

**Формальная постановка задачи
оптимизации расписания с использованием
алгоритма имитации отжига**

Кяжин Никита Олегович

5 ноября 2024 г.

Формальная постановка задачи

Дано:

- Пусть $J = \{j_1, j_2, \dots, j_N\}$ – множество заданий, где N – количество заданий.
- Пусть $\tau = \{t_1, t_2, \dots, t_N\}$ – для каждого задания j_i задано время выполнения $t_i > 0$.
- Пусть $P = \{p_1, p_2, \dots, p_M\}$ – множество процессоров, на которых выполняются задания, где M – количество процессоров.

Расписание:

Расписанием является булева матрица $S^{N \times M}$, в которой $s_{ij} \in \{0, 1\}$, где $i \in 1, \dots, N$, а $j \in 1, \dots, M$. Значение $s_{ij} = 1$ означает, что задание i выполняется на процессоре j , а $s_{ij} = 0$ – что задание i не выполняется на процессоре j .

Обозначим G_j – множество индексов задач, которые выполняются j процессором. Тогда $T_j = \sum_{i \in G_j} t_i$ – время выполнения всех задач, запланированных на j процессор.

Требуется:

Построить расписание $S^{N \times M}$, при котором будет минимизирован критерий, при этом все задания J будут выполнены на множестве процессоров P без прерываний, с учетом ограниченных ресурсов, и не будет пересечений в использовании процессоров.

Минимизируемый критерий:

В зависимости от остатка от деления на 2 контрольной суммы CRC32 от фамилии и инициалов выбирается один из следующих критериев:

- Критерий K_1 (разбалансированность расписания)
- Критерий K_2 (суммарное время ожидания)

$CRC32_{KiazhinNO} = 3618506679$, следовательно выбираем 1 критерий для реализации.

Критерий разбалансированности расписания:

$$K_1 = T_{max} - T_{min} \quad (1)$$

где:

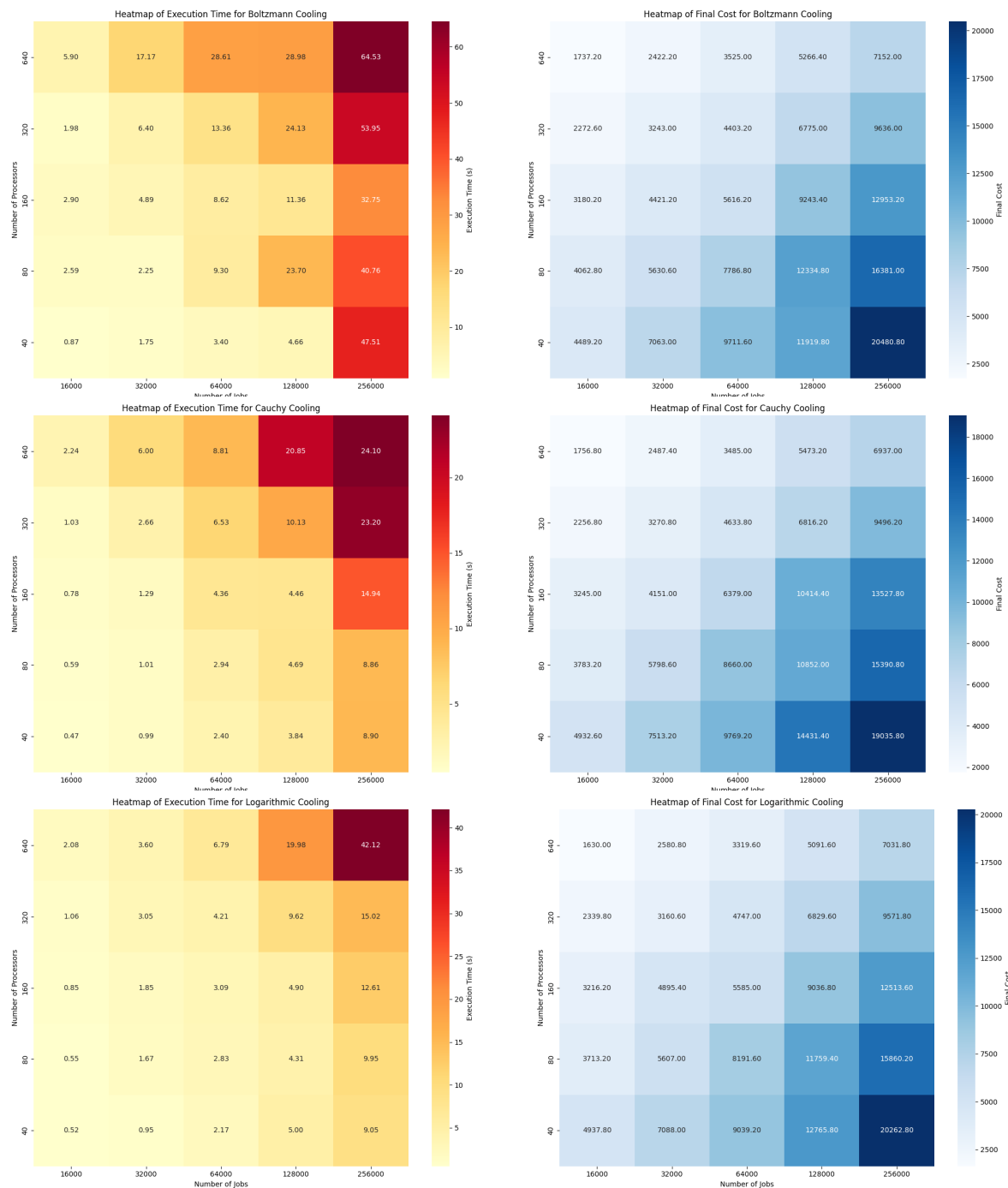
$$T_{max} = \max_{j \in 1, \dots, M} T_j \quad (2)$$

$$T_{min} = \min_{j \in 1, \dots, M} T_j \quad (3)$$

Ограничения

- Каждый процессор $p_j \in P$ в любой момент времени может выполнять не более одного задания.
- Во время выполнения задания процессором, не возникает прерываний.
- Процессор может мгновенно (без прерывания) переключаться между заданиями.
- Каждая строка S_i матрицы $S^{N \times M}$ должна содержать не более одной единицы. (Каждая задача выполняется только на одном процессоре и только один раз)
- Время выполнения $t_i \in \tau$ фиксировано.

Исследование последовательного алгоритма

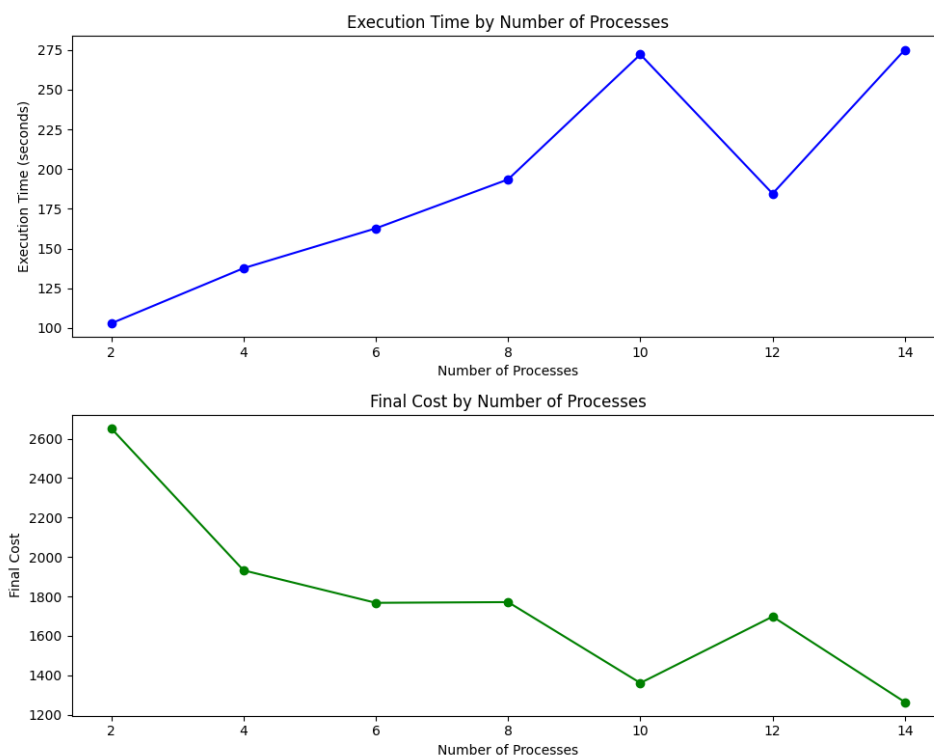


Тепловые карты представляют собой средние значения, полученные в результате пяти запусков последовательной программы.

На основании проведенных исследований можно сделать вывод, что алгоритм понижения температуры на основе модели Больцмана демонстрировал самое длительное время выполнения по сравнению с другими алгоритмами. Тем не менее, это не отразилось на точности вычислений. Все алгоритмы показали сопоставимые результаты с учетом допустимой погрешности.

Кроме того, можно отметить закономерность: с увеличением количества процессоров и уменьшением количества задач точность вычислений возрастает. Это может свидетельствовать о том, что алгоритм имитации отжига сталкивается с большей сложностью при распределении значительного объема заданий на меньшее количество процессоров. При увеличении данных стоит увеличивать и критерий останова, для достижения лучших результатов.

Исследование последовательного алгоритма



В графе представлены средние значения метрик за 5 запусков, были взяты данные из 16000 задач и 40 процессоров. На основе полученных данных можно понять, что параллельный алгоритм работает многократно дольше последовательной версии, однако имеет значимый прирост производительности.