

# Построение многопроцессорного расписания с использованием жадных стратегий и ограниченного перебора

Савицкий Илья

Научный руководитель: к.т.н. доцент Костенко Валерий Алексеевич

19 апреля 2022 г.



## 1 Постановка задачи

- Определения

## 2 Описание алгоритма

- Схема
- Предподсчет
- Пробное размещение работы
- Ограниченный перебор
- Корректировка критичности пути

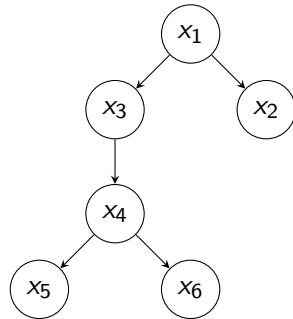
## 3 Текущие результаты



# Постановка задачи

Дано:

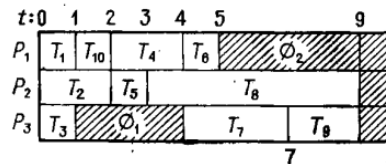
- 1 Ориентированный граф работ  $G$  без циклов, в котором дуги - зависимости по данным, а вершины - задания. Вершин  $n$ , дуг  $m$
- 2 Вычислительная система, состоящая из  $p$  различных процессоров
- 3 Матрица  $C_{ij}$  длительности выполнения работ на процессорах,  $i = 1 \dots n, j = 1 \dots p$
- 4 Матрица  $D_{kl}$  передач данных между процессорами,  $k = 1 \dots p, l = 1 \dots p, D_{kk} = 0$



# Постановка задачи

Требуется:

- 1 Построить расписание  $HP$ , то есть для каждой работы определить время начала ее выполнения и процессор на котором она будет выполняться
- 2 Минимизируемый критерий: время завершения выполнения расписания
- 3 Дополнительные ограничения



## Модель расписания

Множество корректных расписаний  $HP$  задается набором ограничений:

- В расписании не допустимы прерывания
- Интервалы выполнения заявок не пересекаются
- Каждая работа назначена на процессор
- Любую работу обслуживает один процессор

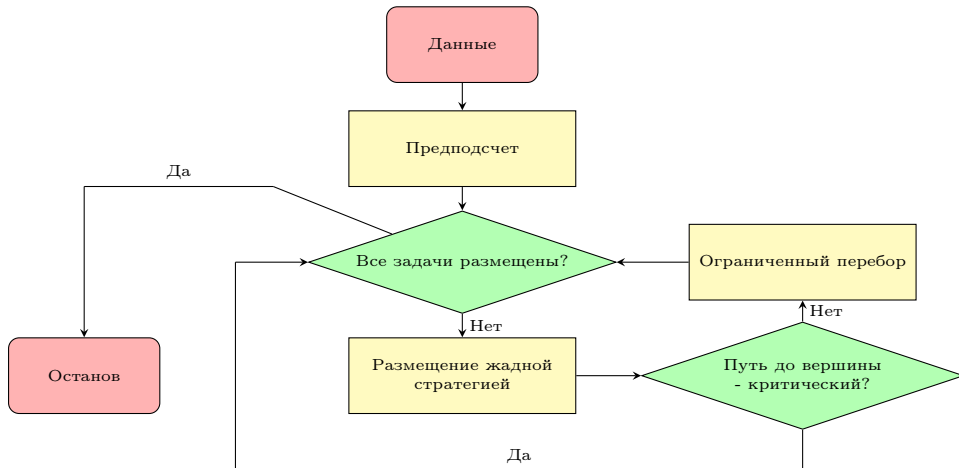


# Постановки задачи

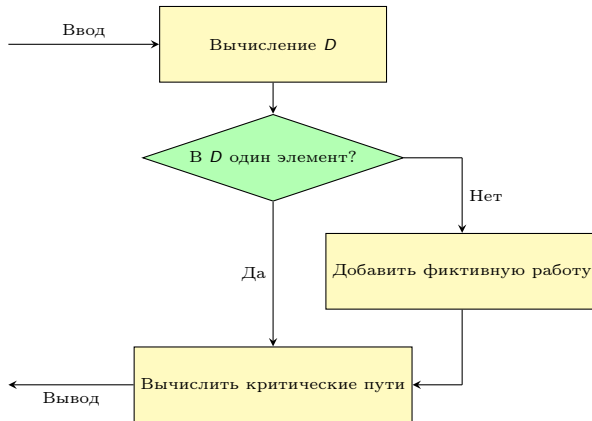
- 1 Задача с однородными процессорами (длительность выполнения работы не зависит от того, на каком процессоре она выполняется) и дополнительными ограничениями на количество передач:
  - $CR = \frac{m_{ip}}{m}$  - количество передач данных между работами на каждый процессор
  - $CR2 = \frac{m_{2e\,dg}}{m}$  - количество дуг, начальный и конечный узлы которых назначены на процессоры, не соединенных напрямую
- 2 Задача с однородными процессорами и контролем сбалансированности распределения работ:
  - $BF = \left( \frac{a_{max} \cdot p}{n} \right)$  - наибольшее, по всем процессорам, количество работ на процессоре
- 3 Задача с неоднородными процессорами, но без дополнительных ограничений на расписание



## Общая схема жадного алгоритма



## Блок-схема предподсчета

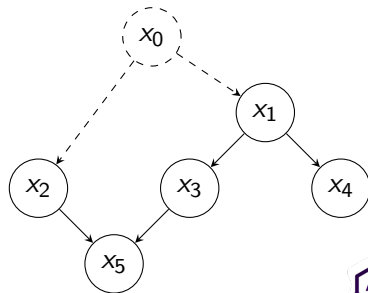




## Предподсчет

- 1 Формируется множество  $D = d_1, d_2, \dots, d_l$ , где  $l$  - количество вершин, доступных для добавления (т.е. у которых нет предшественников в исходном графе)
- 2 Вычисляется вектор  $k$  - вектор длин критических путей от "головной" вершины до каждой вершины графа. В случае, если такой вершины нет - создается фиктивная вершина с нулевой длительностью. Вектор  $k$  заполняется при помощи алгоритма Дейкстры.

Фиктивная вершина



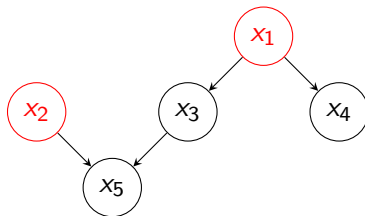
## Блок-схема пробного размещения работы



## Жадный критерий выбора размещения

Из множества  $D$  (выделено красным) выбирается работу по критерию максимальности количества потомков у вершины.

Выбранная вершина



## Пробное размещение работы

Пробное размещение работы производится с учетом жадного и дополнительных критериев.

Жадный критерий - скорейшее завершение работы в расписании.

Способы выбора места:

- 1 Подсчет усредненного взвешенного показателя среди критериев

$$crit = C_1 \cdot GR + C_2 \cdot CR + C_3 \cdot BF$$

,где  $GR$  - время скорейшего завершения работы,  $C_1, C_2, C_3$  - параметры алгоритма

- 2 Допускная система с выбором



## Допускная система выбора

- 1 Список мест размещения работ ранжируется по  $GR$ , после чего отсекаются верхние  $n\%$  работ
- 2 Такие же действия повторяются для каждого дополнительного критерия
- 3 В конечном списке выбрать место по жадному критерию

,где  $n$  - параметр алгоритма

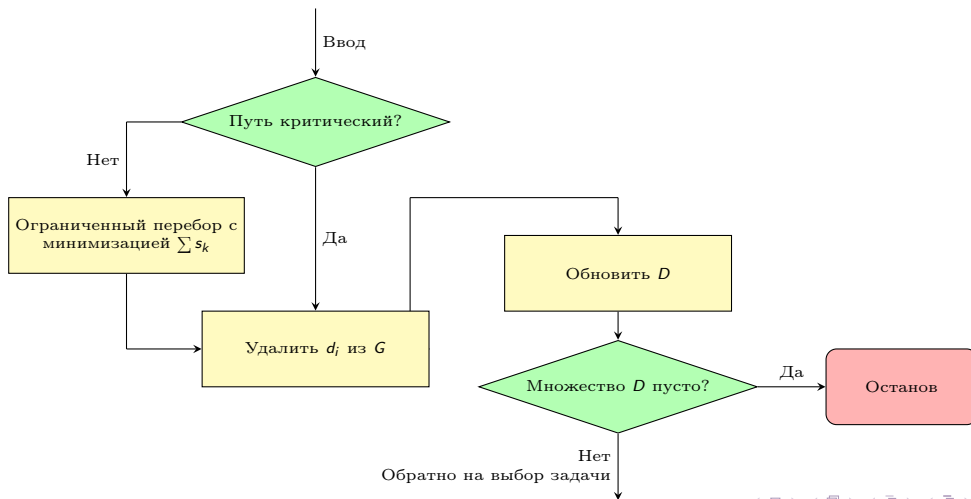


## Процедура ограниченного перебора

После неудачной пробной постановки работы в расписание алгоритм создает набор  $K = k_1, k_2, \dots, k_t$ , состоящий из  $t$  последних добавленных работ ( $t$  – параметр алгоритма). Далее, процедурой полного перебора пробуются различные расписания до тех пор, пока не получится расписание, удовлетворяющее критерию критичности пути до последней поставленной работы и удовлетворяющее дополнительные критерии



## Блок схема корректировки расписания



## Текущие результаты

- 1 Проведен обзор алгоритмов построения списочных расписаний. Цель обзора; выявление жадных критериев и схем ограниченного перебора которые могут быть модифицированы для решения данной задачи.
  - 2 Разработан и реализован алгоритм
- 
- 3 Проведено исследование свойств алгоритма на данных от Хуавей.

