УДК 519.1 DOI: 10.20998/2413-3000.2022.5

В. П. ПРОКОПЕНКОВ

РОЗРОБКА МЕТОДУ ВІДКЛАДЕНИХ РІШЕНЬ ДЛЯ ПОБУДОВИ АЛГОРИТМУ ПОШУКУ ГАМІЛЬТОНОВА ЦИКЛУ НА ГРАФІ

Предмет досліджень – вирішення задачі пошуку гамільтонова циклу на графі, яка відноситься до NP класу складності. Мета роботи – розробка ефективного поліноміального алгоритму її оптимального вирішення. У роботі виконано аналіз проблеми та існуючих методів її вирішення, визначено недоліки цих методів. Показано, що головною перешкодою залишається неможливість сформулювати умови знаходження оптимального рішення. Як наслідок, застосовувані для вирішення цієї задачі методи засновані на переборі допустимих рішень або на інтуїтивних евристиках. Евристичні методи не гарантують відшукання оптимального рішення. Перебірні методи популярні через нескладну лінійну схему пошуку в множині допустимих рішень задачі. Вони дозволяють знайти оптимальне рішення, але вимагають великих витрат часу. У перебірних алгоритмах допустимі рішення можна отримувати використовуючи алгоритми обходу графа, але факторіальні витрати на перебір вимагають скорочення простору перебору, наприклад, використовуючи метод гілок і границь. Цей метод заснований на впорядкованому переборі допустимих рішень, розгляді тільки перспективних з них і відкиданні відразу цілих множин рішень, які не є такими. Для роботи методу важливо визначити функцію вартості часткового рішення, що залежить від певних параметрів, що важко, а може і неможливо для даної задачі. Якщо функція формує ймовірну оцінку, при відкиданні існує ризик втратити оптимальне рішення задачі. Єдиною надійною оцінкою для допустимого рішення залишається довжина циклу, яка, на жаль, стає відомою після його формування. Як альтернатива в статті пропонується новий метод відкладених рішень, згідно з яким одночасно будуються і зберігаються усі можливі часткові рішення. Кожне часткове рішення характеризується своєю оцінкою. На кожному кроці часткове рішення добудовується додаванням до нього вершини, в яку можна перейти з його останньої вершини – будується стільки нових часткових рішень, скільки існує варіантів переходу з його останньої вершини. Сформовані часткові рішення зберігаються, а поточне відпрацьоване рішення видаляється. Для виконання наступного кроку вибирається то часткове рішення, яке має найменшу оцінку довжини. Виконання триває до побудови оптимального рішення. Запропонований метод розв’язує дану задачу, але його застосування для графів великої розмірності вимагає підбору правильної оцінки часткових рішень.

Ключові слова: граф, гамільтонів цикл, складність, NP-повнота, простір перебору, перебірний алгоритм, скорочення перебору, метод відкладених рішень.

В. Ф. ПРОКОПЕНКОВ

РАЗРАБОТКА МЕТОДА ОТЛОЖЕННЫХ РЕШЕНИЙ ДЛЯ ПОСТРОЕНИЯ АЛГОРИТМА ПОИСКА ГАМИЛЬТОНОВА ЦИКЛА НА ГРАФЕ

Предмет исследований – решение задачи поиска гамильтонова цикла на графе, которая относится к NP классу сложности. Цель работы – разработка эффективного полиномиального алгоритма её оптимального решения. В работе выполнен анализ проблемы и существующих методов её решения, определены недостатки этих методов. Показано, что главным препятствием остаётся невозможность сформулировать условия нахождения оптимального решения. Как следствие, применяемые для решения этой задачи методы основаны на переборе допустимых решений или на интуитивных эвристиках. Эвристические методы не гарантируют отыскания оптимального решения. Переборные методы популярны из-за простой линейной схемы поиска в заранее известном множестве допустимых решений задачи. Они позволяют найти оптимальное решение, но требуют больших затрат времени. В переборных алгоритмах допустимые решения можно получать используя алгоритмы обхода графа, но факториальные затраты на перебор требуют сокращения пространства перебора, например, используя метод ветвей и границ. Этот метод основан на упорядоченном переборе допустимых решений, рассмотрении только перспективных из них и отбрасывании сразу целых множеств решений, которые не являются такими. Для работы метода важно определить функцию стоимости частичного решения, зависящую от определённых параметров, что затруднительно, а может и невозможно для рассматриваемой задачи. Если функция формирует вероятностную оценку, при отбрасывании существует риск потерять оптимальное решение задачи. Единственной надёжной оценкой для допустимого решения остаётся длина цикла, которая, к сожалению, становится известной после его формирования. Как альтернатива в статье предлагается новый метод отложенных решений, согласно которому одновременно строятся и сохраняются все возможные частичные решения. Каждое частичное решение характеризуется своей оценкой. На каждом шаге частичное решение достраивается добавлением к нему вершины, в которую можно перейти из его последней вершины – строится столько новых частичных решений, сколько существует вариантов перехода из его последней вершины. Сформированные частичные решения сохраняются, а текущее отработанное решение удаляется. Для выполнения следующего шага выбирается то частичное решение, которое имеет наименьшую оценку длины. Выполнение продолжается до построения оптимального решения. Предложенный метод решает рассматриваемую задачу, но его применение для графов большой размерности требует подбора правильной оценки частичных решений.

Ключевые слова: граф, гамильтонов цикл, сложность, NP-полнота, пространство перебора, переборный алгоритм, сокращение перебора, метод отложенных решений.

V. PROKOPENKOV

DEFERRED SOLUTIONS METHOD DEVELOPMENT FOR CONSTRUCTING A HAMILTONIAN CYCLE SEARCH ALGORITHM ON A GRAPH

The subject of research is the solution of the problem of finding a Hamiltonian cycle on a graph, which belongs to the NP complexity class. The aim of the work is to develop an effective polynomial algorithm for its optimal solution. The paper analyzes the problem and the existing methods of its solution, identifies the shortcomings of these methods. It is showing that the main obstacle remains the inability to formulate the conditions for finding the optimal solution. As a result, the methods for solving this problem based on enumeration over acceptable solutions or on intuitive heuristics. Heuristic methods do not guarantee finding the optimal solution. Enumeration methods are popular because of a simple linear search scheme in a pre-known set of valid solutions to the problem. They allow you to find the optimal solution, but require a lot of time. In enumeration algorithms, valid solutions can be obtain by using graph traversal algorithms, but the factorial cost of enumeration requires reducing the enumeration space, for example, using the branch-and-bound method. This method is based on an ordered search of acceptable solutions, considering only the most promising ones, and discarding at once the whole sets of solutions that are not such. For the method to work, it is important to determine the cost function of a partial solution that depends on certain parameters, which is difficult, and perhaps impossible for the problem under consideration. If the function generates a probabilistic estimate, there is a risk of losing the optimal solution to the problem when it is discarding. The only reliable estimate for a valid solution is the length of the cycle, which, unfortunately, becoming known after its formation. As an alternative, the article proposes a new method of deferred solutions, according to which all possible partial solutions are constructed and stored simultaneously. Each partial solution is characterizing by its own evaluation. At each step, a partial solution is completing by adding a vertex to it, to which you can go from its last vertex – as many new partial solutions are building, as there are options for going from its last vertex. The generated partial solutions are saving, and the current worked-out solution is deleting. To perform the next step, the partial solution that has the smallest length estimate is selecting. Execution continues until the optimal solution is not build. The proposed method solves the problem under consideration, but its application for large graphs requires the selection of a correct estimate of partial solutions.

Keywords: graph, Hamiltonian cycle, complexity, NP-completeness, enumeration space, enumeration algorithm, enumeration reduction, deferred solutions method.

Вступ. Задача пошуку замкнутого шляху на графі відома як гра «кругосвітня подорож» на додекаедрі, запропонована В. Гамільтоном [1], на честь якого замкнутий шлях у графі називається гамільтоновим циклом [2]. Граф, що містить гамільтонів цикл, називається гамільтоновим графом. Пошук гамільтонова циклу у графі, як і перевірка, чи є граф гамільтоновим – задачі, що відносяться до NP класу складності [3]. NP складність обумовлена розміром простору можливих рішень задачі. У графі з n вершин необхідно перебрати $(n-1)!$ можливих рішень.

Усі методи вирішення задачі пошуку гамільтонова циклу можна поділити на евристичні та перебірні. Евристичні методи будуються на інтуїтивних евристиках, вони придатні за витратами часу, але не гарантують оптимальності одержуваного рішення. Перебірні методи використовують лінійну схему пошуку рішення у заздалегідь відомій множині допустимих рішень задачі, але велика розмірність цієї множини призводить до неприпустимо великого часу пошуку.

Проблема полягає в неможливості сформулювати умови знаходження оптимального рішення. У перебірних алгоритмах допустимі рішення можна формувати використовуючи алгоритми обходу графа, а для зменшення часових витрат вдаватися до скорочення простору перебору, використовуючи метод гілок і границь [4]. Цей метод реалізує спрямований перебір допустимих рішень, при якому розглядаються тільки перспективні для отримання оптимуму часткові рішення і відкидаються цілі множини рішень, які не є такими.

Для роботи перебірних методів, що використовують пошук з відкиданням неперспективних рішень, важливо визначити функцію вартості часткових рішень, що залежить від певних параметрів, що буває важко або неможливо. Якщо функція, що формує оцінку, ненадійна або формує імовірнісне значення оцінки, при відкиданні існує ризик втратити оптимальне рішення. Для даної задачі єдиною надійною оцінкою рішення залишається довжина циклу, але, на жаль, вона стає відомою після повного його формування.

Рішення даної задачі як і раніше актуальне для практичних завдань виробництва і для розвитку науки в цілому, тому дослідження в цій галузі тривають.

Аналіз останніх публікацій. У оптимальній постановці задачі пошуку гамільтонова циклу на графі необхідно знайти гамільтонів цикл мінімальної довжини. Граф може бути не гамільтоновим, для перевірки можна використовувати умови, розглянуті в [5-7], але вони не завжди можуть бути застосовні для графів довільної структури [8] для різних задач, наприклад як [9]. Якщо ми знаємо, що граф гамільтонів, у будь-якому випадку рішення необхідно шукати окремо.

Ще кілька років тому справедливо було сказати, що не існує поліноміальних алгоритмів для вирішення задачі в оптимальній постановці [10], а всі алгоритми класифікувалися наступним чином.

Якщо алгоритм, що реалізує певну логіку вирішення задачі, достовірно гарантує отримання оптимального рішення, він називається точним, в іншому випадку евристичним [11]. Перевагою евристичних методів є їх приналежність до P класу складності. У своїй основі ці методи мають логічно обґрунтовані ідеї – евристики, наприклад, мурашиний алгоритм, генетичні та гібридні [12-16]. Вони задовільні за витратами часу, але не гарантують отримання оптимального рішення, а також відрізняються складністю підбору параметрів налаштування. Дослідження тривають, з'являються нові рішення, наприклад, для кубічних графів в [17] було запропоновано рішення складності 1.26^n, а в [18] складність рішення знижена до 1.251^n.

До групи точних методів вирішення даної задачі відносяться методи динамічного програмування складністю 2^n [19-20] і всі перебірні алгоритми експоненціальної складності. Допустимі рішення в перебірних алгоритмах отримують комбінаторними методами, для яких гамільтонів цикл розглядається як перестановка вершин, або використанням алгоритмів обходу графа, наприклад, алгоритм Робертса і Флореса [21]. Великі витрати на перебір вимагають скорочення простору перебору, що можна досягти використанням метода гілок і границь [4] складності $n*\log_2 n$ для відкидання свідомо не оптимальних рішень, але без втрати оптимального.

У роботах [22-23] представлені точний метод і алгоритм вирішення задачі складності n^4. Пошук оптимального рішення задачі зводиться до пошуку замкнутого шляху в новому графі найкоротших шляхів, який будується на основі вихідного графа задачі з використанням алгоритму Дейкстри. Обґрунтування такого рішення логічно – щоб гамільтонів цикл мав найменшу довжину, необхідно щоб він складався з найкоротших шляхів вихідного графа. Розроблений алгоритм гарантує відшукання оптимального рішення і значно скорочує початковий простір пошуку до множини перебору з потужністю $n*(n\text{-}1)$. І все ж, в основі методу і алгоритму [22-23] як і раніше лежить евристика і оптимальне рішення не обчислюється, а відшукується методом перебору, але в значно скороченій за розміром множині перебору по відношенню до початкового простору можливих рішень. Крім того, хоч новий алгоритм і є поліноміальним, ступінь 4 досить великий і при великій кількості вершин вихідного графа виконання алгоритму займає час незадовільний для систем реального часу.

В результаті вивчення стану проблеми можна зробити висновок, що причиною невдачі в розробці ефективного методу є неможливість сформулювати умови знаходження оптимального рішення задачі. Як наслідок, основним способом вирішення як і раніше залишається перебір всіх або більшої частини допустимих рішень і використання інтуїтивних евристичних прийомів.

Мета роботи. Поставлена в роботах [22-23] мета – розробки поліноміального алгоритму пошуку гамільтонова циклу мінімальної довжини досягнута, але виконання алгоритму для графів з великою кількістю вершин займає час незадовільний для систем реального часу. Тому, метою роботи є розробка нового алгоритму вирішення даної задачі з кращими показниками часу.

Постановка задачі. Нехай заданий граф $G = \langle V,E \rangle$, де $V = \left\{ v_i \mid i = \overline{1,n} \right\}$ – це множина вершин, а $E = \left\{ e_{ij} \mid i,j = \overline{1,n}, i \neq j \right\}$ – множина дуг графа. Дуга e_{ij} визначає наявність з'єднання між вершинами v_i та v_j, характеризується відстанню d_{ij}. Нехай задана початкова вершина $v_s \in V$.

Необхідно знайти гамільтонів цикл мінімальної довжини з вершини v_s, тобто кортеж $GC =< v_1, v_2, ..., v_k, ...v_{n-1}, v_n, v_{n+1} >$ з вершин графа *G*, для якого виконуються умови:

1) $v_1 = v_s$;

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

3) для будь-якої пари вершин $v_i, v_j \mid i,j \in \{\overline{1,n}\}$ справедливо: якщо $i \neq j$, то $v_i \neq v_j$;

4) для $\forall v_k \mid k \in \{\overline{2,n}\}$ у графі G існують дуги: $e_{k-1,k}$ – з вершини v_{k-1} до вершини v_k та $e_{k,k+1}$ – з вершини v_k до вершини v_{k+1}.

Метод вирішення задачі. Рішення даної задачі методом перебору передбачає можливість формування кожного рішення з простору можливих рішень, оцінку рішення і відкидання рішення, якщо воно не оптимальне. Для досягнення поставленої мети – підвищення швидкодії перебірного алгоритму, необхідно прискорити процес формування і обробки допустимих рішень і при відкиданні рішень не втратити оптимальне рішення.

У моделі задачі, що подається графом, формування кожного рішення здійснюється обходом графа, що обумовлює покрокову схему формування кожного рішення. Формування рішень має бути безперервним, щоб забезпечити послідовний перебір всього простору можливих рішень.

При обході графа і формуванні рішення, переходячи від вершини до вершини по дугах, важко, в силу не сформульованих умов визначення оптимального рішення, передбачити в якому напрямку необхідно йти, щоб його отримати. Саме тому, найвдалішою схемою обходу графа є схема з поверненнями [21]. Ця схема дозволяє обійти граф повністю, побудувати усі рішення з простору можливих рішень і не пропустити оптимального. На жаль, оцінка розміру простору перебору становить $(n-1)!$ і для великих значень n вона перекреслює усілякі надії на швидке вирішення задачі.

Щоб прискорити процес перебору при збереженні покрокової схеми формування кожного рішення необхідно:

- відмовитися від повної побудови всіх рішень;
- замінити послідовну схему формування рішень на «паралельну»;
- забезпечити можливість відкидання неперспективних рішень;
- виключити можливість втрати оптимального рішення при відкиданні;
- забезпечити можливість повернення до відкладених рішень при необхідності.

Найкраще для досягнення поставленої мети підходить схема відкладених рішень, представлена в [24]. У цій схемі рішення будуються крок за кроком, одночасно і послідовно, як це не здається суперечливим.

У процесі побудови, поки рішення не стало гамільтоновим циклом, воно називається частковим рішенням. Спочатку часткове рішення включає одну початкову вершину.

Кожне рішення будується крок за кроком. На кожному кроці поточне часткове рішення, що розглядається, добудовується додаванням до нього вершини, в яку можна перейти з останньої вершини поточного рішення.

Усі можливі рішення будуються одночасно. Одночасність треба розуміти не в сенсі паралельної обробки на декількох процесорах, а як одночасну побудову множини можливих рішень в процесі пошуку рішення задачі в цілому. Це означає, що при розгляді поточного часткового рішення будується стільки нових часткових рішень, скільки у графі існує варіантів переходу з його останньої вершини в інші вершини.

Часткові рішення будуються послідовно і зберігаються у пам'яті, якщо вони теоретично можуть бути добудовані до повного рішення. Поточне відпрацьоване рішення видаляється з пам'яті.

Відкидання неперспективних рішень в цій схемі розуміється не буквально, а як відкладення цих рішень з можливим поверненням до них знову і реалізується шляхом відмови на поточному кроці продовжувати добудову таких часткових рішення до повних рішень.

Для можливості реалізувати вибір перспективного рішення, яке стане поточним, кожне часткове рішення характеризується своєю оцінкою, яка, наприклад, може бути пов'язана з довжиною вже побудованого шляху в частковому рішенні.

Для виконання наступного кроку алгоритму вибирається те побудоване часткове рішення, яке на поточний момент за оцінкою має найбільші шанси дорости до оптимального рішення.

Виконання схеми можна зупинити після побудови першого рішення задачі. Чи буде отримане рішення оптимальним, залежить від можливості визначити правильну оцінку для порівняння часткових рішень.

Запропонована схема має один істотний недолік – значні витрати на зберігання часткових рішень, але такі витрати пам'яті повинні бути виправдані істотним скороченням часу на пошук оптимального рішення.

Якщо описану проблему формування оцінки часткових рішень вдасться вирішити, можна вважати, що поставлена нами мета буде досягнута.

Для розробки алгоритму виконаємо необхідну формалізацію і обґрунтуємо описаний метод вирішення задачі.

Обґрунтування методу розв'язання задачі. Часткове рішення $PS = < pc, d_{pc} >$ визначається парою елементів: pc – шляху в графі *G*, що визначає часткове рішення та $d_{pc} = | pc |$ – довжини цього шляху. Часткове рішення як шлях pc представляє собою кортеж з вершин графа G – $pc = < v_1, v_2, ..., v_k, ...v_K >$, $K \le n+1 >$, які утворюють шлях в графі *G*, що задовольняє умовам:

1) $v_1 = v_s$, в початковому стані $pc = < v_1 >$, $v_1 = v_s$.

2) для будь-якої пари вершин $v_i, v_j \mid i, j \in \{\overline{1, K}\}$ справедливо: якщо $i \ne j$, то $v_i \ne v_j$;

3) для $\forall v_k \mid k \in \{\overline{2, K-1}\}$ у графі *G* існують дуги: $e_{k-1,k}$ – з вершини v_{k-1} до вершини v_k та $e_{k,k+1}$ – з вершини v_k до вершини v_{k+1}.

4) якщо $K = n+1$ і $v_K = v_s$, то pc стає повним рішенням $gc = < v_1, v_2, ..., v_k, ...v_{n-1}, v_n, v_{n+1} >$.

Довжина часткового шляху $pc = < v_1, v_2, ..., v_k, ...v_K >$, $K \le n+1 >$ визначається як сума ваг $d_{k,k+1}$ відповідних дуг графа $e_{k,k+1}$, зумовленими вершинами в частковому шляху: $| pc | = \sum_{k=1}^{K-1} d_{k,k+1}$.

Для опису графа *G* будемо використовувати списки послідовників його вершин. Для кожної вершини $v_i \in V$ поставимо у відповідність множину $Suc[i] = \{ j \mid \exists e_{i,j} \in E \}$, яка називається списком послідовників вершини v_i, вона включає такі номери $j \mid j \le n$, що визначають вершини v_j графа *G*, до яких можна перейти з вершини v_i. Тоді $\{Suc[i] \mid i = \overline{1, n}\}$ описує граф *G* списками послідовників його вершин.

Для зберігання часткових рішень як структуру даних будемо використовувати чергу відкладених рішень, у якій часткові рішення розташовуються у порядку убування обраної оцінки часткових рішень. Для реалізації методу розв'язання задачі необхідно реалізувати операції над чергою:

- додавання часткового рішення з урахуванням його оцінки, $PQ.Add(PS)$;

- діставання чергового (першого в черзі) часткового рішення для обробки з видаленням його з черги, $PS = PQ.Remove\ (\ )$.

Поточне оброблюване рішення будемо позначати $CurPS$.

Алгоритм розв'язання задачі. Вхідними даними алгоритму, що реалізує метод відкладених рішень є опис графа у формі списків послідовників його вершин і номер початкової вершини. Результатом виконання алгоритму є знайдений гамільтонів цикл *GC*.

Алгоритм.

1. Для кожної вершини j, що належить до $Suc[s]$, списку послідовників початкової вершини s виконати:

1.1 Сформувати початкове часткове рішення $PS_{sj} = < pc, | pc | > = << s >, 0 >$.

1.2 Продовжити шлях у частковому рішенні PS_{sj} вершиною v_j: $pc = < s, j >$.

1.3 Скорегувати довжину часткового рішення PS_j: $| pc | = | pc | + d_{s,j}$.

1.4 Додати часткове рішення PS_{sj} в чергу відкладених рішень: $PQ.Add(PS_{sj})$.

2. Виконувати цикл поки черга відкладених рішень не порожня:

2.1 Визначити поточне часткове рішення для обробки: $CurPS = PQ.Remove\ (\ )$.

2.2 Визначити номер K останньої вершини шляху часткового рішення $CurPS = < pc, | pc | >$.

2.3 Для кожної вершини j, що належить до $Suc[K]$ виконати:

2.3.1 Якщо $j \notin pc$ або $j = s$, сформувати часткове рішення $PS_{Kj} = < pc, | pc | >$:

$pc = pc + j$ – продовжити шлях в частковому рішенні PS_{Kj} вершиною v_j;

$| pc | = | pc | + d_{K,j}$ – скорегувати довжину часткового рішення PS_{Kj}.

2.3.2 Якщо $j = s$ та $K+1 = n$, покласти $GC = PS_{Kj}$, рішення знайдено, перейти до п. 3.

2.3.3 Додати часткове рішення PS_{Kj} до черги відкладених рішень: $PQ.Add(PS_{Kj})$.

3. Зупинитися.

Тестування схеми. Очевидно, що реалізована представленим алгоритмом обчислювальна схема не вимагає доказу коректності. Згідно зі схемою, алгоритм повинен знайти єдине оптимальне рішення.

Як тест було використано не повний граф – додекаедр, опис якого було представлено в [25]. Для пошуку рішення у якості оцінки часткових рішень була обрана реальна довжина шляху (чим менше довжина шляху часткового рішення, тим вище його оцінка), що цілком логічно, так як шуканий гамільтонів цикл повинен мати найменшу довжину.

На рис. 1. представлено оптимальне рішення (7227)<0,4,5,12,11,7,6,19,18,8,9,10,14,13,3,2,15,16,17,1, 0> було знайдено за 0,005 хвилини.

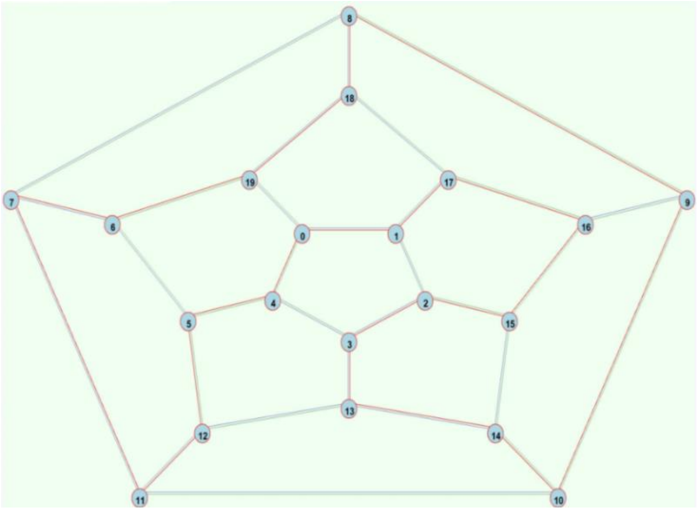

Рис. 1. Знайдене оптимальне рішення

Тестування запропонованої схеми на повному графі з 20 вершин потребувало часу, порівнянного з перебірною схемою всіх допустимих рішень цього графа. Таким чином, тестування показало, що запропонована схема при використанні у якості оцінки реальної довжини шляху дозволяє знаходити оптимальне рішення для неповного графа, але терпить фіаско перед повним графом великої розмірності.

Виникає питання, чи можна виправити зазначений недолік запропонованої схеми вибором іншої оцінки часткових рішень?

Висновок. У статті представлені новий метод і алгоритм пошуку гамільтонова циклу на графі, який показав відмінний результат для неповного графа, що складається з 20 вершин. Успішне застосування нового методу для повного графа великої розмірності вимагає ретельного його аналізу і, як передбачається, залежить від вибору кращої чим використана оцінки часткових рішень.

Список літератури

1. Акимов О. Е. *Дискретная математика. Логика, группы, графы, фракталы.* Москва, 2005. 656 с.
2. Емеличев В. А., Ковалев М. М., Кравцов М. К. *Многогранники, графы, оптимизация*. Москва. 1981. 341 с.
3. Гери М., Джонсон Д. *Вычислительные машины и труднорешаемые задачи*. Москва, 1982. 416 с.
4. 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
5. 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.
6. Оре О. *Теория графов.* Москва, 1980. 336 с.
7. *Гамильтонов граф:* сайт. – URL : https://ru.wikipedia.org/wiki/Гамильтонов_граф. (дата обращения : 4.10.2019).
8. Павленко В. Б. Теоретические аспекты построения гамильтонова цикла. *Теорія оптимальних рішень*. 2011, №10. с. 150–155. сайт . URL : http://dspace.nbuv.gov.ua/bitstream/handle/123456789/46787/22-Pavlenko.pdf?sequence=1 (дата обращения : 4.10.2019).
9. 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
10. Харари Ф. *Теория графов.* Москва, 1973. 300 с.
11. Стивенс Р. *Алгоритмы. Теория и практическое применение*. Москва, 2016. 544 с.
12. *Муравьиный алгоритм* : сайт. URL : https://ru.wikipedia.org/wiki/Муравьиный_алгоритм.
13. 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.
14. Прокопенков В. Ф. Модификация генетического алгоритма поиска гамильтонова цикла на графе. *Международная научная конференция MicroCAD 2016 : Секция №1 : Информационные и управляющие системы.* 2016. С. 32.
15. Прокопенков В. Ф. О возможности нахождения оптимального решения генетическим алгоритмом. *Международная научная конференция MicroCAD 2017 : Секция №1 : Информационные и управляющие системы.* 2017. С. 37.
16. Мартынов А. В., Курейчик В. М. Гибридный алгоритм решения задачи коммивояжера. *Известия ЮФУ. Технические науки*. – 2015. С.36–44.
17. 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

18. Iwama K., Nakashima T. An Improved exact algorithm for cubic graph TSP. *Lecture Notes in Computer Science, 2007.* p. 108 – 117. doi.org/10.1007/978-3-540-73545-8_13.
19. 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
20. 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
21. Roberts S. M., Flores B. An engineering approach to the travelling salesman problem. *Management Science*. 1967. Vol. 13 № 3, p. 269–288. doi.org/10.1287/mnsc.13.3.269
22. Прокопенков В. Ф. Новый метод поиска гамильтонова цикла на графе. *Вісник Національного технічного університету «ХПІ». Серія: Стратегічне управління, управління портфелями, програмами та проектами. 2020. № 2, С.43-49.* doi.org/10.20998/2413-3000.2020.2.6
23. Прокопенков В. Ф. Полиномиальный алгоритм поиска гамильтонова цикла на графе. *Вісник Національного технічного університету «ХПІ». Серія: Стратегічне управління, управління портфелями, програмами та проектами. 2021. № 1(3), С.55-65.* doi.org/10.20998/2413-3000.2021.3.8
24. Prokopenkov V.P. Kozhyn Y.N. Deferred solutions scheme for the problem of finding a Hamiltonian cycle solving. *Международная научная конференция MicroCAD 2021 : Секция №1 : Информационные и управляющие системы.* 2021. С. 16.
25. Прокопенков В. Ф. Параллельный алгоритм поиска гамильтонова цикла на графе. *Международная научная конференция MicroCAD 2015 : Секция №1 : Информационные и управляющие системы.* 2015. С. 25.

References (transliterated)

1. Akimov, O. E. (2005) *Discrete mathematics. Logic, groups, graphs, fractals* [Diskretnaja matematika. Logika, gruppy, grafy, fraktaly], Moscow, 656 p.
2. Emelichev, V. A., Kovalev, M. M., Kravcov, M. K. (1981), *Polyhedra, graphs, opitimization* [Mnogogranniki, grafy, opitimizacija*]*, Moscow, 341 p.
3. Hery, M., Dzhonson, D. (1982), *Computational machines and difficult tasks* [Vychyslytel'nye mashyny y trudnoreshaemye zadachy], *Moscow, 419 p*.
4. Little, J. D. C. (1963) An Algorithm for the Traveling Salesman Problem, *Operations Research,* Vol. 11, № 6, p. 972–989. doi.org/10.1287/opre.11.6.972
5. Doiber, V. A., Kostochka, A. V., Sachs, H. "A shorter proof of *Bolee korotkoe dokazatel'stvo teoremy Diraka o chisle reber hromaticheski kriticheskih grafah*. [Dirac's theorem on the number of edges of chromatically critical graphs"]. *Diskretnyj analiz i issledovanie operacij* [Discrete analysis and operations research] Novosibirsk state University, 1996. pp. 28–34.
6. Ore, O. (2009), *Teoryya hrafov* [The theory of graphs], Moscow, 354 p.
7. *Gamil'tonov_graf* [Hamilton's graf], Available at : https://ru.wikipedia.org/wiki/Гамильтонов_граф. (last accessed: 04.10.2019)
8. Pavlenko, V. B. (2011) Teoreticheskie aspekty postroenija gamil'tonova cikla. [Theoretical aspects of construction of the Hamiltonian cycle], *Teorija optimal'nih rishen* ['Theory of optimal solutions], №10, pp. 150–155. Available at : http://dspace.nbuv.gov.ua/bitstream/handle/123456789/46787/22-Pavlenko.pdf?sequence=1 (last accessed 04.10.2019).
9. Prokopenkov, V. F., Kozhin, Ju., N., Malyh, *O. N.* (2019) 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*, No.1 No. 7. pp. 102-112. doi.org/10.30837/2522-9818.2019.7.102
10. Harari, F. (1973), *Teorija grafov* [Graph theory], Moscow, 300 p.
11. Stivens, R. (2016), *Algoritmy. Teorija i prakticheskoe primenenie* [Algorithms. Theory and practical application], Moscow , 544 p.
12. *Murav'inyj algoritm* [Ant algorithm], Available at : https://ru.wikipedia.org/wiki/Муравьиный_алгоритм. (last accessed: 04.10.2019)
13. Pol, R., Langdon, W. B., McPhee, N. F. (2009), A Field Guide to Genetic Programming, *Genetic Programming and Evolvable Machines,* Vol. 10, No. 2, p. 229 – 230. doi.org/10.1007/s10710-008-9073-y.
14. Prokopenkov, V. F. (2016), 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, p. 32.*
15. Prokopenkov, V. F. (2017), 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, p. 37.*
16. Martynov, A. V., Kurejchik, V. M. (2015) Gibridnyj algoritm reshenija zadachi kommivojazhera. [Hybrid algorithm for solving the traveling salesman problem], *SFU news. Technical science* [*Izvestija JuFU. Tehnicheskie nauki*]. p.36–44.
17. Eppstein, D. (2003), The travelling salesman problem for cubic graphs, *Lecture Notes in Computer Science*, p. 307–318. doi.org/10.1007/978-3-540-45078-8_27
18. Iwama, K., Nakashima, T. (2007), 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.
19. Bellman, R. (1962), Dynamic Programming Treatment of the Travelling Salesman Problem, *Journal of the ACM,* Vol.9 No. 1, p. 61–63. doi.org/10.1145/321105.321111
20. Held, M. (1970), The travelling-salesman problem an minimum spanning trees, *Operations Research,* Vol. 18 No. 6, p. 1138 – 1162. doi.org/10.1287/opre.18.6.1138
21. Roberts, S. M., Flores, B. (1967), An engineering approach to the travelling salesman problem, *Managment Science*, Vol. 13 No. 3, p. 269–288. doi.org/10.1287/mnsc.13.3.269
22. Prokopenkov, V. F. (2020), Novyy metod poiska gamil'tonova tsikla na grafe [A new method for searching a Hamilton cycle on a graf"], *Visnyk Natsionalnoho tekhnichnoho universytetu «KhPI». Seriia: Stratehichne upravlinnia, upravlinnia portfeliamy, prohramamy ta proektamy.* [Bulletin of the National Technical University "KHPI". Series: Strategic Management, portfolio management, programs and projects], № 2, pp.43-49. doi.org/10.20998/2413-3000.2020.2.6
23. Prokopenkov, V. F. (2021), Polynomial algorithm for finding a Hamiltonian cycle on a graph. [Polinomial'nyy algoritm poiska gamil'tonova tsikla na grafe], *Visnyk Natsionalnoho tekhnichnoho universytetu «KhPI». Seriia: Stratehichne upravlinnia, upravlinnia portfeliamy, prohramamy ta proektamy.* [Bulletin of the National Technical University "KHPI". Series: Strategic Management, portfolio management, programs and projects], No. 1(3), pp.55-65. doi.org/10.20998/2413-3000.2021.3.8
24. Prokopenkov V.P. Kozhyn Y.N. (2021), Deferred solutions scheme for the problem of finding a Hamiltonian cycle solving. *International Scientific Conference MicroCAD 2021 : Section No. 1 – Information and Management Systems, p.* 16.
25. Prokopenkov, V. F. (2015), 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, p. 25.*

Надійшла (received) 20.01.2022

Відомості про авторів / Сведения об авторах / About the Authors

Прокопенков Володимир Пилипович (Прокопенков Владимир Филиппович, Prokopenkov Vladymyr) – Національний технічний університет "Харківський політехнічний інститут", старший викладач кафедри системний аналіз та інформаційно-аналітичні технології, м. Харків, Україна; e-mail: prokopenkov.vf@gmail.com; ORCID: https://orcid.org/0000-0003-0084-9832.