

ПРИМЕНЕНИЕ МЕТАЭВРИСТИЧЕСКИХ АЛГОРИТМОВ К РЕШЕНИЮ ЗАДАЧ КЛАСТЕРИЗАЦИИ МЕТОДОМ k -СРЕДНИХ

А.В. Лисин, Р.Т. Файзуллин

Омский государственный технический университет, Омск, Россия

Аннотация

В статье рассматривается подход к сегментации изображений методом k -средних путём сведения задачи кластеризации к задаче непрерывной оптимизации и её решения с помощью метаэвристических алгоритмов на примере алгоритма гравитационного поиска. Обосновывается применимость данного подхода, приводятся результаты численного эксперимента.

Ключевые слова: кластеризация, метод k -средних, обработка изображений, непрерывная оптимизация.

Цитирование: Лисин, А.В. Применение метаэвристических алгоритмов к решению задач кластеризации методом k -средних / А.В. Лисин, Р.Т. Файзуллин // Компьютерная оптика. – 2015. – Т. 39, №3 – С. 406–412.

Введение

Задача кластеризации – это задача разбиения заданного множества объектов на непересекающиеся подмножества, называемые кластерами, таким образом, чтобы объекты, принадлежащие одному кластеру, были в некотором смысле похожи, а объекты из разных кластеров существенно различались.

Определение. Пусть X – множество объектов, а Y – множество меток кластеров. Тогда алгоритмом кластеризации назовём функцию $\alpha: X \rightarrow Y$, которая любому объекту из множества X ставит в соответствие метку (номер) кластера из множества Y .

Метод k -средних – один из наиболее популярных методов кластеризации, находящий применение в большом количестве областей, таких как обработка изображений, кластерный анализ, интеллектуальный анализ данных и т. д. [1]. Задача кластеризации методом k -средних может быть сформулирована следующим образом: пусть дано множество точек $X = \{x_1, \dots, x_n\}$ в метрическом пространстве \mathbb{E}^n . В соответствии с условиями задачи кластеризации необходимо найти разбиение заданного множества X на $k \leq n$ таких подмножеств $S = \{S_1, \dots, S_k\}$, что сумма расстояний точек кластера от их центров минимальна. Фактически необходимо найти

$$\min \sum_{i=1}^k \sum_{x_j \in S_i} \|x_j - \mu_i\|, \tag{1}$$

где μ_i – центр i -го кластера, определяемый по формуле $\mu_i = (1/|S_i|) \sum_{x_j \in S_i} x_j$. Поскольку по условию \mathbb{E}^n – метрическое пространство, $\|x_j - \mu_i\| = d(x_j, \mu_i) \geq 0$, где d – расстояние между точками x_j и μ_i .

Вычислительно задача кластеризации методом k -средних является NP-трудной, а именно: для фиксированных k и размерности пространства d задача решается за время $O(n^{dk+1} \log n)$ [2]. Однако, несмотря на это, существует некоторое количество эвристических алгоритмов, позволяющих быстро находить локальный оптимум для входных данных. Алгоритм Ллойда явля-

ется одним из популярных эвристических алгоритмов, применяемых для приближенного решения задач кластеризации. Кратко алгоритм Ллойда может быть описан последовательностью следующих шагов:

1. Случайным образом выбирается начальный набор центров кластеров из k -элементов $\mu_1^{(1)}, \dots, \mu_k^{(1)}$.
2. Для каждой точки $x_p \in X$ определяется её принадлежность к тому или иному кластеру на шаге t согласно правилу

$$x_p \in S_i^{(t)}, \text{ если } \|x_p - \mu_i^{(t)}\| \leq \|x_p - \mu_j^{(t)}\| \quad \forall j, \\ 1 \leq j \leq k, i \neq j.$$

3. Вычисляются новые значения центров кластеров согласно формуле

$$\mu_i^{(t+1)} = (1/|S_i^{(t)}|) \sum_{x_j \in S_i^{(t)}} x_j.$$

4. Если положения центров кластеров изменились по сравнению с шагом $t-1$, переходят к шагу 2. В противном случае работа алгоритма завершается, и возвращаются найденные центры кластеров [3].

Стоит обратить внимание, что алгоритм Ллойда на каждой итерации вычисляет новые положения центров кластеров как центры масс точек, принадлежащих данному кластеру. Отметим, что данная эвристика очень похожа на использованную в работе [4].

Алгоритм Ллойда имеет следующие недостатки:

1. Не гарантирует достижения глобального минимума функции (1), а лишь одного из локальных минимумов.
2. Результат работы сильно зависит от начального выбора центров кластеров.
3. Число кластеров необходимо знать заранее.

Алгоритм Ллойда используется в качестве базового в большинстве математических и инженерных пакетов ввиду простоты реализации и высокой скорости работы. При этом качество решения задачи кластеризации зачастую оказывается непригодным для дальнейшего анализа, особенно в контексте обработки и анализа изображений, ввиду ограничений, приведённых выше.

Кластеризация в контексте обработки изображений

Важным приложением задачи кластеризации в контексте обработки и анализа изображений является квантование цвета (сегментация изображения). Под квантованием цвета понимается процесс выбора малого числа характерных цветов некоторого изображения. Задачу квантования, таким образом, можно сформулировать как нахождение заданного количества «наилучших» цветов, встречающихся в исходном изображении, и замену всех остальных цветов на наиболее подходящие из найденного множества. При этом кластеризации методом k -средних при помощи алгоритма Ллойда зачастую отдаётся предпочтение ввиду простоты реализации и высокой скорости работы (что особенно актуально в случае обработки потокового видео).

Квантование цвета в том или ином виде зачастую используется как один из этапов процесса анализа изображения. Так, квантование активно используется в моделировании компьютерного зрения, обработке изображений, полученных с телескопов, анализе медицинских снимков и т.д. Резюмируя, можно сказать, что квантование используется в ситуациях, когда необходимо выделить некоторые «области интереса» на изображении или уменьшить количество различных цветов на рисунке.

Отметим, что любое RGB-изображение размера $p \times q$ пикселей может быть представлено как матрица A_{pq} , компоненты которой a_{ij} являются векторами в пространстве \mathbb{Z}_+^3 , все элементы r которого ограничены $r \in [0, 255]$. Однако в работе с целью повышения наглядности графиков мы будем использовать изображения, у которых третья компонента цвета будет равна нулю, т.е. компонентами матрицы A_{pq} будут векторы в пространстве \mathbb{Z}_+^2 (рис. 1). Данное допущение не влияет на результаты, однако позволяет использовать двухмерные графики для представления результатов численных экспериментов.



Рис. 1. Исходное изображение с отфильтрованной синей компонентой в оттенках серого

На рис. 2 показано распределение цветов в исходном изображении.

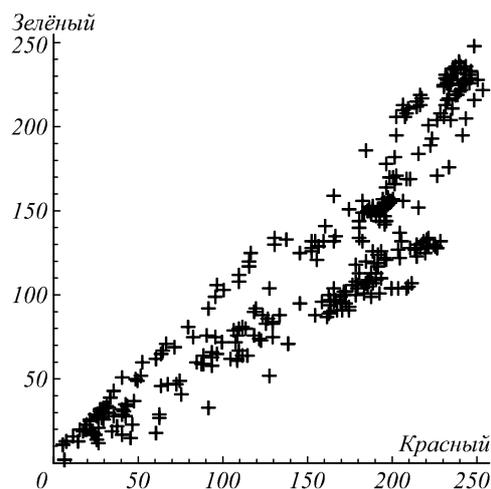


Рис. 2. Распределение цветов в исходном изображении

Кластеризация с помощью метода k -средних как задача непрерывной оптимизации

Рассмотрим задачу (1) в контексте обработки изображений. Как было отмечено, точное решение этой задачи может быть найдено за экспоненциальное время средствами комбинаторной оптимизации. С другой стороны, при некотором фиксированном k задачу (1) можно рассматривать с точки зрения непрерывной оптимизации в пространстве \mathbb{R}_+^{2k} . В этом случае необходимо найти такой вектор $Y^* \in \mathbb{R}_+^{2k}$, что

$$f(Y^*) \leq f(Y), \forall Y \in \mathbb{R}_+^{2k},$$

где $f: \mathbb{R}_+^{2k} \rightarrow \mathbb{R}_+$ – целевая функция вида

$$f(Y) = \sum_{i=1}^{2k-1} \sum_{\lambda \in S_i} \sqrt{(x_r - y_i)^2 + (x_g - y_{i+1})^2}. \quad (2)$$

В функции (2) x_r и x_g – красная и зелёная составляющие цвета некоторой точки изображения соответственно, а Y – вектор, составленный из координат центров кластеров таким образом, что пара (y_0, y_1) является центром первого кластера, (y_2, y_3) – центром второго и т.д. Обобщая: каждый вариант расположения центров k кластеров в пространстве \mathbb{R}_+^2 содержится в $2k$ -мерном векторе $Y = \{y_0, y_1, y_2, y_3, \dots, y_{2k-1}, y_{2k}\}$. Такой вид целевой функции аналогичен использованному в работе [5].

Целевая функция (2) непрерывна, а область её определения компактна [6]. В соответствии с теоремой Вейерштрасса это означает, что предлагаемая целевая функция имеет минимум в области поиска. С другой стороны, способ задания функции не позволяет вычислить её производную, что исключает аналитический подход к поиску глобального минимума, а также не позволяет использовать различные вариации производных методов, требующих вычисления производной в конкретных точках. Несмотря на это, глобальный минимум можно постараться найти с помощью численных методов, не требующих манипуляций с производными. К сожалению, многие эффективные алгоритмы оптимизации могут не дать желаемого результата ввиду потенциально большой

размерности задачи, а также наличия большого числа локальных минимумов и максимумов целевой функции. Например, широко распространённый метод Нелдера–Мида может «застрывать» в локальных экстремумах и не приводит к получению желаемого решения [6]. В этой связи видится разумным использовать для решения задачи алгоритмы, способные «выбираться» из локальных экстремумов. В данной работе для поиска глобального минимума предлагается использовать хорошо зарекомендовавший себя [7] алгоритм гравитационного поиска.

**Метаэвристики
и алгоритм гравитационного поиска**

Метаэвристические алгоритмы (МА) – широкий класс алгоритмов оптимизации, включающий в себя алгоритмы имитации отжига, поиска с запретом, алгоритм роя частиц, генетические алгоритмы и т.д.

Несмотря на то, что внешне МА могут сильно отличаться друг от друга, все они могут быть описаны одной обобщённой формулой. На каждом шаге алгоритм вычисляет очередное решение x^{t+1} согласно следующему правилу:

$$x^{t+1} = A(x^t, p(t)),$$

где A – некоторое отображение, x^t – решение на шаге t , а $p(t)$ – параметры, зависящие от конкретного алгоритма, которые также могут зависеть от номера шага t . Также в параметры $p(t)$ в том или ином виде может входить информация о решениях, полученных на шагах $[1, t-1]$.

Предложенный в [8] алгоритм гравитационного поиска (АГП) – относительно новый метаэвристический алгоритм глобальной оптимизации, основанный на физической модели взаимодействия тел большой массы. Данный алгоритм также можно отнести к классу роевых, т.к. АГП, как и хорошо известный алгоритм роя частиц, основан на развитии многоагентной системы. При этом каждый агент в АГП имеет массу и скорость, а взаимодействие между агентами происходит по правилам классической ньютоновской механики. Таким образом, имеется система из N агентов. Положение i -го агента в пространстве поиска определяется как

$$X_i = (x_i^1, \dots, x_i^d, \dots, x_i^n), \quad i = 1, 2, \dots, N,$$

где x_i^d – координата i -го агента в d -м измерении, а n – размерность пространства поиска.

На шаге t алгоритма сила, действующая на агент i со стороны агента j в d -м направлении, определяется согласно правилу

$$F_{ij}^d = G(t) \frac{M_i(t)M_j(t)}{R_{ij} + \epsilon} (X_j^d(t) - X_i^d(t)), \quad (3)$$

где M_i и M_j – массы i -го и j -го агентов на шаге t , ϵ – некоторая малая константа, а R_{ij} – Евклидово расстояние между агентами. Обратим внимание, что согласно предложению авторов АГП, основанному на экспериментальных результатах, в формуле (3) фигурирует не квадрат расстояния между агентами, как в

формуле классической механики, а обычное Евклидово расстояние.

Результирующая сила, действующая на агент i в d -м направлении, определяется по формуле

$$F_i^d(t) = \sum_{j=1, j \neq i}^N rand_j F_{ij}^d(t), \quad (4)$$

где $rand_i$ – случайное число на отрезке $[0, 1]$. Этот множитель служит для придания стохастических характеристик алгоритму.

В соответствии со вторым законом Ньютона, ускорение i -го агента на шаге t в направлении d определяется как

$$a_i^d(t) = F_i^d(t) / M_i(t), \quad (5)$$

где M_i – масса i -го агента.

Скорость агента на $t+1$ шаге определяется в соответствии с уравнением движения $\partial X_i / \partial t = -(\partial v / \partial X_i)$, решение которого даёт следующую итерационную формулу для вычисления нового положения агента относительно шага t [9]:

$$v_i^d(t+1) = rand_i \cdot v_i^d(t) + a_i^d(t), \quad (6)$$

$$X_i^d(t+1) = X_i^d(t) + v_i^d(t+1), \quad (7)$$

где $rand_i$ – случайное число на отрезке $[0, 1]$.

Гравитационная постоянная G инициализируется в начале работы алгоритма и монотонно убывает с течением времени. G может быть задана формулой

$$G(t) = G_0 e^{-\alpha \frac{t}{t_{max}}},$$

где G_0 – начальное значение, α – некоторая константа, t_{max} – максимальное число итераций алгоритма.

На каждом шаге алгоритма массы агентов пересчитываются в зависимости от значения целевой функции. При этом целевая функция принимает меньшие значения в точках, соответствующих агентам с большей массой. Пересчёт значений масс агентов осуществляется по правилу: сначала вычисляются промежуточные значения масс агентов по формуле

$$\tilde{M}_i(t) = \frac{worst(t) - fit_i(t)}{worst(t) - best(t)},$$

$$best(t) = \min_{j \in 1, \dots, N} fit_j(t),$$

$$worst(t) = \max_{j \in 1, \dots, N} fit_j(t),$$

где $fit_i(t)$ равно значению целевой функции в X_i , а $worst(t)$ и $best(t)$ – худшее и лучшее значения на шаге t соответственно. После чего непосредственно вычисляются массы агентов для данного шага алгоритма

$$M_i(t) = \tilde{M}_i(t) / \sum_{j=1}^n \tilde{M}_j(t).$$

Таким образом, АГП может быть описан в общем виде как следующая последовательность шагов:

1. Определяется пространство поиска (в случае с квантованием цвета пространство поиска \mathbb{R}_+^{2k} , где $\forall r \in \mathbb{R}_+^{2k}, r \in [0, 255]$).

2. Случайным образом инициализируется начальное положение N агентов.
3. Рассчитывается значение целевой функции fit_j для каждого агента.
4. Вычисляются новые значения $G(t)$, $best(t)$, $worst(t)$ и массы агентов $M_i(t)$ для $i=1, \dots, N$ согласно приведённым выше формулам.
5. Рассчитывается значение действующих на агент сил по формулам (3) и (4).
6. Вычисляются новые значения ускорений и скоростей для каждого агента по формулам (5) и (6).
7. Вычисляются новые координаты агентов по формуле (7).
8. Шаги 2–7 повторяются, пока не будет достигнуто условие останова, которым может служить достижение максимального заданного заранее количества итераций.
9. В качестве результата возвращается лучшее найденное положение агента за всё время работы алгоритма.

Квантование цветов с помощью АГП

Вернёмся к задаче квантования цветов. Формула (2) определяет функцию, которая может быть использована в качестве целевой функции для АГП. Тогда последовательность шагов для решения задачи квантования цвета методом k -средних можно записать следующим образом:

1. Случайным образом сгенерировать N векторов Y , где $Y_i = \{y_i^{(1)}, y_i^{(2)}, \dots, y_i^{(k)}, \dots, y_i^{(2k)}\}$. Таким образом, каждый вектор Y_i содержит в себе один из вариантов расположения центров всех k -кластеров. Также определяется максимальное количество итераций t_{max} и устанавливается $t = 1$.
2. Для каждого агента Y_i рассчитывается значение функции (2).
3. Вычисляются новые значения $G(t)$, $best(t)$, $worst(t)$ и массы агентов $M_i(t)$ для $i = 1, \dots, N$.
4. Рассчитывается значение действующих на агент сил.
5. Вычисляются новые значения ускорений и скоростей для каждого агента.
6. Вычисляются новые координаты агентов. Счётчик t увеличивается на единицу.
7. Шаги 2–6 повторяются, пока $t \leq t_{max}$.
8. Возвращается лучшее найденное положение агента за всё время работы алгоритма.

Результаты численного эксперимента

В качестве исходного изображения для численного эксперимента использовалось стандартное изображение «Лена» с отфильтрованной синей компонентой цвета.

На рис. 3 представлен результат работы алгоритма Ллойда для $k = 10$. На рис. 4 представлены соответствующие полученному решению кластеры и их центры.

Для эксперимента на основании нескольких пробных запусков были выбраны следующие параметры:

количество агентов $N = 30$, начальное значение гравитационной постоянной $G = 100$, количество итераций $t_{max} = 250$.



Рис. 3. Результат работы алгоритма Ллойда, $k = 10$

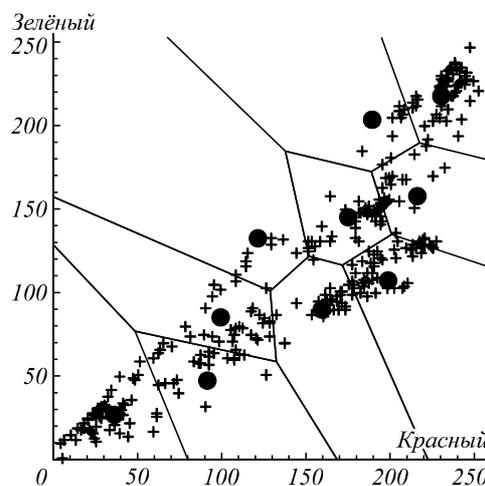


Рис. 4. Кластеры, полученные с помощью алгоритма Ллойда, $k = 10$

Начальное положение кластеров в соответствии с описанием алгоритма выбиралось случайно в диапазоне от 0 до 255. Также в коде алгоритма был предусмотрен случай, когда одна из координат некоторого кластера «выпадала» из данного диапазона. В такой ситуации агент, содержащий «выпавший» центр кластера, изымался, а на его место ставился случайно сгенерированный.

В табл. 1 представлены значения целевой функции (2) для результатов, полученных с помощью АГП и алгоритма Ллойда (из стандартной поставки пакета Wolfram Mathematica) для тестового изображения при различных k . Видно, что во всех тестовых запусках АГП показал лучшие результаты, чем стандартный алгоритм.

Результат квантования цвета, полученный с помощью АГП для $k = 10$, приведён на рис. 5. Можно обратить внимание на меньшую «зернистость» полученного изображения.

На рис. 6 приведены кластеры для $k = 10$, полученные с помощью АГП.

Отметим также интересную особенность решения для $k = 10$, полученного с помощью АГП. Как видно из рис. 6, один из кластеров оказался «пустым».

Табл. 1. Результаты численного эксперимента

k	Алг. Ллойда	АГП
5	$1,64 \times 10^7$	$3,01 \times 10^6$
10	$6,13 \times 10^6$	$1,62 \times 10^6$
15	$4,05 \times 10^6$	$1,08 \times 10^6$
20	$3,12 \times 10^6$	$9,12 \times 10^5$
25	$2,55 \times 10^6$	$7,67 \times 10^5$
30	$2,09 \times 10^6$	$7,07 \times 10^5$
35	$1,65 \times 10^6$	$5,24 \times 10^5$
40	$1,46 \times 10^6$	$5,11 \times 10^5$
45	$1,35 \times 10^6$	$4,23 \times 10^5$
50	$1,26 \times 10^6$	$3,97 \times 10^5$



Рис. 5. Результат работы АГП, $k = 10$

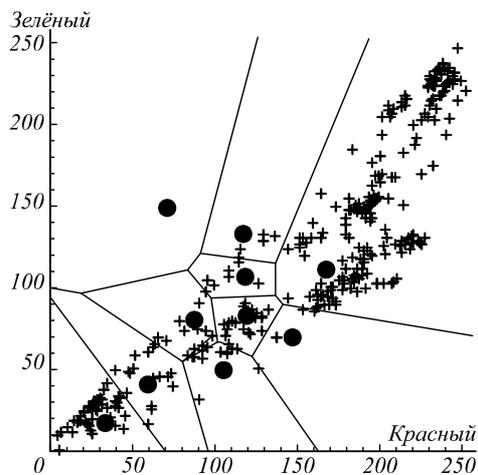


Рис. 6. Кластеры, полученные с помощью АГП, $k = 10$

Это важное свойство алгоритма, т.к. на первый взгляд может показаться логичным ограничить область поиска центров кластеров выпуклой оболочкой всех точек исходного множества. С точки зрения реализации, это довольно простое изменение [10]. С другой стороны, без подобных ограничений можно увидеть, что алгоритм способен находить решения для общего случая, когда количество кластеров k заранее неизвестно. Это свойство выгодно отличает предложенную схему кластеризации от большинства существующих, требующих жёсткого задания количества кластеров.

Анализ сходимости

Как и для большинства алгоритмов роевого типа, анализ сходимости АГП представляет собой нетриви-

альную задачу. В статье [9] приводится сравнительный анализ АГП с алгоритмом роя частиц для поиска глобального экстремума различных унимодальных и мультимодальных тестовых функций. Экспериментальные результаты из [9] говорят о более быстрой сходимости АГП к глобальному оптимуму, чем у алгоритма роя частиц. Несмотря на это, скорость сходимости и точность получаемого решения сильно зависят от выбора начальных параметров. В случае АГП параметры, на которые может влиять пользователь, – это начальное значение гравитационной постоянной G_0 , число итераций t_{max} и количество агентов N . Далее обсудим влияние перечисленных параметров на сходимость алгоритма для примера из этой статьи.

Сила, действующая между двумя агентами, пропорционально зависит от $G(t)$. Эта сила, в свою очередь, определяет расстояние, на которое может переместиться агент на шаге t . Таким образом, выбирая слишком большое значение G_0 , мы рискуем попасть в ситуацию, когда большая сила, действующая на агент, «выталкивает» его из области поиска решений, что заставляет алгоритм отбрасывать некорректное решение и возвращать агент на случайную позицию в области поиска.

Напротив, слишком малое значение G_0 заставляет агенты двигаться слишком медленно, что приводит к медленной сходимости, даже если направление движения агентов соответствует направлению движения к глобальному минимуму.

Максимальное количество итераций t_{max} – второй, наряду с G_0 , параметр, от которого зависит сходимость алгоритма и точность получаемого решения. Большое количество итераций в общем случае означает большую точность получаемых решений. С другой стороны, из-за монотонного убывания $G(t)$, начиная с некоторого шага, скорость изменения положения агентов становится крайне низкой. Поскольку при решении задачи квантования цвета нас интересует только целая часть координат агентов, алгоритм может останавливать работу, как только целая часть координат решения стабилизировалась. Помимо этого, параметры G_0 и t_{max} должны быть согласованы таким образом, чтобы за время работы алгоритма $G(t)$ достигло достаточно малой величины, но достигло её за достаточно большое число шагов.

Количество агентов N , участвующих в поиске, влияет, прежде всего, на поисковые свойства алгоритма (чем больше агентов, тем выше вероятность попадания хотя бы одного из них в область, где присутствует локальный или глобальный минимум целевой функции). С другой стороны, большое количество агентов сильно замедляет работу алгоритма, т.к. в ходе его работы на каждом шаге вычисляется взаимодействие каждого агента с каждым. Этот факт может быть критичным, учитывая большие объёмы вычислений, проделываемых алгоритмом.

Параметры запуска, использованные в работе, были получены экспериментальным путём. Для этого было произведено несколько пробных запусков с раз-

личными параметрами и сравнение получаемых результатов.

Заклучение

В статье продемонстрирован подход к решению задач кластеризации методом k -средних путём сведения их к задачам непрерывной оптимизации и применения к ним метаэвристических алгоритмов оптимизации. В качестве примера рассмотрено решение задачи цветового квантования изображения с помощью алгоритма гравитационного поиска. Предложенный метод продемонстрировал высокое качество результатов по сравнению со стандартными средствами цветового квантования математических пакетов (в частности, из состава математического пакета Wolfram Mathematica) при более низкой скорости работы.

Резюмируя, можно сказать, что предложенный подход к квантованию изображений имеет следующие преимущества:

1. Высокое качество получаемых результатов.
2. Способность находить решения даже при неизвестном количестве кластеров.
3. Простота реализации.

Также у предложенного подхода имеются и недостатки:

1. Относительно низкая скорость работы (что может стать серьезной проблемой при обработке потокового видео).
2. Необходимость переформулирования задачи в виде задачи нелинейного программирования.
3. Зависимость размерностей оптимизационной задачи от количества необходимых кластеров.

Также следует отметить, что несмотря на то, что в статье решение задачи кластеризации привязано к задаче обработки изображений, полученные результаты могут быть применены и в общем случае для кластеризации методом k -средних.

Литература

1. **Сегаран, Т.** Программируем коллективный разум / Т. Сегаран; пер. с англ. – Символ-Плюс, 2008. – 368 с. (T. Segaran. Programming Collective Intelligence. – O'Reilly Media, 2007.)
2. **Inaba, M.** Applications of Weighted Voronoi Diagrams and Randomization to Variance-based K-clustering: (Extended Abstract) / M. Inaba, N. Katoh, H. Imai // Proceedings of the Tenth Annual Symposium on Computational Geometry. – 1994. – Vol. 10 – P. 332–339.
3. **Lloyd, S.** Least squares quantization in PCM / S. Lloyd // Information Theory, IEEE Transactions on. – 1982. – Vol. 28(2) – P. 129-137.
4. **Лисин, А.В.** Эвристический алгоритм поиска приближенного решения задачи Штейнера, основанный на фи-

зических аналогиях / А.В. Лисин, Р.Т. Файзуллин // Компьютерная оптика. – 2013. – Т. 37, № 4. – С. 503-510.

5. **Дулькейт, В.И.** Приближенное решение задачи коммивояжера методом рекурсивного построения вспомогательной кривой / В.И. Дулькейт, Р.Т. Файзуллин // Прикладная дискретная математика. – 2009. – Т. 1, № 3. – С. 72-78.
6. **Sundaram, R.K.** A First Course in Optimization Theory / R.K. Sundaram. – Cambridge University Press, 1996. – 376 p.
7. **Avriel, M.** Nonlinear Programming: Analysis and Methods / M. Avriel – Dover Publishing, 2003. – 544 p.
8. **Duman, S.** Gravitational Search Algorithm for Economic Dispatch with Valve-Point Effects / S. Duman, U. Güvenc, N. Yörükeren // International Review of Electrical Engineering. – 2010. – Vol. 5. – P. 2890-2895.
9. **Rashedi, E.** GSA: A Gravitational Search Algorithm / E. Rashedi, H. Nezamabadi-pour, S. Saryazdi // Information Science. – 2009. – Vol. 179. – P. 2232–2248.
10. **Зиновьев, А.Ю.** Визуализация многомерных данных / А.Ю. Зиновьев. – Красноярск: Изд-во Красноярского государственного технического университета, 2000. – 180 с.
11. **Кормен, Т.** Алгоритмы. Построение и анализ / Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн; пер. с англ. – М.: Издательский дом «Вильямс», 2012. – 1296 с. (T. Cormen, R. Rivest, C. Leiserson. Introduction to algorithms. – McGraw-Hill Science/Engineering/Math, 2001).

References

- [1] Segaran T. Programming Collective Intelligence. O'Reilly Media; 2007.
- [2] Inaba M, Katoh N, Imai H. Applications of Weighted Voronoi Diagrams and Randomization to Variance-based K-clustering: (Extended Abstract). Proceedings of the Tenth Annual Symposium on Computational 1994; 10: 332-339.
- [3] Lloyd S. Least squares quantization in PCM. Information Theory, IEEE Transactions on 1982; Vol. 28(2): 129-137.
- [4] Lisin AV, Faizullin RT. Heuristic algorithm for finding approximate solution of Steiner problem based on physical analogies [in Russian]. Computer Optics 2013; 37(4): 503-510.
- [5] Dulkey VI, Faizullin RT. Approximate solving of travelling salesman problem by recursive plotting of auxiliary curve [in Russian]. Applied Discrete Mathematic 2009; 1(3): 72-78.
- [6] Sundaram RK. A First Course in Optimization Theory. Cambridge University Press; 1996.
- [7] Avriel M. Nonlinear Programming: Analysis and Methods. Dover Publishing; 2003.
- [8] Duman S, Güvenc U, Yörükeren N. Gravitational Search Algorithm for Economic Dispatch with Valve-Point Effects. International Review of Electrical Engineering 2010; 5: 2890-2895.
- [9] Rashedi E, Nezamabadi-pour H, Saryazdi S. GSA: A Gravitational Search Algorithm. Information Science 2009; 179: 2232–2248.
- [10] Zinovjev AU. Multi dimension data visualization. Krasnoyarsk State Technical University Press; 2000.
- [11] Cormen T, Rivest R, Leiserson C. Introduction to algorithms. McGraw-Hill Science/Engineering/Math; 2001.

APPLICATION OF METAHEURISTICS TO K-MEANS CLUSTERING

A.V. Lisin, R.T. Faizullin

Omsk State Technical University, Omsk, Russia

Abstract

We introduce an approach to image segmentation with a k -means algorithm by reducing the clustering problem to a continuous optimization problem, which is solved by metaheuristics algo-

rithms using a gravitational search algorithm as an example. The approach is substantiated and numerical experiment results are discussed.

Keywords: clustering, k -means, image processing, continuous optimization.

Citation: Lisin AV, Faizullin RT. Application of methaheuristics to k -means clustering. Computer Optics 2015; 39(3): 405-12.

Сведения об авторах

Лисин Андрей Владимирович, аспирант Омского государственного технического университета кафедры комплексной защиты информации. Область научных интересов: математическое моделирование, численные методы, визуализация данных, теория алгоритмов.

E-mail: andrey.lisin@gmail.com.

Andrey Vladimirovich Lisin, a graduate school student at Omsk State Technical University, Complex Information Protection department. Research interests: mathematical modeling, numerical methods, data visualization, theory of algorithms.

Файзуллин Рашит Тагирович, доктор технических наук (1999–2014гг.), профессор Омского государственного технического университета кафедры комплексной защиты информации. В списке научных работ Р.Т. Файзуллина более 100 статей, 2 монографии.

E-mail: frt@omgtu.ru.

Rashit Tagirovich Faizullin, Doctor of Technical Sciences (1999–2014), Professor of Omsk State Technical University, Complex Protection of Information department. He is co-author of more when 100 scientific papers, 2 monographs.

*Поступила в редакцию 4 мая 2015 г.
Окончательный вариант – 10 июня 2015 г.*