ISSN 2311-4738 (print), ISSN 2413-3000 (online)

УДК 519.1 DOI: 10.20998/2413-3000.2020.2.6

В. Ф. ПРОКОПЕНКОВ

НОВЫЙ МЕТОД ПОИСКА ГАМИЛЬТОНОВА ЦИКЛА НА ГРАФЕ

В дискретной математике существует много задач, которые относятся к *NP* классу сложности. Решение этих задач имеет как теоретическую, так и практическую ценность. Одной из них является задача поиска гамильтонова цикла на графе. Целью работы является разработка нового метода и алгоритма решения этой задачи предпочтительного имеющимся по затратам времени и качеству получаемого решения. В работе выполнен анализ проблемы и существующих методов её решения, определены недостатки этих методов. Показано, что все известные методы решения этой задачи строятся на реализации перебора вариантов решений либо на интуитивных эвристиках. Первые из них характеризуются неполиномиальными затратами времени, а вторые – не обеспечивают получение оптимального решения. Причиной такого состояния является невозможность сформулировать условия, определяющие оптимальное решение задачи. В таком случае единственно возможным способом решения задачи по-прежнему остаётся перебор вариантов, а для снижения затрат нахождения решения необходимо прибегать к сокращению пространства перебора вариантов. В работе изложены новые принципы нахождения решения и предложен новый метод решения задачи. На основе нового метода разработан полиномиальный алгоритм решения задачи. Поиск гамильтонова цикла в графе сводится к поиску замкнутого пути в новом графе кратчайших путей. Для построения графа кратчайших путей используется алгоритм Дейкстры. Пространство перебора вариантов решения задачи состоит из решений, которые строятся из каждой вершины графа в графе кратчайших путей. Тестирование разработанной программы показало работоспособность разработанного метода и алгоритма решения задачи. Предложенный метод решения существенно сокращает пространство перебора и позволяет находить оптимальное решение с полиномиальной сложностью, как в полном, так и неполном графах. Рассмотренный метод пригоден для параллельной реализации, что даёт дополнительный выигрыш во времени и позволяет на всю мощь использовать параллельные возможности современных многоядерных процессоров.

Ключевые слова: граф, граф кратчайших путей, гамильтонов цикл, сложность, NP-полнота, алгоритм Дейкстры, параллельная обработка.

В. П. ПРОКОПЕНКОВ

НОВИЙ МЕТОД ПОШУКУ ГАМІЛЬТОНА ЦИКЛУ НА ГРАФІ

У дискретній математиці існує багато задач, які відносяться до *NP* класу складності. Вирішення цих задач має як теоретичну, так і практичну цінність. Однією з них є задача пошуку гамільтонова циклу на графі. Метою роботи є розробка нового методу і алгоритму вирішення цієї задачі кращого за наявні за витратами часу і якості одержуваного рішення. В роботі виконаний аналіз проблеми та існуючих методів її вирішення, визначено недоліки цих методів. Показано, що всі відомі методи вирішення цього завдання будуються на реалізації перебору варіантів рішень або на інтуїтивних евристиках. Перші з них характеризуються неполіноміальними витратами часу, а другі – не забезпечують отримання оптимального рішення. Причиною такого стану є неможливість сформулювати умови, що визначають оптимальне рішення задачі. В такому випадку єдино можливим способом вирішення задачі залишається перебір варіантів, а для зниження витрат знаходження рішення необхідно вдаватися до скорочення простору перебору варіантів. У роботі викладені нові принципи знаходження рішення і запропонований новий метод вирішення задачі. На основі нового методу розроблений поліноміальний алгоритм вирішення задачі. Пошук гамільтонова циклу в графі зводиться до пошуку замкнутого шляху в новому графі найкоротших шляхів. Для побудови графа найкоротших шляхів використовується алгоритм Дейкстри. Простір перебору варіантів вирішення задачі складається з рішень, які будуються з кожної вершини графа в графі найкоротших шляхів. Тестування розробленої програми показало працездатність розробленого методу і алгоритму вирішення задачі. Запропонований метод рішення істотно скорочує простір перебору і дозволяє знаходити оптимальне рішення з поліноміальною складністю, як у повному, так і неповному графах. Розглянутий метод придатний для паралельної реалізації, що дає додатковий виграш у часі і дозволяє на всю міць використовувати паралельні можливості сучасних багатоядерних процесорів.

Ключові слова: цінність, енергія, компетентність, управління, проєктна напруженість, ступінь унікальності співробітника.

V. F. PROKOPENKOV

A NEW METHOD FOR SEARCHING A HAMILTON CYCLE ON A GRAPH

In discrete mathematics, there are many problems that belong to the *NP* class of complexity. The solution of these problems has both theoretical and practical value. One of them is the problem of finding a Hamiltonian cycle on a graph. The aim of the work is to develop a new method and algorithm for solving this problem, which are preferable to the existing methods by time and quality of the solution. The paper analyzes the problem and existing methods of solving it, and identifies the disadvantages of these methods. It is shown that all known methods for solving this problem are based on the implementation of enumeration of solutions or on intuitive heuristics. The first of them are characterized by a non-polynomial amount of time for solving, and the second does not provide the optimal solution. The reason for this state is the inability to formulate conditions that determine the optimal solution of the problem. In this case, the only possible way to solve the problem is still to enumerate over the options, and to reduce the cost of finding a solution, you must resort to reducing the space for iterating over the options. The paper presents new principles for finding a solution and offers a new method for solving the problem. On the basis of the new method, a polynomial algorithm for solving the problem is developed. The search for a Hamiltonian cycle in a graph is reduced to the search for a closed path in a new graph of shortest paths. Dijkstra's algorithm is used to construct the shortest paths graph. The space of enumeration of solutions to the problem consists of solutions that are constructed from each vertex of the graph in the shortest paths graph. Testing of the developed program showed the efficiency of the developed method and algorithm for solving the problem. The proposed solution method significantly reduces the search space and allows us to find the optimal solution with polynomial complexity, both in full and incomplete graphs. The considered method is suitable for parallel implementation, which gives an additional gain in time and allows you to fully use the parallel capabilities of modern multi-core processors.

Keywords: graph, shortest path graph, Hamiltonian cycle, complexity, NP-completeness, Dijkstra algorithm, parallel processing.

Введение. Игра "кругосветного путешествия" по додекаэдру, предложенная У. Гамильтоном [1] требовала выйти из одной его вершины, обойти все его вершины по имеющимся ребрам и вернуться в исходную вершину с условием, что одну и ту же вершину при обходе можно посетить не более одного раза. Вершины символизировали крупнейшие города мира, а рёбра – пути между ними. Додекаэдр, многогранник с 12 гранями в форме правильных пятиугольников, можно представить некоторым плоским графом, состоящим из 20 вершин и 30 ребер.

В теории графов, разделе дискретной математики, такой замкнутый путь в графе носит название гамильтонова цикла [2]. Гамильтонов цикл существует не во всяком графе, а если существует, то граф называется гамильтоновым. Задача поиска гамильтонова цикла в графе, как и задача проверки, является ли граф гамильтонов, относится к труднорешаемым или *NP* полным задачам [3]. Не полиномиальная сложность этих задач обусловлена размером пространства решений, которое в общем случае необходимо перебрать, чтобы найти наилучшее решение. Так, если в полном графе n вершин, то из первой вершины существует $n-1$ вариантов перехода, из второй вершины $n-2$ вариантов, из третьей – $n-3$ вариантов и т.д. из предпоследней вершины – 1 вариант, что приводит $n(n-1)(n-2)(n-3)...2\cdot 1=(n-1)!$ вариантам возможных решений, которые необходимо перебрать, чтобы найти наилучшее.

Поиск гамильтонова цикла применяется в разных практических задачах планирования, коммуникаций, транспортных, экономических и т.д. Но на сегодняшний день универсального эффективного метода гарантирующего получение решения за полиномиальное время не существует. Таким образом, очевидна важность и необходимость исследований для разработки новых методов и алгоритмов решения описанной проблемы.

Анализ последних публикаций. Для проверки является ли граф гамильтоновым можно воспользоваться одним из алгебраических критериев Дирака [4], Оре [5], Поша или Хватала [6], но они очень общие и не пригодны для произвольных графов на практике [7].

В производственных задачах, например, [8] не всегда достаточно только знать, что граф является гамильтоновым. В большинстве случаев требуется находить сам гамильтонов цикл и чтобы он имел минимальную длину. Длина цикла оценивается как сумма весов рёбер графа, составляющих цикл. Если в графе ребра в разных направлениях имеют разные веса, то приходится работать с ориентированным графом. В общем случае для задачи поиска гамильтонова цикла эффективных полиномиальных алгоритмов решения не существует [9].

Для поиска гамильтонова цикла в графе все известные методы и алгоритмы можно разделить на точные и эвристические [10]. Первые теоретически гарантируют получение оптимального решения, но для задач большой размерности на практике требуют недопустимо много времени, из-за чего они не применимы. Вторые, не гарантируют получения оптимального решения, строятся на эвристических соображениях, но приемлемы по затратам времени.

Первую группу составляют методы [11, 12] динамического программирования, имеющие сложность порядка 2^n и все переборные алгоритмы. В переборных алгоритмах последовательно перебираются все возможные решения от первого до последнего, и отбирается лучшее по длине искомого цикла. Для формирования проверяемых циклов могут быть использованы комбинаторные методы, так как каждый гамильтонов цикл графа можно рассматривать как перестановку всех вершин графа. Иначе выделять циклы в графе можно в результате обхода графа, начиная с некоторой начальной вершины, реализуя поиск с возвратами, например, алгоритмом Робертса и Флореса [13], имеющим экспоненциальную сложность от числа вершин. Процесс перебора можно сократить, используя метод ветвей и границ [14], отбрасывая на каждом шаге алгоритма заведомо неоптимальные решения.

Эвристические алгоритмы строятся на определённых эвристиках или логически обоснованных идеях, но они не гарантируют отыскания оптимального результата. Примерами таких алгоритмов могут служить муравьиный [15], генетический [16–18] и гибридный [19], их объединяющий, они имеют полиномиальную сложность. Муравьиный алгоритм основывается на поведении муравьиной колонии и моделирует испарение феромонов. Генетические алгоритмы основаны на эволюции и реализуют эвристический поиск, моделируя естественный отбор в природе. Каждый гамильтонов цикл рассматривается как хромосома. На начальном этапе имеется набор таких хромосом (начальная популяция), а на каждом последующем цикле из имеющейся популяции производится новая путём попарного скрещивания и мутации. Генетический алгоритм позволяет в однократном применении получить более одного решения и отобрать лучшее, но с ростом размерности вероятность получения оптимального решения снижается. Кроме того, для муравьиного и генетического алгоритма проблематично подобрать настраивающие их параметры.

Для решения рассматриваемой задачи появляются всё новые алгоритмы. Так для кубических графов в [20] было предложено решение со сложностью 1.26n, в [21] сложность решения понижена до 1.251n.

Цель работы. Разработка метода и алгоритма решения задачи поиска гамильтонова цикла на графе, к которому предъявляются требования: чтобы имел сложность не выше полиномиальной; был точным, т.е. гарантировал получения оптимального решения; был пригоден для любого типа исходного графа.

На сегодняшний день не существует алгоритма поиска гамильтонова цикла, удовлетворяющего перечисленным требованиям. Проблема в разработке такого алгоритма обусловлена тем, что не удаётся найти определяющие факторы, которые однозначно приводят к оптимальному решению задачи. И поэтому, единственно возможным для получения оптимального решения остается перебор всех или большей части возможных решений для выбора лучшего или отыскания интуитивных приёмов, которые впоследствии, возможно, и позволят найти ту теорию, которая приведёт к разработке точного решения.

Если для задачи путём перебора не удаётся за приемлемое время найти оптимальное решение, единственно возможным остаётся сократить область его поиска.

Постановка задачи. Пусть задан граф $G = \langle V, E \rangle$, где $V = \left\{ v_i \mid i = \overline{1,n} \right\}$ – это множество вершин, а $E = \left\{ e_{ij} \mid i, j = \overline{1,n}, i \neq j \right\}$ – множество дуг графа. Дуга e_{ij} определяет наличие соединения между вершинами v_i и v_j и характеризуется расстоянием d_{ij}. Пусть задана начальная вершина $v_s \in V$. Необходимо найти гамильтонов цикл минимальной длины из вершины v_s, т.е. кортеж $< v_1, v_2, ..., v_i, ...v_{n-1}, v_n, v_{n+1} >$ из вершин графа G, для которого выполняются условия:

1) $v_1 = v_s$;

2) $v_{n+1} = v_s$;

3) для любой пары вершин $v_i, v_j \mid i, j \in \{\overline{1,N}\}$ справедливо: если $i \neq j$, то $v_i \neq v_j$;

4) для $\forall v_k \mid k \in \{\overline{2,N}\}$ в графе G существуют дуги: $e_{k-1,k}$ – из вершины v_{k-1} в вершину v_k и $e_{k,k+1}$ – из вершины v_k в вершину v_{k+1}.

Метод решения задачи. Для поиска гамильтонова цикла предлагается новый метод, основанный на алгоритме Дейкстры, определяющем кратчайшие пути в графе. В основу метода положены следующие логические рассуждения.

Гамильтонов цикл кратчайшей длины в графе – это кратчайший замкнутый путь в графе соединяющий все вершины графа. А значит, он не может в своей структуре не содержать кратчайшие пути между некоторыми (или даже всеми) вершинами графа. В идеальном случае он содержит в себе такое подмножество кратчайших путей между вершинами графа и других (не кратчайших путей), которые в совокупности делают его кратчайшим гамильтоновым циклом. Таким образом, логично искать или строить гамильтонов цикл, отталкиваясь от кратчайших путей в графе между вершинами. Например, в графе (рис. 1), вершины которого образуют кольцо, имеется единственный цикл, он же кратчайший и все вершины в нём образованы на кратчайших путях между вершинами. И для такого графа достаточно найти кратчайшие пути (например, используя алгоритм Дейкстры [22]) из какой-либо вершины в остальные и самый длинный из них будет основой для гамильтонова цикла. Для произвольного графа так удачно может сложиться только при определённом стечении обстоятельств.

Если отталкиваться в поиске решения от перебора всех возможных гамильтоновых циклов в графе, то если не перебирать всё пространство решений (а сократить этот перебор), то процедуру поиска необходимо строить так, чтобы в числе сокращенных не оказался гамильтонов цикл минимальной длины. Например, если мы начинаем построения гамильтонова цикла из вершины p, то должны построить все возможные гамильтоновы циклы, начинающиеся из вершины p, построенные на кратчайших путях.

Вполне возможно, что гамильтонова цикла в графе, который строиться на каком-нибудь кратчайшем пути, выходящем из выбранной вершины p, не существует. Если так, то попытка построить гамильтонов цикл из вершины p закончится неудачей, при этом гамильтонов цикл может существовать.

Например, на рис. 2 показан граф, в котором невозможно построить гамильтонов цикл из вершины с номером 1, основываясь на кратчайших путях из этой вершины: <1 – 2>, <1 – 2 – 6>, <1 – 2 – 6 – 3>, <1 – 4> и <1 – 5>.

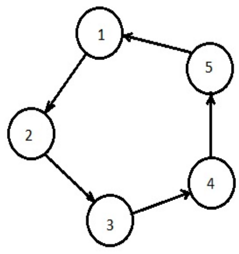

Рис. 1

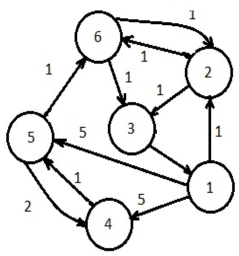

Рис. 2

Попытка построить гамильтонов цикл из вершины 2 приведёт к успеху. Из этой вершины идут такие кратчайшие пути: <2 – 3>, <2 – 3 – 1>, <2 – 3 – 1 – 4>, < 2 – 3 – 1 – 5>, <2 – 6>.

Гамильтонов цикл из вершины с номером 2 собирается как комбинация существующих кратчайших путей в графе из вершины 2 – <2 – 3 – 1 – 4>, из вершины 4 – <4 – 5>, из вершины 5 – <5 – 6> и из вершины 6 – <6 – 2>.

Если в результате мы сможем построить гамильтонов цикл, начиная с какой-либо вершины p (и он будет составлен не только из кратчайших путей), то нельзя однозначно сказать, что это будет цикл минимальной длины. И это так, поскольку в таком случае мы исследуем только одно подпространство пространства решений, которые получаются кратчайшими путями, идущими из вершины p. Следовательно, чтобы не потерять оптимальное решение, мы должны выполнить рассмотренную процедуру для всех остальных вершин графа, рассматривая эти вершины как начальные.

Решения, построенные из каждой вершины графа, формируют пространство поиска оптимального решения задачи. Для определения оптимального решения можно использовать схему, предложенную в [23], которую опишем так.

Пусть имеем задачу T, которая предполагает наличие конечного множества разных начальных условий $\{C_i\}$ для поиска решений и алгоритм A, который применим к заданным начальным условиям C_i и находит одно наилучшее решение в соответствующем этим условиям подмножестве пространства решений. Тогда схема поиска предполагает выполнение таких шагов:

Цикл по i от 1 до $|\{C_i\}|$:

П.1. Найти наилучшее решение $r_i = A(C_i)$ для начальных условий C_i.

П.2. Если лучшее решение R не определено или R хуже текущего решения r_i, то $R = r_i$.

Рассмотренная схема позволяет найти оптимальное решение задачи за полиномиальное время только тогда, когда множество начальных условий $\{C_i\}$ полное и для каждого C_i алгоритм A имеет полиномиальную оценку сложности. Под полнотой надо понимать, что не существует каких-либо других начальных условий не принадлежащих $\{C_i\}$, для которых можно применить алгоритм A.

Предлагаемый метод поиска гамильтонова цикла основывается на рассмотренной схеме. В рамках этой схемы T – это задача поиска гамильтонова цикла минимальной длины. Начальные условия поиска C_i это пара $C_i = (p, path_{pk}) \mid k \in \overline{1,n}, k != p$, где p – это вершина из которой строится гамильтонов цикл; $path_{pk}$ – это кратчайший путь из вершины p в одну из вершин графа, отличную от вершины p. Алгоритм A для заданных начальных условий C_i строит гамильтонов цикл из вершины p, который начинается кратчайшим путём $path_{pk}$ в вершину k (при заданных условиях C_i цикл может не существовать). Для своей реализации процедура построения пути A требует составления графа кратчайших путей $G^* = \langle V^*, E^* \rangle$ такого, что $V^* = V$ – это множество вершин, совпадающее с множеством вершин исходного графа G, а $E^* = \left\{ e^*_{ij} \mid i, j = \overline{1,n}, i \neq j \right\}$ множество дуг. Дуга e^*_{ij} существует, если существует путь длины d^*_{ij} в графе G из вершины v_i в вершину v_j. В качестве таких путей используются кратчайшие пути. Таким образом, построение гамильтонова цикла алгоритмом A сводится к поиску одного (если он существует) пути в графе кратчайших путей G^* из вершины p, проходящем через вершину k.

Поскольку для каждой вершины p в общем случае может существовать $n-1$ начальных условий (по числу вершин графа, за исключением вершины p), то для каждой вершины $p \in V$ алгоритм A будет выполнен $n-1$ раз, а для всех вершин графа – $n*(n-1)$ раз. Заметим, что алгоритм A не должен перебирать варианты построения гамильтонова цикла, например, реализуя схему с возвратами. Поэтому, сложность алгоритма A не превысит порядка $O(n^2)$, а значит, общая сложность алгоритма для предложенного метода составит $O(n^4)$.

Тестирование метода. Для проверки разработанного метода решения была выполнена программная реализация на языке C#, которая подтвердила ожидаемые результаты. Тестирование было выполнено на полном и неполном графах, сгенерированных программным способом [24]. Для тестового графа «додекаэдр» на плоскости, описание которого представлено на рис.3, на рис. 4–5 представлены лучшие найденные решения, полученные программным способом. Наилучшее решение имеет длину гамильтонова цикла равную 7227, а затраченное время на поиск решения составило 0,041 минуты.

Вершина		
№	Координаты	
	x	y
0	694	945
1	905	945
2	971	744
3	800	620
4	628	744
5	438	682
6	267	973
7	39	1047
8	800	1600
9	1560	1047
10	1270	152
11	329	152
12	470	346
13	800	420
14	1129	346
15	1161	682
16	1332	973
17	1023	1107
18	800	1360
19	576	1107

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	0x0	211	0x0	0x0	211	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	200
1	211	0x0	211	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	200	0x0	0x0
2	0x0	211	0x0	211	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	199	0x0	0x0	0x0	0x0
3	0x0	0x0	211	0x0	212	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	200	0x0	0x0	0x0	0x0	0x0	0x0
4	211	0x0	0x0	212	0x0	199	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
5	0x0	0x0	0x0	0x0	199	0x0	337	0x0	0x0	0x0	0x0	0x0	337	0x0	0x0	0x0	0x0	0x0	0x0	0x0
6	0x0	0x0	0x0	0x0	0x0	337	0x0	239	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	336
7	0x0	0x0	0x0	0x0	0x0	0x0	239	0x0	940	0x0	0x0	940	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
8	0x0	0x0	0x0	0x0	0x0	0x0	0x0	940	0x0	939	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	240	0x0
9	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	939	0x0	940	0x0	0x0	0x0	0x0	0x0	239	0x0	0x0	0x0
10	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	940	0x0	941	0x0	0x0	239	0x0	0x0	0x0	0x0	0x0
11	0x0	0x0	0x0	0x0	0x0	0x0	0x0	940	0x0	0x0	941	0x0	239	0x0	0x0	0x0	0x0	0x0	0x0	0x0
12	0x0	0x0	0x0	0x0	0x0	337	0x0	0x0	0x0	0x0	0x0	239	0x0	338	0x0	0x0	0x0	0x0	0x0	0x0
13	0x0	0x0	0x0	200	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	338	0x0	337	0x0	0x0	0x0	0x0	0x0
14	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	239	0x0	0x0	337	0x0	337	0x0	0x0	0x0	0x0
15	0x0	0x0	199	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	337	0x0	337	0x0	0x0	0x0
16	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	239	0x0	0x0	0x0	0x0	0x0	337	0x0	336	0x0	0x0
17	0x0	200	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	336	0x0	337	0x0
18	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	240	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	337	0x0	337
19	200	0x0	0x0	0x0	0x0	0x0	336	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	337	0x0

Рис. 3. Описание тестового графа «Додекаэдр»

```
поток из стартовой вершины 5 через вершину 16
0. 5->16( 6, 1157 ): { 5, 4, 0, 1, 17, 16 }
1. 16->2( 3, 536 ): { 16, 15, 2 }
2. 2->3( 2, 211 ): { 2, 3 }
3. 3->10( 4, 776 ): { 3, 13, 14, 10 }
4. 10->9( 2, 940 ): { 10, 9 }
5. 9->8( 2, 939 ): { 9, 8 }
6. 8->6( 4, 913 ): { 8, 18, 19, 6 }
7. 6->7( 2, 239 ): { 6, 7 }
8. 7->11( 2, 940 ): { 7, 11 }
9. 11->5( 3, 576 ): { 11, 12, 5 }

Цикл:
5->5( 21, 7227 ): { 5, 4, 0, 1, 17, 16, 15, 2, 3, 13, 14, 10, 9, 8, 18, 19, 6, 7, 11, 12, 5 }

поток из стартовой вершины 3 через вершину 8
0. 3->8( 6, 1199 ): { 3, 2, 1, 17, 18, 8 }
1. 8->7( 2, 940 ): { 8, 7 }
2. 7->0( 4, 775 ): { 7, 6, 19, 0 }
3. 0->4( 2, 211 ): { 0, 4 }
4. 4->5( 2, 199 ): { 4, 5 }
5. 5->11( 3, 576 ): { 5, 12, 11 }
6. 11->10( 2, 941 ): { 11, 10 }
7. 10->9( 2, 940 ): { 10, 9 }
8. 9->14( 4, 913 ): { 9, 16, 15, 14 }
9. 14->3( 3, 537 ): { 14, 13, 3 }

Цикл:
3->3( 21, 7231 ): { 3, 2, 1, 17, 18, 8, 7, 6, 19, 0, 4, 5, 12, 11, 10, 9, 16, 15, 14, 13, 3 }

поток из стартовой вершины 8 через вершину 15
0. 8->15( 6, 1187 ): { 8, 18, 17, 1, 2, 15 }
1. 15->9( 3, 576 ): { 15, 16, 9 }
2. 9->10( 2, 940 ): { 9, 10 }
3. 10->3( 4, 776 ): { 10, 14, 13, 3 }
4. 3->0( 3, 423 ): { 3, 4, 0 }
5. 0->6( 3, 536 ): { 0, 19, 6 }
6. 6->5( 2, 337 ): { 6, 5 }
7. 5->11( 3, 576 ): { 5, 12, 11 }
8. 11->7( 2, 940 ): { 11, 7 }
9. 7->8( 2, 940 ): { 7, 8 }

Цикл:
8->8( 21, 7231 ): { 8, 18, 17, 1, 2, 15, 16, 9, 10, 14, 13, 3, 4, 0, 19, 6, 5, 12, 11, 7, 8 }

поток из стартовой вершины 3 через вершину 19
0. 3->19( 4, 623 ): { 3, 4, 0, 19 }
1. 19->6( 2, 336 ): { 19, 6 }
2. 6->5( 2, 337 ): { 6, 5 }
3. 5->11( 3, 576 ): { 5, 12, 11 }
4. 11->7( 2, 940 ): { 11, 7 }
5. 7->8( 2, 940 ): { 7, 8 }
6. 8->1( 4, 777 ): { 8, 18, 17, 1 }
7. 1->2( 2, 211 ): { 1, 2 }
8. 2->9( 4, 775 ): { 2, 15, 16, 9 }
9. 9->10( 2, 940 ): { 9, 10 }
10. 10->3( 4, 776 ): { 10, 14, 13, 3 }

Цикл:
3->3( 21, 7231 ): { 3, 4, 0, 19, 6, 5, 12, 11, 7, 8, 18, 17, 1, 2, 15, 16, 9, 10, 14, 13, 3 }

20.09.2019 15:40:06
Время счета 0,0407523316666667минут

Найденные Решения:
1. 0->0(21,8178):  { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 0  }
2. 0->0(21,8178):  { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 0  }
3. 3->3(21,7231):  { 3, 4, 0, 19, 6, 5, 12, 11, 7, 8, 18, 17, 1, 2, 15, 16, 9, 10, 14, 13, 3  }
4. 3->3(21,7231):  { 3, 2, 1, 17, 18, 8, 7, 6, 19, 0, 4, 5, 12, 11, 10, 9, 16, 15, 14, 13, 3  }
5. 3->3(21,7231):  { 3, 2, 1, 17, 18, 8, 7, 6, 19, 0, 4, 5, 12, 11, 10, 9, 16, 15, 14, 13, 3  }
6. 5->5(21,7227):  { 5, 4, 0, 1, 17, 16, 15, 2, 3, 13, 14, 10, 9, 8, 18, 19, 6, 7, 11, 12, 5  }
7. 8->8(21,7231):  { 8, 18, 17, 1, 2, 15, 16, 9, 10, 14, 13, 3, 4, 0, 19, 6, 5, 12, 11, 7, 8  }
8. 13->13(21,7231):{ 13, 3, 2, 1, 17, 18, 8, 7, 6, 19, 0, 4, 5, 12, 11, 10, 9, 16, 15, 14, 13 }
9. 13->13(21,7231):{ 13, 3, 4, 0, 19, 6, 5, 12, 11, 7, 8, 18, 17, 1, 2, 15, 16, 9, 10, 14, 13 }
10.13->13(21,7231):{ 13, 3, 2, 1, 17, 18, 8, 7, 6, 19, 0, 4, 5, 12, 11, 10, 9, 16, 15, 14, 13 }
11.16->16(21,8178):{ 16, 17, 18, 19, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 }
12.17->17(21,8178):{ 17, 18, 19, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17 }
13.18->18(21,8178):{ 18, 19, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 }
14.18->18(21,8178):{ 18, 19, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 }
15.19->19(21,8178):{ 19, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19 }
16.19->19(21,8178):{ 19, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19 }
17.19->19(21,8178):{ 19, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19 }
<Конец списка>
```

Рис. 4. Фрагмент листинга работы программы

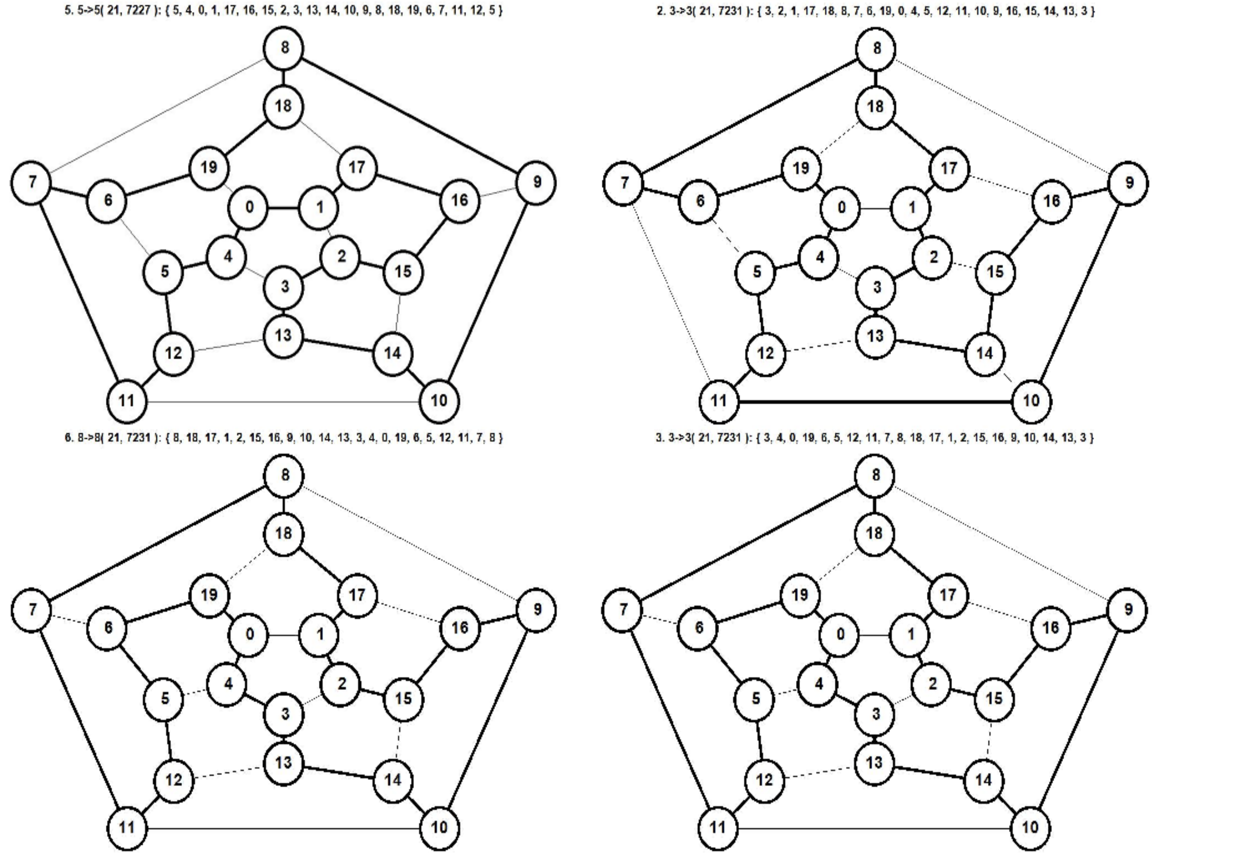

Рис. 5. Лучшие решения

Выводы. В статье предложен новый метод поиска гамильтонова цикла на графе, который программно реализован и протестирован и предусматривает выполнение следующих этапов. На первом этапе определяются кратчайшие пути в графе между каждой парой различных вершин, и формируется граф кратчайших путей. На втором этапе из каждой вершины в графе кратчайших путей строится гамильтонов цикл, проходящий через каждую из $n-1$ вершин, за исключением начальной. Пространство перебора составляют построенные гамильтоновы циклы, число которых не превышает $n*(n-1)$, а лучший из них является оптимальным решением задачи.

Задача поиска гамильтонова цикла в настоящее время не имеет точных алгоритмов решения. Тестирование нового метода показало, что он находит гамильтонов цикл минимальной длины и имеет сложность порядка $O(n^4)$.

Список литературы

1. Акимов О. Е. *Дискретная математика. Логика, группы, графы, фракталы*. Москва : Издатель АКИМОВА, 2005. 656 с.
2. Емеличев В. А., Ковалев М. М., Кравцов М. К. *Многогранники, графы, опитимизация*. Москва : Наука, 1981. 341 с.
3. Гери М., Джонсон Д. *Вычислительные машины и труднорешаемые задачи*. Москва : Мир, 1982. 416 с.
4. Дойбер В.А., Косточка А. В., Закс Х. Более короткое доказательство теоремы Дирака о числе ребер в хроматически критических графах. *Дискретный анализ и исследование операций*. Новосибирский гос.ун-т, 1996. С. 28-34.
5. Оре О. *Теория графов*. Москва : Наука, 1980. 336 с.
6. *Гамильтонов_граф*. URL: https://ru.wikipedia.org/wiki/Гамильтонов_граф (дата обращения : 4 октября 2019).
7. Павленко В. Б. Теоретические аспекты построения гамильтонова цикла. *Теорія оптимальних рішень*. 2011, №10. – С. 150-155. URL: http://dspace.nbuv.gov.ua/bitstream/handle/123456789/46787/22-Pavlenko.pdf?sequence=1 (дата обращения : 4 октября 2019).
8. Прокопенков В. Ф., Кожин Ю. Н., Малых О. Н. Определение оптимального кольцевого маршрута, проходящего через заданное множество пунктов на карте. *Innovative technologies and scientific solutions for industries*. 2019. No.1 (7). С. 102-112. doi.org/10.30837/2522-9818.2019.7.102
9. Харари Ф. *Теория графов*. Москва : Мир, 1973. 300 с.
10. Стивенс Р. *Алгоритмы. Теория и практическое применение*. Москва: Эксмо, 2016. 544 с.
11. Bellman R. Dynamic Programming Treatment of the Travelling Salesman Problem. *Journal of the ACM*. 1962. Vol.9, № 1. P. 61-63. doi.org/10.1145/321105.321111
12. Held M. The travelling-salesman problem an minimum spanning trees. *Operations Research*. 1970. Vol. 18, № 6. P. 1138–1162. doi.org/10.1287/opre.18.6.1138
13. Roberts S. M., Flores B. An engineering approach to the travelling salesman problem. *Managment Science*. 1967. Vol. 13, № 3. P. 269-288. doi.org/10.1287/mnsc.13.3.269
14. Little J. D. C. An Algorithm for the Traveling Salesman Problem. *Operations Research*. 1963. Vol. 11, № 6. P. 972–989. doi.org/10.1287/opre.11.6.972
15. *Муравьиный алгоритм*. URL: https://ru.wikipedia.org/wiki/Муравьиный_алгоритм. (дата обращения : 4 октября 2019).
16. Pol R., Langdon W. B., McPhee N. F. A Field Guide to Genetic Programming. *Genetic Programming and Evolvable Machines*. 2009. Vol. 10, №2. P. 229 – 230. doi.org/10.1007/s10710-008-9073-y.
17. Прокопенков В. Ф. Модификация генетического алгоритма поиска гамильтонова цикла на графе. *Інформаційні технології: наука, техніка, технологія, освіта, здоров'я: Тези доповідей XXIV міжнародної науково-практичної конференції* / за ред. проф. Сокола Є.І. Харків, НТУ «ХПІ», 2016. Ч.І. С.32.

18. Прокопенков В. Ф. О возможности нахождения оптимального решения генетическим алгоритмом. *Інформаційні технології: наука, техніка, технологія, освіта, здоров'я: тези доповідей XXV міжнародної науково-практичної конференції MicroCAD-2017* / за ред. проф. Сокола Є.І. Харків: НТУ «ХПІ», 2017. Ч. І. С.37.
19. Мартынов А. В., Курейчик В. М. Гибридный алгоритм решения задачи коммивояжера. Известия ЮФУ. *Технические науки*. 2015. С. 36-44.
20. Eppstein D. The travelling salesman problem for cubic graphs. *Lecture Notes in Computer Science*. 2003. P.307-318. doi.org/10.1007/978-3-540-45078-8_27
21. Iwama K., Nakashima T. An Improved exact algorithm for cubic graph TSP. *Lecture Notes in Computer Science*. P. 108 – 117. doi.org/10.1007/978-3-540-73545-8_13.
22. Dijkstra E. W. A note on two problems in connexion with graphs. *Numer. Math – Springer Science+Business Media*. 1959. Vol. 1, № 1. P. 269–271. doi.org/10.1007/bf01386390
23. Прокопенков В. Ф., Кожин Ю. Н. Новый подход поиска оптимального решения в переборных NP-задачах. *Інформаційні технології: наука, техніка, технологія, освіта, здоров'я: тези доповідей XXVII міжнародної науково-практичної конференції MicroCAD-2019* / за ред. проф. Сокола Є.І. Харків: НТУ «ХПІ». 2019. Ч. І. С. 38.
24. Прокопенков В. Ф. Параллельный алгоритм поиска гамильтонова цикла на графе. *Інформаційні технології: наука, техніка, технологія, освіта, здоров'я: Тези доповідей XXIII Міжнародної науково-практичної конференції MicroCAD-2015* / за ред. проф. Сокола Є.І. Харків, НТУ «ХПІ». 2015. Ч. І. .25.

References (transliterated)

1. Akimov O. E. *Diskretnaja matematika. Logika, gruppy, grafy, fraktaly* [Discrete mathematics. Logic, groups, graphs, fractals.]. Moscow, Publisher AKIMOV, 2005. 656 p.
2. Emelichev V. A., Kovalev M. M., Kravcov M. K. *Mnogogranniki, grafy, opitimizacija* [Polyhedra, graphs, opitimization]. Moscow, Nauka Publ., 1981. 341 p.
3. Hery M., Dzhonson D. *Vychyslytel'nye mashyny y trudnoreshaemye zadachy* [Computational machines and difficult tasks]. Moscow, Myr. 419 p.
4. Doiber V. A., Kostochka A. V., Sachs H. Bolee korotkoe dokazatel'stvo teoremy Diraka o chisle reber hromaticheski kriticheskih grafah [A shorter proof of Dirac's theorem on the number of edges of chromatically critical graphs]. D*iskretnyj analiz i issledovanie operacij* [Discrete analysis and operations research]. Novosibirsk state University, 1996, pp. 28-34.
5. Ore O. *Teoryya hrafov* [The theory of graphs]. Moscow, Nauka Publ., 1980. 336 p.
6. *Gamil'tonov_graf*. URL: https://ru.wikipedia.org/wiki/Гамильтонов_граф. (accessed: 04.10.2019)
7. Pavlenko V. B. Teoreticheskie aspekty postroenija gamil'tonova cikla [Theoretical aspects of construction of the Hamiltonian cycle] *Teorija optimal'nih rishen'* [Theory of optimal solutions]. 2011, no. 10, pp. 150-155. URL : http://dspace.nbuv.gov.ua/bitstream/-handle/123456789/46787/22-Pavlenko.pdf?sequence=1 (accessed 04.10.2019).
8. Prokopenkov V. F., Kozhin Ju., N., Malyh O. N. Opredelenie optimal'nogo kol'cevogo marshruta, prohodjashhego cherez zadannoe mnozhestvo punktov na karte. [Determination of the optimal circular route passing through a given set of points on the map]. *Innovative technologies and scientific solutions for industries*. 2019, no. 1 (7). pp. 102-112. doi.org/10.30837/2522-9818.2019.7.102
9. Harari F. *Teorija grafov* [Graph theory]. Moscow, Myr, 1973. 300 p.
10. Stivens R. *Algoritmy. Teorija i prakticheskoe primenenie* [Algorithms. Theory and practical application]. Eksmo, Moscow, 2016, 544 p.
11. Bellman R. Dynamic Programming Treatment of the Travelling Salesman Problem. *Journal of the ACM*. 1962, vol.9, no. 1, pp. 61-63. doi.org/10.1145/321105.321111
12. Held M. The travelling-salesman problem an minimum spanning trees. *Operations Research*. 1970, vol. 18, no. 6, pp. 1138 – 1162. doi.org/10.1287/opre.18.6.1138
13. Roberts S. M., Flores B. An engineering approach to the travelling salesman problem. *Managment Science*. 1967, vol. 13, no. 3, pp. 269-288. doi.org/10.1287/mnsc.13.3.269
14. Little J. D. C. An Algorithm for the Traveling Salesman Problem. *Operations Research*. 1963, vol. 11, no. 6, pp. 972–989. doi.org/10.1287/opre.11.6.972
15. *Murav'inyj algoritm* [Ant algorithm]. URL: https://ru.wikipedia.org/wiki/Муравьиный_алгоритм. (accessed: 04.10.2019)
16. Pol R., Langdon W. B., McPhee N. F. A Field Guide to Genetic Programming. *Genetic Programming and Evolvable Machines*. 2009, vol. 10, no. 2, pp. 229 – 230. doi.org/10.1007/s10710-008-9073-y.
17. Prokopenkov V. F. Modifikacija geneticheskogo algoritma poiska gamil'tonova cikla na grafe [Modification of a genetic algorithm for finding a Hamiltonian cycle on a graph]. *International Scientific Conference MicroCAD 2016: Section No. 1 – Information and Management Systems*. 2016, pp. 32.
18. Prokopenkov V. F. O vozmozhnosti nahozhdenija optimal'nogo reshenija geneticheskim algoritmom [On the possibility of finding the optimal solution by genetic algorithm]. *International Scientific Conference MicroCAD 2017: Section No. 1 – Information and Management Systems*. 2017, pp. 37.
19. Martynov A. V., Kurejchik V. M. Gibridnyj algoritm reshenija zadachi kommivojazhera [Hybrid algorithm for solving the traveling salesman problem]. *Izvestija JuFU. Tehnicheskie nauki* [SFU news. Technical science]. 2015. pp.36-44.
20. Eppstein D. The travelling salesman problem for cubic graphs. *Lecture Notes in Computer Science*. 2003, pp. 307-318. doi.org/10.1007/978-3-540-45078-8_27
21. Iwama K., Nakashima T. An Improved exact algorithm for cubic graph TSP. *Lecture Notes in Computer Science*, pp. 108 – 117. doi.org/10.1007/978-3-540-73545-8_13.
22. Dijkstra E. W. A note on two problems in connexion with graphs. *Numer. Math, Springer Science+Business Media*. 1959, vol. 1, no 1. pp. 269–271. doi.org/10.1007/bf01386390
23. Prokopenkov V. F., Kozhin Ju. N. Novyj podhod poiska optimal'nogo reshenija v perebornyh NP-zadachah [A new approach to finding the optimal solution in iterative NP-problems]. *International Scientific Conference MicroCAD 2019: Section No. 1 – Information and Management Systems*. 2019, pp. 38.
24. Prokopenkov V. F. Parallelniy algoritm poiska gamiltonova cikla na grafe [A parallel algorithm for finding a Hamiltonian cycle on a graph]. *International Scientific Conference MicroCAD 2015: Section No. 1 – Information and Management Systems*. 2015, pp. 25.

Поступила (received) 27.11.2019

Відомості про авторів / Сведения об авторах / About the Authors

Прокопенков Владимир Филиппович **(*Прокопенков Володимир Пилипович, Prokopenkov Vladymyr Fylyppovych*)** – Национальный технический университет "Харьковский политехнический институт", старший преподаватель кафедры "Системный анализ и информационно-аналитические технологии", Харьков, Украина; e-mail: prokopenkov.vf@gmail.com.; ORCID: https://orcid.org/0000-0003-0084-9832.