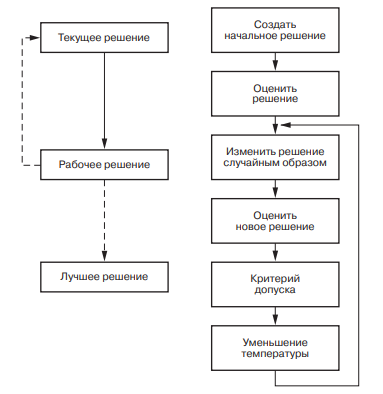
Алгоритм отжига

Мы изучим метод оптимизации, который называется отжигом, или симуляцией восстановления (Simulated annealing).

Как ясно из названия, метод поиска моделирует процесс восстановления. Восстановление – это физический процесс, который заключается в нагреве и последующем контролируемом охлаждении субстанции. В результате получается прочная кристаллическая структура, которая отличается от структуры с дефектами, образующейся при быстром беспорядочном охлаждении. Структура здесь представляет собой кодированное решение, а температура используется для того, чтобы указать, как и когда будут приниматься новые решения.

Свойства структуры зависят от коэффициента охлаждения после того, как субстанция была нагрета до точки плавления. Если структура охлаждалась медленно, будут сформированы крупные кристаллы, что очень полезно для строения субстанции. Если субстанция охлаждается скачкообразно, образуется слабая структура. Чтобы расплавить материал, требуется большое количество энергии. При понижении температуры уменьшается и количество энергии. Чтобы яснее представить процесс восстановления, рассмотрим следующий пример. «Взбалтывание» при высокой температуре сопровождается высокой молекулярной активностью в физической системе. Представьте себе, что вы взбалтываете емкость, в которой находится какая-то поверхность сложной формы. Внутри емкости также имеется шарик, который пытается найти точку равновесия. При высокой температуре шарик может свободно перемещаться по поверхности, а при низкой температуре «взбалтывание» становится менее интенсивным и передвижения шарика сокращаются. Задача заключается в том, чтобы найти точку минимального перемещения при сильном «взбалтывании». При снижении температуры уменьшается вероятность того, что шарик выйдет из точки равновесия. Именно в таком виде процесс поиска заимствуется из восстановления.



Для большинства проблем начальное решение будет случайным. На самом первом шаге оно помещается в текущее решение (Current solution). Другая возможность заключается в том, чтобы загрузить в качестве начального решения уже существующее, возможно, то самое, которое было найдено во время предыдущего поиска. Это предоставляет алгоритму базу, на основании которой выполняется поиск оптимального решения проблемы.

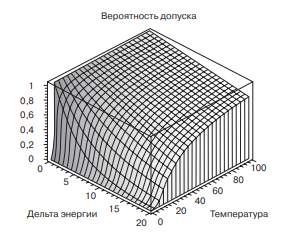
Оценка решения состоит из декодировки текущего решения и выполнения нужного действия, позволяющего понять его целесообразность для решения данной проблемы. Обратите внимание, что закодированное решение может просто состоять из набора переменных. Они будут декодированы из существующего решения, а затем эффективность решения будет оценена на основании того, насколько успешно удалось решить данную задачу.

Поиск решения начинается с копирования текущего решения в рабочее решение (Working solution). Затем мы произвольно модифицируем рабочее решение. Как именно модифицируется рабочее решение, зависит от того, каким образом оно представляется (кодируется). Представьте себе кодировку задачи коммивояжера, в которой каждый элемент представляет собой город. Чтобы выполнить поиск по рабочему решению, мы берем два элемента и переставляем их. Это позволяет сохранить целостность решения, так как при этом не происходит повторения или пропуска города. После выполнения поиска рабочего решения мы оцениваем решение, как было описано ранее. Поиск нового решения основан на методе Монте-Карло (то есть случайным образом).

На этом этапе алгоритма у нас имеется два решения. Первое – это наше оригинальное решение, которое называется текущим решением, а второе – найденное решение, которое именуется рабочим решением. С каждым решением связана определенная энергия, представляющая собой его эффективность (допустим, что чем ниже энергия, тем более эффективно решение). Затем рабочее решение сравнивается с текущим решением. Если рабочее решение имеет меньшую энергию, чем текущее решение (то есть является более предпочтительным), то мы копируем рабочее решение в текущее решение и переходим к этапу снижения температуры. Однако если рабочее решение хуже, чем текущее решение, мы определяем критерий допуска, чтобы выяснить, что следует сделать с текущим рабочим решением. Вероятность допуска основывается на уравнении (которое, в свою очередь, базируется на законе термодинамики):

P(δE) = exp(–δE/T)

При высокой температуре (свыше 60 °С) плохие решения принимаются чаще, чем отбрасываются. Если энергия меньше, вероятность принятия решения выше. При снижении температуры вероятность принятия худшего решения также снижается. При этом более высокий уровень энергии также способствует уменьшению вероятности принятия худшего решения. При высоких температурах симулированное восстановление позволяет принимать худшие решения для того, чтобы произвести более полный поиск решений. При снижении температуры диапазон поиска также уменьшается, пока не достигается равенство при температуре 0°.



После ряда итераций по алгоритму при данной температуре мы ненамного снижаем ее. Существует множество вариантов снижения температуры. В данном примере используется простая геометрическая функция Ti+1 = αTi

Константа α меньше единицы. Возможны и другие стратегии снижения температуры, включая линейные и нелинейные функции.