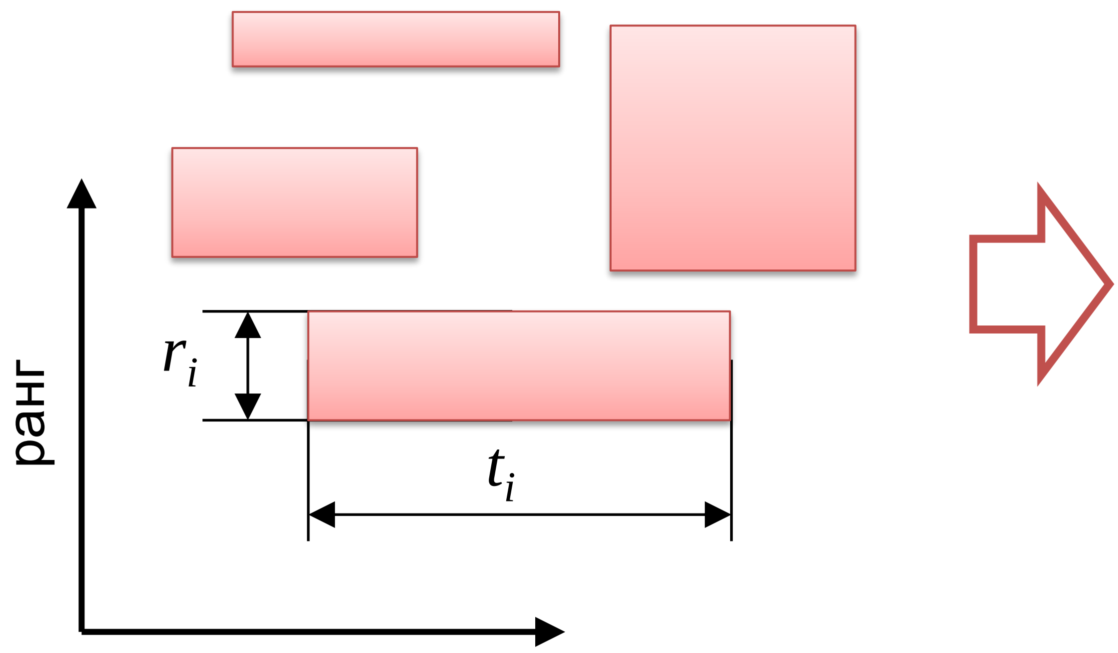
Лекция 2. Мультипрограммный режим обслуживания набора параллельных задач

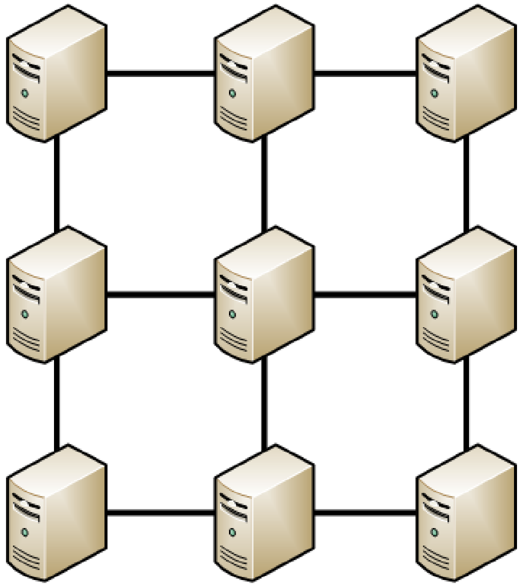
Пазников Алексей Александрович

Ассистент Кафедры вычислительных систем
Сибирский государственный университет
телекоммуникаций и информатики

http://cpct.sibsutis.ru/~apaznikov

Обработка набора задач





Распределённая ВС

Постановка задачи

Имеется **распределённая ВС**, состоящая из N ЭМ

На вход ВС поступает m **задач**.

Каждая задача $i \in \{1, 2, \ldots, m\} = J$ характеризуется:

r_i – ранг параллельной задачи,

t_i – время решения,

Требуется построит **расписание** S решения параллельных задач на ЭМ с целью минимизации суммарного времени их решения

$$S = (\tau_1, \tau_2, \ldots, \tau_m; x_{11}, x_{12}, \ldots, x_{1r_1}; x_{21}, x_{22}, \ldots, x_{2r_2}, \ldots, x_{mr_m})$$

τ_j – время начала решения задачи j

$x_{ij} \in \{1, 2, \ldots, N\}$ – номер ЭМ, на которую направлена ветвь j i-й задачи

Расписание

$J(t) = \{j \in J \mid \tau_j \le t \le \tau_j + t_j\}$ – множество задач, решаемых на распределённой ВС в момент времени t

S – **допустимое**, если оно удовлетворяет условиям:

1. В любой момент времени на ресурсах ВС решается не более n ветвей параллельных задач:

$$\sum_{j \in J(t)} r_j \le n, \quad \forall t \in \Re$$

2. Ветви параллельных задач решаются на разных ЭМ

$$\prod_{j \in J(t)} \prod_{j' \in J(t) \setminus \{j\}} (x_{ji} - x_{j'i'}) \ne 0, \quad \forall t \in R, \quad i = 1, 2, \ldots, r_j,\ i' = 1, 2, \ldots, r_{j'}$$

Ω - множество допустимых решений.

Расписание

3. ЭМ j, на которую назначена задача i

$$x_{ji} \in C,\ j = 1,2,\ldots,m,\ i = 1,2,\ldots,r_j$$

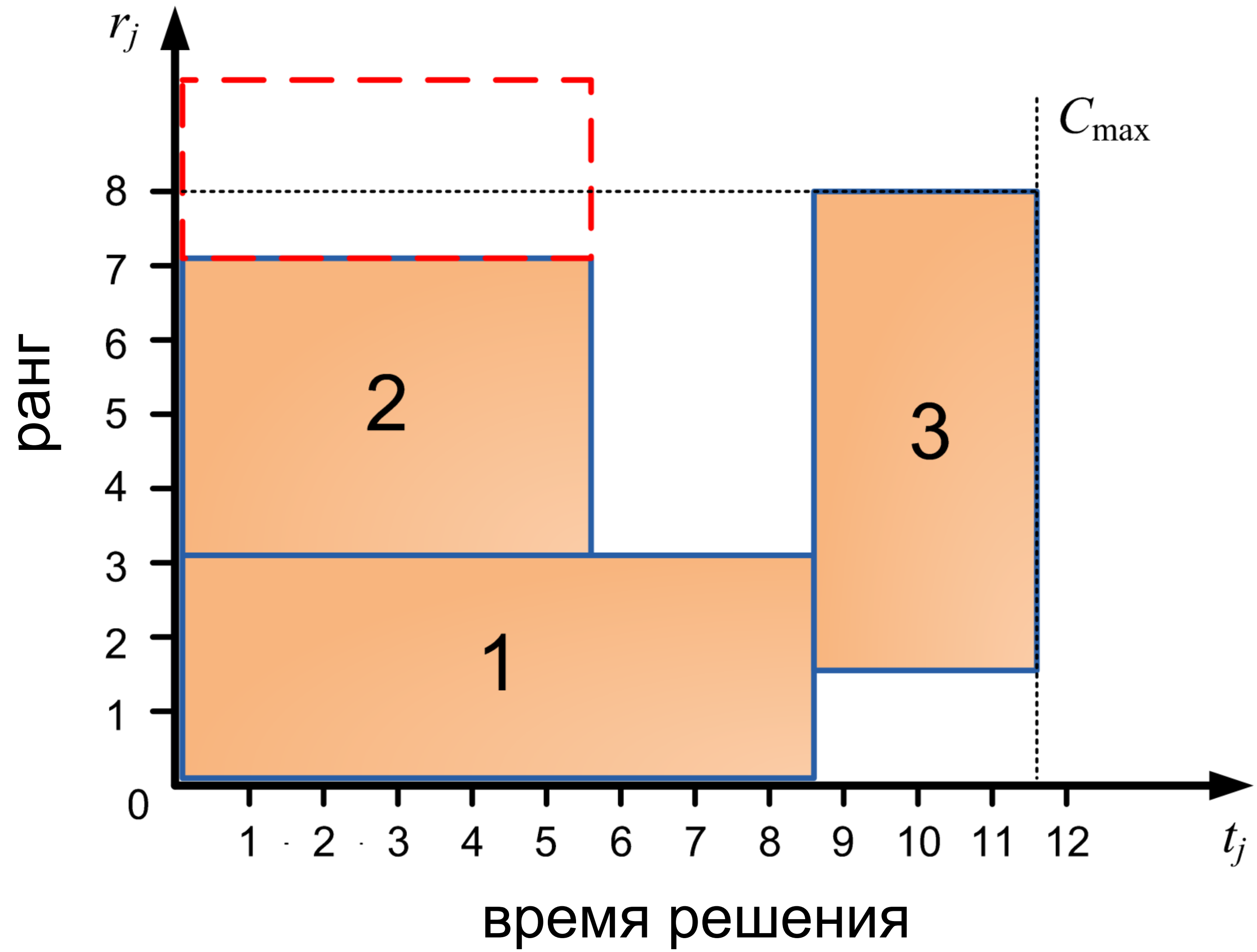
4. Время решения задачи j

$$\tau_j \in \Re,\ j = 1,2,\ldots,m.$$

В качестве **показателя эффективности** расписания будем использовать $C_{\max}$ – максимальное из времён завершения решения задач:

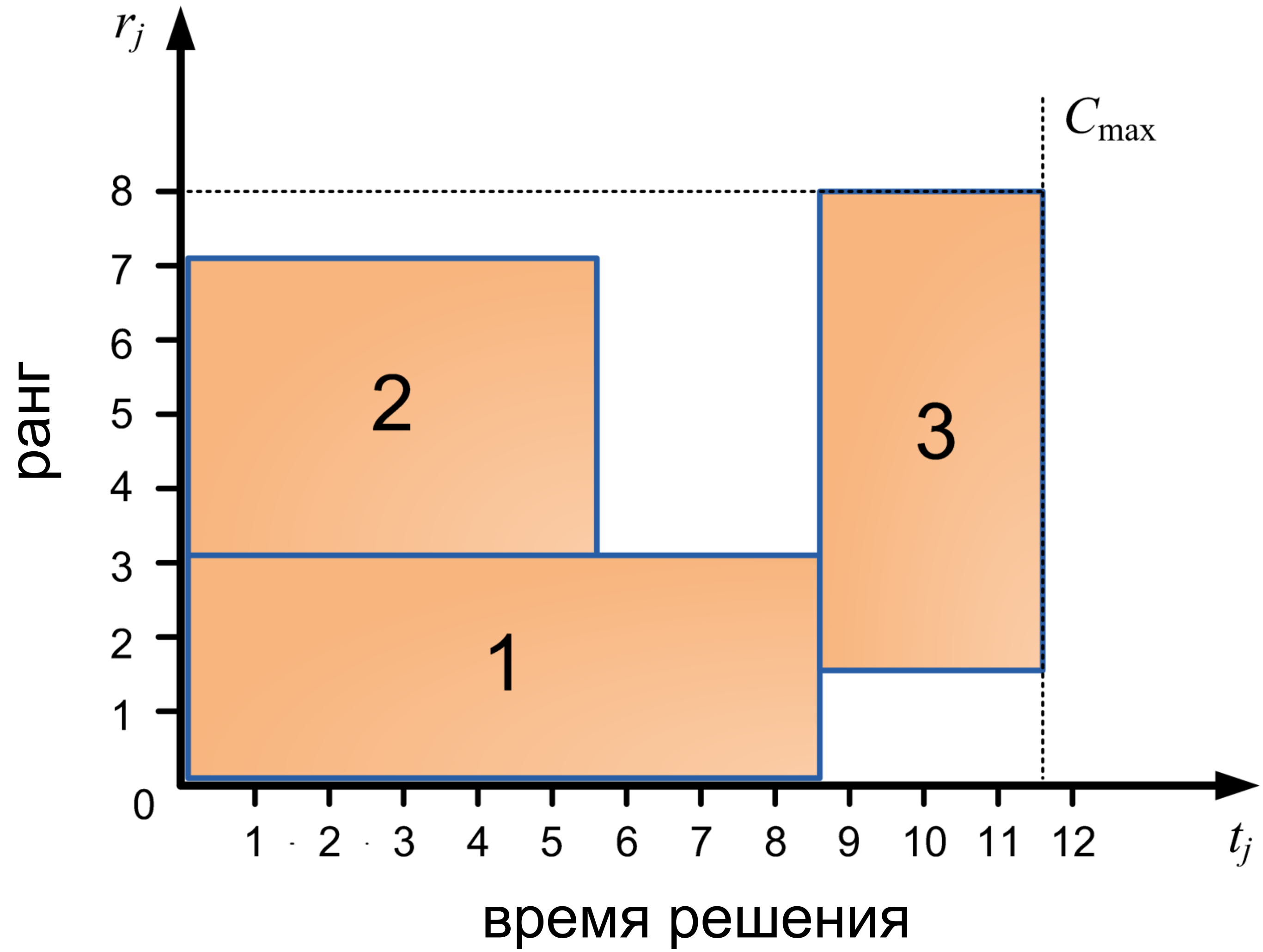
$$C_{\max}(S) = \max\{\tau_j + t_j\}$$

Недопустимое расписание



$S = (0, 0, 9, 0;\ 1, 2, 3;\ 4, 5, 6, 7;\ 3, 4, 5, 6, 7, 8;\ 7, 8, 9, 10)$

Допустимое расписание



S = (0, 0, 9; 1, 2, 3; 4, 5, 6, 7; 3, 4, 5, 6, 7, 8)

Задача построения расписания

Требуется найти допустимое расписание $S \in \Omega$, доставляющее минимум целевой функции $T(S)$:

$$T(S) = \max_{j \in J}\{\tau_j + t_j\} \to \min_{S \in \Omega} \qquad (1)$$

при ограничениях:

$$\sum_{j \in J(t)} r_j \le n, \quad \forall t \in \Re, \qquad (2)$$

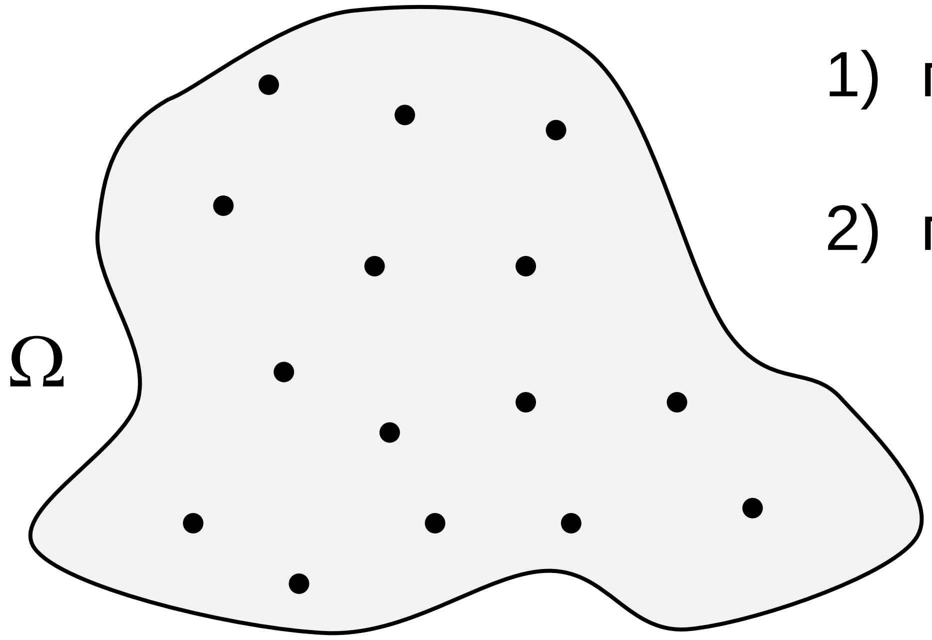
$$\prod_{j \in J(t)} \prod_{j' \in J(t) \setminus \{j\}} (x_{ji} - x_{j'i'}) \ne 0, \quad \forall t \in \Re, \quad i = 1,2,\ldots,r_j,\, i' = 1,2,\ldots,r_{j'} \qquad (3)$$

$$x_{ji} \in C,\; j = 1,2,\ldots,m,\, i = 1,2,\ldots,r_j \qquad (4)$$

$$\tau_j \in \Re,\, j = 1,2,\ldots,m. \qquad (5)$$

Задача (1) – (5) относится к дискретной оптимизации и является трудноразрешаемой

Как искать точное решение?



1) генерируем все решения
2) полный перебор

Это бесперспективно!

Алгоритмы решения задачи (1) – (5)

Необходимо разработать быстрый алгоритм решения задачи (1) – (5) с решением, близким к оптимальному.

1) Быстрый – в смысле вычислительно сложности:

$O(m_2), O(m\log m), \ldots$

2) Обеспечивает решение, близкое к оптимальному:

Верхняя граница: $C'_{\max} = \sum_{j \in J} t_j$

– все задачи решаются последовательно

Нижняя граница: $C''_{\max} = \max\{t_j\}$

– все задачи решаются параллельно

Алгоритмы поиска точного решения

Решение

Приближенные алгоритмы

Основная идея – за полиномимальное время найти удовлетворяющее по точности решение.

Точные алгоритмы

Комбинаторный подход:

- полный перебор (backtracking)
- метод ветвей и границ (branch and bounds)
- ...

(можно распараллелить)

Для целей анализа

Подходы к построению приближенных алгоритмов

Стохастические алгоритмы (локальный поиск)

- имитация отжига (simulated annealing)
- генетические алгоритмы
- муравьиные колонии (ant colony optimization)
- локальный поиск
- поиск с запретами (tabu search)
- метод цепей Монте-Карло

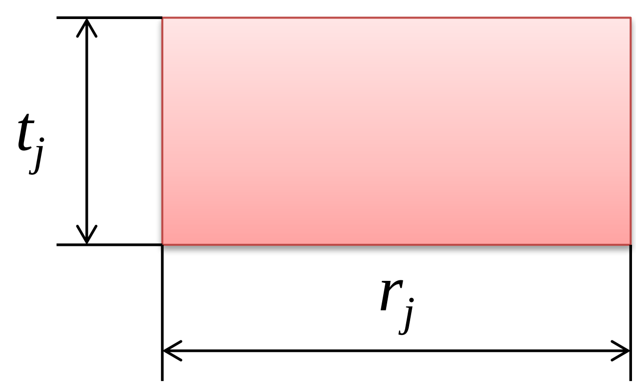
Эвристические алгоритмы (конструируют решение, а не перебирают)

- сведение к задаче упаковки объектов в полуограниченную полосу (strip packing)
- упаковка объектов в контейнеры

Методы решения задачи

Задача (1) – (5) → Задача двумерной упаковки прямоугольников в полуограниченную полосу (2D Strip Packing, 2DSP)

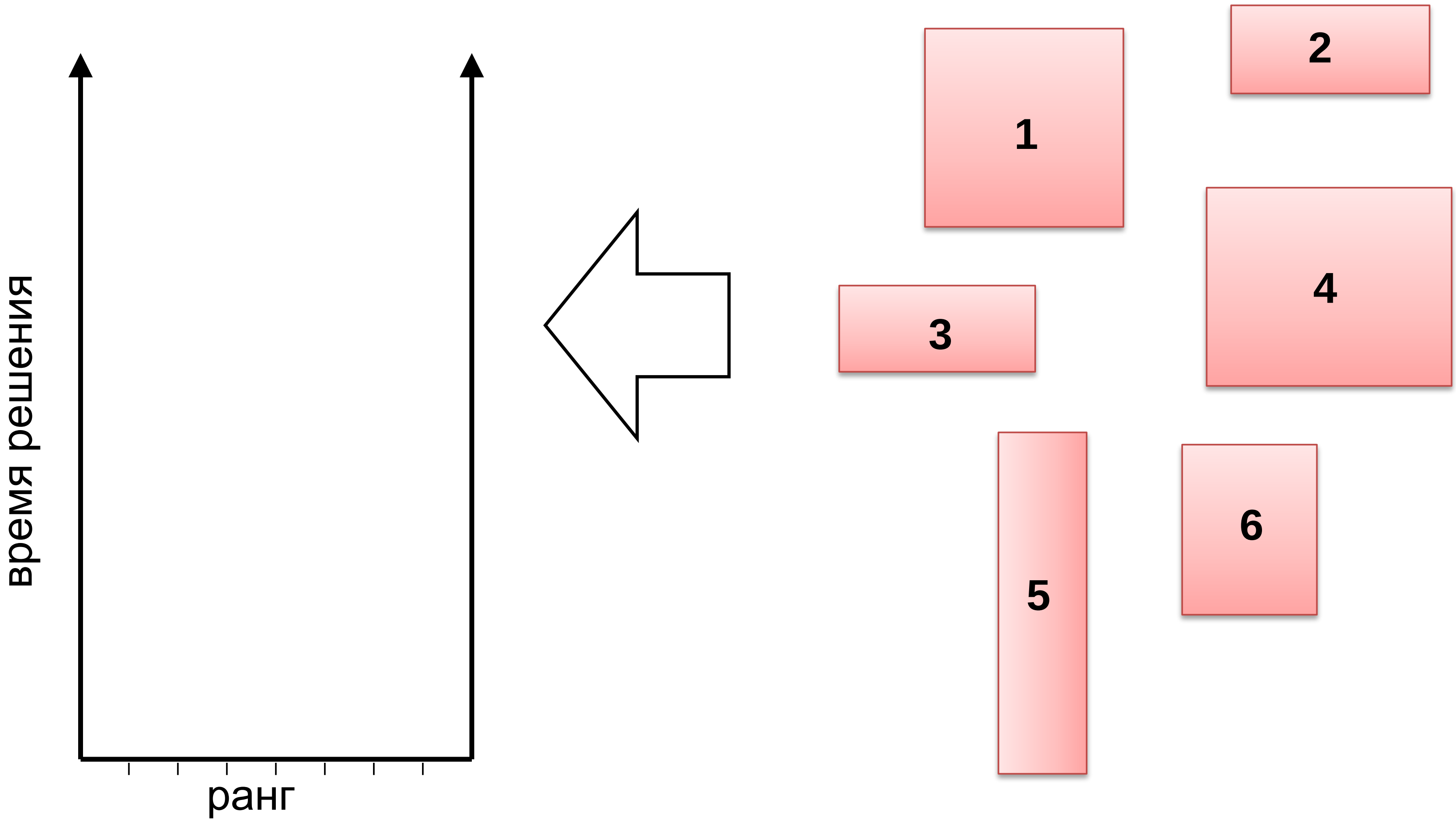
Задача $j \in J$ представляется в виде прямоугольника шириной r_j и высотой t_j.



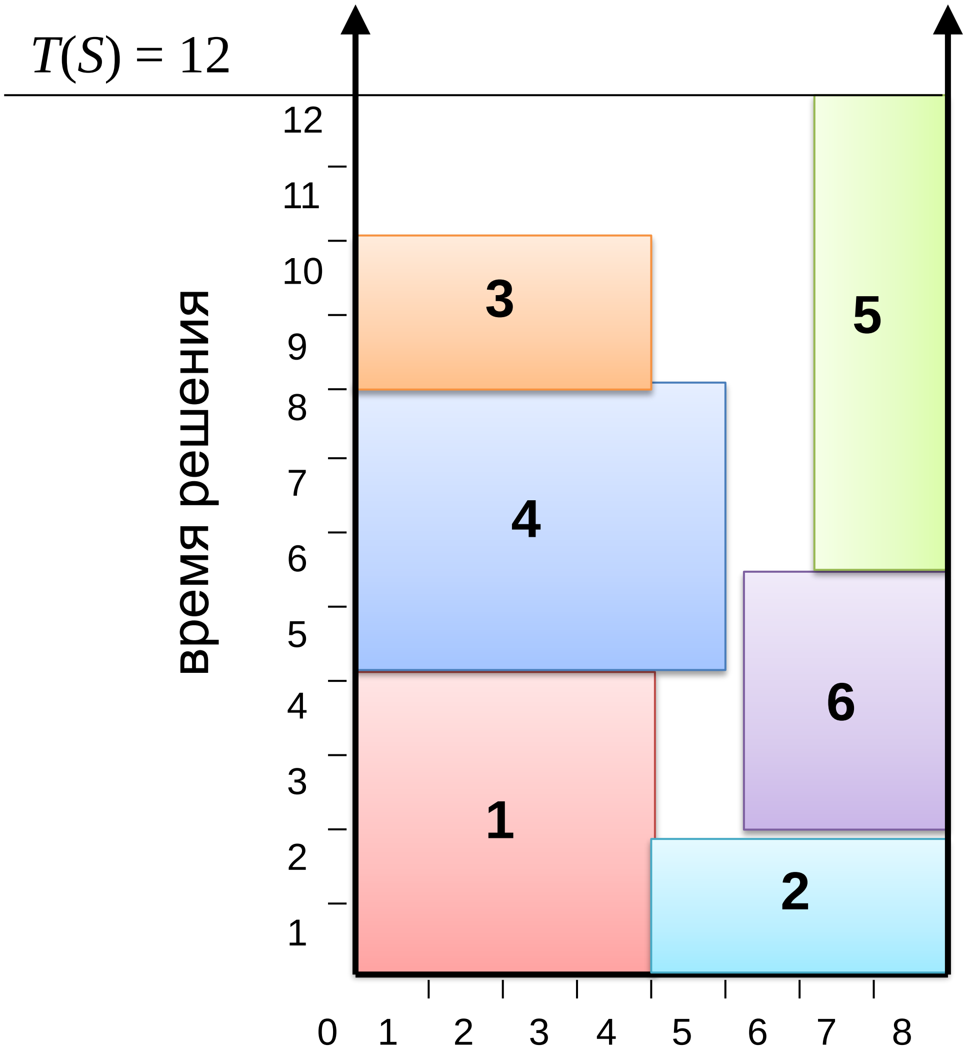
Методы решения задачи

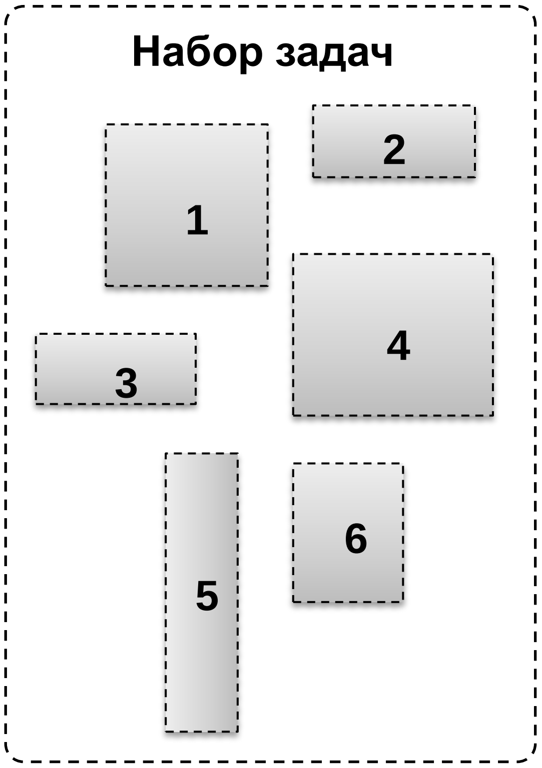
Полуограниченна полоса

Набор задач



Пример упаковки





$S = (0, 0, 8, 4, 5, 2; 1, 2, 3, 4; 5, 6, 7, 8; 1, 2, 3, 4; 1, 2, 3, 4, 5; 7, 8; 6, 7, 8)$

Моделирование

1. Цели исследования?

- точность алгоритмов

2. Модель системы

$n = 32, 64, 128, 256, \ldots$

3. Модель задачи

$r_j = ?,\ t_j = ?,$

Parallel Workload Archives

http://www.cs.huji.ac.il/labs/parallel/workload/logs.html

Моделирование

Вычислительная система

Параллельные задачи

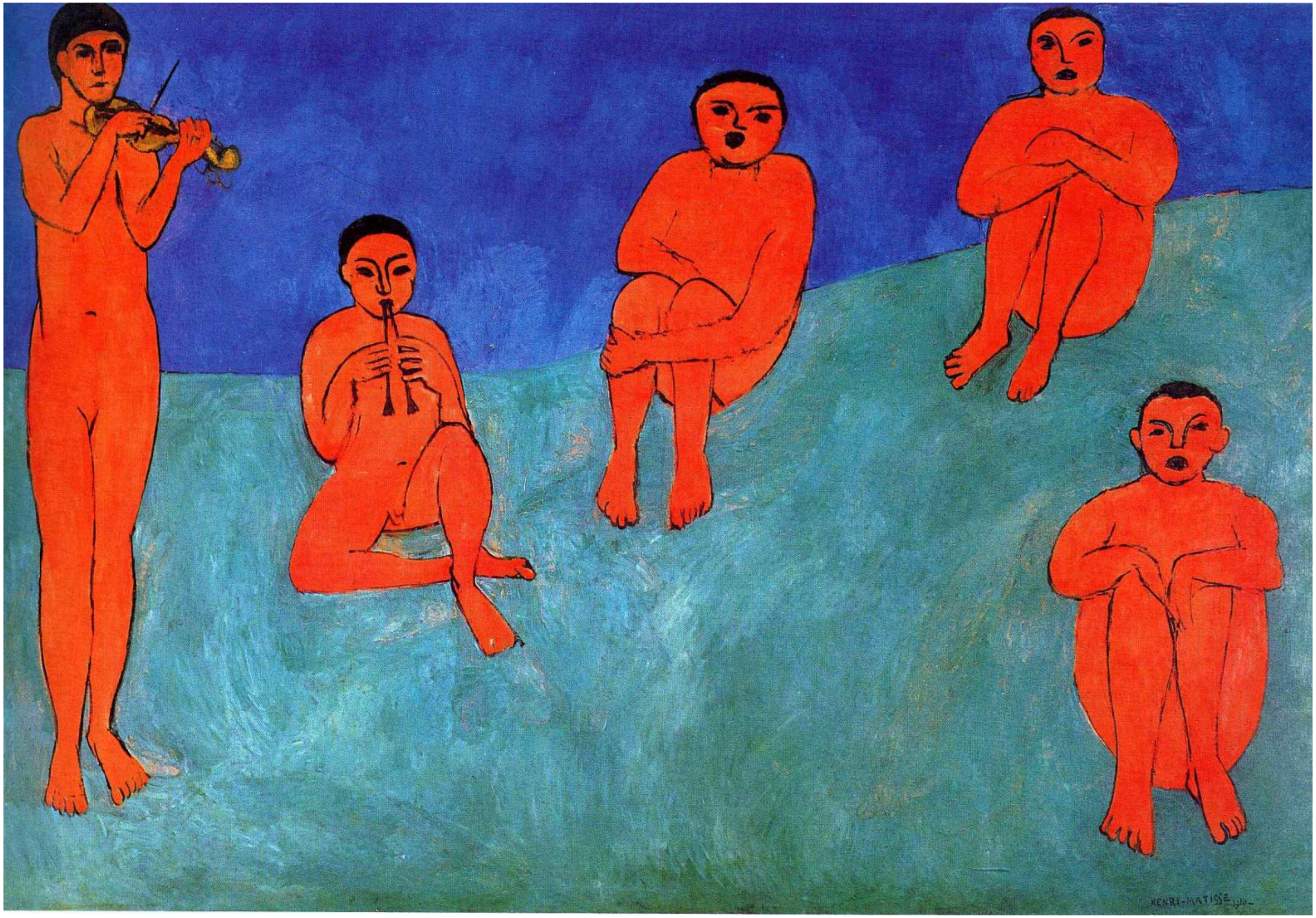
Алгоритмы

Истекшее время: $0.02/t_{alg}$

$C_{\max} = ?$

Относительная погрешность:

$$\varepsilon = \frac{C_{\max} - C''_{\max}}{C''_{\max}}$$



А.Матисс «Музыка»