннее расписание, то машина не простаивает в интервале $\left[\max_{k=1,\dots,j} r_k, C_j(\sigma)\right]$.
- Следовательно, длина этого интервала не превосходит $\sum_{k=1}^{j} p_k$.

$$C_j(\sigma) \leq \max_{k=1,\dots,j} r_k + \sum_{k=1}^{j} p_k \leq 2C_j\left(\sigma^{\mathrm{pr}}\right).$$

# 2-приближение

**Теорема 11.2**

Алгоритм «Перестройка расписания с прерываниями» является 2-приближенным алгоритмом для задачи $1|r_j|\Sigma C_j$.

$$\sum_{j=1}^{n} C_j(\sigma) \leq 2\sum_{j=1}^{n} C_j(\sigma^{\mathrm{pr}}) \leq 2\,\mathrm{OPT}.$$

# $1|r_j|\Sigma w_j C_j$

- Одна машина.
- $\boldsymbol{J} = \{1, \ldots, n\}$ – работы.
- $p_j \geq 0$ – длительность работы $j$.
- $r_j \geq 0$ – время поступления работы $j$.
- $w_j \geq 0$ – вес работы $j$.
- $C_j(\sigma)$ – момент завершения работы в σ.
- Прерывания запрещены.
- Машина обслуживает не более одной работы одновременно.

# Что нужно для алгоритма?

$$C_j\left(\sigma^{\mathrm{pr}}\right) \geq \max_{k=1,\ldots,j} r_k$$

$$C_j\left(\sigma^{\mathrm{pr}}\right) \geq \sum_{k=1}^{j} p_k$$

$$\sum_{j=1}^{n} C_j\left(\sigma^{\mathrm{pr}}\right) \leq \mathrm{OPT}$$

# Что нужно для алгоритма?

$$C_j\left(\sigma^{\mathrm{pr}}\right) \geq \max_{k=1,\dots,j} r_k \qquad C_j\left(\sigma^{\mathrm{pr}}\right) \geq \alpha \max_{k=1,\dots,j} r_k$$

$$C_j\left(\sigma^{\mathrm{pr}}\right) \geq \sum_{k=1}^{j} p_k \qquad C_j\left(\sigma^{\mathrm{pr}}\right) \geq \beta \sum_{k=1}^{j} p_k$$

$$\sum_{j=1}^{n} C_j\left(\sigma^{\mathrm{pr}}\right) \leq \mathrm{OPT} \qquad \sum_{j=1}^{n} C_j\left(\sigma^{*}\right) \leq \mathrm{OPT}$$

Рассмотреть задачу ЛП с переменными $C_j$ такую, что в любом допустимом решении первые два ограничения выполняются с точностью до мультипликативных констант α и β, а третье неравенство выполнено для оптимального решения ЛП.

# Переменные и ограничения

- $C_j$ — момент завершения работы $j$.
- $C_j \geq r_j + p_j$
- Рассмотрим $\sum_{j \in S} p_j C_j$ для произвольного $S \subseteq \boldsymbol{J}$.
- Сумма принимает минимальное значение, когда все работы в $S$ выполняются первыми с момента 0.
- Тогда $C_j(\sigma)$ для $j \in S$ равно $p_j$ + сумма длительностей всех работ, предшествующих $j$ в $\sigma$.
- $\sum_{j \in S} p_j C_j$ должна содержать слагаемое $p_j p_k$ для всех $j,k \in S$.

# Неравенство Кирана

$$\sum_{j\in S} p_j C_j = \sum_{j,k\in S:\, j\le k} p_j p_k =$$

$$= \frac{1}{2}\left(\sum_{j\in S} p_j\right)^2 + \frac{1}{2}\sum_{j\in S} p_j^2 \ge \frac{1}{2}\left(\sum_{j\in S} p_j\right)^2$$

# ЛП $(1|r_j|\Sigma w_j C_j)$

$$\text{minimize} \quad \sum_{j=1}^{n} w_j C_j$$

$$\text{s.t.} \qquad C_j \geq r_j + p_j, \qquad \forall j \in J,$$

$$\sum_{j \in S} p_j C_j \geq \frac{1}{2} \left( \sum_{j \in S} p_j \right)^2, \ \forall S \subseteq J.$$

# Алгоритм $1|r_j|\Sigma w_j C_j$

1. Найти оптимальное решение $\sigma^* = (C_1(\sigma^*), C_2(\sigma^*), \ldots, C_n(\sigma^*))$ задачи ЛП $(1|r_j|\Sigma w_j C_j)$.
2. Упорядочить работы по моментам их завершения в решении $\sigma^*$. Обозначим полученную перестановку через $\pi$.
3. Построить раннее расписание $\sigma$ относительно перестановки $\pi$.
4. **Output** $(\sigma)$

# 3-приближение

**Теорема 11.3**

Алгоритм $1|r_j|\Sigma w_jC_j$ является 3-приближенным алгоритмом для задачи $1|r_j|\Sigma w_jC_j$.

# Доказательство

$$\sum_{j=1}^{n} w_j C_j^* \le \mathrm{OPT}.$$

- Занумеруем работы в порядке их появления в расписании σ*, т.е. $C_1(\sigma^*) \le C_2(\sigma^*) \le \ldots \le C_n(\sigma^*)$.
- Так как σ — раннее расписание, то машина не простаивает в интервале $\left[\max_{k=1,\ldots,j} r_k, C_j(\sigma)\right]$.

$$C_j(\sigma) \le \max_{k=1,\ldots,j} r_k + \sum_{k=1}^{j} p_k$$

$$C_j(\sigma) \le \max_{k=1,\dots,j} r_k + \sum_{k=1}^{j} p_k$$

- Пусть $l \in \{1,\dots,j\}$: $r_l = \max_{k=1,\dots,j} r_k$.
- $C_j(\sigma^*) \ge C_l(\sigma^*)$ и $C_l(\sigma^*) \ge r_l \Rightarrow C_j(\sigma^*) \ge \max_{k=1,\dots,j} r_k$.
- Пусть $S=\{1,\dots,j\}$.
- Так как σ* допустимое решение ЛП $(1|r_j|\Sigma w_j C_j)$, то

$$\sum_{k\in S} p_k C_k(\sigma^*) \ge \frac{1}{2}\left(\sum_{k\in S} p_k\right)^2.$$

- Поскольку $C_1(\sigma^*) \le C_2(\sigma^*) \le \dots \le C_j(\sigma^*)$, то

$$C_j(\sigma^*)\sum_{k\in S} p_k \ge \sum_{k\in S} p_k C_k(\sigma^*) \ge \frac{1}{2}\left(\sum_{k\in S} p_k\right)^2.$$

$$C_j(\sigma^*) \ge \frac{1}{2}\sum_{k\in S} p_k.$$

# Доказательство

$$C_j(\sigma^*) \geq \max_{k=1,\dots,j} r_k \qquad C_j(\sigma^*) \geq \frac{1}{2} \sum_{k \in S} p_k .$$

$$C_j(\sigma) \leq \max_{k=1,\dots,j} r_k + \sum_{k=1}^{j} p_k \leq C_j(\sigma^*) + 2C_j(\sigma^*) = 3C_j(\sigma^*).$$

$$\sum_{j=1}^{n} w_j C_j(\sigma) \leq 3 \sum_{j=1}^{n} w_j C_j(\sigma^*) \leq 3\,\mathrm{OPT}.$$

# Как решать ЛП?

$$\text{minimize} \quad \sum_{j=1}^{n} w_j C_j$$

$$\text{s.t.} \quad C_j \geq r_j + p_j, \quad \forall j \in J \qquad (1),$$

$$\sum_{j \in S} p_j C_j \geq \frac{1}{2}\left(\sum_{j \in S} p_j\right)^2, \ \forall S \subseteq J \qquad (2).$$

# Метод эллипсоидов (грубо)

- Вход: $P = \{Cx \leq d\}$ (полноразмерный или пустой)

*1.* $N=2n((2n+1)\langle C\rangle + n\langle d\rangle - n^3)$, $k = 0$. ($N$ не зависит от числа ограничений)

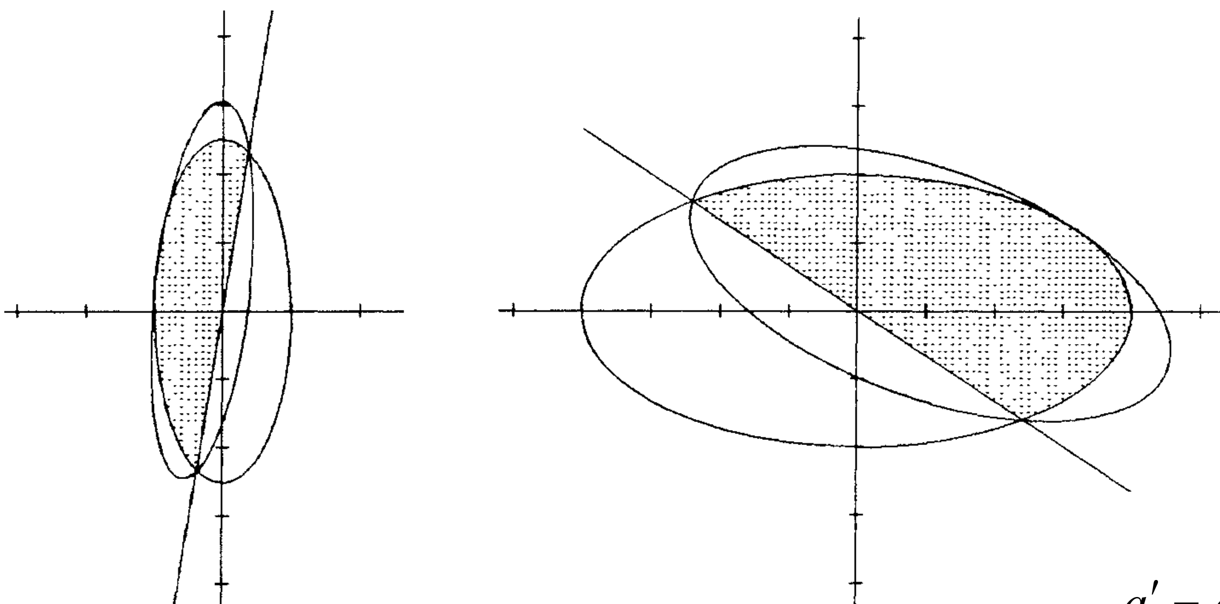
2. Выбрать большой эллипс $E_0(A_0,a_0)$, содержащий $P$.
3. Если $k = N$, то STOP! ($P$ — *пустой.*)
4. Если $a_k \in P$, то STOP! (*Решение найдено.*)
5. Если $a_k \notin P$, то найти неравенство, которое нарушается для $a_k$.

6. Построить $E_{\mathrm{k+1}}(A_{\mathrm{k+1}},a_{\mathrm{k+1}})$, go to 3.

# Лёвнер-Джон эллипсоид

$E = E(A,a) = \{x \in \mathbf{R}^n \mid (x-a)^T A^{-1}(x-a) \le 1\}$

$E'(A,a,c) = E(A,a) \cap \{x \in \mathbf{R}^n \mid c^T x \le c^T a\}$



**(a)** **(b)**

**Figure 3.3**

$$a' = a - \frac{1}{n+1} b, \quad \left( b = \frac{1}{\sqrt{c^T A c}} A c \right)$$

$$A' = \frac{n^2}{n^2-1} \left( A - \frac{2}{n+1} b b^T \right)$$

# Метод эллипсоидов (грубо)

- Вход: $P = \{Cx \le d\}$ (полноразмерный или пустой)

*1.* $N = 2n((2n+1)\langle C\rangle + n\langle d\rangle - n^3)$, $k = 0$. ($N$ не зависит от числа ограничений)

2. Выбрать большой эллипс $E_0(A_0, a_0)$, содержащий $P$.
3. Если $k = n$, то STOP! ($P$ — *пустой*.)
4. **Если $a_k \in P$, то STOP! (*Решение найдено.*)**
5. **Если $a_k \notin P$, то найти неравенство, которое нарушается для $a_k$.**

6. Построить $E_{k+1}(A_{k+1}, a_{k+1})$, go to 3.

**Нужна полиномиальная по времени процедура для шагов 4 и 5!**

# Как проверять неравенства

$$\sum_{j\in S} p_j C_j \geq \frac{1}{2}\left(\sum_{j\in S} p_j\right)^2, \ \forall S \subseteq J\,?$$

- Рассмотрим произвольное решение σ.
- Занумеруем переменные так, что $C_1(\sigma) \leq C_2(\sigma) \leq \ldots \leq C_n(\sigma)$.
- Пусть $S_1 = \{1\}$, $S_2 = \{1, 2\}, \ldots, S_n = \{1, \ldots, n\}$.
- Покажем, что достаточно проверить нарушения ограничений только для множеств $S_1, S_2, \ldots, S_n$.
- Если ни одно из этих ограничений нарушается, то не нарушается ни одно из ограничений ЛП.

# Процедура отделения

**Лемма 11.1**

Если ограничение (2) выполнено для всех множеств $S_1, S_2, \ldots, S_n$, то оно выполнено и для любого множества $S \subseteq \boldsymbol{J}$.

# Доказательство (1)

- Пусть $S \subseteq \boldsymbol{J}$ , на котором нарушается ограничение (2), т.е.

$$\sum_{j\in S} p_j C_j < \frac{1}{2}\left(\sum_{j\in S} p_j\right)^2.$$

- Покажем, что ограничение (2) также нарушается для некоторого множества $S_i$, $i$=1,…, $n$.

$$x = \sum_{j\in S} p_j C_j - \frac{1}{2}\left(\sum_{j\in S} p_j\right)^2$$

- Преобразуем множество $S$ в $S_i$ так, чтобы на каждом шаге значение $x$ уменьшалось

$$x = \sum_{j\in S} p_j C_j - \frac{1}{2}\left(\sum_{j\in S} p_j\right)^2$$

- Удаление работы $k$ уменьшает $x$, если

$$-p_k C_k + p_k \sum_{j\in S\setminus\{k\}} p_j + \tfrac{1}{2} p_k^2 < 0 \Leftrightarrow C_k > \sum_{j\in S\setminus\{k\}} p_j + \tfrac{1}{2} p_k.$$

- Добавление работы $k$ уменьшает $x$, если

$$p_k C_k - p_k \sum_{j\in S} p_j - \tfrac{1}{2} p_k^2 < 0 \Leftrightarrow C_k < \sum_{j\in S} p_j + \tfrac{1}{2} p_k.$$

# Удаление работ

- Пусть $l$ — работа с наибольшим индексом в $S$.
- Если $C_l > \sum_{j \in S \setminus \{l\}} p_j + \frac{1}{2} p_l$, то удалим $l$ из $S$.
- Ограничение (2) нарушается и на $S \setminus \{l\}$.
- Продолжим удаление работ, пока не получим множество $S'$ такое, что для работы $l$ с наибольшим индексом выполнено

$$C_l \leq \sum_{j \in S \setminus \{l\}} p_j + \frac{1}{2} p_l.$$

# Добавление работ

- Предположим, что $S' \neq S_l = \{1, \ldots, l\}$.
- Пусть $k < l$ и $k \notin S_l$ .
- Имеем $$C_k \le C_l \le \sum_{j \in S \setminus \{l\}} p_j + \tfrac{1}{2} p_l < \sum_{j \in S} p_j < \sum_{j \in S} p_j + \tfrac{1}{2} p_k.$$
- Добавление $k$ к $S'$ лишь увеличит $$x = \sum_{j \in S} p_j C_j - \frac{1}{2}\left(\sum_{j \in S} p_j\right)^2.$$
- Добавляя к $S'$ все $k$ ($k < l$ и $k \notin S_l$) получим, что на множестве $S_l$ не выполняются ограничения (2).

# Метод эллипсоидов (1)

- Алгоритм находит эллипсоид в $\mathrm{R}^n$, который содержит все крайние точки ЛП($1|r_j|\Sigma w_j C_j$).
- Пусть $\check{C}$ — центр эллипсоида.
- Алгоритм запускает процедуру отделения с решением $\check{C}$.
- Если $\check{C}$ допустимо, то алгоритм создает ограничение $\Sigma w_j C_j \le \Sigma w_j \check{C}_j$, так как оптимальное решение должно иметь значение не больше чем значение допустимого решения $\check{C}$.
- Если $\check{C}$ недопустимо, то процедура отделения находит ограничение, которое нарушается на решении $\check{C}$.
- В любом случае к ограничениям добавляется гиперплоскость, которая проходит через $\check{C}$ и делит эллипсоид на две части.

# Метод эллипсоидов (2)

- Алгоритм находит новый эллипсоид в $R^n$, который содержит соответствующую половинку предыдущего эллипсоида и в качестве нового решения рассматривает центр нового эллипсоида.
- Этот процесс продолжается пока эллипсоид не уменьшится до размера, в котором будет содержаться только одна крайняя точка, которая и будет оптимальным решением.

# Практика

- Рассмотрим задачу $1||\Sigma w_j C_j$ ,которая отличается от задачи $1|r_j|\Sigma w_j C_j$, рассмотренной в лекции тем, что все работы поступают на обслуживание одновременно в момент 0.
- Пусть работы занумерованы в следующем порядке

$$\frac{w_1}{p_1} \geq \frac{w_2}{p_2} \geq \cdots \geq \frac{w_n}{p_n}.$$

- Докажите, что расписание работ, в котором работы выполняются в указанном выше порядке является оптимальным.