

УДК 347.717
ЛОКАЛЬНАЯ ОПТИМИЗАЦИЯ РЕШЕНИЯ ЗАДАЧИ
КОММИВОЯЖЕРА С ДОПОЛНИТЕЛЬНЫМИ ОГРАНИЧЕНИЯМИ

В. С. БУТОМА

Белорусский государственный университет информатики
и радиоэлектроники
Минск, Беларусь

А. М. БУТОМА

Белорусско-Российский университет
Могилев, Беларусь

Задача коммивояжера является одной из самых известных задач в комбинаторной оптимизации. В классической формулировке данная задача звучит следующим образом: необходимо найти самый выгодный маршрут, проходящий через все города с возвращением в стартовый. Выгодность маршрута определяется минимальным расстоянием и суммарным штрафом.

Известно, что на текущий момент для решения классической задачи коммивояжера применяется алгоритм LKH (Lin-Kernighan Heuristic), который является лучшим по скорости и точности при достаточно большом количестве вершин графа. Однако в реальной жизни часто встречаются случаи, когда задачу нельзя свести к классической модели, и в связи с этим невозможно использовать оптимальный алгоритм в чистом виде.

Для примера рассмотрим не классическую задачу коммивояжера, а задачу, где на каждое 10-е ребро в пути накладывается дополнительный штраф в 10 %, если ребро идет из города не с простым индексом. Данная задача была представлена на ежегодном соревновании по оптимизации Traveling Santa 2018 prime path.

Для решения данной задачи предлагается использовать следующие шаги.

Применим LKH на задаче без дополнительных ограничений, т. е. не учитывая штрафы, и получим базисный путь P для дальнейших преобразований (путь P представляет собой перестановку длины N , где N – это количество городов).

Далее применим две эвристики локальной оптимизации. В общем случае их можно применять как угодно, но для данной задачи будем увеличивать штраф монотонно до максимального. Это делается для следующего:

– сохранение оптимального пути без штрафов на протяжении большого количества итераций;

– при оптимизации и скрещивании путей будут выбираться подпути большего размера.

Первая фаза заключается в нахождении абсолютно точного решения для окна в уже имеющемся пути. Для оптимизации в окне воспользуемся методом динамического программирования по битовым маскам. Пусть $d[mask][i]$ – величина оптимального пути, который заканчивается вершиной под индексом i ($i \in [1 \dots (N - k)]$, где k – это размер окна, в котором применяется оптимизация, $k \leq 25$). При этом посещенные города задаются через $mask$, а именно через значение битов, т. е., если $mask_j = 1$, то город j участвует в пути. В классическом виде рекуррентное соотношение выглядит следующим образом:

$$d[mask][i] = \min(d[mask][i], d[mask \oplus 2^i][j] + cost_{ij}),$$

где $cost_{ij}$ – длина пути между пунктами i и j .

Во второй фазе применим модифицированные k -opt эвристики. В нашей задаче применим 5-opt. Суть заключается в том, чтобы сгенерировать множество всех пятерок, которые удовлетворяют условию, а затем для каждой пятерки найти оптимальный путь.

Пятерки будем генерировать следующим образом: для каждого $i \in [1 \dots N]$ находим множество ближайших соседей (это можно сделать, например, через k -мерное дерево, при этом критерием отбора будет принадлежность области вокруг точки $i \in [1 \dots N]$). Размер области подбирается эмпирически. По каждому набору пятерок индексов $indices$ генерируется множество подпутей из базисного пути P (в общем случае базисный – это текущий оптимальный), т. е. для каждого индекса из $indices$ находится его позиция в P и составляются подпути, перестановки которых могут улучшить итоговый результат. То есть задача сводится снова к нахождению оптимальной перестановки, и эту задачу можно точно решить с помощью динамики по битмаскам, только рекуррентный переход будет немного отличаться от описанного выше.

Данная оптимизация показала себя достаточно эффективной. В частности, с ее помощью на соревновании [1] был получен довольно хороший результат (33 место из 1874).

СПИСОК ЛИТЕРАТУРЫ

1. <https://www.kaggle.com/c/traveling-santa-2018-prime-paths/overview>.