

Е.Е. Полупанова, А.А. Рыбалко**АЛГОРИТМ ПОСЛЕДОВАТЕЛЬНОЙ ГИБРИДИЗАЦИИ ДЛЯ РЕШЕНИЯ ЗАДАЧИ КОММИВОЯЖЕРА**

Задача коммивояжера является задачей комбинаторной оптимизации. В статье приводится постановка данной задачи и предлагается графовая математическая модель, в которой вершины соответствуют городам, а рёбра – это пути между городами, причем предполагается, что граф взвешен. Решение задачи коммивояжера состоит в нахождении гамильтонова цикла минимального веса в полном взвешенном графе. Задача является NP-трудной, поэтому для решения данной задачи используется эвристический подход для получения решения задачи на больших объёмах входных данных. Эвристика заключается в применении для решения задачи коммивояжера гибридизации двух алгоритмов: алгоритма имитации отжига и алгоритма ближайшего соседа. Для решения задачи коммивояжера используется последовательная схема гибридизации. Основная идея заключается в том, что на стартовом наборе решений запускается метод ближайшего соседа, а затем лучшее решение, полученное на первом этапе, подается на вход алгоритму имитации отжига. В статье подробно освещены построение, блок-схемы гибридного алгоритма, алгоритма имитации отжига и метода ближайшего соседа. Далее в статье приводится описание пользовательского интерфейса приложения, написанного на Typescript. Приложение использует реальную карту местности для решения задачи коммивояжера. В последней части статьи освещается сравнительный анализ эффективности работы алгоритмов: сравнение точности и времени работы разработанного гибридного алгоритма, алгоритма имитации отжига и метода ближайшего соседа на различных входных наборах данных. Удалось установить, что разработанный гибридный алгоритм находится на втором месте по скорости работы и на первом по качеству решения среди реализованных алгоритмов. Кроме того, разработанное решение имеет высокую экономическую и практическую ценность ввиду того, что приложение для решения задачи коммивояжера, а следовательно, приложение для навигации по маршруту может заменить существующие аналоги или же оно может быть использовано в каких-либо узконаправленных областях, а также в логистике.

Задача коммивояжера; гибридный алгоритм; метод ближайшего соседа; алгоритм имитации отжига; веб-приложение; маршруты; навигатор; Typescript; react-leaflet.

E.E. Polupanova, A.A. Rybalko**SEQUENTIAL HYBRIDIZATION ALGORITHM FOR THE TRAVELING SALESMAN PROBLEM SOLVING**

The traveling salesman problem is a combinatorial optimization problem. The article presents a statement of this problem and proposes a graph mathematical model in which vertices correspond to cities, and edges are paths between cities, and it is assumed that the graph is weighted. The solution of the traveling salesman problem consists in finding the minimum weight Hamiltonian cycle in a complete weighted graph. The problem is NP-hard, so a heuristic approach is used to solve this problem and speed up the solution of the problem on large volumes of input data. The heuristic consists in applying hybridization of two algorithms to solve the traveling salesman problem: the annealing simulation algorithm and the nearest neighbor algorithm. Sequential hybridization scheme is used to solve the traveling salesman problem. The basic idea is that the nearest neighbor method is launched on the initial set of solutions, and then the best solution of the first stage is fed to the annealing simulation algorithm. The article details the construction, flowcharts of the hybrid algorithm, the annealing simulation algorithm, and the nearest neighbor method. The article goes on to describe the user interface of the application written in Typescript. The application uses an area map as a solution to the traveling salesman problem. In the last part of the article, a comparative analysis of the algorithms' performance is highlighted: a comparison of the accuracy and operating time of the developed hybrid algorithm, the annealing

simulation algorithm, and the nearest neighbor method for different input data sets. It was established that the developed hybrid algorithm is in second place in terms of speed and in first place in terms of solution quality among the implemented algorithms. In addition, the developed solution has a high economic and practical value because an application for solving the traveling salesman problem, and therefore an application for route navigation, can replace existing analogues or it can be used in any narrowly focused areas, as well as in logistics.

Traveling salesman problem; hybrid algorithm; nearest neighbor method; annealing simulation algorithm; web application; route; navigator; Typescript; react-leaflet.

Введение. Алгоритмы поисковой оптимизации нужны для решения тех или иных комбинаторных задач с целью получения решения задачи, максимально приближенного к оптимальному, за приемлемое время. Они используются при решении NP-полных проблем, таких как задача коммивояжера [1], задача упаковки ранца и др. Именно поэтому эти алгоритмы всегда были востребованы и сейчас они активно развиваются.

Алгоритмов поисковой оптимизации на сегодняшний день достаточно много и все они подразделяются на множество видов, таких как биоинспирированные, роевые, популяционные, генетические, стохастические и другие [2–4].

В соответствии с классификацией Ванга (X. Wang) выделяют три категории гибридных алгоритмов: вложенные (embedded) алгоритмы, алгоритмы типа препроцессор/постпроцессор (preprocessor/postprocessor), коалгоритмы (co-algorithms) [2].

В типе препроцессор/постпроцессор выделяют два класса методов гибридизации: последовательная и конвейерная.

В данной работе рассматривается гибридный алгоритм, построенный на алгоритме имитации отжига и методе ближайшего соседа, использующий принципы последовательной гибридизации, а также приводятся экспериментальные исследования реализованных алгоритмов.

Постановка задачи. Задача коммивояжера формулируется следующим образом: коммивояжер, выходящий из какого-нибудь города, желает посетить $(n - 1)$ других городов и вернуться к исходному [1], n – число городов.

Предлагается графовая математическая модель, в которой вершины соответствуют городам, а рёбра – это пути между городами. Предполагается, что граф взвешен, т.е. каждому ребру графа соответствует некоторый вес $c > 0$, и между каждой парой вершин существует ребро, т.е. граф полный.

Таким образом, решение задачи коммивояжера – это нахождение гамильтонова цикла минимального веса в полном взвешенном графе.

Данная задача возникает в обширном классе таких приложений, как, например, распознавание траекторий, образов, построение оптимальных схем движения и т.д. [5].

Существуют симметричная и ассиметричная задачи коммивояжера. Симметричная задача предполагает, что граф неориентированный. В ассиметричной задаче необходимо учитывать направление рёбер графа, так как работаем с ориентированным графом. В данной работе рассматривается решение ассиметричной задачи коммивояжера. Виды данной задачи подробно описаны в [6].

Алгоритм полного перебора решений задачи имеет сложность $O(n!)$. При малых n задача решается за приемлемое время, но если, к примеру, $n=100$, то задача будет решаться сотню миллионов лет. Однако, благодаря стохастическим алгоритмам оптимизации задачу можно решить за приемлемое время.

Задача коммивояжера является NP-трудной, то есть не доказано существования алгоритма точного решения задачи за экспоненциальное время. Поэтому на практике, для решения данной задачи, часто применяют эвристические методы [7–15]. Под эвристическими алгоритмами (“эвристиками”) чаще всего понимают

алгоритмы, для которых отсутствует либо неизвестна оценка точности [16]. В то же время порой именно эвристики являются единственным способом получить решение, максимально близкое к оптимальному, за приемлемое время.

Математическая постановка задачи. Математически, данную задачу можно сформулировать следующим образом: дан граф $H = (V, E)$, где V – непустое множество вершин, E – непустое множество рёбер. Каждому ребру (i, j) , где $i, j \in V$, можно сопоставить критерий выгодности маршрута $len_{ij} \geq 0$, который может быть расстоянием между городами, временем или стоимостью поездки.

Гамильтоновым циклом называется маршрут, включающий ровно по одному разу каждую вершину графа.

Решением задачи будет кратчайший гамильтонов цикл в графе. Для оценки качества решения используется целевая функция (1):

$$F_{len} = \left| \sum_{i=1}^{n-1} (m_{\phi(i)\phi(i+1)}) + m_{\phi(n-1)\phi(0)} \right|, \quad (1)$$

где F_{len} – целевая функция всего решения;

$M = (m_{ij})$ – это матрица размера $n \times n$, где (m_{ij}) – вес ребра (i, j) .

$\phi = (\phi_i)$ – это матрица размера n , где (ϕ_i) – индекс пункта в маршруте ϕ .

Целью задачи коммивояжера является минимизация представленной выше целевой функции.

Алгоритм имитации отжига. Алгоритм имитации отжига был предложен Н. Метрополисом в 1953 г. За основу работы был взят принцип процесса отжига металлов, который применяют в металлургии. Алгоритм относится к вероятностным методам решения. Метод отжига служит для поиска глобального минимума некоторой функции [17]. Ключевым моментом в таких подходах является случайный выбор одного из нескольких возможных решений вместо анализа каждого [18]. Алгоритм относят к методам глобального поиска.

Классический алгоритм имитации отжига предполагает следующие шаги:

а) задать начальное решение s и начальную температуру $T = T_{max}$, где T_{max} – максимальная температура;

б) пока $T > 0$, повторить N раз следующие действия:

1) выбрать новое решение s' из окрестности решений s ;

2) рассчитать изменение целевой функции (длины маршрута) $\Delta E = E(s') - E(s)$;

3) если $\Delta E \leq 0$, принять $s = s'$; иначе: посчитать $P_k = \exp(-\Delta E/T)$, сгенерировать число P из интервала $[0, 1]$ и если $P_k > P$, то принять новое решение $s = s'$; иначе проигнорировать его;

в) уменьшить температуру T , приняв $T = \alpha T$, где α – коэффициент понижения температуры, выбираемый из интервала $(0, 1)$ и если достигнут критерий останова, выйти из цикла, иначе перейти к шагу б).

Из формулы вероятности принятия нового решения $\exp(-\Delta E/T)$ видим, что эта вероятность зависит от текущей температуры и от разницы между целевыми функциями текущего и нового решений, и чем выше температура и меньше изменение целевой функции, тем больше вероятность того, что новое решение будет принято за текущее. Постепенно температура понижается и вероятность принятия нового решения существенно уменьшается.

Алгоритм имитации отжига не отбрасывает новое решение, которое хуже текущего, а принимает его с некоторой вероятностью. Именно это свойство алгоритма помогает ему выбираться из локальных оптимумов и достигать глобальных оптимумов [19].

Блок-схема алгоритма имитации отжига представлена на рис. 1.

Рассмотрим подробно шаги алгоритма имитации отжига:

а) задать максимальную температуру, начальный путь, количество итераций общего цикла, в котором понижается температура – N , и количество итераций отжига при определённой температуре – L . Берём $T = T_{\max}$.

б) пока счётчик итераций меньше или равен N выполнить L заданных итераций отжига.

1) С помощью псевдослучайных чисел $n1$ и $n2$ инвертируем путь в текущем пути от $n1$ до $n2$ и получаем новый путь – кандидат.

2) Считаем длину нового пути.

3) Если длина нового пути меньше или равна длине текущего, то принять новый путь в качестве текущего. Иначе, сгенерировать псевдослучайное число R из промежутка $[0, 1]$ и принять новое решение с вероятностью $P_k = \exp(-\Delta E/T)$, т.е. если $P_k > R$, то принимаем решение, иначе пропускаем его.

в) уменьшаем температуру в соответствии с установленным законом.

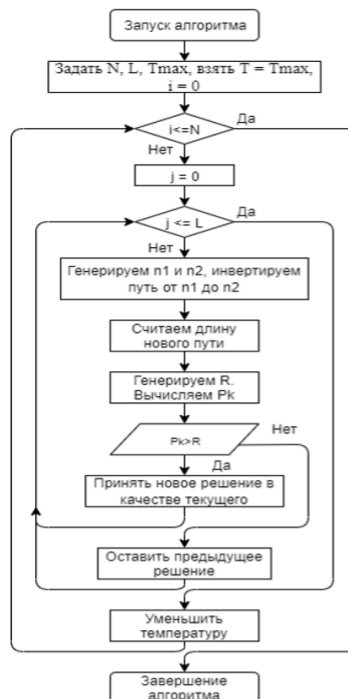


Рис. 1. Блок-схема алгоритма имитации отжига

Метод ближайшего соседа. Метод ближайшего соседа является жадным алгоритмом, т. е. он не всегда даёт оптимальные решения. Однако из-за его простоты, этот алгоритм отличается высокой скоростью работы. Алгоритм сильно зависит от топологии задачи. Например, если вершины графа выстроятся в круг, то тогда алгоритм найдет точное решение задачи.

Классический метод ближайшего соседа предполагает следующие шаги:

а) из всех не посещённых вершин выбрать одну вершину V и пометить её, как посещённую;

б) выбрать ближайшую к V вершину U ;

в) выбрать вершину U , как текущую и отметить её, как посещённую;

г) если посещены все вершины, то завершить алгоритм, иначе вернуться к шагу б).

Изначальный алгоритм был немного улучшен тем, что сравниваются пути от каждой вершины графа.

На рис. 2 представлена блок-схема метода ближайшего соседа.

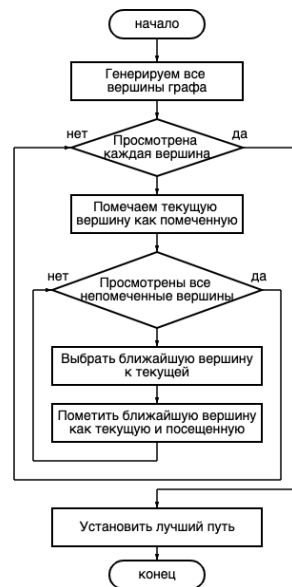


Рис. 2. Блок-схема метода ближайшего соседа

Гибридный алгоритм. В данной работе был разработан последовательный гибридный алгоритм решения задачи коммивояжера: сначала запускается метод ближайшего соседа, а затем на найденном решении запускается алгоритм имитации отжига, улучшая тем самым полученный результат.

На рис. 3 представлена блок-схема гибридного алгоритма. В блоке алгоритма имитации отжига при запуске алгоритма начальным решением выступает лучшее решение метода ближайшего соседа.

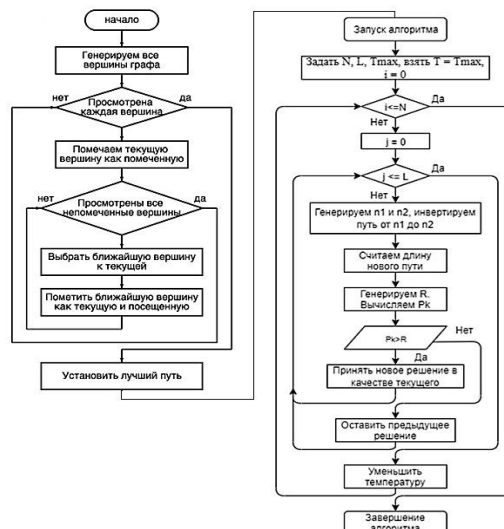


Рис. 3. Блок-схема гибридного алгоритма

Приложение. Для решения задачи коммивояжера и исследования реализованных алгоритмов было разработано веб-приложение. В качестве языка разработки был выбран Typescript – этот язык полностью идентичен языку Javascript, но поддерживает типизацию, благодаря чему программы писать намного проще и удобнее.

В качестве технологии отображения карты была выбрана библиотека react-leaflet, которая предоставляет широкие возможности для работы с картой. Библиотека работает в связке с библиотекой React.js – библиотекой для построения интерфейсов [20].

Основная идея приложения – пользователь отмечает желаемые точки на карте, тем самым создавая пункты маршрута; по нажатию кнопки расчёта запускается гибридный алгоритм, который определяет кратчайший путь передвижения на автомобиле. Алгоритм работает с матрицей продолжительности (времени), в которой элемент определяется продолжительностью (в секундах) прохождения пути от точки i к точке j .

Расчёт матрицы продолжительности происходит на удаленном сервере при помощи библиотеки OSRM – (Open Source Routing Machine) с использованием мультиуровневого алгоритма Дейкстры. В расчёте матрицы учитывается дистанция, приоритетность дорог, правила дорожного движения, и не учитываются какие-либо динамические факторы – текущие погодные условия, пробки и т.д.

Все точки маршрута хранятся в состоянии `points`, каждая точка (объект класса `LatLng`) хранит широту, долготу и функцию расчёта дистанции от неё самой, до какой-либо другой точки.

Этапы работы построения маршрута:

- 1) отсылаем запрос на сервер OSRM с массивом `points`;
- 2) получаем ответ от сервера – вычисленную матрицу продолжительности на основе массива `points`;
- 3) запуск гибридного алгоритма на основе матрицы продолжительности;
- 4) по сети от сервера OSRM получаем информацию о старом и новом маршрутах;
- 5) с помощью метода `createRoute` рисуем оптимальный маршрут на карте;
- 6) отображаем информацию о старом и новом маршрутах.

На рис. 4 представлен маршрут, построенный для 100 точек при помощи гибридного алгоритма, а на рис. 5 – модальное окно с информацией о старом и новом маршрутах.

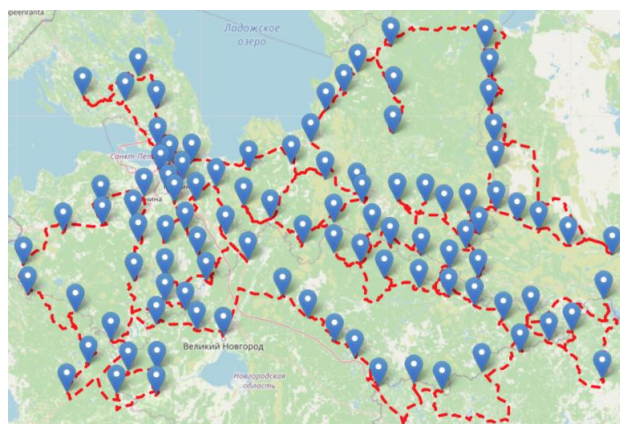


Рис. 4. Маршрут для 100 точек, построенный при помощи гибридного алгоритма

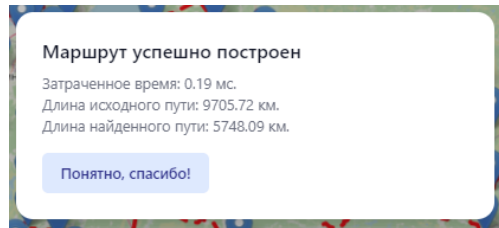


Рис. 5. Модальное окно с информацией о старом и новом маршрутах

Библиотека leaflet-routing-machine при построении маршрута предоставляет список всех событий маршрута, которым он должен следовать.

На рис. 6 представлен фрагмент списка инструкций маршрута для 100 вершин. Эти инструкции на английском языке имитируют реальный навигатор, но без озвучивания и отслеживания местоположения пользователя в реальном времени.

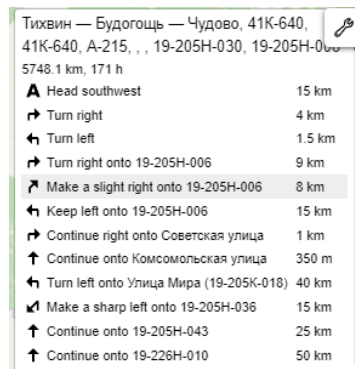


Рис. 6. Фрагмент списка событий маршрута для 100 вершин

Анализ эффективности реализованных алгоритмов. На рис. 7 представлен график зависимости времени работы от числа вершин для алгоритма имитации отжига, метода ближайшего соседа и гибридного алгоритма. Из графика видно, что гибридный алгоритм по скорости выполнения уступает методу ближайшего соседа, но обгоняет алгоритм имитации отжига.

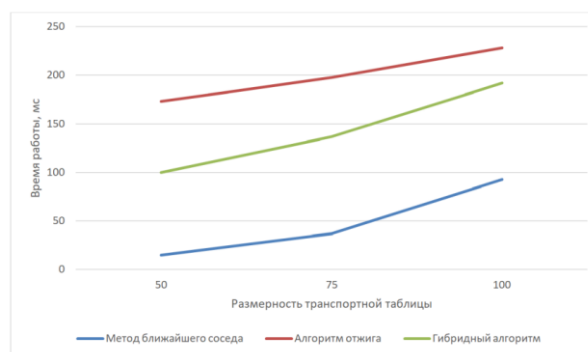


Рис. 7. Зависимость времени работы от числа вершин для алгоритма имитации отжига, алгоритма ближайшего соседа и гибридного алгоритма

Сложность алгоритма имитации отжига – $O(n)$, метода ближайшего соседа – $O(n^2)$, гибридного алгоритма – $O(n^2)$, где n – число вершин.

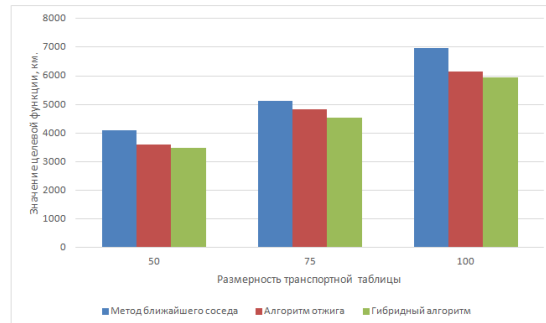


Рис. 8. Фрагмент списка событий маршрута для 100 вершин

На рис. 8 представлена диаграмма зависимости значения целевой функции от числа вершин. По диаграмме видно, что гибридный алгоритм опережает по качеству решения алгоритм имитации отжига и алгоритм ближайшего соседа.

Заключение. Задача коммивояжера является NP-трудной, поэтому для её решения в работе был предложен эвристический подход для получения решения на больших объёмах входных данных. Суть эвристики заключалась в применении для решения задачи коммивояжера гибридизации двух алгоритмов: алгоритма ближайшего соседа и алгоритма имитации отжига.

Алгоритм имитации отжига отличается от всех эвристических алгоритмов тем, что он допускает в процессе поиска шаги, приводящие к увеличению целевой функции. Метод ближайшего соседа, в свою очередь, относится к числу жадных алгоритмов, т.е. на каждом шаге выбирает наилучшее решение. Гибридизация алгоритмов заключалась в последовательном применении сначала метода ближайшего соседа, а затем на найденном решении алгоритма имитации отжига для улучшения полученного результата.

Для демонстрации работы гибридного алгоритма реализовано приложение для расчёта оптимального маршрута и решения задачи коммивояжера при помощи алгоритмов оптимизации на реальной карте местности с использованием библиотеки react-leaflet.

В результате экспериментальных исследований удалось установить, что гибридный алгоритм опережает по качеству алгоритм имитации отжига, но уступает методу ближайшего соседа по скорости решения.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Мудров В.И. Задача о коммивояжёре. – М.: Знание, 1969. – 64 с.
2. Карпенко А.П. Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой. – М.: МГТУ им. Н.Э. Баумана, 2014. – 446 с.
3. Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы: учеб. пособие. – 2-е изд. – М.: Физматлит, 2006. – 368 с.
4. Саймон Д. Алгоритмы эволюционной оптимизации. – М.: Изд-во: ДМК Пресс, 2020. – 1002 с.
5. Борознов В.О. Исследование решения задачи коммивояжера // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. – 2009. – № 2. – С. 147-151.
6. Колесников А.В., Кириков И.А., Листопад С.В. Решение сложных задач коммивояжера методами функциональных гибридных интеллектуальных систем / под ред. А.В. Колесникова. – М.: ИПИ РАН, 2011. – 295 с.

7. *Morán-Mirabal L.F., González-Velarde J.L., Resende M.G.C.* Randomized heuristics for the family traveling salesperson problem // *International Transactions in Operational Research*. – 2014. – Vol. 21, Issue 1. – P. 41-57. – DOI: 10.1111/itor.12026.
8. *Nagata Y.* High-Order Entropy-Based Population Diversity Measures in the Traveling Salesman Problem // *Evolutionary Computation*. – 2020. – Vol. 28, Issue 4. – P. 595-619. – DOI: 10.1162/evco_a_00268.
9. *Pop P., Matei O., Pinte C.* A two-level diploid genetic based algorithm for solving the family traveling salesman problem // *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'18)*. Association for Computing Machinery, New York, NY, USA, 2018. – P. 340-346. – DOI: 10.1145/3205455.3205545.
10. *Bernardino R., Paías A.* Solving the family traveling salesman problem // *European Journal of Operational Research*. – 2018. – Vol. 267, Issue 2. – P. 453-466. – DOI: 10.1016/j.ejor.2017.11.063.
11. *Cacchiani V., Muritiba A.E.F., Negreiros M., Toth P.* A multistart heuristic for the equality generalized traveling salesman problem // *Networks*. – 2011. – Vol. 57, Issue 3. – P. 231-239. – DOI: 10.1002/net.20421.
12. *Garn W.* Balanced dynamic multiple travelling salesmen: Algorithms and continuous approximations // *Computers & Operations Research*. – 2021. – Vol. 136. – Article 105509. – DOI: 10.1016/j.cor.2021.105509.
13. *Рыбалко А.А., Полупанова Е.Е.* Полиалгоритм решения задачи коммивояжера // *Современные проблемы математики, информатики и моделирования: Матер. III Всероссийской научно-практической конференции молодых учёных; Краснодар: Краснодарский ЦНТИ - филиал ФГБУ "РЭА" Минэнерго России, 2021. – Т. 1. – С. 208-213.*
14. *Полупанова Е.Е., Поляков А.С.* Популяционный алгоритм решения задачи коммивояжера // *Современные информационные технологии и ИТ-образование; Москва: Фонд содействия развитию интернет-медиа, ИТ-образования, человеческого потенциала Лига интернет-медиа, 2021. – Т. 17, № 2. – С. 324-333.*
15. *Курейчик В.М., Гогохия Л.Р.* Решения задачи коммивояжера с применением генетического оператора старения // *Тр. международного научно-технического конгресса "интеллектуальные системы и информационные технологии - 2020" ("ИС & ИТ-2020", "is&it'20")*; научное издание: в 2-х т. – 2020. – С. 7-16.
16. *Сергиенко И.В., Гуляницкий Л.Ф., Сиренко С.И.* Классификация прикладных методов комбинаторной оптимизации // *Кибернетика и системный анализ*. – 2009. – № 5. – С. 71-83.
17. Метод отжига. – URL: <http://www.math.spbu.ru/user/gran/sb1/lopatin.pdf> (дата обращения: 10.05.2023).
18. *Курсанов М.Н.* Графы в Maple. Задачи, алгоритмы, программы. – М.: Изд-во Физматлит, 2007. – 167 с.
19. *Осовский С.* Нейронные сети для обработки информации. – М.: Финансы и статистика, 2002. – 343 с.
20. React.js. – URL: <https://ru.reactjs.org/> (дата обращения: 07.05.2023).

REFERENCES

1. *Mudrov V.I.* Zadacha o kommivoyazhere [The problem of a traveling salesman]. Moscow: Znanie, 1969, 64 p.
2. *Karpenko A.P.* Sovremennye algoritmy poiskovoy optimizatsii. Algoritmy, vdokhnovlennyye prirodoy [Modern search engine optimization algorithms. Algorithms inspired by nature]. Moscow: MGTU im. N.E. Bauman, 2014, 446 p.
3. *Gladkov L.A., Kureychik V.V., Kureychik V.M.* Geneticheskie algoritmy: ucheb. posobie [Genetic algorithms: training manual]. 2nd ed. Moscow: Fizmatlit, 2006, 368 p.
4. *Saymon D.* Algoritmy evolyutsionnoy optimizatsii [Evolutionary optimization algorithms]. Moscow: Izd-vo: DMK Press, 2020, 1002 p.
5. *Boroznov V.O.* Issledovanie resheniya zadachi kommivoyazhera [Investigation of the solution of the traveling salesman problem], *Vestnik Astrakhanskogo gosudarstvennogo tekhnicheskogo universiteta. Seriya: Upravlenie, vychislitel'naya tekhnika i informatika* [Vestnik of Astrakhan State Technical University. Series: Management, Computer Sciences and Informatics], 2009, No. 2, pp. 147-151.
6. *Kolesnikov A.V., Kirikov I.A., Listopad S.V.* Reshenie slozhnykh zadach kommivoyazhera metodami funktsional'nykh gibridnykh intellektual'nykh sistem [Listopad Solving complex problems of a traveling salesman by methods of functional hybrid intelligent systems], ed. by A.V. Kolesnikova. Moscow: IPI RAN, 2011, 295 p.

7. Morán-Mirabal L.F., González-Velarde J.L., Resende M.G.C. Randomized heuristics for the family traveling salesperson problem, *International Transactions in Operational Research*, 2014, Vol. 21, Issue 1, pp. 41-57. DOI: 10.1111/itor.12026.
8. Nagata Y. High-Order Entropy-Based Population Diversity Measures in the Traveling Salesman Problem, *Evolutionary Computation*, 2020, Vol. 28, Issue 4, pp. 595-619. DOI: 10.1162/evco_a_00268.
9. Pop P., Matei O., Pinte C. A two-level diploid genetic based algorithm for solving the family traveling salesman problem, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'18)*. Association for Computing Machinery, New York, NY, USA, 2018, pp. 340-346. DOI: 10.1145/3205455.3205545.
10. Bernardino R., Paia A. Solving the family traveling salesman problem, *European Journal of Operational Research*, 2018, Vol. 267, Issue 2, pp. 453-466. DOI: 10.1016/j.ejor.2017.11.063.
11. Cacchiani V., Muritiba A.E.F., Negreiros M., Toth P. A multistart heuristic for the equality generalized traveling salesman problem, *Networks*, 2011, Vol. 57, Issue 3, pp. 231-239. DOI: 10.1002/net.20421.
12. Garn W. Balanced dynamic multiple travelling salesmen: Algorithms and continuous approximations, *Computers & Operations Research*, 2021, Vol. 136, Article 105509. DOI: 10.1016/j.cor.2021.105509.
13. Rybalko A.A., Polupanova E.E. Polialgoritm resheniya zadachi kommvoyazhera [Polyalgorithm of solving the traveling salesman problem], *Sovremennye problemy matematiki, informatiki i modelirovaniya: Mater. III Vserossiyskoy nauchno-prakticheskoy konferentsii molodykh uchenykh; Krasnodar: Krasnodarskiy TSNTI - filial FGBU "REA" Minenergo Rossii, 2021* [Modern problems of mathematics, computer science and modeling – materials of the III All-Russian Scientific and Practical Conference of Young Scientists, Krasnodar, Krasnodar Central Research Institute - Branch of the Federal State Budgetary Institution "REA" of the Ministry of Energy of Russia, 2021], Vol. 1, pp. 208-213.
14. Polupanova E.E., Polyakov A.S. Populyatsionnyy algoritm resheniya zadachi kommvoyazhera [Population algorithm for solving the traveling salesman problem], *Sovremennye informatsionnye tekhnologii i IT-obrazovanie; Moskva: Fond sodeystviya razvitiyu internet-media, IT-obrazovaniya, chelovecheskogo potentsiala Liga internet-media, 2021* [Modern Information Technologies and IT Education; Moscow, Foundation for the Promotion of Internet Media, IT Education, Human Potential, Internet Media League, 2021], Vol. 17, No. 2, pp. 324-333.
15. Kureychik V.M., Gogokhiya L.R. Resheniya zadachi kommvoyazhera s primeneniem geneticheskogo operatora stareniya [Goghia Solving the traveling salesman problem using the genetic operator of aging], *Tr. mezhdunarodnogo nauchno-tekhnicheskogo kongressa "intellektual'nye sistemy i informatsionnye tekhnologii - 2020" ("IS & IT-2020", "is&it'20")*; *nauchnoe izdanie* [Proceedings of the International Scientific and Technical Congress "Intelligent Systems and Information Technologies - 2020" ("IS & IT-2020", "is&it'20"). scientific publication]; in 2nd vol., 2020, pp. 7-16.
16. Sergienko I.V., Gulyanitskiy L.F., Sirenko S.I. Klassifikatsiya prikladnykh metodov kombinatornoy optimizatsii [Classification of applied methods of combinatorial optimization], *Kibernetika i sistemnyy analiz* [Cybernetics and system analysis], 2009, No. 5, pp. 71-83.
17. Metod otzhiga [Annealing method]. Available at: <http://www.math.spbu.ru/user/gran/sb1/lopatin.pdf> (accessed 10 May 2023).
18. Kirsanov M.N. Grafy v Maple. Zadachi, algoritmy, programmy [Graphs in Maple. Tasks, algorithms, programs]. Moscow: Izd-vo Fizmatlit, 2007, 167 p.
19. Osovskiy S. Neyronnye seti dlya obrabotki informatsii [Neural networks for information processing]. Moscow: Finansy i statistika, 2002, 343 p.
20. React.js. Available at: <https://ru.reactjs.org/> (accessed 07 May 2023).

Статью рекомендовал к опубликованию д.т.н., профессор В.Н. Марков.

Полупанова Елена Евгеньевна – Кубанский государственный университет; e-mail: jjienka@mail.ru; г. Краснодар, Россия; тел.: +79284013301; кафедра вычислительных технологий; к.т.н.; доцент; ORCID: <https://orcid.org/0000-0002-0364-1132>.

Рыбалко Александр Андреевич – e-mail: alexmobilepost@gmail.ru; тел.: +79282563456, кафедра вычислительных технологий; магистрант.

Polupanova Elena Evgenievna – Kuban State University; e-mail: jienka@mail.ru; Krasnodar, Russia; phone: +79284013301; the department of computational technologies; cand. of eng. sc.; associate professor; ORCID: <https://orcid.org/0000-0002-0364-1132>.

Rybalko Alexandr Andreevich – e-mail: alexmobilepost@gmail.ru; phone: +79282563456, the department of computational technologies; master's degree student.

УДК 004.8

DOI 10.18522/2311-3103-2023-3-118-125

Е.С. Подоплелова

АНАЛИЗ МЕТОДОВ МНОГОКРИТЕРИАЛЬНОГО ПРИНЯТИЯ РЕШЕНИЙ НА ПРИМЕРЕ ЗАДАЧИ РАНЖИРОВАНИЯ

Данная работа посвящена отбору и сравнению популярных традиционных методов многокритериального принятия решений. В статье представляется обзор существующих работ последних лет на тему их сравнения, выделены основные критерии, а также наиболее значимые результаты. Далее, был рассмотрен пример реализации СППР (системы поддержки принятия решений) по рекомендации такого метода пользователю, который включает описание не только основных методов, но и их модификаций, выделяя исчерпывающую таксономию методов многокритериального анализа в целом. Для отбора методов в этой статье были использованы международные базы научных публикаций: Science Direct, Google Scholar и IEEE Xplore. Были произведены определенные настройки поиска для получения работ, соответствующих запросу. На следующем этапе описывается задача ранжирования альтернатив для демонстрации результатов применения выбранных методов. В качестве метода распределения весов критериев использовался метод анализа иерархий (МАИ). Результаты вычислений представлены в таблицах и графически. Метрикой оценки было принято считать устойчивость метода к количеству альтернатив и критериев, а также чувствительность к весам критериев. На текущем шаге исследования были отобраны следующие методы: TOPSIS, WASPAS, VIKOR, PROMETHEE и ELECTRE. В результате исследования были определены оптимальные методы (по соотношению вычислительной сложности к устойчивости) для их дальнейшего использования в разработке СППР, метод ELECTRE было принято использовать как дополнительный инструмент при большом количестве альтернатив для отсеивания наименее привлекательных. PROMETHEE показал высокую чувствительность к изменению весов и сложности вычислений, поэтому был исключен из дальнейших этапов разработки. VIKOR и TOPSIS показали наилучшую устойчивость при простоте вычислений.

Методы многокритериального принятия решений; система поддержки принятия решений; TOPSIS; VIKOR; Метод анализа иерархий; ELECTRE, PROMETHEE; ранжирование альтернатив.

E.S. Podoplelova

SELECTION OF MULTI-CRITERIA ANALYSIS METHODS ON THE EXAMPLE OF THE PROBLEM OF RANKING

This work is devoted to the selection and comparison of popular traditional methods of multi-criteria decision making. The article presents an overview of the existing works of recent years on the topic of their comparison, highlights the main criteria, as well as the most significant results. Further, an example of the implementation of a DSS (decision support system) was considered on the recommendation of such a method to the user, which includes a description of not only the main methods, but also their modifications, highlighting an exhaustive taxonomy of multi-criteria analysis methods in general. For the selection of methods in this article, international databases of scientific publications were used: Science Direct, Google Scholar and IEEE Xplore. Certain search settings have been made to retrieve jobs that match the query. The next step de-