

УДК 519.6
DOI: 10.15827/0236-235X.125.055-062

Дата подачи статьи: 05.12.18
2019. Т. 32. № 1. С. 055–062

О применении жадных алгоритмов в некоторых задачах дискретной математики

В.А. Бойков¹, к.ф.-м.н., старший научный сотрудник, vboykov@inbox.ru

¹ Одинцовский филиал МГИМО МИД России, г. Одинцово, 143007, Россия

Алгоритмы, основанные на идее локальной оптимальности, кажутся естественными и являются соблазном при решении задач оптимизации. Однако задачи оптимизации, которые будут рассмотрены в данной статье, многошаговые, а получение жадными алгоритмами оптимального решения в многошаговых задачах, вообще говоря, не гарантировано. Это будет показано на примерах решения транспортной задачи, задачи о кратчайшем расстоянии между городами на заданной сети дорог и задачи коммивояжера. Объектами проведенного исследования являются жадные алгоритмы, примененные к решениям таких задач.

В работе приводится пример парадоксального решения одной транспортной задачи малой размерности. При ее решении одним из жадных алгоритмов строится план перевозок продукции. Однако этот план не является оптимальным и обладает парадоксальным свойством. А именно: по маршруту, который оказывается самым дешевым в оптимальном плане, перевозка не осуществляется. Оптимальное решение рассмотренной задачи достигается с помощью пакета Mathcad.

На примере задачи о кратчайшем расстоянии показано, что жадный алгоритм тоже не приводит к оптимальному пути. А три контрпримера на евклидовых графах показывают, что, вообще говоря, невозможно получить оптимальный маршрут даже при расчете вариантов на несколько шагов вперед.

В качестве третьего примера применения жадного алгоритма для решения задачи коммивояжера рассматривается метод ближайших городов. С его помощью описывается последовательное построение гамильтонова цикла. Приведенная версия алгоритма застрахована от получения несвязных графов в процессе решения. Далее длина полученного гамильтонова цикла используется как верхняя оценка при реализации простейшей версии метода ветвей и границ. Оптимальность полученного решения проверяется с помощью программы, выполненной в пакете Mathcad.

В рассмотренных примерах решения, полученные с помощью жадных алгоритмов, используются в качестве начального приближения для дальнейшей оптимизации целевой функции.

Ключевые слова: жадный алгоритм, транспортная задача, парадоксальное решение транспортной задачи, задача о кратчайшем расстоянии, задача коммивояжера, задача коммивояжера как задача математического программирования, метод ветвей и границ.

Впервые жадный алгоритм (greedy algorithm) – алгоритм, заключающийся в принятии локально оптимальных решений на каждом этапе, был описан в работе по комбинаторной оптимизации [1]. Поток исследований, последовавших за этой публикацией, обобщен в работах [2, 3]. Основным направлением исследования являлось выявление задач, в которых оптимальное решение получалось жадными алгоритмами. Приводилось обоснование оптимальности жадных алгоритмов для их решения. В одной из последних фундаментальных книг рассматриваются приложения жадных алгоритмов для получения оптимальных решений [4]. В частности, рассматриваются такие задачи, как планирование заданий и построение кодов Хаффмана в теории кодирования.

Пример парадоксального решения транспортной задачи

Поводом для данного исследования послужил пример парадокса, возникающего при решении одной транспортной задачи малой размерности [5].

Постановка транспортной задачи заключается в следующем. Имеются два пункта (A_1, A_2) производства продукции и два пункта (B_1, B_2) потребления этой продукции. Объемы производства (предложения) продукции в пунктах A_i равны a_i ($i = 1, 2$). Объемы потре-

бления (спроса) продукции в пунктах B_j равны b_j ($j = 1, 2$). Заданы удельные затраты на перевозку единицы продукции c_{ij} ($i = 1, 2, j = 1, 2$) по маршруту из пункта A_i в пункт B_j .

Обозначим x_{ij} ($i = 1, 2, j = 1, 2$) объем продукции, перевозимой из пункта A_i в пункт B_j .

Требуется спланировать объемы перевозок по заданным маршрутам из пунктов A_1 и A_2 в пункты B_1 и B_2 так, чтобы суммарные затраты на все перевозки были минимальными, то есть требуется найти такие объемы перевозок, которые обеспечивают минимум

$$\text{функции: } \min f(\mathbf{X}), \quad f(\mathbf{X}) = \sum_{i=1}^2 \sum_{j=1}^2 c_{ij} x_{ij}. \quad (1)$$

Здесь \mathbf{X} – таблица (матрица), составленная из величин объемов перевозок по заданным маршрутам,

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{pmatrix}.$$

При этом требуется, чтобы объемы перевозок удовлетворяли условиям

$$\begin{cases} x_{11} + x_{12} = a_1, & x_{21} + x_{22} = a_2, \\ x_{11} + x_{21} = b_1, & x_{12} + x_{22} = b_2, \end{cases} \quad (2)$$

$$(x_{ij} \geq 0, i, j = 1, 2).$$

Первые два равенства означают, что вся продукция из пунктов A_1 и A_2 вывезена. Два равенства, записан-

ные во второй строке, означают, что в пункты B_1 и B_2 завезена продукция в требуемом объеме. Интерпретация условий неотрицательности на объемы перевозок очевидна.

Чтобы удовлетворить весь спрос, необходимо и достаточно, чтобы суммарный спрос был равен суммарному предложению, то есть должны выполняться условия баланса:

$$a_1 + a_2 = b_1 + b_2. \tag{3}$$

Обсудим решение транспортной задачи жадным алгоритмом – *методом минимальной стоимости* (ММС), которое многим покажется естественным или интуитивно очевидным. Опишем это решение по шагам для исходных данных из [5]:

$$C = \begin{pmatrix} 12 & 16 \\ 8 & 10 \end{pmatrix}, \quad a = \begin{pmatrix} 5 \\ 7 \end{pmatrix}, \quad b = \begin{pmatrix} 4 \\ 8 \end{pmatrix}.$$

Шаг 1. Найдем маршрут минимальной стоимости, то есть соответствующий минимальному элементу матрицы C , а именно: $\min c_{ij} = c_{21} = 8; i = 2, j = 1$.

Тогда из пункта A_2 в пункт B_1 планируем перевозку: $x_{21} := \min\{a_2, b_1\} = \min\{7, 4\} = 4$.

Пересчитываем величины спроса на продукцию и остатков продукции:

$$b_1 := b_1 - 4 = 4 - 4 = 0, \quad a_2 := a_2 - 4 = 7 - 4 = 3.$$

Поскольку в пункт B_1 продукция завезена полностью, из пункта A_1 в пункт B_1 уже ничего везти не надо. Поэтому полагаем $x_{11} := 0$.

Итак, в матрице перевозок X остаются незаполненными элементы второго столбца x_{12}, x_{22} ; $X = \begin{pmatrix} 0 & - \\ 4 & - \end{pmatrix}$.

Шаг 2. Как и на первом шаге, в матрице C найдем минимальный элемент, соответствующий этим перевозкам, то есть $\min\{c_{12}, c_{22}\} = \min\{16, 10\} = 10 = c_{22}; i = 2, j = 2$.

Это означает, что из пункта A_2 в пункт B_2 планируем перевозку по тому же правилу:

$$x_{22} := \min\{a_2, b_2\} = \min\{3, 7\} = 3.$$

Пересчитываем величины спроса на продукцию и остатков продукции:

$$b_2 := b_2 - 3 = 8 - 3 = 5, \quad a_2 := a_2 - 4 = 3 - 3 = 0.$$

В матрицу перевозок X записываем найденную перевозку $x_{22} = 3$. Остается незаполненным элемент x_{12} ;

$$X = \begin{pmatrix} 0 & - \\ 4 & 3 \end{pmatrix}.$$

Шаг 3. Минимальный элемент матрицы C , соответствующий незаполненному элементу матрицы X , есть элемент c_{12} . Элемент x_{12} заполняем по тому же правилу: $x_{12} := \min\{a_1, b_2\} = \min\{5, 5\} = 5$.

Пересчитываем величины спроса на продукцию и остатков продукции:

$$b_2 := b_2 - 5 = 5 - 5 = 0, \quad a_1 := a_1 - 5 = 5 - 5 = 0.$$

Как видим, вся продукция из пунктов A_1 и A_2 вывезена в тех объемах, в которых она там произведена. В пункты B_1 и B_2 завезено столько продукции, сколько требовалось. Это свойство следует из равенства суммарного спроса и суммарного предложения (3).

Получили план перевозок, который обозначим как

$$X_{\text{ММС}} = \begin{pmatrix} 0 & 5 \\ 4 & 3 \end{pmatrix}.$$

Вычислим суммарные затраты на запланированные объемы перевозок: $f(X_{\text{ММС}}) = c_{11} \cdot x_{11} + c_{12} \cdot x_{12} + c_{21} \cdot x_{21} + c_{22} \cdot x_{22} = 12 \cdot 0 + 16 \cdot 5 + 8 \cdot 4 + 10 \cdot 3 = 142$.

Описанный метод получения плана перевозок называется ММС.

Проверим оптимальность решения задачи. Продемонстрируем, как легко и наглядно получается наилучший план перевозок с использованием математического пакета Mathcad. Приведем решение рассмотренной транспортной задачи в Mathcad (табл. 1).

Таблица 1

Программа Mathcad. Решение транспортной задачи

Table 1

Mathcad. The solution of a transportation problem

Оператор Mathcad	Комментарий
ORIGIN := 1	Встроенная функция Mathcad, задающая минимальные значения индексов матрицы
$C := \begin{pmatrix} 12 & 16 \\ 8 & 10 \end{pmatrix}$ $a := \begin{pmatrix} 5 \\ 7 \end{pmatrix} \quad b := \begin{pmatrix} 4 \\ 8 \end{pmatrix}$	Исходные данные транспортной задачи
$f(X) := \sum_{i=1}^2 \sum_{j=1}^2 (C_{ij} \cdot X_{ij})$	Формула вычисления суммарных транспортных затрат
$X := \begin{pmatrix} 11 & 12 \\ 21 & 22 \end{pmatrix}$	Начальное приближение для значений элементов матрицы перевозок (могут быть любыми числами)
Given	Начало блока решения задач
$X_{1,1} + X_{1,2} = a_1$ $X_{2,1} + X_{2,2} = a_2 \quad X \geq 0$ $X_{1,1} + X_{2,1} = b_1$ $X_{1,2} + X_{2,2} = b_2$	Ограничения для транспортной задачи. Неравенство означает, что все элементы матрицы X больше или равны нулю
$X_{\text{opt}} := \text{Minimize}(f, X)$ $X_{\text{opt}} = \begin{pmatrix} 4 & 1 \\ 0 & 7 \end{pmatrix}$	Вызов функции <i>Minimize</i> , которая находит такую матрицу X , которая минимизирует функцию f . $X_{\text{opt}} :=$ присвоение оптимальной матрицы
$f(X_{\text{opt}}) = 134$	Вычисление значений суммарных затрат для матрицы перевозок X_{opt}

Обратим внимание, что значение суммарных затрат на перевозку грузов по плану, построенному пакетом Mathcad, получилось меньше, чем при решении ММС, а именно: $f(X_{\text{ММС}}) = 142 > 134 = f(X_{\text{opt}})$.

Кроме того, в оптимальном плане перевозок перевозка по маршруту минимальной стоимости не осуществляется ($c_{21} = 8$, но $x_{21}^{\text{opt}} = 0$).

Проверим, действительно ли решение, полученное компьютером, оптимальное. Для вычислителей, использующих компьютер для решения каких-либо задач, возникает вопрос о достоверности ответа.

Для этого часто используются необходимые и достаточные условия оптимальности. Условия оптимальности для транспортной задачи с m пунктами производства продукции и n пунктами потребления продукции приведены ниже.

Теорема [6]. Для оптимальности плана перевозок

$$X^* = \begin{pmatrix} x_{11}^* & \dots & x_{1n}^* \\ \dots & \dots & \dots \\ x_{m1}^* & \dots & x_{mn}^* \end{pmatrix}$$

в транспортной задаче необходимо и достаточно, чтобы существовали числа u_i и v_j ($i = \overline{1, m}, j = \overline{1, n}$), называемые *потенциалами*, такие, чтобы выполнялись условия:

а) для $x_{ij}^* > 0$ выполняются равенства $u_i + v_j = c_{ij}$;
 б) для остальных x_{ij}^* выполняются неравенства $u_i + v_j \leq c_{ij}$.

Проверим, является ли найденный план $X_{ММС}$ оптимальным. В данном случае $m = 2, n = 2$. Заметим, что неизвестных четыре (u_1, u_2, v_1, v_2), а положительных перевозок в плане X_{opt} три ($x_{11} = 4, x_{12} = 1, x_{22} = 7$). То есть условия а) задают систему трех линейных уравнений (по количеству положительных перевозок) с четырьмя неизвестными. Общее решение такой системы линейных уравнений зависит от одной переменной, значения которой можно назначать произвольно.

Составим систему линейных уравнений а) для полученного плана X_{opt} , выданного компьютером:

$$\begin{cases} \text{для } x_{11} = 4 > 0 & \text{запишем уравнение } u_1 + v_1 = c_{11} = 12, \\ \text{для } x_{12} = 1 > 0 & \text{запишем уравнение } u_1 + v_2 = c_{12} = 16, \\ \text{для } x_{22} = 7 > 0 & \text{запишем уравнение } u_2 + v_2 = c_{22} = 10. \end{cases}$$

Положив в 1-м и 2-м уравнениях $u_1 = 0$, получаем потенциалы $v_1 = 12, v_2 = 16$. Из третьего уравнения следует $u_2 = 10 - v_2 = 10 - 16 = -6$.

Проверим условие б) для нулевой перевозки $x_{21} = 0$. Оказывается, что $u_2 + v_1 = -6 + 12 = 6 < c_{21} = 8$. Это означает, что данный план перевозок X_{opt} является оптимальным в рассматриваемой транспортной задаче, то есть минимизирует суммарную стоимость перевозок.

Итак, выяснили, что ММС не дает оптимального решения транспортной задачи сразу. Этот метод используется лишь для получения начального плана перевозок, который затем по шагам преобразуется в оптимальный план специальными вычислительными методами, например, методом потенциалов [7].

В чем же парадокс решения транспортной задачи? План перевозок строился по принципу – на каждом шаге выбирается перевозка по маршруту, соответствующему минимальной стоимости, однако ММС (жадный алгоритм) не дал оптимального решения.

Таким образом, очевидно, что имеются такие транспортные задачи, оптимальное решение которых не дает маршрут минимальной стоимости. В рассмотренном примере такой маршрут один – x_{21} .

Применение жадных алгоритмов в задаче о кратчайшем пути

Рассмотрим пример (рис. 1). Пусть имеются 10 городов. Номера городов на схеме указаны в кружках. Стрелки означают дороги (пути возможных переходов из города в город), а числа над/под стрелками – длины соответствующих дорог (расстояния между городами). Требуется найти кратчайший путь из города 1 в город 10.

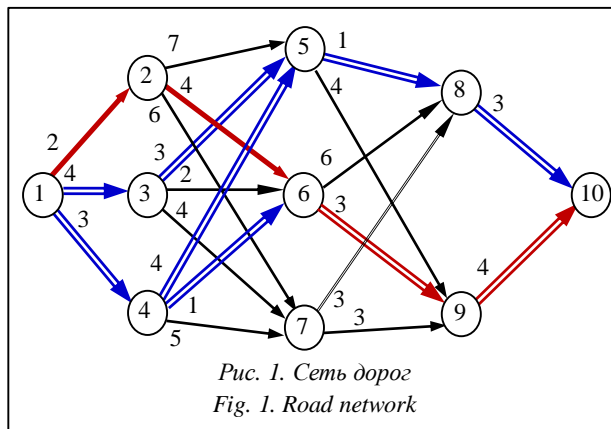


Рис. 1. Сеть дорог
Fig. 1. Road network

Попытаемся решить эту задачу жадным алгоритмом – методом ближайшего города (МБГ) (рис. 2). Маршрут № 1, полученный этим методом, на рисунке обозначен жирными стрелками.

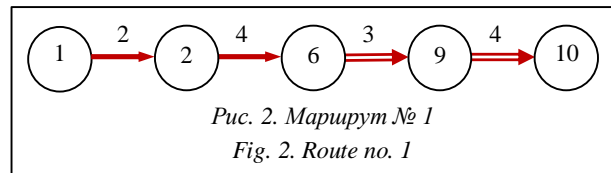


Рис. 2. Маршрут № 1
Fig. 2. Route no. 1

Длина пройденного пути $F_1^{МБГ} = 2 + 4 + 3 + 4 = 13$.

Эту же задачу можно решить, например, методом динамического программирования Р. Беллмана [8].

Оптимальные последовательности прохождения городов (маршруты 2, 3, 4 соответственно) представлены на рисунке 3.

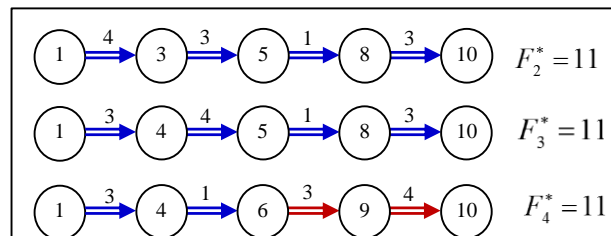


Рис. 3. Маршруты 2, 3, 4
Fig. 3. Routes no. 2, 3, 4

Справа от оптимальных траекторий выписаны оптимальные значения F_2^* , F_3^* , F_4^* суммарных пройденных расстояний по соответствующим маршрутам.

Итак, с помощью жадного алгоритма МБГ не удалось сразу получить оптимальный маршрут для перехода из города 1 в город 10 ($13 = F_1^{МБГ} > F^* = 11$).

Для выяснения природы этого явления продемонстрируем несколько геометрических контрпримеров, показывающих, что расчет вариантов даже на несколько шагов вперед (но не до конца), вообще говоря, не приводит к оптимальному решению.

Рассмотрим задачу о поиске кратчайшего пути из точки A в точку B по сети дорог, соединяющих конечное множество заданных точек.

Контрпример 1. Даны точки A, B, C, M , соединенные дорогами, которые обозначены сплошными линиями (рис. 4). Пунктирный отрезок AB не входит в заданную сеть дорог. Требуется найти кратчайший путь из точки A в точку B . Действуя жадным алгоритмом, из точки A переходим в ближайшую точку C и проигрываем, так как, несмотря на то, что $AC < AM$, выполняется неравенство $AC + CM + MB > AM + MB$.

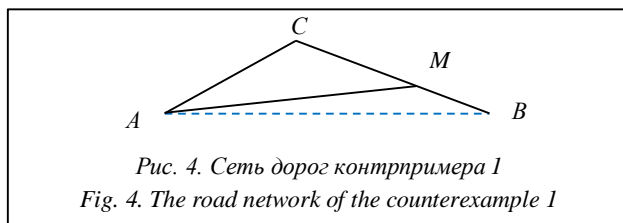


Рис. 4. Сеть дорог контрпримера 1
Fig. 4. The road network of the counterexample 1

Жадных решателей не спасает и алгоритм расчета на два хода вперед.

Контрпример 2. Заданы точки A, B, C, D_1, D_2, M , соединенные дорогами, которые обозначены сплошными линиями (рис. 5). Пунктирный отрезок AB не входит в заданную сеть дорог. Требуется найти кратчайший путь из точки A в точку B по существующей сети дорог. Рассмотрим сеть дорог, такую, что

$$AD_1 + D_1D_2 < AC + CM, \tag{4}$$

но $AD_1 + D_1D_2 + D_2M + MB > AC + CM + MB. \tag{5}$

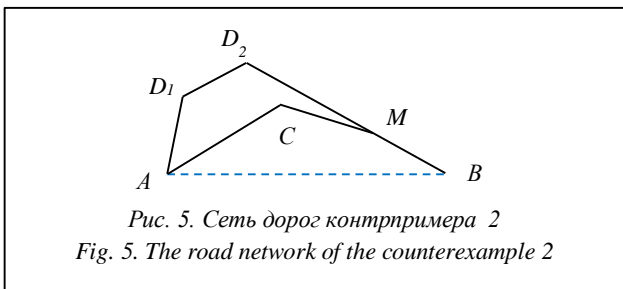


Рис. 5. Сеть дорог контрпримера 2
Fig. 5. The road network of the counterexample 2

Жадный решатель, находящийся в точке A , вычисляет сумму расстояний на два хода вперед. Убеждается, что выполняется неравенство (4), осуществляет переход из точки A в точку D_2 и, очевидно, проигрывает по сумме пройденного расстояния из точки A в точку B (см. неравенство (5)).

Жадных решателей не спасает и алгоритм расчета на n ($n > 2$) ходов вперед.

Контрпример 3. Заданы точки $A, D_1, D_2, \dots, D_{n-1}, D_n, M, B, C_1, C_2, \dots, C_{n-1}$, соединенные дорогами (рис. 6). Дороги обозначены сплошными линиями, пунктирами – дороги из точки D_2 в D_{n-1} и из точки C_2 в C_{n-1} . Пунктирный отрезок AB не входит в заданную сеть дорог. Требуется найти кратчайший путь из точки A в точку B по существующей сети дорог. По аналогии с предыдущими примерами построим такую сеть дорог, что выполняются два неравенства:

$$AD_1 + D_1D_2 + \dots + D_{n-1}D_n < AC_1 + C_1C_2 + \dots + C_{n-1}M, \tag{6}$$

$$\text{но } AD_1 + D_1D_2 + \dots + D_{n-1}D_n + D_nM + MB > AC_1 + C_1C_2 + \dots + C_{n-1}M + MB. \tag{7}$$

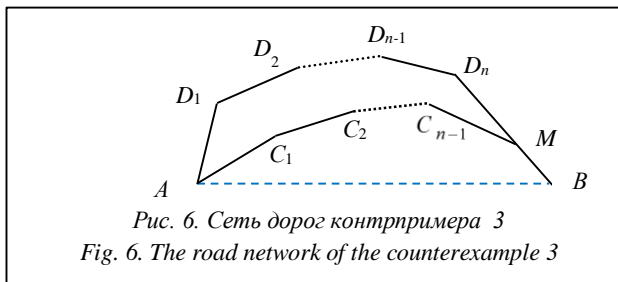


Рис. 6. Сеть дорог контрпримера 3
Fig. 6. The road network of the counterexample 3

И в этом случае жадный решатель, находящийся в точке A , вычисляет сумму расстояний на n ходов вперед, убеждается, что выполняется неравенство (6), и осуществляет переход из точки A в точку D_n и, очевидно, проигрывает по сумме пройденного расстояния из точки A в точку B (см. неравенство (7)).

Рассмотренная задача поиска кратчайшего пути имеет переборный характер. Математики-вычислители постоянно работают над специальными алгоритмами решения таких задач для большого количества точек. Практика решения этих задач показывает, что идеи так называемых жадных алгоритмов, вообще говоря, не проходят.

Использование жадного алгоритма в задаче коммивояжера

Постановка задачи коммивояжера. Имеются n городов (точек). Задана матрица расстояний между городами $C[n \times n] = (c_{ij})$, где c_{ij} – расстояние между городами i и j ($i, j = \overline{1, n}$). Выехав из одного города, коммивояжер должен побывать во всех остальных городах по одному разу и вернуться в исходный город, то есть маршрут должен быть замкнутым. Такие маршруты называются *гамильтоновыми циклами*.

Требуется найти такой гамильтонов цикл, который минимизирует пройденный путь.

Суммарное пройденное расстояние по гамильтонову циклу и требование его минимизации можно записать в виде

$$c_{i_1 i_2} + c_{i_2 i_3} + \dots + c_{i_{n-1} i_n} + c_{i_n i_1} \rightarrow \min. \tag{8}$$

Минимум ищется по всем перестановкам (i_1, i_2, \dots, i_n) номеров городов $(1, 2, \dots, n)$. Количество таких перестановок равно $n!$. Количество различных гамильтоновых циклов в n раз меньше, то есть $(n - 1)!$.

При достаточно больших значениях n решение задачи коммивояжера, основанное на методе полного перебора, становится проблематичным. Поэтому представляют интерес методы решения задачи коммивояжера, не связанные с полным перебором вариантов.

Формализация задачи коммивояжера как задачи математического программирования

Введем переменные, отражающие факт перехода из точки i в точку j :

$$x_{ij} = \begin{cases} 1, & \text{если коммивояжер из точки } i \\ & \text{переезжает в точку } j, \\ 0 & \text{в противном случае,} \end{cases} \quad (9)$$

где $i, j = \overline{1, n}$.

Тогда задача ставится следующим образом [9].

Найти значения булевых переменных x_{ij} , которые удовлетворяют условиям

$$\min f(X), \quad f(X) = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}, \quad (10)$$

с ограничениями вида

$$\sum_{j=1}^n x_{ij} = 1 \quad \text{для всех } i = \overline{1, n}; \quad (11)$$

$$\sum_{i=1}^n x_{ij} = 1 \quad \text{для всех } j = \overline{1, n}; \quad (12)$$

$$u_i - u_j + n \cdot x_{ij} \leq n - 1; \quad i, j = \overline{2, n}; \quad i \neq j. \quad (13)$$

Будем считать, что все $c_{ii} = +\infty, i = \overline{1, n}, c_{ij} \geq 0$ и, вообще говоря, $c_{ij} \neq c_{ji}$.

Ограничения (11) и (12) означают, что из каждой точки можно выйти и войти в каждую точку ровно один раз. Ограничения (13) гарантируют невозможность распада гамильтонова цикла на несвязные циклы.

Задача (9)–(13) представляет собой задачу булева линейного программирования.

Используя эту формализацию, решим задачу коммивояжера с помощью математического пакета Mathcad (табл. 2). Пусть задана матрица расстояний между четырьмя городами ($n = 4$).

Замечание. Поскольку оптимальные значения элементов матрицы $X = X_{\text{opt}}$ получились булевыми, ограничения (9) выполнены автоматически.

Рассмотрим решение задачи методом ветвей и границ [10]. Для решения задачи коммивояжера этот метод был предложен в 1963 году американскими математиками Литтл, Мурти и др. (Little, Murty, Sweeney, Karel).

Идея метода ветвей и границ применительно к задаче коммивояжера проста.

Таблица 2

Программа Mathcad. Решение задачи коммивояжера
Mathcad. The solution of the traveling salesman problem

Оператор Mathcad	Комментарий
ORIGIN := 1	Встроенная функция Mathcad, задающая минимальные значения индексов матрицы
$w := 10^{13} \quad C := \begin{pmatrix} w & 3 & 1 & 16 \\ 5 & w & 4 & 3 \\ 2 & 3 & w & 9 \\ 8 & 7 & 6 & w \end{pmatrix}$ $X := \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7 \end{pmatrix} \quad u := \begin{pmatrix} 0 \\ 0 \\ 0 \\ 8 \end{pmatrix}$	Исходные данные: C – матрица расстояний между городами; w – большое число; X – начальное значение матрицы исходных маршрутов; u – вспомогательные переменные
$f(X, u) := \sum_{i=1}^4 \sum_{j=1}^4 (C_{ij} \cdot X_{ij})$	Формула вычисления суммарной длины гамильтонова цикла для матрицы маршрутов X – формула (9)
Given	Начало блока решения задач
$\sum_{i=1}^4 X_{<i>} = 1$ $u_2 - u_2 + 4 \cdot X_{2,2} \leq 3$ $u_3 - u_2 + 4 \cdot X_{3,2} \leq 3$ $u_4 - u_2 + 4 \cdot X_{4,2} \leq 3$ $\sum_{j=1}^4 (X^T)^{<j>} = 1$ $u_2 - u_3 + 4 \cdot X_{2,3} \leq 3$ $u_3 - u_3 + 4 \cdot X_{3,3} \leq 3$ $u_4 - u_3 + 4 \cdot X_{4,3} \leq 3$ $X \geq 0$ $u_2 - u_4 + 4 \cdot X_{2,4} \leq 3$ $u_3 - u_4 + 4 \cdot X_{3,4} \leq 3$ $u_4 - u_4 + 4 \cdot X_{4,4} \leq 3$	Ограничения (10)–(12) для задачи коммивояжера без условий булевости элементов матрицы X ($n = 4, n - 1 = 3$)
$v := \text{Minimize}(f, X, u) = \begin{pmatrix} \{4, 4\} \\ \{4, 1\} \end{pmatrix}$ $X_{\text{opt}} := v_1 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$ $f(X_{\text{opt}}, v_2) = 14$	Вызов функции Minimize, находящей такую матрицу X , которая минимизирует функцию f . X_{opt} := присвоение оптимальной матрицы маршрутов. Вычисление значений суммарной длины оптимального гамильтонова цикла (X_{opt})
$M := C_{1,3} + C_{3,2} + C_{2,4} + C_{4,1} = 15$	Вычисление значений суммарной длины оптимального гамильтонова цикла, полученного МБГ

Ветвление основано на следующем соображении: проезд из любого города i в любой другой город j может либо принадлежать оптимальному циклу, либо не

принадлежать. В процессе решения необходимо сделать вывод – принадлежит ли путь из города i в другой город j оптимальному циклу. Для этого используется понятие *границы*, которая позволяет сделать отсев неперспективных гамильтоновых циклов. Этот прием часто позволяет избежать полного перебора вариантов, то есть уменьшить количество вычислений. Однако метод ветвей и границ не гарантирует, что процесс решения задачи коммивояжера не сведется к полному перебору всех возможных гамильтоновых циклов.

Изложим идею метода ветвей и границ с использованием аналога метода минимальной стоимости для решения транспортной задачи. Для задачи коммивояжера аналог этого метода – МБГ.

Первый этап. Построение гамильтонова цикла с помощью МГБ.

Шаг 1. В матрице C найдем минимальный элемент. В данном случае $\min c_{ij} = c_{13} = 1$ ($i = 1, j = 3$). Это означает, что для первого города ближайшим к нему является третий город. Составим список городов, в которых уже побывали, $A = \{1, 3\}$, и список городов, в которых еще не были, $B = \{2, 4\}$.

Очевидно, $A \cup B = \{1, 2, 3, 4\}$ – множество всех городов. Контролировать состояние двух списков необходимо для предотвращения преждевременного возвращения в исходный город.

Шаг 2. Из третьего города переходим в ближайший город среди тех городов, в которых еще не были. Находим $\min_{j \in B} c_{3j} = c_{32} = 3$ ($j = 2$), то есть из третьего города переходим во второй.

Обновляем списки городов A и B : $A = \{1, 3, 2\}$, $B = \{4\}$.

Шаг 3. Из второго города переходим в ближайший город среди тех городов, в которых еще не были. Находим $\min_{j \in B} c_{2j} = c_{24} = 8$ ($j = 4$), то есть из второго города переходим в четвертый.

Обновляем списки городов A и B : $A = \{1, 3, 2, 4\}$, $B = \{\emptyset\}$.

Шаг 4. Из четвертого города переходим в первый, так как все города пройдены (множество B не содержит элементов), то есть наступил момент возвращения в исходный город.

Шаг 5. Подсчитываем длину l пройденного пути по построенному маршруту $(1, 3, 2, 4)$: $l(1, 3, 2, 4) = c_{13} + c_{32} + c_{24} + c_{41} = 1 + 3 + 3 + 8 = 15$. $M = 15$.

Число M объявим верхней границей для искомой минимальной длины гамильтонова цикла.

Очевидно, что минимум среди всех длин гамильтоновых циклов удовлетворяет неравенству

$$\min \left(c_{i_1 i_2} + c_{i_2 i_3} + c_{i_3 i_4} + c_{i_4 i_1} \right) \leq M.$$

Итак, с помощью МБГ найдена оценка сверху для оптимального значения целевой функции.

Второй этап. На втором этапе решения задачи коммивояжера продемонстрируем алгоритм метода вет-

вей и границ, использующий найденную границу как критерий отсева неперспективных гамильтоновых циклов. Изобразим перебор вариантов ветвления схематично (рис. 7).

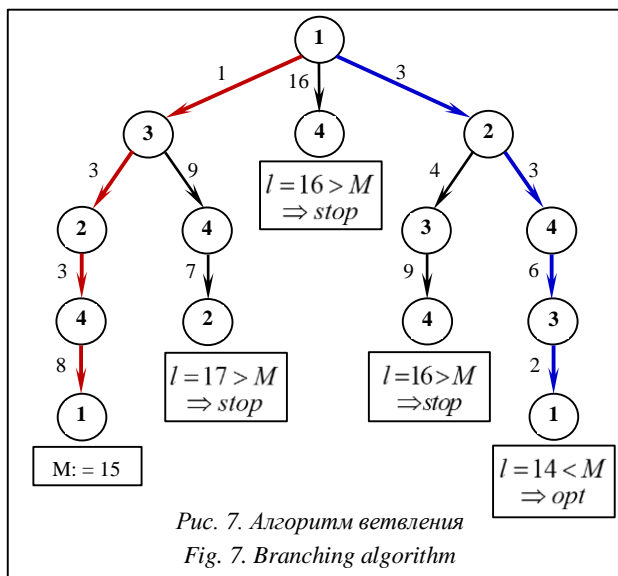


Рис. 7. Алгоритм ветвления
Fig. 7. Branching algorithm

На рисунке 7 представлено дерево ветвления вариантов построения гамильтоновых циклов.

Левая ветка дерева построена с помощью МБГ. Начав движение из города 1, последовательно переходим через города 3, 2, 4 и, пройдя все города, возвращаемся в начальный город 1. При этом вычисляем длину пройденного пути (гамильтонова цикла). Она равна 15.

Эту величину принимаем за оценку сверху (границу) для дальнейшего перебора вариантов ($M = 15$).

Рассмотрим ветку построенного дерева $(1 \rightarrow 3 \rightarrow 4 \rightarrow 2)$. При этом на каждом переходе от города к городу вычисляем накопленную сумму пройденного пути ($l = 1 + 9 + 7 = 17 > M \Rightarrow stop$). Поскольку получена величина, большая оценки сверху, дальнейшее продолжение этого маршрута нецелесообразно.

Следующая ветка для рассмотрения $(1 \rightarrow 4)$. Для этой ветки имеем ($l = 16 > M \Rightarrow stop$), потому и ее отбрасываем как неперспективную.

Рассмотрим ветку дерева $(1 \rightarrow 2 \rightarrow 3 \rightarrow 4)$. Для нее получаем ($l = 3 + 4 + 9 = 16 > M \Rightarrow stop$). Снова отбрасываем ветку как неперспективную.

Осталась последняя ветка дерева $(1 \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow 1)$. Обратим внимание, что накопленная сумма пройденных расстояний равна ($l = 3 + 3 + 6 + 2 = 14 < M \Rightarrow opt$). В этом случае оказалось, что установлен рекорд – сумма пройденного пути меньше предыдущего рекорда.

Обновляем величину рекорда (верхняя граница) ($M = l = 14$). Если бы дерево вариантов имело больше веток, для отсева неперспективных вариантов использовалось бы уже обновленное значение верхней границы. Поскольку рассмотрены все варианты, последний гамильтонов цикл оптимальный.

Заметим, что при последовательном переборе вариантов циклов в данном примере не пришлось строить все циклы до конца и вычислять их длину. В этом достоинство метода ветвей и границ. Существенным является получение хорошей оценки (границы) для оптимального значения длины гамильтонова цикла. Это позволяет как можно раньше отсекал неперспективные ветки дерева вариантов (гамильтоновы циклы) и тем самым уменьшать объем вычислительной работы.

В данном примере жадный алгоритм (МБГ) дал неплохую верхнюю границу длины гамильтонова цикла.

Оценим выигрш от применения данного метода для решения рассмотренной задачи коммивояжера. Будем считать, что трудоемкость решения задачи определяется количеством сложений для вычисления длины цикла. Количество сложений при полном переборе вариантов равно количеству дуг графа $4! = 24$. В рассматриваемом случае затрачено 16 сложений, что составляет примерно 66,7 % от общего числа вариантов. Однако метод ветвей и границ не дает гарантии, что не придется осуществлять полный перебор вариантов.

Рассмотренные примеры показывают, что решение оптимизационных многошаговых задач не является тривиальной задачей и интуитивно очевидные жадные алгоритмы редко приводят к оптимальным решениям

без дополнительных вычислительных процедур. Для решения рассмотренных задач используются специальные математические алгоритмы и программы, широко применяемые в прикладных задачах. При этом жадные алгоритмы часто используют для получения начального приближения к оптимальному решению.

Литература

1. Edmonds J. Matroids and the greedy algorithm. *Math. Programming*, 1971, vol. 1, no. 2, pp. 127–136.
2. Пападимитриу Х., Стайглиц К. Комбинаторная оптимизация: Алгоритмы и сложность; [пер. с англ.]. М.: Мир, 1984. 510 с.
3. Дасгупта С., Пападимитриу Х., Вазирани У. Алгоритмы; [пер. с англ.; под ред. А. Шеня]. М.: МЦНМО, 2014. 320 с.
4. Кормен Т.Х., Лейзерсон Ч.И., Ривест Р.Л., Штайн К. Алгоритмы: построение и анализ; [пер. с англ.]. М.: Вильямс, 2013. 1328 с.
5. Очков В.Ф., Богомолова Е.П., Иванов Д.А. Физико-математические этюды с Mathcad и Интернет. СПб: Лань, 2016. 388 с.
6. Юдин Д.Б., Гольштейн Е.Г. Задачи и методы линейного программирования. Задачи транспортного типа. М.: Либроком, 2010. 184 с.
7. Грешилов А.А. Прикладные задачи математического программирования. М.: Логос, 2006. 288 с.
8. Беллман Р. Динамическое программирование; [пер. с англ.; под ред. Н.Н. Воробьева]. М.: Иностран. лит-ра, 1960. 400 с.
9. Костевич Л.С., Лапко А.А. Исследование операций. Теория игр. Минск: Выш. шк., 2008. 368 с.
10. Таха Х.А. Введение в исследование операций; [пер. с англ.]. М.: Вильямс, 2005. 912 с.

On the application of greedy algorithms in some problems of discrete mathematics

V.A. Boykov¹, Ph.D. (Physics and Mathematics), Senior Researcher, vboykov@inbox.ru

¹ Odintsovo branch of MGIMO University, Odintsovo, 143007, Russian Federation

Abstract. The algorithms that are based on the idea of local optimality seem natural and tempting when solving optimization problems. However, the optimization problems discussed in the paper are multistage. In this case, the obtaining an optimal solution in a multistage problem by greedy algorithms is not guaranteed generally. This fact is demonstrated by the examples of solving a transport problem, the problem of the shortest distance between cities on a given road network, and the traveling salesman problem.

The research objects are greedy algorithms applied to solving the same problems described in this paper.

The paper gives an example of a paradoxical solution of a small dimension transportation problem. When solving the problem, one of greedy algorithms constructs a product transportation plan. However, this plan is not optimal and has a paradoxical property. Namely, no transportation by the route that is the cheapest in the optimal plan. The optimal solution of the considered problem is given by mathematical package Mathcad.

The fact that the greedy algorithm does not show the optimal path is shown on the example of the shortest distance problem. Three counterexamples on Euclidean graphs show that it is impossible to obtain an optimal route even when calculating options several steps ahead.

The third example of applying the greedy algorithm to solve the traveling salesman problem is the nearest city method. The method describes the sequential construction of the Hamiltonian cycle. The above version of the algorithm is secured from obtaining non-connected graphs during a solution process. Further, the length of the Hamiltonian cycle is used as an upper bound when implementing the simplest version of the branch and bound method. The program made in Mathcad checks the optimality of the obtained solution.

In the considered examples, the solutions obtained by greedy algorithms are used as an initial approximation for further optimization of the target function.

Keywords: greedy algorithms, transport problem, paradoxical solution of the transport problem, shortest distance problem, traveling salesman problem, traveling salesman problem as a mathematical programming problem, branch and bound method.

References

1. Edmonds J. Matroids and the greedy algorithm. *Math. Programming*. North-Holland Publ. Co. 1971, vol. 1, no. 2, pp. 127–136.
2. Papadimitriou Ch.H., Steiglitz K. *Combinatorial optimization: Algorithms and Complexity*. Dover Publ., 1998, 528 p. (Russ. ed.: Moscow, Mir Publ., 1984, 510 p.).
3. Dasgupta S., Papadimitriou Ch.H., Vazirani U. *Algorithms*. Science Engineering & Math Publ., 2011, 336 p. (Russ. ed.: Shen A. Moscow, MCCME Publ., 2014, 320 p.).
4. Cormen Th.H., Leiserson Ch.I., Rivest R.L., Stein Cl. *Introduction to Algorithms*. 3rd ed., The MIT Press, 2009, 1312 p. (Russ. ed.: Moscow, Vilyams Publ., 2013, 1328 p.).
5. Ochkov V.F., Bogomolova E.P., Ivanov D.A. *Physics and Mathematics with Mathcad and the Internet*. St. Petersburg, Lan Publ., 2016, 388 p.
6. Yudin D.B., Golstein E.G. *Problems and Methods of Linear Programming. Transport Problems*. Moscow, Librokom Publ., 2010, 184 p.
7. Greshilov A.A. *Applied Problems of Mathematical Programming*. 2nd ed., Moscow, Logos Publ., 2006, 288 p.
8. Bellman R.E. *Dynamic Programming*. Princeton Univ. Press, 1957 (Russ. ed.: Moscow, 1960, 400 p.).
9. Kostevich L.S., Lapko A.A. *Operations Research. Game Theory*. 2nd ed., Minsk, Vysh. shk. Publ., 2008, 368 p.
10. Taha H.A. *Operations Research: An Introduction*. 7th ed. Prentice Hall, 2002 (Russ. ed.: Moscow, Villiams Publ., 2005, 912 p.).

Примеры библиографического описания статьи

1. Бойков В.А. О применении жадных алгоритмов в некоторых задачах дискретной математики // Программные продукты и системы. 2019. Т. 32. № 1. С. 55–62. DOI: 10.15827/0236-235X.125.055-062.
2. Boykov V.A. On the application of greedy algorithms in some problems of discrete mathematics. *Software & Systems*. 2019, vol. 32, no. 1, pp. 55–62 (in Russ.). DOI: 10.15827/0236-235X.125.055-062.