

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ В ДИСКРЕТНОЙ МАТЕМАТИКЕ

УДК 519.7

ЗАДАЧА КОММИВОЯЖЁРА: ПРИБЛИЖЁННЫЙ АЛГОРИТМ ПО МЕТОДУ ВЕТВЕЙ И ГРАНИЦ С ГАРАНТИРОВАННОЙ ТОЧНОСТЬЮ¹

Ю. Л. Костюк

*Национальный исследовательский Томский государственный университет,
г. Томск, Россия*

Для решения задачи коммивояжёра с матрицей расстояний порядка n предлагается приближённый алгоритм на основе метода ветвей и границ. Для отсекается используется увеличенная оценка снизу текущего частичного решения, гарантирующая заранее заданную величину ε погрешности всего решения. Проведён вычислительный эксперимент для матриц расстояний четырёх видов распределений, среди них для равномерного случайного (несимметричного) распределения, а также для матриц евклидовых расстояний между случайными точками (симметричного распределения). В последнем случае дополнительно применён локальный поиск. Получены оценки степени p для функции полиномиальной трудоёмкости $O(n^p)$ для разных видов распределений и различных величин погрешности ε . Для равномерного случайного распределения полученная оценка степени p оказалась близка к 2,8 в диапазоне n до 1000 и средней погрешности около 1 %.

Ключевые слова: задача коммивояжёра, метод ветвей и границ, приближённый алгоритм, локальный поиск, вычислительный эксперимент.

DOI 10.17223/20710410/45/12

THE TRAVELING SALESMAN PROBLEM: APPROXIMATE ALGORITHM BY BRANCH-AND-BOUND METHOD WITH GUARANTEED PRECISION

Yu. L. Kostyuk

Tomsk State University, Tomsk, Russia

E-mail: kostyuk_y_l@sibmail.com

To solve the traveling salesman problem with distance matrix of order n , we propose an approximate algorithm based on the branch-and-bound method. For clipping, an increased least estimate of the current partial solution is used. This guarantees a predetermined value ε of the whole solution error. A computational experiment for distance matrices of four kinds of distributions was carried out. A uniform random (asymmetric) distribution as well as matrices of Euclidean distances between random

¹Результаты получены в рамках выполнения госзадания Минобрнауки России, проект № 2.4218.2017/4.6.

points (a symmetric distribution) were used. In the latter case, a local search was additionally applied. Estimates for the power p in the polynomial computational complexity $O(n^p)$ of the algorithm for various kinds of distributions and various values of error ε are obtained. For a uniform random distribution and $n \leq 1000$, the obtained estimate of the power p turned out to be close to 2.8 and of an average value of error to be 1 %.

Keywords: *traveling salesman problem, branch-and-border method, approximate algorithm, local search, computational experiment.*

Введение

В известном алгоритме Литтла [1], реализующим метод ветвей и границ для задачи коммивояжёра (ЗК), на каждом шаге обработки матрицы расстояний вычисляется нижняя оценка стоимости оптимального решения. В [2] предложена модификация алгоритма Литтла путём вычисления более точных нижних оценок. В результате увеличивается эффективность отсека неперспективных маршрутов и тем самым значительно сокращается время вычислений. В [3] предложен способ дополнительного уточнения нижних оценок. Как показал вычислительный эксперимент, алгоритм [3] имеет преимущество перед алгоритмом Литтла как на симметричных, так и на несимметричных матрицах. Тем не менее его трудоёмкость экспоненциальная, поэтому единственным способом решения ЗК большой размерности остаётся поиск приближённого решения. При этом в общем случае приближённое решение с гарантированной точностью можно находить лишь алгоритмами с экспоненциальной трудоёмкостью [4, 5]. Таким образом, поиск более эффективных приближённых алгоритмов с заданной степенью приближения к точному решению представляется актуальным.

В настоящей работе рассмотрен приближённый алгоритм с гарантированной точностью решения, имеющий полиномиальную трудоёмкость в ограниченном диапазоне n , который построен на основе модифицированного алгоритма с уточнением нижней границы [3].

1. Приближённый алгоритм

Алгоритм реализует поиск вглубь на дереве решений, первоначально корень дерева содержит исходную матрицу расстояний M порядка n , а длина текущего маршрута полагается бесконечно большой. В результате обработки матрицы вычисляется нижняя оценка стоимости оптимального маршрута, а в каждой строке и каждом столбце матрицы появляется не менее чем по одному нулю. Затем выбирается один из нулевых элементов матрицы M_{ij} , который определяет переход в маршруте из вершины i в вершину j . Создаётся левый узел в дереве решений, содержащий копию матрицы расстояний без строки i и столбца j , т. е. порядка $n - 1$. Альтернативный вариант маршрута, в котором переход из вершины i в вершину j заведомо отсутствует, сохраняется в исходной матрице M , для этого в ней элемент M_{ij} заменяется на бесконечность.

На следующем шаге алгоритм переходит к обработке левого узла дерева решений, если по матрице расстояний в узле можно построить новый маршрут коммивояжёра меньшей стоимости, чем текущий маршрут. В противном случае выполняется отсечение этих неперспективных вариантов, для чего алгоритм переходит к обработке правого узла, т. е. осуществляет обратный ход. После обработки очередного узла с матрицей расстояний порядка n создаётся новый левый узел с матрицей порядка $n - 1$ и т. д., пока не будет построен новый текущий маршрут коммивояжёра.

Таким образом, на всех шагах работы алгоритма дерево решений представляется списком узлов от корня влево до висячего узла.

Отсечение бесперспективных вариантов выполняется следующим образом. Пусть $S_{\min}$ — стоимость наилучшего на текущий момент построенного маршрута (в начальный момент $S_{\min}$ равно бесконечности), S_m — нижняя оценка для нового частично построенного маршрута в текущий момент, состоящего из отрезков. В методе ветвей и границ проверка $S_m \geq S_{\min}$ отсекает все бесперспективные варианты, которые можно построить на основе частичного маршрута. В предлагаемом приближённом алгоритме проверка $S_m(1 + \varepsilon) \geq S_{\min}$ гарантирует, что наименьший найденный маршрут не будет превышать оптимальный маршрут более чем в $(1 + \varepsilon)$ раз. Чтобы среднюю погрешность сделать на порядок меньше по сравнению с заданной величиной ε , предлагается более гибкая проверка в виде

$$S_m \geq S_{\min}/(1 + \varepsilon(m)),$$

где функция $\varepsilon(m)$ зависит от количества m отрезков частично построенного маршрута, т. е. размера текущей матрицы, и от параметра n_0 следующим образом:

$$\varepsilon(m) = \begin{cases} 0, & \text{если } m \leq n_0, \\ \varepsilon(m - n_0)/(n - n_0), & \text{если } m > n_0. \end{cases}$$

Параметр n_0 имеет следующий смысл: если порядок m текущей матрицы не превышает n_0 , то проверка на отсечение выполняется так же, как в точном алгоритме: $S_m \geq S_{\min}$, а если m больше n_0 , то проверка выполняется с учётом погрешности, увеличивающейся линейно от нуля до ε при возрастании m от n_0 до n .

Общая структура алгоритма аналогична структуре алгоритма с улучшенной нижней оценкой [3].

Как показал вычислительный эксперимент, в некоторых случаях полученный приближённый маршрут можно значительно улучшить, применив к нему локальный поиск [6] с окном фиксированного размера k . Пусть нумерация вершин в маршруте $P_1, P_2, \dots, P_n$. Для всех i от 0 до $n - 1$ последовательно рассматривается группа вершин в окне $P_{i \bmod n+1}, \dots, P_{(i+k-1) \bmod n+1}$, с учётом замыкания маршрута (после вершины P_n следует вершина P_1). Для этой группы вершин выбирается перестановка, минимизирующая маршрут. Так как k является константой, общее время работы алгоритма при этом возрастает на $O(n)$.

Исходный текст модифицированного алгоритма с улучшенной нижней границей и приближённого алгоритма на языке Паскаль (Delphi) доступен в сети интернет [7].

2. Вычислительный эксперимент

В вычислительном эксперименте на модельных данных проверялась сравнительная трудоёмкость следующих четырёх алгоритмов:

- 1) алгоритма Литтла (А-1);
- 2) модифицированного алгоритма с улучшенной нижней границей (А-2);
- 3) приближённого алгоритма с заданной гарантированной точностью решения ε (А-3).
- 4) приближённого алгоритма с заданной величиной ε и с дополнительным локальным поиском (А-4).

Алгоритмы написаны на языке Паскаль в системе Delphi, вычисления производились на компьютере с процессором Athlon (частота 2,9 ГГц). Исходными данными для

всех алгоритмов являлись одни и те же сгенерированные экземпляры матриц расстояний (каждая матрица задаёт свою ЗК).

В табл. 1–8 представлено среднее время вычисления маршрута (в секундах) для следующих четырёх видов распределений элементов матриц:

- 1) случайные матрицы, элементы в них равномерно распределённые независимые случайные целые числа из диапазона от 0 до 1000, матрицы несимметричные;
- 2) случайные матрицы, такие же, как в первом распределении, в которых затем вычислены кратчайшие расстояния алгоритмом Флойда [8] и этими кратчайшими расстояниями заменены исходные элементы матрицы; в результате для расстояний выполняется неравенство треугольника;
- 3) матрицы евклидовых расстояний, в которых для n равномерно распределённых независимых случайных точек на плоскости с координатами из диапазона от 0 до 1000 вычислены расстояния между точками, матрицы симметричные;
- 4) матрицы евклидовых расстояний, такие же, как в первом распределении, в которых для каждой пары расстояний M_{ij} и M_{ji} случайно выбрано одно из них и его длина увеличена в 2 раза, после чего вычислены все кратчайшие расстояния алгоритмом Флойда и этими кратчайшими расстояниями заменены исходные элементы матрицы; полученные матрицы несимметричные, но с выполнением неравенства треугольника.

Распределения выбраны из следующих соображений. Равномерное случайное (первый вид распределения) и евклидовы расстояния (третий вид) — стандартные для ЗК. Для практических применений важен случай, когда расстояния несимметричные, но выполняется неравенство треугольника, как в распределениях 2 и 4. При этом четвёртый вид распределения задаёт почти евклидовы расстояния, когда отношение расстояний M_{ij} и M_{ji} не превышает 2.

Для каждого размера матриц n генерировалось от 400 до 4000 вариантов, чем меньше n , тем больше вариантов. В таблицах отсутствуют те результаты вычислений, для получения которых потребовалось бы слишком много времени.

Т а б л и ц а 1

Результаты эксперимента для матриц первого вида, $n \leq 100$

n	40	50	60	70	80	90	100
A-1	0,077	0,69	5,2	—	—	—	—
A-2	0,010	0,036	0,12	0,40	1,24	3,65	8,1
A-3, $\varepsilon = 0,05$	0,0055	0,0113	0,021	0,043	0,078	0,12	0,18
$\bar{\varepsilon}$	0,05 %	0,08 %	0,11 %	0,14 %	0,16 %	0,21 %	0,21 %
A-3, $\varepsilon = 0,1$	0,0039	0,0058	0,0085	0,013	0,016	0,021	0,025
$\bar{\varepsilon}$	0,18 %	0,31 %	0,46 %	0,55 %	0,61 %	0,75 %	0,75 %
A-3, $\varepsilon = 0,15$	0,0032	0,0046	0,0062	0,0091	0,011	0,015	0,018
$\bar{\varepsilon}$	0,3 %	0,5 %	0,7 %	0,8 %	0,9 %	1,1 %	1,1 %

Т а б л и ц а 2

Результаты эксперимента для матриц первого вида,
 $n \leq 1000$

n	150	200	300	400	500	700	1000
A-3, $\varepsilon = 0,1$	0,074	0,15	0,47	10	—	—	—
A-3, $\varepsilon = 0,15$	0,053	0,11	0,36	0,84	1,6	4,6	18
$\Delta\bar{\varepsilon}$	0,32 %	0,27 %	0,26 %	0,23 %	—	—	—

В табл. 1 и 2 представлены результаты вычислений для матриц первого вида. Здесь параметр n_0 задавался равным 25, это означает, что если некоторый частичный маршрут состоит не более чем из 25 отрезков, то для всех вариантов, строящихся на основе данного частичного маршрута, ищется в точности оптимальный маршрут. Оказалось, что здесь локальный поиск не даёт никакого улучшения маршрута, поэтому в таблицах он не отражён. Для алгоритма А-3 в том случае, когда получено точное решение, приведено среднее значение погрешности $\bar{\varepsilon}$ в процентах. В случае, когда точное решение отсутствует, приведено среднее превышение погрешности $\Delta\bar{\varepsilon}$ по сравнению с наиболее близким к точному решением, полученным для соответствующей размерности n .

Т а б л и ц а 3
Результаты эксперимента для матриц второго вида,
 $n \leq 50$

n	25	30	35	40	45	50
A-1	0,017	0,13	1,2	4,3	21	—
A-2	0,003	0,015	0,076	0,6	1,7	14
A-3, $\varepsilon = 0,05$	0,001	0,005	0,015	0,09	0,2	0,7
$\bar{\varepsilon}$	0,17 %	0,19 %	0,28 %	0,31 %	0,38 %	0,4 %
A-3, $\varepsilon = 0,1$	0,0006	0,0018	0,003	0,0035	0,013	0,035
$\bar{\varepsilon}$	0,4 %	0,43 %	0,6 %	0,7 %	0,8 %	0,85 %
A-3, $\varepsilon = 0,15$	0,0005	0,0011	0,002	0,0028	0,004	0,006
$\bar{\varepsilon}$	0,56 %	0,6 %	0,8 %	0,85 %	1,1 %	1,1 %

Т а б л и ц а 4
Результаты эксперимента для матриц
второго вида, $n \leq 110$

n	60	70	80	90	100	110
A-3, $\varepsilon = 0,05$	1,7	—	—	—	—	—
A-3, $\varepsilon = 0,1$	0,1	3	—	—	—	—
$\Delta\bar{\varepsilon}$	0,5 %	—	—	—	—	—
A-3, $\varepsilon = 0,15$	0,02	0,17	0,5	0,7	1,2	3
$\Delta\bar{\varepsilon}$	0,7 %	0,3 %	—	—	—	—

В табл. 3 и 4 представлены результаты вычислений для матриц второго вида. Здесь параметр n_0 задавался равным 10. Оказалось, что здесь, как и для матриц первого распределения, локальный поиск не даёт никакого улучшения маршрута, поэтому в таблицах он не отражён. Для алгоритма А-3 приведена также погрешность $\bar{\varepsilon}$ или превышение погрешности $\Delta\bar{\varepsilon}$ в процентах.

В табл. 5 и 6 представлены результаты вычислений для матриц третьего вида. Здесь параметр n_0 задавался равным 10. Для алгоритмов А-3 и А-4 приведена также погрешность $\bar{\varepsilon}$ или превышение погрешности $\Delta\bar{\varepsilon}$ в процентах. Размер окна k для локального поиска в алгоритме А-4 задавался равным 11.

В табл. 7 и 8 представлены результаты вычислений для матриц четвёртого вида. Условия проведения эксперимента полностью идентичны предыдущему случаю.

Из таблиц видно, что среднее время работы как точных, так и приближённых алгоритмов зависит, в первую очередь, от вида распределения элементов матриц. При одинаковой размерности для первого вида матриц время работы точных алгоритмов наименьшее, а для третьего — наибольшее, причём разница достигает сотен раз. Ещё больше разница во времени для приближённых алгоритмов.

Т а б л и ц а 5
Результаты эксперимента для матриц
третьего вида, $n \leq 40$

n	25	30	35	40
A-1	0,58	13	–	–
A-2	0,11	0,57	4,3	17
A-3, $\varepsilon = 0,05$ $\bar{\varepsilon}$	0,036 0,5 %	0,14 0,52 %	0,46 0,58 %	2,1 0,62 %
A-4, $\varepsilon = 0,05$ $\bar{\varepsilon}$	0,086 0,06 %	0,19 0,08 %	0,52 0,09 %	2,2 0,18 %
A-3, $\varepsilon = 0,1$ $\bar{\varepsilon}$	0,014 1,8 %	0,04 1,9 %	0,11 2,3 %	0,3 2,4 %
A-4, $\varepsilon = 0,1$ $\bar{\varepsilon}$	0,07 0,16 %	0,09 0,3 %	0,17 0,7 %	0,37 0,8 %
A-3, $\varepsilon = 0,15$ $\bar{\varepsilon}$	0,007 3,2 %	0,017 3,6 %	0,035 4,3 %	0,08 4,5 %
A-4, $\varepsilon = 0,15$ $\bar{\varepsilon}$	0,06 0,3 %	0,07 0,6 %	0,095 1,2 %	0,15 1,4 %

Т а б л и ц а 6
Результаты эксперимента для
матриц третьего вида, $n \leq 60$

n	45	50	55	60
A-3, $\varepsilon = 0,05$	5,7	22	–	–
A-4, $\varepsilon = 0,05$	5,8	22	–	–
A-3, $\varepsilon = 0,1$ $\Delta \bar{\varepsilon}$	0,8 1,9 %	2,3 2,1 %	11 –	– –
A-4, $\varepsilon = 0,1$ $\Delta \bar{\varepsilon}$	0,9 0,7 %	2,4 0,8 %	11 –	– –
A-3, $\varepsilon = 0,15$ $\Delta \bar{\varepsilon}$	0,2 4 %	0,43 4 %	1,2 2,8 %	4 –
A-4, $\varepsilon = 0,15$ $\Delta \bar{\varepsilon}$	0,3 1,6 %	0,51 1,8 %	1,3 1,2 %	4,1 –

Т а б л и ц а 7
Результаты эксперимента для матриц
четвёртого вида, $n \leq 45$

n	30	35	40	45
A-1	4	26	–	–
A-2	0,32	1,5	3,6	15
A-3, $\varepsilon = 0,05$ $\bar{\varepsilon}$	0,05 0,5 %	0,2 0,5 %	0,3 0,55 %	1 0,55 %
A-4, $\varepsilon = 0,05$ $\bar{\varepsilon}$	0,11 0,15 %	0,26 0,2 %	0,36 0,2 %	1,1 0,2 %
A-3, $\varepsilon = 0,1$ $\bar{\varepsilon}$	0,018 1,7 %	0,06 2 %	0,08 2,2 %	0,2 2,2 %
A-4, $\varepsilon = 0,1$ $\bar{\varepsilon}$	0,08 0,6 %	0,13 0,7 %	0,15 0,9 %	0,3 0,9 %
A-3, $\varepsilon = 0,15$ $\bar{\varepsilon}$	0,007 3,2 %	0,02 3,4 %	0,03 3,9 %	0,04 4,1 %
A-4, $\varepsilon = 0,15$ $\bar{\varepsilon}$	0,07 1,1 %	0,09 1,3 %	0,11 1,5 %	0,13 1,8 %

Т а б л и ц а 8
Результаты эксперимента для
матриц четвёртого вида, $n \leq 80$

n	50	60	70	80
A-3, $\varepsilon = 0,05$	3,6	–	–	–
A-4, $\varepsilon = 0,05$	3,7	–	–	–
A-3, $\varepsilon = 0,1$	0,6	3,1	10	–
$\Delta\bar{\varepsilon}$	1,5 %	–	–	–
A-4, $\varepsilon = 0,1$	0,7	3,2	10,1	–
$\Delta\bar{\varepsilon}$	0,9 %	–	–	–
A-3, $\varepsilon = 0,15$	0,12	0,3	1,6	4
$\Delta\bar{\varepsilon}$	3,5 %	2 %	2,6 %	–
A-4, $\varepsilon = 0,15$	0,22	0,4	1,7	4,1
$\Delta\bar{\varepsilon}$	1,7 %	1,2 %	1,3 %	–

Для третьего и четвёртого вида матриц дополнительный локальный поиск уменьшает среднюю погрешность в 2–3 раза при незначительном увеличении времени. В целом, средняя погрешность $\bar{\varepsilon}$ оказывается меньше гарантированной погрешности ε на порядок.

На основе проведённого эксперимента установлено, что средняя трудоёмкость $T(n)$ приближённых алгоритмов в ограниченном диапазоне n имеет полиномиальную зависимость вида $T(n) = a \cdot n^p$. Для получения оценки константы p будем рассматривать отношение трудоёмкости при некотором n_i к трудоёмкости при n_{i-1} для возрастающей последовательности $n_1, n_2, \dots$:

$$\frac{T(n_i)}{T(n_{i-1})} = \left(\frac{n_i}{n_{i-1}} \right)^p. \quad (1)$$

Будем минимизировать сумму квадратов отклонений логарифмов отношений экспериментальных данных времени работы t_i и трудоёмкости (1):

$$\sum_i \left(\ln \frac{t_i}{t_{i-1}} - p \ln \frac{n_i}{n_{i-1}} \right)^2 \rightarrow \min_p. \quad (2)$$

После дифференцирования левой части выражения (2) по параметру p и приравнивания производной к нулю получим формулу для вычисления оценки $\hat{p}$:

$$\hat{p} = \sum_i \ln \frac{t_i}{t_{i-1}} \ln \frac{n_i}{n_{i-1}} / \sum_i \left(\ln \frac{n_i}{n_{i-1}} \right)^2. \quad (3)$$

В табл. 9 приведены полученные по формуле (3) оценки $\hat{p}$ при гарантированной погрешности $\varepsilon = 0,15$ для различных видов матриц. Для третьего и четвёртого видов оценка $\hat{p}$ приведена для приближённого алгоритма с последующей локальной оптимизацией.

Т а б л и ц а 9
Оценка параметра p для различных видов матриц

Вид матриц	1	2	3	4
n	От 40 до 1000	От 25 до 110	От 25 до 60	От 30 до 80
$\hat{p}$	2,8	5,9	4,0	4,2

Заключение

Как показал вычислительный эксперимент, для матриц с равномерно случайными расстояниями предложенный приближённый алгоритм способен находить решение ЗК со средней погрешностью, близкой к 1 %, в диапазоне до $n = 1000$, с трудоёмкостью, растущей как $n^{2,8}$. Для матриц расстояний с другими распределениями, где время работы приближённого алгоритма приемлемо (не более 20 с на обычном компьютере), диапазон n существенно меньше. Однако и в этом случае время работы (по сравнению с временем работы точного алгоритма) уменьшается в десятки и сотни раз при заданной гарантированной погрешности 15 % и средней погрешности, близкой к 2 %.

ЛИТЕРАТУРА

1. *Little J. D. C., Murty K. G., Sweeney D. W., and Karel C.* An algorithm for the Traveling Salesman Problem // *Operat. Res.* 1963. No. 11. P. 972–989.
2. *Костюк Ю. Л.* Эффективная реализация алгоритма решения задачи коммивояжёра методом ветвей и границ // *Прикладная дискретная математика.* 2013. № 2(20). С. 78–90.
3. *Костюк Ю. Л.* Задача коммивояжёра: улучшенная нижняя граница в методе ветвей и границ // *Прикладная дискретная математика.* 2013. № 4(22). С. 73–81.
4. *Меламед И. И., Сергеев С. И., Сигал И. Х.* Задача коммивояжёра. Точные алгоритмы // *Автоматика и телемеханика.* 1989. № 10. С. 3–29.
5. *Меламед И. И., Сергеев С. И., Сигал И. Х.* Задача коммивояжёра. Приближенные алгоритмы // *Автоматика и телемеханика.* 1989. № 11. С. 3–26.
6. *Пападимитриу Х., Стайглиц К.* Комбинаторная оптимизация: Алгоритмы и сложность: пер. с англ. М.: Мир, 1985. 510 с.
7. *Костюк Ю. Л.* Модифицированный алгоритм решения задачи коммивояжёра методом ветвей и границ с улучшенной нижней границей, вычисляющий точное или приближённое решение при заданной гарантированной точности. Программа на языке Паскаль в системе Delphi. 2017 г., апрель. <http://www.inf.tsu.ru/Decanat/Staff.nsf/people/KostjukJuL>
8. *Ахо А., Хопкрофт Дж., Ульман Дж.* Построение и анализ вычислительных алгоритмов: пер. с англ. М.: Мир, 1979. 536 с.

REFERENCES

1. *Little J. D. C., Murty K. G., Sweeney D. W., and Karel C.* An algorithm for the Traveling Salesman Problem. *Operat. Res.*, 1963, no. 11, pp. 972–989.
2. *Kostyuk Yu. L.* Effektivnaya realizaciya algoritma resheniya zadachi kommivoyazhera metodom vetvei i granic [Effective implementation of algorithm for solving the travelling salesman problem by the branch-and-border method]. *Prikladnaya Diskretnaya Matematika*, 2013, no. 2(20), pp. 78–90. (in Russian)
3. *Kostyuk Yu. L.* Zadacha kommivoyazhera: uluchshennaya nizhnyaya granica v metode vetvei i granic [The travelling salesman problem: improved lower bounds in the branch-and-border method]. *Prikladnaya Diskretnaya Matematika*, 2013, no. 4(22), pp. 73–81. (in Russian)
4. *Melamed I. I., Sergeev S. I., and Sigal I. K.* The traveling salesman problem. Exact methods. *Autom. Remote Control*, 1989, vol. 50, no. 10, pp. 1303–1324.
5. *Melamed I. I., Sergeev S. I., and Sigal I. K.* The traveling salesman problem. Approximate algorithm. *Modifitsirovannyi algoritm resheniya zadachi kommivoyazhera metodom vetvey i granits s uluchshennoy nizhney granitse, vychislyayushchiy tochnoe ili priblizhennoe reshenie pri zadannoy garantirovannoy tochnosti.* Programma na yazyke Paskal' v sisteme Delphi. 2017 g., aprel'. *ms. Autom. Remote Control*, 1989, vol. 50, no. 11, pp. 1459–1479.
6. *Papadimitriou C. H. and Steiglitz K.* Combinatorial Optimization: Algorithms and Complexity. N.J., Prentice Hall, 1982.

7. *Kostyuk Yu. L.* A modified algorithm for solving the traveling salesman problem using the branch and bound method with an improved lower bound, which calculates an exact or approximate solution for a given guaranteed accuracy. Pascal program in Delphi system. April, 2017. <http://www.inf.tsu.ru/Decanat/Staff.nsf/people/KostjukJuL>
8. *Aho A. V., Hopcroft J. E., and Ullman J. D.* The Design and Analysis of Computer Algorithms. Reading, Massachusetts, Addison-Wesley Publ., 1976.