

Backtracking Algorithms for Constructing the Hamiltonian Decomposition of a 4-regular Multigraph

A. V. Korostil¹, A. V. Nikolaev¹

DOI: [10.18255/1818-1015-2021-1-6-21](https://doi.org/10.18255/1818-1015-2021-1-6-21)

¹P. G. Demidov Yaroslavl State University, 14 Sovetskaya, Yaroslavl 150003, Russia.

MSC2020: 05C85, 52B05

Research article

Full text in Russian

Received February 15, 2021

After revision March 10, 2021

Accepted March 12, 2021

We consider a Hamiltonian decomposition problem of partitioning a regular graph into edge-disjoint Hamiltonian cycles. It is known that verifying vertex non-adjacency in the 1-skeleton of the symmetric and asymmetric traveling salesperson polytopes is an NP-complete problem. On the other hand, a sufficient condition for two vertices to be non-adjacent can be formulated as a combinatorial problem of finding a Hamiltonian decomposition of a 4-regular multigraph. We present two backtracking algorithms for verifying vertex non-adjacency in the 1-skeleton of the traveling salesperson polytope and constructing a Hamiltonian decomposition: an algorithm based on a simple path extension and an algorithm based on the chain edge fixing procedure.

Based on the results of the computational experiments for undirected multigraphs, both backtracking algorithms lost to the known heuristic general variable neighborhood search algorithm. However, for directed multigraphs, the algorithm based on chain fixing of edges showed comparable results with heuristics on instances with existing solutions, and better results on instances of the problem where the Hamiltonian decomposition does not exist.

Keywords: Hamiltonian decomposition, traveling salesperson polytope, 1-skeleton, vertex adjacency, backtracking

INFORMATION ABOUT THE AUTHORS

Alexander V. Korostil | orcid.org/0000-0003-1881-0207. E-mail: av.korostil@gmail.com
postgraduate student.

Andrei V. Nikolaev | orcid.org/0000-0003-4705-2409. E-mail: andrei.v.nikolaev@gmail.com
correspondence author | PhD, associate professor.

Funding: This work was supported by P. G. Demidov Yaroslavl State University Project № VIP-016.

For citation: A. V. Korostil and A. V. Nikolaev, "Backtracking Algorithms for Constructing the Hamiltonian Decomposition of a 4-regular Multigraph", *Modeling and analysis of information systems*, vol. 28, no. 1, pp. 6-21, 2021.

Алгоритмы поиска с возвратом для построения гамильтонова разложения 4-регулярного мультиграфа

А. В. Коростиль¹, А. В. Николаев¹

DOI: [10.18255/1818-1015-2021-1-6-21](https://doi.org/10.18255/1818-1015-2021-1-6-21)

¹Ярославский государственный университет им. П. Г. Демидова, ул. Советская, 14, г. Ярославль, 150003 Россия.

УДК 519.16, 004.021, 514.172.45

Научная статья

Полный текст на русском языке

Получена 15 февраля 2021 г.

После доработки 10 марта 2021 г.

Принята к публикации 12 марта 2021 г.

Рассматривается задача построения гамильтонова разложения регулярного мультиграфа на гамильтоновы циклы без общих рёбер. Известно, что проверка несмежности вершин в полиэдральных графах симметричного и асимметричного многогранников коммивояжёра является NP-полной задачей. С другой стороны, достаточное условие несмежности вершин можно сформулировать в виде комбинаторной задачи построения гамильтонова разложения 4-регулярного мультиграфа. В статье представлены два алгоритма поиска с возвратом для проверки несмежности вершин в полиэдральном графе коммивояжёра и построения гамильтонова разложения 4-регулярного мультиграфа: алгоритм на основе последовательного расширения простого пути и алгоритм на основе процедуры цепного фиксирования рёбер.

По результатам вычислительных экспериментов для неориентированных мультиграфов оба переборных алгоритма проиграли известному эвристическому алгоритму поиска с переменными окрестностями. Однако для ориентированных мультиграфов алгоритм на основе цепного фиксирования рёбер показал сопоставимые результаты с эвристиками на экземплярах задачи, имеющих решение, и лучшие результаты на экземплярах задачи, для которых гамильтонова разложения не существует.

Ключевые слова: Гамильтоново разложение, многогранник коммивояжёра, полиэдральный граф, смежность вершин, поиск с возвратом

ИНФОРМАЦИЯ ОБ АВТОРАХ

Александр Васильевич Коростиль | orcid.org/0000-0003-1881-0207. E-mail: av.korostil@gmail.com
аспирант.

Андрей Валерьевич Николаев | orcid.org/0000-0003-4705-2409. E-mail: andrei.v.nikolaev@gmail.com
автор для корреспонденции | канд. физ.-мат. наук, доцент.

Финансирование: Работа выполнена в рамках инициативной НИР ЯрГУ им. П. Г. Демидова № VIP-016.

Для цитирования: A. V. Korostil and A. V. Nikolaev, "Backtracking Algorithms for Constructing the Hamiltonian Decomposition of a 4-regular Multigraph", *Modeling and analysis of information systems*, vol. 28, no. 1, pp. 6-21, 2021.

Введение

Гамильтоновым разложением регулярного мультиграфа называется разбиение множества его рёбер на гамильтоновы циклы. Задача поиска в заданном регулярном графе гамильтоновых циклов без общих рёбер находит применение в комбинаторной оптимизации [1], теории кодирования [2, 3], алгоритмах распределенного интеллектуального анализа данных [4], анализе взаимосвязанных сетей [5] и других областях. См. также теоретические результаты по оценке числа гамильтоновых разложений регулярных графов [6]. В данной работе задача построения гамильтонова разложения возникает в области полиэдральной комбинаторики.

1. Многогранник коммивояжёра

Рассматривается классическая постановка задачи коммивояжёра: задан полный взвешенный граф (или орграф) $K_n = (V, E)$, найти гамильтонов цикл минимального веса. Обозначим через HC_n множество всех гамильтоновых циклов в графе K_n и сопоставим каждому гамильтонову циклу $x \in HC_n$ характеристический вектор $x^v \in \mathbb{R}^E$ по следующему правилу:

$$x_e^v = \begin{cases} 1, & \text{если цикл } x \text{ содержит ребро } e, \\ 0, & \text{в противном случае.} \end{cases}$$

Многогранник

$$STSP(n) = \text{conv}\{x^v \mid x \in HC_n\}$$

называется *многогранником симметричной задачи коммивояжёра*.

Многогранник асимметричной задачи коммивояжёра $ATSP(n)$ определяется аналогично как выпуклая оболочка характеристических векторов всех возможных гамильтоновых циклов в полном орграфе K_n .

Подход к решению задачи коммивояжёра с помощью целочисленного линейного программирования был впервые представлен в классической работе Данцига, Фалкерсона и Джонсона для 49 городов США [7]. Лучшие на данный момент точные алгоритмы для решения задачи коммивояжёра основаны на частичном описании фасет многогранника коммивояжёра и методе ветвей и отсечений для целочисленного линейного программирования [8].

Полиэдральным графом многогранника называется граф, вершинами которого являются вершины многогранника, а рёбрами – геометрические рёбра, т.е. одномерные грани. Исследование полиэдральных графов представляет интерес, так как, с одной стороны, некоторые комбинаторные алгоритмы для таких задач как совершенное паросочетание, покрытие множества, независимое множество, ранжирование объектов, задачи с нечёткими мерами и ряд других основаны на отношении смежности вершин в полиэдральном графе и технике локального поиска (когда от текущего решения переход осуществляется к «лучшему» решению среди смежных) [9–13]. С другой стороны, различные характеристики полиэдрального графа задачи, такие как диаметр и кликовое число (число вершин в наибольшей клике), служат оценками сложности для различных моделей вычислений и классов алгоритмов [14–16].

К сожалению, на пути исследования полиэдрального графа многогранника коммивояжёра встаёт классический результат Пападимитриу.

Теорема 1 (Пападимитриу [17]). *Задача проверки несмежности вершин в многогранниках $STSP(n)$ симметричной и $ATSP(n)$ асимметричной задач коммивояжёра является NP-полной.*

Отметим, что дополнительная задача проверки смежности вершин в полиэдральном графе многогранника коммивояжёра будет co-NP-полной.

2. Гамильтоново разложение и достаточное условие несмежности вершин

В связи с результатом Пападимитриу о NP-полноте проверки несмежности вершин в полиэдральном графе многогранника коммивояжёра особый интерес представляют различные достаточные условия несмежности. В частности, известны полиномиально разрешимые достаточные условия для пирамидальных циклов [18], пирамидальных циклов с шагами назад [19] и «родословных» [20, 21]. В данной работе рассматривается наиболее общее из известных – достаточное условие Рао.

Пусть $x = (V, E_x)$ и $y = (V, E_y)$ – два гамильтоновых цикла на множестве вершин V . Обозначим через $x \cup y$ 4-регулярный неориентированный (2-регулярный ориентированный) мультиграф $(V, E_x \cup E_y)$, который содержит копию каждого ребра x и y . Отметим, что если два цикла содержат одно и то же ребро e , то в мультиграф $x \cup y$ добавляются обе копии ребра.

Лемма 1 (Рао [22]). *Пусть x, y – два гамильтоновых цикла. Если мультиграф $x \cup y$ содержит два гамильтоновых цикла z и w без общих рёбер, отличных от x и y , то соответствующие вершины x^v и y^v многогранника коммивояжёра STSP(n) (или ATSP(n)) несмежны.*

С геометрической точки зрения достаточное условие Рао означает, что отрезок, соединяющий вершины x^v и y^v , пересекается с отрезком, соединяющим две другие вершины z^v и w^v многогранника коммивояжёра, следовательно, они не могут быть смежны. Пример выполненного достаточного условия приведён на Рис. 1.

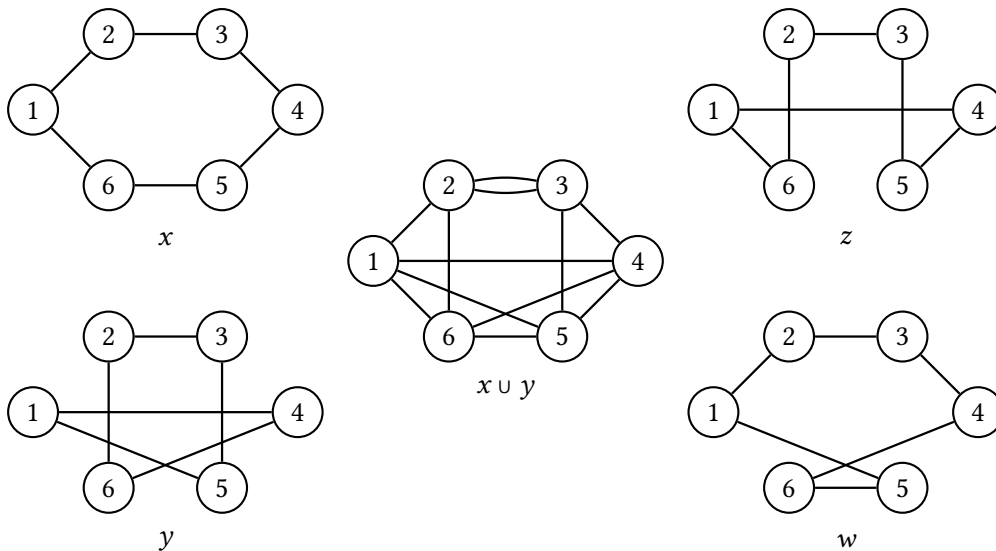


Fig. 1. An example of a satisfied sufficient condition for nonadjacency

Рис. 1. Пример выполненного достаточного условия несмежности

Таким образом, проверка достаточного условия несмежности вершин в полиэдральном графе многогранника коммивояжёра сводится к поиску двух различных гамильтоновых разложений в 4-регулярном мультиграфе $x \cup y$. Сформулируем достаточное условие в виде комбинаторной задачи.

Условие. Заданы два гамильтоновых цикла x и y .

Задача. Найдутся ли в мультиграфе $x \cup y$ два гамильтоновых цикла z и w без общих рёбер, отличные от x и y ?

Отметим, что проверка содержит ли произвольный граф гамильтоново разложение является NP-полной задачей уже для 4-регулярных неориентированных мультиграфов и 2-регулярных ориентированных мультиграфов [23].

Ранее задача проверки несмежности вершин в полиэдральном графе коммивояжёра и построения гамильтонова разложения 4-регулярного мультиграфа исследовалась в работах [24, 25], где был предложен ряд эвристических алгоритмов на основе построения покрытия графа циклами без общих вершин: имитация отжига и поиск с переменными окрестностями. Эвристические алгоритмы оказались очень эффективными на экземплярах задачи имеющих решение, особенно на неориентированных графах. Однако на экземплярах задачи не имеющих решения эвристики сталкиваются со значительными затруднениями. В данной работе рассматриваются два точных переборных алгоритма поиска с возвратом построения гамильтонова разложения 4-регулярного мультиграфа.

3. Алгоритм поиск с возвратом на основе построения простого пути

Напомним, что *поиск с возвратом (backtracking)* – один из общих методов нахождения решения задачи полным перебором. Процедура заключается в последовательном расширении частичного решения. Если на очередном шаге такое расширение провести не удастся, то возвращаются к более короткому частичному решению и продолжают поиск дальше [26].

В работе [27] был представлен алгоритм поиска с возвратом для задачи построения гамильтонова разложения 4-регулярного мультиграфа на основе построения простого пути. Ниже приводится его модифицированная версия.

Пусть здесь и далее частичное решение состоит из двух компонент z и w . Идея алгоритма заключается в последовательном построении простого пути в компоненте z . При этом рёбра, не попавшие в z , отправляются в компоненту w .

Рассмотрим пример мультиграфа $x \cup y$, приведённый на Рис. 1. Построим для него частичное решение, соответствующее простому пути $2 - 3 - 5 - 6$ (Рис. 2). Здесь сплошными выделены рёбра компоненты z , пунктирными – рёбра компоненты w . Так как степень каждой вершины в мультиграфе $x \cup y$ равна 4, то продолжить простой путь в z из вершины 6 можно не более чем тремя способами: $(6, 1)$, $(6, 2)$ и $(6, 4)$. При этом какое бы ребро мы не выбрали для продолжения пути, так как уже два ребра инцидентных вершине 6 добавлены в z , то два других ребра могут попасть только в компоненту w .

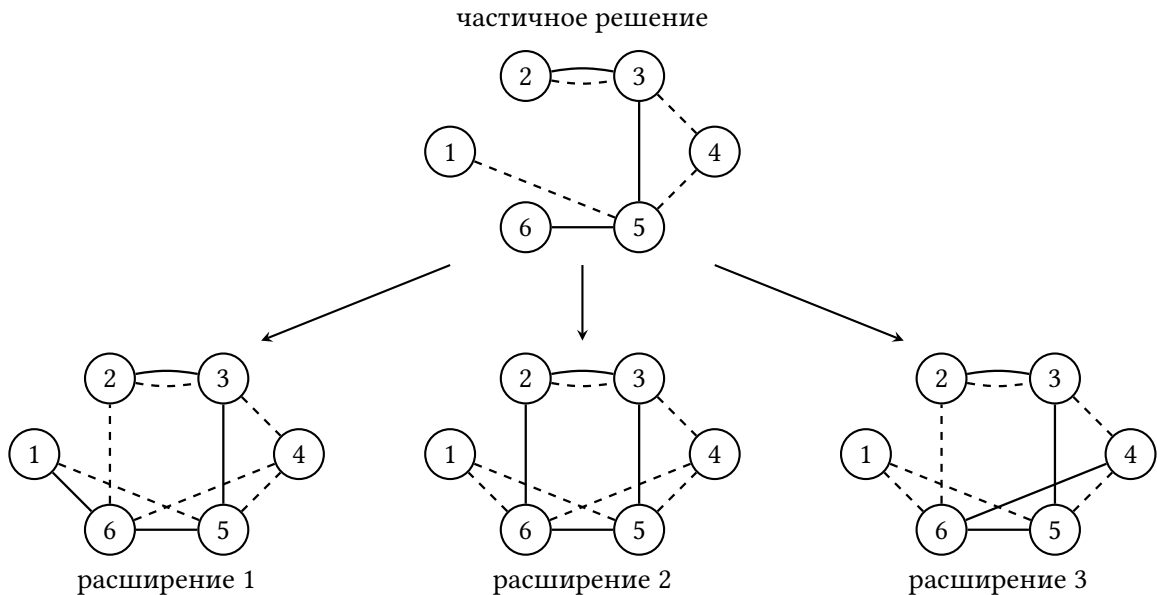


Fig. 2. Backtracking based on simple path expansion

Рис. 2. Поиск с возвратом на основе расширения простого пути

Мы последовательно рассматриваем все три варианта и для каждого проверяем корректность частичного решения:

- расширение 1 недопустимо, так как содержит цикл на вершинах 2, 3, 4, 6 в компоненте w ;
- расширение 2 недопустимо, так как содержит цикл на вершинах 2, 3, 5, 6 в компоненте z ;
- расширение 3 является корректным.

В общем случае условия корректности частичного решения имеют следующий вид:

- в z содержится простой путь (по построению) или гамильтонов цикл;
- в w содержится лес (граф без циклов) со степенью вершин не более 2 или гамильтонов цикл.

Если вариант расширения частичного решения оказался некорректным, то мы откатываемся на шаг назад и рассматриваем следующий вариант. Если все три варианта расширения недопустимы, то данное частичное решение невозможно расширить и мы также возвращаемся на шаг назад.

Общая схема поиска с возвратом для неориентированных графов приведена в псевдокоде Алгоритма 1.

Алгоритм 1 Поиск с возвратом на основе построения простого пути

```

1: procedure Backtracking_Simple_Path( $z, w, (i, j), x \cup y$ )
2:   Добавить ребро  $(i, j)$  в  $z$ 
3:   if Вершине  $i$  в  $z$  инцидентны 2 ребра then
4:     | Добавить свободные ребра инцидентные  $i$  в  $w$ 
5:   end if
6:   if частичное решение  $z, w$  не является корректным then
7:     | return ▷ Вернуться на шаг назад
8:   end if
9:   if  $z$  и  $w$  гамильтоновы циклы (отличные от  $x$  и  $y$ ) then
10:    | return гамильтоново разложение  $z$  и  $w$  ▷ Решение найдено
11:  end if
12:  Отсортировать рёбра  $(j, k)$  из вершины  $j$  в порядке возрастания свободных степеней  $k$ 
13:  for каждого свободного ребра  $(j, k)$  из вершины  $j$  do
14:    |  $z, w \leftarrow$  Backtracking_Simple_Path( $z, w, (j, k), x \cup y$ )
15:    | if гамильтоново разложение найдено then
16:      | | return гамильтоново разложение  $z$  и  $w$ 
17:    | end if
18:  end for
19: end procedure

20: procedure Algorithm_Simple_Path( $x \cup y$ )
21:    $z, w \leftarrow \emptyset$ 
22:   Выбрать начальное ребро  $(i, j)$  мультиграфа  $x \cup y$ 
23:    $z, w \leftarrow$  Backtracking_Simple_Path( $z, w, (i, j), x \cup y$ )
24:   if  $z$  и  $w$  найдены then
25:     | return Найдено гамильтоново разложение  $z$  и  $w$  (вершины многогранника несмежны)
26:   end if
27:   return Гамильтоново разложение не найдено (вершины возможно смежны)
28: end procedure

```

Для ориентированных графов алгоритм отличается только тем, что из каждой вершины мультиграфа $x \cup y$ выходит ровно два ребра, следовательно, существует не более двух вариантов расширения простого пути в z . Все остальные шаги полностью аналогичны.

4. Алгоритм поиска с возвратом на основе цепного фиксирования рёбер

Второй алгоритм поиска с возвратом основан на процедуре цепного фиксирования рёбер мультиграфа в компонентах z и w . Алгоритмы для ориентированных и неориентированных графов несколько отличаются, поэтому опишем их по отдельности.

4.1. Ориентированные мультиграфы

Рассмотрим ориентированный 2-регулярный мультиграф $x \cup y$, для которого полустепени захода и исхода каждой вершины равны двум. Выберем некоторое ребро (i, j) и зафиксируем его в компоненте z , тогда второе ребро (i, k) , выходящее из i , и второе ребро (h, j) , входящее в j , очевидно не могут попасть в z . Мы зафиксируем эти рёбра в w (Рис. 3).

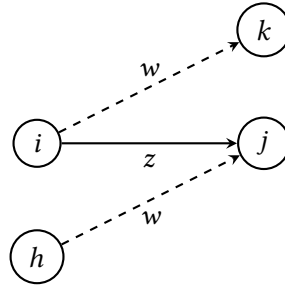


Fig. 3. Fixing the edge (i, j) to z

Рис. 3. Фиксирование ребра (i, j) в z

Основная идея алгоритма заключается в том, что рёбра (i, k) и (h, j) , зафиксированные в w , в свою очередь, запускают рекурсивные цепочки фиксирования рёбер в z и т. д.

Рассмотрим в качестве примера ориентированный 2-регулярный мультиграф, приведённый на Рис. 4. Отметим, что мультиграф содержит две копии ребра $(2, 3)$. Одинаковые ребра не могут попасть в один гамильтонов цикл, поэтому зафиксируем по одной копии в z и w . Выберем некоторое ребро, например $(1, 2)$, и зафиксируем его в компоненте z , тогда:

- 1) ребро $(1, 2)$ зафиксировано в z , следовательно, рёбра $(1, 4)$ и $(6, 2)$ отправляются в w ;
- 2) ребро $(1, 4)$ зафиксировано в w , следовательно, ребро $(3, 4)$ отправляется в z , ребро $(6, 2)$ зафиксировано в w , следовательно, ребро $(6, 1)$ отправляется в z ;
- 3) ребро $(3, 4)$ зафиксировано в z , следовательно, ребро $(3, 5)$ отправляется в w , ребро $(6, 1)$ зафиксировано в z , следовательно, ребро $(5, 1)$ отправляется в w ;
- 4) ребро $(3, 5)$ зафиксировано в w , следовательно, ребро $(4, 5)$ отправляется в z , ребро $(5, 1)$ зафиксировано в w , следовательно, ребро $(5, 6)$ отправляется в z .

На выходе только по одному зафиксированному в z ребру $(1, 2)$ мы получаем однозначно гамильтоновы циклы z и w (Рис. 4). Учитывая, что ребро $(1, 2)$ должно принадлежать хотя бы одному гамильтонову циклу, то данный 2-регулярный ориентированный мультиграф содержит единственное гамильтоново разложение, которое и было найдено.

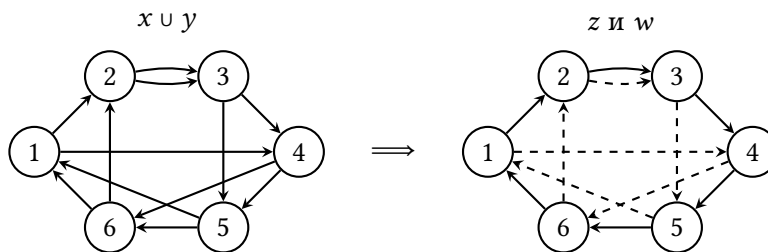


Fig. 4. The result of fixing the edge $(1, 2)$ in z

Рис. 4. Результат фиксирования ребра $(1, 2)$ в z

Общая схема поиска с возвратом для ориентированных мультиграфов приведена в псевдокоде Алгоритма 2.

Алгоритм 2 Поиск с возвратом с цепным фиксированием рёбер для ориентированных графов

```

1: procedure Chain_Edge_Fixing_Directed( $(i, j)$  в  $z$ )                                ▷ Рекурсивное фиксирование рёбер
2:   Зафиксировать ребро  $(i, j)$  в  $z$ 
3:   if ребро  $(i, k)$  не зафиксировано then
4:     | Chain_Edge_Fixing_Directed( $(i, k)$  в  $w$ )
5:   end if
6:   if ребро  $(h, j)$  не зафиксировано then
7:     | Chain_Edge_Fixing_Directed( $(h, j)$  в  $w$ )
8:   end if
9: end procedure

10: procedure Backtracking_Chain_Edge_Fixing_Directed( $z, w, (i, j), x \cup y$ )
11:   Chain_Edge_Fixing_Directed( $(i, j)$  в  $z$ )
12:   if  $z$  и  $w$  – гамильтоновы циклы (отличные от  $x$  и  $y$ ) then
13:     | return  $z$  и  $w$  – решение
14:   end if
15:   if  $z$  или  $w$  содержит не гамильтонов цикл then                                ▷ Частичное решение некорректно
16:     | return                                                                    ▷ Вернуться на шаг назад
17:   end if
18:   Выбрать вершину  $i'$ , из которой выходят свободные рёбра
19:   for каждого свободного ребра  $(i', j')$  из вершины  $i'$  do
20:     |  $z, w \leftarrow$  Backtracking_Chain_Edge_Fixing_Directed( $z, w, (i', j'), x \cup y$ )
21:     | if гамильтоново разложение найдено then
22:       | return гамильтоново разложение  $z$  и  $w$ 
23:     | end if
24:   end for
25: end procedure

26: procedure Algorithm_Chain_Edge_Fixing_Directed( $x \cup y$ )
27:    $z, w \leftarrow \emptyset$ 
28:   Найти совпадающие рёбра в  $x \cup y$  и зафиксировать в  $z$  и  $w$  по одной копии
29:   Выбрать свободное ребро  $(i, j)$  мультиграфа  $x \cup y$ 
30:    $z, w \leftarrow$  Backtracking_Chain_Edge_Fixing_Directed( $z, w, (i, j), x \cup y$ )
31:   if  $z$  и  $w$  найдены then
32:     | return Найдено гамильтоново разложение  $z$  и  $w$  (вершины многогранника несмежны)
33:   end if
34:   return Гамильтоново разложение не найдено (вершины возможно смежны)
35: end procedure

```

На шаге предварительной обработки данных (строка 28, Алгоритм 2) мы находим все совпадающие рёбра в мультиграфе $x \cup y$ и фиксируем в z и w по одной копии, так как эти рёбра не могут попасть в один гамильтонов цикл.

Частичное решение считается корректным, если компоненты z и w представляют собой ориентированные ациклические графы с полустепенями захода и исхода не более одного или ориентированные гамильтоновы циклы (контур). По построению, полуступени вершин в z и w не

могут быть равны двум, так что для проверки корректности достаточно убедиться в отсутствии не гамильтоновых циклов.

В отличие от алгоритма поиска с возвратом на основе построения простого пути (Алгоритм 1) вершину i' для построения расширения частичного решения (строка 18, Алгоритм 2) имеет смысл выбирать так, чтобы цепочки фиксирования запускались с обоих концов ребра (i', j') . Чем больше рёбер будет зафиксировано на одном шаге, тем меньше окажется глубина рекурсии.

Отметим, что мы не рассматриваем различные варианты при выборе и фиксировании первого свободного ребра (i, j) (строка 29, Алгоритм 2), так как это ребро должно попасть в одну из компонент решения. Назовём компонентой z ту, что содержит ребро (i, j) .

Следует также отметить, что хотя процедура Chain_Edge_Fixing_Directed (строки 1-9, Алгоритм 2) рекурсивного фиксирования рёбер на каждом шаге вызывает до двух своих подзадач, общая трудоёмкость линейна ($O(V)$), так как каждое ребро может быть зафиксировано не более одного раза, и $|E| = 2|V|$.

4.2. Неориентированные мультиграфы

Псевдокод алгоритма для неориентированных мультиграфов приведён в Алгоритме 3.

Алгоритм 3 Поиск с возвратом с цепным фиксированием рёбер для неориентированных графов

```

1: procedure Chain_Edge_Fixing_Undirected( $(i, j)$  в  $z$ )
2:   Зафиксировать ребро  $(i, j)$  в  $z$ 
3:   if в  $z$  вершине  $i$  инцидентны 2 зафиксированных ребра then
4:     | Chain_Edge_Fixing_Undirected( $(i, k)$  в  $w$ )           ▷ Зафиксировать два других ребра в  $w$ 
5:     | Chain_Edge_Fixing_Undirected( $(i, h)$  в  $w$ )
6:   end if
7:   if в  $z$  вершине  $j$  инцидентны 2 зафиксированных ребра then
8:     | Chain_Edge_Fixing_Undirected( $(j, k)$  в  $w$ )           ▷ Зафиксировать два других ребра в  $w$ 
9:     | Chain_Edge_Fixing_Undirected( $(j, h)$  в  $w$ )
10:  end if
11: end procedure

12: procedure Backtracking_Chain_Edge_Fixing_Undirected( $z, w, (i, j), x \cup y$ )
13:  Chain_Edge_Fixing_Undirected( $(i, j)$  в  $z$ )
14:  if  $z$  и  $w$  – гамильтоновы циклы (отличные от  $x$  и  $y$ ) then
15:    | return  $z$  и  $w$  – решение
16:  end if
17:  if  $z$  или  $w$  содержит не гамильтонов цикл then           ▷ Частичное решение некорректно
18:    | return                                                   ▷ Вернуться на шаг назад
19:  end if
20:  Выбрать вершину  $i$  с минимальной степенью  $d$  свободных рёбер
21:  Отсортировать рёбра  $(i, j_k)$  из вершины  $i$  в порядке возрастания свободных степеней  $j_k$ 
22:  for  $k \leftarrow 1$  to  $d$  do
23:    |  $z, w \leftarrow$  Backtracking_Chain_Edge_Fixing_Undirected( $z, w, (i, j_k), x \cup y$ )
24:    | if гамильтоново разложение найдено then
25:      | | return гамильтоново разложение  $z$  и  $w$ 
26:    | end if
27:  end for
28: end procedure

```

Процедура фиксирования рёбер срабатывает, как только в одном из частичных решений вершине инцидентны 2 ребра, тогда два других ребра отправляется в другое частичное решение (строки 3-10, Алгоритм 3).

Частичное решение считается корректным, если компоненты z и w представляют собой леса со степенью вершин не более двух или гамильтоновы циклы. По построению, мы добавляем в z и w по одному ребру за раз. Причём, как только степень какой-то вершины в одной из компонент становится равной двум, то два оставшихся ребра отправляются в другую компоненту. Таким образом, для проверки корректности частичного решения достаточно убедиться, что z и w не содержат не гамильтоновых циклов.

Отметим, что на каждом шаге для расширения частичного решения мы выбираем вершину i с минимальной степенью по свободным рёбрам с целью уменьшить ветвление рекурсии. Выходящие из i рёбра рассматриваются в порядке возрастания степеней смежных вершин по свободным рёбрам (строки 20-27, Алгоритм 3).

4.3. Смежность вершин и предварительная обработка данных

Отметим, что рассматриваемые в данной статье алгоритмы разработаны непосредственно для задачи построения гамильтонова разложения 4-регулярного мультиграфа $x \cup y$. Алгоритмы обращаются напрямую к циклам x и y только при проверке, что найденное разложение z и w отлично от исходного. Если опустить эту проверку, то мы получим алгоритмы для построения гамильтонова разложения без какой-либо привязки к полиэдральной комбинаторике. В противном случае, если нас интересует именно смежность вершин в полиэдральном графе многогранника коммивояжера, то рассматриваемые алгоритмы имеет смысл усилить, добавив шаг предварительной обработки данных, на котором будут проверены известные полиномиальные достаточные условия несмежности вершин: пирамидальные циклы [18], пирамидальные циклы с шагами назад [19], «родословные» [20, 21] и т. д.

5. Вычислительные эксперименты

Алгоритмы тестировались на случайных ориентированных и неориентированных гамильтоновых циклах. Для каждого размера графа с помощью алгоритма тасования Фишера-Йетса [28] были сгенерированы 100 пар случайных перестановок с равномерным распределением вероятности.

Для сравнения были выбраны 3 алгоритма:

- поиск с возвратом на основе построения простого пути (BSP);
- поиск с возвратом на основе цепного фиксирования рёбер (BCEF);
- эвристический алгоритм поиска с переменными окрестностями (GVNS) из работы [25], который является модификацией алгоритма имитации отжига [24] и работает на основе построения покрытия циклами без общих вершин через поиск совершенного паросочетания и нескольких операций объединения циклов.

Алгоритмы поиска с возвратом реализованы на Python, для эвристического алгоритма поиска с переменными окрестностями взята готовая реализация на Node.js [25]. Вычислительные эксперименты проведены на машине с Intel(R) Core(TM) i5-4460 с CPU 3.20GHz и 16GB RAM.

Результаты вычислительных экспериментов для неориентированных графов приведены в Таблице 1 и графике на Рис. 5. Результаты для ориентированных графов представлены в Таблице 2 и графиках на Рис. 6 и Рис. 7.

Для каждого набора из 100 тестов приведено среднее время работы алгоритмов в секундах отдельно по задачам имеющим решение и по задачам не имеющим решения. Для двух алгоритмов поиска с возвратом (BSP и BCEF) было установлено ограничение по времени в 2 часа на каждый набор из 100 тестовых задач. Соответственно, в таблицах указано, сколько задач из 100 алгоритм успел решить за 2 часа. Для эвристического алгоритма поиска с переменными окрестностями (GVNS)

Table 1. Computational results for 100 random undirected Hamiltonian cycles

Таблица 1. Вычислительные результаты для 100 неориентированных гамильтоновых циклов

	BSP				BCEF				GVNS			
	Есть решение		Нет решения		Есть решение		Нет решения		Есть решение		«Нет решения»	
V	N	время (с)	N	время (с)	N	время (с)	N	время (с)	N	время (с)	N	время (с)
32	100	0.030	-	-	100	0.015	-	-	100	0.001	-	-
48	100	0.151	-	-	100	0.032	-	-	100	0.002	-	-
64	100	0.154	-	-	100	0.052	-	-	100	0.003	-	-
96	100	0.908	-	-	100	0.09	-	-	100	0.007	-	-
128	100	3.045	-	-	100	0.157	-	-	100	0.012	-	-
192	100	4.257	-	-	100	0.22	-	-	100	0.023	-	-
256	100	19.625	-	-	100	0.487	-	-	100	0.033	-	-
384	9	80.037	-	-	100	1.278	-	-	100	0.079	-	-
512	0	-	-	-	100	2.447	-	-	100	0.121	-	-
768	0	-	-	-	100	3.591	-	-	100	0.272	-	-
1024	0	-	-	-	100	6.305	-	-	100	0.468	-	-
1536	0	-	-	-	100	8.799	-	-	100	1.011	-	-
2048	0	-	-	-	100	18.499	-	-	100	1.821	-	-
3072	0	-	-	-	100	45.542	-	-	100	3.852	-	-
4096	0	-	-	-	86	82.973	-	-	100	7.768	-	-

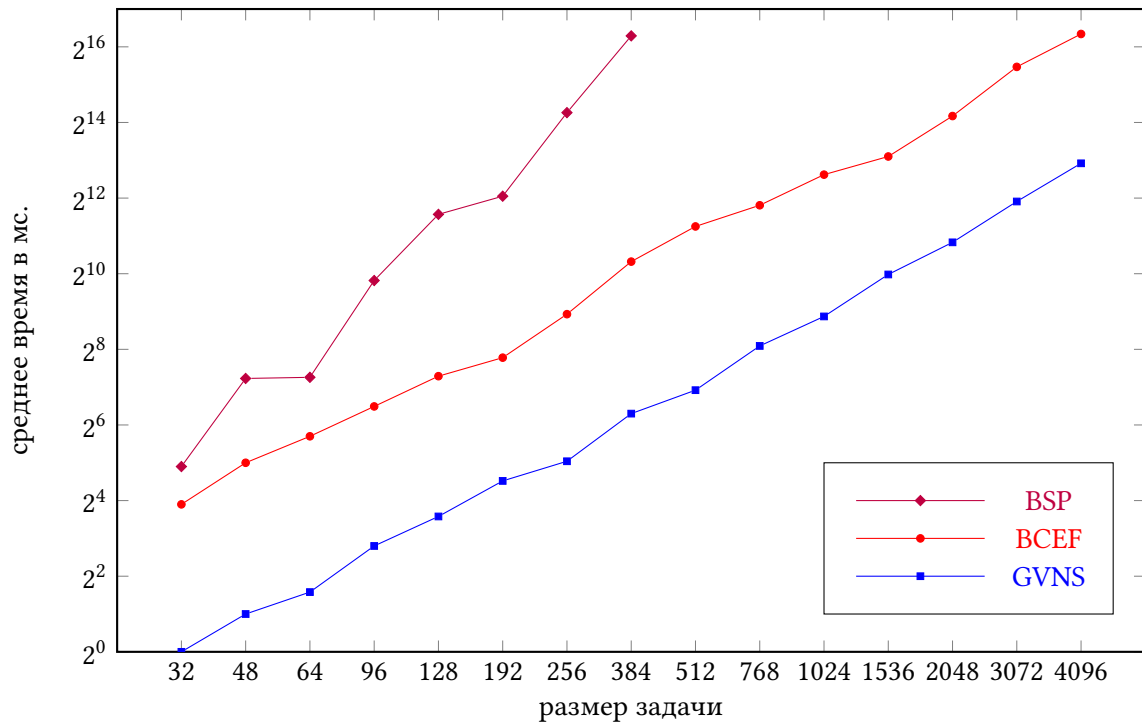


Fig. 5. Computational results for undirected graphs

Рис. 5. Вычислительные результаты для неориентированных графов

Table 2. Computational results for 100 random directed Hamiltonian cycles**Таблица 2.** Вычислительные результаты для 100 ориентированных гамильтоновых циклов

	BSP				BCEF				GVNS			
	Есть решение		Нет решения		Есть решение		Нет решения		Есть решение		«Нет решения»	
V	N	время (с)	N	время (с)	N	время (с)	N	время (с)	N	время (с)	N	время (с)
32	30	0.069	70	0.592	30	0.006	70	0.011	30	0.001	70	0.248
48	20	1.838	80	17.005	20	0.011	80	0.023	20	0.002	80	0.440
64	8	36.850	17	403.016	20	0.026	80	0.044	20	0.002	80	0.714
96	0	-	0	-	19	0.032	81	0.086	19	0.005	81	1.330
128	-	-	-	-	18	0.055	82	0.141	18	0.009	82	2.312
192	-	-	-	-	21	0.059	79	0.262	21	0.014	79	4.569
256	-	-	-	-	25	0.134	75	0.628	25	0.093	75	9.120
384	-	-	-	-	20	0.083	80	1.646	20	0.104	80	19.215
512	-	-	-	-	22	0.143	78	2.131	22	0.112	78	29.048
768	-	-	-	-	19	0.426	81	2.241	19	0.404	81	55.991
1024	-	-	-	-	17	0.847	83	10.545	17	1.679	83	60.000
1536	-	-	-	-	16	0.369	84	7.844	16	0.943	84	60.000
2048	-	-	-	-	15	1.651	85	16.179	14	1.987	86	60.000
3072	-	-	-	-	21	1.059	79	46.589	21	2.999	79	60.000
4096	-	-	-	-	17	1.551	78	91.963	18	4.846	82	60.000

было установлено ограничение в 1 минуту на каждый тест. Это обусловлено тем, что эвристический алгоритм обладает односторонней ошибкой. Если алгоритм находит решение, то решение существует. Однако, эвристический алгоритм не может гарантировать, что решения задачи не существует, лишь, что решение не найдено за заданное время или число итераций.

Известно, что случайные неориентированные регулярные графы обладают гамильтоновым разложением с очень высокой вероятностью [29]. Действительно, для всех тестовых задач на неориентированных мультиграфах (Таблица 1) существовало гамильтоново разложение на циклы, отличные от исходных, и вершины многогранника коммивояжера были не смежны. С геометрической точки зрения это означает, что степени вершин в полиэдральном графе значительно меньше общего числа вершин, поэтому две случайные вершины с высокой вероятностью оказываются не смежными.

По результатам тестирования для неориентированных гамильтоновых циклов оба алгоритма поиска с возвратом проиграли эвристическому алгоритму поиска с переменными окрестностями (GVNS). За отведённое время алгоритм на основе построения простого пути (BSP) решил 709 задач, алгоритм на основе цепного фиксирования рёбер (BCEF) – 1486 задач, и поиск с переменными окрестностями (GVNS) решил все 1500 задач из 1500. По времени работы на решённых задачах GVNS оказался в среднем в 13 раз быстрее чем BCEF и в 640 раз быстрее чем BSP. Среди двух алгоритмов поиска с возвратом, алгоритм на основе цепного фиксирования рёбер (BCEF) оказался значительно эффективнее, решив все тестовые задачи до 3072 вершин и показав время работы на решённых задачах в среднем в 48 раз быстрее, чем поиск на основе простого пути (BSP).

Результаты вычислительных экспериментов для ориентированных циклов (Таблица 2, Рис. 6 и 7) оказались принципиально иными. Только примерно 20% случайных тестовых задач имели решение. Отметим, что это не означает, что соответствующие вершины многогранника коммивояжера смежны, так как задача построения гамильтонова разложения проверяет лишь достаточное условие несмежности вершин в полиэдральном графе.

Худшие результаты из трёх рассмотренных алгоритмов показал поиск с возвратом на основе построения простого пути (BSP). За отведённое на тест время алгоритм решил всего 225 тестовых задач из 1500, не решив ни одной задачи для графов на более чем 64 вершинах. В то время как алгоритм цепного фиксирования рёбер (BCEF) решил 1495 задач, а поиск с переменными

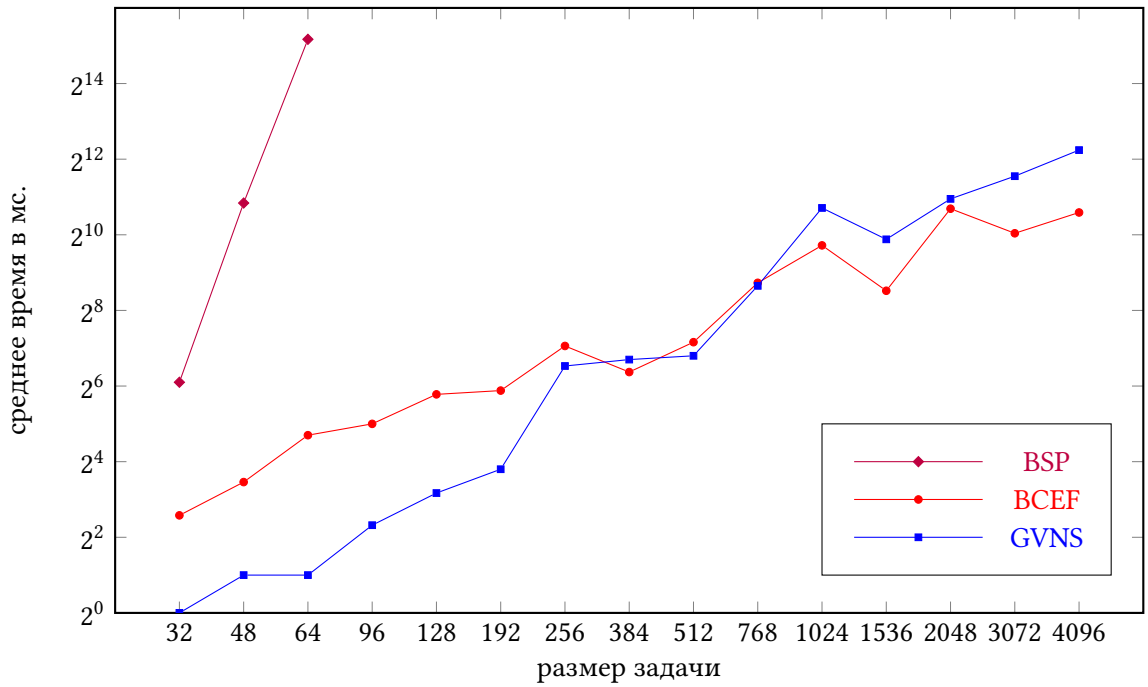


Fig. 6. Computational results for directed graphs with solution

Рис. 6. Вычислительные результаты для ориентированных графов в задачах с решением

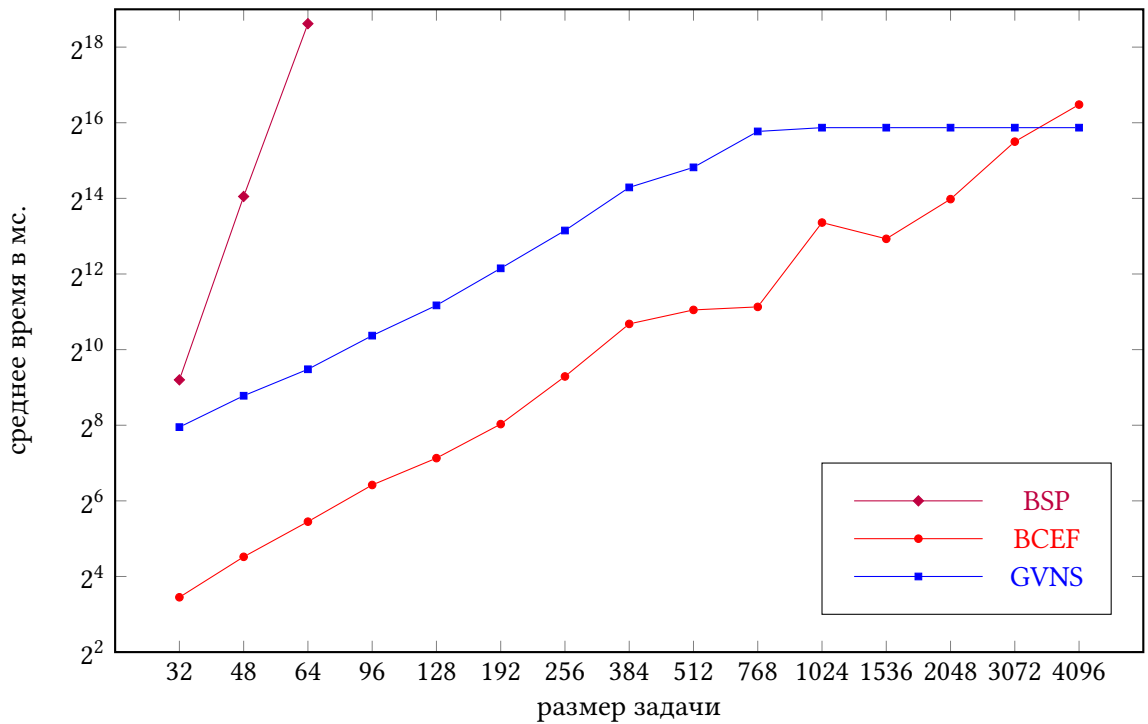


Fig. 7. Computational results for directed graphs without solution

Рис. 7. Вычислительные результаты для ориентированных графов в задачах без решения

окрестностями (GVNS) справился со всеми тестовыми задачами. По времени работы на задачах, имеющих решение, эвристический алгоритм GVNS показал преимущество над BCEF на небольших графах до 512 вершин в среднем в 6 раз. Однако, на графах с более 512 вершинами уже BCEF оказался в среднем в 2 раза быстрее, чем GVNS. В целом, на задачах, для которых решение существует, алгоритмы BCEF и GVNS показали сопоставимые результаты. Некоторый разброс производительности может быть обусловлен разными реализациями алгоритмов. Тем не менее, отметим, что при увеличении размера графа преимущество BCEF над GVNS нарастало. С другой стороны, эвристический алгоритм поиска с переменными окрестностями (GVNS) столкнулся со значительными затруднениями на задачах не имеющих решений. Алгоритм не может определить этот сценарий и выходит лишь при достижении ограничения на время работы или число итераций. На подобных задачах алгоритм на основе цепного фиксирования рёбер (BCEF) оказался в среднем в 16 раз быстрее, чем GVNS. Отчасти это означает, что порог итераций для эвристического алгоритма можно было бы понизить. Однако в таком случае возникла бы опасность потерять существующие решения.

Заключение

В работе были рассмотрены два алгоритма поиска с возвратом для задачи построения гамильтонова разложения 4-регулярного мультиграфа. По результатам вычислительных экспериментов на случайных ориентированных и неориентированных мультиграфах алгоритм поиска с возвратом на основе цепного фиксирования рёбер оказался значительно эффективнее, чем поиск с возвратом на основе построения простого пути. Кроме того, на ориентированных мультиграфах алгоритм цепного фиксирования рёбер показал сопоставимые результаты с ранее известным эвристическим алгоритмом поиска с переменными окрестностями на тестовых задачах, имеющих решение, и значительно превзошёл эвристику на задачах, для которых решения не существует.

Рассматриваемые алгоритмы поиска с возвратом были разработаны для задачи проверки несмежности вершин в полиэдральном графе многогранника коммивояжёра. Однако они также могут быть применены непосредственно к задаче построения гамильтонова разложения регулярного мультиграфа и множеству её приложений.

References

- [1] J. Krarup, “The peripatetic salesman and some related unsolved problems”, in *Combinatorial Programming: Methods and Applications*, vol. 19, Springer Netherlands, 1995, pp. 173–178. DOI: [10.1007/978-94-011-7557-9_8](https://doi.org/10.1007/978-94-011-7557-9_8).
- [2] M. M. Bae and B. Bose, “Edge disjoint Hamiltonian cycles in k -ary n -cubes and hypercubes”, *IEEE Transactions on Computers*, vol. 52, no. 10, pp. 1271–1284, 2003. DOI: [10.1109/TC.2003.1234525](https://doi.org/10.1109/TC.2003.1234525).
- [3] R. F. Bailey, “Error-correcting codes from permutation groups”, *Discrete Mathematics*, vol. 309, no. 13, pp. 4253–4265, 2009. DOI: [10.1016/j.disc.2008.12.027](https://doi.org/10.1016/j.disc.2008.12.027).
- [4] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Y. Zhu, “Tools for Privacy Preserving Distributed Data Mining”, *SIGKDD Explor. Newsl.*, vol. 4, no. 2, pp. 28–34, 2002. DOI: [10.1145/772862.772867](https://doi.org/10.1145/772862.772867).
- [5] R. W. Hung, “Embedding two edge-disjoint Hamiltonian cycles into locally twisted cubes”, *Theoretical Computer Science*, vol. 412, no. 35, pp. 4747–4753, 2011. DOI: [10.1016/j.tcs.2011.05.004](https://doi.org/10.1016/j.tcs.2011.05.004).
- [6] R. Glebov, Z. Luria, and B. Sudakov, “The number of Hamiltonian decompositions of regular graphs”, *Israel Journal of Mathematics*, vol. 222, no. 1, pp. 91–108, 2017. DOI: [10.1007/s11856-017-1583-y](https://doi.org/10.1007/s11856-017-1583-y).
- [7] G. Dantzig, R. Fulkerson, and S. Johnson, “Solution of a Large-Scale Traveling-Salesman Problem”, *Journal of the Operations Research Society of America*, vol. 2, no. 4, pp. 393–410, 1954. DOI: [10.1287/opre.2.4.393](https://doi.org/10.1287/opre.2.4.393).

- [8] D. L. Applegate, R. E. Bixby, V. Chvatál, and W. J. Cook, *The Traveling Salesman Problem: A Computational Study*. Princeton University Press, 2006, ISBN: 9780691129938.
- [9] N. E. Aguilera, R. D. Katz, and P. B. Tolomei, “Vertex adjacencies in the set covering polyhedron”, *Discrete Applied Mathematics*, vol. 218, pp. 40–56, 2017. DOI: [10.1016/j.dam.2016.10.024](https://doi.org/10.1016/j.dam.2016.10.024).
- [10] M. L. Balinski, “Signature Methods for the Assignment Problem”, *Operations Research*, vol. 33, no. 3, pp. 527–536, 1985. DOI: [10.1287/opre.33.3.527](https://doi.org/10.1287/opre.33.3.527).
- [11] C. R. Chegireddy and H. W. Hamacher, “Algorithms for finding K -best perfect matchings”, *Discrete Applied Mathematics*, vol. 18, no. 2, pp. 155–165, 1987. DOI: [10.1016/0166-218X\(87\)90017-5](https://doi.org/10.1016/0166-218X(87)90017-5).
- [12] E. F. Combarro and P. Miranda, “Adjacency on the order polytope with applications to the theory of fuzzy measures”, *Fuzzy Sets and Systems*, vol. 161, no. 5, pp. 619–641, 2010. DOI: [10.1016/j.fss.2009.05.004](https://doi.org/10.1016/j.fss.2009.05.004).
- [13] H. N. Gabow, “Two Algorithms for Generating Weighted Spanning Trees in Order”, *SIAM Journal on Computing*, vol. 6, no. 1, pp. 139–150, 1977. DOI: [10.1137/0206011](https://doi.org/10.1137/0206011).
- [14] V. A. Bondarenko, “Nonpolynomial lower bounds for the complexity of the traveling salesman problem in a class of algorithms”, *Automation and Remote Control*, vol. 44, no. 9, pp. 1137–1142, 1983.
- [15] V. Bondarenko and A. Nikolaev, “On Graphs of the Cone Decompositions for the Min-Cut and Max-Cut Problems”, *International Journal of Mathematics and Mathematical Sciences*, vol. 2016, p. 7 863 650, 2016. DOI: [10.1155/2016/7863650](https://doi.org/10.1155/2016/7863650).
- [16] M. Grötschel and M. Padberg, “Polyhedral theory”, in *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, John Wiley, Chichester, 1985, pp. 251–305.
- [17] C. H. Papadimitriou, “The adjacency relation on the traveling salesman polytope is NP-Complete”, *Mathematical Programming*, vol. 14, no. 1, pp. 312–324, 1978. DOI: [10.1007/BF01588973](https://doi.org/10.1007/BF01588973).
- [18] V. A. Bondarenko and A. V. Nikolaev, “On the Skeleton of the Polytope of Pyramidal Tours”, *Journal of Applied and Industrial Mathematics*, vol. 12, no. 1, pp. 9–18, 2018. DOI: [10.1134/S1990478918010027](https://doi.org/10.1134/S1990478918010027).
- [19] A. Nikolaev, “On vertex adjacencies in the polytope of pyramidal tours with step-backs”, in *Mathematical Optimization Theory and Operations Research. MOTOR 2019*, ser. LNCS, vol. 11548, Springer, 2019, pp. 247–263. DOI: [10.1007/978-3-030-22629-9_18](https://doi.org/10.1007/978-3-030-22629-9_18).
- [20] T. S. Arthanari, “On pedigree polytopes and Hamiltonian cycles”, *Discrete Mathematics*, vol. 306, no. 14, pp. 1474–1492, 2006. DOI: [10.1016/j.disc.2005.11.030](https://doi.org/10.1016/j.disc.2005.11.030).
- [21] T. S. Arthanari, “Study of the pedigree polytope and a sufficiency condition for nonadjacency in the tour polytope”, *Discrete Optimization*, vol. 10, no. 3, pp. 224–232, 2013. DOI: [10.1016/j.disopt.2013.07.001](https://doi.org/10.1016/j.disopt.2013.07.001).
- [22] M. R. Rao, “Adjacency of the Traveling Salesman Tours and 0 – 1 Vertices”, *SIAM Journal on Applied Mathematics*, vol. 30, no. 2, pp. 191–198, 1976. DOI: [10.1137/0130021](https://doi.org/10.1137/0130021).
- [23] B. Péroche, “NP-completeness of some problems of partitioning and covering in graphs”, *Discrete Applied Mathematics*, vol. 8, no. 2, pp. 195–208, 1984. DOI: [10.1016/0166-218X\(84\)90101-X](https://doi.org/10.1016/0166-218X(84)90101-X).
- [24] A. Kozlova and A. Nikolaev, “Simulated annealing approach to verify vertex adjacencies in the traveling salesperson polytope”, in *Mathematical Optimization Theory and Operations Research. MOTOR 2019*, ser. LNCS, vol. 11548, Springer, 2019, pp. 374–389. DOI: [10.1007/978-3-030-22629-9_26](https://doi.org/10.1007/978-3-030-22629-9_26).
- [25] A. Nikolaev and A. Kozlova, “Hamiltonian decomposition and verifying vertex adjacency in 1-skeleton of the traveling salesperson polytope by variable neighborhood search”, *Journal of Combinatorial Optimization*, 2020. DOI: [10.1007/s10878-020-00652-7](https://doi.org/10.1007/s10878-020-00652-7).

- [26] S. S. Skiena, *The Algorithm Design Manual*. Springer, 2008. DOI: [10.1007/978-1-84800-070-4](https://doi.org/10.1007/978-1-84800-070-4).
- [27] A. V. Korostil and A. V. Nikolaev, “Algoritm poiska s vozvratom dlya postroeniya gamil’tonova razlozheniya 4-regulyarnogo mul’tigrafa (Backtracking algorithm to construct the Hamiltonian decomposition of a 4-regular multigraph)”, in *Zametki po informatike i matematike (Notes on Computer Science and Mathematics)*, vol. 12, P.G. Demidov Yaroslavl State University, 2020, pp. 91–97.
- [28] D. E. Knuth, *The Art of Computer Programming, Volume 2 (3rd Ed.): Seminumerical Algorithms*. USA: Addison-Wesley Longman Publishing Co., Inc., 1997, ISBN: 0201896842. DOI: [10.5555/270146](https://doi.org/10.5555/270146).
- [29] J. H. Kim and N. C. Wormald, “Random Matchings Which Induce Hamilton Cycles and Hamiltonian Decompositions of Random Regular Graphs”, *Journal of Combinatorial Theory, Series B*, vol. 81, no. 1, pp. 20–44, 2001. DOI: [10.1006/jctb.2000.1991](https://doi.org/10.1006/jctb.2000.1991).