УДК 519.863+УДК 519.854

Методы ветвей и сечений для задачи маршрутизации с ограничением по грузоподъемности

©2012 г. *Е.С.Чеписова*
Нижегородский государственный университет им. Н.И. Лобачевского
mmes@mm.unn.ru

В статье рассматриваются методы решения задачи маршрутизации с ограничением по грузоподъемности, основанные на применении метода ветвей и границ, метода отсечений и их комбинации.

Ключевые слова: задача маршрутизации транспорта, алгоритмы сечений, метод ветвей и границ.

Одной из ключевых функций систем поддержки принятия решений в области транспортной логистики является построение маршрутов различного назначения, эффективных с точки зрения стоимости объезда. Данная статья посвящена исследованию задачи нахождения маршрутов для посещения заданного множества адресов некоторым количеством единиц транспортных средств с обязательным возвращением в начальное местоположение после окончания поездки. Математическая формулировка этой задачи известна как задача маршрутизации транспорта (*Vehicle Routing Problem*). Базовой версией *VRP* является задача маршрутизации с ограничением по грузоподъемности (*CVRP).*

Задача *CVRP* может быть сформулирована следующим образом. Задается полный неориентированный граф $G=(V,E)$, в котором $V=\{0, 1, ..., n\}$- множество всех вершин, 0 - вершина, в которой построенные маршруты должны начинаться и заканчиваться (склад), *A* - множество ребер $\{(i, j) \mid i \neq j\}$. $V_c = V \setminus \{0\}$ - множество из *n* целевых вершин для посещения (вершин клиентов). Стоимость проезда между вершинами *i* и *j* обозначается через c_{ij}, и считается что расходы симметричные, т.е., $c_{ij}=c_{ji}$. Имеется парк идентичных транспортных средств, каждое мощности $Q>0$. Каждый клиент *i* выдвигает целочисленный спрос q_i, $0<q_i \le Q$. Клиент должен быть обслужен одним транспортным средством и транспортное средство не может обслужить множество клиентов, чей спрос превышает его емкость. Требуется построить *K* маршрутов минимальной суммарной стоимости, которые начинаются и заканчиваются в вершине 0, и каждая вершина из V' должна быть включена в маршрут одного и только одного транспортного средства.

Задача *CVRP* является *NP*-трудной и обобщает хорошо известную задачу коммивояжера и задачу об упаковке. Вычислительная сложность *CVRP* обуславливает то, что наиболее интенсивно ведется поиск приближенных алгоритмов [1]. Тем не менее, задачи небольшого и среднего размеров могут быть решены точно. Многие точные алгоритмы основаны на поиске по дереву решений [2]. Для того чтобы, избежать перебор «неперспективных» вариантов, могут быть использованы два типа оценок: нижние границы для значений целевой функции на подмножестве допустимых решений и верхние границы для оптимального значения целевой функции. Для получения таких оценок обычно ставится и решается некоторая оптимизационная задача, называемая оценочной. Часто она получается из данной либо исключением некоторых условий задачи, например, условий целочисленности переменных, либо заменой целевой функции $f(x)$ минорантой $g(x) \leq f(x)$.

Метод ветвей и границ заключается в разбиении задачи на упрощенные подзадачи, путем фиксирования значения переменной ветвления. Каждая из этих подзадач решается тем же способом; т.е. разветвляется другая переменная. На каждом шаге проводятся испытания, чтобы проверить может ли быть удалена вся ветвь, т.е. нет лучшего решения, чем то, известное в настоящее время, которое можно найти в этой ветке. Одним из известных методов ветвей и границ для задачи *CVRP* является метод *K*-дерева, аналогичный алгоритму 1-дерева для симметричной задачи коммивояжера [3,4]. Пусть количество используемых транспортных средств фиксировано и равно *K*. Симметричная задача *CVRP* моделируется как задача определения *K*-дерева, то есть связного графа с *K* циклами, такого, что вершина 0 имеет степень 2*K*, степени вершин-клиентов равны двум и выполняются дополнительные ограничения на грузоподъемность транспортного средства. В качестве ослабленной задачи рассматривается задача построения *K*-дерева минимального веса, вершина 0 в котором имеет степень $2K$:

$$\min \sum_{e \in E} c_e x_e ,$$

$$\sum_{e \in \delta(0)} x_e = 2K,$$

$$\sum_{e \in \delta(S)} x_e \geq 1, \qquad S \subseteq V \setminus \{0\}, S \neq \varnothing,$$

$$x_e \in \{0,1\}, \forall e \in E.$$

Поскольку любое допустимое решение задачи маршрутизации представляет собой *K*-дерево, минимальный вес *K*-дерева дает оценку снизу на оптимальное значение целевой функции задачи *CVRP*.

Асимптотически точную нижнюю оценку можно получить, используя постановку задачи маршрутизации в виде задачи разбиения множества [5]. Пусть $\{1,...,R\}$ - индексное множество всех возможных маршрутов, а c_r - длина маршрута *r*. Для каждого клиента *i* =1,2,…,*n* и каждого маршрута *r* = 1,2,…,*R* полагают, α_{ir} =1, если клиент *i* попадает в маршрут *r* , α_{ir} =0 в противном случае. Также для каждого *r* = 1,2,…*R* пусть y_r = 1, если маршрут *r* входит в оптимальное решение и y_r=0 в противном случае. В формулировке разбиения множества для *VRP* задача заключается в выборе множества допустимых маршрутов минимальной стоимости, при условии, что каждый клиент включается в несколько маршрутов. Задача разбиения множества имеет вид:

$$Min\sum_{r=1}^{R} c_r y_r$$

$$\sum_{r=1}^{R} \alpha_{ir} y_r \geq 1, \forall i = 1,2,...,n$$

$$y_r \in \{0,1\}, \forall r = 1,2,..., R.$$

Линейная релаксация этой задачи может быть решена с помощью метода генерации столбцов [6]. Если решение задачи является целочисленным, то оно будет оптимальным решением задачи *CVRP* , в противном случае полученное решение можно использовать для применения метода ветвей и границ. В качестве переменной ветвления используется нецелое значение y_r для некоторого маршрута *r*, и в подзадачах y_r равна либо 1, либо 0. Решение, найденное с помощью этого метода, будет лучшим целым решением среди всех решений, составленных из столбцов, присутствующих в конце шага метода генерации столбцов. Это решение не обязательно будет оптимальным для *VRP*, но оно может быть близким к нему.

Другой подход к решению поставленной задачи целочисленного программирования заключается в применении метода отсечений [7]. Первоначально решается обычная задача линейного программирования, полученная из исходной задачи отбрасыванием требования целочисленности. Если полученное решение является целочисленным, то оно будет также решением исходной задачи. Если нет, то к ограничениям исходной задачи добавляется новое линейное ограничение, обладающее двумя свойствами: полученное нецелочисленное решение ему не удовлетворяет; все целочисленные точки допустимого множества исходной задачи ему удовлетворяют. Таким образом, это ограничение отсекает» данное дробное решение. Различные способы построения отсечений для задачи маршрутизации приводятся в [5].

Перспективным методом решения задачи *CVRP* является метод, в котором сочетается и поиск по дереву решений и метод отсечений [8]. Для решения *CVRP* методом ветвей и сечений используют двухиндексную постановку. Пусть $V_c = V \setminus \{0\}$ - множество клиентов. Для каждого множества $S \subseteq V_c$ через $q(S)$ обозначается $\sum_{i \in S} q_i$, $\delta(S)$ - это множество ребер из графа G ровно с одной концевой вершиной в *S*, *E(S)* – это множество ребер из *G* с несколькими концевыми вершинами в *S*, и $r(S)$ - минимальное число транспортных средств, необходимых для обслуживания клиентов из *S*. Пусть x_{ij} характеризует то, сколько раз автомобиль проходит между вершинами i и j, для заданного множества $F \subseteq E$, пусть $x(F)$ обозначает $\sum_{e \in F} x_e$. Тогда постановка целочисленного программирования имеет вид:

$$\min \sum_{e \in E} c_e x_e$$

$$x(\delta(\{i\})) = 2 \quad (i = 1,\dots,n) \tag{1}$$

$$x(\delta(S)) \geq 2r(S) \quad (S \subseteq V_c, |S| \geq 2) \tag{2}$$

$$x_{ij} \in \{0,1\} \quad (1 \leq i < j \leq n) \tag{3}$$

$$x_{ij} \in \{0,1,2\} \quad (i = 0,\ j = 1,\dots,n). \tag{4}$$

Равенство (1) обеспечивает, то, чтобы клиент посещался только один раз. Неравенство (2) накладывает ограничения на мощности транспортных средств и гарантирует, чтобы маршруты были связаны между собой. Постановка останется в силе, если заменить $r(S)$ в правой части (2) на очевидную нижнюю границу $k(S) = [q(S)/Q]$, которую дает, так называемое, округленное неравенство мощности. Ограничения (3) и (4) – условия целочисленности.

На каждом шаге алгоритма метода ветвей и границ решается линейная релаксация задачи (1)-(4). Полученная при этом нижняя граница затем может быть улучшена добавлением отсечений. Выделяют следующие виды неравенств [8]:

- неравенство ёмкости (*RCIs*);

- оформленные неравенства ёмкости (*FCIs*):

$$x(\delta(S)) + \sum_{i=1}^{p} x(\delta(S_i)) \geq 2r(S,\Omega) + 2\sum_{i=1}^{p} r(S_i)\,;$$

- неравенство созвездия и частичного созвездия:

$$\alpha x(E(N)) + \beta x(E(N:S)) \leq \gamma,\ \ \alpha x(E(N)) + \beta x(E(C:S)) \leq \gamma\,;$$

- отсечения Гомори:

$$\sum_{i \in NB} \min\left\{ \frac{f(\alpha_i)}{f(x_j^*)}, \frac{1 - f(\alpha_i)}{1 - f(x_j^*)} \right\} x_i \geq 1.$$

Для поиска неравенств конкретного класса, нарушаемых для дробного решения релаксированной задачи используются точный и эвристические алгоритмы сечений.

Список литературы

1. Laporte G. The vehicle routing problem: an overview of exact and approximate algorithms.// Eur. J. Opl. – 1992. – Res. 59. – pp. 345–358.

2. Кристофидес Н. Теория графов. Алгоритмический подход. – М.: Мир, 1978. – 432 с.

3. Christofides N., Mingozzi A., Toth P. Exact algorithms for the vehicle routing problem based on the spanning tree and shortest path relaxations.// Math. Program. 20. – 1981. – pp. 255–282.

4. Fisher M. Optimal solution of vehicle routing problem using minimum k-trees.// Operations Res. 42. – 1994. – pp. 626–642.

5. Bramel J, Simchi-Levi D. The Logic of Logistics: Theory, Algorithms and Applications for Logistics Management. – NY.: Springer-Verlag. -1993. - 281 p.

6. Мину М. Математическое программирование. - М.: Наука, 1990. – 488 с.

7. Корбут А.А., Финкельштейн Ю.Ю. Дискретное программирование. - М.: Наука, 1969.- 368 с.

8. Lysgaard J., Letchford A. N., Eglese R. W. A new branch-and-cut algorithm for the capacitated vehicle routing problem. // Math. Program. – 2004. - Ser. A 100. – pp. 423–445.