

ПРИМЕНЕНИЕ SMART-ТАБЛИЦ И ОЦЕНОК ЦЕЛЕВОЙ ФУНКЦИИ ДЛЯ СНИЖЕНИЯ РАЗМЕРНОСТИ И УСКОРЕНИЯ РЕШЕНИЯ ЗАДАЧ CONSTRAINED CLUSTERING

© 2021 А. А. Зуенко✉, О. В. Фридман, О. Н. Зуенко

*Институт информатики и математического моделирования —
обособленное подразделение Федерального исследовательского центра
«Кольский научный центр Российской академии наук» (ИИММ КНЦ РАН)
ул. Ферсмана, 24А, 184209 Апатиты, Мурманской обл., Российская Федерация*

Аннотация. В статье представлен комплексный подход к точному решению задач *Constrained Clustering*, то есть задач кластеризации, предполагающих анализ, помимо матрицы расстояний, фоновых знаний о необходимости/недопустимости вхождения некоторых объектов в те или иные кластеры. Подход реализован в рамках парадигмы программирования в ограничениях (*Constraint Programming*), ориентированной на построение процедур систематического поиска (процедур обхода дерева поиска) для решения сложных комбинаторных задач. При этом, вся исходная информация о задаче выражается с помощью ограничений, то есть качественных и количественных зависимостей. Существенная сложность заключается в том, что в современных средах и библиотеках программирования в ограничениях обработка качественных ограничений, которыми, в частности, являются правила отнесения объектов к одному или различным кластерам, производится недостаточно эффективно. Таким образом, представляется актуальной разработка способов ускорения обработки подобных ограничений. В статье предлагается представлять и обрабатывать качественные ограничения в форме табличных ограничений нового типа, а именно *smart*-таблиц *D*-типа. Для *smart*-таблиц *D*-типа разработаны высокоэффективные процедуры вывода на ограничениях, осуществляющие раннее отсечение неперспективных ветвей дерева поиска. Другое направление работ, которое активно развивается в настоящих исследованиях, связано с уменьшением количества ограничений, используемых для представления задачи, и с упрощением их вида. Предлагается генерировать ограничения лишь для некоторых пар объектов, основываясь на интервальной оценке для оптимального значения критерия кластеризации. Для получения данной оценки используется ранее предложенный авторами метод иерархической кластеризации, который позволяет анализировать ограничения на комбинации пар объектов внутри кластера. Предложенный подход позволяет находить все варианты разбиений, обеспечивающие глобальный оптимум целевой функции для рассматриваемых задач *Constrained Clustering* высокой размерности. Разработанный подход проиллюстрирован на примере задачи выявления зон участка горного массива с различной степенью сейсмической активности.

Ключевые слова: программирование в ограничениях, кластерный анализ, мультимножества, кластеризация с частичным привлечением учителя, *smart*-таблицы.

✉ Зуенко Александр Анатольевич
e-mail: zuenko@iimm.ru



Контент доступен под лицензией Creative Commons Attribution 4.0 License.
The content is available under Creative Commons Attribution 4.0 License.

ВВЕДЕНИЕ

Недостатком большинства «классических» методов кластеризации является невозможность учитывать пользовательские ограничения на допустимые/недопустимые комбинации объектов внутри кластера, поскольку не всегда отнесение объекта в ближайший (по расстоянию) кластер является семантически корректной операцией. Для решения этой проблемы в настоящее время развивается направление исследований, именуемое кластеризацией с частичным привлечением учителя (*semi-supervised clustering* или *Constrained Clustering*) [1–5], в рамках которого при отнесении объектов к одному или различным кластерам анализируются не только расстояния между объектами, но и дополнительные фоновые знания о предметной области. Количество классов и сами классы неизвестны, но для некоторых пар объектов известно, например, что они попадают или не попадают в один кластер.

Анализ методов решения задач *Constrained Clustering* показал, что, как и в случае задачи классического кластерного анализа, большинство из них являются приближенными, а не точными. Данные методы не предназначены для систематического исследования пространства поиска и, в общем случае, не позволяют найти глобальный оптимум при рассмотрении задачи *Constrained Clustering* как задачи комбинаторной оптимизации. Появление мощных SAT-солверов и технологии программирования в ограничениях (*Constraint Programming*) [6–8] позволило решать многие практически значимые задачи комбинаторного поиска, характеризующиеся большой размерностью, с помощью точных методов. Однако проведенный анализ показал, что данные средства декларативного программирования всё ещё редко применяются в задачах *Data Mining*, в целом, и в задачах *Constrained Clustering*, в частности. В связи с изложенным, представляется перспективной разработка точных методов решения задач *Constrained Clustering* в различных постановках.

В настоящих исследованиях задачу *Constrained Clustering* предложено решать в рам-

ках парадигмы программирования в ограничениях с использованием нового типа табличных ограничений — *smart-таблиц D-muna* [9]. Помимо разработки способа представления задачи *Constrained Clustering* с использованием *smart-таблиц*, значительное внимание уделено вопросу уменьшения количества и упрощению ограничений задачи в процессе её постановки. Предлагаемые методы в совокупности обеспечивают возможность получения глобального оптимума для задач высокой размерности.

Разработанный подход в статье проиллюстрирован на примере задачи исследования степени влияния горно-геологических факторов на уровень техногенной сейсмической активности участков горного массива.

1. МАТЕРИАЛЫ И МЕТОДЫ

1.1. Представление задачи *Constrained Clustering* в рамках парадигмы *Constraint Programming*

Опишем предлагаемый подход на примере задачи кластеризации с критерием минимизации диаметра разбиения.

Пусть задача состоит в том, что требуется разбить n объектов $O = \{o_1, \dots, o_n\}$ на k кластеров таким образом, чтобы диаметр разбиения был минимальным среди всех возможных разбиений, то есть $D \rightarrow \min$, где D — диаметр разбиения. Напомним, что диаметр разбиения — это максимальный диаметр для всех кластеров разбиения. Диаметр кластера — это максимальное расстояние между любыми двумя точками, принадлежащими данному кластеру. Описываемая модель позволяет искать разбиение при условии, что не задано точное число k итоговых кластеров, а задан лишь интервал $k \in [k_{\min}, k_{\max}]$.

В качестве базовой модели для решения задачи *Constrained Clustering* была использована модель, описанная в [10]. Для обозначения точек (объектов) кластеров используются переменные $G = \{G_1, \dots, G_n\}$. Доменом каждой переменной выступает множество индексов возможных кластеров $\{1, \dots, k_{\max}\}$. Присваивание $G_i = c$ обозначает, что точка o_i лежит в кластере c .

Помимо матрицы расстояний между объектами $[d_{ij}]$, при постановке задачи *Constrained Clustering* задается ряд ограничений. Перечислим упомянутые ограничения [10].

1. *Precede*($[G_1, \dots, G_n], [1..k_{max}]$) — ограничение, которое необходимо задавать для того, чтобы избежать симметричных решений: каждому из возможных разбиений, содержащих по меньшей мере k_{min} различных кластеров и самое большее k_{max} различных кластеров, должно соответствовать ровно одно полное присваивание значений переменных из $G = \{G_1, \dots, G_n\}$.

2. *AtLeast*($G, k_{min}, 1$) — условие на нижнюю границу k_{min} рассматриваемого интервала $[k_{min}, k_{max}]$. Данное ограничение предписывает, чтобы в результирующем полном присваивании хотя бы одна из переменных $G = \{G_1, \dots, G_n\}$ принимала значение k_{min} . Условие на верхнюю границу k_{max} интервала для k учитывается при задании множества возможных индексов кластеров — $\{1, \dots, k_{max}\}$.

3. При решении задачи минимизации диаметра разбиения для каждой пары объектов, то есть для каждого элемента матрицы $[d_{ij}]$, должно быть сгенерировано ограничение вида:

$$(d_{ij} > D) \rightarrow (G_i \neq G_j). \quad (1)$$

Здесь d_{ij} — это константа, обозначающая расстояние между объектами o_i и o_j . Переменная D обозначает диаметр разбиения и изначально принимает значения из интервала $[d_{min}, d_{max}]$, где d_{min} и d_{max} — это минимальный и максимальный элемент матрицы $[d_{ij}]$.

Кроме перечисленных обязательных ограничений, согласно [10], в модель могут быть добавлены дополнительные пользовательские ограничения, например:

1) *AtLeast*(G, G_i, α) — ограничения на минимальный размер любого кластера в разбиении: количество точек в любом кластере при любом разбиении должно быть не меньше заданного числа α ;

2) *AtMost*(G, c, β) — ограничения на максимальный размер кластера с индексом $c \in \{1, \dots, k_{max}\}$: количество точек в любом кластере при любом разбиении должно быть не больше заданного числа β .

3) $G_i = G_j$ — ограничения *must-link*, предписывающее, что при любом разбиении пара объектов o_i и o_j должна попадать в один кластер.

4) $G_i \neq G_j$ — ограничение *cannot-link*, предписывающее, что при любом разбиении пара объектов o_i и o_j не попадает в один кластер.

В базовой модели [10] существенная проблема состоит в организации эффективной обработки нечисловых ограничений вида (1). Дело в том, что в рамках существующих сред программирования в ограничениях обработка нечисловых ограничений осуществляется недостаточно эффективно, особенно в случае задач удовлетворения ограничений высокой размерности.

Таким образом, актуальным направлением исследований представляется разработка способов ускорения обработки подобных ограничений. Другое направление работ, которое активно развивается в настоящих исследованиях, связано с уменьшением количества ограничений, используемых для представления задачи, и упрощением их вида. В ходе исследований было предложено генерировать ограничения вида (1) не для всех пар объектов, а лишь для некоторых, что позволяет существенно снизить размерность решаемой задачи.

Кратко поясним основную идею (рис. 1). Пусть в результате применения некоторых методов нам удалось «сузить» исходный интервал возможных значений диаметра разбиения с $D \in [d_{min}, d_{max}]$ до $D \in [d_1, d_2]$, где d_1 и d_2 — это новые верхняя и нижняя границы интервала. В таком случае для тех элементов матрицы расстояний $[d_{ij}]$, для которых справедливо $d_{ij} \in [d_{min}, d_1]$ ограничений генерировать не требуется, так как утверждение (1) будет тождественно истинным. Для тех элементов матрицы $[d_{ij}]$, для которых выполняется $d_{ij} \in (d_2, d_{max}]$ ограничение вида (1) упрощается и становится ограничением вида $(G_i \neq G_j)$, которое проще обрабатывать. Останется сгенерировать ограничения вида (1) только для объектов, для которых $d_{ij} \in [d_1, d_2]$. Число оставшихся объектов, как правило, значительно меньше, чем количество исходных.

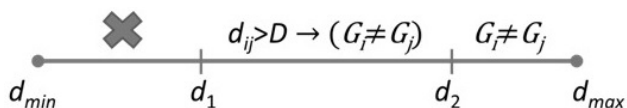


Рис. 1. Возможности упрощения вида ограничений и снижения их количества [Fig. 1. Possibilities of simplifying the form of constraints and reducing their number]

1.2. Представление задачи *Constrained Clustering* с использованием *smart*-таблиц *D*-типа и организация логического вывода

Как уже упоминалось, в рамках современных систем программирования в ограничениях обработка качественных ограничений (логические формулы, продукционные правила, и т. п.) осуществляется недостаточно эффективно. Наиболее перспективным подходом к представлению и обработке качественных зависимостей следует признать подход, основанный на применении их специализированного табличного представления. Известные виды табличных ограничений, такие как *compressed*-таблицы и *smart*-таблицы [11–14], хорошо подходят для моделирования дизъюнктивных нормальных форм логических формул. Однако с помощью известных типов табличных ограничений не всегда целесообразно описывать некоторые виды знаний, например продукционные правила, поскольку при этом порождаются громоздкие конструкции. Рассмотрим пример.

Пример 1. Пусть имеется правило $(4 > D) \rightarrow (G_1 \neq G_2)$, сгенерированное при анализе некоторой матрицы расстояний $[d_{ij}]$. Раскрытие в данном выражении импликации приводит к записи: $(4 \leq D) \vee (G_1 \neq G_2)$. С помощью известных типов *smart*-таблиц это ограничение может быть выражено так:

$$\begin{array}{c} D \quad G_1G_2 \\ \left[\begin{array}{cc} \geq 4 & * \\ * & \neq \end{array} \right]. \end{array}$$

В самой верхней строке записан заголовок отношения (ограничения), причем один из атрибутов отношения, а именно G_1G_2 , является составным, то есть состоит из двух простых атрибутов G_1 и G_2 . Данное табличное представление при двух значащих компонентах содержит две фиктивных компоненты «*», обо-

значающие весь диапазон возможных значений соответствующей переменной. Таким образом, в некоторых случаях использование табличных ограничений известных типов является нерациональным с точки зрения компактного представления информации.

В настоящей работе логические выражения вида (1) предлагается представлять и обрабатывать с помощью специального вида табличных ограничений — *smart*-таблиц *D*-типа, который был ранее разработан одним из авторов статьи [9].

В отличие от широко распространенных *smart*-таблиц *C*-типа, которые соответствуют дизъюнктивным нормальным формам логических формул с элементарными одно- и двуместными предикатами, *smart*-таблицы *D*-типа соответствуют конъюнктивным нормальным формам таких формул. Авторам не известны работы, где в контексте решения задач удовлетворения ограничений рассматривались бы структуры, подобные предлагаемым *smart*-таблицам *D*-типа.

Представленное выше правило может быть смоделировано следующей *smart*-таблицей *D*-типа, состоящей из одной строки:

$$\begin{array}{c} D \quad G_1G_2 \\] \geq 4 \quad \neq [\end{array}$$

Таблицы *D*-типа записываются с помощью перевернутых квадратных скобок.

Приведем еще один пример *smart*-таблицы *D*-типа, которая моделирует сразу несколько подобных правил:

Пример 2. Пусть имеется 5 правил: $(480 > D) \rightarrow (G_4 \neq G_7)$, $(832 > D) \rightarrow (G_5 \neq G_6)$, $(810 > D) \rightarrow (G_7 \neq G_8)$, $(478 > D) \rightarrow (G_7 \neq G_{14})$, $(650 > D) \rightarrow (G_8 \neq G_{10})$.

Тогда соответствующая этим правилам таблица *D*-типа будет выглядеть следующим образом:

$$\begin{array}{c} D \quad G_4G_7 \quad G_5G_6 \quad G_7G_8 \quad G_7G_{14} \quad G_8G_{10} \\ \left[\begin{array}{cccccc} \geq 480 & \neq & \emptyset & \emptyset & \emptyset & \emptyset \\ \geq 832 & \emptyset & \neq & \emptyset & \emptyset & \emptyset \\ \geq 810 & \emptyset & \emptyset & \neq & \emptyset & \emptyset \\ \geq 478 & \emptyset & \emptyset & \emptyset & \neq & \emptyset \\ \geq 650 & \emptyset & \emptyset & \emptyset & \emptyset & \neq \end{array} \right]. \end{array}$$

Символ « \emptyset » обозначает компоненту, не содержащую ни одного значения. Каждая строка данной таблицы соответствует некоторому правилу. При программном представлении *smart*-таблиц *D*-типа компоненты « \emptyset » не хранятся в оперативной памяти компьютера. Тот факт, что несколько правил записываются и обрабатываются в виде единой *smart*-таблицы *D*-типа, позволяет снизить вычислительные расходы на вызовы процедур обработки ограничений и дает возможность удалять больше «лишних» значений из доменов переменных, то есть значений, которые не входят ни в одно решение задачи удовлетворения ограничений.

Отличительной особенностью предлагаемых *smart*-таблиц *D*-типа по сравнению с обычными таблицами и *compressed*-таблицами является наличие составных атрибутов в заголовке матрицы (схеме отношения). Значениями компонент составных атрибутов могут быть отношения из предопределённого множества. В приведённой в примере 2 *smart*-таблице *D*-типа имеется, в частности составной атрибут G_4G_7 , в качестве компонент могут выступать отношения сравнения $\{=, \neq\}$ на паре простых атрибутов G_4 и G_7 .

При организации рассуждений на данных структурах ключевым понятием является понятие кванта информации. Квант — это совокупность значений переменной, рассматриваемая как единое целое. Множество квантов для некоторого домена представляет собой разбиение соответствующего домена или компоненты. Так домен переменной G_4G_7 разбивается на два кванта. Первым квантом является отношение « $G_4 = G_7$ », вторым квантом — отношение « $G_4 \neq G_7$ ». Для простых атрибутов G_i с дискретными областями определения в качестве квантов обычно выступают элементы домена, то есть значения $\{1, 2, 3, \dots\}$.

Любой метод решения задачи удовлетворения ограничений содержит в себе две компоненты: компоненту, реализующую поиск, и компоненту, реализующую распространение ограничений (логический вывод на ограничениях). Как правило, в качестве компоненты, реализующей поиск, выступает один из мето-

дов поиска в глубину с возвратами. Вывод на ограничениях сводится к усечению областей определения переменных и характеризуется полиномиальной временной сложностью. Компонента, реализующая вывод, разрабатывается отдельно под каждый тип ограничений. В случае *smart*-таблиц *D*-типа, помимо редуцирования доменов переменных, в результате вывода на ограничениях удаляются некоторые строки, столбцы, компоненты, значения компонент таблиц. Вывод на ограничениях, представленных в виде *smart*-таблиц *D*-типа, предлагается осуществлять с использованием следующих утверждений [9]:

Утверждение 1. Если хотя бы одна строка *smart*-таблицы *D*-типа пуста (содержит все пустые компоненты), то таблица пуста (соответствующая задача удовлетворения ограничений не имеет решения).

Утверждение 2. Если все компоненты некоторого атрибута пусты, то данный атрибут можно удалить из *smart*-таблицы *D*-типа (удаляются все компоненты, стоящие в соответствующем столбце).

Утверждение 3. Если в *smart*-таблице *D*-типа есть строка (*smart*-кортеж), содержащая лишь одну непустую компоненту, то все кванты, не входящие в эту компоненту, удаляются из соответствующего домена.

Утверждение 4. Если строка *smart*-таблицы *D*-типа содержит хотя бы одну полную компоненту, то она удаляется.

Утверждение 5. Если компонента атрибута *smart*-таблицы *D*-типа содержит квант, не принадлежащий соответствующему домену, то квант удаляется из компоненты.

Утверждение 6. Если в *smart*-таблице *D*-типа усечён один или несколько доменов простых атрибутов, которые формируют некоторый составной атрибут, то: из домена составного атрибута исключаются кванты, которые обращаются в пустое множество при новых доменах соответствующих простых атрибутов.

Утверждение 7. В случае конкретизации домена сложного атрибута должны быть конкретизированы и домены соответствующих простых атрибутов с учётом вновь выведенного домена составного атрибута.

1.3. Предлагаемый подход к повышению эффективности процесса решения задачи *Constrained Clustering*

Рассмотрим предлагаемый подход более детально. В рамках подхода процесс решения разбивается на несколько этапов.

1 этап. Оценить диапазон значений, в который должен попадать искомый оптимальный диаметр разбиения. Для нахождения первоначальной оценки оптимального значения диаметра разбиения предлагается использовать метод FPF (Furthest Point First) [15]. Если считать, что диаметр разбиения, полученного в результате применения метода FPF, равен d , тогда оптимальный диаметр разбиения может быть оценен как: $D \in [(d/2), d]$. Видно, что чем точнее оценена нижняя граница, тем менее точной оказывается верхняя, и наоборот. Как правило, этот метод позволяет получить хорошую оценку для нижней границы оцениваемого параметра. На основе полученной оценки генерируются ограничения *cannot-link* для тех пар кластеров, для которых $d_{ij} > d$.

2 этап. Выполнить конкретизацию верхней границы найденного на этапе 1 интервала $D \in [(d/2), d]$. Для этого применяется разработанный авторами метод агломеративной иерархической кластеризации с учетом пользовательских ограничений [16]. При анализе возможности/необходимости слияния кластеров в рамках авторского метода принимаются во внимание пользовательские ограничения *cannot-link* и *must-link*. Применение данного метода повышает эффективность вычислительных процедур и позволяет сократить перебор вариантов объединения кластеров. В результате данного этапа получаем некоторое разбиение объектов на k_{min} кластеров. Для данного разбиения вычисляем диаметр, который служит верхней оценкой для возможных значений переменной D .

3 этап. Сгенерировать ограничения в виде *smart*-таблиц D -типа для систематического решения задачи *Constrained Clustering*. Оценка значений критерия кластеризации, полученная на предыдущих этапах, позволяет генери-

ровать ограничения не для всех пар кластеризуемых объектов, как было описано ранее.

4 этап. Решить сгенерированную на предыдущем этапе задачу *Constrained Clustering*. В ходе систематического поиска не происходит полного перебора вариантов, поскольку неперспективные ветви дерева поиска отсекаются в результате применения метода распространения ограничений для случая *smart*-таблиц D -типа и предложенных эвристики выбора наилучшего преемника текущего состояния. При анализе наилучшего преемника выбирается переменная, домен которой содержит наименьшее количество значений. Выбор значения переменной производится согласно следующей эвристике: поскольку переменная представляет один из кластеризуемых объектов, а её значение — номер кластера, то присваиваем переменной номер того кластера, который ближе к рассматриваемому объекту.

При применении подхода отсутствуют принципиальные ограничения на способ представления кластеризуемых объектов. Это может быть обычная объектно-признаковая таблица. Также для представления объектов могут использоваться мультимножества. В последнем случае в качестве кластеризуемого объекта может выступать агрегированное множество объектов предметной области, а сама кластеризация ориентирована на групповое принятие решений с учетом мнений нескольких экспертов [17].

В рамках подхода осуществляется поиск всех решений задачи *Constrained Clustering*, то есть всех разбиений объектов, обеспечивающих оптимальное значение критерия кластеризации. Эти решения в дальнейшем могут использоваться для того, чтобы понять, какие из объектов образуют ядро кластеров, а какие при различных разбиениях попадают в различные кластеры.

2. РЕЗУЛЬТАТЫ И ИХ ОБСУЖДЕНИЕ

Рассмотрим пример, где в качестве объектов кластеризации выступают 14 пространственных ячеек, на которые разбит один из

участков высоконапряженного массива горных пород. Подробное описание объекта исследований приведено в работе [16], где авторами был предложен метод иерархической кластеризации с учетом пользовательских ограничений. В отличие от упомянутой работы, где предложен «жадный» метод поиска, в ходе настоящих исследований разработан метод систематического поиска всех разбиений, доставляющих глобальный оптимум задачи *Constrained Clustering*, а также используется другой способ представления кластеризуемых объектов и отличная метрика.

Напомним, что в рассматриваемом примере цель кластеризации — выявление зон с различной степенью сейсмической активности. Каждое сейсмическое событие, отнесенное к некоторой пространственной ячейке, описывается определенным набором факторов, оказывающих, по мнению экспертов, влияние на возникновение сейсмических событий. В качестве таких факторов выступают геология и тектоника месторождения и прилегающего к нему района, геометрия и динамика горных работ. При различных сочетаниях влияющих факторов может наблюдаться различный уровень сейсмической активности.

Исходные данные, агрегирующие оценки двух экспертов, представлены в виде таблицы с описанием совокупности влияющих факторов с учетом числа сейсмособытий и с представлением групп сейсмособытий в виде совокупности мультимножеств.

В качестве признаков (факторов) использовались: *R1* — разлом 1; *R2* — разлом 2; *OP* — границы очистного пространства; *OPv* — границы очистного пространства вышележащего горизонта; *W* — выработки;

RT — рудное тело; *VP* — вмещающие породы; *RT/VP* — рудное тело/вмещающие породы; *WB* — висячий бок рудной залежи; *LB* — лежащий бок рудной залежи, *N* — число сейсмособытий в ячейке.

В табл. 1 представлен фрагмент списка исходных данных для проведения кластеризации. Каждая строка табл. 1 представляет собой мультимножество. В каждой ячейке таблицы, соответствующей некоторому фактору, приведены значения кратности для двух элементов мультимножества: 0 — данный фактор не оказывает влияния и 1 — данный фактор оказывает влияние. Подобная таблица может отображать мнение как одного эксперта, так и представлять совокупность мнений нескольких экспертов в случае группового принятия решений. В табл. 2 представлены результаты расчетов расстояний между исходными кластерами (один объект — один кластер).

В расчетах использовалась метрика:

$$d_{11}(o_i, o_j) = \sum_{l=1}^n |k_{A_i}(x^l) - k_{A_j}(x^l)|, \quad (2)$$

где A_i и A_j — мультимножества, соответствующие объектам o_i и o_j .

В приводимом примере требуется выполнить кластеризацию пространственных ячеек на три кластера таким образом, чтобы диаметр разбиения был бы минимально возможным. Каждый кластер соответствует некоторому уровню сейсмической активности: кластер *A* — низкий уровень активности, кластер *B* — средний уровень активности, кластер *C* — высокий уровень активности.

Рассмотрим по этапам применение предложенного подхода на описанном примере.

Первый этап состоит в применении метода FPF для предварительной оценки диаметра

Таблица 1. Представление информации о кластеризуемых объектах
[Table 1. Representing information about clustered objects]

| Номер ячейки | R1 | R2 | OP | OPv | RT | VP | RT/VP | W | WB | LB | N |
|--------------|-----|-----|-----|-----|-----|-----|-------|-----|-----|-----|---|
| 1 | 2 0 | 2 0 | 2 0 | 2 0 | 2 0 | 0 2 | 2 0 | 2 0 | 0 2 | 2 0 | 2 |
| 2 | 0 2 | 2 0 | 2 0 | 2 0 | 2 0 | 0 2 | 2 0 | 2 0 | 0 2 | 2 0 | 2 |
| 3 | 6 0 | 6 0 | 6 0 | 6 0 | 6 0 | 0 6 | 6 0 | 6 0 | 0 6 | 6 0 | 6 |
| ... | | | | | | | | | | | |

Таблица 2. Матрица расстояний
[Table 2. Destination matrix]

| | | | | | | | | | | | | | | |
|----|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 1 | 0 | 4 | 40 | 84 | 1062 | 1264 | 446 | 1088 | 242 | 442 | 104 | 328 | 10 | 66 |
| 2 | 4 | 0 | 44 | 80 | 1066 | 1262 | 450 | 1084 | 246 | 438 | 108 | 324 | 10 | 62 |
| 3 | 40 | 44 | 0 | 52 | 1046 | 1272 | 438 | 1104 | 250 | 454 | 112 | 344 | 60 | 58 |
| 4 | 84 | 80 | 52 | 0 | 1050 | 1260 | 480 | 1100 | 270 | 450 | 140 | 340 | 100 | 72 |
| 5 | 1062 | 1066 | 1046 | 1050 | 0 | 832 | 980 | 1294 | 1020 | 1156 | 1030 | 1006 | 1070 | 1098 |
| 6 | 1264 | 1262 | 1272 | 1260 | 832 | 0 | 1260 | 936 | 1260 | 1088 | 1250 | 1132 | 1260 | 1288 |
| 7 | 446 | 450 | 438 | 480 | 980 | 1260 | 0 | 810 | 200 | 104 | 410 | 386 | 450 | 478 |
| 8 | 1088 | 1084 | 1104 | 1100 | 1294 | 936 | 810 | 0 | 930 | 650 | 1080 | 1016 | 1080 | 1108 |
| 9 | 242 | 246 | 250 | 270 | 1020 | 1260 | 200 | 930 | 0 | 280 | 210 | 270 | 250 | 278 |
| 10 | 442 | 438 | 454 | 450 | 1156 | 1088 | 104 | 650 | 280 | 0 | 430 | 366 | 430 | 458 |
| 11 | 104 | 108 | 112 | 140 | 1030 | 1250 | 410 | 1080 | 210 | 430 | 0 | 240 | 100 | 114 |
| 12 | 328 | 324 | 344 | 340 | 1006 | 1132 | 386 | 1016 | 270 | 366 | 240 | 0 | 320 | 320 |
| 13 | 10 | 10 | 60 | 100 | 1070 | 1260 | 450 | 1080 | 250 | 430 | 100 | 320 | 0 | 70 |
| 14 | 66 | 62 | 58 | 72 | 1098 | 1288 | 478 | 1108 | 278 | 458 | 114 | 320 | 70 | 0 |

разбиения. После применения метода FPF имеем: в кластер *A* попадают объекты 1–4, 11–14; в кластер *B* — объекты 7–10; в кластер *C* — объекты 5, 6. Диаметр кластера кластера *A* — 344, диаметр кластера *B* — 930, диаметр кластера *C* — 832. Тогда диаметр разбиения — 930. Оценка для возможных значений диаметра разбиения будет выглядеть следующим образом: $D \in [465, 930]$.

Далее в рамках второго этапа предлагаемого подхода проводится кластеризация сейсмособытий согласно модифицированному методу иерархической кластеризации [16] с учетом полученных с помощью метода FPF ограничений *cannot link*: ($d_{ij} > 930$). В табл. 2 темным цветом помечены комбинации объектов, на которые наложены ограничения *cannot link*. На каждом шаге иерархической кластеризации расстояния между объединенными кластерами и остальными пересчитываются, при этом на объединенные кластеры распространяются ограничения, наложенные на исходные кластеры. При расчете расстояний в обработку берутся только допустимые сочетания (не помеченные цветом в таблице расстояний), что позволяет на каждом шаге существенно сократить объем вычислений. В табл. 3 приведен последний шаг кластеризации (не осталось ячеек, которые можно объединить).

Таблица 3. Разбиение, полученное после иерархической кластеризации
[Table 3. Partition obtained after hierarchical clustering]

| | | | |
|---------------|---------------|----------|------|
| | 1–4, 9, 11–14 | 7, 8, 10 | 5, 6 |
| 1–4, 9, 11–14 | 0 | 1718 | 1912 |
| 7, 8, 10 | 1718 | 0 | 1786 |
| 5, 6 | 1912 | 1786 | 0 |

На рис. 2 в виде дендрограммы представлен ход авторского метода иерархической кластеризации с учетом ограничений на сочетаемость кластеризуемых объектов.

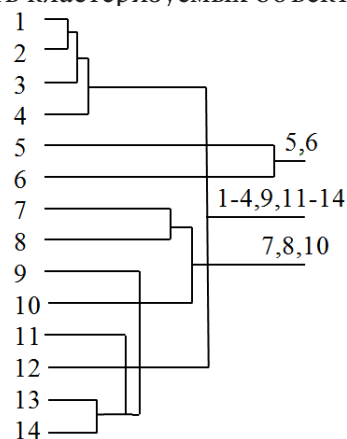


Рис. 2. Результаты иерархической кластеризации
[Fig. 2. Hierarchical clustering results]

Результаты иерархической кластеризации (результаты второго этапа): кластер A — объекты 1–4, 9, 11–14; кластер B — объекты 7, 8, 10; кластер C — объекты 5, 6. Диаметры полученных кластеров, 344, 810 и 832 соответственно. Таким образом, уточнена верхняя граница кластеризации: метод FPF давал оценку диаметра разбиения $D \in [465, 930]$, а после уточнения интервал для оценки диаметра разбиения стал $D \in [465, 832]$.

В рамках третьего этапа подхода осуществляется генерация ограничений в форме smart-таблиц D -типа. Выполненная на предыдущих этапах оценка диаметра разбиения позволяет генерировать ограничения далеко не для всех ячеек матрицы расстояний (комбинаций пар объектов). Очевидно, что для элементов матрицы расстояний, для которых $d_{ij} < 465$, нет необходимости генерировать ограничения. Для элементов матрицы расстояний, для которых $d_{ij} > 832$ генерируются ограничения вида $(G_i \neq G_j)$. Элементов же матрицы расстояний, которые удовлетворяют условию $d_{ij} \in [465, 832]$, остается всего пять (если смотреть на верхний треугольник матрицы расстояний в табл. 2): $d_{4,7} = 480$, $d_{5,6} = 832$, $d_{7,8} = 810$, $d_{7,14} = 478$, $d_{8,10} = 650$. Следовательно, для рассматриваемой задачи *Constrained Clustering* будет сгенерировано всего пять ограничений вида (1). Выше в примере 2 приводится описание этих пяти правил в форме smart-таблицы D -типа. Заметим, что без подготовительных этапов, состоящих в применении метода FPF и иерархической кластеризации с учетом ограничений, количество генерируемых ограничений вида (1) оценивалось бы величиной $(14 \cdot 13) / 2 = 91$.

В рамках четвертого этапа производится систематический поиск решений для сгенерированной на третьем этапе задачи удовлетворения ограничений. В результате систематического поиска было установлено, что оптимальным значением критерия кластеризации является $D = 832$, а также получены все три оптимальных решения рассматриваемой задачи *Constrained Clustering*.

Первое решение: в класс A попадают объекты 1, 2, 3, 4, 9, 10, 11, 12, 13, 14; в класс B — объекты 7, 8; в класс C — объекты 5, 6.

Второе решение: в класс A попадают объекты 1; 2; 3; 4; 7; 9; 11; 12; 13; 14; в класс B — объекты 8, 10; в класс C — объекты 5, 6.

Третье решение: в класс A попадают объекты 1; 2; 3; 4; 9; 11; 12; 13; 14; в класс B — объекты 7, 8, 10; в класс C — объекты 5, 6.

На основе полученных решений можно сделать вывод, что *при любом* разбиении в класс A попадают объекты 1, 2, 3, 4, 9, 11, 12, 13, 14; в класс B — объект 8; в класс C — объекты 5, 6. Объекты 7 и 10 иногда попадают в класс A (в одном из трех оптимальных решений), а иногда в класс B (в двух из трех оптимальных решений). Принадлежность этих объектов кластерам наглядно отображена в табл. 4, где также показано наличие упомянутых ранее влияющих факторов в той и иной ячейке. В столбце N приводится количество сейсмических событий, произошедших в той или иной ячейке.

Далее поставим цель выявить влияние перечисленных в табл. 4 факторов на уровень сейсмической активности. В качестве инструмента были использованы «деревья решений» (вкладка «Data Mining») в среде «Statistica 13» [18]. Обучающая выборка была сформирована из всех объектов, кроме объектов 7 и 10. В итоге, получено дерево решений, представленное на рис. 3 (цифра 1 соответствует классу A , цифра 2 — классу B , цифра 3 — классу C). На основе полученного дерева можно сформулировать правила, описывающие влияние рассмотренных факторов среды на уровень сейсмической активности.

Первое правило: Если ячейка относится к висячему боку и там имеются выработки, то уровень сейсмической активности оценивается как высокий.

Второе правило: Если ячейка относится к висячему боку и там отсутствуют выработки, то уровень сейсмической активности оценивается как низкий.

Третье правило: Если ячейка не относится к висячему боку и в ней располагается граница очистного пространства текущего горизонта, то уровень сейсмической активности оценивается как средний.

Четвертое правило: Если ячейка не относится к висячему боку и в ней не располагает-

Таблица 4. Разбиение пространственных ячеек по уровням сейсмической активности на основе всех полученных решений задачи *Constrained Clustering*
 [Table 4. Partitioning spatial cells by seismic activity levels, based on all solutions obtained to *Constrained Clustering* problem]

| Ячейка | R1 | R2 | OP | OPv | RT | VP | RT/VP | W | WB | LB | N | Кластер |
|--------|----|----|----|-----|----|----|-------|---|----|----|-----|---------|
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 2 | A |
| 2 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 2 | A |
| 3 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 6 | A |
| 4 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 10 | A |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 107 | C |
| 6 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 126 | C |
| 7 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 45 | A, B |
| 8 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 108 | B |
| 9 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 25 | A |
| 10 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 43 | A, B |
| 11 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 10 | A |
| 12 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 32 | A |
| 13 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | A |
| 14 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 7 | A |

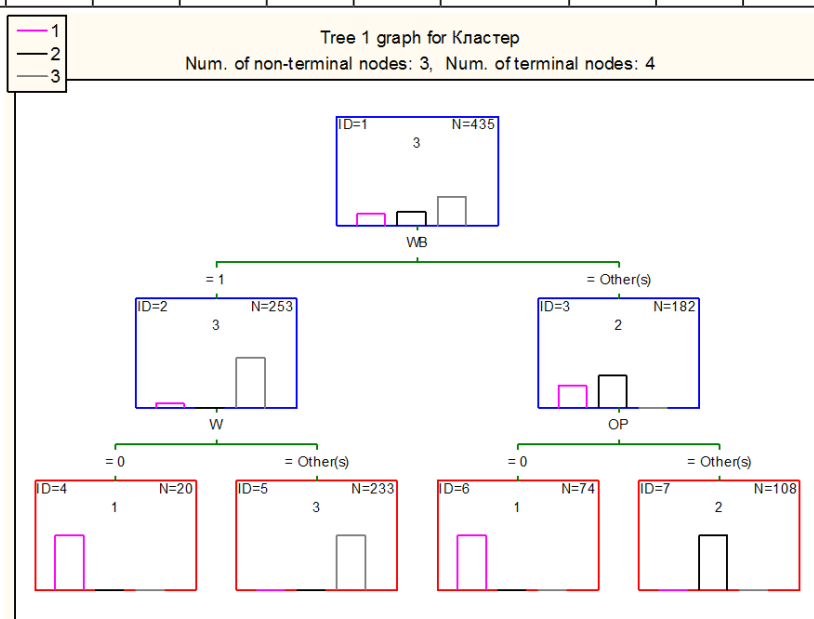


Рис. 3. Получение правил на основе деревьев решений
 [Fig. 3. Generating rules based on Decision Trees]

ся граница очистного пространства текущего горизонта, то уровень сейсмической активности оценивается как низкий.

ЗАКЛЮЧЕНИЕ

В статье предложен комплексный подход к решению задач *Constrained Clustering*, реа-

лизованый в рамках парадигмы *Constraint Programming*, то есть задача *Constrained Clustering* формализуется в виде задачи удовлетворения ограничений, а итоговые кластеры формируются в результате систематического исследования альтернатив разбиения. При этом существенная сложность заключается в организации эффективной обработки ка-

качественных ограничений, представляющих собой правила отнесения объектов к одному или разным кластерам. Предлагается представлять и обрабатывать качественные ограничения в форме табличных ограничений нового типа, а именно *smart*-таблиц *D*-типа. Также значительное внимание в статье уделено вопросу снижения количества и упрощению ограничений задачи *Constrained Clustering*. Предлагается генерировать ограничения не для всех пар объектов, а лишь для некоторых, основываясь на интервальной оценке для оптимального значения критерия кластеризации. Для получения данной оценки используется ранее предложенный авторами метод иерархической кластеризации, который позволяет анализировать ограничения на допустимость/недопустимость комбинаций объектов внутри кластера. Предложенный подход позволяет находить все варианты разбиений, обеспечивающие глобальный оптимум целевой функции для рассматриваемых задач *Constrained Clustering* высокой размерности. Применение предлагаемого подхода позволяет существенно снизить размерность исходной задачи и трудоемкость вычислений.

Проведенные исследования подтвердили применимость методов *Constrained Clustering* для изучения техногенной сейсмичности и показали принципиальную возможность оценки с их помощью степени влияния горно-геологических факторов на сейсмические события. Полученные результаты кластеризации были использованы для предварительного выделения зон с различной сейсмоактивностью. Результаты исследований позволяют судить о перспективности дальнейшего развития методов кластеризации для задач поддержки управления природно-промышленными комплексами как сложными объектами в многомерных фазовых пространствах.

БЛАГОДАРНОСТИ

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 20-07-00708а.

КОНФЛИКТ ИНТЕРЕСОВ

Авторы декларируют отсутствие явных и потенциальных конфликтов интересов, связанных с публикацией настоящей статьи.

СПИСОК ЛИТЕРАТУРЫ

1. *Guns, T.* k-Pattern set mining under constraints / T. Guns, S. Nijssen, L. D. Raedt // IEEE Transactions on Knowledge and Data Engineering, – 2013. – V. 25, № 2. – P. 402–418.
2. *Métivier, J.-P.* Constrained Clustering Using SAT / J.-P. Métivier, et al. // Proceedings of the 11th International Symposium on Advances in Intelligent Data Analysis, – 2012. – P. 207–218.
3. *Davidson, I.* A SAT-based Framework for Efficient Constrained Clustering / I. Davidson, S. S. Ravi, L. Shamis // Proceedings of the 10th SIAM International Conference on Data Mining, – 2010. – P. 94–105.
4. *Mueller, M.* Integer Linear Programming Models for Constrained Clustering / M. Mueller, S. Kramer // Proceedings of the 13th International Conference on Discovery Science, – 2010. – P. 159–173.
5. *Babaki, B.* Constrained Clustering using Column Generation / B. Babaki, T. Guns, S. Nijssen // Proceedings of the 11th International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, – 2014. – P. 438–454.
6. *Bartak, R.* Constraint Programming: In Pursuit of the Holy Grail / R. Bartak // Proceedings of the Week of Doctoral Students (WDS99), Part IV. – Prague: MatFyzPress, – 1999. – P. 555–564.
7. *Ruttkey, Zs.* Constraint satisfaction a survey / Zs. Ruttkey // CWI Quarterly, – 1998. – Vol. 11. – P. 163–214.
8. *Russel, S.* Artificial Intelligence: A Modern Approach. 3rd edition / S. Russel, P. Norvig. – Prentice Hall, – 2010. – 1132 p.
9. *Zuenko, A. A.* Representation and Processing of Qualitative Constraints Using a New Type of Smart Tables / A.A. Zuenko // Proceedings of the 4th International Conference on Computer Science and Application Engineering

(CSAE '20). ACM New York, NY, USA. – 2020. Article № 45. – P. 1–7.

10. *Dao, T.-B.-H.* Constrained clustering by constraint programming / T.-B.-H. Dao, K.-C. Duong, C. Vrain // *Artificial Intelligence*, – 2017. – V. 244. – P. 70–94.

11. *Xia, W.* Optimizing STR algorithms with tuple compression / W. Xia, R.H.C. Yap // *CP 2013. LNCS*, 8124. – 2013. – P. 724–732.

12. *Perez, G.* Improving GAC-4 for table and MDD constraints / G. Perez, J.C. Regin // *CP 2014. LNCS*, 8656. – 2014. – P. 606–621.

13. *Katsirelos, G.* A compression algorithm for large arity extensional constraints / G. Katsirelos, T. Walsh // *CP 2007. LNCS*, 4741, – 2007. – P. 379–393.

14. *Mairy, J.* The Smart Table Constraint / J. Mairy, Y. Deville, C. Lecoutre // In: Michel L. (eds.) *Integration of AI and OR Techniques in Constraint Programming. CPAIOR 2015. Lecture Notes in Computer Science*. – Springer. Cham, – 2015. – V. 9075. – P. 271–287.

15. *Gonzalez, T.* Clustering to Minimize the Maximum Intercluster Distance / T. Gonzalez // *Theoretical Computer Science*, – 1985. – V. 38. – P. 293–306.

16. *Зуенко, А. А.* Применение методов ограниченной кластеризации для исследования техногенной сейсмичности / А. А. Зуенко, О. В. Фридман, О. Г. Журавлева // *Вестник ВГУ, серия: Системный анализ и информационные технологии*, – 2019. – № 3. – С. 29–41.

17. *Петровский, А. Б.* Методы групповой классификации многопризнаковых объектов (часть 1) / А. Б. Петровский // *Искусственный интеллект и принятие решений*. – 2009. – № 3. – С. 3–14.

18. *Боровиков, В. П.* Популярное введение в современный анализ данных и машинное обучение на STATISTICA / В. П. Боровиков. – Москва : Горячая Линия. Телеком, 2021. – 354 с.

Зуенко Александр Анатольевич — канд. техн. наук, ведущий научный сотрудник, Институт информатики и математического моделирования — обособленное подразделение Федерального государственного бюджетного учреждения Федерального исследовательского центра «Кольский научный центр Российской академии наук» (ИИММ КНЦ РАН).

E-mail: zuenko@iimm.ru

ORCID iD: <https://orcid.org/0000-0002-7165-6651>

Фридман Ольга Владимировна — канд. техн. наук, старший научный сотрудник, Институт информатики и математического моделирования — обособленное подразделение Федерального государственного бюджетного учреждения Федерального исследовательского центра «Кольский научный центр Российской академии наук» (ИИММ КНЦ РАН).

E-mail: ofridman@iimm.ru

ORCID iD: <https://orcid.org/0000-0003-1897-4922>

Зуенко Ольга Николаевна — стажер-исследователь, аспирант ИИММ КНЦ РАН, Институт информатики и математического моделирования — обособленное подразделение Федерального государственного бюджетного учреждения Федерального исследовательского центра «Кольский научный центр Российской академии наук» (ИИММ КНЦ РАН)

E-mail: ozuenko@iimm.ru

ORCID iD: <https://orcid.org/0000-0001-5431-7538>

APPLICATION OF SMART-TABLES AND ESTIMATES OF OBJECTIVE FUNCTIONS TO REDUCE THE DIMENSION AND ACCELERATE THE SOLUTION OF THE CONSTRAINED CLUSTERING PROBLEMS

© 2021 A. A. Zuenko✉, O. V. Fridman, O. N. Zuenko

*Institute of Informatics and Mathematical Modeling — Subdivision of the Federal Research Centre «Kola Science Centre of the Russian Academy of Sciences» (IIMM KSC RAS)
24A, Fersman Street, 184209 Apatity, Murmansk region, Russian Federation*

Annotation. The article presents an integrated approach to the accurate solution of *Constrained Clustering* problems, that is clustering problems involving analysis, in addition to distance matrix, of background knowledge about necessity/infeasibility of the occurrence of some objects into a particular cluster. The approach is implemented in the framework of Constraint Programming Paradigm, focused on the constructions of systematic search procedures (traversal of the search tree) for solving complex combinatorial problems. Moreover, all the initial information about the problem is expressed with constraints, that is, qualitative and quantitative relations. A significant complexity lies in the fact that in today's environments and libraries of constraint programming the processing of qualitative constraints, which in particular, are the rules for assigning objects to one or different clusters, is not done efficiently enough. Thus, it seems relevant to develop ways to accelerate the processing of such constraints. The article proposes to represent and process the qualitative constraints in the form of table constraints of a new type, namely *smart-tables* of *D*-type. For the *smart-tables* of *D*-type high efficient constraint inference procedures have been developed that perform early pruning unpromising branches of the search tree. Another area of work, which is actively developing in this research, is related with reducing the number of constraints used to represent the problem and simplifying their form. It is proposed to generate the constraints only for some pairs of objects based on an interval estimate for the optimal value of the clustering criterion. To obtain this estimate, the hierarchical clustering method previously proposed by the authors is used, which allows you to analyze the constraints for the combinations of pairs of objects within a cluster. The proposed approach allows you to find all the partitions variants that provide global optimum of objective function for the considered *Constrained Clustering* problems of high dimension. The developed approach is illustrated by example of the problem of identifying zones of rock mass with different seismic activity.

Keywords: constraint programming, cluster analysis, multi-sets, semi-supervised clustering, *smart-tables*.

ACKNOWLEDGMENTS

The reported study was funded by RFBR, project number № 20-07-00708a.

CONFLICT OF INTEREST

The authors declare the absence of obvious and potential conflicts of interest related to the publication of this article.

✉ Zuenko Alexander A.
e-mail: zuenko@iimm.ru

REFERENCE

1. Babaki B., Guns T., Nijssen S. (2014) Constrained Clustering using Column Generation: Proceedings of the 11th International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. P. 438–454.
2. Bartak R. (1999) Constraint Programming: In Pursuit of the Holy Grail, Proceedings of the Week of Doctoral Students (WDS99), Part IV. Prague: MatFyzPress. P. 555–564.

3. Borovikov V. P. (2017) Popular introduction to modern data analysis and machine learning on STATISTICA. Moscow: Hot Line. Telecom.
4. Dao T.-B.-H., Duong K.-C., Vrain C. Constrained clustering by constraint programming. *Artificial Intelligence*. 244. P. 70–94.
5. Davidson I. Ravi S. S., Shamis L. (2010) A SAT-based Framework for Efficient Constrained Clustering: Proceedings of the 10th SIAM International Conference on Data Mining. P. 94–105.
6. Gonzalez T. (1985) Clustering to Minimize the Maximum Intercluster Distance. *Theoretical Computer Science*. 38. P. 293–306.
7. Guns T., Nijssen S., Raedt L. D. (2013) k-Pattern set mining under constraints. *IEEE Transactions on Knowledge and Data Engineering*. 25(2). P. 402–418.
8. Katsirelos G., Walsh T. (2007) A compression algorithm for large arity extensional constraints. *CP 2007. LNCS*, 4741. P. 379–393.
9. Mairy J., Deville Y., Lecoutre C. (2015) The Smart Table Constraint. In: Michel, L. (eds.) *Integration of AI and OR Techniques in Constraint Programming. CPAIOR 2015. Lecture Notes in Computer Science*. Springer. Cham. 9075. P. 271–287.
10. Métivier J.-P. (2012) Constrained Clustering Using SAT: Proceedings of the 11th International Symposium on Advances in Intelligent Data Analysis. P. 207–218.
11. Mueller M., Kramer S. (2010) Integer Linear Programming Models for Constrained: Proceedings of the 13th International Conference on Discovery Science. P. 159–173.
12. Perez G. (2014) Improving GAC-4 for table and MDD constraints. *CP 2014. LNCS*. 8656. P. 606–621.
13. Petrovsky A. B. (2009) Methods for group classification of multi-feature objects (part 1). *Artificial intelligence and decision making*. 3. P. 3–14. (in Russian).
14. Russel S. & Norvig P. (2010) *Artificial Intelligence: A Modern Approach*. 3rd edition. Prentice Hall.
15. Ruttkay Zs. (1998) Constraint satisfaction a survey. *CWI Quarterly*. 11. P. 163–214.
16. Xia W. (2013) Optimizing STR algorithms with tuple compression. *CP 2013. LNCS*. 8124. P. 724–732.
17. Zuenko A. A. (2020) Representation and Processing of Qualitative Constraints Using a New Type of Smart Tables: Proceedings of the 4th International Conference on Computer Science and Application Engineering (CSAE '20). ACM New York, NY, USA, Article № 45. P. 1–7.
18. Zuenko A. A., Fridman O. V., Zhuravleva O. G. (2019) Application of limited clustering methods for the study of technogenic seismicity. *Vestnik VSU, series: System analysis and information technology*. 3. P. 29–41. (in Russian).

Zuenko Alexander A. — PhD, leading researcher, Institute of Informatics and Mathematical Modeling – Subdivision of the Federal Research Centre “Kola Science Centre of the Russian Academy of Sciences” (IIMM KSC RAS).

E-mail: zuenko@iimm.ru

ORCID iD: <https://orcid.org/0000-0002-7165-6651>

Fridman Olga V. — PhD, senior researcher, Institute of Informatics and Mathematical Modeling – Subdivision of the Federal Research Centre “Kola Science Centre of the Russian Academy of Sciences” (IIMM KSC RAS).

E-mail: ofridman@iimm.ru

ORCID iD: <https://orcid.org/0000-0003-1897-4922>

Zuenko Olga N. – trainee researcher, PhD student, Institute of Informatics and Mathematical Modeling – Subdivision of the Federal Research Centre “Kola Science Centre of the Russian Academy of Sciences” (IIMM KSC RAS).

E-mail: ozuenko@iimm.ru

ORCID iD: <https://orcid.org/0000-0001-5431-7538>