



Московский государственный университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра автоматизации систем вычислительных комплексов

Отчет по заданию практикума
Алгоритм имитации отжига

Выполнил:
Хасанов Дмитрий Максимович, 421 группа

Москва, 2022

Содержание

1	Введение	2
2	Формальная постановка задачи	2
3	Исследование производительности реализации	2

1 Введение

С помощью алгоритма имитации отжига необходимо решить **задачу**:

Дано N независимых работ, для каждой работы задано время выполнения. Требуется построить расписание выполнения работ без прерываний на M процессорах.

На расписании должно достигаться минимальное значение критерия: суммарное время ожидания (т.е. сумма, по всем работам, времён завершения работ).

2 Формальная постановка задачи

Расписание $S = \{s_i | i = 1, \dots, N\}$, где:

- $s_i = (p_i, n_i)$ – состояние работы с номером i :
 - p_i – номер процессора, на котором запущена работа
 - n_i – номер работы на процессоре
- N — общее число работ

ДАНО:

- M – число процессоров;
- $X = (X_1, X_2, \dots, X_N)$, где:
 - $X_i \in \mathbb{N}$ – длительность работы с номером i
 - N – общее число работ

ТРЕБУЕТСЯ:

Получить расписание S , минимизирующее требуемый критерий.

Для подсчета времени T завершения работы X с состоянием $s = (p, n)$ воспользуемся формулой:

$$T = \sum_{i: p_i=p, n_i < n} X_i$$

МИНИМИЗИРУЕМЫЙ КРИТЕРИЙ:

$$\min_S \sum_{i=1}^N T_i$$

3 Исследование производительности реализации

Было необходимо экспериментально получить ответы на вопросы:

- При каких количествах процессоров и работ последовательный алгоритм ИО работает больше 1 минуты хотя бы с одним из законов понижения температуры
- С каким из законов понижения температуры на таких «тяжелых» входных данных алгоритм работает дольше? Находит ли он при этом лучшие решения, чем при других законах понижения температуры?

Ход экспериментов:

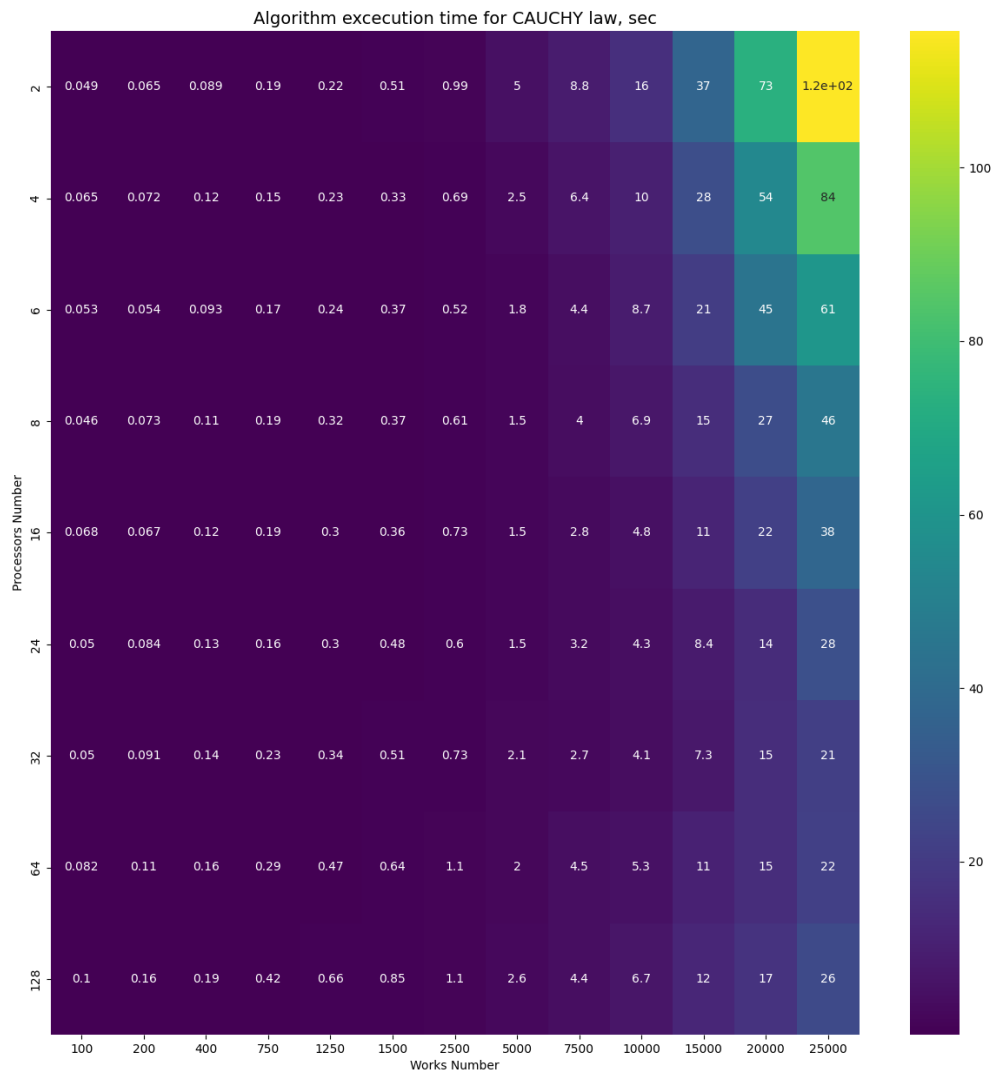
Запускался алгоритм с законом понижения температуры Коши на параметрах nProc и nTasks по сетке. На одной паре данных проводилось 5 запусков, результаты усреднялись.

Входные параметры: длительность работ в отрезке [5, 25], изначально располагаются на одном процессоре.

Также проводились отдельные запуски при исследовании работы алгоритма с разными законами понижения температуры на «тяжелых» данных. Перебирались (nProc, nTasks, Law), по 2 запуска, результаты усреднялись.

Результаты:

1. Алгоритм работает дольше 1 минуты при числе работ больших 25000 и малом числе процессоров (2-4).



2. На таких «тяжелых» данных алгоритм работает дольше всего с законом понижения температуры Больцмана. Это логично, исходя из того что $\frac{1}{\log x}$ убывает медленнее чем $\frac{1}{x}$. Относительно других алгоритмов решение становится лучше, но лишь на несколько сотых долей процента от изначального значения минимизируемого критерия.

Law comparison on "big" data

