**References**

1. **Rusakov S. G.** Modelirovanie nelinejnyh radiochastotnyh shem v sistemah avtomatizacii shemotehnicheskogo proektirovanija (Simulation of nonlinear radio frequency circuits in circuit design automation systems), *Avtomatizacija Proektirovanija,* 1997, no. 2.
2. **Duan X., Mayaram K.** An Efficient and Robust Method for Ring-Oscillator Simulation Using the Harmonic-Balance Method, *IEEE Trans. Computer-Aided Design*, 2005, vol. 24, no. 8, pp. 1225—1233.
3. **Gourary M. M., Rusakov S. G., Ulyanov S. L., Zharov M. M., Gullapalli K. K., Mulvaney B. J.** Simulation of high-Q oscillators, *IEEE/ACM International Conference on Computer-Aided Design,* Nov. 1998, pp. 162—169.
4. **Gourary M. M., Rusakov S. G., Ulyanov S. L., Zharov M. M., Gullapalli, K. K., Mulvaney B. J.** A robust and efficient oscillator analysis technique using harmonic balance, *Computer Methods in Applied Mechanics and Engineering*, Jan. 2000, vol. 181, pp. 451—466.
5. **Docking S., Sachdev M.** An Analytical Equation for the Oscillation Frequency of High-Frequency Ring Oscillators, *IEEE Journal of Solid-State Circuits*, February 2003, vol. 39, N. 2.
6. **Selting P., Zheng Q.** Numerical stability analysys of oscillating integrated circuits, *ELSEVIER J. of Computational and Applied Mathematics*, 1997, vol. 82, pp. 367—378.

УДК 519.854.2 DOI: 10.17587/it.25.590-595

**М. В. Ульянов,** вед. науч. сотр., проф., д-р техн. наук, проф., e-mail: muljanov@mail.ru,
Институт проблем управления им. В. А. Трапезникова РАН,
Московский государственный университет им. М. В. Ломоносова,
**М. И. Фомичёв,** аспирант, преподаватель, e-mail: michan94@yandex.ru,
Нижегородский государственный технический университет им. Р. Е. Алексеева,
Национальный исследовательский университет "Высшая школа экономики"

# Сравнительный анализ комбинаций метода ветвей и границ с метаэвристическими алгоритмами для решения асимметричной задачи коммивояжера[1]

*Алгоритм, реализующий метод ветвей и границ, для решения задачи коммивояжера — один из востребованных точных алгоритмов ее решения. Метаэвристические алгоритмы решения этой задачи не гарантируют получения точного решения, но работают "быстро". Для сокращения числа вершин порожденного дерева решений в методе ветвей и границ можно использовать решение, полученное метаэвристическим алгоритмом. За счет выбора метаэвристического алгоритма и его комбинации с методом ветвей и границ можно получить выигрыш во времени получения точного решения. Такой выбор необходимо подтверждать экспериментальными данными о временной эффективности программной реализации такого комбинированного алгоритма. В данной статье рассматриваются некоторые метаэвристические алгоритмы и комбинация таких алгоритмов с классической реализацией метода ветвей и границ для решения асимметричной задачи коммивояжера. Приводятся данные экспериментального исследования среднего времени получения точного решения для диапазона размерности задачи от 30 до 45 и даются рекомендации по выбору метаэвристического алгоритма.*

***Ключевые слова:*** *задача коммивояжера, метод ветвей и границ, предвычисленный тур, сложность индивидуальной задачи, комбинированные алгоритмы, метаэвристические алгоритмы*

## Введение

В современном мире промедление в секунду или даже долю секунды может стоить миллионы рублей. Заинтересованному лицу важно получить точный ответ на вопрос в кратчайшие сроки. Но, к сожалению, даже при нынешних вычислительных мощностях многие задачи не могут быть решены точно за приемлемое время.

Задача коммивояжера является одной из таких задач. Целый ряд практических постановок в области бизнеса и логистики сводится к классической задаче коммивояжера. Она заключается в поиске гамильтонова цикла с минимальной стоимостью в полном асимметричном графе. Одним из самых известных точных алгоритмов решения задачи коммивояжера является алгоритм метода ветвей и границ, предложенный Дж. Литл, К. Мурти, Д. Суини и К. Кэрол в 1963 г. [1]. Однако данный алгоритм имеет экспоненциальную временную сложность по размерности задачи (числу вершин

[1]Работа выполнена при поддержке гранта РФФИ 18-07-00656.

в графе), и, как результат, ряд прикладных задач большой размерности не может быть точно решен этим методом за время, приемлемое для лица, принимающего решение.

Вместе с тем существует ряд алгоритмов, которые не гарантируют нахождение оптимального решения, но позволяют находить достаточно "хорошие" решения за приемлемое с практической точки зрения время. Такие алгоритмы принято называть "метаэвристическими алгоритмами". В данной статье представлено сравнение типичных представителей различных категорий метаэвристических алгоритмов. Кроме того, наличие метаэвристических алгоритмов решения задачи коммивояжера не означает отказа от возможности получения точных решений этой задачи. Различные идеи, например, использование дополнительной памяти [2], позволяют сократить время решения задачи, но этого явно недостаточно. В данной статье исследуется подход, при котором для сокращения времени решения задачи коммивояжера предлагается сократить поисковое дерево решений с помощью тура коммивояжера (назовем этот тур предвычисленным), полученного до начала работы метода ветвей и границ некоторым метаэвристическим алгоритмом. Отметим, что есть смысл использовать только такой метаэвристический алгоритм, который обеспечит не только существенное сокращение поискового дерева решений, а следовательно, и сокращение времени работы метода ветвей и границ, но и сокращение суммарного времени решения, включающего в себя и время выполнения программной реализации самого метаэвристического алгоритма.

## Задача коммивояжера. Терминология и обозначения

Для однозначности понимания изложения приведем используемую далее терминологию и обозначения, связанные с задачей коммивояжера. *Асимметричная задача коммивояжера* — это задача нахождения гамильтонова цикла минимального веса в полном взвешенном ориентированном графе без собственных петель.

Полный взвешенный ориентированный граф с $n$ вершинами задается взвешенной матрицей смежности $A$ размерности $n$, элементы которой определяют веса дуг между вершинами полного графа. Собственные петли запрещаются специальными значениями на главной диагонали матрицы.

В данной работе будет использоваться терминология, принятая для задачи коммивояжера. Гамильтонов цикл в полном графе называется *туром*, тогда задача коммивояжера заключается в нахождении тура минимального веса. Матрица смежности $A$ называется *матрицей стоимостей*, поскольку в содержательной постановке задачи коммивояжера веса дуг интерпретируются как стоимости проезда между городами. Число вершин графа $n$, определяющее размерность матрицы стоимостей, называется *размерностью задачи коммивояжера*. В статье далее рассматривается асимметричная задача коммивояжера, которая представляет собой наиболее общую и вычислительно трудную постановку задачи.

Очевидно, что начальная вершина тура может быть выбрана произвольно. Таким образом, множество всех возможных туров определяется всеми возможными перестановками остальных вершин и имеет мощность $(n - 1)!$.

Под термином *индивидуальная задача* будем понимать конкретный экземпляр задачи коммивояжера — это задача, заданная конкретной матрицей стоимостей $A$ размерности $n$.

Одним из точных алгоритмов решения задачи коммивояжера является классический алгоритм [1], реализующий метод ветвей и границ. Идея метода ветвей и границ состоит в том, чтобы разделить текущее множество туров на два множества: одно, которое, весьма вероятно, содержит оптимальный тур, и другое, которое, вероятно, этого тура не содержит.

Опираясь на работу Д. Э. Кнута [3], под *сложностью* индивидуальной задачи коммивояжера, заданной матрицей $A$, будем понимать $c(A)$ — число порожденных алгоритмом ветвей и границ вершин поискового дерева решений. Исследование сложности индивидуальной задачи коммивояжера позволяет абстрагироваться от конкретной реализации алгоритма метода ветвей и границ и программно-аппаратных особенностей компьютера, на котором экспериментально исследуются задача коммивояжера и алгоритмы ее решения.

## Постановка задачи

В классическом методе ветвей и границ для решения задачи коммивояжера предполагается, что невозможно начать отсекать поддеревья поискового дерева решений до тех пор, пока мы не найдем какой-либо тур. Ввиду особенности задачи дерево решений может серьезно разрастись, пока первый тур не будет найден. Использование предвычисленного (метаэвристическим алгоритмом) тура (известного до начала работы метода ветвей и границ) может сократить размеры дерева решений, т. е. сложность индивидуальной задачи. Такой комбинированный подход может при хороших значениях предвычисленного тура:

- уменьшить время расчета оптимального тура, потому что отсекается необходимость создавать неперспективные узлы дерева решений и посещать их;
- уменьшить объем выделенной памяти, поскольку пропадает необходимость хранить неперспективные узлы дерева.

Таким образом, данный подход направлен на сокращение и времени расчета и объема требуемой памяти. Однако при его реализации возникают несколько вопросов. Очевидно, что чем ближе стоимость предвычисленного тура к точному решению, тем меньше времени потребуется для нахождения оптимального решения. Вместе с тем, предвычисленный тур должен быть найден достаточно быстро. Это значит, что время, затраченное на работу классического метода ветвей и границ, по крайней мере, должно быть не больше, чем суммарное время работы метаэвристического алгоритма и алгоритма метода ветвей и границ с предвычисленным туром. Другими словами, использование предвычисленного тура должно быть оправданным и рациональным.

На основе этих рассуждений сформулируем постановку задачи экспериментального исследования: определить, оправдано ли использование различных метаэвристических алгоритмов для расчета предвычисленного тура.

## Эвристические алгоритмы получения предвычисленного тура

Эвристические алгоритмы — это алгоритмы, которые не гарантируют нахождение точного решения, однако решения, найденные этими алгоритмами, в определенной мере близки к оптимальным и при этом работают за "приемлемое" время [15]. В отличие от точных алгоритмов эвристические алгоритмы обычно достаточно просты в реализации и работают быстрее. Все множество эвристических алгоритмов можно разделить на три типа:

- жадные [4];
- роевые [5—7];
- улучшающие решения [8, 9].

Кроме того, существует множество различных алгоритмов, которые предназначены для решения частных случаев задачи (например, для решения метрической задачи коммивояжера [10]).

Жадный алгоритм является одним из самых простых в реализации и понимании принципа работы. Он основывается на выборе самого дешевого перехода между городами последовательно для каждого города, начиная с первого и заканчивая $n$-м, запрещая возможные подциклы. Нет никаких сомнений, что этот алгоритм не гарантирует точное решение задачи коммивояжера, более того, тур, полученный с помощью жадного алгоритма, может быть очень далек от оптимального. В то же время он работает достаточно быстро ($O(n^2)$ в худшем случае).

Классическим представителем роевых алгоритмов является муравьиный алгоритм. Муравьиный алгоритм для задачи коммивояжера был предложен коллективом авторов во главе с Марко Дориго в 1991 г. [5]. Он заключается в эмуляции поведения муравьиной колонии. Его точность зависит от следующих параметров:

- время жизни колонии;
- число муравьев;
- начальное расположение муравьев;
- начальное количество феромона;
- объем выделенного феромона;
- коэффициент испарения;
- число элитных муравьев.

С детальным описанием алгоритма можно ознакомиться в работе [11], общая схема работы представлена ниже:

1. Ввод матрицы расстояний $D$.
2. Инициализация параметров алгоритма — $\alpha$, $\beta$, $e$, $Q$.
3. Инициализация ребер — присвоение видимости $\eta_{ij}$ и начальной концентрации феромона.
4. Размещение муравьев в случайно выбранные города без совпадений.
5. Выбор начального кратчайшего маршрута.
6. Цикл по времени жизни колонии $t = \overline{1, t_{\max}}$.

Что касается алгоритмов, улучшающих решения, то в 1973 г. Ш. Лин и Б. Керниган представили эффективный эвристический алгоритм для задачи коммивояжера (алгоритм Лина—Кернигана) [12]. Он основан на идее итерационного улучшения случайно полученного тура. Как показали экспериментальные результаты, этот алгоритм достаточно часто находит даже глобально оптимальные решения. В то же время сложность алгоритма составляет приблизительно $O(n^{2,2})$ [12].

Позднее, в 2000 г. К. Хелсгаун предложил модифицированную реализацию упомянутого алгоритма (алгоритм Лина—Кернигана—Хелсгауна) [13]. Этот алгоритм достаточно часто находит оптимальное решение за приемлемое время даже для задач большой размерности.

Рассмотренные алгоритмы спроектированы для решения симметричной задачи коммивояжера. Однако с использованием метода трансформации, предложенного в работе [14], любая несимметричная задача коммивояжера (размерности $n$) может быть преобразована в симметричную задачу коммивояжера (размерности $2n$). К сожалению, эта трансформация также влияет на время решения несимметричной задачи коммивояжера алгоритмом Лина—Кернигана—Хелсгауна.

Основная идея алгоритма Лина—Кернигана—Хелсгауна заключается в том, чтобы найти некоторое допустимое решение, после чего выделить два множества дуг, таких что если все дуги первого множества удалить из найденного тура и заменить их дугами из второго множества, то результатом будет тур, который лучше (дешевле) предыдущего. Процесс переноса дуг повторяется до тех пор, пока не станет невозможно сформировать такие два множества дуг. Все ограничения на множества дуг и особенности их выбора подробно описаны в работе [13].

Эффективность алгоритма Лина—Кернигана—Хелсгауна достигается, в первую очередь, за счет эффективной стратегии поиска множеств дуг, описанных выше. Поиск построен на основе ограничения перемещений, которые определяются множеством возможных кандидатов [13].

Автором работы [15] предпринята попытка разработать алгоритм, который будет находить оптимальные или очень близкие к оптимальным решения достаточно быстро. Однако при этом не уделено внимание вопросу сложности реализации, в результате чего последняя программная реализация, представленная автором, занимает порядка 10 000 строк исходного кода [15].

## Экспериментальный анализ

Эксперименты проводили на стационарном компьютере со следующими характеристиками:

- процессор: Intel i7 3770K 3800 MHz;
- оперативная память: Kingston KHX1600C9D3P1 16 Гбайт;
- материнская плата: ASRock Fatal1ty Z370 Gaming K6;
- операционная система: GIGABYTE GA-Z77X-D3H.

Для минимизации шумов операционной системы фоновые процессы, которые не нужны для исследования, были отключены, а также отсутствовал графический пользовательский интерфейс, а управление операционной системой осуществлялось посредством командной строки.

Алгоритмы реализованы на языке С++ и скомпилированы в исполняемый файл с помощью компилятора: gcc 4.8.5 20150623.

Для экспериментального анализа были сгенерированы несимметричные задачи коммивояжера для размерностей от 30 до 45. Объем выборки для каждой размерности составил 100 000 индивидуальных задач. Веса дуг асимметричных матриц сгенерированы равномерным случайным генератором случайных чисел в диапазоне [1, 106).

В таблице представлено усредненное по экспериментальной выборке время работы алгоритмов в микросекундах. В таблице приняты следующие обозначения:

- $ARS$ — классический алгоритм, реализующий метод ветвей и границ;
- $ARS_{greedy}$ — алгоритм, реализующий метод ветвей и границ с расчетом предвычисленного тура с помощью жадного алгоритма;
- $ARS_{ant\_k}$ — алгоритм, реализующий метод ветвей и границ с расчетом предвычисленного тура с помощью муравьиного алгоритма, параметр $k$ — время жизни колонии, были исследованы три варианта со временем жизни колонии $2n$, $n$, $n/2$, где $n$ — размерность задачи;
- $ARS_{LKH}$ — алгоритм, реализующий метод ветвей и границ с расчетом предвычисленного тура с помощью алгоритма Лина—Кернигана—Хелсгауна.

На основе полученных экспериментальных результатов можно прийти к заключению, что предвычисленный тур, полученный с помощью жадного алгоритма, не оказывает существенного влияния на время работы классического алгоритма. Использование муравьиного алгоритма в качестве алгоритма для расчета предвычисленного тура является нерациональным выбором для всех исследованных размерностей задачи. С ростом размера задачи алгоритм Лина—Кернигана—Хелсгауна демонстрирует увеличение влияния предвычисленного тура на сокращение сум-

**Результаты экспериментов**

| $n$ | $\bar{t}_{ARS}(n)$ | $\bar{t}_{ARS_{greedy}}(n)$ | $\bar{t}_{ARS_{ant\_2n}}(n)$ | $\bar{t}_{ARS_{ant\_n}}(n)$ | $\bar{t}_{ARS_{ant\_n/2}}(n)$ | $\bar{t}_{ARS_{LKH}}(n)$ |
|---|---|---|---|---|---|---|
| 30 | 36 064 | 35 991 | 103 126 | 69 154 | 52 666 | 47 132 |
| 31 | 46 189 | 45 997 | 122 431 | 83 832 | 64 393 | 57 493 |
| 32 | 58 595 | 58 421 | 145 123 | 101 319 | 79 980 | 71 539 |
| 33 | 75 763 | 75 637 | 173 840 | 124 189 | 99 335 | 89 269 |
| 34 | 95 272 | 95 074 | 205 753 | 150 077 | 122 596 | 109 342 |
| 35 | 122 840 | 122 526 | 246 564 | 184 048 | 152 449 | 137 296 |
| 36 | 160 253 | 159 982 | 298 927 | 228 747 | 194 414 | 174 565 |
| 37 | 200 241 | 199 928 | 356 744 | 276 314 | 237 227 | 214 465 |
| 38 | 268 470 | 286 486 | 448 657 | 353 119 | 310 908 | 281 061 |
| 39 | 331 927 | 331 922 | 532 891 | 425 789 | 377 933 | 345 382 |
| 40 | 426 732 | 426 760 | 650 774 | 530 512 | 478 796 | 437 931 |
| 41 | 549 072 | 548 974 | 797 805 | 663 296 | 604 769 | 556 979 |
| 42 | 682 981 | 682 063 | 958 466 | 807 605 | 745 104 | 687 072 |
| 43 | 862 458 | 861 599 | 1 144 220 | 995 092 | 928 952 | 859 534 |
| 44 | 1 050 941 | 1 049 858 | 1 356 959 | 1 200 666 | 1 124 756 | 1 040 319 |
| 45 | 1 398 876 | 1 398 777 | 1 733 965 | 1 560 646 | 1 478 504 | 1 355 826 |

марного времени точного решения задачи коммивояжера.

## Заключение

Таким образом, на основании полученных экспериментальных результатов можно сделать следующие выводы:

- предвычисленные туры, полученные с помощью жадного алгоритма и муравьиного эвристического алгоритма, не представляют в рамках изучаемой задачи практической ценности, поскольку суммарное время работы метода ветвей и границ и таких эвристических алгоритмов больше, чем время работы "чистого" метода ветвей и границ;
- по полученным экспериментальным данным, начиная с $n = 43$, комбинация алгоритма метода ветвей и границ с алгоритмом Лина—Кернигана—Хелсгауна работает быстрее, чем классический алгоритм метода ветвей и границ для асимметричной задачи коммивояжера, по крайней мере, по оценке в среднем.

Авторы видят дальнейшее развитие исследования в более детальном анализе распределения значений времени работ программных реализаций исследованных алгоритмов.

**Список литературы**


1. **Little J. D. C., Murty K. G., Sweeney D. W., Karel C.** An algorithm for the traveling salesman problem // Operations Research. 1963. Vol. 11. P. 972—989.
2. **Ulyanov M. V., Fomichev M. I.** Resource characteristics of ways to organize a decision tree in the branch-and-bound method for the traveling salesmen problem // Business Informatics. 2015. № 4(34). P. 38—46.
3. **Knuth D. E.** Estimating the efficiency of backtracking programs // Mathematics of Computing. 1975. Vol. 29. P. 121—136.
4. **Applegate D. L., Bixby R. E., Chvatal V., Cook W. J.** The traveling salesman problem: A computational study. Princeton, NJ: Princeton University Press, 2006.
5. **Colorni A., Dorigo M., Maniezzo V.** Distributed optimization by ant colonies // Proceedings of the First European Conference on Artificial Life (ECAL 91). Paris, France, 11—13 December 1991. P. 134—142.
6. **Dorigo M., Gambardella L. M.** Ant colony system: A cooperative learning approach to the traveling salesman problem // IEEE Transactions of Evolutionary Computation. 1997. Vol. 1, N. 1. P. 53—66.
7. **Bonavear E., Dorigo M.** Swarm intelligence: From natural to artificial systems. Oxford, UK: Oxford University Press, 1999.
8. **Gamboa D., Rego C., Glover F.** Data structures and ejection chains for solving large scale traveling salesman problems // European Journal of Operational Research. 2005. Vol. 160, N. 1. P. 154—171.
9. **Kaplan H., Lewenstein M., Shafrir N., Sviridenko M.** Approximation algorithms for asymmetric TSP by decomposing directed regular multigraphs // Journal of the ACM. 2005. Vol. 52, N. 4. P. 602—626.
10. **Mömke T., Svensson O.** Removing and adding edges for the traveling salesman problem // Journal of the ACM. 2016. Vol. 63, N. 1. P. 315—336.

11. **Stutzle T., Hoos H. H.** MAX-MIN ant system // Future Generation Computer Systems. 2000. N. 16. P. 889—914.

12. **Lin S., Kernighan B. W.** An effective heuristic algorithm for the traveling-salesman problem // Operations Research. 1973. Vol. 21, N. 2. P. 498—516.

13. **Helsgaun K.** An effective implementation of the Lin—Kernighan traveling salesman heuristic // European Journal of Operational Research. 2000. Vol. 126, N. 1. P. 106—130.

14. **Jonker R., Volgenant T.** Transforming asymmetric into symmetric traveling salesman problems // Operations Research Letters. 1983. N. 2. P. 161—163.

15. **Helsgaun K.** An extension of the Lin—Kernighan—Helsgaun TSP solver for constrained traveling salesman and vehicle routing problems // Technical Report. Roskilde: Roskilde University, 2017.

**M. V. Ulyanov,** Leading Researcher, Professor, e-mail: muljanov@mail.ru,
V. A. Trapeznikov Institute of Control Sciences of the Russian Academy of Sciences,
M. V. Lomonosov Moscow State University,
**M. I. Fomichev,** Doctoral Student, Nizhny Novgorod State Technical University n.a. R. E. Alekseev,
Lecturer of Faculty of Computer Science, School of Software Engineering National Research University
Higher School of Economics, e-mail: michan94@yandex.ru

# Comparative Analysis of the Branch and Bound Method Combinations with Metaheuristic Algorithms for Solving the Asymmetric Traveling Salesman Problem

*The algorithm that implements the Branch and Bound method for solving the Traveling Salesman Problem is one of the common exact algorithms for solving it. Metaheuristic algorithms for solving this problem do not guarantee obtaining an exact solution, however they work "quickly". In order to reduce the nodes number of the generated decision tree in the Branch and Bo und method, you can use the solution obtained by the metaheuristic algorithm. By choosing a metaheuristic algorithm and its combination with the Branch and Bound method, it is possible to gain time for obtaining an exact solution. Such a choice must be confirmed by experimental data on the time efficiency of the software implementation of such a combined algorithm. This article discusses some metaheuristic algorithms and a combination of such algorithms with the classical implementation of the Branch and Bound method for solving the asymmetric Traveling Salesman Problem. The data of an experimental study of the average time of obtaining an exact solution for the range of the dimension of the problem from 30 to 45 are given, and recommendations are given on the choice of a metaheuristic algorithm.*

***Keywords:*** *asymmetric Traveling Salesman Problem, branch and bound method, precomputed tour, complexity of the individual problem, combined algorithm, metaheuristic algorithm*

DOI: 10.17587/it.25.590-595

**References**

1. **Little J. D. C., Murty K. G., Sweeney D. W., Karel C.** An algorithm for the traveling salesman problem, *Operations Research*, 1963, vol. 11, pp. 972—989.

2. **Ulyanov M. V., Fomichev M. I.** Resource characteristics of ways to organize a decision tree in the branch-and-bound method for the traveling salesmen problem, *Business Informatics*, 2015, no. 4(34), pp. 38—46.

3. **Knuth D. E.** Estimating the efficiency of backtracking programs, *Mathematics of Computing*, 1975, vol. 29, pp. 121—136.

4. **Applegate D. L., Bixby R. E., Chvatal V., Cook W. J.** The traveling salesman problem: A computational study, Princeton, NJ, Princeton University Press, 2006.

5. **Colorni A., Dorigo M., Maniezzo V.** Distributed optimization by ant colonies, *Proceedings of the First European Conference on Artificial Life (ECAL 91),* Paris, France, 11—13 December 1991, pp. 134—142.

6. **Dorigo M., Gambardella L. M.** Ant colony system: A cooperative learning approach to the traveling salesman problem, *IEEE Transactions of Evolutionary Computation*, 1997, vol. 1, no. 1, pp. 53—66.

7. **Bonavear E., Dorigo M.** Swarm intelligence: From natural to artificial systems, Oxford, UK, Oxford University Press, 1999.

8. **Gamboa D., Rego C., Glover F.** Data structures and ejection chains for solving large scale traveling salesman problems, *European Journal of Operational Research*, 2005, vol. 160, no. 1, pp. 154—171.

9. **Kaplan H., Lewenstein M., Shafrir N., Sviridenko M.** Approximation algorithms for asymmetric TSP by decomposing directed regular multigraphs, *Journal of the ACM*, 2005, vol. 52, no. 4, pp. 602—626.

10. **Mömke T., Svensson O.** Removing and adding edges for the traveling salesman problem, *Journal of the ACM*, 2016, vol. 63, no. 1, pp. 315—336.

11. **Stutzle T., Hoos H. H.** MAX-MIN ant system, *Future Generation Computer Systems*, 2000, no. 16, pp. 889—914.

12. **Lin S., Kernighan B. W.** An effective heuristic algorithm for the traveling-salesman problem, *Operations Research*, 1973, vol. 21, no. 2, pp. 498—516.

13. **Helsgaun K.** An effective implementation of the Lin—Kernighan traveling salesman heuristic, *European Journal of Operational Research*, 2000, vol. 126, no. 1, pp. 106—130.

14. **Jonker R., Volgenant T.** Transforming asymmetric into symmetric traveling salesman problems, *Operations Research Letters*, 1983, no. 2, pp. 161—163.

15. **Helsgaun K.** An extension of the Lin—Kernighan—Helsgaun TSP solver for constrained traveling salesman and vehicle routing problems. Technical Report, Roskilde, Roskilde University, 2017.