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 № 6

Выполнил студент группы ....КС-30 ....Сидоров Сергей Александрович
Ссылка на репозиторий: https://github.com/MUCTR-IKT-CPP/Sidorov.S.A_KS-30_2sem/tree/main/lab6

Приняли: .... Пысин Максим Дмитриевич
....Краснов Дмитрий Олегович

Дата сдачи: 01.06.21

Оглавление

Описание задачи.

Реализовать алгоритм отжига для поиска глобального оптимума(минимума) произвольной функции. В качестве примера взять функцию $F(x) = x^2 + 10 - 10cos(2\pi x)$. Сам алгоритм выглядит следующим образом:

1) Задать начальное значение (можно выбирать случайно).
2) Изменить значение температуры при помощи заданной функции t_k, где k это номер итерации, получив температуру t_k.
3) Сгенерировать новую точку x_{k+1}, с которой будет сравниваться текущий вариант (возможна случайная генерация, или использование какой-либо функции от температуры).
4) Вычислить значение искомой функции $f(x)$ в точке x_{k+1} и вычислить разницу между $f(x_{k+1}) - f(x_k) = dF$.
5) Проверка решения на вероятность принятий: 1 при $dF < 0$; $P(x_k, x_{k+1}) = exp(-\frac{dF}{t_k})$
6) Проверяем критерий завершения, критерием является некоторая температура окончания.

Воспользуемся вариантом быстрого отжига: $t_k = \frac{T_0}{k} A(x) = x + TC(0,1)$, где C это случайно число сгенерированное при помощи распределения коши, в C++.

Описание модели.

Алгоритм имитации отжига (англ. simulated annealing) — эвристический алгоритм глобальной оптимизации, особенно эффективный при решении дискретных и комбинаторных задач.

Алгоритм вдохновлён процессом отжига в металлургии — техники, заключающейся в нагревании и контролируемом охлаждении металла, чтобы увеличить его кристаллизованность и уменьшить дефекты. Симулированние отжига в переборных задачах может быть использовано для приближённого нахождения глобального минимума функций с большим количеством свободных переменных.

Алгоритм вероятностный и не даёт почти никаких гарантий сходимости, однако хорошо работает на практике при решении NP-полных задач.

Для примера будем рассматривать задачу коммивояжёра:

- Есть n городов, соединённых между собой дорогами. Необходимо проложить между ними кратчайший замкнутый маршрут, проходящий через каждый город только один раз.

Пусть имеется некоторая функция $f(x)$ от состояния x, которую мы хотим минимизировать. В данном случае x это перестановка вершин (городов) в том порядке, в котором мы будем их посещать, а $f(x)$ это длина соответствующего пути.

Возьмём в качестве базового решения какое-то состояние x_0(например, случайную перестановку) и будем пытаться его улучшать.

Введём температуру t — какое-то действительное число (изначально равное единице), которое будет изменяться в течение оптимизации и влиять на вероятность перейти в соседнее состояние.

Пока не придём к оптимальному решению или пока не закончится время, будем повторять следующие шаги:

1) Уменьшим температуру $t_k = T(t_{k-1})$.

2) Выберем случайного соседа x — то есть какое-то состояние y, которое может быть получено из x каким-то минимальным изменением.
3) С вероятностью $p(f(x), f(y), t_k)$ сделаем присвоение $x \leftarrow y$.

В каждом шаге есть много свободы при реализации. Основные эвристические соображения следующие:

1) В начале оптимизации наше решение и так плохое, и мы можем позволить себе высокую температуру и риск перейти в состояние хуже. В конце наоборот — наше решение почти оптимальное, и мы не хотим терять прогресс. Температура должна быть высокой в начале и медленно уменьшаться к концу.
2) Алгоритм будет работать лучше, если функция $f(x)$ «гладкая» относительно этого изменения, то есть изменяется не сильно.
3) Вероятность должна быть меньше, если новое состояние хуже, чем старое. Также вероятность должна быть больше при высокой температуре.

Например, можно действовать так:

1) $t_k = \gamma * t_{k-1}$, где γ это какое-то число, близкое к единице (например, 0.99). Оно должно зависеть от планируемого количества итераций: оптимизация при низкой температуре почти ничего не будет менять.
2) В случае с перестановками этим минимальным изменением может быть, например, своп двух случайных элементов.
3) Если y не хуже, то есть $f(y) \leq f(x)$, то переходим в него в любом случае. Иначе делаем переход в y, с вероятностью $p = e^{\frac{f(x)-f(y)}{t_k}}$ – это экспонента отрицательного числа, и она даст вероятность в промежутке (0, 1).

В выборе конкретных эвристик не существует «золотого правила». Все компоненты алгоритма сильно зависят друг от друга и от задачи.

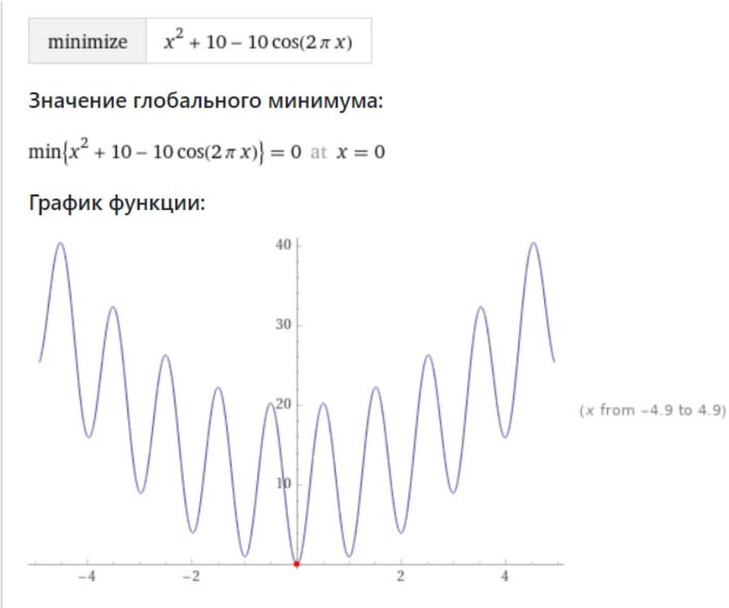
Выполнение задачи.

Задаем начальное значение. Изменяем значение температуры при помощи заданной функции t_k, где k это номер итерации, получив температуру t_k. Генерируем новую точку x_{k+1}, с которой будет сравниваться текущий вариант.

Вычисляем значение искомой функции $f(x)$ в точке x_{k+1} и вычислим разницу между $f(x_{k+1}) - f(x_k) = dF$.

Проверяем решение на вероятность принятий, проверяем критерий завершения, критерием является некоторая температура окончания.

Заданная функция будет иметь следующий вид:



Заключение.

В данной лабораторной работе был реализован метод отжига. Метод работает правильно с некоторой погрешностью. Для более точного результата следует подбирать параметры.