Markov Vladimir Vasilievich – Southern Federal University; e-mail: vvmarkov@sfedu.ru; 44, Nekrasovskiy lane, Taganrog, 347928, Russia; phone: +78634371651; the department of computer aided design; associate professor.

Kravchenko Yury Alekseevich – e-mail: yakravchenko@sfedu.ru; the department of computer aided design; associate professor.

Kuzmina Maria Anatolyevna – e-mail: kuzmina.maria.tti@gmail.com; the department of computer aided design, graduate student.

УДК 002.53:004.89 DOI 10.23683/2311-3103-2018-4-185-197

А.Н. Нацкевич, И.О. Курситыс

КОМБИНИРОВАННЫЙ БИОИНСПИРИРОВАННЫЙ АЛГОРИТМ ДЛЯ РЕШЕНИЯ ЗАДАЧИ КЛАСТЕРИЗАЦИИ ДАННЫХ*

Статья посвящена решению одной из популярнейших задач интеллектуального анализа данных – задачи кластеризации. Кластеризация – объединение в группы схожих объектов – является одной из фундаментальных задач в области анализа данных и Data Mining. Список прикладных областей, где она применяется, широк: сегментация изображений, маркетинг, борьба с мошенничеством, прогнозирование, анализ текстов и многие другие. Решение данной задачи приобретает особую актуальность в условиях постоянно растущего объема генерируемых, передаваемых и обрабатываемых данных. Авторами исследована задача кластеризации, приведены постановка задачи, основные формулы для ее решения, а так же целевая функция. Проведен аналитический обзор существующих алгоритмов, таких как: алгоритмы иерархической кластеризации, квадратичной ошибки, алгоритмы k-means и c-means, алгоритмы, основанные на теории графов. Отмечены основные достоинства и недостатки рассмотренных алгоритмов. Предложено использовать методы биоинспирированного поиска для решения задачи кластеризации, обоснована актуальность применения биоинспирированных моделей и методов для решения NP-полных задач, к классу которых относится и исследуемая задача. Отмечен вклад ученых в решение данной проблемы – биоинспирированные алгоритмы, такие как метод роя частиц, муравьиный алгоритм, пчелиный алгоритм, алгоритм бактериальной оптимизации, алгоритм кукушки и многие другие подобные методы успешно применяются для решения задачи кластеризации. Предложен комбинированный биоинспирированный алгоритм, применяющий последовательно муравьиный алгоритм и алгоритм летучих мышей. Раскрыты основные идеи алгоритмов, приведены схемы решения задачи, кодирования решений. Реализован метод локального поиска для алгоритма летучих мышей. Проведены экспериментальные исследования на тестовых примерах (бенчмарках), которые доказывают эффективность разработанного алгоритма по сравнению с алгоритмами k-средних и генетическим алгоритмом. В ходе проведения экспериментов определена временная сложность разработанного комбинированного биоинспирированного алгоритма. Предложенную модель комбинированого решения задачи планируется в дальнейшем использовать для работы ранее разработанного бустинга алгоритмов, который работает с несколькими алгоритмами и позволяет найти лучшее решение из решений, полученных разными биоинспирированными алгоритмами.

Кластеризация; биоинспирированный алгоритм; муравьиный алгоритм; алгоритм летучих мышей; роевый алгоритмы; роевой интеллект; искуственный интеллект.

* Работа выполнена при поддержке РФФИ (проект № 16-07-00703)..

A.N. Natskevich, I.O. Kursitys

COMBINED BIOINSPIRED ALGORITHM FOR SOLVING THE CLUSTERING PROBLEM

The article is devoted to solving the clustering problem, which is one of the most important and popular problem in intelligent data analysis. Clustering, which means uniting the similar elements in groups, is one of the fundamental problem in Data Mining. Application of solving this problem includes image segmentation, marketing, protection from financial fraud, forecasting, text analysis and many other fields. A constantly growing scope of generated, transferred and processed data determines the significance of the problem. The authors investigate the clustering problem, provide the problem statement, the main mathematical formulas and the objective function needed for solving. The article consists of the analytical review of the popular algorithms, such as hierarchical optimization, squared error, k-means, c-means and graph-based algorithms. The authors note their benefits and shortcomings. The clustering problem is NP-complete, which determines the advantages of application of bioinspired models and methods for solving the mentioned problem. The related works of famous researchers are given in the article, such as: particle swarm optimization, ant colony optimization, artificial bee colony algorithm, bacteria colony optimization, cuckoo search algorithm, which demonstrate the effectiveness of bioinspired algorithms in terms of solving the clustering problem. The authors propose a combined bioinspired algorithm which applies the ant colony optimization and the bat algorithm successively. The main ideas of the algorithms, their flowcharts and solutions encoding schemes are provided herein. The local search method is implemented in the bat algorithm. The experiments carried out with benchmarks demonstrate the effectiveness of the proposed algorithm in comparison with the k-means algorithm and the genetic one. During the experimental research the authors managed to define the developed combined bioinspired algorithm time complexity. The authors are planning to apply the suggested combined solution for boosting of the algorithms, which works with several algorithms and reveals the best solution among several ones obtained with different bioinspired algorithms.

Clustering; bioinspired algorithm; ant colony optimization; bat algorithm; swarm intelligence, artificial intelligence.

Введение. В современном мире одной из ключевых характеристик развития общества является развитие информационных технологий и, как следствие, непрерывный рост объема генерируемой, передаваемой и обрабатываемой информации. В связи с этим, актуальным направлением сегодня является разработка методов обработки больших массивов данных (Big Data), а так же их интеллектуальный анализ, который позволяет обнаруживать в наборах данных неявные закономерности, что существенно упрощает решение задач последующей обработки данных.

Проблема роста больших массивов данных обосновывает актуальность создания новых масштабируемых алгоритмов анализа данных, которые способны выдать хорошие результаты при условии оптимальных временных затрат. Одним их наиболее часто и успешно используемых методов анализа данных является кластеризация, что связано с необходимостью деления огромного количества постоянно растущего объема данных на кластеры [1, 20] для последующего упрощения их обработки с целью выделения информации и решения различных научных проблем.

Кластеризация – объединение в группы схожих объектов – является одной из фундаментальных задач в области анализа данных и Data Mining. Список прикладных областей, где она применяется, широк: сегментация изображений, маркетинг, борьба с мошенничеством, прогнозирование, анализ текстов и многие другие. Кластеризация в Data Mining приобретает ценность тогда, когда она выступает одним из этапов анализа данных, построения законченного аналитического решения. Как правило, кластеризация применяется для того, чтобы произвести так называемое сжатие данных, т.е. сократить объём используемых данных за счёт того, что внутри кластера объекты не различаются (рассматриваются как один объект).

1. Постановка задачи кластеризации. В работе [2] авторами приведена следующая постановка задачи кластеризации данных.

Пусть X={xi | i=1,2,…,n} – множество объектов, каждый объект описывается множеством атрибутов (признаков конкретного объекта) A={aj | j=1,2,...,m}. Y={yl | l=1,2,...,k} – множество кластеров, по которым необходимо распределить объекты. Каждый кластер содержит центроид cl∈C, описывающий средние параметры множества объектов, входящих в данный кластер. Задана функция расстояния между объектами P(xi , xj). Требуется разбить множество объектов на непересекающиеся подмножества так, чтобы каждый кластер состоял из объектов, близких по метрике p. В процессе решения каждому объекту приписывается номер кластера l. Алгоритм кластеризации – это функция a: X -> Y, которая любому объекту xi X ставит в соответствие номер кластера l. Количество кластеров при этом может быть известно заранее или определяться в процессе работы алгоритма.

В качестве метрики p выбрана Евклидова метрика. Формула для подсчета расстояния между двумя объектами выглядит следующим образом:

$$d = \sqrt{\sum_{k=1}^{m}(x_i^k - x_j^k)^2}. \quad (1)$$

Для определения значений объектов используется шкалирование по методу Минимакс. Используется шкала [0, 1]. Общая формула для шкалирования выглядит следующим образом:

$$a_i' = \frac{(a_i - \min(a_i))}{(\max(a_i) - \min(a_i))}, \quad (2)$$

где α_i'– новое нормализованное значение i атрибута объекта, – ненормализованное значение атрибута объекта, min(αi) – минимальное значение i атрибута, max(αi) – максимальное значение атрибута α.

Центроид кластера j определяется по следующей формуле:

$$c_j' = \frac{1}{S_j}\sum_{x_i \in S_j}\sum_{a^k \in x} a_i^k + c_j^k. \quad (3)$$

Решением задачи кластеризации является множество V'={Y'l|l=1,2,…,k}. Запланированным вариантом решения *V'* является разбиение множества объектов по множеству кластеров.

В качестве оценки решения *V'* рассматривается целевая функция, имеющая следующий вид:

$$F = \frac{P^o}{P^i} \to max, \quad (4)$$

где P_o – среднее межкластерное расстояние, P_i – среднее внутрикластерное расстояние.

Рассмотрим подробнее механизм организации бустинга с помощью использования биоинспирированного алгоритма.

При этом формула для подсчета внутрикластерного расстояния имеет следующий вид:

$$P^i = \frac{1}{X}\sum_{j=1}^{n}\sum_{i=1}^{n} p(x_i, c_j) \to min, \quad (5)$$

где p – расстояние, которое вычисляется по формуле выбранной метрики, $x \in X$ – текущий элемент, $c \in C$ – центроид данного кластера, k – общее количество элементов, l – количество элементов в конкретном j кластере.

Среднее межкластерное расстояние описывает расстояние между объектами, входящими в состав различных кластеров и определяется по следующей формуле:

$$P^o = \frac{1}{U}\sum_{u \in U} p(u_i, u) \to max, \quad (6)$$

где p – расстояние с учетом выбранной метрики, u_i – рассматриваемый центроид, u – центроид, относительно которого вычисляется среднее межкластерное расстояние, n – общее количество кластеров.

2. Обзор алгоритмов решения поставленной задачи.

Алгоритмы иерархической кластеризации. Среди алгоритмов иерархической кластеризации выделяются два типа: восходящие и нисходящие алгоритмы. Нисходящие алгоритмы работают по принципу «сверху-вниз»: в начале все объекты помещаются в один кластер, который затем разбивается на все более мелкие кластеры. Более распространены восходящие алгоритмы, которые в начале работы помещают каждый объект в отдельный кластер, а затем объединяют кластеры во все более крупные, пока все объекты выборки не будут содержаться в одном кластере. Результаты таких алгоритмов обычно представляют в виде дерева. К недостатку иерархических алгоритмов можно отнести систему полных разбиений, которая может являться излишней в контексте решаемой задачи [3, 19].

Алгоритмы квадратичной ошибки. Задачу кластеризации можно рассматривать как построение оптимального разбиения объектов на группы. При этом оптимальность может быть определена как требование минимизации среднеквадратической ошибки разбиения:

$$e^2(X, L) = \sum_{j=1}^{K} \sum_{i=1}^{n_j} \left\| x_i^{(j)} - c_j \right\|^2, \quad (7)$$

где c_j — «центр масс» кластера j, K – количество кластеров, n – количество точек.

Самым распространенным алгоритмом этой категории является метод k-средних (k-means). Этот алгоритм строит заданное число кластеров, расположенных как можно дальше друг от друга. Работа алгоритма делится на несколько этапов:

1. Случайно выбрать k точек, являющихся начальными «центрами масс» кластеров.

2. Отнести каждый объект к кластеру с ближайшим «центром масс».

3. Пересчитать «центры масс» кластеров согласно их текущему составу.

4. Если критерий остановки алгоритма не удовлетворен, вернуться к п. 2.

В качестве критерия остановки работы алгоритма обычно выбирают минимальное изменение среднеквадратической ошибки. К недостаткам данного алгоритма можно отнести необходимость задавать количество кластеров для разбиения [3].

Нечеткие алгоритмы. Наиболее популярным алгоритмом нечеткой кластеризации является алгоритм c-средних (c-means). Он представляет собой модификацию метода k-средних. Шаги работы алгоритма:

1. Выбрать начальное нечеткое разбиение n объектов на k кластеров путем выбора матрицы принадлежности U размера (n, k).

2. Используя матрицу U, найти значение критерия нечеткой ошибки:

$$E^2(X, L) = \sum_{i=1}^{N} \sum_{k=1}^{K} U_{ik} \left\| x_i^{(k)} - c_k \right\|^2, \quad (8)$$

где ck – «центр масс» нечеткого кластера k, K – количество кластеров, N – количество точек.

3. Перегруппировать объекты с целью уменьшения этого значения критерия нечеткой ошибки.

4. Возвращаться в п. 2 до тех пор, пока изменения матрицы U не станут незначительными. Этот алгоритм может не подойти, если заранее неизвестно число кластеров, либо необходимо однозначно отнести каждый объект к одному кластеру.

Алгоритмы, основанные на теории графов. Суть таких алгоритмов заключается в том, что выборка объектов представляется в виде графа G=(V, E), вершинам которого соответствуют объекты, а ребра имеют вес, равный «расстоянию» между объектами. Достоинством графовых алгоритмов кластеризации являются нагляд-

ность, относительная простота реализации и возможность внесения различных усовершенствований, основанные на геометрических соображениях. Основными алгоритмам являются алгоритм выделения связных компонент, алгоритм построения минимального покрывающего (остовного) дерева и алгоритм послойной кластеризации.

Задача кластеризации относится к категории NP-полных задач. Для решения таких задач в настоящее время широко используются недетерминированные (стохастические), работающие одновременно с большим количеством текущих решений (многоагентные) алгоритмы, являющиеся более эффективными и универсальным.

За последние несколько десятилетий наблюдается широкое распространение и развитие методов и алгоритмов, инспирированных эволюцией (эволюционные алгоритмы, генетические алгоритмы, методы имитации отжига), и коллективным поведением особей в природе (муравьиный алгоритм, алгоритм пчелиной колонии, метод роя частиц, алгоритм поведения светлячков, алгоритм стаи серых волков, алгоритм летучих мышей) [8–10].

Исследованию биоинспирированных алгоритмов для решения задачи кластеризации посвятили свои труды такие ученые, как Van, D.M. и A.P. Engelbrecht [4], которые предложили оптимизацию кластеризации данных методом роя частиц; Shelokar [5], предложивший муравьиный алгоритм для кластеризации; Kao [6], предложивший гибридный алгоритм, использующий метод роя частиц и k-means; Zhang [7], использовавший алгоритм пчелиной колонии, X. Yan, предложивший гибридный пчелиный алгоритм с использованием оператора кроссинговера из классического генетического алгоритма, M.Wan [8], предложивший кластеризацию алгоритмом бактериальной оптимизации, а также J. Senthilnatha, Vipul Dasb, Omkara и V. Mani [9], предложившие алгоритм кукушки для решения задачи кластеризации. Все вышеуказанные алгоритмы показали свою эффективность при решении задачи кластеризации за полиномиальное время [10, 15–18]

Таким образом, использование биоинспирированных методов и алгоритмов, а также их комбинаций, является популярным и эффективным решением NP-полных задач большой размерности и конкретно – задачи кластеризации.

В данной работе для решения задачи кластеризации предлагается использовать гибридный алгоритм, комбинирующий последовательно метод поведения муравьиной колонии и метод летучих мышей.

3. Алгоритм муравьиной колонии. Поиск решений осуществляется в ориентированном двудольном графе (рис. 1) на основе итерационного выполнения базовой техники алгоритма поиска лучшего решения. Работа поисковой процедуры начинается с построения в соответствии со спецификой решаемой задачи графа поиска решений. Граф поиска решений представлен в виде выражения $H^l=(E \cup W, U^l)$, где $E=\{e_i \mid i=1,2,\ldots,n\}$ – первая доля, описывающая множество объектов для кластеризации, а $W=\{w_j \mid j=1,2,\ldots,m\}$ – вторая доля, описывающая множество кластеров. Ребро U_{ij} связывает вершину $e_i \in E$ с $W_j \in W$. $U^l = \{u_j \mid j=1,2, \ldots, m\}$ – множество ребер, связывающих вершины множества E с вершинами множества W, l – номер агента, получившего текущее решение. Ребро указывает на возможность принадлежности текущего объекта кластеру, которое в дальнейшем учитывается при работе алгоритма k-means. Для наглядности представления конкретного решения V^l сгруппируем множество заданий W в подмножества [2].

В работе поиск решения Vl сводится к поиску на полном двудольном графе H_{nm} такого решения H^l, для которого оценка F^l имеет минимальное значение. Первоначально, для каждого объекта выбираются объекты, атрибуты которых счита-

ются начальными атрибутами центроидов каждого класса. Объекты выбираются таким образом, чтобы согласно метрике ρ они были наиболее далекими друг от друга. Сами объекты на этапе определения центроида не распределяются.

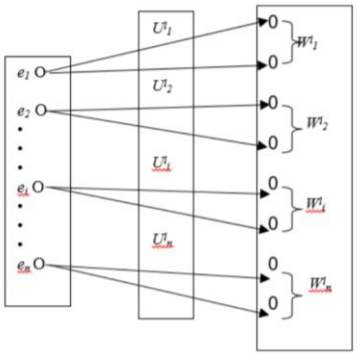

Рис. 1. Модель поиска решений

На каждой итерации l выполняется три этапа. На первом этапе каждый муравей строит решение (разбиение на кластеры), на втором этапе откладывается феромон, на третьем этапе осуществляется испарение феромона.

Моделирование поведения муравьёв связано с анализом количества феромона на ребрах графа H_{nm} и близости к центроиду кластера. На начальном этапе на всех ребрах U графа H_{nm} откладывается одинаковое (небольшое) количество феромона Q/v, где v=|U|. Формируется список объектов. Обозначим граф Hnm после отложения на нем на итерации t феромона, как $H_{nm}(t)$. На первом этапе каждый муравей осуществляет распределение объектов по кластерам. Для этого муравей просматривает последовательно множество объектов и на каждом шаге для рассматриваемого объекта случайным образом выбирается кластер. Вероятность распределения объекта в кластер зависит от веса каждого ребра, который определяется по формуле:

$$f_{ij} = (\alpha d(x_i, y_j) + \beta \tau_{i,j}), \quad (9)$$

где α – коэффициент, определяющий вес критерия расстояния конкретного объекта от центроида кластера при распределении; d – расстояние с учетом используемой метрики; x∈X– текущий i объект для распределения; c∈C – центроид j кластера; β – коэффициент, определяющий вес критерия количества отложенного феромона; τ_i – количество феромона отложенного на ребре $u_{i,j}$. Формула для определения вероятности $p_{i,j}$ распределения объекта x_i в кластер y_j Y выглядит следующим образом:

$$P_{i,j} = \frac{(\alpha d(x_i, y_j) + \beta \tau_{i,j})}{\sum_{j=1}^{n} (\alpha d(x_i, y_j) + \beta \tau_{i,j})}. \quad (10)$$

Таким образом, последовательно, для каждого элемента, муравей определяет кластер, в который необходимо распределить каждый конкретный объект. Каждый l-й агент формирует на ребрах графа Hnm(t-1) свой собственный граф – решение Hl(t), определяется решение Vl(t) и оценка решения Fl(t).

После того, как каждый агент построил свое решение, происходит формирования центроидов кластеров по следующей формуле:

$$c_j' = \frac{1}{S_j} \sum_{x_i \in S_j} x_i, \quad (11)$$

где c'j – новое значение центроида j кластера; xi – объект, входящий в состав кластера S, при этом xϵ S, n – количество объектов в j кластере.

На втором этапе итерации t, каждый агент откладывает феромон на рёбрах графа H_{nm}(t-1), соответствующих ребрам построенного граф – решения Hl(t), в количестве пропорциональном функции качества Fl(t). Формула, определяющая количество феромона, которое должен отложить каждый конкретный муравей, выглядит следующим образом:

$$q_{i,j} = \frac{q_{i,j} \cdot F^b}{F^c} \quad (12)$$

где q – количество феромона, которое должен отложить муравей; F^b – лучшее найденное решение; F^c – текущее решение.

После того, как каждый агент сформировал решение и отложил феромон, на третьем этапе итерации t происходит общее испарение феромона на ребрах двудольного графа Hnm(t). Испарение проходит на всех ребрах множества U, при этом коэффициент испарения A задается априорно. Формула для определения количества оставшегося после испарения феромона на ребре имеет вид:

$$\tau_{ij} = \tau_{ij} \cdot (1 - A),$$

где A – коэффициент испарения; τ – количество феромона на i ребре. Данный процесс происходит определенное количество итераций, после чего происходит смена алгоритмов и начинает работу алгоритм летучих мышей.

4. Алгоритм летучих мышей. Большинство видов летучих мышей обладает совершенными средствами эхолокации, которая используется ими для обнаружения добычи и препятствий, а также для обеспечения возможности разместиться в темноте на насесте. Параметры лоцирующего звукового импульса летучих мышей различных видов меняются в широких пределах, отражая их различные охотничьи стратегии. Большинство мышей используют короткие частотно-модулированные в пределах примерно одной октавы сигналы. Некоторые виды не используют частотную модуляцию своих звуковых импульсов [11–18].

Разработанный алгоритм летучих мышей соответствует следующим правилам:

1) все летучие мыши используют эхолокацию, чтобы анализировать расстояние, а также иметь различие между добычей и природными препятствиями;

2) летучие мыши перемещаются случайным образом со скоростью Vi в позиции xi с фиксированной частотой fmin, изменяемой длиной волны λ и громкостью A0, чтобы найти добычу. Они могут автоматически регулировать длину волны испускаемого импульса и интенсивность импульса r[0,1], зависящих от близости цели;

3) громкость изменяется от большего (положительного) A0 к меньшему постоянному значению Amin [11–18].

Начальным этапом является инициализация начальной популяции летучих мышей (решений) размерностью n: X = {x1, x2…, xn}.

Положения всех летучих мышей в начальный момент задаются случайным образом. Каждый поисковый агент обладает следующими характеристиками: текущее положение xi, скорость движения vi, громкость сигнала Ai, частота сигнала ωi, интенсивность сигнала ri. Перемещение каждого агента происходит следующим образом [8, 4–15] (блоки 5, 6 и 7 соответственно):

$$\omega_i = \omega_{min} + (\omega_{max} - \omega_{min}) \cdot \beta, \quad (13)$$

$$v_i^t = v_i^{t-1} + (x_i^t - x_{best}) \cdot w_i, \quad (14)$$

$$x_i^t = x_i^{t-1} + v_i^t, \quad (15)$$

где β – случайная константа от 0 до 1, x_{best} – лучшее глобальное решение, $\omega_i \epsilon [\omega_{min}, \omega_{max}]$.

Для определения новых положений летучих мышей, каждая из них осуществляет локальный поиск в окрестности своего текущего положения путем случайного блуждания (блок 9):

$$x_i^{new} = x_i^{old} + \varepsilon \cdot A^t, \quad (16)$$

где ε – случайное число из интервала [-1, 1], At – средняя громкость всех летучих мышей в текущей популяции.

Изменение параметров звуковых сигналов осуществляется по формулам (блок 12):

$$A_i^{t+1} = \alpha \cdot A_i^t, \quad (17)$$

$$r_i^{t+1} = r_i^0 \cdot (1 - e^{-\gamma t}), \quad (18)$$

где рекомендуемые параметры α = 0.9 и γ = 0.9.

С увеличением числа итераций громкость испускаемых звуковых сигналов будет уменьшаться, а их интенсивность увеличиваться, моделируя этим приближение летучей мыши к цели [14].

В рамках данного алгоритма решение закодировано в виде вектора строки размерностью k × m, где k – количество кластеров, а m – количество объектов в кластере.

На рис. 2 продемонстрировано кодирование решения, где C1 – центроид кластера 1, C2 – центроид кластера 2, Ck – центроид кластера k.

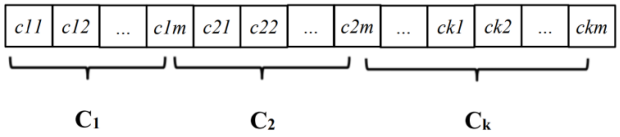

Рис. 2. Кодирование решения для алгоритма летучих мышей

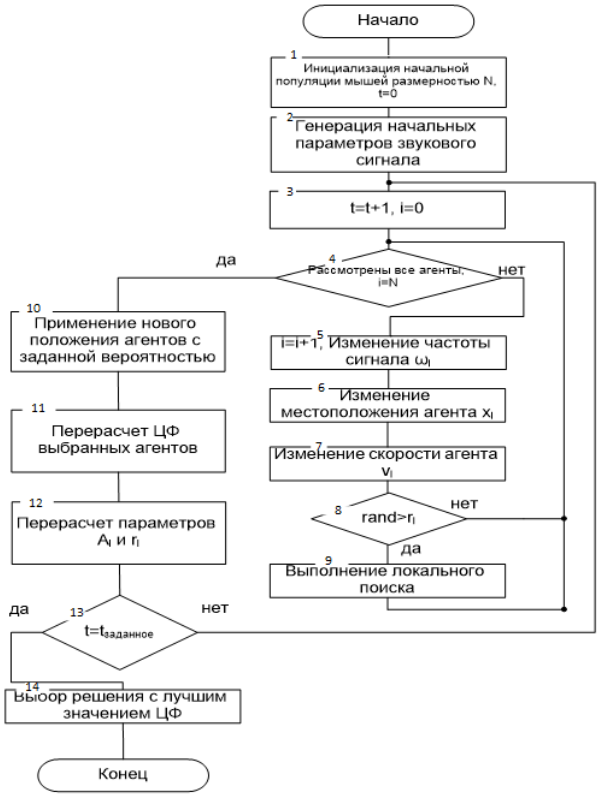

Рис. 3. Схема работы алгоритма летучих мышей

5. Экспериментальные исследования. Экспериментальные исследования проводились на основе работы алгоритмов с разным количеством объектов и кластеров с целью выяснить следующие параметры полученного комбинированного алгоритма:

1) временная сложность алгоритма;
2) эффективность алгоритма по сравнению с другими алгоритмами кластеризации.

Для проведения экспериментов использовались следующие тестовые примеры (бенчмарки) (табл. 1) [10]:

Таблица 1

Описание тестовых примеров

№	Название	Число атрибутов	Число кластеров	Число объектов (объектов в каждом кластере)
1	ArtDataset1	2	4	600(150,150,150,150)
2	ArtDataset2	3	5	250(50,50,50,50,50)
3	Iris	4	3	150(50,50,50)
4	Thyroid	5	3	215(150,35,30)

Временная сложность комбинированного алгоритма состоит из двух частей: временная сложность муравьиного алгоритма проанализирована в работе [2] и составляет O(t*n*2*m), где t – число итераций, n – число агентов, m – количество элементов для кластеризации. Временная сложность алгоритма летучих мышей имеет квадратичный характер и может быть выражена формулой O(α*m2). Таким образом, временная сложность комбинированного биоинспирированного алгоритма кластеризации составляет O(t*n*2*m+ α*m2).

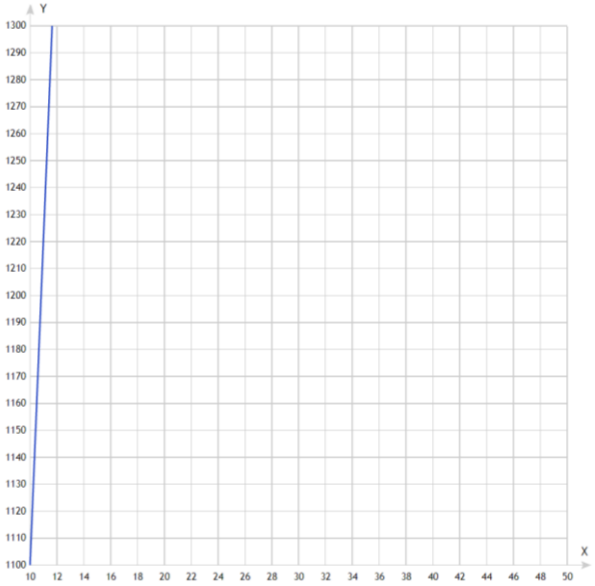

Рис. 4. График временной сложности алгоритма

Алгоритмы сравниваются по критерию суммарного внутрикластерного расстояния, определяющегося как сумма расстояний между каждым объектом и центроидом кластера, в который размещен объект.

Вычисление эффективности алгоритмов производилось с учетом следующих параметров:

1. Муравьиный алгоритм: количество итераций – 100, количество агентов – 50;
2. Алгоритм летучих мышей: количество итераций – 300, количество агентов – 50, коэффициенты α и γ = 0,9, максимальная частота сигнала – k × m, минимальная частота сигнала – 0.

Таблица 2

Сравнение эффективности алгоритмов кластеризации

№	Название теста	k-means	Генетический алгоритм	Разработанный комбинированный алгоритм
1	ArtDataset1	531.529	530.124	527.874
2	ArtDataset2	1728.798	1727.126	1726.153
3	Iris	97.326	96.012	95.656
4	Thyroid	1978.333	1923.586	1866.467

Таким образом, на основании проведенных экспериментов, можно сделать вывод, что разработанный алгоритм в среднем на 5 % эффективнее алгоритма k-means и на 3 % эффективнее генетического алгоритма.

Выводы. Данная работа посвящена решению одной из важнейших задач в области интеллектуального анализа данных – задачи кластеризации. В первом разделе проведен анализ задачи кластеризации. Приведена постановка задачи и основные формулы для ее решения. Определена целевая функция для решения задачи.

Во втором разделе проведен аналитический обзор популярных алгоритмов решения задачи кластеризации, отмечены их достоинства и недостатки. Обоснована актуальность использования биоинспирированных алгоритмов. Задача кластеризации является NP-полной, что обуславливает необходимость использования недетерминированных стохастических методов для ее решения. К таким относятся эволюционные (генетический алгоритм, алгоритм имитации отжига, эволюционный алгоритм), а также активно развивающиеся биоинспирированные алгоритмы, такие как алгоритм роя частиц, муравьиный алгоритм, пчелиный алгоритм, алгоритм светлячков, алгоритм летучих мышей.

В третьем разделе авторами раскрыты основные идеи муравьиного алгоритма для решения задачи кластеризации. Предложен комбинированный алгоритм, состоящий из последовательного применения муравьиного алгоритма и алгоритма летучих мышей, который описан в четвертом разделе. Приведена схемы кодирования решения и схема алгоритма летучих мышей.

Результаты экспериментальных исследований, приведенные в пятом разделе, позволяют сделать вывод, что разработанный алгоритм превосходит известные алгоритмы в среднем на 5 %. Временная сложность составляет O(t*n*2*m+ α*m2).

Полученные результаты планируется в дальнейшем использовать для работы с моделью бустинга, предложенной в работе [14], который работает с несколькими алгоритмами, что позволяет найти лучшее решение во время работы сразу нескольких биоинспирированных алгоритмов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Ka-Chun Wong*. A Short Survey on Data Clustering Algorithms // IEEE Second International Conference on Soft Computing and Machine Intelligence, 2015.
2. *Кравченко Ю.А., Нацкевич А.Н.* Модель решения задачи кластеризации данных на основе использования бустинга алгоритмов адаптивного поведения муравьиной колонии и k-средних // Известия ЮФУ. Технические науки. – 2017. – № 7 (192). – С. 90-102.

3. *Ершов К.С., Романова Т.Н.* Анализ и классификация алгоритмов кластеризации // Новые информационные технологии в автоматизированных системах. – 2016. – Вып. 19. – С. 274-279.
4. *Van D.M. and Engelbrecht A.P.* Data clustering using particle swarm optimization // Proceedings of The Congress on Evolutionary Computation. – 2003. – P. 215-220.
5. *Shelokar P.S., Jayaraman V.K. and Kulkarni B.D.* An Ant Colony Approach for Clustering // Analytica Chimica Acta. – 2004. – Vol. 509, No. 2. – P. 187-195.
6. *Yi-Tung Kao, Erwie Zahara and I-Wei Kao.* A hybridized approach to data clustering // Expert Systems with Applications. – 2008. – Vol. 34, No. 3. – P. 1754-1762.
7. *Changsheng Zhang, Dantong Ouyang and Jiaxu Ning.* An artificial bee colony approach for clustering // Expert Systems with Applications. – 2010. – Vol. 37, No. 7. – P. 4761-4767,
8. *Miao Wan, Lixiang Li, Jinghua Xiao, Cong Wang and Yixian Yang,* Data clustering using bacterial foraging optimization // Journal of Intelligent Information Systems. – 2012. – Vol. 38, No. 2. – P. 321-341.
9. *Senthilnath J., Vipul Das, Omkar S.N. and Mani V.* Clustering using Levy Flight Cuckoo Search // Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications, Advances in Intelligent Systems and Computing. – 2012. – Vol. 202. – P. 65-75.
10. *Jensi R., Wiselin Jiji G*. MBA-LF: a new data clustering method using modified bat algorithm and Levy flight // ICTACT J. Soft Comput. – 2015. – No. 6. – P. 1093-1101.
11. *Кулиев Э.В., Лежебоков А.А., Кравченко Ю.А.* Роевой алгоритм поисковой оптимизации на основе моделирования поведения летучих мышей // Известия ЮФУ. Технические науки. – 2016. – № 7 (180). – С. 53-62.
12. *Частикова В.А., Новикова Е.Ф.* Алгоритм летучих мышей для решения задачи глобальной оптимизации // Научные труды КубГТУ (электронный сетевой политематический журнал). – 2015. – № 2. – URL: http://ntk.kubstu.ru/file/348 (дата обращения: 15.12.2017).
13. *Красношлык Н.А.* Решение задачи глобальной оптимизации модифицированным алгоритмом летучих мышей // Радіоелектроніка, інформатика, управління. – 2015. – № 4 (35). – С. 96-103.
14. *Кравченко Ю.А., Нацкевич А.Н., Курситыс И.О.* Бустинг биоинспирированных алгоритмов для решения задачи кластеризации // Международная конференция по мягким вычислениям и измерениям. – 2018. – Т. 1. – С. 777-780.
15. *Bova V.V., Kravchenko Y.A., Kureichik V.V.* Development of distributed information systems: ontological approach // Advances in Intelligent Systems and Computing. – 2015. – Vol. 349. – P. 113-122.
16. *Кравченко Ю.А.* Технология анализа надежности адаптивных информационных сред // Известия ЮФУ. Технические науки. – 2010. – № 12 (113). – С. 103-108.
17. *Кравченко Ю.А.* Оценка когнитивной активности пользователя в системах поддержки принятия решений // Известия ЮФУ. Технические науки. – 2009. – № 4 (93). – С. 113-117.
18. *Родзин С.И., Курейчик В.В.* Теоретические вопросы и современные проблемы развития когнитивных биоинспирированных алгоритмов оптимизации // Кибернетика и программирование. – 2017. – № 3. – С. 51-79.
19. *Donkuan X. Yingjie T.A.* comprehensive survey of clustering algorithms // Annals of Data Science. – 2015. – Vol. 2, Issue 2. – P. 165-193.
20. *Kravchenko Y.A., Kuliev E.V., Kursitys I.O.:* Information's semantic search, classification, structuring and integration objectives in the knowledge management context problems // In: 8th IEEE International Conference on Application of Information and Communication Technologies (AICT 2016). – IEEE Press, Baku, Azerbaijan, 2016. – P. 136-141.

REFERENCES

1. *Ka-Chun Wong*. A Short Survey on Data Clustering Algorithms, *IEEE Second International Conference on Soft Computing and Machine Intelligence, 2015.*
2. *Kravchenko Yu.A., Natskevich A.N*. Model' resheniya zadachi klasterizatsii dannykh na osnove ispol'zovaniya bustinga algoritmov adaptivnogo povedeniya murav'inoy kolonii i k-srednikh [A model for solving the problem of data clustering based on the use of boosting algorithms of adaptive behavior of ant colony and k-means], *Izvestiya YUFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2017, No. 7 (192), pp. 90-102.

3. *Ershov K.S., Romanova T.N.* Analiz i klassifikatsiya algoritmov klasterizatsii [Analysis and classification of clustering algorithms], *Novye informatsionnye tekhnologii v avtomatizirovannykh sistemakh* [New information technologies in automated systems], 2016, Issue 19, pp. 274-279.
4. *Van D.M. and Engelbrecht A.P.* Data clustering using particle swarm optimization, *Proceedings of the Congress on Evolutionary Computation*, 2003, pp. 215-220.
5. *Shelokar P.S., Jayaraman V.K. and Kulkarni B.D.* An Ant Colony Approach for Clustering, *Analytica Chimica Acta*, 2004, Vol. 509, No. 2, pp. 187-195.
6. *Yi-Tung Kao, Erwie Zahara and I-Wei Kao.* A hybridized approach to data clustering, *Expert Systems with Applications*, 2008, Vol. 34, No. 3, pp. 1754-1762.
7. *Changsheng Zhang, Dantong Ouyang and Jiaxu Ning.* An artificial bee colony approach for clustering, *Expert Systems with Applications*, 2010, Vol. 37, No. 7, pp. 4761-4767,
8. *Miao Wan, Lixiang Li, Jinghua Xiao, Cong Wang and Yixian Yang,* Data clustering using bacterial foraging optimization, *Journal of Intelligent Information Systems*, 2012, Vol. 38, No. 2, pp. 321-341.
9. *Senthilnath J., Vipul Das, Omkar S.N. and Mani V.* Clustering using Levy Flight Cuckoo Search, *Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications, Advances in Intelligent Systems and Computing*, 2012, Vol. 202, pp. 65-75.
10. *Jensi R., Wiselin Jiji G*. MBA-LF: a new data clustering method using modified bat algorithm and Levy flight, *ICTACT J. Soft Comput.*, 2015, No. 6, pp. 1093-1101.
11. *Kuliev E.V., Lezhebokov A.A., Kravchenko Yu.A.* Roevoy algoritm poiskovoy optimizatsii na osnove modelirovaniya povedeniya letuchikh myshey [Swarm algorithm search engine optimization is based on modeling the behavior of bats], *Izvestiya YUFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2016, No. 7 (180), pp. 53-62.
12. *Chastikova V.A., Novikova E.F.* Algoritm letuchikh myshey dlya resheniya zadachi global'noy optimizatsii [The bats algorithm for solving global optimization problems], *Nauchnye trudy KubGTU (elektronnyy setevoy politematicheskiy zhurnal)* [Научные труды КубГТУ (электронный сетевой политематический журнал)], 2015, No. 2. Available at: http://ntk.kubstu.ru/file/348 (accessed 15 December2017).
13. *Krasnoshlyk N.A.* Reshenie zadachi global'noy optimizatsii modifitsirovannym algoritmom letuchikh myshey [The global optimization problem the modified algorithm of bats], *Radioelektronika, informatika, upravlinnya* [Radoactive, Informatics, management], 2015, No. 4 (35), pp. 96-103.
14. *Kravchenko Yu.A., Natskevich A.N., Kursitys I.O.* Busting bioinspirirovannykh algoritmov dlya resheniya zadachi klasterizatsii [Boosting bioinspired algorithms for solving the clustering problem], *Mezhdunarodnaya konferentsiya po myagkim vy-chisleniyam i izmereniyam* [International conference on soft computing and measurements], 2018, Vol. 1, pp. 777-780.
15. *Bova V.V., Kravchenko Y.A., Kureichik V.V.* Development of distributed information systems: ontological approach, *Advances in Intelligent Systems and Computing*, 2015, Vol. 349, pp. 113-122.
16. *Kravchenko Yu.A.* Tekhnologiya analiza nadezhnosti adaptivnykh informatsionnykh sred [Technology reliability analysis of adaptive information environments], *Izvestiya YUFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2010, No. 12 (113), pp. 103-108.
17. *Kravchenko Yu.A.* Otsenka kognitivnoy aktivnosti pol'zovatelya v sistemakh podderzhki prinyatiya resheniy [Evaluation of cognitive activity of the user in decision support systems], *Izvestiya YUFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2009, No. 4 (93), pp. 113-117.
18. *Rodzin S.I., Kureychik V.V.* Teoreticheskie voprosy i sovremennye problemy razvitiya kognitivnykh bioinspirirovannykh algoritmov optimizatsii [Theoretical questions and contemporary problems of the development of cognitive bio-inspired algorithms for optimization], *Kibernetika i programmirovanie* [Cybernetics and programming], 2017, No. 3, pp. 51-79.
19. *Donkuan X. Yingjie T.A.* comprehensive survey of clustering algorithms, *Annals of Data Science*, 2015, Vol. 2, Issue 2, pp. 165-193.
20. *Kravchenko Y.A., Kuliev E.V., Kursitys I.O.:* Information's semantic search, classification, structuring and integration objectives in the knowledge management context problems, *In: 8th IEEE International Conference on Application of Information and Communication Technologies (AICT 2016).* IEEE Press, Baku, Azerbaijan, 2016, pp. 136-141.

Статью рекомендовал к опубликованию д.т.н., профессор В.И. Финаев.

Нацкевич Александр Николаевич – Южный федеральный университет; e-mail: natskevich.a.n@gmail.com; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 88634371651; кафедра систем автоматизированного проектирования; аспирант.

Курситыс Илона Олеговна – e-mail: i.kursitys@mail.ru; кафедра систем автоматизированного проектирования; аспирант.

Natskevich Alexander Nikolaevich – Southern Federal University; e-mail: natskevich.a.n@gmail.com; 44, Nekrasovskiy, Taganrog, 347928, Russia; phone: +78634371651; the department of computer aided design; graduate student.

Kursitys Ilona Olegovna – e-mail: natskevich.a.n@gmail.com; the department of computer aided design, graduate student.