УДК 519.6

© 2020 г. **И.П. Рожнов,**
Л.А. Казаковцев, д-р техн. наук
(Сибирский университет науки и технологий, Красноярск)

ПОДХОД К РАЗРАБОТКЕ АЛГОРИТМОВ АВТОМАТИЧЕСКОЙ ГРУППИРОВКИ НА ОСНОВЕ ПАРАМЕТРИЧЕСКИХ ОПТИМИЗАЦИОННЫХ МОДЕЛЕЙ

Предложен новый подход к разработке алгоритмов автоматической группировки, основанных на параметрических оптимизационных моделях, с комбинированным применением алгоритмов поиска с чередующимися рандомизированными окрестностями и жадных агломеративных эвристических процедур. Показано, что новые алгоритмы, разработанные с использованием нового подхода, позволяют получать более точный и стабильный результат (по достигаемому значению целевой функции) в сравнении с известными алгоритмами автоматической группировки за фиксированное время, позволяющее использовать алгоритмы в интерактивном режиме принятия решений для практических задач.

Ключевые слова: кластеризация, автоматическая группировка, k-средних, k-медоид, СЕМ-алгоритм.

DOI: 10.22250/isu.2020.63.24-37

Введение

В связи с ускоренным ростом объемов данных растет и потребность в современных средствах и системах сбора, хранения и обработки массивов данных, вследствие чего увеличивается их многообразие. Все возрастающее использование массивов данных большой размерности стимулирует повышенный интерес к разработке и применению методов и средств обработки и анализа этих массивов. Одним из перспективных направлений является кластерный анализ, который позволяет упорядочивать (группировать) объекты в однородные группы (кластеры), а решение задачи автоматической группи-

ровки (кластеризации) сводится к разработке алгоритма, способного обнаружить эти группы без использования предварительно маркированных данных.

Существуют производственные задачи автоматической группировки объектов, которые должны быть решены сравнительно быстро, при этом результат должен быть таким, чтобы известными методами его трудно было бы улучшить без значительного увеличения временных затрат [1].

Анализ существующих проблем применения методов автоматической группировки объектов, к которым предъявляются высокие требования по точности и стабильности результата, показывает дефицит алгоритмов, способных выдавать за фиксированное время результаты, которые было бы трудно улучшить известными методами и которые бы обеспечивали стабильность получаемых результатов при многократных запусках алгоритма. При этом известные алгоритмы, удовлетворяющие таким требованиям (например, метода жадных эвристик [2 – 4]), требуют значительных вычислительных затрат. Отмечая некоторый дефицит компромиссных по качеству результата и времени счета методов автоматической группировки (под качеством будем понимать точность – близость значения целевой функции к глобальному оптимуму), в настоящем исследовании ставится задача разработать усовершенствованные алгоритмы для задач автоматической группировки, в которых предъявляются высокие требования к точности и стабильности результата [1, 2].

Алгоритмы автоматической группировки

Современные методы кластерного анализа предлагают широкий выбор средств выявления разнородных по совокупности параметров групп. В основе широкого круга методов, – таких, как k -средних [5], k -медоид [6], а также метод автоматической группировки, основанный на решении p -медианной задачи [7], лежит идея минимизации расстояний между объектами одной группы (кластера). Наиболее распространенным из подобных методов является метод k -средних (k -means). Собственно, алгоритм k -средних является алгоритмом локальной оптимизации, и его результат зависит от выбора начальных значений (усредненных параметров центров или центроидов групп – кластеров). В то же время метод выявления различных по параметрам групп объектов должен давать воспроизводимые результаты.

Задача k -средних состоит в нахождении таких k центров кластеров $X_1 \dots X_k$ в d -мерном пространстве, чтобы сумма квадратов расстояний от них

до заданных точек A_i ($A_1, \dots, A_N$) была минимальна

$$\arg \min F(X_1, \dots, X_k) = \sum_{i=1}^N \min_{j \in \{1, k\}} \|X_j - A_i\|^2. \quad (1)$$

Одноименный алгоритм последовательно улучшает известное решение, позволяя найти локальный минимум. Это простой и быстрый алгоритм, применимый к широчайшему классу задач. Алгоритм имеет ограничения – в начале решения необходимо задать число групп k , на которые разбиваются объекты, а результат сильно зависит от начального решения, как правило, выбираемого случайно.

Алгоритм 1.1 Алгоритм k-средних [5]

Дано: векторы данных $A_1 \dots A_N$, k начальных центров кластеров $X_1 \dots X_k$

выполнять

1. Составить кластер C_i векторов данных для каждого центра X_i так, чтобы для каждого вектора данных его центр был ближайшим.
2. Рассчитать новое значение центра X_i для каждого кластера,

пока шаги 1-2 приводят к каким-либо изменениям.

Алгоритм для задачи k-медоид – Partitioning Around Medoids (PAM) – был предложен Кауфманом (Leonard Kaufman) и Руссивом (Peter J. Rousseeuw) [6]. Он похож на алгоритм k-средних: работа обоих основана на попытках минимизировать ошибку, но PAM работает с медоидами – объектами, являющимися частью исходного множества и представляющими группу, в которую они включены, а k-means работает с центроидами – искусственно созданными объектами, представляющими кластер. Алгоритм PAM разделяет множество из N объектов на k кластеров (N и k являются входными данными алгоритма). Алгоритм работает с матрицей расстояний, его цель – минимизировать расстояние между представителями каждого кластера и его членами.

Алгоритм 1.2 PAM-процедура [6]

Фаза Build:

1. Выбрать k объектов в качестве медоид.
2. Построить матрицу расстояний, если она не была задана.
3. Отнести каждый объект к ближайшей медоиде.

Фаза Swap:

4. Для каждого кластера найти объекты, снижающие суммарное расстояние, и если такие объекты есть, выбрать те, которые снижают его сильнее всего, в качестве медоид.
5. Если хотя бы одна медоид поменялась, вернуться к шагу 3, иначе завершить алгоритм.

В число популярных методов автоматической группировки входит и алгоритм Expectation Maximization (ЕМ-алгоритм – максимизация математи-

ческого ожидания) [8]. Основная идея алгоритма состоит в искусственном введении вспомогательного вектора скрытых переменных, что сводит сложную оптимизационную задачу к двум шагам:

1) Е-шаг – последовательность итераций по пересчету скрытых переменных по текущему приближению вектора параметров;

2) М-шаг – максимизация правдоподобия (для нахождения следующего приближения вектора).

Классификационный ЕМ-алгоритм (Classification Expectation Maximization – CEM) – модификация ЕМ-алгоритма работает по принципу четкой классификации данных выборки. В этом случае каждый объект относится к единственному кластеру. CEM-алгоритм почти совпадает с другой модификацией – SEM (Stochastic EM), только у первого на каждом шаге вводится детерминированное правило, что данные относятся лишь к одному кластеру, для которого вычислили максимальную апостериорную вероятность. Таким образом, CEM-алгоритм в отличие от ЕМ решает задачу четкой кластеризации [9].

Для дальнейшего изложения каждую из процедур – k -средних, k -медоид и CEM – обозначим как двухшаговый алгоритм локального поиска. Тем более, что сами процедуры k -средних и k -медоид являются, по сути, алгоритмами поиска с чередующимися окрестностями. Далее при решении задач k -средних в качестве двухшагового алгоритма локального поиска будет реализован Алгоритмом 1.1, соответственно для k -медоид – Алгоритмом 1.2 и максимизации функции правдоподобия – CEM-алгоритмом.

Методы локального поиска получили дальнейшее развитие в метаэвристиках (методах оптимизации, многократно использующих простые правила или эвристики для достижения оптимального или субоптимального решения, характеризующихся большей устойчивостью). Поиск с чередующимися окрестностями (VNS – Variable Neighborhoods Search) – популярный метод решения задач дискретной оптимизации Н. Младеновича и П. Хансена [10], который позволяет находить хорошие субоптимальные решения достаточно больших задач автоматической группировки. Идея состоит в систематическом изменении вида окрестности в ходе локального поиска. Гибкость и высокая эффективность объясняют ее конкурентоспособность при решении NP-трудных задач.

Обозначим через N_k , $k = 1, \dots, k_{max}$ конечное множество видов окрестностей, предварительно выбранных для локального поиска. Метод с чередующимися окрестностями опирается на то, что локальный минимум в одной окрестности не обязательно является локальным минимумом в другой

окрестности, при этом глобальный минимум является локальным в любой окрестности. Кроме того, в среднем локальные минимумы ближе к глобальному, чем случайно выбранная точка, и они расположены близко друг к другу. Это позволяет сузить область поиска глобального оптимума, используя информацию об уже обнаруженных локальных оптимумах. Эта гипотеза лежит в основе различных операторов скрещивания (crossover) для генетических алгоритмов и других подходов.

Основная схема локального поиска с чередующимися окрестностями (VNS).

Алгоритм 2 VNS (Variable Neighborhoods Search) [10]

Шаг 1. Выбрать окрестности O_k , $k = 1, \dots, k_{max}$ и начальную точку x .

Шаг 2. Повторять, пока не выполнен критерий остановки.

2.1. $k = 1$.

2.2. Повторять до тех пор, пока $k \leq k_{max}$:

2.2.1. случайно выбрать точку $x' \in O_k(x)$;

2.2.2. применить локальный спуск с начальной точки x' , не меняя координат, по которым x и x' совпадают. Полученный локальный оптимум обозначается x'' ;

2.2.3. если $F(x'') < F(x)$, то полагается $x = x''$, $k = 1$, иначе $k = k + 1$.

Критерием остановки может служить максимальное время счета или максимальное число итераций без смены лучшего найденного решения. При решении задач большой размерности сложность выполнения одной итерации становится весьма значительной и требуются новые подходы для разработки эффективных методов локального поиска.

Повысить точность методов автоматической группировки позволяет новый подход к разработке алгоритмов автоматической группировки, основанных на параметрических оптимизационных моделях, с комбинированным применением алгоритмов поиска с чередующимися рандомизированными окрестностями и жадных агломеративных эвристических процедур.

Алгоритмы автоматической группировки метода жадных эвристик

Жадная агломеративная эвристическая процедура для задачи k -средних и аналогичных задач состоит из двух шагов [2 – 4]. Пусть имеются два известных (родительских) решения задачи (первое из которых, например, является лучшим из известных), представленных множествами центров кластеров S . Вначале множества родительских решений объединяются. Получаем промежуточное недопустимое (с избыточным числом кластеров) решение. Затем производится последовательное уменьшение числа центров. Каждый раз отсекается тот центр, удаление которого дает наименее существенное ухудшение значения целевой функции. Ниже представлен алго-

ритм работы базовой жадной агломеративной эвристической процедуры:

Алгоритм 3.1 Базовая жадная агломеративная эвристическая процедура (Greedy)

Дано: начальное число кластеров K , требуемое число кластеров $k < K$.

1. Случайным образом определить начальное решение $S = \{X_1, \dots, X_k\}$.
2. Передать решение S в двухшаговый алгоритм локального поиска, получить новое, улучшенное решение S .

пока $K \neq k$

для каждого $i' \in \{1, \dots, K\}$

3.1. $S' = S \setminus \{X_{i'}\}$.

- 3.2. Передать решение S' в двухшаговый алгоритм локального поиска, выполнить от 1 до 3 итераций алгоритма, полученное значение (значение целевой функции) сохранить в $F'_{i'}$.

конец цикла

4. $i'' = \arg \max_{i'=1, \dots, k} F'_{i'}$.

5. Получить решение $S'' = S \setminus \{X_{i''}\}$, улучшить его с помощью двухшагового алгоритма локального поиска.

конец цикла

Производительность Алгоритма 3.1 при больших объемах данных во время выполнения расчетов становится проблемой, особенно когда найти правильный параметр k (число кластеров) в практических задачах можно только путем выполнения нескольких запусков с разным количеством кластеров. При увеличении количества кластеров Алгоритм 3.1 на шаге 2 начинает работать все медленнее (алгоритм требует все большего количества итераций, и каждая итерация требует возрастающих вычислительных ресурсов), поэтому мы внесли изменения и реализовали удаление кластеров не по одному, а по несколько за одну итерацию [11].

Алгоритм 3.2 Базовая жадная агломеративная эвристическая процедура для задач с большим числом кластеров

Дано: начальное число кластеров K , необходимое количество кластеров $k < K$, $k > 50$, первоначальное решение S , $|S|=K$.

1. Улучшить решение S двухшаговым алгоритмом локального поиска (если это возможно).

пока $K \neq k$

для каждого $i' \in \{1, \dots, K\}$

2. $S' = S \setminus \{X_{i'}\}$. Вычислить $F'_{i'} = F(S')$, где $F(\cdot)$ значение целевой функции (например, (1) для задачи k -средних).

конец цикла

3. Установить S_{elim} из n_{elim} центроидов, $S_{elim} \subset S$, $|S_{elim}| = n_{elim}$, с минимальными значениями $F'_{i'}$. Здесь, $n_{elim} = \max\{1, 0.2 \cdot (|S| - k)\}$.
4. Получить новое решение $S = S \setminus S_{elim}$, $K = K - 1$ и улучшить его с помощью двухшагового алгоритма локального поиска.

конец цикла

Предложены эвристические процедуры [12], модифицирующие известное решение на основе второго известного решения (см. Алгоритмы 3.1 и 3.2).

Алгоритм 4 Жадная процедура 1

Дано: множества центров кластеров $S' = \{X'_1, \dots, X'_k\}$ и $S'' = \{X''_1, \dots, X''_k\}$

для каждого $i' \in \{1, K\}$

1. Объединить S' с элементом множества S'' : $S = S' \cup \{X''_{i'}\}$.
2. Запустить базовую жадную агломеративную эвристическую процедуру (Алгоритм 3.1 или 3.2) с S в качестве начального решения. Полученный результат (полученное множество, а также значение целевой функции) сохранить.
3. Возвратить в качестве результата лучшее (по значению целевой функции) из решений, полученных на шаге 2.

конец цикла

Возможен вариант, в котором множества объединяются частично, при этом первое множество берется полностью, а из второго множества выбирается случайным образом случайное число элементов.

Алгоритм 5 Жадная процедура 2

Дано: см. Алгоритм 4.

1. Выбрать случайное $r' \in [0;1]$. Присвоить $r = [(k/2 - 2) r'^2] + 2$.
Здесь $[.]$ – целая часть числа.
2. **Для** i **от** 1 **до** $k - r$
 - 2.1. Сформировать случайно выбранное подмножество S''' элементов множества S'' мощности r . Объединить множества $S = S' \cup S'''$.
 - 2.2. Запустить базовую жадную агломеративную эвристическую процедуру (Алгоритм 3.1 или 3.2) с этими объединенными множествами в качестве начального решения.

конец цикла

Более простой вариант Алгоритма 4 представлен ниже, уже с полным объединением множеств.

Алгоритм 6 Жадная процедура 3

Дано см. Алгоритм 4.

1. Объединить множества $S = S' \cup S''$.
 2. Запустить базовую жадную агломеративную эвристическую процедуру (Алгоритм 3.1 или 3.2) с S в качестве начального решения.
-

Данные алгоритмы могут использоваться в составе различных стратегий глобального поиска, а в качестве окрестностей, в которых производится поиск решения, используются множества решений, производные («дочерние») по отношению к решению S' , образованные комбинированием его элементов с элементами некоторого решения S'' и применением базовой жадной агломеративной эвристической процедуры (Алгоритм 3.1 или 3.2).

Алгоритмы 4-6 могут осуществлять поиск в окрестностях известного промежуточного решения S' , где решения, принадлежащие окрестности, образуются добавлением элементов другого решения S'' с последующим удалением «лишних» центров кластеров жадной агломеративной эвристической процедурой. Таким образом, второе решение S'' является параметром окрестности, выбираемым случайным образом (рандомизированным).

Подход к разработке алгоритмов автоматической группировки, основанных на параметрических оптимизационных моделях

Комбинации алгоритмов метода жадных эвристик с чередующимися окрестностями для задач k-средних, k-медоид и известных алгоритмов j-means и СЕМ были рассмотрены в [9, 12 – 14].

Общую схему предлагаемого нового подхода к разработке алгоритмов автоматической группировки, основанных на параметрических оптимизационных моделях, с комбинированным применением алгоритмов поиска с чередующимися рандомизированными окрестностями и жадных агломеративных эвристических процедур, можно описать следующим образом:

Алгоритм 7 GH-VNS (Greedy Heuristic in the Variable Neighborhood Search – Жадная эвристика в поиске с чередующимися окрестностями)

1. Получить решение S , запустив двухшаговый алгоритм локального поиска из случайным образом сгенерированного начального решения.
2. $O = O_{start}$ (номер окрестности поиска).
3. $i = 0, j = 0$ (количество безрезультатных итераций в конкретной окрестности и в целом по алгоритму).

пока $j < j_{max}$

пока $i < i_{max}$

4. **Если** не выполняются условия ОСТАНОВА (превышение лимита времени), **то** получить решение S' , запустив двухшаговый алгоритм локального поиска из случайного начального решения.

повторять

5. В зависимости от значения O (возможны значения 1, 2 или 3), запустить Алгоритм Жадная процедура 1 либо 2 или 3 соответственно с начальными решениями S и S' . Так, окрестность определяется способом включения центров кластеров из вто-

рого известного решения и параметром окрестности – вторым известным решением.

если новое решение лучше, чем S , **то**
записать новый результат в S , $i = 0, j = 0$.

иначе выйти из цикла.

конец цикла

6. $i = i + 1$.

конец цикла

7. $i = 0, j = j + 1, O = O + 1$, **если** $O > 3$, **то** $O = 1$.

конец цикла

В данном алгоритме i_{max} – число безрезультатных поисков в окрестности, а j_{max} – число безрезультатных переключений окрестностей. Значения этих двух параметров важны при расчетах. Мы использовали значения: $i_{max} = 2k, j_{max} = 2$.

Параметр O_{start} , задающий номер окрестности, с которой начинается поиск, особенно важен. В зависимости от значения O_{start} далее алгоритмы обозначены GH-VNS1, GH-VNS2, GH-VNS3 (для задачи k -средних соответственно – k -GH-VNS1, k -GH-VNS2, k -GH-VNS3).

Важным является способ получения второго решения S' на Шаге 4. По умолчанию второе решение содержит число центров, равное числу центров в решении S . Мы также использовали модификации Алгоритма 7, в которых число центров в решении S' выбирается случайным образом из множества $\{2, |S|\}$, где $|S|$ – число центров в решении S . В этом случае алгоритмы названы GH-VNS1-RND, GH-VNS2-RND, GH-VNS3-RND.

Результаты вычислительных экспериментов

Для нашего исследования мы использовали классические наборы данных из репозитория UCI (Machine Learning Repository) [15] и Clustering basic benchmark [16]. В качестве тестовых наборов данных также были использованы результаты неразрушающих тестовых испытаний сборных производственных партий электрорадиоизделий (ЭРИ), проведенных в специализированном тестовом центре АО «ИТЦ – НПО ПМ» (Железногорск) для комплектации бортовой аппаратуры космических аппаратов [1, 2].

Для всех наборов данных было выполнено по 30 попыток запуска каждого из алгоритмов. Фиксировались только лучшие результаты, достигнутые в каждой попытке, затем из этих результатов по каждому алгоритму были рассчитаны значения целевой функции: минимальное и максимальное значения (Min, Max), среднее значение (Среднее) и среднеквадратичное отклонение (СКО). Алгоритмы j -means и k -средних были запущены в режиме

мультистарта.

Лучшие значения целевой функции (минимальное значение, среднее значение и среднеквадратичное отклонение) выделены полужирным курсивом в табл. 1 – 5.

В табл. 1 приведены результаты вычислительных экспериментов по набору данных BIRCH3 (100000 векторов данных, каждый размерностью 2) 100 кластеров, 6 часов.

В табл. 2 представлены результаты вычислительных экспериментов по набору данных KDDCUP04BioNormed (145751 векторов данных, каждый размерностью 74) 200 кластеров, 24 часа.

Таблица 1

Алгоритм	Значение целевой функции			
	Min	Max	Среднее	СКО
k-средних	7,92474E+13	8,87404E+13	8,31599E+13	3,088140E+12
j-means	3,76222E+13	3,7965E+13	3,77715E+13	0,116211E+12
k-GH-VNS1	3,72537E+13	3,77474E+13	3,74703E+13	0,171124E+12
k-GH-VNS2	4,21378E+13	5,01871E+13	4,52349E+13	4,333462E+12
k-GH-VNS3	3,72525E+13	3,74572E+13	3,73745E+13	0,074315E+12
k-GH-VNS1-RND	3,72541E+13	3,77687E+13	3,74943E+13	0,185483E+12
k-GH-VNS2-RND	3,83257E+13	4,61847E+13	4,0815E+13	2,543163E+12
k-GH-VNS3-RND	3,73131E+13	3,75242E+13	3,74164E+13	0,061831E+12

Таблица 2

Алгоритм	Значение целевой функции			
	Min	Max	Среднее	СКО
k-средних	5 336 446	5 381 386	5 366 144	25 722,4
j-means	5 330 344	5 382 908	5 355 903	26 785,8
k-GH-VNS1	5 294 620	5 307 828	5 301 224	9 339,5
k-GH-VNS2	5 440 814	5 476 140	5 458 477	24 979,5
k-GH-VNS1-RND	5 310 067	5 340 849	5 325 458	21 765,7
k-GH-VNS2-RND	5 368 527	5 399 695	5 384 111	22 039,6

Для более полного сравнения полученных результатов вычислительных экспериментов новых алгоритмов были использованы данные вычислительных экспериментов, полученных Л.А Казаковцевым. [2] над наборами данных тестовых испытаний партии изделий промышленной продукции различными модификациями генетического алгоритма (табл. 3). Для расчетов были использованы сборные партии изделий промышленной продукции 1526TL1 – 3 партии (1234 векторов данных, каждый размерностью 157).

В табл. 3 показаны результаты вычислительных экспериментов над результатами испытаний производственных партий изделий 1526TL1 (10 кла-

стеров, 1 минута), где использованы следующие аббревиатуры и сокращения: ГА – генетический алгоритм; ЖЭ – жадная эвристика; ГАЖЭ – генетический с жадной эвристикой с вещественным алфавитом; ЛП – локальный поиск; ГА ФП – генетический алгоритм с рекомбинацией подножеств фиксированной длины.

Таблица 3

Алгоритм	Значение целевой функции			
	Min	Max	Среднее	СКО
k-средних	43 842,10	43 844,66	43 843,38	0,8346
j-means	43 841,97	43 843,51	43 842,59	0,4487
k-GH-VNS1	43 841,97	43 844,18	43 842,34	0,9000
k-GH-VNS2	43 841,97	43 844,18	43 843,46	1,0817
k-GH-VNS3	43 841,97	43 842,10	43 841,99	0,0424
ГАЖЭ+ЛП	43 842,10	43 845,73	43 843,72	1,3199
ГАЖЭ вещ., $\sigma_e=0.25$	43 841,98	43 844,18	43 842,6	0,6762
ГАЖЭ вещ.частич., $\sigma_e=0,25$	43 841,98	43 841,98	43 841,98	1,53E-11
ГА ФП	43 841,98	43 842,34	43 842,10	0,0945
ГА классич.	43 842,10	43 842,88	43 842,44	0,2349
Детерм. ЖЭ, $\sigma_e=0.25$	45 113,56	45 113,56	45 113,56	0,0000
Детерм. ЖЭ, $\sigma_e=0.001$	45 021,21	45 021,21	45 021,21	0,0000

В табл. 4 представлены результаты вычислительных экспериментов по набору данных *ionosphere* (351 векторов данных, каждый размерностью 35) 10 кластеров, 60 секунд, 30 попыток, манхэттенское расстояние.

Таблица 4

Алгоритм	Значение целевой функции		
	Min (рекорд)	Среднее	СКО
PAM	2 688,57	2 704,17	12,3308
PAM-GH-VNS1	2 607,21	2 607,25	0,1497
PAM-GH-VNS2	2 607,21	2 607,43	0,4303
PAM-GH-VNS3	2 607,21	2 607,34	0,4159
GA-FULL	2 608,22	2 624,97	9,5896
GA-ONE	2 608,69	2 625,18	10,7757

В табл. 4 GA-FULL – генетический алгоритм с жадной эвристикой с вещественным алфавитом, GA-ONE – генетический алгоритм, в котором Алгоритм 5 используется в качестве процедуры кроссинговера.

В табл. 5 приведены результаты вычислительных экспериментов по наборам данных тестовых испытаний партии изделий промышленной продукции (10 кластеров, 2 минуты, 30 попыток).

Таблица 5

Алгоритм	Значение целевой функции			
	Min	Max	Среднее	СКО
3OT122A (767 векторов данных, каждый размерностью 13)				
CEM	120 947,6	146 428,5	135 777,6	7 985,6992
CEM-GH-VNS1	121 256,5	152 729,1	143 956,0	8 708,6293
CEM-GH-VNS2	123 664,4	158 759,2	143 028,5	10 294,3992
CEM-GH-VNS3	128 282,2	155 761,9	143 506,9	10 058,8266
1526TL1 (1234 векторов данных, каждый размерностью 157)				
CEM	354 007,3	416 538,4	384 883,4	20 792,8068
CEM-GH-VNS1	376 137,1	477 124,5	438 109,4	29 964,0641
CEM-GH-VNS2	345 072,6	487 498,3	444 378,1	43 575,3282
CEM-GH-VNS3	379 352,3	516 777,8	456 271,4	38 323,0246

В настоящее время в кластерном анализе существует тенденция к применению коллективных методов [17]. Алгоритмы кластерного анализа не являются универсальными: каждый алгоритм имеет свою особую область применения. В том случае, если рассматриваемая область содержит различные типы наборов данных, для выделения кластеров приходится применять не один определенный алгоритм, а набор различных алгоритмов. Ансамблевый (коллективный) подход позволяет снизить зависимость конечного решения от выбранных параметров исходных алгоритмов и получить более устойчивое решение [18].

Заключение

Результаты вычислительных экспериментов показали, что новые алгоритмы метода жадных эвристик для задач автоматической группировки с повышенными требованиями к точности результата (по значению целевой функции), с применением алгоритмов поиска с чередующимися рандомизированными окрестностями (GH-VNS), имеют более стабильные (меньшее среднеквадратичное отклонение целевой функции) и более точные (меньшее среднее значение целевой функции) результаты и, следовательно, лучшие показатели в сравнении с классическими алгоритмами (k-средних, j-means, РАМ и CEM).

В то же время с ростом числа кластеров и объема выборки сравнительная эффективность нового подхода, основанного на параметрических оптимизационных моделях, с комбинированным применением алгоритмов поиска с чередующимися рандомизированными окрестностями и жадных агломеративных эвристических процедур, повышается, и для больших наборов данных новые алгоритмы имеют преимущество при фиксированном времени

работы алгоритма.

Однако стоит отметить, что при значительном увеличении времени расчетов известные генетические алгоритмы метода жадных эвристик показывают немного лучшие результаты в сравнении с предложенными новыми алгоритмами. Тем не менее, можно говорить о конкурентоспособности новых алгоритмов как в сравнении с классическими алгоритмами k-средних, РАМ и j-means, так и с генетическими алгоритмами, включая алгоритмы метода жадных эвристик, а также с детерминированными алгоритмами.

ЛИТЕРАТУРА

1. Рожнов И.П., Орлов В.И., Гудыма М.Н., Казаковцев В.Л. Составление оптимальных ансамблей алгоритмов кластеризации // Системы управления и информационные технологии. – 2018. – № 2(72). – С. 31-35.
2. Казаковцев Л.А. Метод жадных эвристик для систем автоматической группировки объектов: Дис... д-ра техн. наук. – Красноярск. 2016.
3. Kazakovtsev L.A., Antamoshkin A.N. Greedy Heuristic Method for Location Problems // Вестник Сибирского государственного аэрокосмического университета им. академика М.Ф. Решетнева. – 2015. – Т. 16, № 2. – С. 317-325.
4. Kazakovtsev L.A., Antamoshkin A.N. Genetic Algorithm with Fast Greedy Heuristic for Clustering and Location Problems // Informatica (Ljubljana). – 2014. – Т. 38, № 3. – С. 229-240.
5. Farahani R.Z., Hekmatfar M. (eds.) Facility location: Concepts, models, algorithms and case studies // Berlin Heidelberg. – Springer-Verlag. – 2009. – P. 549.
6. L. Kaufman, P.J. Rousseeuw Clustering by means of Medoids. Statistical Data Analysis Based on the L1-Norm and Related Methods. – Springer US. – 1987. – P. 405-416.
7. Антамошкин А.Н., Казаковцев Л.А. Алгоритм случайного поиска для обобщенной задачи Вебера в дискретных координатах // Информатика и системы управления. – 2013. – № 1 (35). – С. 87-98.
8. Королев В.Ю. EM-алгоритм, его модификации и их применение к задаче разделения смесей вероятностных распределений. Теоретический обзор. – М.: ИПИ РАН. – 2007. – С. 94.
9. Rozhnov I., Kazakovtsev L., Bezhitskaya E., Bezhitskiy S. Improved Classification EM algorithm for the Problem of Separating Semiconductor Device Production Batches // В сб.: IOP Conference Series: Materials Science and Engineering International Workshop "Advanced Technologies in Material Science, Mechanical and Automation Engineering – MIP: Engineering – 2019". Krasnoyarsk. – 2019. – Vol. 537. DOI:10.1088/1757-899X/537/5/052032.
10. Mladenovic N., Hansen P. Variable neighborhood search // Comput. Oper. Res. – 1997. – Vol. 24. – P. 1097-1100.
11. Рожнов И.П., Казаковцев В.Л. Реализация жадных эвристических алгоритмов кластеризации для массивно-параллельных систем // Системы управления и информационные технологии. – 2019. – № 2(76). – С. 36-40.
12. Рожнов И.П., Казаковцев Л.А., Гудыма М.Н., Казаковцев В.Л. Алгоритм для задачи к-

- средних с рандомизированными чередующимися окрестностями // Системы управления и информационные технологии. – 2018. – № 3(73). – С. 46-51.
13. Рожнов И.П. Алгоритмы с чередованием жадных эвристических процедур для дискретных задач кластеризации // Системы управления и информационные технологии. – 2019. – № 1(75). – С. 49-55.
 14. Казаковцев Л.А., Рожнов И.П., Шестаков П.Ф. Усовершенствованный СЕМ-алгоритм для данных большой размерности // Наука и образование: опыт, проблемы, перспективы развития. Ч.2. – Красноярск: КГАУ, 2019. – С. 280-284.
 15. UCI Machine Learning Repository [Электронный ресурс]. – Режим доступа: <http://archive.ics.uci.edu/ml>
 16. Clustering basic benchmark [Электронный ресурс]. – Режим доступа: <http://cs.joensuu.fi/sipu/datasets>.
 17. Бериков В.В. Классификация данных с применением коллектива алгоритмов кластерного анализа // Знания-Онтологии-Теории (ЗОНТ-2015). – 2015. – С. 29-38.
 18. Rozhnov I., Orlov V., Kazakovtsev L. Ensembles of clustering algorithms for problem of detection of homogeneous production batches of semiconductor devices // CEUR Workshop Proceedings Ser. OPTA-SCL 2018 – Proceedings of the School-Seminar on Optimization Problems and their Applications. CEUR-WS, 2018. – Vol. 2098. – P. 338-348.

Статья представлена к публикации членом редколлегии А.Д. Плутенко.

E-mail:

Рожнов Иван Павлович – ris2005@mail.ru;

Казаковцев Лев Александрович – levk@bk.ru.