

---

---

**ПРОГРАММНАЯ ИНЖЕНЕРИЯ, ТЕСТИРОВАНИЕ  
И ВЕРИФИКАЦИЯ ПРОГРАММ**

---

---

УДК 004.42

## **АЛГОРИТМ КЛАСТЕРИЗАЦИИ МНОЖЕСТВА ДЕТАЛЕЙ ПО ЧЕРТЕЖАМ**

© 2020 г. В. Н. Кучуганов<sup>а,\*</sup>, А. В. Кучуганов<sup>а,\*\*</sup>, Д. Р. Касимов<sup>а,\*\*\*</sup>

<sup>а</sup> ФГБОУ ВО «Ижевский государственный технический университет им. М.Т. Калашникова»  
426069 Ижевск, ул. Студенческая, 7, Россия

\*E-mail: kuchuganov@istu.ru

\*\*E-mail: Aleks\_KAV@udm.ru

\*\*\*E-mail: kasden@mail.ru

Поступила в редакцию 15.01.2019 г.

После доработки 19.05.2019 г.

Принята к публикации 27.05.2019 г.

В статье для задач производственной логистики предлагается алгоритм кластеризации заданного множества деталей по чертежам. С целью ускорения процесса выполняется фаззификация значений каждого параметра и поиск локальных максимумов на N-мерной гистограмме, отличающийся способом выборки соседних векторов, который состоит в пересчете координат из N-мерного пространства в одномерное и обратно и сравнении с координатами соседних векторов. За счет этого алгоритм осуществляет поиск вершин кластеров за один проход и не требует создания множества локальных списков или графов соседства векторов, благодаря чему повышается эффективность алгоритма кластеризации. Приведены результаты эксперимента по автоматическому группированию деталей на основе чертежей.

DOI: 10.31857/S0132347420010045

### 1. ВВЕДЕНИЕ

Классификаторы деталей достаточно широко применяются в машиностроении с целью унификации конструкций, технологий и технологической оснастки [1]. Но применение их для разработки групповых технологий, обеспечивающих повышение эффективности производства, сдерживается проблемами автоматизации процесса группирования (кластеризации) деталей по их чертежам или 3D геометрическим моделям. Оперативная кластеризация деталей по геометрическим, техническим, технологическим, производственным параметрам могла бы повысить степень унификации технологических процессов, повысить обоснованность и качество нормирования труда, оптимизировать планы производства. Для этого необходимо привлечение методов Image Mining – автоматического извлечения признаков формы и относительного расположения структурных элементов из чертежей, 3D геометрических моделей, технологических эскизов.

На современных предприятиях применяются системы класса PLM (Product Lifecycle Management), решающие задачи накопления и управления всей информацией, возникающей на этапах конструкторского и технологического проектирования, изготовления, сопровождения и т.д. [2]. Благодаря этому практически у каждого чертежа имеется подробный “паспорт”, что позволяет ис-

пользовать эту информацию и для автоматизации группирования деталей.

Целью данной работы является создание эффективного алгоритма кластеризации машиностроительных деталей по набору параметров, включающего:

- геометрические признаки, которые могут быть извлечены из чертежей деталей с помощью алгоритмов автоматической векторизации чертежей, структурного анализа и распознавания типовых конструктивных элементов (ТКЭ);

- дополнительные технологические параметры, которые могут быть получены из электронного архива чертежей.

Особенностью предлагаемого алгоритма является фаззификация значений каждого параметра, что помимо ускорения процесса кластеризации также дает пользователю отчетливую обратную связь в качественных терминах.

Статья организована следующим образом. В разделе 2 анализируются существующие методы кластеризации объектов, в том числе по параметрам формы. В разделе 3 описывается укрупненный алгоритм кластеризации. В разделе 4 рассматриваются функции пересчета координат из одномерного массива в многомерное пространство и обратно, посредством которых устанавливается соседство (близость) объектов. В разделе 5 описа-

ны эксперименты по автоматическому группированию деталей на основе чертежей. Заключение по работе дается в разделе 6.

## 2. ОБЗОР МЕТОДОВ КЛАСТЕРИЗАЦИИ

В настоящее время существует множество методов кластерного анализа. С их помощью аналитики получают возможность классифицировать информацию об объектах, т.е. на основе известной информации группировать объекты в целях анализа, лучшего понимания, принятия управленческих решений.

Наиболее популярными можно считать методы, использующие подход на основе  $k$ -средних, и гистограммные методы.

Алгоритм кластеризации методом  $k$ -средних ( $k$ -means) [3] минимизирует среднеквадратичное отклонение векторов признаков с целью получения  $k$ -кластеров – компактных множеств объектов, близких по свойствам. Алгоритм является итерационным, т.е. требует значительных ресурсов. Недостатком считается отсутствие возможности определить реально существующее количество кластеров. Чтобы уменьшить объем вычислений обычно сокращают количество итераций, но при этом появляется риск пропустить кластеры, важные для решаемой прикладной задачи.

Гистограммные методы отыскивают все вершины кластеров за один проход, но это требует слишком больших вычислений. Чтобы ускорить процесс, предварительно осуществляют квантование пространства путем округления значений параметров. Так, в работе [4] предлагается осуществлять разбиение векторного пространства на множество подпространств и строить многомерную гистограмму полученных векторов, исключая пустые подпространства. Список этих векторов (подпространств) организован по типу алфавитного словаря, т.е. в порядке возрастания значений цифр очередной координаты. Для удобства анализа соседних подпространств строится множество локальных графов векторов и их соседей. Далее применяется гистограммный алгоритм Нарендры [5], не требующий задания числа кластеров или количества итераций. Время классификации линейно зависит от числа различных векторов. С помощью алгоритма осуществлялась сегментация областей на спутниковых снимках по ряду спектральных каналов.

Кластеризация широко применяется для сегментации изображений по цветовым характеристикам пикселей. Так, в статье [6] проблема сегментации изображения решается посредством иерархической последовательности кусочно-постоянных приближений с использованием метода кластеризации Уорда [7], применяющего принципы дисперсионного анализа для оценки

расстояний между кластерами. Мерой различия объектов служит евклидово расстояние, которое, к сожалению, невозможно применить к атрибутам, которые по своей сути являются качественными и не могут быть выражены в количественной шкале. А в задаче группирования деталей такого рода атрибуты (например: вид детали, тип заготовки, группа материала и т.п.) имеют существенное значение, и их не следует игнорировать.

Нечеткие классификаторы часто применяют в системах поддержки принятия управленческих решений при анализе больших объемов данных. В работе [8] задачу построения нечетких классификаторов, формулируемую как задачу отбора признаков, генерации базы нечетких правил, оптимизации условий правил, предлагается решать путем отбора классифицирующих признаков на основе бинарного гармонического поиска [9], использования метаэвристических алгоритмов непрерывного гармонического поиска, являющихся дальнейшим развитием эволюционных вычислений, а также путем генерации базы правил по экстремальным значениям признаков. Эффективность предлагаемого подхода проверена на реальных данных из KEEL и KDD Cup хранилищ текстовых сообщений. К сожалению, для построения классификаторов в задачах производственной логистики такой принцип, как правило, не подходит, поскольку производственники – конструкторы, технологи, плановики – обычно сами решают (назначают) критерии группирования, которые они считают важнейшими в той или иной производственной ситуации.

Для кластеризации объектов по параметрам формы обычно используют статистические характеристики, в частности, скелетные и контурные дескрипторы. Так в работе [10] для повышения надежности кластеризации предлагается алгоритм спектральной кластеризации, основанный на спектральном анализе множества пикселей, составляющих изображение. Алгоритм не требует представления формы в виде вектора.

В работе [11] предлагается осуществлять кластеризацию плоских фигур путем формирования графов скелетов и обнаружения общей структуры фигур на основе результатов сопоставления графов. Используется стандартная агломеративная итерационная схема кластеризации. В каждой итерации графы скелетов из двух кластеров объединяются в один граф, вычисляется мера сходства в качестве критерия близости кластеров. Эксперименты выполнены на популярных тестах для форм – рисунках людей, животных, рыб и т.п. – проблема весьма актуальная в задачах поиска изображений по контенту (CBIR – Content-Based Image Retrieval).

Коммерчески предлагаемая система CADFind [12, 13] способна выполнять поиск машинострои-

тельных чертежей и 3D моделей по произвольному образцу, задаваемому пользователем, а также производить автоматическое группирование деталей по их чертежам и 3D моделям. Система автоматически идентифицирует потенциальные семейства деталей и позволяет уточнять их в интерактивном режиме. В доступных описаниях принципов работы этой системы указывается, что она индексирует геометрию детали, а также текстовую информацию, связанную с ней, включая материал, техпроцесс. Извлекаемая из чертежа/3D модели информация представляется в виде кода групповой технологии (GT кода) – строки буквенно-цифровых символов. Сопоставление деталей происходит путем нечеткого сравнения соответствующих GT кодов. К сожалению, ни на официальном сайте продукта, ни в публикациях авторов нам не удалось найти более подробной информации о том, какие именно геометрические признаки детали описывает формируемый GT код, каким образом осуществляется автоматическое извлечение этих признаков из чертежей и 3D моделей, в чем заключается нечеткий характер процедуры сравнения GT кодов. Не раскрывается также и применяемый метод кластеризации, на котором базируется процесс автоматического группирования деталей. Остается только сделать вывод о том, что проблема весьма актуальна, а нацеленные на ее решение программные системы – востребованы.

В части контентного поиска машиностроительных чертежей по образцу существуют также другие исследовательские работы (например, [14, 15]). Однако в этих работах не делается акцент на задаче автоматической классификации деталей.

### 3. УКРУПНЕННЫЙ АЛГОРИТМ КЛАСТЕРИЗАЦИИ

Предлагаемый алгоритм кластеризации относится к категории гистограммных и состоит в следующем:

**1. Подготовка исходных данных.** Подбор репрезентативной выборки объектов, заполнение массива измерений (таблица 1), где  $M$  – количество объектов в обучающей выборке,  $N$  – количество параметров.

**2. Фаззификация значений атрибутов,** описывающих объекты. Предварительная (перед собственно кластеризацией) фаззификация параметров существенно сокращает объем вычислений.

В результате фаззификации для каждого параметра  $p_i$ ,  $i \in \{1 \dots N\}$  с диапазоном значений  $D_i$  получаем линейно упорядоченное множество качественных значений параметра  $F_i = \{p_{ij}\}$ , где  $j \in \{1 \dots h(i)\}$ ,  $h(i)$  – количество качественных значений параметра  $p_i$  и (сюрьективную) функцию  $val_i: D_i \rightarrow F_i$ , которая сопоставляет каждому возможному значению параметра  $p_i$  соответствующее

**Таблица 1.** Структура массива измерений

Объекты	Параметры			
	1	2	$\vdots$	N
1				
2				
.				
.				
M				

щее качественное значение. За счет этого диапазон значений каждого параметра  $p_i$  разбивается на конечное количество качественных значений  $D_{ij} = \{x \in D_i | val_i(x) = p_{ij}\}$ .

**3. Построение N-мерной гистограммы.** Гиперпространство содержит не более чем  $H^N$  локусов (ячеек, подпространств), где  $H$  – максимальное количество различных качественных значений параметра (например, по каждому из параметров может быть получено одно из  $H = 5$  возможных качественных значений). В цикле по массиву измерений сформировать массив локусов (таблица 2), где для каждого локуса указать количество объектов в нем, т.е. объектов, имеющих соответствующие качественные значения всех  $N$  параметров. В массиве измерений отметить, к какому локусу относится объект. Номер кластера на данном этапе отсутствует.

**4. Поиск локальных максимумов в N-мерном пространстве измерений.** Для этого с помощью *локального анализатора окрестности локуса* просканировать пространство и найти такие локусы, в которых по каждому из  $N$  направлений выполняется условие:

$$\forall l = 1..L (\exists i \in \{i_1, \dots, i_N\} (w(F1(i_1, \dots, h(i) - 1, \dots, i_N))) < w(l) \geq w(F1(i_1, \dots, h(i) + 1, \dots, i_N)))) \Rightarrow L_l \in \{C\}, \quad (1)$$

где:  $l = 1..L$  – индекс локуса в одномерном массиве  $L$  локусов;

$F1(i_1, \dots, i_N)$  – функция, вычисляющая положение локуса в одномерном массиве  $L$ , где  $i_1, \dots, i_N$  – координаты локуса в  $N$ -мерном пространстве измерений;

$h(i) = 0..H$  – положение локуса вдоль  $i$ -той оси координат,  $i \in \{i_1, \dots, i_N\}$ ;

$w(l)$ ,  $w(F1)$  – вес локуса (количество объектов, по своим значениям попавших в соответствующее подпространство);

$\{C\}$  – множество локальных максимумов.

Найденные локусы в гиперпространстве параметров служат “вершинами” кластеров – скопления объектов с близкими характеристиками. Получаем список кластеров

Таблица 2. Массив локусов

Локус	Координаты (качественные) локуса				№ кластера	Вес (количество объектов)
	1	2	...	N		
1					1	w(1)
.					.	.
.					.	.
.					.	.
L					K	w(L)

$$c_k = \langle w(k), x_1, \dots, x_N \rangle, \quad k = 1..K,$$

где:  $x_1, \dots, x_N$  – координаты вершины кластера, удовлетворяющего условию (1) в  $N$ -мерном пространстве.

**5. Определение границ кластеров.** Для этого из вершины кластера  $c_k$ , имеющего координаты  $x_1, \dots, x_N$ , выше описанный локальный анализатор запускается рекурсивно. Если вес  $w(F1(x_1, \dots, h(i) \pm 1, \dots, x_N))$  соседнего по какому-либо  $i$ -тому направлению локуса меньше или равен весу локуса, принадлежащего кластеру, то этот соседний локус присоединяется к тому же кластеру:

$$\begin{aligned} \forall l \in \{1..H^N\} (\exists i \in \{x_1, \dots, x_N\} \\ w(F1(x_1, \dots, h(i) \pm 1, \dots, x_N)) \\ \leq w(F1(x_1, \dots, x_N))) \\ \Rightarrow L(F1(x_1, \dots, h(i) \pm 1, \dots, x_N)) \in c_k; \end{aligned}$$

где:  $N$  – количество параметров;

$h(i) = 0..H$  – положение локуса вдоль  $i$ -той оси координат,  $i \in \{x_1, \dots, x_N\}$ ;

$H$  – количество возможных значений  $i$ -го параметра;

$l$  – индекс локуса в одномерном массиве локусов  $L$ ;

$L(F1(x_1, \dots, h(i) \pm 1, \dots, x_N))$  – локус соседний с анализируемым;

$c_k$  – кластер с номером  $k$ .

Очевидно, таким же образом процесс кластеризации можно осуществлять в каждом из полученных кластеров – фаззифицировать и кластеризовать, по вышеописанному алгоритму не всю “обучающую” выборку, а уже полученный кластер, что дает возможность более детально анализировать интересующие области пространства измерений и детализировать классификацию объектов.

Функция, определяющая принадлежность нового объекта тому или иному кластеру – это мера относительной близости, вычисляемая как расстояние в  $N$ -мерном метрическом пространстве между объектом и каждым из “центров” кластеров.

#### 4. ФУНКЦИИ ПЕРЕСЧЕТА КООРДИНАТ

Пусть  $N$  – количество осей пространства параметров,  $H$  – количество возможных значений каждого параметра.

Создадим одномерный (линейный) массив  $L$  локусов. Количество элементов в массиве будет  $H^N$ .

Координаты  $\langle x_1, x_2, \dots, x_N \rangle$  локуса (ячейки) в  $N$ -мерном пространстве запишем в виде числа в позиционной системе счисления с основанием  $H$ . Это число содержит  $N$  цифр, а каждая цифра может принимать значения от 0 до  $H - 1$ .

Чтобы вычислить значение индекса  $l$  элемента в одномерном массиве  $L$ , достаточно отобразить координаты локуса в  $H$ -ричное число и перевести это число из позиционной системы счисления с основанием  $H$  в десятичную систему счисления. И наоборот, если известен номер локуса в одномерном массиве, то для определения его координат в  $N$ -мерном пространстве нужно перевести этот номер в  $H$ -ричное число. Для выборки ближайших к нему соседних по  $i$ -той оси локусов нужно увеличить или уменьшить на 1  $i$ -тую цифру  $H$ -ричного числа и полученные числа перевести в десятичные. Чтобы проверить всех ближайших соседей, то же самое надо сделать по всем другим цифрам  $H$ -ричного числа.

Если в одномерном массиве не хранить пустые локусы, то в оставшихся должны быть в качестве ключа записаны их координаты с целью извлечения нужных локусов. Выбор того или другого способа хранения информации зависит от ожидаемого числа пустых ячеек.

С другой стороны, существование пустых локусов может оказаться интересным для какой-то предметной области, например, в социологических исследованиях, где параметры, как правило, качественные.

Перевод  $H$ -ричного числа в десятичную систему и обратно (см. например, [16]) осуществляется по формулам:

$$X = \sum_{k=-n}^m R_k * B^k, \quad (2)$$

где  $X$  – значение числа в десятичной системе,  $B$  – основание  $H$ -ричной системы счисления,  $R$  – цифра  $H$ -ричного числа,  $k$  – позиция цифры в записи числа.

Алгоритм поиска центров кластеров выглядит следующим образом:

```

k := 0;
FOR l = 1..L // Цикл по всем локусам, L = HN
  IF p[l] = 0 THEN Continue(l); // Переход к следующему l
  FN(l, ZH); // Перевод l → ZH в H-ричную систему счисления (с основанием H)
  // Сформировать массив (или список) из цифр числа ZH
  FOR i = 1..N // Цикл по цифрам числа ZH
    // Сформировать число ZL такое, что i-тая цифра ZL меньше на 1,
    // чем i-тая цифра в ZH
    F1(ZL, ll); // Выдает индекс ll соседнего локуса в одномерном массиве L
    IF p[l] < p[ll] THEN Continue(l); // Переход к следующему l
    // Сформировать ZR такое, что i-тая цифра ZR больше на 1, чем i-тая цифра в ZH
    F1(ZR, lr); // Выдает индекс lr соседнего локуса в одномерном массиве L
    IF p[l] ≤ p[lr] THEN Continue(l); // Переход к следующему l
  END i
  k := k + 1; c[k] := ⟨ p[l], l, ZH ⟩ // Центр k-го кластера
END l

```

Функции  $FN$ ,  $F1$  — это несколько команд. Можно не искать границы кластеров. Обычно объект относят к тому центру, до которого ближе.

Сложность алгоритма не превосходит значения функции  $O$ :

$$O(H^N \times N),$$

где  $N$  — количество параметров;

$H$  — максимальное количество различных качественных значений параметра.

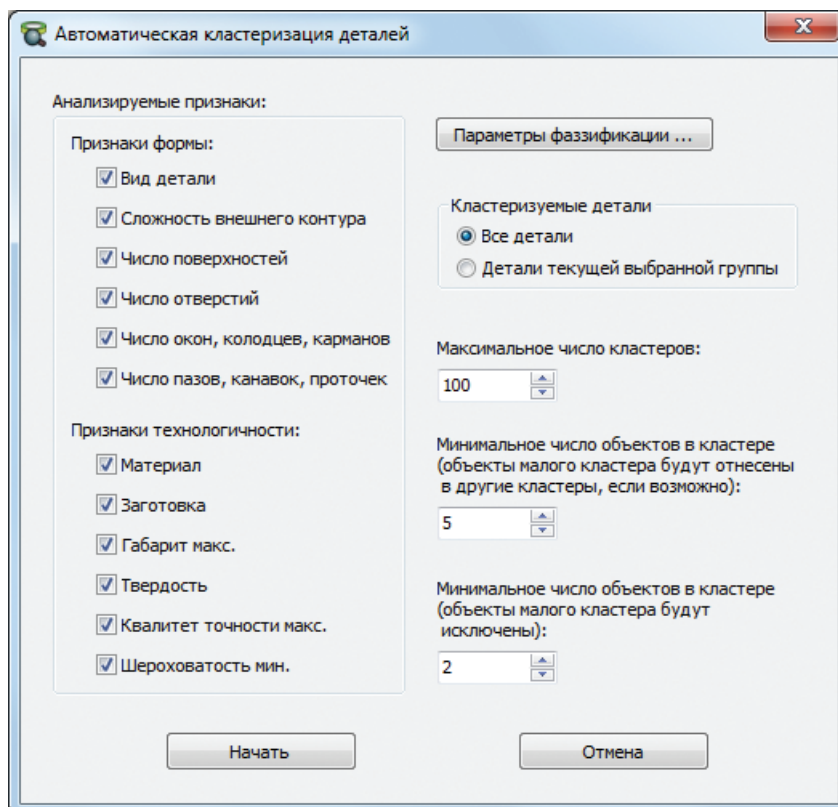
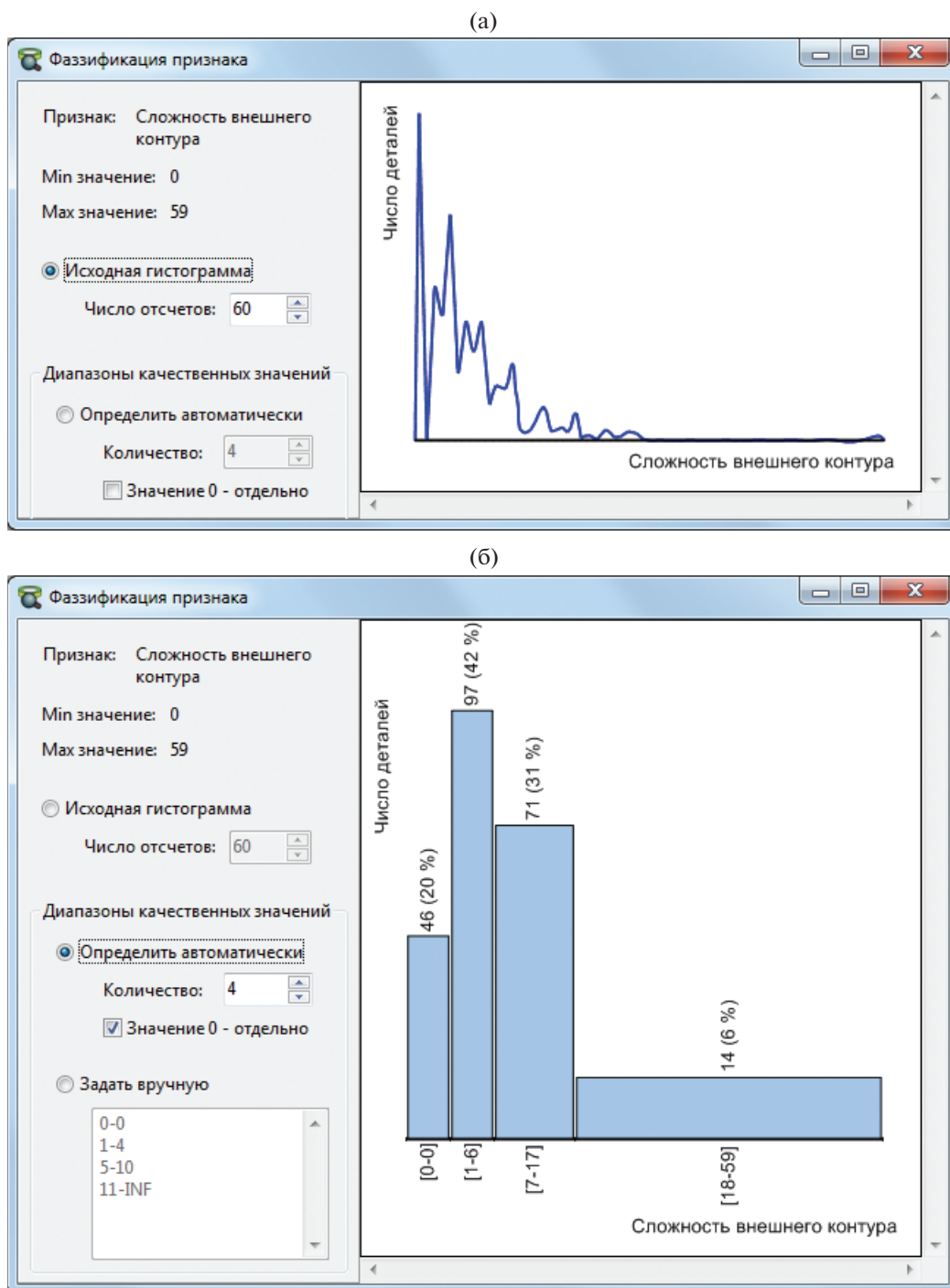


Рис. 1. Окно настроек кластеризации.



**Рис. 2.** Инструментарий перевода количественных значений признака в качественные: а) исходная гистограмма распределения количественных величин; б) диапазоны качественных значений, предложенные системой (автоматически).

В практических задачах это зачастую оказывается меньше, чем линейная зависимость от числа анализируемых объектов, поскольку редко требуется оперировать более чем 5 качественными значениями параметра, и числом анализируемых па-

раметров обычно управляет лицо, принимающее решение, с целью повышения наглядности результатов. А если параметров много, то применяют *факторный анализ*, заменяя группу свойств одним обобщенным свойством.

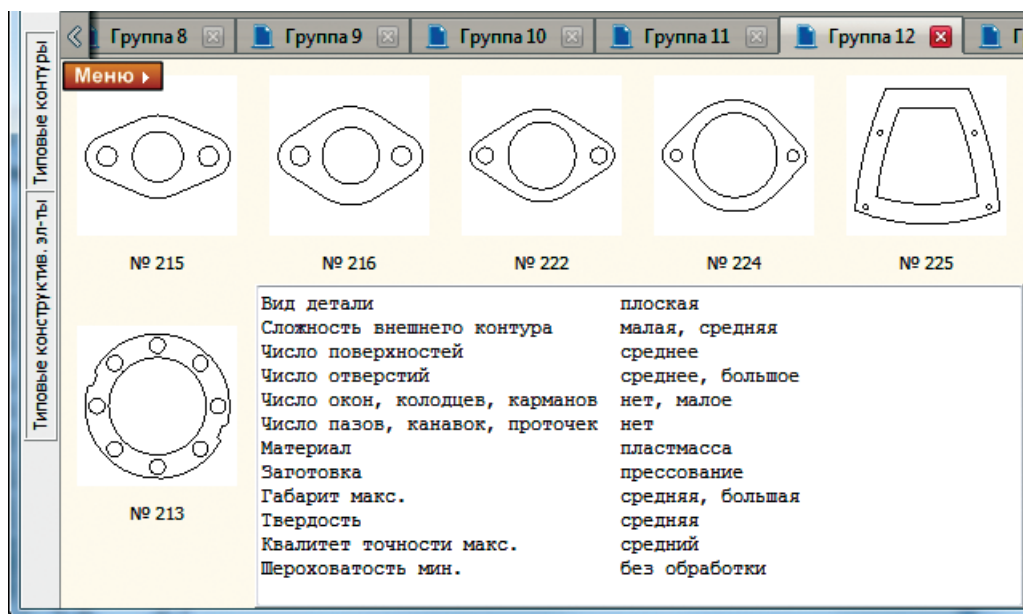


Рис. 3. Вывод результатов кластеризации.

### 5. ЭКСПЕРИМЕНТ

Алгоритм кластеризации, описанный выше, был использован для синтеза классификатора деталей.

В эксперименте анализировались такие параметры деталей как: вид детали (корпусная, плоская, тело вращения); сложность внешнего контура (количество перегибов – смен знака приращения угла наклона); число поверхностей, рассчитываемое на основе выделения минимальных примыкающих контуров; количество простых и ступенчатых отверстий; количество окон, колодцев и карманов; количество пазов, канавок и проточек. Дополнительные параметры классификации: группа материала (сталь, чугун, силумин, пластмасса); заготовка (пруток, плита, литье, прессование, прокат); габариты; твердость; квалитет точности; шероховатость (без обработки, шлифование/точение, полирование, зеркальная). Общее количе-

ство анализируемых параметров – 12. Числовые параметры преобразовывались в качественные значения (нет, мало, средне, много).

На рис. 1 показано окно запуска процесса кластеризации деталей. Здесь пользователь, обычно, конструктор-технолог, выбирает анализируемые признаки, с помощью инструмента “Параметры фаззификации” корректирует функции преобразования количественных значений признаков в качественные, указывает множество кластеризуемых деталей и задает ограничения на количество и размер целевых кластеров. Чем больше кластеров он хочет получить, тем точнее будет выполнена кластеризация. Соответственно, придется разрабатывать больше групповых техпроцессов.

Процесс перевода количественных значений признаков в качественные является одним из инструментов управления точностью и производительностью процесса кластеризации. На рис. 2 показано окно программы, содержащее средства анализа отдельного признака. Первое, что видит эксперт в данном окне – гистограмму распределения значения признака в выборке деталей (рис. 2а). По нажатию на какой-либо отсчет гистограммы, на экране отображаются примеры деталей, имеющих соответствующее значение признака. Получив общую характеристику набора данных, эксперт затем определяет:

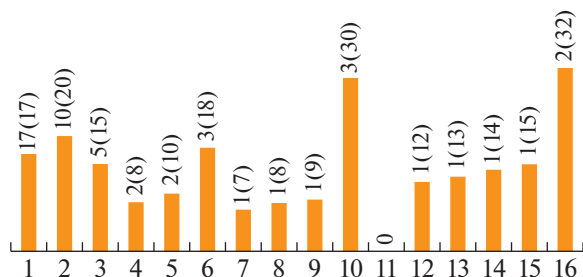


Рис. 4. Распределение деталей по кластерам (по горизонтальной оси – размер кластера, по вертикальной оси – число кластеров и общее количество деталей в них).

1) число качественных значений признака, которыми целесообразно оперировать в процессе кластеризации;

2) соответствия между качественными и количественными значениями.

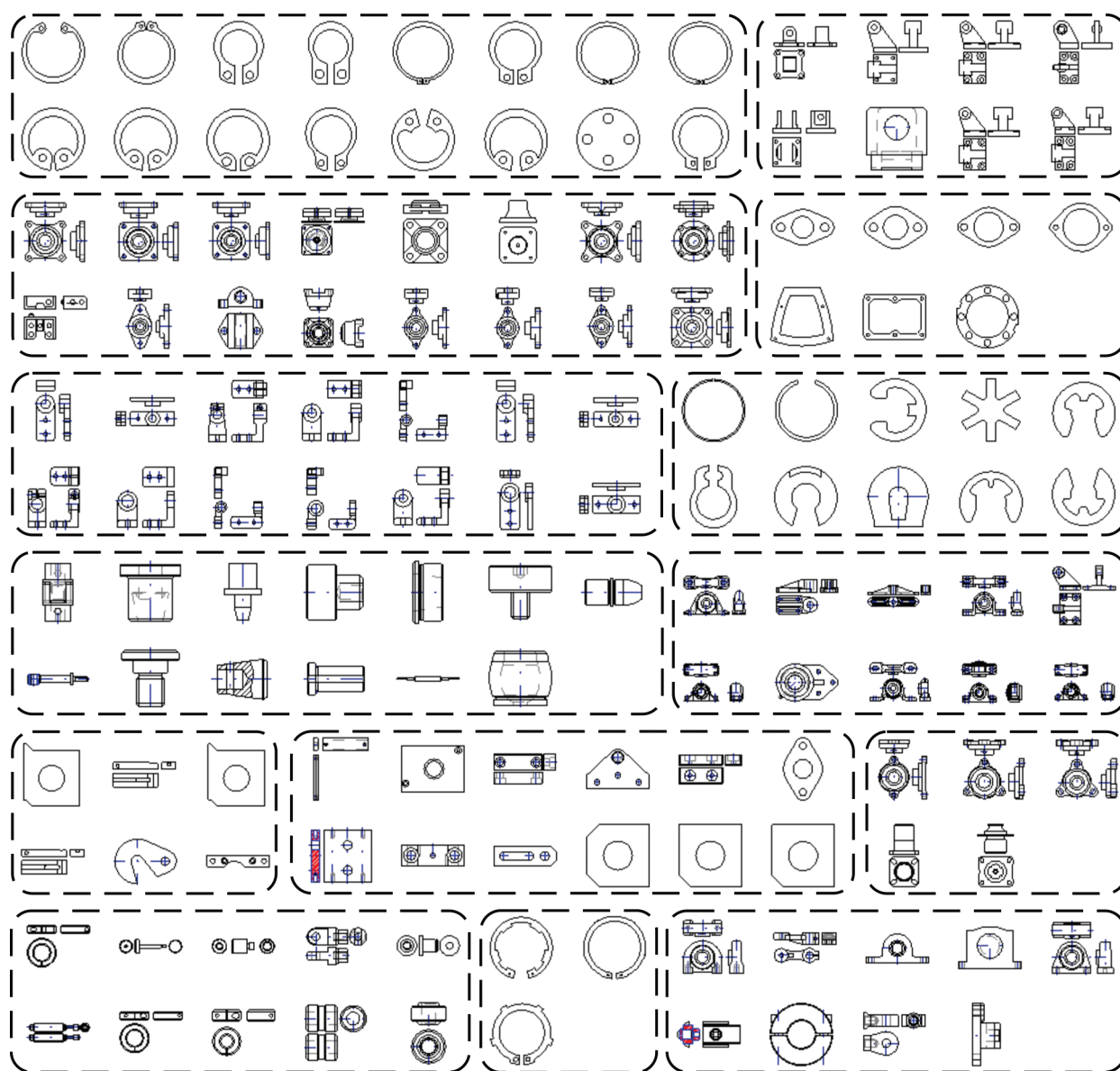


Рис. 5. Примеры полученных кластеров.

Качественное значение представляет определенный диапазон количественных величин. Основная задача, возлагаемая на эксперта, состоит в установлении границ диапазонов. Для упрощения этого процесса, мы предусмотрели возможность автоматического получения рекомендуемых границ диапазонов (рис. 26). Программа выявляет на исходной гистограмме локальные максимумы (в количестве, равном заданному числу целевых диапазонов). В примере на рис. 26 были получены следующие локальные максимумы (отсчеты): 0, 4, 12, 25. Затем к ним программа относит остальные отсчеты гистограммы по критерию близости. В результате формируется опор-

ная разбивка на диапазоны, которую эксперт может затем подкорректировать в ручном режиме.

На рис. 3 приведен пример результата кластеризации. При визуализации полученных кластеров помимо чертежей деталей отображаются также значения их признаков в качественном представлении. Это помогает инженеру интерпретировать результаты.

Для экспериментального исследования был подготовлен набор данных, содержащий 98 чертежей корпусных деталей, 82 чертежа плоских деталей и 48 чертежей тел вращения, всего 228 деталей. Чертежи деталей и часть данных о материале и габаритах взяты из открытых онлайн-каталогов (TraceParts, PARTcommunity, Mitutoyo) и архивов



студенческих работ. Значения дополнительных технологических параметров, отсутствующие в первоисточнике, задавались вручную экспертом-технологом. При необходимости выполнялась векторизация сканированных чертежей [17]. Извлечение признаков формы и распознавание ТКЭ на чертежах осуществляла разработанная подсистема графического поиска чертежей [18].

*Синтаксическое распознавание* ТКЭ на чертежах осуществлялось с помощью порождающих грамматик, их описывающих. Ниже показаны примеры функций описания конструктивных элементов деталей: “окно” – сквозная прорезь в стенке, “колодец”, “карман” – углубление, у которого один край совпадает с краем детали:

$\langle \text{Элемент} \rangle ::= \text{type}, x_s, y_s, x_e, y_e, x_c, y_c, r, \alpha, \beta,$   
где  $\text{type}$  – тип элемента,  $\text{type} \in \{\text{Line}, \text{Arc}, \text{Circle}\};$   
 $x_s, y_s$  – координаты точки начала элемента;  
 $x_e, y_e$  – координаты точки конца элемента;  
 $x_c, y_c$  – координаты центра дуги/окружности;  
 $r$  – радиус дуги/окружности;  
 $\alpha, \beta$  – углы векторов от центра к концам дуги.  
 $\langle \text{Цепочка элементов} \rangle ::= e_1, e_2, \dots, e_k : \langle \text{Усл1} \rangle, \langle \text{Усл2} \rangle,$   
где  $\langle \text{Усл1} \rangle ::= \forall i \in [1..k] (e_i \text{ is } \langle \text{Элемент} \rangle \wedge e_i.\text{type} \neq \text{Circle});$   
 $\langle \text{Усл2} \rangle ::= \forall i \in [1..k-1] (\text{Соединяются}(e_i, e_{i+1}) \wedge \forall j \in [i+2..k] \neg \text{Пересекаются}(e_i, e_j)).$   
 $\langle \text{Контур} \rangle ::= (\langle \text{Цепочка элементов} \rangle : \langle \text{Усл1} \rangle) | (\langle \text{Элемент} \rangle : \langle \text{Усл2} \rangle),$   
где  $\langle \text{Усл1} \rangle ::= \text{Соединяются}(e_k, e_1),$   
 $\langle \text{Усл2} \rangle ::= \text{type} = \text{Circle}.$   
 $\langle \text{Узел} \rangle ::= e_1, e_2, e_3 : \langle \text{Усл1} \rangle, \langle \text{Усл2} \rangle,$   
где  $\langle \text{Усл1} \rangle ::= \forall i \in [1..3] e_i \text{ is } \langle \text{Элемент} \rangle;$   
 $\langle \text{Усл2} \rangle ::= \text{Соединяются}(e_1, e_2) \wedge \text{Соединяются}(e_2, e_3) \wedge \text{Соединяются}(e_1, e_3).$   
 $\langle \text{2СвязанныхКонтура} \rangle ::= C_1, C_2 : \langle \text{Усл1} \rangle, \langle \text{Усл2} \rangle,$   
где  $\langle \text{Усл1} \rangle ::= (C_1 \text{ is } \langle \text{Контур} \rangle) \wedge (C_2 \text{ is } \langle \text{Контур} \rangle);$   
 $\langle \text{Усл2} \rangle ::= \text{ПроекционноСвязаны}(C_1, C_2).$   
 $\langle \text{Окно} \rangle ::= \langle \text{2СвязанныхКонтура} \rangle : \langle \text{Усл1} \rangle, \dots, \langle \text{Усл3} \rangle,$   
где  $\langle \text{Усл1} \rangle ::= \neg \exists U (U \text{ is } \langle \text{Узел} \rangle \wedge U \cap C_1 \neq \emptyset);$   
 $\langle \text{Усл2} \rangle ::= \forall i \in [1..k] e_i \in C_1 (\text{ПриращениеУглаВектораКасательной}(e_i, e_{i+1}) \leq 0);$   
 $\langle \text{Усл3} \rangle ::= \exists_{\geq 4} U (U \text{ is } \langle \text{Узел} \rangle \wedge U \cap C_2 \neq \emptyset).$   
 $\langle \text{Колодец} \rangle ::= \langle \text{2СвязанныхКонтура} \rangle : \langle \text{Усл1} \rangle, \dots, \langle \text{Усл3} \rangle,$   
где  $\langle \text{Усл1} \rangle ::= \neg \exists U (U \text{ is } \langle \text{Узел} \rangle \wedge U \cap C_1 \neq \emptyset);$   
 $\langle \text{Усл2} \rangle ::= \forall i \in [1..k] e_i \in C_1 (\text{ПриращениеУглаВектораКасательной}(e_i, e_{i+1}) \leq 0);$   
 $\langle \text{Усл3} \rangle ::= \exists_{=2} U (U \text{ is } \langle \text{Узел} \rangle \wedge U \cap C_2 \neq \emptyset).$   
 $\langle \text{Карман} \rangle ::= \langle \text{2СвязанныхКонтура} \rangle : \langle \text{Усл1} \rangle, \langle \text{Усл2} \rangle,$   
где  $\langle \text{Усл1} \rangle ::= \exists_{=2} U (U \text{ is } \langle \text{Узел} \rangle \wedge U \cap C_1 \neq \emptyset);$   
 $\langle \text{Усл2} \rangle ::= \exists_{=2} U (U \text{ is } \langle \text{Узел} \rangle \wedge U \cap C_2 \neq \emptyset).$

Описание ТКЭ выполнено с помощью аппарата формальных грамматик и первопорядковой логики предикатов.

На этапе определения локальных максимумов было получено 89 кластерообразующих точек. В результате рекурсивного поиска границ кластеров была получена 51 группа деталей. Общее время работы алгоритма с учетом загрузки информации из базы данных составило 8 секунд на персональном компьютере.

На рис. 4 показана характеристика полученных кластеров. В значимые кластеры (размером от 2 шт.), представляющие пользу для последующей реализации групповой технологии, было отнесено 211 деталей (92.5%).

На рис. 5 показаны некоторые из полученных кластеров.

С точки зрения эксперта-технолога, в значимых кластерах неверно классифицированными оказались 13 деталей, и еще 3 детали, попавшие в кластеры-синглтоны, следовало отнести в значи-

мые кластеры. В целом, точность кластеризации составила 0.93.

Некоторые ошибки классификации были обусловлены тем, что не всегда удавалось правильно извлечь признаки геометрической формы из чертежей или используемый набор характеристик был недостаточен. Требуется усовершенствовать методы автоматического анализа чертежей, ввести новые признаки, учитывающие относительное расположение структурных элементов.

Некоторые кластеры (7 шт.) требуют повторной (иерархической) кластеризации – это удобнее для разработки групповых технологий. Чем меньше размер группы деталей, тем проще технологю с ней работать.

## 6. ЗАКЛЮЧЕНИЕ

Таким образом, предлагаемый алгоритм кластеризации множества деталей по чертежам на основе N-мерной гистограммы отличается от известных способом выборки соседних векторов, который состоит в пересчете координат из N-мерного пространства в одномерное и обратно и сравнении с координатами соседних векторов. Алгоритм осуществляет поиск вершин кластеров за один проход. Основным достоинством алгоритма является отсутствие необходимости создания множества локальных списков или графов соседних векторов, благодаря чему повышается эффективность алгоритма кластеризации. Также следует отметить, что алгоритм осуществляет кластеризацию деталей в терминах, обычно употребляемых технологами, за счет механизма перевода количественных значений признаков в качественные.

Оперативная (например, в ходе планирования позаказного многономенклатурного производства) кластеризация деталей по геометрическим, техническим, технологическим, производственным параметрам позволяет повысить степень унификации технологических процессов и за счет этого оптимизировать план производства.

## 7. БЛАГОДАРНОСТИ

Исследование алгоритма выполнено Касимовым Д.Р. за счет гранта Российского научного фонда (проект № 18-71-00109).

## СПИСОК ЛИТЕРАТУРЫ

1. *Halevi G.* Expectations and Disappointments of Industrial Innovations. Lecture Notes in Management and Industrial Engineering. Springer, 2017. 131 p.

2. *Barladian B.Kh., Voloboy A.G., Galaktionov V.A., Shapiro L.Z.* Integration of Realistic Computer Graphics into Computer-Aided Design and Product Lifecycle Management Systems // Programming and Computer Software. 2018. V. 44. Is. 4. P. 225–232.
3. *MacQueen J.* Some methods for classification and analysis of multivariate observations // In Proc. 5th Berkeley Symp. on Math. Statistics and Probability. 1967. P. 281–297.
4. *Sidorova V.S.* Histogram Hierarchical Algorithm and the Reduction of the Dimensionality of the Spectral Features Space // Zh. Sib. Fed. Univ., Ser. Tekhn. Tekhnol. 2017. V. 10. Is. 6. P. 714–722.
5. *Narendra P.M., Goldberg M.* A non-parametric clustering scheme for LANDSAT // Pattern Recognition. 1977. V. 9. Is. 4. P. 207–215.
6. *Kharinov M.V.* Pixel clustering for color image segmentation // Programming and Computer Software. 2015. V. 41. Is. 5. P. 258–266.
7. *Ward J.H.* Hierarchical grouping to optimize an objective function // Journal of the American Statistical Association. 1963. V. 58. Is. 301. P. 236–244.
8. *Hodashinsky I.A., Mekh M.A.* Fuzzy classifier design using harmonic search methods // Programming and Computer Software. 2017. V. 43. Is. 1. P. 37–46.
9. *Wang L., Yang R., Xu Y., Niu Q., Pardalos P.M., Fei M.* An improved adaptive binary harmony search algorithm // Information Sciences. 2013. V. 232. P. 58–87.
10. *Bai S., Liu Z., Bai X.* Co-spectral for robust shape clustering // Pattern Recognition Letters. 2016. V. 83. P. 3. P. 388–394.
11. *Shen W., Wang Y., Bai X., Wang H., Latecki L.J.* Shape clustering: Common structure discovery // Pattern Recognition. 2013. V. 46. Is. 2. P. 539–550.
12. *Barton J., Love D.* Retrieving designs from a sketch using an automated GT coding and classification system // Production Planning & Control. 2005. V. 16. Is. 8. P. 763–773.
13. CADFind. URL: <http://www.cadfind3d.com/>
14. *Liu R., Wang Y., Baba T., Masumoto D.* Shape detection from line drawings with local neighborhood structure // Pattern Recognition. 2010. V. 43. Is. 5. P. 1907–1916.
15. *Fonseca M.J., Ferreira A., Jorge J.A.* Sketch-Based Retrieval of Vector Drawings // Sketch-based Interfaces and Modeling. 2011. V. 12. P. 181–204.
16. *Knuth D.* The Art of Computer Programming. Volume 2: Seminumerical Algorithms. 3-rd Ed. Massachusetts: Addison–Wesley, 1997. 762 p.
17. *Kasimov D.R., Kuchuganov A.V., Kuchuganov V.N.* Vectorization of Raster Mechanical Drawings on the Base of Ternary Segmentation and Soft Computing // Programming and Computer Software. 2017. V. 43. Is. 6. P. 337–344.
18. *Kasimov D.R., Kuchuganov A.V., Kuchuganov V.N.* Individual Strategies in the Tasks of Graphical Retrieval of Technical Drawings // Journal of Visual Languages and Computing. 2015. V. 28. P. 134–146.