

УДК 658.52.011.56

ОБ ИСПОЛЬЗОВАНИИ АЛГОРИТМА ЛИТЛА ДЛЯ РЕШЕНИЯ ЗАДАЧИ О КОММИВОЯЖЕРЕ ПРИ РАЗРАБОТКЕ ПРИГОРОДНЫХ МАРШРУТОВ

**С.А. Канцедал, профессор, д.т.н., Западнодонбасский частный институт
экономики и управления, г. Павлоград, М.В. Костикова, доцент, к.т.н.,
И.В. Скрипина, ст. преподаватель, ХНАДУ**

Аннотация. Описывается алгоритм Литла, предложенный для точного решения задачи о коммивояжере. Представлены характеристики времени решения задач о коммивояжере алгоритмом Литла и его модификацией для случаев с симметричной и несимметричной матрицей расстояний. Даны границы размеров задач, для которых целесообразно использование алгоритма при составлении маршрутной сети.

Ключевые слова: коммивояжер, расстояние, алгоритм, матрица, границы, эксперимент, маршрутная сеть.

ПРО ВИКОРИСТАННЯ АЛГОРИТМУ ЛІТЛА ДЛЯ РОЗВ'ЯЗАННЯ ЗАДАЧІ ПРО КОМІВОЯЖЕРА ПРИ РОЗРОБЦІ ПРИМІСЬКИХ МАРШРУТІВ

**С.А. Канцедал, професор, д.т.н., Західнодонбаський приватний інститут економіки
і управління, м. Павлоград, М.В. Костікова, доцент, к.т.н.,
І.В. Скрипіна, ст. викладач, ХНАДУ**

Анотація. Описується алгоритм Літла, запропонований для точного розв'язання задачі про комівояжера. Представлено характеристики часу розв'язання задач про комівояжера алгоритмом Літла і його модифікацією для випадків із симетричною і несиметричною матрицею відстаней. Надано межі розмірів задач, для яких є доцільним використання алгоритму при складанні маршрутної мережі.

Ключові слова: комівояжер, відстань, алгоритм, матриця, межі, експеримент, маршрутна мережа.

ON THE USE OF LYTLE'S ALGORITHM FOR SOLVING TRAVELING SALESMAN PROBLEM AT DEVELOPING SUBURBAN ROUTE

**S. Kantsedal, Professor, Doctor of Technical Science, West Donbass Institution of
Economics and Management, Pavlograd, M. Kostikova, Associate Professor,
Candidate of Technical Science, I. Skrypina, senior teacher, KhNAHU**

Abstract. Lytle's algorithm is described as proposed for an accurate solution of the salesman Problem. Statistical characteristics of solution duration with lytle's algorithm of some problems and of their modifications are specified. On the basis of the results obtained the limits for the algorithm practical specification in the preparation of the route network are given.

Key words: salesman, distance, algorithm, matrix, boundaries, experiment, route network.

Введение

Рынок услуг автоперевозчиков на пригородных и районных маршрутах в своем развитии

претерпел традиционные для большинства сфер отечественной экономики этапы – от хаоса до упорядочения. Процветающая в действующем СССР система крупных авто-

транспортных предприятий (АТП) в новых экономических условиях себя не оправдала. Большинство из этих АТП работали убыточно либо вовсе прекратили свою деятельность. На рынке начали обосновываться частные предприниматели, которые более адекватно реагировали на изменение ситуации. По мере того, как у частного бизнеса появлялось больше средств, менялся и рынок, как и положено во всем цивилизованном мире, – в сторону укрупнения. Сегодня на транспортном рынке работают серьезные инвестиции, появились предприятия, имеющие в своем парке от 20 до 100 машин. Либерализация экономики и переход к свободному рынку транспортных услуг коренным образом изменили механизм взаимоотношений всех участников перевозочного процесса. В настоящее время госадминистрации различных областей начинают работу по составлению единых маршрутных сетей области, обслуживать которые перевозчики будут только через открытые конкурсы.

Для успешного проведения этих работ, конечно, необходимо привлекать современные научные разработки и высокопроизводительные компьютеры. Одной из задач, решение которой позволит оптимизировать разработку таких сетей, является задача о коммивояжере.

Анализ публикаций

Формулировку задачи коммивояжера как задачи целочисленного линейного программирования можно найти в [1, 2].

Для решения этой задачи разработаны точные, приближенные и эвристические алгоритмы. Точные алгоритмы представлены в рамках схемы ветвей и границ, а также метода динамического программирования [1, 3, 4]. Эти алгоритмы носят экспоненциальный характер, т.е. время, затрачиваемое ими на решение различных задач при увеличении количества городов, растет экспоненциально. При этом алгоритм, реализующий метод динамического программирования, как по количеству выполняемых операций, так и по требуемой для хранения данных памяти ЭВМ, уступает методу ветвей и границ [4].

Впервые метод ветвей и границ как метод оптимизации был продемонстрирован группой американских математиков, возглавляе-

мой Литлом, именно на задаче о коммивояжере [5]. Разработанный ими алгоритм позволял решать задачи с количеством городов до 30 и несимметричной матрицей C (матрица расстояний между городами) не более чем за 8 минут работы центрального процессора. Задачи с симметричной матрицей решались существенно дольше. Было также установлено, что время счета растет экспоненциально с увеличением количества городов. Причем добавление 10 городов увеличивает время примерно на порядок.

С нашей точки зрения, статистические исследования алгоритма Литла недостаточны для того, чтобы сделать правдоподобные заключения о его рабочих характеристиках. Кроме того, он допускает ряд модификаций, которые могут улучшить эти характеристики.

Цель и постановка задачи

Задача о коммивояжере формулируется следующим образом. Имеется n городов $1, 2, \dots, i, \dots, n$ и расстояний между ними, представленных квадратной матрицей $C = \|c_{ij}\|_{n \times n}$, $i = 1, 2, \dots, n$, $j = 1, 2, \dots, n$. Коммивояжер выезжает из некоторого базового города, в качестве которого путем перенумерации всегда можно принять город n , объезжает все остальные города, побывав в каждом по одному разу, и возвращается в исходный город. Требуется найти такой порядок объезда городов, который бы минимизировал общую стоимость объезда.

Различают несколько разновидностей задачи, обусловленных свойствами элементов матрицы C . Если расстояние из города i в город j и наоборот – из j в i – одинаково $\forall i, j$, задачу называют задачей с симметричной матрицей расстояний. В противном случае, т.е. когда $c_{ij} \neq c_{ji}$, $i = 1, 2, \dots, n$, $j = 1, 2, \dots, n$, $i \neq j$, задачу характеризует несимметричная матрица C . Когда для любой тройки городов i, j, k справедливо условие $c_{ij} + c_{jk} \geq c_{ik}$, говорят, что задача удовлетворяет условию треугольника. В противном случае это условие ей не свойственно. Самая общая постановка задачи та, для которой матрица расстояний несимметричная и для ее элементов не выполняется условие треугольника.

Известно несколько математических формулировок задачи: сетевая, комбинаторная, целочисленного линейного программирования. Наиболее плодотворными из них для построения и обоснования решающих алгоритмов являются первые две.

Если каждому городу $1, 2, \dots, n$ поставить в соответствие вершины v_1, v_2, \dots, v_n и каждую пару вершин соединить взвешенными ребрами, отражающими стоимости проезда из города в город в ту и другую сторону, получим мультиграф, в котором требуется найти покрывающий (Гамильтонов) цикл, наименьшей длины. Для задачи с симметричной матрицей такой цикл следует искать в обычном графе.

Комбинаторная формулировка определяется тем, что любой Гамильтонов цикл – это замкнутый путь в мультиграфе, включающий все вершины $1, 2, \dots, n$. Тогда, поскольку одна (базовая) вершина фиксирована, любой путь – это перестановка π на множестве чисел $1, 2, \dots, n-1$, замкнутая на первоначальную вершину. Следовательно, среди множества перестановок $(n-1)!$ требуется найти такую циклическую перестановку $\pi^* = (i_1, i_2, \dots, i_{n-1})$, для которой суммарная стоимость $c_{i_1 i_2} + c_{i_2 i_3} + \dots + c_{i_{n-2} i_{n-1}} + c_{i_{n-1} i_1}$ будет минимальной.

Цель настоящей работы – обосновать модификации алгоритма Литла и на основании эксперимента определить в целом границы практического применения рассматриваемого алгоритма. Для достижения указанной цели решались следующие задачи: а) конструирование модифицированного алгоритма Литла; б) программная реализация разработанного алгоритма; в) проведение экспериментальных исследований алгоритма.

Алгоритм и его модификации

По существу метод ветвей и границ в оптимизационной задаче комбинаторного типа сокращает полный перебор решений, позволяя получить для задач разумной размерности точное решение в приемлемые сроки. Степень сокращения перебора, вообще говоря, не определена и сильно зависит от многих факторов. Однако метод содержит инструменты, позволяющие во многих случаях оказывать воздействие на эту интенсивность

и, таким образом, добиваться требуемого решения.

Идея метода состоит в том, что исходное множество решений M по некоторому правилу разбивается на конечное число непересекающихся подмножеств M_i , таких, что $\cup M_i = M$, $M_i \cap M_j = \emptyset$, $i \neq j$. Далее, по определенному признаку, выбирается одно из подмножеств M_i и, в свою очередь, по принятому правилу разбивается на конечное число непересекающихся подмножеств M_k , таких, что $\cup M_k = M_i$; $M_k \cap M_l = \emptyset$, $k \neq l$. Процесс разбиения подмножеств продолжается до тех пор, пока не получают подмножество, содержащее один элемент π . Для него вычисляется значение оптимизируемой функции $F(\pi)$, часто называемое рекордом.

Процедуру разбиения исходного множества на подмножества удобно представлять как построение корневого ориентированного z -ярусного дерева, в котором вершины – это множества, а дуги ориентированы непосредственно от порождающих множеств к порождаемым. В такой интерпретации очередное разбиение множества представляется как разветвление соответствующей ветви дерева, в связи с чем процедуру разбиения называют ветвлением.

Процесс ветвления сопровождается вычислением для исходного множества M и каждого из порождаемых подмножеств нижних (верхних) границ оптимизируемой функции на этих подмножествах – оценок точных значений функции. Именно эти границы являются инструментом метода, сокращающим полный перебор. Однако это достигается только в том случае, когда нижние границы функции от предка к потомку не убывают, а верхние – не возрастают. При этом в конечной вершине дерева (листе) границы трансформируются в значение оптимизирующей функции, соответствующее маршруту ветвления. Действительно, если в случае минимизации функции $F(\pi)$ найдено некоторое решение задачи $F_r(\pi)$ и для очередного подмножества M_k вычислена нижняя граница $f(M_k) \geq F_r(\pi)$, то, в силу $f(M_{k+1}) \geq f(M_k)$, подмножество M_k не будет продуцировать решений $F(\pi)$, меньших известного решения $F_r(\pi)$. Вследствие этого все решения, порождаемые последователь-

ным разбиением M_k , могут быть отброшены. В дереве поиска это равнозначно отсечению ветвей, выходящих из вершины, отображаемой множеством M_k . Аналогичным образом дело обстоит и при максимизации функции, только отсечение осуществляют тогда, когда $f(M_k) \leq F_r(\pi)$.

Степень отсечения ветвей, а равно и сокращение поиска, существенно зависит от близости оценок к точным значениям функции на подмножествах. Однако, как показывает опыт вычислений, получение качественных оценок всегда сопряжено с ростом его объема. В связи с этим общее сокращение времени решения задачи может быть не достигнуто. Поэтому обычно ищут надлежащий компромисс между сложностью решения оценочных задач и интенсивностью отсечения. Если он достигается, то, как правило, экспериментальным путем.

Во многих случаях эффект отсечения становится весьма ощутимым, если в задаче минимизации $F(\pi)$ используется верхняя граница функции $F(\pi)$, а в задаче максимизации – нижняя граница функции $F(\pi)$. Такие границы с небольшими затратами времени могут быть вычислены до применения метода, а в процессе его реализации уже на начальной стадии разбиения множеств и вычисления оценок применяться для отсечения ветвей в дереве поиска.

Практика решения различных задач методом ветвей и границ показывает, что и способ ветвления, и его стратегия (правило выбора очередного множества для разбиения), и вычисление границ $F(\pi)$ на подмножествах специфичны для каждой задачи. Вместе с тем на основании этой же практики выработаны рекомендации, которых, по возможности, следует придерживаться при разработке алгоритма. Схема ветвления должна порождать меньше вершин на верхних ярусах дерева решений. Стратегия – способствовать получению меньшего (большого) рекорда за первый проход дерева от корня к листу. Решение оценочных задач – не слишком затратным способом вычислять нижние (верхние) границы, близкие к точным границам оптимизируемой функции на подмножествах.

В алгоритме Литла используется схема ветвления, в результате применения которой за

один проход от корня к листу строится бинарное дерево решений, содержащее $n-2$ яруса. Корень дерева отображает все множество Гамильтоновых циклов задачи. Ветви, выходящие из корня и остальных вершин, определяются выбором одного конкретного ребра, указывающего направление переезда из города в город. Таким образом, каждое множество разделяется на два подмножества: содержащее все Гамильтоновы циклы с данным переездом из города в город и циклы, не содержащие таких переездов.

Вычисление нижних границ основано на том факте, что в любой Гамильтонов цикл входят только по одному разу элементы (ребра с соответствующей стоимостью) из каждой строки и столбца матрицы C . Например, c_{12} , c_{23} , c_{34} , c_{41} для матрицы 4×4 . Поэтому если из некоторой строки или столбца матрицы C вычесть константу h , что называется приведением матрицы, стоимость всех объездов (в том числе и оптимальных) уменьшится ровно на эту величину. Иными словами, в результате этой операции изменение состава ребер оптимального маршрута не произойдет, а стоимость его будет представляться величиной $F(\pi^*) = h + F'(\pi^*)$, где $F'(\pi^*)$ – стоимость оптимального маршрута в приведенной матрице C' .

Если в качестве константы h в некоторой строке матрицы C выбрать минимальный элемент, то эту константу можно считать нижней границей $g = h$ функции $F(\pi)$. Однако такая граница слишком груба. Более точная граница в корне дерева получается последовательным вычитанием минимальных элементов из строк и столбцов матрицы C с дальнейшим суммированием этих отрицательных элементов, т.е. $g = \sum_{j=1}^n \sum_{i=1}^n \min c_{ij}$.

Такая процедура называется полным приведением матрицы C .

Вычисление нижних границ функции $F(\pi)$ на подмножествах циклов, содержащих выbranное ребро (i, j) и не содержащих его, одинаково в том отношении, что к текущей нижней границе g_S функции $F(\pi)$ на подмножестве, которое разделяется, добавляется некоторое число h , и различно в том смысле, что эти числа разные и получаются различными путями. В первом случае в приведен-

ной матрице, соответствующей разбиваемому множеству, вычеркивается i -я строка и j -й столбец, ребро (i, j) добавляется к списку выбранных ребер, устанавливается замкнутый цикл с началом p и концом q и блокируется его образование в матрице C введением элемента $c_{qp} = \infty$. Затем осуществляется полное приведение видоизмененной матрицы, находится добавка h и вычисляется новая нижняя граница $g = g_s + h$ на подмножествах циклов, содержащих ребро (i, j) . Вторая добавка h к g_s получается в результате выполнения операции ветвления, которая определяет ребро (i, j) . Для этого в каждой строке матрицы C , соответствующей разделяемому множеству, находится минимальный элемент, не равный нулю, фиксируется соответствующий столбец, находится в нем минимальный элемент, также не равный нулю, после чего суммируются эти элементы. Далее среди полученных сумм находится наибольшая h , которая определяет ребро (i, j) и прибавляется к текущей границе. В результате получаем $g = g_s + h$.

Стратегия ветвления, т.е. выбор очередного множества для разбиения, реализуется так: из всех висячих вершин дерева поиска, отображающих подмножества циклов, выбирается та вершина, для которой нижняя граница минимальна.

Исходя из изложенного, алгоритм Литла может быть представлен следующим образом.

Шаг 1. Положить рекорд $R = \infty$.

Шаг 2. Выполнить первое приведение матрицы C и найти $g_s = \sum_{j=1}^n \sum_{i=1}^n \min c_{ij}$.

Шаг 3. Выполнить ветвление, т.е. определить максимальную добавку h и ребро (i, j) .

Шаг 4. Вычислить нижнюю границу $g = g_s + h$ для множества циклов, не содержащих ребро (i, j) ; запомнить эту границу и ребро (i, j) .

Шаг 5. В текущей матрице C вычеркнуть строку i , столбец j , установить подцикл (p, q) и заблокировать его образование, положив $c_{qp} = \infty$; привести матрицу C и найти добавку h .

Шаг 6. Вычислить нижнюю границу $W = g_s + h$ для множества циклов, содержащих ребро (i, j) .

Шаг 7. Если $Z > n - 2$, перейти к шагу 10.

Шаг 8. Если $W > R$, перейти к шагу 10.

Шаг 9. Положить $R = W$, добавить два ребра к списку ребер, запомнить этот список как лучший текущий цикл.

Шаг 10. Выбрать висячую вершину дерева поиска с наименьшей нижней границей g .

Шаг 11. Если $g \geq R$, остановиться.

Шаг 12. Получить матрицу C для ветвления в выбранной висячей вершине и вернуться к шагу 3.

В описанный алгоритм внесены изменения: добавлено использование верхней границы, применяются другая стратегия ветвления и другой способ получения матрицы C для ветвления в выбранной висячей вершине. В качестве верхней границы R используется стоимость объезда, полученная модифицированным эвристическим алгоритмом «ехать в ближайший город» [4].

Что касается способа получения матрицы C для ветвления в выбранной висячей вершине дерева поиска, то в целях сокращения объемов вычислений, особенно на нижних ярусах дерева, он заменен запоминанием матриц C для вершин ветвления. Для ветвления выбирается вершина, определяющая все циклы, включающие выбранное ребро (i, j) .

Результаты эксперимента

Экспериментальные исследования проводились для того, чтобы определить временные характеристики алгоритма и, таким образом, установить предельные границы его практического применения. С этой целью были составлены три программы полного перебора (генерации всех перестановок чисел $1, 2, \dots, n$ с вычислением стоимости объезда) модифицированного эвристического алгоритма «ехать в ближайший город» и собственно алгоритма метода ветвей и границ. Программа прямого перебора использовалась для тестирования программы метода ветвей и границ.

Непосредственные задачи эксперимента состояли в следующем:

1) для ряда серий задач размером $n = 10, 12, 15, 17, 20, 22, 25, 27, 30$ получить наименьшие, средние и наибольшие времена счета, а также другие статистические харак-

теристики с использованием верхней границы R и без нее;

2) по статистическим данным построить эмпирические зависимости времен решения задачи как функции ее размера;

3) оценить погрешности эвристического алгоритма.

Каждая серия состояла из 100 задач со случайными временами переезда, равномерно распределенными на интервале $(0, 100)$. Генерировались как симметричные, так и несимметричные матрицы C . Погрешности эвристического алгоритма определялись по выражению $\Delta = (S_l - S_v) / S_v$, где S_l – минимальная, средняя или максимальная стоимости объезда, найденные эвристическим алгоритмом, а S_v – те же стоимости, найденные методом ветвей и границ. Статистические данные обрабатывались средствами пакета *STATISTICA 6* фирмы *StatSoft*.

Данные расчетов свидетельствуют, прежде всего, о том, что использование верхней границы R как для задач с симметричной матрицей расстояний C , так и с несимметричной, не улучшает временные характеристики алгоритма Литла. Наоборот, оно в незначительной степени их ухудшает. Объяснить этот результат можно влиянием двух факторов. Во-первых, модифицированный алгоритм «ехать в ближайший город», используемый для вычисления верхней границы R , дает ее значения, весьма далекие от оптимальных решений, и тем самым, не обеспечивает эффективного отсеечения ветвей в дереве поиска. Во-вторых, сложность алгоритма $O(n^3)$, и это требует определенных затрат времени на его реализацию, которые не покрываются ожидаемым сокращением времени за счет эффективности отсеечения.

Средние времена счета t_S приближены экспоненциальной зависимостью $t = \exp(a + b \cdot n)$, в результате чего получены эмпирические формулы зависимости времени счета t от размера задачи n :

а) для задач с симметричной матрицей C : $t = \exp(-10,46 + 0,338 \cdot n)$, без использования R ; $t = \exp(-10,47 + 0,339 \cdot n)$, с использованием R ;

б) для задач с несимметричной матрицей C : $t = \exp(-10,39 + 0,251 \cdot n)$, без использования R ; $t = \exp(-8,67 + 0,183 \cdot n)$, с использованием R .

Используя эти зависимости, методом экстраполяции получаем, что задачу с 50 городами и симметричной матрицей расстояний C рассматриваемым алгоритмом в среднем можно решать за 636–649 секунд, задачу с несимметричной матрицей C – за 8,7 – 1,7 секунд.

Минимальные и максимальные времена счета t_m, t_{mx} существенно отличаются от среднего времени t_S . Например, для задач с 30 городами и симметричной матрицей расстояний C максимальное время больше среднего примерно в 60 раз, для задач с несимметричной матрицей C – примерно в 50 раз. Это свидетельствует о достаточно высоком размахе колебаний времени решения задач рассматриваемым алгоритмом. В то же время средние характеристики алгоритма статистически достаточно устойчивы, что подтверждается низким стандартным отклонением S_{KO} и узким 95-процентным доверительным интервалом среднего значения времени счета.

Говоря о качестве модифицированного алгоритма «ехать в ближайший город», следует сказать, что этот алгоритм мало пригоден для практических применений. Средние погрешности алгоритма, хотя и растут линейно (эмпирические зависимости приведены ниже), достаточно высоки, и, что характерно, для задач с несимметричной матрицей расстояний C они выше, чем для симметричной матрицы C : $\Delta_S = -1,85 + 1,17 \cdot n$, матрица C – симметричная; $\Delta_S = 6,49 + 1,46 \cdot n$, матрица C – несимметричная.

Методом экстраполяции получаем, что для задач с 50 городами эвристический алгоритм по отношению к оптимальному решению будет давать для симметричной матрицы C и несимметричной матрицы C средние погрешности соответственно 56,6 % и 80,3 %.

Выводы

Таким образом, в целом экспериментальные исследования алгоритма Литла показали, что его целесообразно использовать в массовом

порядке для решения задач размерности до 50 городов преимущественно с несимметричной матрицей расстояний S .

Научная новизна разработки состоит в конструировании и экспериментальном исследовании алгоритма Литла и его модификации, касающейся введения верхней границы минимизируемой функции, предназначенной для повышения быстродействия алгоритма.

Практическая значимость исследований состоит в том, что на основании статистической обработки полученных экспериментальных данных даются правдоподобные заключения о рабочих характеристиках алгоритма, что позволило достаточно точно определить границы практического применения алгоритма при составлении маршрутной сети.

Литература

1. Корбут А.А. Дискретное программирование / А.А. Корбут, Ю.Ю. Финкельштейн. – М.: Наука, 1969. – 368 с.
2. Пападимитриу Х. Комбинаторная оптимизация. Алгоритмы и сложность / Х. Пападимитриу, К. Стайглиц. – М.: Мир, 1985. – 510 с.
3. Гудман С. Введение в анализ и разработку алгоритмов / С. Гудман, С. Хидетниши. – М.: Мир, 1981. – 366 с.
4. Конвей Р.В. Теория расписаний / Р.В. Конвей, В.П. Максвелл, Л.В. Миллер. – М.: Наука, 1975. – 360 с.
5. Литл Дж. Алгоритм для решения задачи о коммивояжере / Дж. Литл, К. Мурти, Д. Суини, К. Кэрел. // Экономика и математические методы. – 1965. – Вып. 1, № 1. – С. 94 – 107.

Рецензент: П.Ф. Горбачев, профессор, д.т.н., ХНАДУ.

Статья поступила в редакцию 21 марта 2012 г.
