**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Российский химико-технологический университет имени Д.И. Менделеева»**

**Кафедра информационных компьютерных технологий**

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 10**

Выполнил студент группы КС-30 Лобачев Дмитрий Сергеевич

Ссылка на репозиторий: https://github.com/MUCTR-IKT-CPP/DSLobachev\_30/blob/main/Algorithms/Laba10/Laba10.cpp

Приняли: Пысин Максим Дмитриевич

Краснов Дмитрий Олегович

Лобанов Алексей Владимирович

Крашенинников Роман Сергеевич

Дата сдачи: 03.05.2023

**Оглавление**

[Описание задачи 3](#_Toc134042597)

[Описание метода/модели 4](#_Toc134042598)

[Выполнение задачи. 5](#_Toc134042599)

[Результат работы 6](#_Toc134042600)

[Заключение 6](#_Toc134042601)

# Описание задачи

В рамках лабораторной работы необходимо реализовать алгоритм отжига для поиска глобального оптимума(минимума) произвольной функции. В качестве примера взять функцию .

Сам алгоритм выглядит следующим образом:

1. Задать начальное значение (можно выбирать случайно)..
2. Изменить значение температуры при помощи заданной функции , где k это номер итерации, получив температуру .
3. Сгенерировать новую точку , с которой будет сравниваться текущий вариант (возможна случайная генерация, или использование какой-либо функции от температуры).
4. Вычислить значение искомой функции в точке и вычислить разницу между
5. Проверка решения на вероятность принятий:
6. Проверяем критерий завершения, критерием является некоторая температура окончания.

Воспользуемся вариантом быстрого отжига: , где C это случайноe число, сгенерированное при помощи распределения коши.

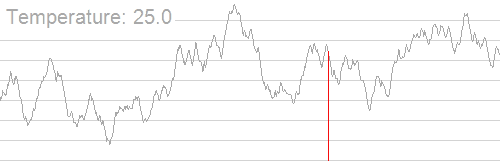
# Описание метода/модели

Алгоритм имитации отжига (англ. Simulated annealing) — общий алгоритмический метод решения задачи глобальной оптимизации, особенно дискретной и комбинаторной оптимизации. Один из примеров методов Монте-Карло.

Алгоритм основывается на имитации физического процесса, который происходит при кристаллизации вещества, в том числе при отжиге металлов. Предполагается, что атомы вещества уже почти выстроены в кристаллическую решётку, но ещё допустимы переходы отдельных атомов из одной ячейки в другую. Активность атомов тем больше, чем выше температура, которую постепенно понижают, что приводит к тому, что вероятность переходов в состояния с большей энергией уменьшается. Устойчивая кристаллическая решётка соответствует минимуму энергии атомов, поэтому атом либо переходит в состояние с меньшим уровнем энергии, либо остаётся на месте. (Этот алгоритм также называется алгоритмом Н. Метрополиса, по имени его автора).

Алгоритм генерирует процесс случайного блуждания по пространству состояний . Решение ищется последовательным вычислением точек пространства : каждая точка, начиная с , «претендует» на то, чтобы лучше предыдущих приближать решение. На каждом шаге алгоритм вычисляет новую точку и понижает значение величины (изначально положительной), понимаемой как «температура».

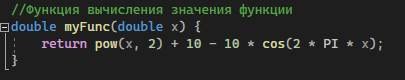
Пример работы алгоритма:



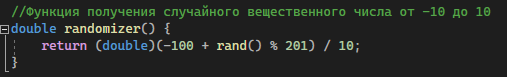
# Выполнение задачи.

Для реализации данного метода сортировки использовался язык программирования C++.

1. Функция вычисления значения функции myFunc()

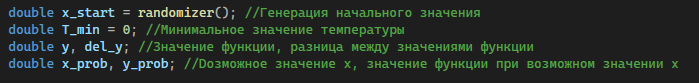


1. Функция получения случайного вещественного числа от -10 до 10 randomizer()



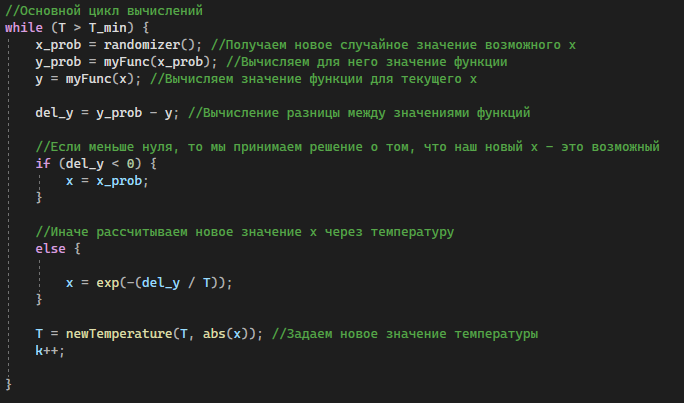
**Выполнение работы**

Задаем начальные значения для работы алгоритма:



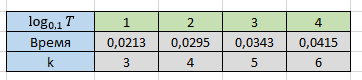


Начинаем производить расчеты в цикле, а именно, генерируем новое случайное значение х и проверяем разницу между значениями функций в старой и новой точках. После этого мы принимаем решение, если данная разница меньше нуля, то нашим новым значением х станет то сгенерированное значение, если же нет, то вычисляем новое значение по формуле и после этого вычисляем новое значение температуры:



# Результат работы

В ходе выполнения работы были получены следующие значения для Tmin = {0.1, 0.01, 0.001, 0.0001, 0}



Также были выполнены расчеты при Tmin = 0:



# Заключение

По результатам работы можно сделать вывод, что при уменьшении минимального значения температуры увеличивается количество итераций и, соответственно, итоговое затраченное время, которое требуется для окончания вычислений. Это объясняется тем, что температура в данной задаче может выступать как точность расчетов, что как раз и влияет на время вычисления.