

# Жадные алгоритмы для построения многопроцессорного списочного расписания

Савицкий Илья

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

27 апреля 2023 г.



# Цели и задачи ВКР

Целью ВКР является разработка детерминированного алгоритма построения многопроцессорного расписания с дополнительными ограничениями.

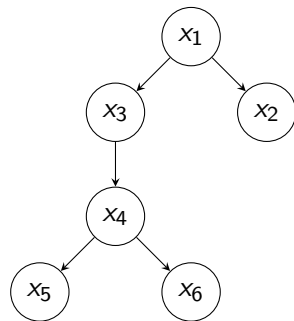
Для достижения указанной цели требуется:

- ❶ Провести аналитический обзор алгоритмов построения списочных расписаний с целью выявления алгоритмов, которые возможно модифицировать под поставленную задачу и имеют хорошую возможность масштабирования.
- ❷ Разработать и реализовать алгоритмы.
- ❸ Провести исследование качества решений и временной сложности алгоритмов.



# Постановка задачи

- 1 Граф потока управления  $G$  без циклов, в котором дуги - зависимости по данным, а вершины - задания. Вершин  $n$ , дуг  $m$
- 2 Вычислительная система, состоящая из  $p$  процессоров.
- 3 Матрица  $C_{n \times p}$  времени выполнения работ на процессорах. Каждая строка этой матрицы - длины выполнения  $n$ -й задачи на  $p$  процессорах.
- 4 Время  $d$ , затрачиваемое на межпроцессорную передачу.



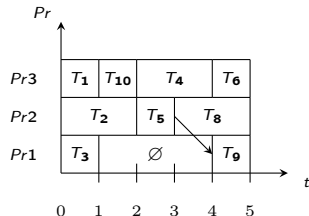
Граф потока данных



# Постановка задачи

Требуется:

- 1 Построить расписание  $HP$ , то есть для  $i$ -й работы определить время начала ее выполнения  $s_i$  и процессор  $p_i$  на котором она будет выполняться;
- 2 Минимизируемый критерий: время завершения выполнения расписания.



# Ограничение на корректность расписания

Множество корректных расписаний  $HP$  задается набором ограничений:

- В расписании не допустимы прерывания;
- Интервалы выполнения работ не пересекаются;
- Каждая работа назначена на процессор;
- Любую работу обслуживает один процессор;
- Частичный порядок, заданный графом потока управления  $G$ , сохранен в  $HP$ .



- ❶ Задача без дополнительных ограничений.
- ❷ Задача с дополнительным ограничением на количество передач:
  - $CR = \frac{m_{ip}}{m} < 0.4$ , где  $m_{ip}$  - количество межпроцессорных передач в расписании.



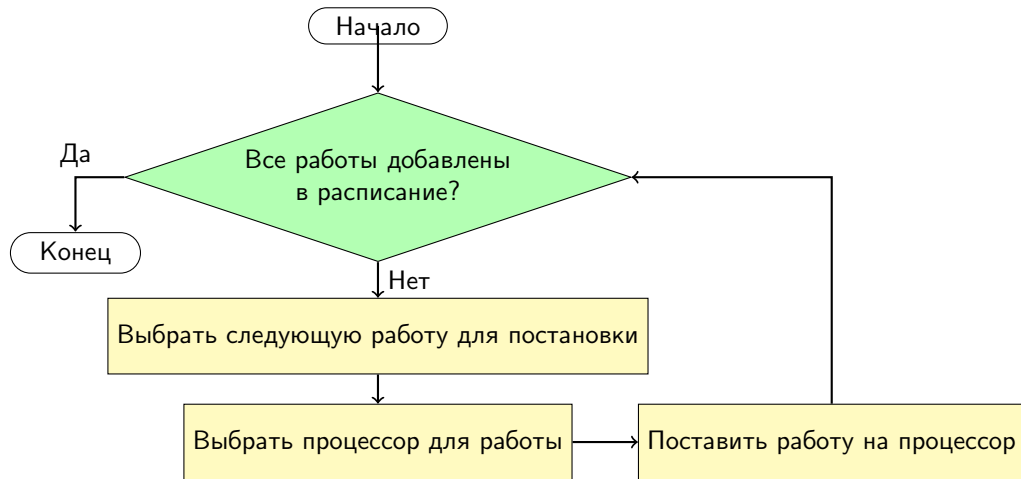
# Обзор предметной области

Проведен обзор детерминированных алгоритмов, которые возможно модифицировать под поставленную задачу и имеют хорошую возможность масштабирования.

Название алгоритма	Возможность модификации алгоритма	Возможность масштабирования алгоритма
Метод ветвей и границ	✓	✗
Метод динамического программирования	✓	✗
Алгоритм поиска максимального потока	✗	✓
Жадные алгоритмы	✓	✓



# Общая схема жадных алгоритмов построения расписания





# Жадный алгоритм с выбором по числу потомков

- 1 Выбор следующей работы на постановку - критерий GC1
- 2 Выбор процессора для работы
  - Для  $CR$  - из изначально заданного распределения
  - Для  $NO$  - по критерию GC2

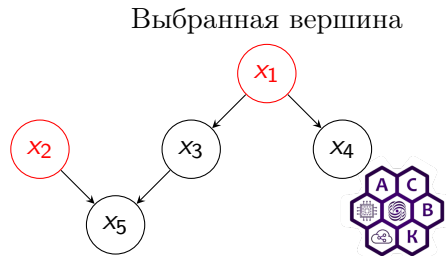
Зададим множество доступных для добавления вершин  $D = (d_1, d_2, \dots, d_l)$ , где  $l$  - количество вершин, доступных для добавления.

## Критерий GC1:

Из множества  $D$  выбирается работа по критерию GC1 максимальности количества потомков у вершины.

## Критерий GC2:

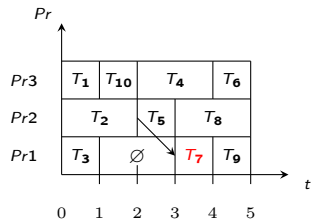
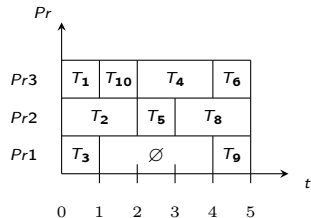
Работа ставится на процессор, на котором время завершения работы будет минимальным.



# Алгоритм постановки работы на процессор

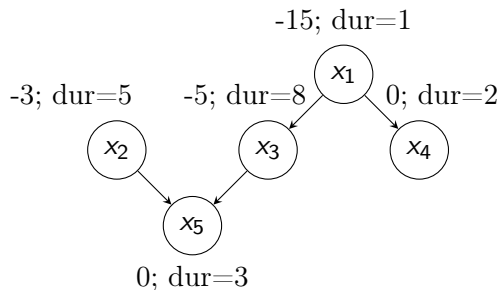
При постановке требуется найти такое минимальное время  $t$ , чтобы

- 1 Все передачи данных завершились до  $t$ ;
- 2 Существует интервал простоя длительности не меньше времени выполнения работы, начинающийся в  $t$ .



# Жадный алгоритм с фиктивными директивными сроками

- 1 Выбор следующей работы на постановку - в порядке возрастания фиктивных директивных сроков;
- 2 Выбор процессора для работы:
  - Для  $CR$  - из изначально заданного распределения
  - Для  $NO$  - по критерию  $GC2$



Распространение директивных сроков по графу потока управления.  
Все межпроцессорные передачи равны 2.  
Только  $x_2$  и  $x_5$  находятся на разных процессорах.



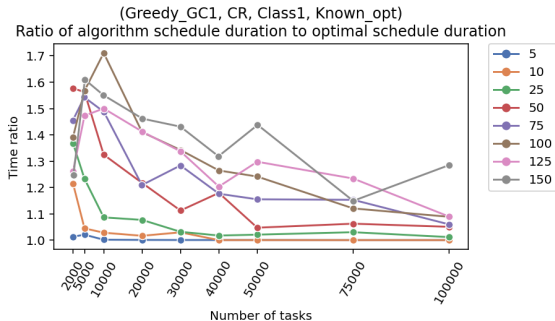
# Наборы данных для исследования

Для исследования качества решений и временной сложности алгоритма были созданы следующие наборы данных:

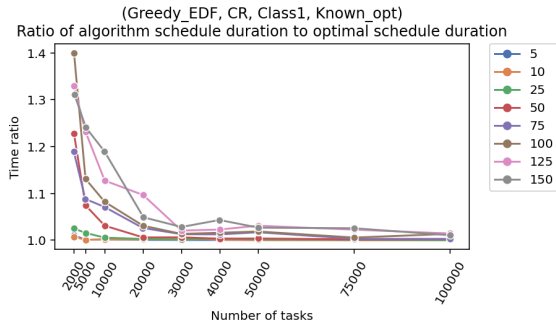
- 1 Набор данных с известным оптимумом.
- 2 Набор данных, основанных на слоистых данных.
- 3 Набор данных для построения расписания на неоднородных процессорах.



# Точность полученного расписания. CR



(a) Жадный алгоритм с выбором по числу потомков

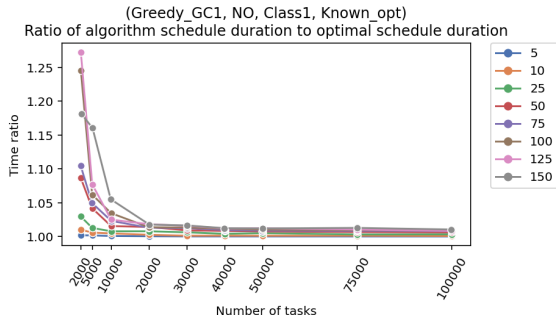


(b) Жадный алгоритм с фиктивными директивными сроками

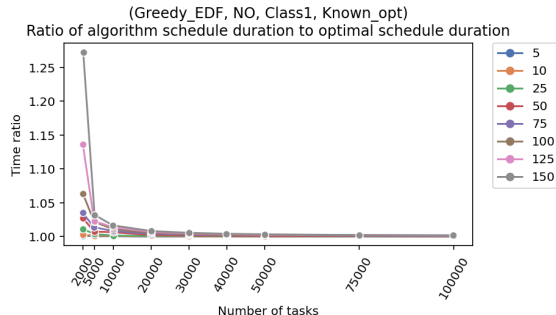
Качество решений алгоритмов на данных с известным оптимумом, постановка с дополнительным ограничением на межпроцессорные передачи



# Точность полученного расписания. NO



(a) Жадный алгоритм с выбором по числу потомков

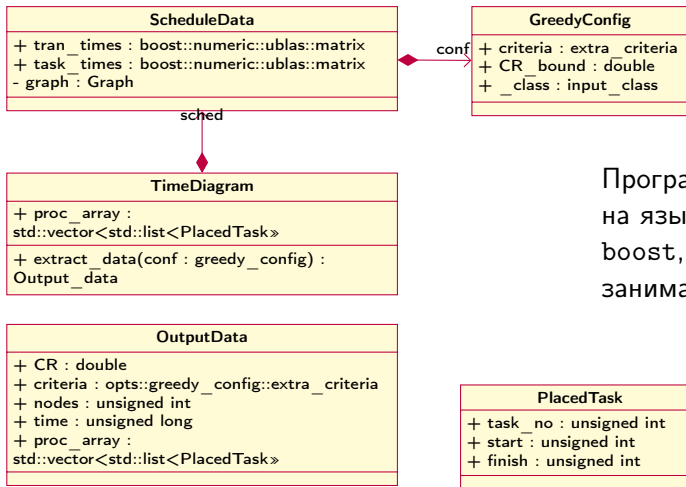


(b) Жадный алгоритм с фиктивными директивными сроками

Качество решений алгоритмов на данных с известным оптимумом, постановка без дополнительных ограничений



# Программная реализация алгоритма



Программная реализация была выполнена на языке C++, с использованием библиотек boost, METIS, json и toml. Реализация занимает 1689 строк.



- 1 Проведен аналитический обзор алгоритмов построения списочных расписаний с целью выявления алгоритмов, которые возможно модифицировать под поставленную задачу и имеют хорошую возможность масштабирования, по результатам которого были выбраны жадные алгоритмы.
- 2 Разработаны и реализованы алгоритмы, основанные на различных жадных критериях.
- 3 Проведено исследование свойств алгоритма, которое показало низкую вычислительную сложность и среднее отклонение от оптимума в 30% для жадного алгоритма с выбором по числу потомков и до 5-10% для жадного алгоритма с фиктивными директивными сроками.



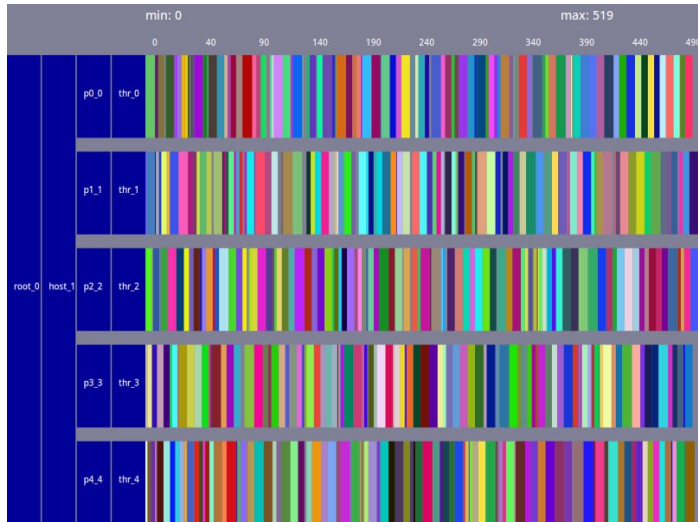


# Критерии модификации алгоритма

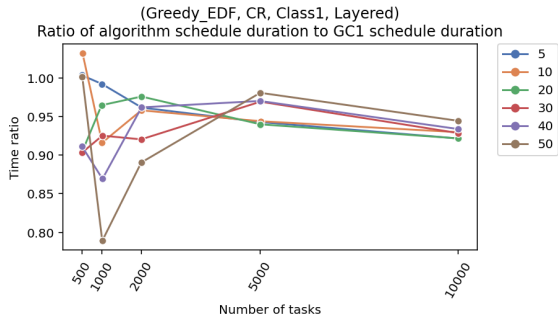
- 1 Возможность учета количества межпроцессорных передач.
- 2 Учет затрат на межпроцессорные передачи.



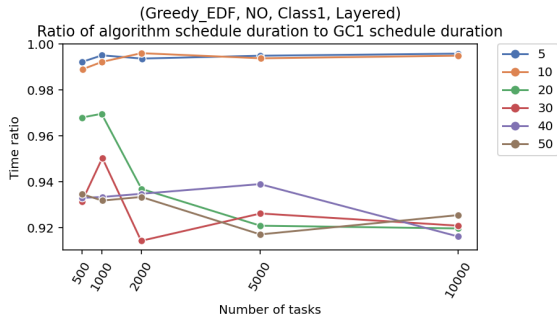
# Проблема проверки алгоритма на данных с известным оптимумом



# Точность полученного расписания. CR и NO



(a) При постановке CR

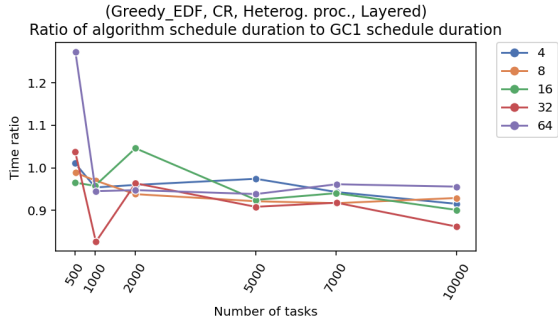


(b) При постановке NO

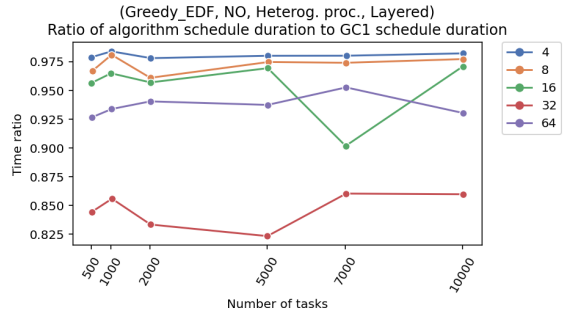
Отношение длительности работы алгоритма с фиктивными директивными сроками к длительности работы жадного алгоритма на данных, основанных на слоистых графах



# Точность полученного расписания. CR и NO



(a) При постановке CR



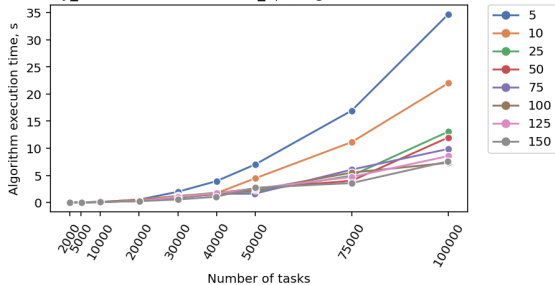
(b) При постановке NO

Отношение длительности работы алгоритма с фиктивными директивными сроками к длительности работы жадного алгоритма на данных, основанных на неоднородных процессорах



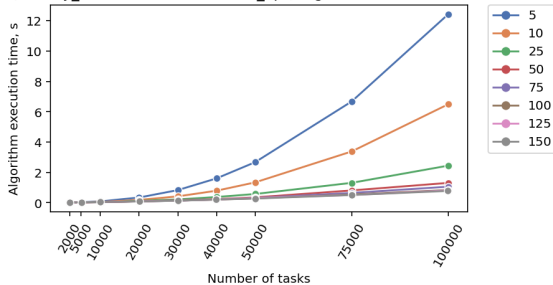
# Время выполнения программы. CR и NO.

(Greedy\_GC1, CR, Class1, Known\_opt) Algorithm execution time, s



(a) Жадный алгоритм с выбором по числу потомков

(Greedy\_EDF, CR, Class1, Known\_opt) Algorithm execution time, s



(b) Жадный алгоритм с фиктивными директивными сроками

Время выполнения алгоритма на данных с известным оптимумом, в секундах

