

УДК 519.16

DOI: 10.25559/SITITO.15.201901.99-106

Исследование влияния асимметрии на сложность решения задачи коммивояжера методом ветвей и границ

Н. В. Мамонов^{1,2}

¹ Московский государственный университет имени М.В. Ломоносова, г. Москва, Россия

² ООО «Гудфокаст», г. Москва, Россия

mnv95@mail.ru

Аннотация

Одной из самых известных задач комбинаторной оптимизации является задача коммивояжера. Эта задача является важной и вместе с тем трудноразрешимой, что связано с экспоненциальной сложностью точных методов решения. Она возникает в обширном классе приложений: в построении оптимальных схем движения, распознавании траекторий и образов. Однако, целый ряд научных исследований, в частности, вопросы генетики и биоинформатики, требуют получения точного решения. На основе классической реализации метода ветвей и границ в работе проводится исследование влияния асимметрии на сложность решения. Поскольку проводимое исследование требует анализа большого числа экспериментально полученных данных, возникает необходимость повышения скорости проведения расчетов. Выбранный для работы метод ветвей и границ допускает получение большей скорости перебора решений за счет хранения в памяти состояний активных листовых вершин дерева решений. На эффективность реализации алгоритма влияет также выбор структур данных для быстрого нахождения следующей рассматриваемой вершины. На основании полученных результатов и проведенного анализа установлена зависимость сложности решения задачи от асимметричности матрицы. Некоторые из вариантов внесения асимметрии приводят к сокращению сложности решения. Изучение влияния асимметрии на сложность задачи коммивояжера на примере классического алгоритма решения – метода ветвей и границ – представляет интерес для дальнейшего предиктивного анализа сложности индивидуальных задач.

Ключевые слова: асимметричная задача коммивояжера, метод ветвей и границ, структуры данных, дерево решений.

Для цитирования: Мамонов Н. В. Исследование влияния асимметрии на сложность решения задачи коммивояжера методом ветвей и границ // Современные информационные технологии и ИТ-образование. 2019. Т. 15, № 1. С. 99-106. DOI: 10.25559/SITITO.15.201901.99-106

© Мамонов Н.В., 2019



Контент доступен под лицензией Creative Commons Attribution 4.0 License.
The content is available under Creative Commons Attribution 4.0 License.



The Study of the Influence of Asymmetry on the Complexity of Solving the Traveling Salesman Problem by the Branch and Bound Method

N. V. Mamonov^{1,2}

¹ Lomonosov Moscow State University, Moscow, Russia

² GoodsForecast, Moscow, Russia

mnv95@mail.ru

Abstract

The traveling salesman problem is the one of the most famous combinatorial optimization problems. This problem is important and at the same time difficult to solve, due to the exponential complexity of the exact methods of solution. It occurs in an extensive class of applications: in the construction of optimal motion schemes, recognition of tracks and images. However, a number of scientific studies, in particular, genetics and bioinformatics, require an accurate solution. This study is concerned with the problem of asymmetry influence on the complexity of the solution, which obtained with the classical implementation of branches and bounds method. Since the study requires the analysis of numerous experimental data, there is a need to increase the speed of calculations. The selected approach using branch and bounds method allows gaining an extra speed of search in decision tree by maintaining in-memory state of the active list nodes of the tree. The efficiency of the implementation of the algorithm is also affected by the choice of data structures for the rapid finding of the next vertex under consideration. Based on the results and the analysis, the dependence of the complexity of the problem solution on the asymmetry of the matrix is established. Some asymmetry options reduce the complexity of the solution. The study of the influence of asymmetry on the traveling salesman problem's complexity on the example of the classical solution algorithm – the method of branches and bounds – is of interest to further predictive analysis of the complexity of individual problems.

Keywords: asymmetric traveling salesman problem, branch and bound method, data structures, decision tree.

For citation: Mamonov N.V. The Study of the Influence of Asymmetry on the Complexity of Solving the Traveling Salesman Problem by the Branch and Bound Method. *Sovremennye informacionnye tehnologii i IT-obrazovanie* = Modern Information Technologies and IT-Education. 2019; 15(1):99-106. DOI: 10.25559/SITITO.15.201901.99-106



Введение

Как известно, задача коммивояжера состоит в том, что торговец должен посетить несколько городов, продавая в них товары. Между городами проложены пути, имеющие некоторую стоимость. В общем случае, стоимость проезда между двумя городами в каждом из направлений может отличаться. Маршрут, проходящий через все указанные города ровно по одному разу, называется туром. Тур минимальной стоимости, следуя которому коммивояжер посещает города, является решением задачи. Возможный простейший алгоритм решения – полный перебор всех возможных вариантов. Однако, учитывая факториальное количество туров и реальные размерности задачи, такой подход оказывается неприменимым на практике.

Задача коммивояжера относится к задачам большой вычислительной сложности, теория алгоритмической сложности относит ее к классу NP-трудных задач. Несмотря на многолетнюю историю исследования данной проблемы, точные методы решения имеют экспоненциальную сложность. А алгоритмы, с помощью которых можно получить оптимальный тур, принадлежат к классу NPRH [1], то есть к классу алгоритмов с высокой параметрической зависимостью трудоемкости при фиксированной размерности.

Математическая постановка задачи строится в терминах теории графов: города соответствуют вершинам графа, а ребра между вершинами – это пути сообщения между городами. Дуги графа дополнительно нагружаются информацией о стоимости пути между соответствующими вершинами. В случае, если стоимость проезда не зависит от направления движения, то задача называется симметричной, иначе – асимметричной. В общем случае, асимметричная задача коммивояжера моделируется ориентированным графом. Таким образом, рассматривается полный ориентированный связный граф на n вершинах без собственных петель, задаваемый взвешенной матрицей стоимостей $C = (c_{ij})$. В матрице стоимостей диа-

гональные элементы, отражающие стоимость дуги в ту же вершину – то есть петли, будут задаваться как $c_{ii} = \infty$. Тогда решение задачи коммивояжера формулируется как нахождение гамильтонова цикла минимального веса в построенном графе.

Задача коммивояжера в чистом виде применяется при планировании, логистике и изготовлении микросхем. Работы последних лет продемонстрировали применение задачи коммивояжера к решению подзадач в различных областях, например, при определении аминокислотной и нуклеотидной последовательности биополимеров, в результате чего получают формальное описание первичной структуры линейной макромолекулы в виде последовательности мономеров в текстовом виде [2]. Еще одной областью, в которой применяется задача коммивояжера, является астрономия: в процессе исследования космоса необходимо минимизировать время, необходимое на перемещение телескопа между направлениями на различные небесные тела.

Для решения поставленной задачи нахождения оптимального тура для коммивояжера существуют различные методы. Обычно, все множество методов решения задачи коммивояжера разделяют на точные, субоптимальные, которые дают приближенное решение задачи, и методы, использующие особенности конкретной задачи. К эвристическим методам в

большинстве случаев относятся такие, которые находят не самый эффективный маршрут, а некоторое приближение. Методы, основанные на эвристиках и аппроксимационных алгоритмах, позволяют получить приближенное решение за приемлемое время. Одним из таких методов является метод ближайшего соседа. Расширением этого метода является применение динамического программирования для решения задачи на фрагментах с дальнейшим объединением в один тур. Современные методы показывают ошибку около 2-3% на задачах очень большой размерности (миллионы городов). В основном, удовлетворительное приближение показывают различные эвристические эволюционные алгоритмы, в частности генетический и муравьиный.

Точность решения, получаемого с помощью этих алгоритмов, оказывается достаточной для многих бизнес задач, сводящихся к классической задаче коммивояжера. Однако, целый ряд научных исследований, в частности вопросы генетики и биоинформатики, требуют получения точного решения.

Точные алгоритмы позволяют находить оптимальный путь, при этом, с помощью различных методов, уменьшая количество рассматриваемых перебором туров. К таким алгоритмам относится алгоритм полного перебора – наиболее простой подход к решению задачи. Производится перебор всех возможных перестановок и последующий выбор оптимального пути. Сложность алгоритма в таком случае имеет оценку $O(n!)$. Ввиду надэкспоненциальной сложности задачи, очевидно, что алгоритм полного перебора не может быть применен в реальных задачах, на достаточно больших размерностях входа.

Точное решение задачи коммивояжера дает также классический метод ветвей и границ, и его различные модификации [3]. Метод получается развитием метода полного перебора, добавлением проверок, по результатам которых может быть приостановлено рассмотрение ветви решений и сделан переход на более перспективную ветвь, которая, вероятно, содержит оптимальное решение. Именно на основе классической реализации метода ветвей и границ строится дальнейшее исследование влияния асимметрии на сложность решения.

Поскольку проводимое исследование требует анализа большого числа экспериментально полученных данных, возникает необходимость повышения скорости проведения расчетов. Выбранный для работы метод ветвей и границ допускает получение большей скорости перебора решений за счет хранения в памяти состояний активных листовых вершин дерева решений. Таким образом, расходуя дополнительный объем памяти, исключаются повторные вычисления промежуточных результатов. Также на эффективность реализации алгоритма влияет выбор структур данных для быстрого выбора следующей рассматриваемой вершины.

Метод ветвей и границ

Метод ветвей и границ – алгоритм оптимизации перебора, цель которого состоит в нахождении состояния, при котором функция стоимости достигает экстремума. Такой подход использует древовидное представление решений. Для решения задачи коммивояжера рассматривается классический алгоритм, авторами которого являются Литл, Мерти, Суини и Карел [4].

Метод состоит из двух важных действий: операции ветвления и операции отсечения. Операция ветвления предполагает раз-



биение всего множества решений на два подмножества, что в дальнейшем приводит к сокращению перебора. Для каждого из подмножеств вычисляется оценка, которая позволяет отсекал такие подмножества решений, которые точно не содержат минимального тура. Такое действие называется построением границ, и в задаче поиска минимума оценка производится для нижней границы множества. Таким образом, метод приводит к исследованию древовидной модели пространства решений.

В задаче коммивояжера исходным множеством решений является множество всех туров, на котором минимизируется целевой функционал, задающий стоимость тура. Для построения метода предлагаются алгоритмы для двух основных процедур – ветвления и построения границ.

Операция ветвления. Рассматривается асимметричная задача с n вершинами и матрицей стоимостей C . Работа алгоритма начинается с корня поискового дерева решений, который будет соответствовать полному множеству всех возможных туров, мощность которого составляет $(n-1)!$. Ветви, выходя-

щие из корня, определяются выбором одного ребра, например, ребра (k, l) . При выборе ребра разбиения (k, l) целью ста-

вится получение двух подмножеств исходного множества решений, таких, что одно с высокой вероятностью содержит оптимальный тур и второе, в котором искомый тур отсутствует. Для выбора ребра, которое, вполне вероятно, входит в оптимальный тур предлагается специальный алгоритм. Множество R разделяется на два множества $\{k, l\}$ и $\overline{\{k, l\}}$. Во

множество $\{k, l\}$ входят все туры из R , содержащие ребро (k, l) , т.е. проходящие через него, а во множество $\overline{\{k, l\}}$ –

туры, не содержащие это ребро. Из вышеизложенных рассуждений ясно, что если процесс ветвления будет организован так, что на каждом шаге выбирается «правильное» ребро, то весь процесс будет завершён за n шагов.

В процессе построения дерева с каждой его вершиной связывается нижняя оценка стоимости любого тура из множества, соответствующего рассматриваемой вершине. Чем точнее будет произведена нижняя оценка множества, тем эффективнее будет работа алгоритма: предположим, что уже получен определенный полный тур T со стоимостью $s(T)$. Если нижняя

граница, связанная с множеством туров, представленных некоторой вершиной поискового дерева, больше, чем $s(T)$, то

до конца процесса поиска не нужно рассматривать эту и все следующие за ней вершины. В реализации это приводит к усечению поискового дерева решений путем отбрасывания всех листьев дерева, имеющих стоимость большую, чем $s(T)$.

Алгоритм решения задачи коммивояжера методом ветвей и границ

Пусть R – полное множество туров на n городах. X – рассматриваемая на текущей итерации вершина дерева поиска. Через \bar{Y} и Y обозначаются подмножества, следующие непосредственно за вершиной X , первое из которых не содержит

ребро ветвления (k, l) , а второе, содержащее это ребро. Нижние оценки для множеств \bar{Y} и Y обозначаются через $w(\bar{Y})$ и $w(Y)$. Оптимальный тур, известный алгоритму на данный

момент, обозначается через S . Причем на шаге инициализации текущий оптимальный тур устанавливается равным бесконечности.

Алгоритм решения задачи коммивояжера методом ветвей и границ.

Вход: C – матрица стоимостей,
 n – размерность.

Выход: Оптимальное решение со стоимостью S .

Шаг 1. Инициализация переменных.

Шаг 2. Приведение матрицы стоимостей C . Установка корня дерева решений $X = R$.

while ($w(X) < S$)

Шаг 3. Выбор ребра для следующего ветвления.

Шаг 4. Создание вершины \bar{Y} . Вычисление $w(\bar{Y})$.

Шаг 5. Создание вершины Y . Вычисление $w(Y)$.

if ($\text{size}(C_Y) == 2$)

Шаг 6. Проведение исчерпывающей оценки для Y .

if ($w(Y) < S$)

$S = w(Y)$; Запоминаем новый оптимальный тур.

endif

endif

Шаг 7. Выбор следующей вершины, и установка X .

Шаг 8. Пересчет новой матрицы C , соответствующей выбранной вершине, либо чтение матрицы из памяти.

end

Оптимальный тур хранится в памяти, стоимость тура S . ■

Описание программной реализации

В процессе решения задачи коммивояжера методом ветвей и границ происходит порождение дерева решений, которое необходимо хранить в памяти компьютера. От способов организации поискового дерева зависит общая ресурсная эффективность реализации. Выбор способа внутреннего представления дерева определяется особенностями организации работы с ним, а также частотой использования базовых операций вставки, удаления и поиска. Обычно в литературе рассматриваются подходы, основанные на самобалансирующихся деревьях или бинарной куче.

Детальный анализ алгоритма и проведенные эксперименты демонстрируют экспоненциальный рост количества порожденных вершин дерева решений с увеличением размерности задачи. Исходя из этого и учитывая шаг 7 алгоритма, следует, что к структуре хранения вершин дерева решений предъявляется требование быстрого получения подходящей листовой вершины, то есть такой вершины, у которой оценка туров минимальная. Также необходимо, чтобы выбранная структура данных поддерживала эффективную операцию вставки новых элементов. По итогам исследования в качестве структуры данных, удовлетворяющей описанным требованиям, был выбран упорядоченный список.



Добавление новой вершины, порожденной на текущем шаге, в упорядоченный список происходит только в двух случаях:

- Если на предыдущих шагах не получено ни одного тура. В этом случае не известна стоимость ни одного полного маршрута и необходимо добавлять все порожденные вершины.
- Если некоторый оптимальный тур получен, и оценка текущей вершины меньше либо равна стоимости найденного тура. В этом случае именно рассматриваемый на текущем шаге маршрут может обладать меньшей стоимостью, чем найденный ранее тур.

Одной из ключевых операций в методе ветвей и границ является операция усечения матрицы, которая заключается в исключении строки и столбца соответствующим текущему ребру ветвления. На этапах сохранения и восстановления усеченной матрицы возникает проблема ее компактного хранения. В качестве альтернатив рассматривались следующие варианты:

- сохранение исходной матрицы, в которой активные строки и столбы помечались особым образом,
- хранение массивов переиндексации отдельно от усеченной матрицы,
- расширение матрицы для хранения реальных индексов строк и столбцов.

Первый способ отличается как повышенным потреблением памяти на хранение исходных матриц, так и снижением производительности из-за непоследовательного доступа к памяти при пропуске неактивных строк и столбцов, соответствующих уже вычеркнутым на предыдущих этапах ребрам. Выбор между вторым и третьим вариантами во многом определяется используемым компилятором и средой выполнения. Исследования показали, что третий вариант оказывается предпочтительнее второго в силу повышения локальности памяти, что в свою очередь позволяет эффективнее утилизировать кэш-память, снижая количество обращений в оперативную память.

Помимо возможности выбора различных структур данных для организации хранения, на временную сложность алгоритма влияет способ работы с матрицами внутри метода ветвей и границ. Как показано выше, в процессе работы алгоритм производит переходы между активной веткой решения и веткой, потенциально принадлежащей пути с меньшим весом. В процессе перехода меняется текущее состояние матрицы стоимостей и соответствующие нижние и верхние оценки стоимости тура. Возможны два варианта организации смены рассматриваемой ветви в дереве решения: путем выполнения пересчета матрицы стоимостей или сохранения всех полученных в листовых вершинах конфигураций усеченных матриц. Здесь прослеживается классическая проблема оптимального соотношения времени и памяти. В первом варианте реализации метода ветвей и границ дополнительное процессорное время требуется на пересчет усеченных матриц стоимостей, во втором необходима дополнительная память на хранение этих матриц. В работах [5] на основе экспериментальных данных подтверждена гипотеза об эффективности подхода, использующего хранение матриц, и приведены численные оценки зависимости потребления памяти от длины входа. Авторы сформулировали рекомендации по практическому использованию реализации метода ветвей и границ с хранением усеченных матриц, на основе спрогнозированного времени вы-

полнения и затрат памяти по полученному тренду.

Реализация алгоритма для поставленной задачи исследования влияния асимметрии матрицы стоимостей производилась на языке C++. В качестве основного алгоритма использовался описанный ранее метод ветвей и границ с хранением листовых вершин с рассчитанными матрицами стоимостей.

В таком случае происходило ускорение работы алгоритма за счет большего расхода памяти на хранение результатов прошлых расчетов. С целью экономии памяти алгоритм не хранит все дерево целиком, в памяти остаются только листовые вершины, которые впоследствии могут стать активными.

Сложность задачи коммивояжера, решаемой с помощью метода ветвей и границ, оценивается как полное число порожденных вершин дерева решений, а не фактическое время выполнения алгоритма. Выбор такой оценки позволяет абстрагироваться от аппаратной и вычислительной среды, в которой проводятся эксперименты, а также от влияния нюансов реализации на конечную сложность задачи. На сложность влияют процедуры алгоритма метода ветвей и границ, а именно стратегия выбора ребра ветвления, поиска следующей вершины дерева решений и проведение оценки [6]. Все описанные процедуры остаются неизменными для проводимых экспериментов.

Алгоритм работает с вещественными матрицами стоимостей, что несколько повышает затраты на операции с числами, однако является необходимым в условиях поставленной задачи. Для хранения матрицы стоимостей разработана специальная структура данных, позволяющая эффективно производить основные операции метода ветвей и границ – приведение и редукция. Для хранения в матрице отсутствующих переходов, например, петель, то есть диагональных элементов матрицы стоимостей, либо ребер, исключенных из тура, используется максимально возможное значение для типа double. В ходе работы алгоритма возникает потребность в работе с дугами, например, их объединение для предотвращения появления циклов. С целью повышения эффективности работы метода, реализован специальный класс списка, проводящий необходимые операции над ребрами, составляющими текущий тур.

Для каждой размерности алгоритм генерирует несколько симметричных матриц стоимостей. Это необходимо для сглаживания результатов оценки сложности. Для каждой матрицы вычисляется сложность алгоритма как количество порожденных вершин дерева решений. Затем с заданным шагом генерируются дельты, вносимые в симметричную матрицу стоимостей. Для полученных асимметричных матриц получаются соответствующие сложности. Результаты измерений обрабатываются и выводятся в файл с дополнительной информацией, необходимой для анализа.

Варианты внесения асимметрии в матрицу

Для упрощения проведения экспериментов сначала генерируется множество симметричных матриц разных размерностей, а затем в каждую случайную матрицу вносятся различные виды асимметрии. Величина, на которую изменяется элемент матрицы, задается как процент от находящегося в ячейке значения.

Рассматриваются следующие способы внесения асимметрии:



- Все элементы над главной диагональю получают положительное приращение, а все элементы ниже главной диагонали получают отрицательное приращение ($2Eps+/-$)
- Все элементы над главной диагональю получают отрицательное приращение, а все элементы ниже главной диагонали получают положительное приращение ($2Eps-/+$)
- Только элементы над главной диагональю получают положительное приращение ($Eps+$)
- Все дуги, выходящие из некоторой случайно выбранной вершины, получают приращение; технически все элементы некоторой строки матрицы получают приращение ($EpsRow$)
- Все дуги, входящие в некоторую случайно выбранную вершину, получают приращение ($EpsColumn$)
- Одна случайно выбранная дуга получает приращение ($EpsElem$)

Результаты проведения экспериментов

Ниже приведены графики с результатами измерения сложности для задачи с размерностью 5 – 30 городов (рис. 1 - 6). Асимметричность вводилась указанными выше способами.



Рис. 1. Зависимость сложности от eps, 5 городов
Fig. 1. Dependence of eps complexity, 5 cities

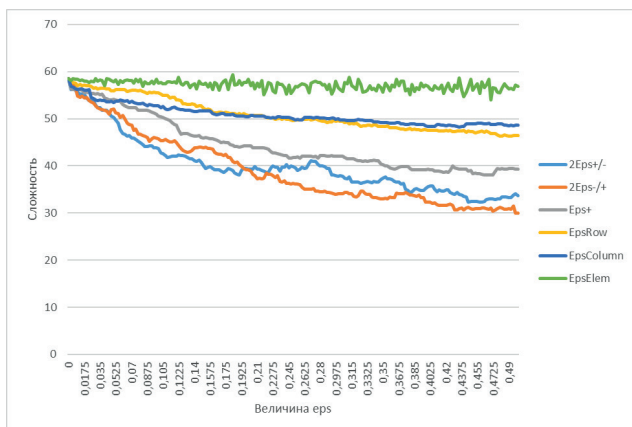


Рис. 2. Зависимость сложности от eps, 10 городов
Fig. 2. Dependence of eps complexity, 10 cities

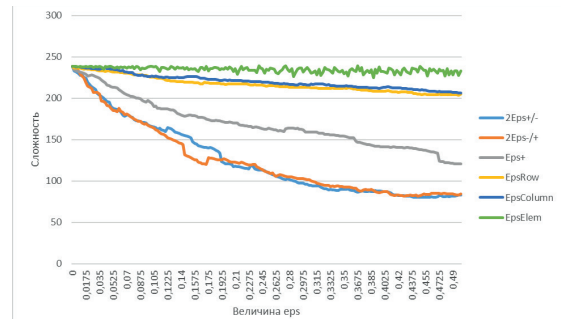


Рис. 3. Зависимость сложности от eps, 15 городов
Fig. 3. Dependence of eps complexity, 15 cities

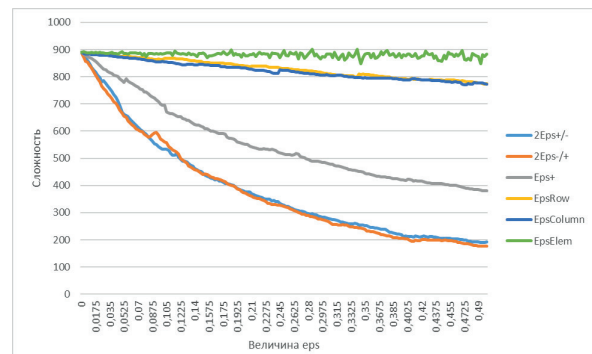


Рис. 4. Зависимость сложности от eps, 20 городов
Fig. 4. Dependence of eps complexity, 20 cities

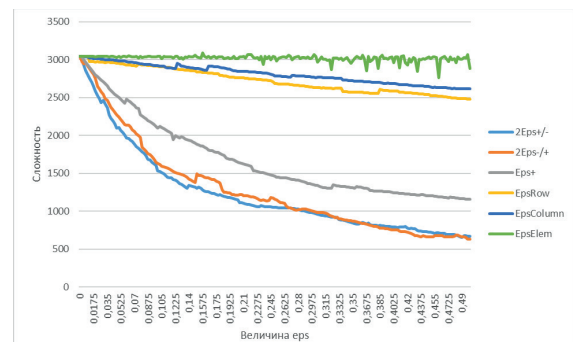


Рис. 5. Зависимость сложности от eps, 25 городов
Fig. 5. Dependence of eps complexity, 25 cities

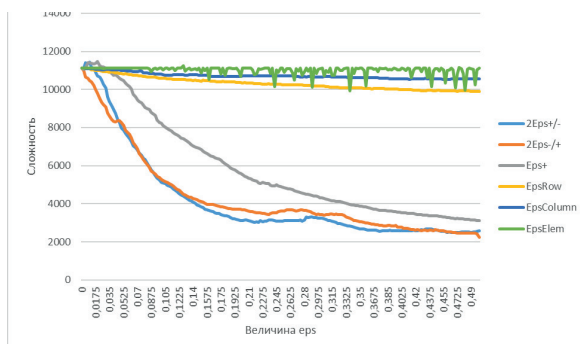


Рис. 6. Зависимость сложности от eps, 30 городов
Fig. 6. Dependence of eps complexity, 30 cities



Для каждой размерности производилось 100 генераций симметричных матриц, для каждой из которых вычислялось значение сложности на всем диапазоне величин внесения асимметрии.

Как показали результаты экспериментов, для всех размерностей характерен скачок уменьшения сложности при внесении асимметричности. С ростом размерности коэффициент уменьшения сложности растет. Это связано с тем, что в процессе построения дерева решений для симметричной задачи происходит большое количество смен вершин, так как нижние оценки располагаются близко друг к другу. При внесении асимметрии алгоритм лучше отсекает множества, не содержащие оптимального тура, и дерево становится меньше в ширину.

Заметен общий тренд: первые два варианта внесения асимметрии с ростом размерности задачи становятся неотличимыми. Изменение всех входящих/выходящих дуг одной вершины с ростом размерности значительно уменьшает влияние на результат. Изменение одного случайного элемента, как и предполагалось, не оказывает систематического влияния на сложность решения задачи.

Закключение

В работе рассмотрен классический метод ветвей и границ, использующий дополнительную память для хранения промежуточных матриц стоимостей, применительно к задаче коммивояжера. Реализован эффективный стенд для проведения экспериментов по изучению влияния асимметрии на сложность задачи коммивояжера. Полученные результаты и проведенный анализ для некоторых методов внесения асимметрии в матрицу стоимостей показывают, что асимметрия приводит к сокращению сложности решения задачи, при этом наибольшее сокращение (более, чем в три раза) наблюдается при асимметрии типа $2\text{Eps}+/-$ или $2\text{Eps}-/+$ для задач размерности 30.

Список использованных источников

- [1] Ульянов М.В. Ресурсно-эффективные компьютерные алгоритмы. Разработка и анализ. М.: Физматлит, 2008. 304 с.
- [2] Tang N.C., Chilkoti A. Combinatorial codon scrambling enables scalable gene synthesis and amplification of repetitive proteins // *Nature Materials*. 2016. Vol. 15, Issue 4. Pp. 419-424. DOI: 10.1038/nmat4521
- [3] Костевич Л.С. Математическое программирование. Информационные технологии оптимальных решений. Мн.: Новое знание, 2003. 424 с.
- [4] Гудман С., Хидетниemi С. Введение в разработку и анализ алгоритмов. М.: Мир, 1981. 368 с.
- [5] Ulyanov M.V., Fomichev M.I. Resource characteristics of ways to organize a decision tree in the branch-and-bound method for the traveling salesman problem // *Business Informatics*. 2015. No. 4(34). Pp. 38-46. DOI: 10.17323/1998-0663.2015.4.38.46
- [6] Goloveshkin V.A., Zhukova G.N., Ulyanov M.V., Fomichev M.I. Probabilistic Prediction of the Complexity of Traveling Salesman Problems Based on Approximating the Complexity Distribution from Experimental Data // *Automation and Remote Control*. 2018. Vol. 79. Issue 7. Pp. 1296-1310. DOI: 10.1134/S0005117918070093
- [7] Lobjois L., Lemaitre M. Branch and Bound Algorithm Selection by Performance Prediction // *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98)*. American Association for Artificial Intelligence, AAAI Press, Menlo Park, CA, 1998. URL: <https://aaai.org/Papers/AAAI/1998/AAAI98-050.pdf> (дата обращения: 13.01.2019).
- [8] Bektaş T., Gouveia L., Martínez-Sykora A., Salazar-González J. Balanced vehicle routing: Polyhedral analysis and branch-and-cut algorithm // *European Journal of Operational Research*. 2019. Vol. 273, Issue 2. Pp. 452-463. DOI: 10.1016/j.ejor.2018.08.034
- [9] Arigliano A., Ghiani G., Grieco A., Guerriero E., Plana I. Time-dependent asymmetric traveling salesman problem with time windows: Properties and an exact algorithm // *Discrete Applied Mathematics*. 2019. Vol. 261. Pp. 28-39. DOI: 10.1016/j.dam.2018.09.017
- [10] Hore S., Chatterjee A., Dewanji A. Improving variable neighborhood search to solve the traveling salesman problem // *Applied Soft Computing*. 2018. Vol. 68. Pp. 83-91. DOI: 10.1016/j.asoc.2018.03.048
- [11] Gouveia L., Leitner M., Ruthmair M. Extended formulations and branch-and-cut algorithms for the Black-and-White Traveling Salesman Problem // *European Journal of Operational Research*. 2017. Vol. 262, Issue 3. Pp. 908-928. DOI: 10.1016/j.ejor.2017.04.061
- [12] Huang Y., Zhong C. Detecting list-colored graph motifs in biological networks using branch-and-bound strategy // *Computers in Biology and Medicine*. 2019. Vol. 107. Pp. 1-9. DOI: 10.1016/j.combiomed.2019.01.025
- [13] Toriello A. Optimal Toll Design: A Lower Bound Framework for the Asymmetric Traveling Salesman Problem // *Mathematical Programming*. 2014. Vol. 144, Issue 1-2. Pp. 247-264. DOI: 10.1007/s10107-013-0631-6
- [14] Sergeev S. Nonlinear resolving functions for the travelling salesman problem // *Automation and Remote Control*. 2013. Vol. 74, Issue 6. Pp. 978-994. DOI: 10.1134/S0005117913060088
- [15] Charikar M., Goemans M.X., Karloff H. On the Integrality Ratio for the Asymmetric Traveling Salesman Problem // *Mathematics of Operations Research*. 2006. Vol. 31, Issue 2. P. 245-252. DOI: 10.1287/moor.1060.0191
- [16] Subramanyam A., Gounaris C. A branch-and-cut framework for the consistent traveling salesman problem // *European Journal of Operational Research*. 2016. Vol. 248, Issue 2. Pp. 384-395. DOI: 10.1016/j.ejor.2015.07.030
- [17] Головешкин В.А., Жукова Г.Н., Ульянов М.В., Фомичев М.И. Использование квантильных коэффициентов асимметрии и эксцесса для оценки сложности решения задачи коммивояжера // *International journal of open information technologies*. 2016. Т. 4, № 12. С. 7-12. URL: <https://elibrary.ru/item.asp?id=27543353> (дата обращения: 13.01.2019).
- [18] Liškiewicz M., Schuster M. A new upper bound for the traveling salesman problem in cubic graphs // *Journal of Discrete Algorithms*. 2014. Vol. 27. Pp. 1-20. DOI: 10.1016/j.jda.2014.02.001
- [19] Kara I., Derya T. Formulations for Minimizing Tour Duration of the Traveling Salesman Problem with Time Windows // *Procedia Economics and Finance*. 2015. Vol. 26. Pp. 1026-1034. DOI: 10.1016/S2212-5671(15)00926-0



- [20] LaRusic J., Punnen A.P. The asymmetric bottleneck traveling salesman problem: Algorithms, complexity and empirical analysis // *Computers & Operations Research*. 2014. Vol. 43. Pp. 20-35. DOI: 10.1016/j.cor.2013.08.005

Поступила 13.01.2019; принята к публикации 25.02.2019;
опубликована онлайн 19.04.2019.

Об авторе:

Мамонов Николай Викторович, магистрант, факультет вычислительной математики и кибернетики, Московский государственный университет имени М.В. Ломоносова (119991, Россия, г. Москва, Ленинские горы, д. 1); математик-программист, ООО «Гудфокаст» (117246, Россия, г. Москва, Научный проезд, д. 19, оф. 501), ORCID: <http://orcid.org/0000-0002-6088-7366>, mnv95@mail.ru

Автор прочитал и одобрил окончательный вариант рукописи.

References

- [1] Ulyanov M.V. Resource-efficient computer algorithms. Development and analysis. М.: Fizmatlit, 2008. (In Russ.)
- [2] Tang N.C., Chilkoti A. Combinatorial codon scrambling enables scalable gene synthesis and amplification of repetitive proteins. *Nature Materials*. 2016; 15(4):419-424. (In Eng.) DOI: 10.1038/nmat4521
- [3] Kostevich L.S. Mathematical programming: information technologies of optimal solutions. Мн.: Novoye znaniye, 2003. (In Russ.)
- [4] Goodman S.E., Hedetniemi S.T. Introduction to the Design and Analysis of Algorithms. NY: McGraw-Hill, 1977. (In Eng.)
- [5] Ulyanov M.V., Fomichev M.I. Resource characteristics of ways to organize a decision tree in the branch-and-bound method for the traveling salesman problem. *Business Informatics*. 2015; 4(34):38-46. (In Eng.) DOI: 10.17323/1998-0663.2015.4.38.46
- [6] Goloveshkin V.A., Zhukova G.N., Ulyanov M.V., Fomichev M.I. Probabilistic Prediction of the Complexity of Traveling Salesman Problems Based on Approximating the Complexity Distribution from Experimental Data. *Automation and Remote Control*. 2018; 79(7):1296-1310. (In Eng.) DOI: 10.1134/S0005117918070093
- [7] Lobjois L., Lemaitre M. Branch and Bound Algorithm Selection by Performance Prediction. Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98). American Association for Artificial Intelligence, AAAI Press, Menlo Park, CA, 1998. Available at: <https://aaai.org/Papers/AAAI/1998/AAAI98-050.pdf> (accessed 13.01.2019). (In Eng.)
- [8] Bektaş T., Gouveia L., Martínez-Sykora A., Salazar-González J. Balanced vehicle routing: Polyhedral analysis and branch-and-cut algorithm. *European Journal of Operational Research*. 2019; 273(2):452-463. (In Eng.) DOI: 10.1016/j.ejor.2018.08.034
- [9] Arigliano A., Ghiani G., Grieco A., Guerriero E., Plana I. Time-dependent asymmetric traveling salesman problem with time windows: Properties and an exact algorithm. *Discrete Applied Mathematics*. 2019; 261:28-39. (In Eng.) DOI: 10.1016/j.dam.2018.09.017
- [10] Hore S., Chatterjee A., Dewanji A. Improving variable neighborhood search to solve the traveling salesman problem. *Applied Soft Computing*. 2018; 68:83-91. (In Eng.) DOI: 10.1016/j.asoc.2018.03.048
- [11] Gouveia L., Leitner M., Ruthmair M. Extended formulations and branch-and-cut algorithms for the Black-and-White Traveling Salesman Problem. *European Journal of Operational Research*. 2017; 262(3):908-928. DOI: 10.1016/j.ejor.2017.04.061
- [12] Huang Y., Zhong C. Detecting list-colored graph motifs in biological networks using branch-and-bound strategy. *Computers in Biology and Medicine*. 2019; 107:1-9. (In Eng.) DOI: 10.1016/j.compbiomed.2019.01.025
- [13] Toriello A. Optimal Toll Design: A Lower Bound Framework for the Asymmetric Traveling Salesman Problem. *Mathematical Programming*. 2014; 144(1-2):247-264. (In Eng.) DOI: 10.1007/s10107-013-0631-6
- [14] Sergeev S. Nonlinear resolving functions for the travelling salesman problem. *Automation and Remote Control*. 2013; 74(6):978-994. (In Eng.) DOI: 10.1134/S0005117913060088
- [15] Charikar M., Goemans M.X., Karloff H. On the Integrality Ratio for the Asymmetric Traveling Salesman Problem. *Mathematics of Operations Research*. 2006; 31(2):245-252. DOI: 10.1287/moor.1060.0191
- [16] Subramanyam A., Gounaris C. A branch-and-cut framework for the consistent traveling salesman problem. *European Journal of Operational Research*. 2016; 248(2):384-395. (In Eng.) DOI: 10.1016/j.ejor.2015.07.030
- [17] Goloveshkin V.A., Zhukova G.N., Ulyanov M.V., Fomichev M.I. The estimation of the complexity of solving a particular travelling salesman problem by quantile-based measures for skewness and kurtosis. *International Journal of Open Information Technologies*. 2016; 4(12):7-12. Available at: <https://elibrary.ru/item.asp?id=27543353> (accessed 13.01.2019). (In Russ.)
- [18] Liškiewicz M., Schuster M. A new upper bound for the traveling salesman problem in cubic graphs. *Journal of Discrete Algorithms*. 2014; 27:1-20. (In Eng.) DOI: 10.1016/j.jda.2014.02.001
- [19] Kara I., Derya T. Formulations for Minimizing Tour Duration of the Traveling Salesman Problem with Time Windows. *Procedia Economics and Finance*. 2015; 26:1026-1034. (In Eng.) DOI: 10.1016/S2212-5671(15)00926-0
- [20] LaRusic J., Punnen A.P. The asymmetric bottleneck traveling salesman problem: Algorithms, complexity and empirical analysis. *Computers & Operations Research*. 2014; 43:20-35. (In Eng.) DOI: 10.1016/j.cor.2013.08.005

Submitted 13.01.2019; revised 25.02.2019;
published online 19.04.2019.

About the author:

Nikolai V. Mamonov, Master's Degree student, Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University (1 Leninskie Gory, Moscow 119991, Russia); Software Developer, GoodsForecast (19 Nauchnyj pr., Moscow 117246, Russia), ORCID: <http://orcid.org/0000-0002-6088-7366>, mnv95@mail.ru

The author has read and approved the final manuscript.

