# ЛОКАЛЬНЫЙ ПОИСК

Для решения оптимизационных задач зачастую используют алгоритмы локального поиска. Алгоритм локального поиска начинает работу от некоторого исходного решения, найденного либо каким-то другим алгоритмом, либо сгенерированного случайным образом. Алгоритм локального поиска представляет собой итеративный процесс. На каждой итерации происходит переход от текущего решения к соседнему решению с меньшим или большим (в зависимости от того, какая рассматривается задача: минимизации или максимизации) значением целевой функции. Процесс повторяется до тех пор, пока не будет достигнут локальный оптимум. При этом соседнее решение не обязательно должно быть наилучшим в окрестности. Также критерий оценки оптимальности решений не должно изменяться на протяжении всего процесса.

Для любого решения *s* оптимизационной проблемы должно быть задано некоторое множество *N*(*s*) соседних решений, которые незначительно отличаются от *s*. *N*(*s*) называется окрестностью *s*. Семейство всех окрестностей называется структурой окрестностей. В более строгой формулировке структурой окрестностей называется отображение *N*, которое каждому допустимому решению *s* задачи ставит в соответствие множество решений *N*(*s*). Однако, отношение соседства не всегда симметрично, т.е. если *s* является соседом *s*′, то не обязательно *s*′ будет соседом *s*.

Обозначим через *m*(*s*) значение целевой функции некоторой оптимизационной проблемы при найденном решении *s*. Решение *s* называется локально-минимальным (локально-максимальным) по отношению к структуре окрестностей *N*, если

*m*(*s*) ≤ *m*(*s*′) (*m*(*s*) ≥ *m*(*s*′)) для всех *s*′ ∈ *N*(*s*).

Другими словами *s* является локально-оптимальным, если оно лучшее в *N*(*s*).

Для решения *s* поиск лучшего решения в окрестности *N*(*s*) может быть осуществлен двумя способами:

а) выбирается лучшее решение во всей окрестности;

б) выбирается первое лучшее решение, когда окрестность каким-либо образом перебирается. При этом перебор сразу же прекращается, как только будет найдено первое лучшее решение.

Локальным поиском называется алгоритм, который для оптимизационной задачи просматривает подмножество допустимых решений до момента, когда будет найдено локально-оптимальное решение относительно структуры окрестностей *N*.

Структуру окрестностей также можно задать в помощь ориентированного графа.

Графом соседства (окрестностей) для оптимизационной задачи называется взвешенный ориентированный граф *GN*(*x*) = (*V*(*x*), *A*) с множеством вершин *V*(*x*), совпадающим с множеством допустимых решений задачи, и множеством дуг *A* = {(*s*1, *s*2) | *s*2 ∈ *N*(*s*1)}. Веса в графе приписаны вершинам и равны соответствующим значениям целевой функции.

Граф окрестностей (neighborhood graph) еще называют ландшафтом целевой функции (landscape, fitness landscape). При определении структуры окрестностей *N* важно следить за тем, чтобы граф *GN*(*x*) был строго связен, т.е. для каждой пары вершин *s* и *s*′ существовал путь из *s* в *s*′. Если такое свойство не выполняется, то стремятся получить свойство вполне связности, когда из любой вершины существует путь в вершину *s*\* с минимальным (максимальным) значением целевой функции. Если и это свойство не выполняется, то нет уверенности в том, что будет достигнут глобальный оптимум. Нужно либо ограничиться локальными оптимумами, либо переопределить функцию окрестности.

Переход от одного состояния к другому в графе есть дуга. Локальному поиску соответствует путь в графе, который заканчивается в вершине sr ∈ V(x) такой, что m(sr) ≤ m(s) (m(sr) ≥ m(s)) ∀s, что (sr, s) ∈ A. Однако, локальный поиск может исследовать экспоненциальное количество решений прежде, чем он найдет оптимум. Чтобы решить эту проблему, часто используют правило остановки.

В общем случае алгоритмы локального поиска имеют следующую схему.

Input: Исходная задача;

Output: Решение s;

begin

s := некоторое допустимое решение s0;

repeat

Выбрать еще не рассмотренное решение s′ ∈ N(s);

if m(s′) < m(s) (m(s′) > m(s))

s := s′;

until все решения из N не будут рассмотрены;

end.

Поведение алгоритма локального поиска зависит от следующих факторов:

1. Структуры окрестностей N. Размер окрестности выбирается на основе компромисса между целью получения хорошего улучшения при переходе к новому состоянию и целью сокращения времени просмотра одной окрестности. Обычно окрестность порождается с помощью некоторой операции локального изменения текущего решения.
2. Начального решения s0. Его можно находить с помощью некоторого алгоритма, который выдает хорошее приближение, или с помощью случайной генерации. Замечено, что первый случай предпочтительнее, поскольку время спуска к оптимуму сокращается.
3. Стратегия выбора новых решений. Как и было сказано выше, следующее решение выбирается либо как лучшее в данной окрестности, либо как первое лучшее по отношению к текущему решению.

Алгоритмы локального поиска широко применяются для решения NP-трудных задач дискретной оптимизации. Кроме того, некоторые полиномиально разрешимые задачи могут легко решаться локальным спуском. Ниже приведены примеры таких задач.

1. Линейное программирование. Алгоритм симплекс-метода можно представить как движение по вершинам многогранника допустимой области. Вершина не является оптимальной, если существует смежная с ней вершина с меньшим значением целевой функции. Базисное допустимое решение не является оптимальным, если и только если оно может быть улучшено локальным изменением базиса, т.е. заменой одной базисной переменной на небазисную.
2. Минимальное остовное дерево. Остовное дерево не является оптимальным, если локальной перестройкой, добавляя одно ребро и удаляя из образовавшегося цикла другое ребро, можно получить новое остовное дерево с меньшим суммарным весом. Операция локальной перестройки задает отношение соседства на множестве остовных деревьев.
3. Максимальное паросочетание. Паросочетание не является максимальным, если существует увеличивающийся путь. Два паросочетания называют соседними, если их симметрическая разность образует путь. Аналогичные утверждения справедливы для взвешенных паросочетаний, совершенных паросочетаний минимального веса, задач о максимальном потоке.