# Многопроцессорное расписание

Габдрахманов Азат

26 ноября 2024 г.

# Введение

- Многопроцессорное расписание — важная задача в области теории расписаний и оптимизации.
- Применяется в компьютерных системах, производственных процессах, логистике и других областях.
- Цель: оптимальное распределение заданий между процессорами для достижения наилучшего результата.

- https://en.wikipedia.org/wiki/Flow-shop$_s$*cheduling*

# Формальное определение задачи

**Условие:**

- Задано множество заданий $T$
- Число процессоров $m$
- Длительность каждого задания $l(t)$ для $t \in T$
- Общий директивный срок $D$

**Вопрос:** Существует ли $m$-процессорное расписание для заданий из $T$, которое удовлетворяет общему директивному сроку $D$?

**Формальное условие:**

$$\exists\, o : T \to \mathbb{Z}, \quad \forall u \geq 0, \quad \#\{t \in T \mid o(t) \leq u < o(t) + l(t)\} \leq m$$

и

$$\forall t \in T, \quad o(t) + l(t) \leq D$$

# Общие виды заданий

- **Процессоры с разной производительностью:**
  - Некоторые процессоры могут выполнять задания быстрее других.
  - Время выполнения задания на процессоре $P_j$: $l(t)/s_j$, где $s_j$ — скорость процессора.
- **Зависимости между заданиями:**
  - Некоторые задания не могут начинаться до завершения других.
- **Приоритеты заданий:**
  - Задания имеют разные приоритеты, влияющие на порядок их выполнения.

# Частные случаи задачи

- **Одинаковое время выполнения всех заданий:**
  - $l(t) = l$ для всех $t \in T$
  - Проблема становится аналогом задачи разбиения множества (Partition)
  - В этом случае задача может быть решена за полиномиальное время.
- **Ограниченное количество процессоров:**
  - Для фиксированного $m$ задача может быть проще решаемой.
- **Отсутствие дедлайнов:**
  - Задача упрощается, если нет ограничений по времени завершения.

# Доказательство NP-полноты

**Класс NP:**

- Задачи, для которых решение можно проверить за полиномиальное время.

**Сведение из известной NP-полной задачи:**

- Используем задачу Partition или 3-распределение (3-Partition) для доказательства.
- Показываем, что любая NP-задача может быть сведена к многопроцессорному расписанию.

**Основные шаги доказательства:**

1. Выбираем известную NP-полную задачу.
2. Строим полиномиальное сведение от этой задачи к задаче многопроцессорного расписания.
3. Показываем, что решение одной задачи эквивалентно решению другой.

**Вывод:** Задача многопроцессорного расписания является NP-полной.

# Применение задачи многопроцессорного расписания

- **Компьютерные системы:**
  - Распределение процессов и потоков между ядрами процессора.
- **Производственные системы:**
  - Планирование производственных процессов на различных машинах или линиях.
- **Логистика:**
  - Организация доставки и распределения ресурсов.
- **Облачные вычисления:**
  - Оптимизация использования виртуальных машин и серверов.

# Заключение

- Многопроцессорное расписание — фундаментальная задача в оптимизации и теории сложности.
- Задача NP-полная, что делает её решение для больших экземпляров вычислительно сложным.
- Практическое применение в различных областях требует использования эвристических и приближенных методов.
- Понимание сложности задачи важно для разработки эффективных алгоритмов распределения ресурсов.

# Список литературы


- M. R. Garey, D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, 1979.
- R. Graham. *Scheduling: Theory, Algorithms, and Systems*. Prentice Hall, 1966.
- K. Pruhs, J. R. Rice. *Scheduling on Multiprocessors: Algorithms and Complexity*. Springer, 1997.