Метод имитации отжига для задачи коммивояжёра

Денисов Н.С.

Кафедра исследования операций

весна 2021 г.

Постановка задачи

Пусть имеется множество городов, каждый из которых представлен точкой на плоскости с координатами (x, y). Тогда задача коммивояжёра — обойти все города с наименьшими затратами, побывав в каждом из них только один раз.

Постановка задачи

Возможные варианты метрик между произвольными городами x_i и x_j:

- $\rho(x_i, x_j) = t_{ij}$ — время, необходимое для перемещения из города x_i в город x_j.
- $\rho(x_i, x_j) = p_{ij}$ — стоимость пути из города x_i в город x_j.
- $\rho(x_i, x_j) = c_{ij}$ — расстояние между городами x_i и x_j.

Остановимся на последней метрике, так как её вычисление инвариантно относительно пары городов, в то время как первая и вторая метрики требуют дополнительных сведений — например, скорость передвижения или расценки на перемещение соотвественно. Однако, алгоритм решения поставленной задачи не зависит от выбора метрики и требует лишь адаптации целевой функции.

Описание алгоритма

Алгоритм имитации отжига является эвристическим и описывает реальный физический процесс, происходящий в металлах при закалке. Алгоритм имитации отжига похож на градиентный спуск, но за счёт случайности выбора промежуточной точки должен попадать в локальные минимумы реже, чем градиентный спуск.

Описание алгоритма

В своей работе алгоритм использует четыре функции

Описание алгоритма

Целевая функция (функция суммарного расстояния):

$$E_i = E(x^i) = \sum_{1}^{|C|-1} c_{k\ k+1} + \sqrt{(x_{|C|} - x_1)^2 + (y_{|C|} - y_i)^2},$$

где i — номер текущей итерации, C — множество городов, а $|C|$ — его мощность.

Описание алгоритма

Функция убывания температуры (определеяет количество итераций и скорость изменения температуры):

$$Q_i = \frac{T_{max}}{i},$$

где i — номер текущего шага.
Критиерий продолжения итераций:

$$Q_i \geq T_{min}$$

Описание алгоритма

Функция генерации очередной последовательности городов:

$$Ax^t = x^{t+1},$$

где A — некий оператор, $A : \mathbb{R}^n \to \mathbb{R}^n$, который каким-то образом преобразует текущую последовательность городов.

В программной реализации — это перестановка двух случайных компонент вектора последовательности городов.

Описание алгоритма

Функция вычисления вероятности принятия новой последовательности городов:

$$P(\overline{x^*} \to \overline{x_{i+1}} | \overline{x_i}) = \begin{cases} 1 & , F(\overline{x^*}) - F(\overline{x_i}) < 0 \\ exp(-\frac{F(\overline{x^*}) - F(\overline{x_i})}{Q_i}) & , F(\overline{x^*}) - F(\overline{x_i}) \geq 0 \end{cases}$$

Если новая последовательность городов $\overline{x^*}$ даёт улучшение целевой функции, то она принимается однозначно. Иначе, она принимается с некоторой вероятностью $p \in (0, 1)$, которая убывает с увеличением номера итерации (следует из генерации Q_i).

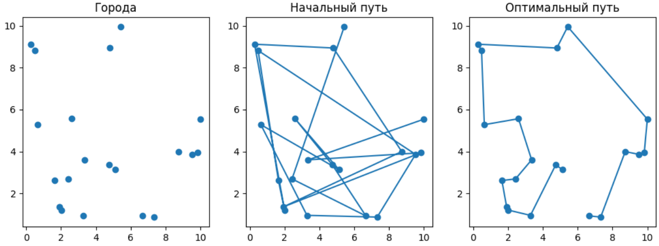
Описание алгоритма

Последовательность шагов работы алгоритма:

1. Случайным образом генерируем начальную последовательность городов — вектор неповторяющихся натуральных чисел x^0 (номера городов). Вычисляем целевую функцию E_0.
2. Очередной вектор x^i, $i = 1, 2, \ldots$ получаем перестановкой двух случайных компонент вектора x^{i-1}. Вычисляем целевую функцию E_i. Вычисляем вероятность P перехода к вектору x^i. Если она равна 1 — принимаем вектор x^i. Иначе генерируем случайное число $\xi \in [0, 1]$. Если $\xi \leq P$, то принимаем вектор x^i, иначе оставляем текущий вектор и повторяем итерации.
3. Продолжаем итерации (пункт 2), пока выполняется условие $Q_i \geq T_{min}$.

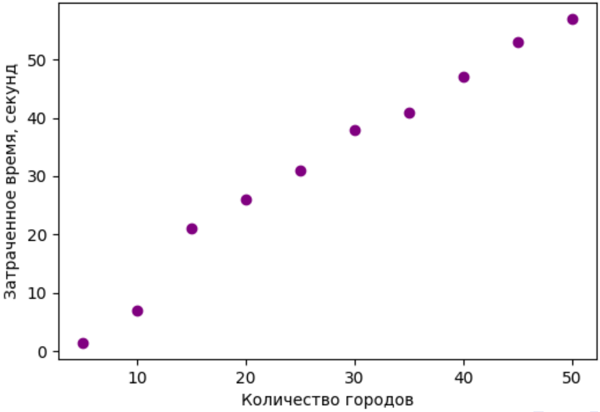
Результат работы алгоритма

На изображении снизу показаны множество городов, начальная (случайная) последоательность городов и конечная, то есть оптимальная, последовательность городов:



Время работы алгоритма

Скорость работы алгоритма находится между $O(\ln(n))$ и $O(n)$, но всё же ближе к линейной, где n — количество городов.



Заключение

Алгоритм имитации отжига — мощный инструмент для задач дискретной оптимизации. Он прост в реализации, обладает хорошей производительностью, успешно выполняет поставленную задачу, а также предоставляет возможность опитимизации некоторых его частей — например, функции убывания температуры или функции генерации новых точек.