электронный журнал
МОЛОДЕЖНЫЙ НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК
Издатель ФГБОУ ВПО "МГТУ им. Н.Э. Баумана". Эл No. ФС77-51038.

 # 03, март 2016

УДК 004.023

Метод имитации отжига и его применение при решении оптимизационных задач

Белянова М.А., *студент*
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Системы обработки информации и управления»

Научный руководитель: Гапанюк Ю.Е. , к.т.н.,доцент,
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Системы обработки информации и управления»
gapyu@bmstu.ru

Впервые информация о методе имитации отжига была опубликована Киркпатриком, Гелаттом и Веччи в журнале “Science” от 13 мая 1983 года. Между статистической механикой, а конкретнее, поведением систем с большим количеством степеней свободы и тепловым равновесием при конечной температуре, и вариативностью методов оптимизации, основанных на комбинаторике (т.е. поиском минимума функции, зависящей от большого количества параметров) существует глубокая и крайне полезная связь. Суть этого метода была позаимствована из статистической физики, а конкретнее, из отжига. В статье [1] отжиг определяется, как процесс, при котором сначала однородное металлическое тело расплавляют и затем медленно понижают температуру, удерживая его на температуре близкой к минимальной, чтобы в металле не возникло дефектов, и чтобы он не образовал стекла без кристаллической решётки, характеризующегося только локально оптимальными структурами.

От того, как расположены атомы в металлическом теле, зависит энергия, которой оно обладает. Тело стремится сделать её количество как можно меньше, но существует ненулевая вероятность, с которой тело сможет перейти в состояние с большей энергией, чем текущее. При достижении теплового равновесия состояние тела уже не может измениться, и на этом факте основывается метод имитации отжига.

Впрочем, существует проблема, заключающаяся в том, что, если просто двигаться по направлению уменьшения энергии, существует возможность застрять в локальном минимуме и остановиться в недостаточно стабильном состоянии. Для того,

чтобы этого не произошло, используется алгоритм, который учитывает возможность перехода в состояние с более высоким значением энергии.

Энергии достаточно сложно придумать очевидный эквивалент в оптимизируемой системе, но при правильном выборе этого параметра-эквивалента можно получить достоверные эвристические решения комбинаторных проблем оптимизации, потому что многие из них могут быть сведены к минимизации или максимизации какой-либо величины.

Алгоритм состоит из нескольких шагов. Каждый шаг состоит из случайного изменения текущего состояния и создания состояния, соседнего текущему. Соседнее состояние определяется нуждами задачи. Как только у нас есть новое решение, генерируется ответ ценовой функции, который определяет, может ли только что сгенерированное состояние заменить текущее. Если изменение ценовой функции отрицательно, то новое состояние сразу же заменяет текущее, в противном случае, оно принимается, если удовлетворяет критерию Метрополиса. Данный критерий основан на вероятности Больцмана [2]:

$$P(E) \propto \mathrm{e}^{-\frac{E}{kT}},$$

где E – энергия системы, k – константа Больцмана, T – текущая температура, а $P(E)$ – вероятность того, что система в состоянии, имеющем энергию E.

Критерий Метрополиса гласит, что если разница между значениями функции стоимости текущего и нового состояния больше или равна нулю, то из равномерного распределения генерируется случайное число δ, и если выполняется, что

$$\delta \leq e^{-\frac{\Delta E}{T}},$$

то созданное состояние принимается, как текущее. Если нет, то текущее состояние не меняется вовсе. Чтобы применить алгоритм к нужной задаче, необходимо выполнить четыре принципиально важных действия:

1) Представить себе вид возможных решений



Рис. 1. Блок-схема алгоритма метода имитации отжига

2) Определить функцию стоимости

3) Определить механизм создания соседних состояний

4) Разработать механизм «остывания»

Чтобы «остывание» велось корректно, необходимо определить четыре параметра: изначальная температура, правило, по которому температура меняется, количество итераций, которые должны быть произведены на каждом шаге и критерий остановки поиска.

Блок-схема алгоритма метода имитации отжига приводится на рис. 1.

Существует несколько вариантов того, как можно организовать механизм «остывания», часто используется геометрическое правило остывания. По этому правилу температура считается по формуле

$T_{i+1} = cT_i$, где c – константа, которая обычно немногим меньше 1. Также температурная функция может быть задана как $T(n) = \frac{T_{max}}{n}$, где n — количество уже пройденных шагов, и в дальнейшем мы будем пользоваться именно этой формулой.

Имитация отжига может быть смоделирована с использованием цепей Маркова. В данной модели мы рассматриваем имитацию отжига как процесс, в котором создаётся цепь Маркова, содержащая последовательность случайных состояний, в которой каждое состояние взаимодействует только с соседними состояниями-кандидатами, а состояния-кандидаты не зависят от состояния, которое было до текущего.

Пусть $\xi(k)$ — случайная величина, показывающая состояние, выпавшее на *k*-том шаге. Тогда введём функцию

$$P_{i,j}(k) = \mathbb{P}(\xi(k) = j \mid \xi(k-1) = i) = \begin{cases} G_{ij}(t_k) A_{ij}(\mathrm{t}_k), & i \neq j; \\ 1 - \sum\limits_{\substack{j \in S \\ j \neq i}} G_{ij}(t_k) A_{ij}(\mathrm{t}_k), & i = j, , \end{cases}$$

где $G_{ij}(t)$ — это вероятность того, что при температуре t состоянием-кандидатом для состояния i будет состояние j, а $A_{ij}(t)$ — вероятность того, что из I будет произведён переход в *j* (при температуре *t*). Выше функция A уже была описана, как

$$A_{ij}(t) = \begin{cases} 1, & E(j) < E(i); \\ e^{\frac{E(i)-E(j)}{t}}, & E(j) \geq E(i). \end{cases}$$

Считается, что все соседние состояния j равновероятны, поэтому функция $G_{ij}(t)$ определяется, как

$$G_{ij}(t) = \begin{cases} \dfrac{1}{|N(i)|}, j \in N(i); \\ 0, \quad j \notin N(i). \end{cases}$$

Выбор нового кандидата обычно не зависит от температуры. Для подходящих параметров известна следующая теорема [3]:

Если есть набор (S, f) и подходящая функция поиска соседних состояний, то после значительного числа переходов с фиксированным значением контрольного параметра, метод имитации отжига найдёт решение $i \in S$ с вероятностью

$$\mathbb{P}_c(\xi = i) = q_i(c) = \frac{\mathrm{e}^{-\frac{E(i)}{t}}}{N_0(t)},$$

где N_0 определена, как

$$N_0(t) = \sum_{j \in S} \mathrm{e}^{-\frac{E(i)}{t}}.$$

Заинтересовавшиеся доказательством могут ознакомиться с ним в источнике [3].

Из этой теоремы следует, что для заданных выше параметров справедливо следующее:

$$\lim_{t \to 0_+} q_i(t) = q_i* = \frac{\chi_{S*}(i)}{|S*|},$$

где S* — это множество оптимальных решений, а характеристическая функция $\chi_M : S \to \{0,1\}$ возвращает 1 только от элементов, принадлежащих M.

Из вышеописанного видно, что метод имитации отжига может найти оптимальное решение с вероятностью 1, если на вход подать бесконечное число проходов.

Применение метода имитации отжига в NP-полных задачах

NP-полной называется задача, которая лежит в классе NP, и к которой сводится любая задача из класса NP. Класс NP — набор проблем разрешимости (то есть, задач вида "определить, принадлежит ли это число такому-то множеству"), которое при некоторых дополнительных сведениях можно решить за полином. Эквивалентно, класс NP можно определить, как содержащий задачи, которые можно за полиномиальное время решить на недетерминированной машине Тьюринга. [4]

Если доказано, что задача NP-полная, то считается, что лучше воспользоваться приближённым алгоритмом вместо того, чтобы пытаться найти алгоритм, решающий её точно. К таковым относится, например, задача поиска гамильтонова пути, задача китайского почтальона, задача раскраски графов и т.п.

В данной статье приводятся результаты выполнения алгоритма метода имитации отжига для задачи четырёх красок и задачи коммивояжёра. Программа, использующая метод имитации отжига для этих задач, написана на языке Python с использованием библиотеки PIL, предназначенной для работы с растровой графикой.

Задача четырёх красок

Пусть у нас есть планарный граф, каждая из вершин которого имеет случайный цвет из четырёх заданных. Необходимо перекрасить вершины этого графа в эти же четыре цвета так, чтобы никакие две вершины одного цвета не были связаны ребром. Эквивалентом энергии в данной задаче пусть служит количество рёбер, соединяющих вершины одного цвета. Множеством состояний назовём множество всех способов раскрасить вершины в четыре цвета. Соседними назовём два состояния, которые отличаются цветом не более, чем одной вершины. Пусть $T_{max} = 20$, $T_{min} = 1 \times 10^{-3}$. Начальные условия и результат выполнения программы можно увидеть на рис. 2 и рис. 3.

Сначала формируется список пар координат точек, затем случайным образом генерируются цвета, после чего эти два списка объединяются в список пар точка-цвет. После этого генерируется список рёбер в формате [точка i, точка j] и список кандидатов. Пока что они пусты. Затем генерируются рёбра. Они генерируются случайным образом, и

если граф получается не планарным, два пересекающихся ребра меняются местами до тех пор, пока не останется ни одной пары рёбер, которые бы пересекались. После этого мы рисуем получившийся граф на холсте (см. рис. 2).

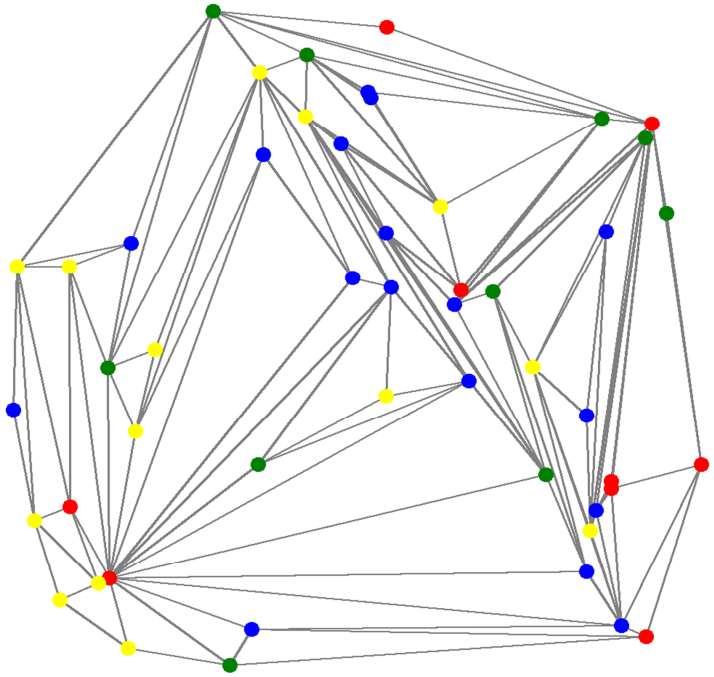


Рис. 2. Начальное состояние задачи четырёх красок

Затем начинается сама работа программы. Мы задаём максимальную и минимальную температуры. Так как было решено вычислять текущую температуру по формуле $T(n) = \frac{T_{max}}{n}$, минимальная температура должна быть строго больше нуля. Номер шага перед началом алгоритма равен единице, считаем следующим образом: 1) выбираем случайное состояние-кандидат, 2) считаем разницу энергий (напомню, энергией в данной задаче считается количество рёбер, соединяющих вершины одного цвета, 3) если она меньше нуля, или если выполняется критерий Метрополиса, переходим в состояние-кандидат, 4) инкрементируем номер шага, 5) высчитываем текущую температуру, 6) повторяем пункты 1-5, пока текущая температура больше минимальной.

После выполнения алгоритма, рисуем на холсте итоговый результат, который представлен в данной статье на рис. 3.

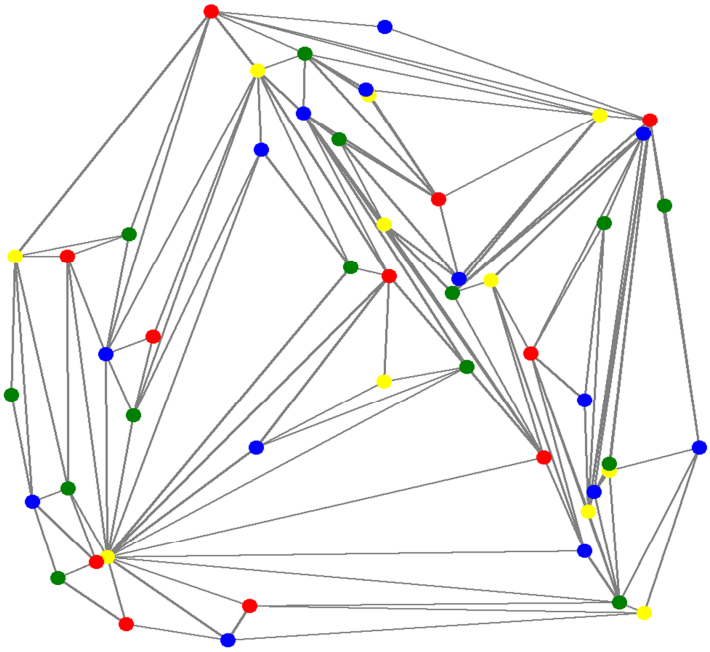


Рис. 3. Конечное состояние задачи четырёх красок

Задача коммивояжёра

На квадрате 10x10 расположено n точек, необходимо найти гамильтонов цикл наименьшей длины, проходящий через все эти точки. Пусть состоянием будет какой-либо гамильтонов цикл через все вершины, а два состояния будут соседними, если одно из них получается инвертированием какого-либо пути в этом цикле. Возьмём за энергию длину цикла. Пусть $T_{max} = 20, T_{min} = 1 \times 10^{-3}$, а в начальном состоянии гамильтонов цикл проходит по вершинам случайным образом.

Опишем работу программы. Сначала зададим граф, как множество точек со случайными координатами $(x;y)$, и зададим проходящий через все точки графа случайный гамильтонов цикл. Выведем изображение этого графа на экран. Зададим начальную и минимальную(конечную) температуры, шаг, который перед началом выполнения цикла равен единице, и, пока текущая температура не будет равна минимальной, выполняем следующие действия: 1) генерируем новое состояние-кандидат, как состояние, у которого рёбра, идущие из двух случайных вершин текущего состояния, поменяны местами (то есть, ребро, идущее из вершины i в вершину $i+1$ и ребро, идущее из вершины j в вершину $j+1$, становятся рёбрами из i в j и из $i+1$ в $j+1$), 2) считаем его энергию, как длину цикла 3) проверяем разницу энергий, 4) если она меньше нуля, или если выполняется критерий Метрополиса, заменяем текущее состояние состоянием

кандидатом, 5) инкрементируем номер шага, 6) высчитываем текущую температуру. Визуализация начальных условий и итоговый граф представлены на рис. 4 и рис. 5.

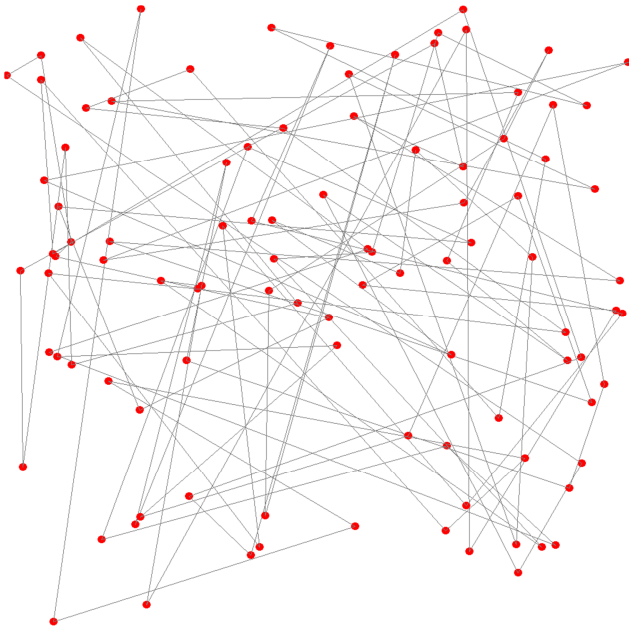


Рис. 4. Начальное состояние задачи коммивояжёра

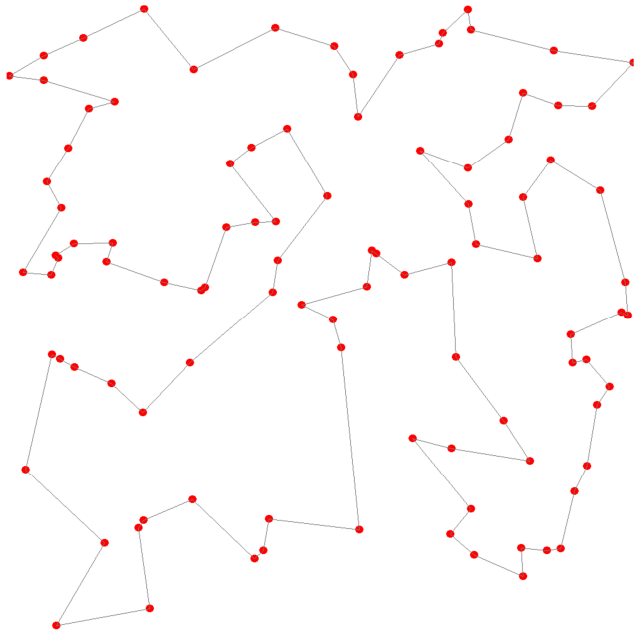


Рис. 5. Конечное состояние задачи коммивояжёра

Можно заметить, что алгоритм приводит результаты работы, похожие на оптимальные. То, насколько близко к оптимальному будет результат, зависит от конечной температуры и функции, определяющей текущую температуру на k-том шаге.

Заключение

Имитация отжига применяется при решении большого количества различных по своей направленности оптимизационных задач. Используя этот метод, необходимо помнить, что точность решения зависит от времени выполнения программы. Но если задачу невозможно решить определённым алгоритмом, имитация отжига достойна того, чтобы обратить на себя внимание за счёт своей гибкости и лёгкости в применении.

Список литературы

[1]. Kirkpatrick S., Gelatt Jr C.D., Vecchi M.P. Optimization by simulated annealing // Science, 1983. no. 220. P. 671-680.

[2]. Burke E.K., Kendall G. Search Methodologies Introductory Tutorials in Optimization and Decision Support Techniques. New-York: Springer, 2005. 620 p.

[3]. Pham D.T., Karaboga D. Intelligent Optimisation Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks. London: Springer, 2000. 500 p.

[4]. Венбо М. Современная криптография: теория и практика: пер. с англ. М.: Издательский дом «Вильямс», 2005. 768 с. [Venbo M. Modern Cryptography: Theory and Practice. New Jersey: Prentice Hall PTR, 2004. 648 p.]