

Исследование алгоритма имитации отжига для задачи построения расписаний

Лысцов Илья Андреевич
Вариант 2

5 ноября 2024 г.

Формальная постановка задачи

Дано:

- $P = \{p_i \mid i = 1, 2, \dots, N\}$ — множество независимых работ, где N — общее количество работ.
- $W = \{w_i \mid i = 1, 2, \dots, N\}$ — множество времен выполнения каждой работы p_i , где w_i — время выполнения работы p_i .
- M — количество процессоров, $M \in \mathbb{N}$.

Требуется:

Построить расписание выполнения всех работ из множества P на M процессорах. Расписание определяется как отображение:

$$S : p_i \rightarrow (m, o_i)$$

где:

- $m \in \{1, 2, \dots, M\}$ — номер процессора, на котором выполняется работа p_i ,
- $o_i \in \mathbb{N}$ — порядок выполнения работы p_i на процессоре m . Для каждой пары работ p_i и p_j , выполняющихся на одном и том же процессоре m , порядок выполнения уникален:

$$\forall i, j : S(p_i) = (m, o_i), S(p_j) = (m, o_j), i \neq j \Rightarrow o_i \neq o_j.$$

Порядок выполнения нумеруется последовательно: если на процессоре m выполняются k работ, то $o_i \in \{1, 2, \dots, k\}$.

Время завершения работы

Для вычисления времени завершения работы p_i на процессоре m используется следующее правило. Если работа p_i имеет порядок выполнения o_i , то время её завершения f_i вычисляется как сумма времен выполнения всех предшествующих работ, которые выполняются на процессоре m , плюс её собственное время выполнения:

$$f_i = \sum_{\substack{j : o_j < o_i, \\ S(p_j) = (m, o_j)}} w_j + w_i.$$

Минимизируемый критерий:

Необходимо минимизировать критерий K — суммарное время завершения всех работ, определяемое как:

$$K = \sum_{i=1}^N f_i$$

где f_i — время завершения работы p_i , вычисленное через её порядок выполнения.

Экспериментальное исследование

Входные данные

Для генерации входных данных была написана отдельная функция, принимающая на вход число процессоров, число работ, диапазон длительностей работ. Функция создаёт CSV файл, содержащий описание набора работ.

Длительность каждой работы при исследовании находилась в диапазоне от 1 до 100: $w_i \in \{1, \dots, 100\}$.

Последовательный алгоритм

Исследование проводилось на параллельной реализации с заданным числом процессов 1.

На рисунках 1, 2, 3 изображены зависимости времени работы алгоритма от размера входных данных для разных законов понижения температуры.

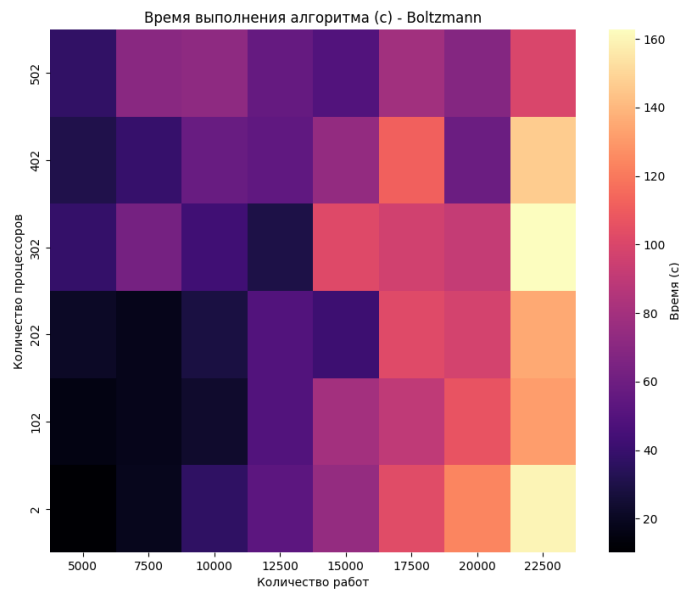


Рис. 1: Тепловая карта для закона Больцмана

Из рисунков видно, что алгоритм, использующий закон Больцмана, работает дольше всего на одинаковых входных данных. При этом улучшения критерия К по сравнению с использованием других законов охлаждения не было замечено.

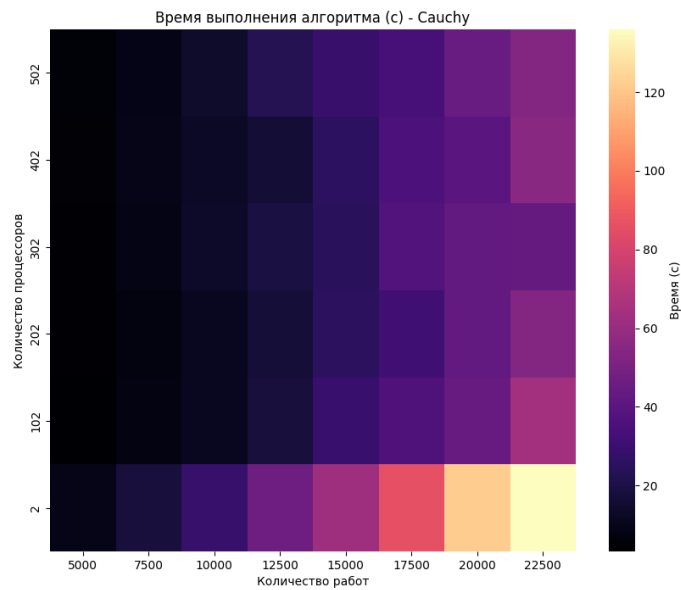


Рис. 2: Тепловая карта для закона Коши

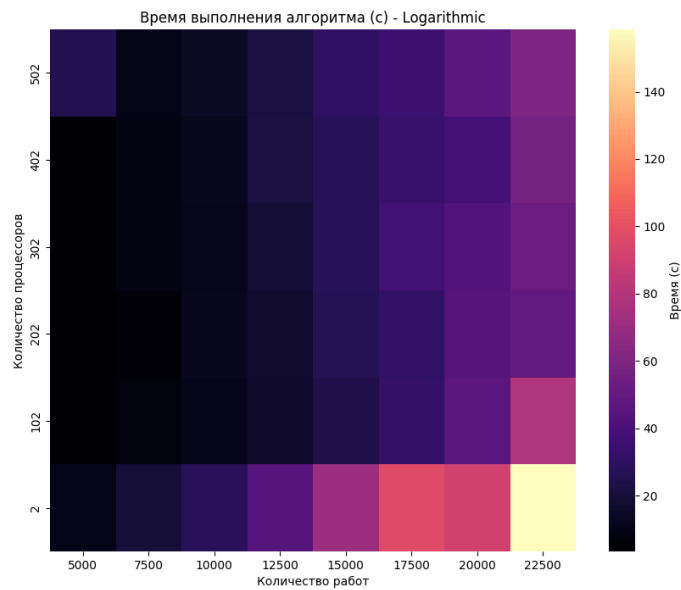


Рис. 3: Тепловая карта для логарифмического закона

Параллельный алгоритм

Для проведения эксперимента был выбран закон Коши с числом процессоров 102 и числом задач 22500. Зависимость времени работы от числа процессов в параллельном алгоритме представлена на рисунке 4. Зависимость значения критерия К от числа процессов изображена на рисунке 5.

Из графиков видно, что при увеличении числа процессов время выполнения программы растёт. При этом программа с большим числом процессов находит более точное решение.

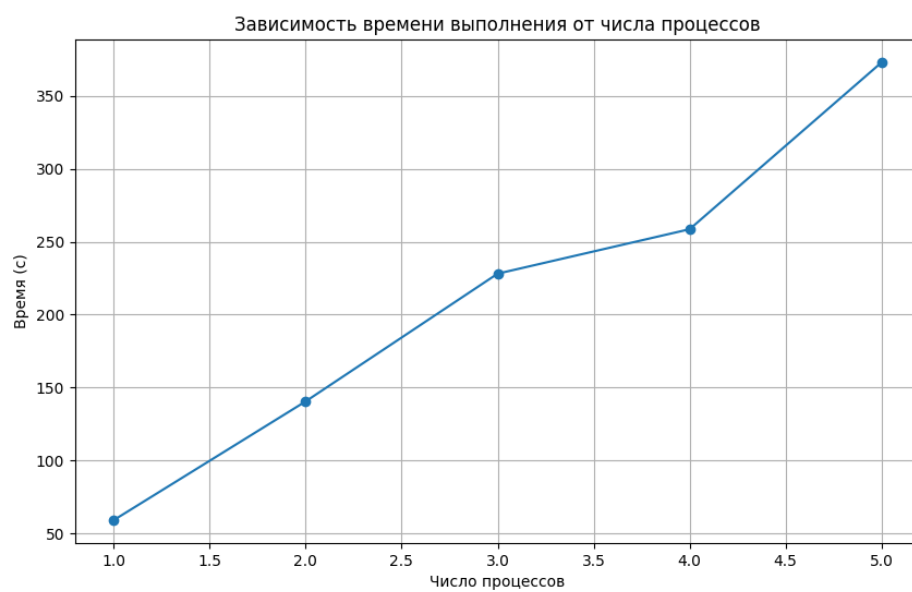


Рис. 4: График зависимости времени выполнения от числа процессов

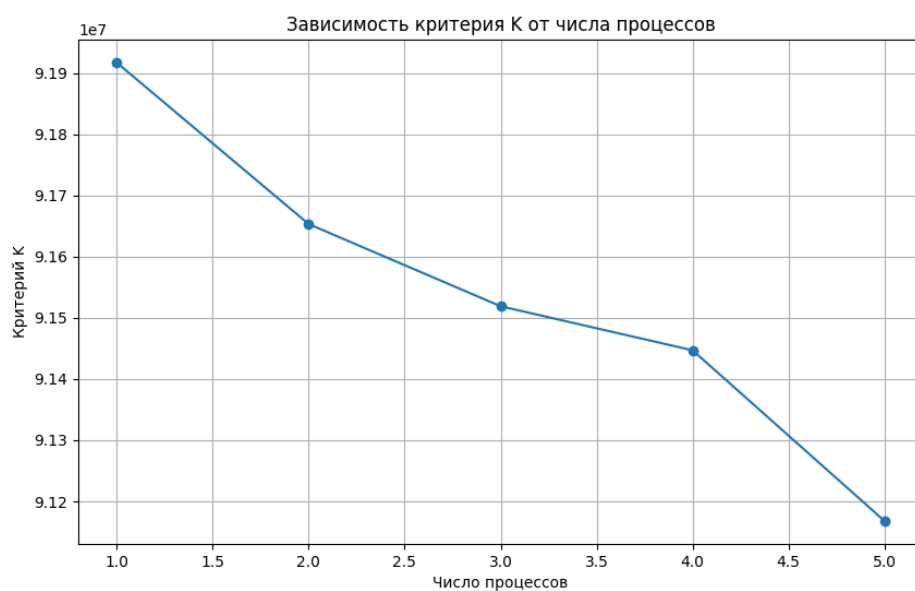


Рис. 5: График зависимости критерия К от числа процессов