УДК 002.53:004.89 DOI 10.23683/2311-3103-2019-3-32-43

**Ю.А. Кравченко, А.Н. Нацкевич**

# АЛГОРИТМ МОДЕЛИРОВАНИЯ ПОВЕДЕНИЯ МУРАВЬИНОЙ КОЛОНИИ ДЛЯ РЕШЕНИЯ ЗАДАЧИ КЛАСТЕРИЗАЦИИ НА ОСНОВЕ ИСПОЛЬЗОВАНИЯ ДИНАМИЧЕСКОГО КОДИРОВАНИЯ*

*Рассматриваются способы применения модифицированных многоагентных алгоритмов для решения задачи кластеризации. Представлен модифицированный биоинспирированный алгоритм моделирования поведения муравьиной колонии. Также описываются сравнительные характеристики модифицированнного алгоритма и канонического. Представлен аналитический обзор различных перспективных разработок среди классических и современных алгоритмов решения задачи кластеризации, дана оценка их достоинств и недостатков. Рассмотрены особенности представленного модифицированного алгоритма. Эвристика представленного алгоритма заключается в модификации отдельных конструктивных особенностей для таких этапов работы алгоритма, как этап получения первичного решения и этап проведения локального поиска. Также предлагается использовать принцип динамической смены кодировки решений в момент выполнения локального поиска для получения более оптимальных результатов среди глобальных решений. Для этапа получения первичного решения предлагается использовать равномерное покрытие пространства поиска решений, локальный поиск предлагается проводить среди кластеров, содержащих граничные элементы. Проведенные экспериментальные исследования показали, что применение разработанного алгоритма позволяет получать решения не уступающие или превосходящие по качеству решения канонического алгоритма, временные затраты в данном случае остаются такими же или имеют незначительные улучшения.*

*Кластеризация; эволюционное моделирование; роевые алгоритмы; машинное обучение; биоинспирированные алгоритмы.*

**Yu.A. Kravchenko, A.N. Natskevich**

# THE MODEL OF BOOSTING BIOINSPIRED ALGORITHMS FOR SOLVING PROBLEMS OF CLASSIFICATION AND CLUSTERING

*In the article methods of application of boosting models for solving clustering and classification problems are considered, comparative characteristics of these models are described. A boosting model has also been developed to solve the clustering problem. The statement of the problem is given. An analytical review of some promising developments among modern and classical clustering algorithms is presented, their advantages and disadvantages are estimated. A modified boosting algorithm for solving the clustering problem is presented. The approaches of boosting and bagging are compared, the merits and drawbacks of the approaches considered are estimated. The review of algorithms used in the process of boosting is given. As an example of solving the problem of data clustering, a new model for solving optimization problems is presented, based on the use of clustering algorithms weighted set and their boosting based on the ideas of bioinspired algorithms. The heuristic of the proposed boosting algorithm is the use of a probability matrix, which allows a weighted estimation of the learning algorithms quality result to obtain the highest quality of the solution to the clustering problem, and also use weighted data sets containing information on the probability of each individual element occurrence in a particular cluster. The conducted researches showed that the solutions obtained by using the algorithm boosting approach allow to obtain results that are not inferior or superior in quality to the variants obtained by the known algorithms.*

*Clustering; evolutionary modeling; swarm algorithms; machine learning; bioinspired algorithms.*

* Работа выполнена при поддержке РФФИ (проекты: № 17–07–00446, № 18–07–00050).

**Введение.** При решении различных научных задач может возникнуть необходимость решения задачи кластеризации данных. Решение задачи в современных условиях может быть затруднено по ряду причин. Во-первых – высокая скорость роста объема данных. Одним из примеров является статистика IBM, согласно которой каждый год генерируется минимум 2.5 эксабайта данных [2]. Во-вторых – многие наборы данных являются слабоструктурированными, что затрудняет решение задачи кластеризации [1, 3].

Для решения данной задачи было разработано большое количество алгоритмов, которые отличаются друг от друга временными затратами, алгоритмической сложностью и различными особенностями эксплуатации.

Среди современных методов одними из достаточно эффективных являются многоагентные методы, работающие с множеством решений. Например, существует ряд алгоритмов, базирующихся на использовании эволюционного моделирования [4–12]. В процессе работы алгоритмы этого класса генерируют сразу несколько решений, лучшее из которых получается путем интеграции набора решений определенным методом. Например, методом голосования. Основное преимущество этого класса алгоритмов заключается в обеспечении возможности использования моделей распараллеливания. Основные недостатки – достаточно большая алгоритмическая сложность некоторых методов и большое количество начальных параметров, от которых сильно зависит качество полученных решений. [13].

Также стоит отметить классификацию методов кластеризации, полученную такими учеными, как Donkuan, X. Yingjie T., анализ результатов исследований которых представлен в работе [13]. Они разделили имеющиеся алгоритмы кластеризации на две большие группы: классические и современные.

Подобная структура также использовалась в других классификациях [1, 14]

Однако, несмотря на наличие множества алгоритмов решения задачи, невозможно подобрать алгоритм, который способен эффективно решить любой набор данных, поступающий на вход. Таким образом, актуальной становится проблема повышения эффективности решения задачи кластеризации за счет использования многоагентных методов, нацеленных на обработку большого количество решений.

Также известным является тот факт, что решение задачи может сильно зависеть от различных конструктивных особенностей алгоритма. Например, от процесса поиска начального решения или от процесса выполнения локального поиска. Эти процессы могут быть улучшены, например, за счет использования динамической кодировки или за счет использования равномерного распределения центроидов по пространству поиска решений на этапе получения начального решения.

Предложенная модификация алгоритма моделирования муравьиной колонии сводится к улучшению отдельных процессов работы алгоритма. А именно, процесса получения начального решения и процесса выполнения локального поиска.

**1. Постановка задачи кластеризации.** Рассмотрим подробнее задачу кластеризации. Имеется множество объектов. Это множество должно быть разбито на множество кластеров таким образом, чтобы каждый отдельный кластер включал наиболее схожие друг с другом объекты. Схожесть объектов оценивается с помощью выбранной метрики. Одна из известных метрик – евклидова. Количество кластеров может задаваться заранее или определяться алгоритмом в процессе работы. Каждый элемент входного набора данных может быть определен в любой из кластеров.

Пусть $O=\{o_1, o_2, ..., o_{|o|}\}$ – множество объектов набора данных. Каждый объект множества описывается множеством признаков объекта $A = \{a_1, a_2, ..., a_{|A|}\}$. Дано множество кластеров $Y = \{y_1, y_2, ..., y_{|Y|}\}$, по которому необходимо распределить все объекты. Среднее значение всех объектов кластера описывается центрои-

дом $c_i \in C$. Также задана метрика расстояния между объектами $p(x_i, x_j)$. Требуется разбить поступающее на вход множество объектов на непересекающиеся подмножества (кластеры). Также необходимо, чтобы каждый кластер состоял из объектов, близких друг к другу с учетом выбранной метрики схожести.

Таким образом алгоритм кластеризации представляет собой функцию $a: O \to Y$, которая любому объекту $x_i \in X$ ставит в соответствие кластер $y_i \in Y$.

В данной статье в качестве метрики схожести $p$ рассматривается Евклидова метрика. Формула для подсчета несхожести объектов с учетом данной метрики выглядит следующим образом:

$$d = \sqrt{\sum_{k=1}^{|A|}(a_i^k - a_j^k)^2},$$

где $a_j^k$ – $k$ атрибут объекта $o_i$, $a_j^k$ – $k$ атрибут объекта $x_j$.

Для подсчета средних значений элементов конкретного кластера используются центроиды. Формула для определения центроида $j$ кластера выглядит следующим образом:

$$c_j' = \frac{1}{S_j}\sum_{x_i \in S_j}\sum_{a^k \in x} a_i^k + c_j^k,$$

где $a^k_j$ – $k$ атрибут объекта $o_i$, $c^k_j$ – $k$ атрибут центроида $c_j$.

Для представления решения используется вектор параметров $V=\{v_1, v_2, \ldots, v_{|V|}\}$. Полученным решением $V'$ является разбиение множества объектов по множеству кластеров типа $(x_1, y_1), \ldots, (x_n, y_n)$.

Изначально для проведения оценки решения используется вектор варьируемых параметров $X\{x_1, x_2, \ldots, x_{|X|}\}$, каждый элемент которого представляет собой центроид $x_i = c_i$.

Для оценки полученного решения используется следующая целевая функция:

$$F = \frac{P^o}{P^i} \to max,$$

где $P^i$ – среднее внутрикластерное расстояние, $P^o$ – среднее межкластерное расстояние.

Подсчет внутрекластерного расстояния для всех кластеров имеет вид:

$$P^i = \frac{1}{O}\sum_{i=1}^{|O|}\sum_{j=1}^{|C|} p(o_i, c_j) \to min,$$

где $p$ – расстояние с учетом выбранной метрики, $o \in O$ – элемент из набора данных, $c \in C$ – центроид кластера.

Среднее межкластерное расстояние описывает расстояние между центроидами всех классов и определяется по следующей формуле:

$$P^o = \frac{1}{C}\sum_{i=1}^{|C|} p(c_i, c) \to max,$$

где $p$ – расстояние между центроидами с учетом выбранной метрики, $c_i$ – рассматриваемый центроид, $c$ – центроид, относительно которого вычисляется среднее межкластерное расстояние.

**2. Алгоритм моделирования поведения муравьев с динамической колировкой решения.** Введем ряд обозначений, использующихся для формального описания алгоритма.

$S = \{s_1, s_2, \ldots, s_{|S|}\}$ – вектор множества агентов.

$X = \{x_1, x_2, \ldots, x_{|X|}\}$ – вектор варьируемых параметров, использующихся при кодировке решения.

$X_i = \{x_1, x_2, \ldots, x_{|X|}\}$ – вектор варьируемых параметров, определяющих положение агента.

$C = \{c_1, c_2, \ldots, c_{|C|}\}$ – вектор, определяющий множество центроидов.
$F(X)$ – целевая функция
$F(X^*)$ – искомое значение целевой функции.
$A = \{a1, a2, \ldots, a|A|\}$ – вектор, определяющий пространство признаков объекта.
$R^{|A|}$ – $A$ - мерное пространство поиска решений.
$d(o_i, o_j)$ – функция оценки несхожести между объектами с учетом выбранной метрики.

Как описывалось выше, алгоритм содержит ряд модифицированных процессов, направленных на улучшение качества решения задачи кластеризации.

Рассмотрим ряд предложенных модификаций более подробно. Первая модификация касается используемой кодировки полученных решений. Для получения оптимальных решений необходимо использовать оптимальную кодировку агента. В зависимости от выбранной кодировки - агент может описывать как множество центроидов, так и множество отдельных элементов конкретного решения. Рассмотрим три типа кодировки агента.

1. Агент представляет собой множество центроидов $s_j = (c_1, \ldots, c_n)$. Такая кодировка использовалась различными учеными при решении задачи кластеризации начиная с алгоритма PSO [15]. При использовании данной кодировки получается, что каждый агент представляет собой решение, а множество агентов – множество решений. Особенность использования данной кодировки - больший разброс решений для дальнейшей оптимизации. Вариант кодировки является оптимальным в случае, если необходимо произвести эффективный глобальный поиск. В процессе работы алгоритма происходит перемещение всех центроидов в соответствии с формулой передвижения агента, соответствующей конкретному алгоритму.
2. Агент представляет центроид $s_j = c_j$. При условии использования данной кодировки множество агентов представляет собой одно конкретное решение из множества решений. Использование данной кодировки оптимально при локальной оптимизации конкретного найденного глобального решения. Также особенностью данной кодировки является автоматическое регулирование количества кластеров. Количество кластеров в данном случае равно количеству агентов. Разброс полученных решений может быть идентичен первой кодировке с той разницей, что количество получаемых на каждой итерации решений задается пользователем, а не определяется количеством агентов.
3. Агент представляет собой конкретный элемент из множества элементов $s_j = o_j$. В данном случае агенты обрабатывают часть одного решения из множества представленных решений. Подобная техника точно имитирует поведение муравьёв в живой природе при построении кладбищ для очистки пространства в муравьином гнезде [16]. Впервые алгоритм был представлен такими учеными, как Lumer и B. Faieta [17]. Авторами предлагалось использовать в качестве пространства поиска решений *2d* сетку для имитации поведения муравьев. Плюс использования подобного пространства поиска решений – сведение данных любой сложности представления к *2d* для простоты визуализации и более точная имитации поведения муравьев в реальной среде. Основной минус метода – использование большого количества дополнительной памяти для осуществления расчётов. Еще один минус метода – сложность реализации с учетом обработки Big Data.

Для получения более оптимального решения задачи кластеризации предлагается использовать различные кодировки решений для различного типа поиска. Для глобального поиска – первый вариант кодировки, описанной выше, для локального поиска - второй вариант.

Основные положения алгоритма. В каждой точке передвижения, описывающейся вектором центроидов или конкретным центроидом агент (муравей) оставляет феромонную точку $p(x_i)$, интенсивность которой равна $b_0$. Испарение феромона

происходит итеративно в соответствии со степенью устойчивости $p_a$, заданной пользователем. Количество феромона постепенно уменьшается на каждой следующей итерации по следующей формуле:

$$p(x_i) = \begin{cases} p_a * p(x_i), & p_a * p(x_i) > 0 \\ 0, & \text{иначе} \end{cases},$$

где $p_a$ – степень устойчивости феромнона, $p(x_i)$ – количество феромона на координатах $x_i$.

Совокупность всех феромонных точек образует феромонный след.

Все муравьи делятся на муравьев, осуществляющих глобальный поиск $S^g$ и муравьев, осуществляющих локальный поиск $S^l$. Общее количество муравьев $|S| = |S^g| + |S^l|$.

Этап инициализации алгоритма. Создается $|S^g|$ регионов глобального поиска. Кодировка представления агентов определяется множеством центроидов и выглядит следующим образом: $X_i^g = \{c_{1, \ldots,} c_n\}$, *где* $c_i$ – центроид. Создается $|S^l|$ регионов локального поиска. Кодировка агента локального поиска представляет собой отдельный центроид $X_i^l = c_i$. Количество агентов локального поиска может быть задано двумя различными способами. Первый вариант – определение количества пользователем. В данном случае, возможно уменьшение алгоритмической сложности, но есть вероятность, что при наличии большого количества центроидов точность выполнения локального поиска упадет. Второй вариант – устанавливать количество агентов равным количеству центроидов. В данном случае есть риск получения большого количества агентов в случае обработки большого количества данных, но точность нахождения оптимальных локальных решений может сильно возрасти.

Различия с каноническим алгоритмом заключается также в методе задания регионов поиска. В случае решения задачи кластеризации регион поиска представляет собой множество центроидов. Количество центроидов определяется в процессе работы алгоритма. Изначально центроиды создаются таким образом, чтобы покрыть все пространство поиска решений используя при этом определенный интервал.

В качестве примера приведем формула для набора данных, каждый элемент которого содержит два параметра. Такой элемент может быть представлен в двумерном пространстве. Формула для определения позиций кластеров выглядит следующим образом:

$$c_{x,y} = \sum_{x=0}^{a_x^{max}/10} \sum_{y=0}^{a_y^{max}/10} (x * a_x^{max}/a_{del}, y * a_y^{max}/a_{del}),$$

где $c_{x,y}$ – центроид с двумя параметрами, $a_x^{max}$ – максимальное значение $x$ атрибута, $a_y^{max}$ – максимальное значение $y$ атрибута, $a_{del}$ – показатель точности, заданный пользователем. Чем больше это число, тем большим количеством центроидов будет покрыто пространство поиска решений. Поскольку алгоритм является случайным – внутри интервала возможно использования функции случайного присваивания позиции центроида.

Опишем этап оптимизации алгоритма:

1. Производим поиск глобального решения с помощью использования канала стигмергии и агентов из числа $S^g$. Кодировка каждого конкретного агента представляет собой множество центроидов.

2. В каждом из регионов глобального поиска вычислим значение целевой функции $F^*$.

3. Среди регионов выбирается лучшее решение в соответствии с множеством полученных целевых функций.

4. Среди худших решений проводим получение нового решения случайным образом.

5. Производим оптимизацию лучшего решения путем осуществления локального поиска муравьями из числа $S^l$ на основе использования канала стигмергии. Кодировка каждого конкретного агента представляет собой кластер.

6. Если найден локальный регион с лучшим значением целевой функции – переставляем муравья $s_i^g$ в данный регион и инициализируем новую феромонную точку.

7. Моделируем испарение феромона.

Этап завершения оптимизации включает в себя проверку условия окончания итераций. Если условие было выполнено – останавливаем алгоритм. Если нет – повторяем этап оптимизации.

Рассмотрим подробнее этап глобального поиска:

1. Случайным образом выбираем муравья $s_i$ из числа $S^g$.

2. С учетом интереса муравья вычисляем координаты центра тяжести путем поиска среди активных феромонных точек. Формула для вычисления центра тяжести выглядит следующим образом:

$$W_i = \sum_{j,k} \omega_{i,j,k} * X_{j,k}, \;\; j \in (1, \dots, |S^g|), \;\; k \in (1, \dots, |\varphi_j|),$$

где $X_{j,k}$ – k-я точка феромонного следа $f_j$.

$$\omega_{i,j,k} = \frac{\omega_{i,j,k}}{\sum_{i=1}^{|S^g|} \sum_{m=1}^{|\varphi_j|} \omega_{i,l,m}},$$

где $\omega_i$ нормированный интерес i-го муравья к k-й точке феромонного следа $f_j$.

3. Координаты нового положения муравья вычисляются по следующей формуле:

$$X_i' = X_i + V_i,$$

где $V_i$ – шаг в направлении ($X_i$ – $\omega_i$). Величина шага измеряется по формуле $abs(b_{beg} - b_{inc})$, где $b_{beg}$ – начальная длина шага, $b_{inc}$ – инкремент шага в соответствии с итерацией.

Этап локального поиска с учетом стигмергии осуществляется следующим образом:

1. Инициализируем феромонные точки с учетом выбранной кодировки. Координаты феромонных точек равны координатам центроидов.

2. случайным образом выбираем муравья $s_i$ из числа $|S^l|$.

3. С учетом интереса данного муравья вычисляем координаты центра тяжести феромонных точек.

4. Вычислим координаты нового положения муравья.

Рассмотрим подробнее третий пункт. В случае нахождения глобального решения определение скорости движения каждого отдельного муравья определяется в соответствии с каноническим алгоритмом. Однако, при осуществлении локального поиска есть определенные различия.

В отличии от канонического алгоритма, в соответствии с которым центр тяжести феромонных точек вычислялся по всем активным феромонным точками с учетом текущей позиции муравья, в данном алгоритме, наиболее «интересными» точками для муравья являются те точки, которые включают в себя наиболее большое количество граничных элементов.

Рассмотрим подробнее понятие граничных элементов. Граничными называются такие элементы, которые могут быть отнести к двум и более центроидам в приблизительно равной степени. Использование подобной концепции становится актуальной в случае использования алгоритма создания равномерного покрытия пространства поиска решений цетроидами.

Предположим, агент, кодировка которого представляет собой центроид, оставил феромонную точку в позиции центроида $c_i$. В случае, если в окрестности данного центроида существует еще ряд центроидов, которые имеют с ним граничные элементы – феромонная точка является для муравьёв, представляющих собой граничные центроиды, более интересной.

Формула определения того, является ли элемент граничным, выглядит следующим образом:

$$f(x_{i,j,k}) = \begin{cases} |p(x_i, c_j) - p(x_i, c_j)| > x_f, & 1 \\ \text{иначе}, & 0 \end{cases},$$

где $x_i$ – рассматриваемый элемент, $c_i$ – $i$ центроид, $c_j$ – $j$ центроид, $x_f$ – степень схожести, определяющая, насколько вероятным является отнесение элемента сразу нескольким кластерам. Для корректной работы параметры должны быть нормализованы. Число задается пользователем, однако, в последующих статьях автором будет проведена работа по автоматической адаптации данного параметра. Функция возвращает значение «истина» (1) в случае, если элемент является граничным и «ложь» (0) – во всех остальных случаях.

В случаях, если кластер не имеет граничных элементов, скорее всего, позиция центроида определена верно и все элементы, входящие в состав кластера, имеют схожие параметры в соответствии с заданной метрикой.

Введем множество, определяющее совокупность муравьев, включающих в свой состав агентов, которые имеют граничные элементы относительно центроида $c_i$, который определяет текущий муравей: $h_i = \{s_i^1, \ldots, s_i^n\}$. Совокупность всех множеств муравьев, содержащих взаимосвязи между каждыми двумя или более конкретными центроидами обозначим через $H = \{h_i, \ldots, h_{|C|}\}$, где $|C|$ – общее количество муравьев, равное количеству центроидов при условии поиска локального решения.

Как говорилось выше, для определения общей скорости передвижения муравья, выбирается ряд центроидов в соответствии с интересом конкретного муравья. Приведем формулу определения центра тяжести в соответствии с интересом конкретного муравья.

$$W_i = \sum_{j,k} \omega_{i,j,k} X_{j,k},\ j \in [1, \ldots, |a_i|], k \in [1, \ldots, |\varphi_i|],$$

где $X_{j,k}$ – $k$ феромонная точка из множества точек, определяющих муравьёв, содержащих центроиды с граничными элементами $A$, $\omega_{i,j,k}$ – нормированный интерес муравья. Определяется по следующей формуле:

$$\omega_{i,j,k} = \frac{\omega_{i,j,k}}{\sum_{i=1}^{|a_i|} \sum_{m=1}^{|\varphi_i|} \omega_{i,j,m}}.$$

Координаты нового положения муравья вычисляются по формуле:

$$X_i' = X_i + (V_i),$$

где $X_i$ – текущая позиция муравья $s_i$, $V_i$ – шаг в направлении $(X_i - W_i)$. Величина шага измеряется таким же образом, как и в случае получения глобального решения. Как и в случае с вычислением скорости агента при осуществлении глобального поиска - величина шага измеряется по формуле $abs(b_{beg} - b_{inc})$, где $b_{beg}$ – начальная длина шага, $b_{inc}$ – инкремент шага в соответствии с итерацией.

Интерес муравья $s_i$ к $k$-й точке феромонного следа $a_i$ определяется по следующей формуле:

$$\omega_{i,j,k} = \frac{\rho}{2} \theta \exp(-\rho_{i,j,k}),\ i,j \in (1, \ldots, |h_i|),\ k \in (1, \ldots, \varphi_j),$$

где $p(t)$ – среднее текущее расстояние между двумя муравьями популяции, $\theta$- количество феромона в конкретной точки $p_{i,j,k}$ – евклидово расстояние между регионом $X_i$ и той же феромонной точкой.

Рассмотрим подробнее процесс определения схожести центроидов с точки зрения количества граничных элементов. Формула для объединения центроидов в один кластер выглядит следующим образом:

$$f(x_g^{i,j}) = \begin{cases} x_g^{i,j} > x_p, & 1 \\ \text{иначе}, & 0 \end{cases},$$

где $x_g^{i,j}$ – процент граничных элементов от общего количества элементов, находящихся близко к центроидам $c_i$ и $c_j$, $x_p$ – заранее определенный процент количества граничных элементов центроида, при достижении которого считается, что оба центроида определяют один кластер.

В качестве условия окончания итераций можно использовать достижение изначально заданного числа итераций или стагнацию вычислительного процесса в течении определенного количества итераций.

**3. Экспериментальные исследования.** Основная цель проведенного исследования эффективности – проверка качества решений, полученных с помощью разработанной модификации муравьиного алгоритма при решении задачи кластеризации данных. Для проверки использовался ряд бенчмарков с заранее известным оптимумом.

Экспериментальные исследования проводились на вычислительной машине, обладающей мощностью порядка 140 Гигафлопс.

Основные используемые параметры алгоритма: количество итераций алгоритма $t$ – 5000, параметр затухания феромона – 0.99, количество агентов, участвующих в процессе поиска решений $|S|$ - 50. Из них для осуществление глобального поиска используется $|S^g| = 35$, для осуществления локального поиска используется $|S^l| = 15$.

В процессе экспериментов разработанный алгоритм сравнивался с каноническим циклическим муравьиным алгоритмом и с алгоритмом роя частиц. Алгоритмическая сложность разработанного алгоритма вычисляется следующим образом: $O(|S^g|*|O|^2 + |S^l|*|O|^2)$, что в целом может быть сведено к $O(|S|^2 + |O|^2)$.

График зависимости времени от количества объектов в наборе данных представлен на рис. 1:


*Рис. 1. Временная сложность алгоритма*

Приведем табл. 1, демонстрирующую сравнение характеристик времени работы алгоритмов.

Таблица 1

**Сравнение характеристик времени работы алгоритмов**

| Размерность базы данных | Время работы модифицированного алгоритма | Канонический муравьиный алгоритм | Алгоритм роя частиц |
|---|---|---|---|
| 1000 | 1.59 | 1.61 | 1.59 |
| 2000 | 1.64 | 1.68 | 1.65 |
| 5000 | 3.82 | 3.90 | 3.92 |
| 10000 | 11.14 | 11.69 | 11.57 |
| 20000 | 22.80 | 23.60 | 23.50 |

Кроме временных параметров алгоритмы сравнивались по критерию процентного количество неверно кластеризованных решений ici (incorrectly clustered instances). Результаты сравнения приведены в табл. 2.

Таблица 2

**Сравнения качества полученных решений**

| Набор данных | Число кластеров | Алгоритм роя частиц (ici) | Канонический алгоритм (ici) | Модифицированный алгоритм (ici) |
|---|---|---|---|---|
| Ионосфера | 10 | 28.5 | 17.6 | 7.2 |
| Ионосфера | 20 | 26.3 | 16.5 | 5.4 |
| Ирис | 10 | 22.3 | 9.3 | 4.7 |
| Ирис | 20 | 18.6 | 7.3 | 3.4 |
| Ирис | 30 | 15.1 | 5.3 | 1.1 |

В процессе проведения экспериментальных исследований было выявлено, что разработанный алгоритм дает небольшое временное преимущество (в пределах 1 %), которое может быть увеличено за счет оптимизации некоторых особенностей алгоритма.

Сравнительный анализ качества работы алгоритмов, результаты которого приведены в табл. 2 показал, что решения, полученные с помощью использования модифицированного алгоритма, отличаются в лучшую сторону по сравнению с аналогами.

**Заключение.** В результаты проведенного аналитического исследования можно сделать вывод, что модификация отдельных процессов работы алгоритма может дать значительный прирост качества найденных решений при соизмеримых временных затратах.

Для проведения экспериментальных исследований была разработана программная реализация описанного алгоритма для решения задачи кластеризации и проведено ее сравнение с каноническим алгоритмом и алгоритмом роя частиц по таким параметрам, как время работы и процент некорректно кластеризованных объектов данных (*ici*).

Дальнейшим улучшением алгоритма станет уменьшение параметров, которые необходимо задавать пользователю. Например, для определения граничных центроидов используется параметр $x_f$, представляющий собой степень схожести между двумя элементами, при достижении которой элементы считаются граничными. Алгоритм может работать более эффективно, если число будет определяться автоматически в процессе работы. Данная оптимизация будет проведена автором в дальнейшем. Также стоит рассмотреть попытку автоматизации задания изначального количества феромона. Данный параметр также задается пользователем.

Кроме того, время работы алгоритма может быть уменьшено в случае использования параллельных парадигм программирования, как на этапе глобального поиска, так и на этапе локального поиска [18–20]. Подобное исследование планируется провести в будущем.

Стоит отметить, что разработанный алгоритм подходит для оценки качества существующей классификации. В данном случае, если с учетом существующей классификации, поступающие на вход элементы классифицируются неверно – есть вероятность, что задачу стоит свести к задаче кластеризации и игнорировать существующую классификацию.

БИБЛИОГРАФИЧЕСКИЙСПИСОК


1. *Ka-Chun Wong.* A Short Survey on Data Clustering Algorithms // IEEE Second International Conference on Soft Computing and Machine Intelligence, 2015.
2. IBM Consumer products industry blog. Industry insights. – https://www.ibm.com/blogs/ insights-on-business/consumer-products/2-5-quintillion-bytes-of-data-created-every-day-how-does-cpg-retail-manage-it/ (дата обращения: 20.05.2019).
3. *Mayr A, Binder H, Gefeller O, Schmid M.* The Evolution of Boosting Algorithms – From Machine Learning to Statistical Modelling // Methods Inf. Med. – 2014. – Vol. 53. – P. 419-427.
4. *Зайцев А.А., Курейчик В.В., Полупанов А.А.* Обзор эволюционных методов оптимизации на основе роевого интеллекта // Известия ЮФУ. Технические науки. – 2010. – № 12 (113). – С. 7-12.
5. *Kureichik V.V., Kravchenko Y.A.* Bioinspired algorithm applied to solve the travelling salesman problem // World Applied Sciences Journal. – 2013. – Vol. 22, No. 12. – P. 1789-1797.
6. *Gladkov L.A., Kureichik V.V., Kravchenko Y.A.* Evolutionary algorithm for extremal subsets comprehension in graphs // World Applied Sciences Journal. – 2013. – Vol. 27, No. 9. – P. 1212-1217.
7. *Курейчик В.В., Курейчик В.М., Сороколетов П.В.* Анализ и обзор моделей эволюции // Известия РАН. Теория и системы управления. – 2007. – № 5. – С. 114-126.
8. *Родзин С.И., Курейчик В.В.* Состояние, проблемы и перспективы развития биоэвристик // Программные системы и вычислительные методы. – 2016. – № 2. – С. 158-172.
9. *Курейчик В.В., Бова В.В., Курейчик Вл.Вл*. Комбинированный поиск при проектировании // Образовательные ресурсы и технологии. – 2014. – № 2 (5). – С. 90-94.
10. *Курейчик В.В., Курейчик Вл.Вл.* Биоиспирированный поиск при проектировании и управлении // Известия ЮФУ. Технические науки. – 2012. – № 11 (136). – С. 178-183.
11. *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
12. *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.
13. *Donkuan, X. Yingjie T.* A comprehensive survey of clastering algorithms // Annals of Data Science. – 2015. – Vol. 2, Issue 2. – P. 165-193.
14. Survay of clustering algorithms. – https://scholarsmine.mst.edu/cgi/viewcontent. cgi?article=1763&context=ele_comeng_facwork (дата обращения: 25.05.2019).
15. Data clustering using particle swarm optimization. – http://citeseerx.ist.psu.edu/viewdoc/ download?doi=10.1.1.468.819&rep=rep1&type=pdf (дата обращения: 20.05.2019).
16. Chretien. Organisation Spatiale du Materiel Provenant de L'excavation du nidchez Messor Barbarus et des Cadavres d'ouvrieres chez Lasius niger (Hymenopterae: Formicidae), Ph.D. thesis, Universite Libre dr Bruxelles, 1996.
17. *Lumer and B. Faieta.* Diversity and adaptation in Populations of ClusteringAnts // in Third international Conference on simulation of Adaptive Behavior: Fromanimals to Animats. – MIT Press, Cambridge, 1994. – Vol. 3. – P. 489-508.
18. *Курейчик В.М., Курейчик В.В., Родзин С.И.* Модели параллелизма эволюционных вычислений // Вестник Ростовского государственного университета путей сообщения. – 2011. – № 3 (43). – С. 93-97.

19. *Курейчик В.М., Курейчик В.В., Родзин С.И., Гладков Л.А.* Основы теории эволюционных вычислений. – Ростов-на-Дону: ЮФУ, 2010.
20. *Родзин С.И., Курейчик В.В.* Теоретические вопросы и современные проблемы развития когнитивных биоинспирированных алгоритмов оптимизации // Кибернетика и программирование. – 2017. – № 3. – С. 51-79.

REFERENCES

1. *Ka-Chun Wong.* A Short Survey on Data Clustering Algorithms, *IEEE Second International Conference on Soft Computing and Machine Intelligence*, *2015*.
2. IBM Consumer products industry blog. Industry insights. Available at: https://www.ibm.com/blogs/insights-on-business/consumer-products/2-5-quintillion-bytes-of-data-created-every-day-how-does-cpg-retail-manage-it/ (accessed 20 May 2019).
3. *Mayr A, Binder H, Gefeller O, Schmid M.* The Evolution of Boosting Algorithms – From Machine Learning to Statistical Modelling, *Methods Inf. Med*., 2014, Vol. 53, pp. 419-427.
4. *Zaytsev A.A., Kureychik V.V., Polupanov A.A.* Obzor evolyutsionnykh metodov optimizatsii na osnove roevogo intellekta [Overview of evolutionary optimization techniques based on swarm intelligence], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2010, No. 12 (113), pp. 7-12.
5. *Kureichik V.V., Kravchenko Y.A.* Bioinspired algorithm applied to solve the travelling salesman problem, *World Applied Sciences Journal*, 2013, Vol. 22, No. 12, pp. 1789-1797.
6. *Gladkov L.A., Kureichik V.V., Kravchenko Y.A.* Evolutionary algorithm for extremal subsets comprehension in graphs, *World Applied Sciences Journal*, 2013, Vol. 27, No. 9, pp. 1212-1217.
7. *Kureychik V.V., Kureychik V.M., Sorokoletov P.V.* Analiz i obzor modeley evolyutsii [Analysis and review of models of evolution], *Izvestiya RAN. Teoriya i sistemy upravleniya* [News of Russian Academy of Sciences. Theory and control systems], 2007, No. 5, pp. 114-126.
8. *Rodzin S.I., Kureychik V.V.* Sostoyanie, problemy i perspektivy razvitiya bioevristik [Status, problems and prospects of bio-heuristics development], *Programmnye sistemy i vychislitel'nye metody* [Software systems and computational methods], 2016, No. 2, pp. 158-172.
9. *Kureychik V.V., Bova V.V., Kureychik Vl.Vl.* Kombinirovannyy poisk pri proektirovanii [Combined search in design], *Obrazovatel'nye resursy i tekhnologii* [Educational resources and technologies], 2014, No. 2 (5), pp. 90-94.
10. *Kureychik V.V., Kureychik Vl.Vl.* Bioispirirovannyy poisk pri proektirovanii i upravlenii [Biospherology search in the design and management], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2012, No. 11 (136), pp. 178-183.
11. *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 [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
12. *Kravchenko Yu.A., Natskevich A.N., Kursitys I.O.* Busting bioinspirirovannykh algoritmov dlya resheniya zadachi klasterizatsii [Boosting of bioinspired algorithms for solving the clustering problem], *Mezhdunarodnaya konferentsiya po myagkim vychisleniyam i izmereniyam* [International conference on soft computing and measurements], 2018, Vol. 1, pp. 777-780.
13. *Donkuan, X. Yingjie T.* A comprehensive survey of clastering algorithms, *Annals of Data Science*, 2015, Vol. 2, Issue 2, pp. 165-193.
14. Survay of clustering algorithms. Available at: https://scholarsmine.mst.edu/cgi/viewcontent.cgi?article=1763&context=ele_comeng_facwork (accessed 25 May 2019).
15. Data clustering using particle swarm optimization. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.468.819&rep=rep1&type=pdf (accessed 20 May 2019).
16. Chretien. Organisation Spatiale du Materiel Provenant de L'excavation du nidchez Messor Barbarus et des Cadavres d'ouvrieres chez Lasius niger (Hymenopterae:Formicidae), Ph.D. thesis, Universite Libre dr Bruxelles, 1996.
17. *Lumer and B. Faieta.* Diversity and adaptation in Populations of ClusteringAnts, *in Third international Conference on simulation of Adaptive Behavior: Fromanimals to Animats*. MIT Press, Cambridge, 1994, Vol. 3, pp. 489-508.

18. *Kureychik V.M., Kureychik V.V., Rodzin S.I.* Modeli parallelizma evolyutsionnykh vychisleniy [Models of parallelism of evolutionary calculations], *Vestnik Rostovskogo gosudarstvennogo universiteta putey soobshcheniya* [Bulletin of Rostov state University of railway engineering], 2011, No. 3 (43), pp. 93-97.
19. *Kureychik V.M., Kureychik V.V., Rodzin S.I., Gladkov L.A.* Osnovy teorii evolyutsionnykh vychisleniy [Fundamentals of the theory of evolutionary computation]. Rostov-on-Don: YuFU, 2010.
20. *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.

Статью рекомендовал к опубликованию к.т.н. С.Г. Буланов.

**Кравченко Юрий Алексеевич** – Южный федеральный университет; e-mail: yakravchenko@sfedu.ru; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 88634371651; кафедра систем автоматизированного проектирования; доцент.

**Нацкевич Александр Николаевич** – e-mail: natskevich.a.n@gmail.com; кафедра систем автоматизированного проектирования; аспирант.

**Kravchenko Yury Alekseevich** – Southern Federal University; e-mail: yakravchenko@sfedu.ru; 44, Nekrasovskiy lane, Taganrog, 347928, Russia; phone: +78634371651; the department of computer aided design; associate professor.

**Natskevich Alexander Nikolaevich** – e-mail: natskevich.a.n@gmail.com; the department of computer aided design; graduate student.

УДК 004.032 DOI 10.23683/2311-3103-2019-3-43-50

**В.С. Потапов**

## РЕАЛИЗАЦИЯ АЛГОРИТМА ПРЕОБРАЗОВАНИЯ КЛАССИЧЕСКОГО ИЗОБРАЖЕНИЯ В КВАНТОВОЕ СОСТОЯНИЕ, ВЫДЕЛЕНИЯ ГРАНИЦ И ПРЕОБРАЗОВАНИЯ ПОЛУТОНОВОГО ИЗОБРАЖЕНИЯ В БИНАРНОЕ*

*Данная статья посвящена решению задачи исследования и разработки методов функционирования квантовых алгоритмов и моделей квантовых вычислительных устройств. Квантовый алгоритм, реализованный в работе, позволяет произвести преобразование классического изображения в квантовое состояние, выделения границ и преобразование полутонового изображения в бинарное, показывает возможности квантовой теории информации в интерпретации классических задач. Целью работы является компьютерное моделирование квантового алгоритма для решения задачи преобразования классического изображения с использованием квантовых вычислительных средств и методов, изучение существующих алгоритмов распознавания образов и создание эффективной модели распознавания с помощью свойств и методов квантовых вычислений. Данная статья посвящена решению задачи исследования и разработки методов функционирования квантовых алгоритмов и моделей квантовых вычислительных устройств. Актуальность данных исследований заключается в математическом и программном моделировании и реализации квантового алгоритма для решения классов задач классического характера. Научная новизна данного направления в первую очередь выражается в постоянном обновлении и дополнении поля квантовых исследований по ряду направлений, а компьютерная симуляция квантовых физических явлений и особенностей слабо освещена в мире. В настоящее время во многих передовых странах мира интенсивно ведутся научно-исследовательские работы по разработке и созданию квантовых компьютеров и их программного обеспечения, наблюдается*

* Работа выполнена при финансовой поддержке РФФИ, грант № НК 19-07-01082.