ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Октябрь–декабрь 2021. Т. 28, № 4. С. 5–60

УДК 519.87 DOI 10.33048/daio.2021.28.714

# ДИСКРЕТНЫЕ ЗАДАЧИ РАЗМЕЩЕНИЯ В МАШИННОМ ОБУЧЕНИИ

*И. Л. Васильев*[a], *А. В. Ушаков*[b]

Институт динамики систем и теории управления им. В. М. Матросова,
ул. Лермонтова, 134, 664033 Иркутск, Россия

E-mail: [a] vil@icc.ru, [b] aushakov@icc.ru

**Аннотация.** Задачи размещения составляют довольно широкий класс оптимизационных задач, являющихся одними из базовых объектов исследования в области комбинаторной оптимизации и исследования операций. Помимо большого числа экономических приложений, такие задачи нашли широкое применение и в области машинного обучения. Например, задача кластеризации может быть представлена как задача размещения, в которой необходимо разделить множество клиентов на кластеры, обслуживаемые открытыми предприятиями. В настоящем обзоре планируется кратко проследить, как идеи и подходы, возникшие в области теории размещения, привели к появлению современных популярных алгоритмов машинного обучения, реализованных в большинстве коммерческих пакетов прикладных программ и технических вычислений. Помимо этого, предполагается провести обзор современных точных методов и эвристик, а также некоторых обобщений базовых задач и алгоритмов, возникающих непосредственно в практических приложениях из анализа данных. Отметим, что основное внимание будет уделено дискретным задачам размещения, лежащим, например, в основе многих популярных алгоритмов кластеризации (PAM, affinity propagation и т. д.) Поскольку объём современных данных создаёт существенные трудности для классических алгоритмов ввиду их высокой вычислительной сложности, в обзоре будут рассмотрены современные подходы к реализации упомянутых алгоритмов для анализа больших массивов данных. Библиогр. 138.

**Ключевые слова:** машинное обучение, задачи размещения, кластеризация.

Исследование выполнено при финансовой поддержке Российского фонда фундаментальных исследований (проект № 20–17–50233).

## Введение

Задачи размещения представляют собой обширный класс оптимизационных задач, вызывающих широкий интерес со стороны многочисленных исследователей в целом ряде областей, таких как математическое программирование, исследования операций, компьютерные науки и т. д. Несмотря на разнообразие, задачи размещения объединяет необходимость выработки оптимальной стратегии принятия двух основных решений: выбора мест для размещения в них предприятий с целью обслуживания некоторого заданного множества потребителей, а также назначение для каждого потребителя предприятия для получения им обслуживания. Определение оптимальной стратегии зависит от конкретной задачи и заданных ограничений. Она может предполагать, например, минимизацию суммарных издержек обслуживания, суммарной стоимости размещения предприятий, расстояния между потребителями и открытыми предприятиями и т. д. Под предприятиями могут пониматься не только непосредственно производственные мощности, но и разнообразные объекты инфраструктуры, оборудование (камеры, датчики) и т. д. К настоящему моменту задачи размещения нашли приложение как в государственном (например, размещение пожарных станций и станций скорой помощи, почтовых отделений, госпиталей, образовательных учреждений и т. д.), так и частном (размещение банковских отделений, логистических парков, складов, центров дистрибуции) секторах экономики. Отметим, что в широком смысле задачи размещения могут быть разделены на три типа — непрерывные, дискретные и задачи на графах — в зависимости от того, каким образом определяются возможные места для размещения предприятий. В зависимости от структуры ограничений и целевой функции задачи каждого класса также могут быть условно разделены на три группы: минисуммные, минимаксные и задачи покрытия.

Помимо классических, так называемых экономических и географических приложений, задачи размещения нашли широкое применение и в машинном обучении. Если говорить о приложениях задач размещения в реальных практических задачах, то их применению в прикладных задачах машинного обучения посвящено намного больше работ, чем приложениям, непосредственно связанным с экономическим размещением предприятий. В рамках машинного обучения традиционным приложением задач размещения является область методов обучения без учителя, прежде всего кластеризация, являющаяся одной из базовых техник для первичного анализа данных, а также составной частью более комплексных подходов. Классическая задача кластеризации состоит в разделении заданного множества объектов (элементов данных) на непересекающиеся группы (кластеры) таким образом, чтобы каждый кластер состоял

из максимально схожих между собой объектов, в то время как объекты из разных кластеров были различными. Несмотря на большое количество подходов к задаче кластеризации, наиболее популярный тип алгоритмов направлен на поиск решения некоторой невыпуклой задачи оптимизации, целью которой является минимизация расстояний («максимизация схожести») между объектами кластера и его представителем. Таким образом, задача кластеризации может быть представлена как задача размещения, в которой необходимо разбить заданное множество потребителей на непересекающиеся группы, каждая из которых получает обслуживание из некоторого открытого предприятия.

Считается, что основа современных исследований в области задач размещения была заложена с выходом известных работ Л. Купера [1, 2], в которых одними из первых было рассмотрено обобщение так называемой задачи Вебера на случай нескольких предприятий, известное в настоящее время как задача Вебера с несколькими источниками (multi-source Weber problem) или как непрерывная задача о $p$-медиане. Предполагается, что задано множество точек на плоскости (потребителей) $a^j \in \mathbb{R}^2$, $j \in J$, $|J| = n$, для которых задан спрос $w_{ij}$, вообще говоря, зависящий от мест размещения предприятий. Задача состоит в поиске точек $c^i \in \mathbb{R}^2$, $i \in I$, $|I| = p$, для размещения в них предприятий таким образом, чтобы минимизировать взвешенную сумму евклидовых расстояний между потребителями и обслуживающими их предприятиями, т. е.

$$\min_{C \subset \mathbb{R}^2} \Big\{ \sum_{j \in J} \min_{i \in I} w_{ij} \|a^j - c^i\|,\ |C| = p \Big\}. \tag{1}$$

В [2] для решения задачи был предложен ряд алгоритмов, считающихся теперь классическими в области размещения, например жадный алгоритм, предполагающий поиск мест размещения предприятий только в точках, соответствующих потребителям. Однако наиболее известным алгоритмом является так называемая альтернирующая эвристика, или алгоритм размещения-назначения (alternate location-allocation algorithm), или алгоритм Купера.

В его основе лежит наблюдение, что задача выбора мест размещения предприятий и задача присоединения потребителей к открытым предприятиям по отдельности просты. Например, если известны назначения потребителей (т. е. известно, какие потребители обслуживаются каким предприятием), то точки $c^i$ могут быть найдены для каждого подмножества потребителей $J_i$ с помощью алгоритма Вейсфельда (используя условия оптимальности первого порядка) [3]. С другой стороны, если известны точки размещения предприятий, то каждый потребитель присоединяется к предприятию, обеспечивающему минимальную стоимость

обслуживания. Таким образом, алгоритм Купера состоит в последовательном повторении процедур поиска точек размещения предприятий и назначения к ним потребителей. Если потребители $a^j$ представляют собой точки в многомерном пространстве и обладают единичным спросом $w_{ij} = 1$, а расстояния между предприятиями и потребителями заданы квадратичной евклидовой нормой, то задача (1) представляет собой не что иное, как известную в области машинного обучения задачу поиска минимума суммы квадратов (minimum-sum-of-squares clustering) или задачу о $k$-средних ($k$-means). Последняя на сегодняшний день является наиболее популярной, известной и широко используемой моделью кластеризации. Поскольку $w_{ij} = 1$, минимальную стоимость обслуживания потребителя обеспечивает ближайшее к нему предприятие. С другой стороны, так как расстояния заданы квадратичной нормой, задача поиска точки размещения предприятий $c^i$ для каждого заданного подмножества потребителей $J_i$ сводится к вычислению центра масс (среднего значения или центроида) векторов $a^j$, $j \in J_i$, т. е. $c^i = \frac{1}{|J_i|} \sum\limits_{j \in J_i} a^j$.

Алгоритм Купера для задачи о $k$-средних представляет собой известный в области машинного обучения одноимённый алгоритм $k$-средних ($k$-means), или алгоритм Ллойда, или алгоритм динамических ядер, который является в настоящий момент возможно самым популярным и известным алгоритмом кластеризации. Фактически, алгоритм Купера является одним из первых алгоритмов кластеризации.

Считается, что идея описанного алгоритма предложена С. Ллойдом в 1957 г., хотя его работа впервые опубликована только в 1982 г. [4]. При этом стоит отметить, что алгоритм Ллойда является частным случаем алгоритма Купера. Схожий алгоритм предложен также Э. В. Форги [5], а его варианты для непрерывной задачи, в которой места размещения потребителей заданы с помощью вероятностной меры, известны ещё с конца 1950-х гг. Алгоритм Купера может быть с лёгкостью обобщён для других случаев выбора метрики в задаче (1). Отметим, что центр масс оказывается оптимальным местом размещения предприятия для достаточно широкого класса мер расстояния, известного как дивергенции Брэгмана, к которому, в частности, относится квадрат евклидовой нормы [6]. Помимо алгоритма Ллойда в литературе часто под алгоритмом $k$-средних подразумевают так называемый алгоритм Маккуина, в котором после начального выбора центров кластеров объекты последовательно присоединяются к ближайшему центру, причём после присоединения нового объекта соответствующие центры обновляются [7].

На этом примере можно видеть, что многие модели и методы, возникшие в области задач размещения, являются фундаментальными для области анализа данных и машинного обучения. На протяжении своей истории задачи размещения и алгоритмы их решения привлекали огромное

внимание исследователей в области машинного обучения, исследования операции и приближённых алгоритмов. Поскольку исследования в каждом из направлений велись во многом обособленно, многие идеи и подходы, возникшие в одном сообществе, оказались малоизвестными для исследователей в других. Прежде всего это справедливо для приложений, где уровень проникновения новых идей традиционно остаётся низким. Несмотря на то, что в машинном обучении модели и методы, традиционные для задач размещения, используются достаточно широко, они зачастую явно не ассоциируются с этой областью, а потому во многих работах не всегда отчётливо прослеживается современное состояние исследований.

В данной статье планируется провести обзор идей и подходов, возникших в области теории размещения, которые нашли применение в машинном обучении и привели, например, к появлению современных популярных алгоритмов кластеризации, реализованных в большинстве программных пакетов по машинному обучению и техническим вычислениям. С другой стороны, будут рассмотрены подходы к решению задач размещения, возникшие в области машинного обучения. Поскольку непрерывные задачи размещения, в частности задача о $k$-средних, представляют собой чрезвычайно широкую область исследований, в данном обзоре основное внимание будет уделено дискретным задачам. В первую очередь будет проведён обзор подходов, известных и широко используемых в области машинного обучения, например, традиционных и современных алгоритмов кластеризации. Также будут кратко рассмотрены наиболее эффективные приближённые алгоритмы, точные методы и эвристики. Так как современное развитие технологий привело к накоплению больших массивов разнородных данных, анализ которых с применением имеющихся алгоритмов зачастую затруднён или попросту невозможен, планируется также рассмотреть современные подходы к реализации упомянутых алгоритмов для анализа больших массивов данных.

## 1. Постановки базовых задач

Рассмотрим следующий дискретный вариант обобщённой задачи Вебера, в котором задано множество точек (потребителей) $u^i \in \mathbb{R}^n$, $i \in I = \{1, \ldots, m\}$, и размещение предприятий допускается только среди потребителей, т. е. в конечном фиксированном числе мест. Можно предположить, что каждая точка представляет элемент данных, обладающий $n$ характеристиками, а расстояния $d_{ij} = d(u^i, u^j)$ задают степень «непохожести» между элементами $u^i$ и $u^j$, $i, j \in I$, или стоимость включения элемента $j$ в кластер $i$. Отметим, что хотя в классической постановке $d_{ij}$ предполагаются заданными евклидовой метрикой, в рассматриваемом случае вместо этого можно использовать другие способы определения

«расстояний», в первую очередь различные обобщения метрики (псевдометрики и т. д.). Также отметим, что в отличие от классической постановки обобщённой задачи Вебера точки предполагаются заданными в многомерном пространстве, а не на плоскости. Задача кластеризации тогда состоит в выборе подмножества точек $C \subset U$, $|C| = p$, называемых представителями кластера, таких что сумма расстояний между точками и ближайшими представителями минимальна, т. е.

$$\min_{S \subset I} \left\{ \sum_{j=1}^{m} \min_{i \in S} d_{ij},\ |S| = p \right\}.$$

Отметим, что все точки, для которых $i$ — ближайший представитель, составляют кластер.

Задача может быть представлена в целочисленном виде. Для этого вводятся булевы переменные $y_i$, принимающие значение 1, если точка $i$ является представителем кластера, а также переменные $x_{ij}$, равные единице, если элемент $j$ включён в кластер $i$ (представитель $i$ является для него ближайшим). В этом случае задача может быть записана так:

$$\min_{(x,y)} \sum_{i=1}^{m} \sum_{j=1}^{m} d_{ij} x_{ij}, \tag{2}$$

$$\sum_{i=1}^{m} x_{ij} = 1, \quad j = 1, \ldots, m, \tag{3}$$

$$\sum_{j=1}^{m} x_{ij} \leqslant m y_i, \quad i = 1, \ldots, m, \tag{4}$$

$$\sum_{i=1}^{m} y_i = p, \tag{5}$$

$$y_i, x_{ij} \in \{0, 1\}, \quad i, j = 1, \ldots, m. \tag{6}$$

Целевая функция (2) минимизирует суммарное расстояние между элементами данных и ближайшим представителем, ограничения (3) гарантируют, что каждый элемент присоединён только к одному кластеру, неравенства (4) обеспечивают присоединение элементов только к представителям, ограничения (5) и (6) задают условие на количество кластеров и целочисленность переменных.

Такая постановка задачи кластеризации была впервые предложена в [8] и является первой моделью кластеризации, сформулированной в виде задачи целочисленного линейного программирования. Модель была

вдохновлена известной обзорной статьей М. Л. Балинского [9], в частности, представленной в ней целочисленной постановкой так называемой простейшей задачи размещения предприятий. Последняя является вариантом задачи (2)–(6), в которой множество потребителей и множество пунктов размещения предприятий представлены отдельными множествами $J$ и $I$. Более того, для каждого $i \in I$ задана стоимость размещения в нём предприятия $f_i$, а $d_{ij}$ задают стоимость обслуживания потребителя $j$ предприятием $i$. В задаче требуется разместить предприятия таким образом, чтобы минимизировать суммарную стоимость обслуживания клиентов и размещения предприятий, т. е.

$$\min_{(x,y)} \sum_{i\in I} \sum_{j\in J} d_{ij}x_{ij} + \sum_{i\in I} f_i y_i, \tag{7}$$

$$\sum_{i\in I} x_{ij} = 1, \quad j \in J, \tag{8}$$

$$x_{ij} \leqslant y_i, \quad i \in I,\, j \in J, \tag{9}$$

$$y_i \in \{0,1\}, \quad x_{ij} \geqslant 0, \quad i \in I,\, j \in J. \tag{10}$$

Несмотря на явную отсылку к работе М. Л. Балинского, задача (2)–(6) использует несколько иной вид ограничений, связывающих переменные $y_i$ и $x_{ij}$. Действительно, если предположить, что $|I| = |J| = m$, то задача (7)–(10) содержит $m(1 + m)$ булевых переменных и $m(1 + m) + 1$ ограничений. В то же время, задача (2)–(6) содержит лишь $2m + 1$ ограничений, что достигается за счёт суммирования условий (9) для всех $j \in J$. Несмотря на то, что такой подход позволяет существенно уменьшить число ограничений, его недостатком является тот факт, что линейная релаксация задачи в общем случае даёт худшую двойственную оценку оптимального значения, поскольку множество (3), (4) вложено в (8), (9) при $y_i, x_{ij} \in [0, 1]$. Отметим, что ограничения (9) часто называют ограничениями Балинского, а (4) — ограничениями Эфроимсона — Рея, поскольку последние были впервые использованы в [10] в методе ветвей и границ для простейшей задачи размещения. Заметим также, что ограничения на целочисленность переменных $x_{ij}$ в (10) заменены условиями неотрицательности, поскольку всякое оптимальное решение задачи может быть представлено в виде эквивалентного решения, в котором все $x_{ij}$ принимают только булевы значения.

Независимо от работы Х. Д. Винода [8] представленная формулировка задачи кластеризации (2)–(6), использующая ограничения Балинского, была также предложена в известной работе [11] как задача размещения пунктов обслуживания, связанных с потребителями дорожной сетью. Отметим, что формулировка, представленная в [11], теперь считается классической.

Обобщённая задача Вебера носит также название непрерывной задачи о $p$-медиане, поскольку её решение для случая одного предприятия при единичном спросе потребителей представляет собой геометрическую медиану. Отметим, что для одномерного случая геометрическая медиана совпадает с медианой. Интересно, что описанный выше дискретный вариант обобщённой задачи Вебера также носит название задачи о $p$-медиане. При этом, говоря просто о задаче о $p$-медиане, подразумевают именно её дискретный вариант. Помимо связи с задачей Вебера такое название возникло благодаря работам С. Л. Хакими [12,13], в которых исследовался вариант фактически аналогичной задачи размещения предприятий, но на графе (поиск так называемой $p$-медианы графа). Задача рассматривалась в приложении к размещению заданного числа коммутаторов в некоторой коммуникационной сети, ответственных за обработку и передачу данных от одного клиента другому с целью минимизации суммарной длины линий связи. Причём в первоначальной постановке предполагалось, что каждое предприятие (медиана) может размещаться в любой точке сети. Тем не менее, сперва для одной медианы [12], а затем и для $p$-медианы [13] Хакими доказал так называемое свойство Хакими, утверждающее, что хотя бы одно оптимальное решение задачи предполагает размещение предприятий в вершинах сети. Это свойство фактически сводит задачу о $p$-медиане на графе к дискретной задаче, в которой расстояния задаются кратчайшим путём, соединяющим две соответствующие вершины.

Отметим особо, что (дискретная) задача о $p$-медиане (2)–(6) также широко известна в области машинного обучения как задача о $k$-медоидах. Это название стало популярным с выходом работы [14], в которой авторы посчитали, что название $p$-медиана может внести путаницу с понятием геометрической медианы, т. е. с непрерывной задачей. В области приближённых алгоритмов задача носит название задачи о $k$-медиане (см., например, [15], где предложен первый приближённый алгоритм с константной оценкой точности), причём её действительно часто путают с непрерывной версией, где расстояния заданы метрикой (чаще всего евклидовой) [16]. В непрерывном виде задачу часто исследуют вместе с обобщённой задачей о $k$-средних, в которой вместо квадрата евклидова расстояния рассматривается квадрат произвольной метрики. С другой стороны, задача о $k$-средних может быть сведена к дискретной задаче, в которой число возможных центров кластеров ограничено некоторым фиксированным числом точек (т. е. фактически к задаче о $p$-медиане) с произвольно малой потерей гарантированной точности [17].

Задачей о $k$-медиане также часто называют вариант обобщённой задачи Вебера, в котором расстояния задаются метрикой $l_1$ (манхэттенские расстояния). Это задача также встречается под названием задачи

о $k$-средних относительно метрики $l_1$. Её решение для случая одного предприятия при единичном спросе потребителей называется геометрическим медианным абсолютным отклонением.

Задача о $p$-медиане на графе NP-трудна даже в случае планарного графа с максимальной степенью вершин равной 3, однако полиномиально разрешима на дереве [18]. В непрерывной постановке задача NP-трудна даже на плоскости, в случае если расстояния заданы евклидовой, манхэттенской метрикой [19], а также квадратом евклидова расстояния (задача о $k$-средних) [20], однако полиномиально разрешима на вещественной прямой [21]. Непрерывная задача о $p$-медиане, в которой расстояния заданы квадратом евклидовой метрики, NP-трудна для произвольной размерности $n$ при $p = 2$ [22]. Дискретный вариант задачи, в котором расстояния заданы евклидовой метрикой, также является NP-трудной задачей [23].

Некоторые последующие работы были посвящены уменьшению размерности классической постановки задачи о $p$-медиане с сохранением качества линейной релаксации. Первые попытки были сделаны в [24], где авторы предложили использовать ограничения Эфроимсона — Рея, а ограничения Балинского для каждого потребителя включить в формулировку только для некоторого фиксированного числа $t$ ближайших предприятий. Это позволило уменьшить число ограничений и в то же время сохранить «качество» решения линейной релаксации. Более того, в статье было замечено, что каждый клиент может быть присоединён в худшем случае лишь к $(m-p+1)$-му ближайшему предприятию (представителю кластера), тем самым число переменных $x_{ij}$ задачи может быть уменьшено соответствующим образом. В [25] была предложена так называемая постановка COBRA, в которой размерность задачи предлагается уменьшать за счёт исключения «эквивалентных» $x_{ij}$. Предполагается, что если для двух различных потребителей $j_1, j_2 \in I$ выполнено

$$O_{j_1 t} = \{l \in I \mid d_{lj_1} < d_{tj_1}\} = \{l \in I \mid d_{lj_2} < d_{tj_2}\} = O_{j_2 t},$$

то для оптимального решения задачи справедливо $x_{tj_1} = x_{tj_2}$. Другими словами, если два потребителя $j_1$ и $j_2$ имеют одно и то же $t$-е ближайшее предприятие и множество $1, \ldots, t-1$ ближайших предприятий совпадает, то размер формулировки может быть сокращён за счёт замены переменной $x_{tj_2}$ на $x_{tj_1}$.

Другой подход к постановке дискретных задач размещения, идейно близкий к задаче о покрытии множеств, был предложен в известной работе [26] и обобщён для задачи о $p$-медиане в [27]. Предположим, что $d_{ii} = 0$, $d_{ij} \geqslant 0$, $i, j \in I$, $i \neq j$, и для каждого $j \in I$ вектор $(\bar{d}_{1j}, \bar{d}_{2j}, \ldots, d_{G_j j})$ представляет собой упорядоченные по возрастанию, неповторяющиеся расстояния до всех $i \in I$, $i \neq j$, т. е. $0 = \bar{d}_{1j} < \bar{d}_{2j} <$

$\cdots < \bar{d}_{G_j j} = \max\limits_{i=1,\ldots,n} d_{ij}$. Введём множество переменных $z_{kj}$, названных в [27] *кумулятивными*, таких что

$$z_{kj} = \begin{cases} 1, & \text{если расстояние от объекта } j \text{ до ближайшего} \\ & \text{представителя равно по крайней мере } \bar{d}_{kj},\ j \in I, \\ & 2 \leqslant k \leqslant G_j, \\ 0 & \text{в противном случае.} \end{cases}$$

Задача о $p$-медиане в этом случае может быть записана следующим образом:

$$\min \sum_{j=1}^{m} \sum_{k=2}^{G_j} (\bar{d}_{kj} - \bar{d}_{k-1,j}) z_{kj}, \tag{11}$$

$$z_{kj} + \sum_{i \in I \,|\, d_{ij} < \bar{d}_{kj}} y_i \geqslant 1, \quad j = 1, \ldots, m,\ 2 \leqslant k \leqslant G_j, \tag{12}$$

$$\sum_{i=1}^{m} y_i = p, \tag{13}$$

$$y_i \in \{0, 1\}, \quad i = 1, \ldots, m, \tag{14}$$

$$z_{kj} \geqslant 0, \quad j = 1, \ldots, m,\ 2 \leqslant k \leqslant G_j. \tag{15}$$

Поскольку все расстояния упорядочены, целевая функция принимает только положительные значения. Ограничения (12) гарантируют, что переменная $z_{kj}$ принимает значение 1, если не существует представителя кластера на расстоянии ближе чем $\bar{d}_{kj}$. В противном случае переменная принимает значение 0, так как рассматривается задача на минимум. Интересно, что указанная формулировка не содержит переменных, указывающих, каким образом объекты присоединяются к кластерам. Вместо этого переменные $z_{kj}$ характеризуют минимальные расстояния до представителя кластера. Представленная формулировка и классическая постановка задачи содержат один и тот же набор переменных $y_i$. В то же время вторая содержит $m^2$ переменных $x_{ij}$ и $m^2$ соответствующих ограничений. В наихудшем случае, если все расстояния от каждого $j \in I$ до всех других объектов различны, количество переменных $z_{kj}$ и соответствующих ограничений в задаче (11)–(15) также равно $m^2$. Однако для каждого повторяющегося расстояния формулировка будет содержать на одно ограничение и переменную меньше, что может существенно сократить размерность задачи для некоторых практических задач кластеризации, в которых матрицы расстояний разрежены или содержат большое число дубликатов. Несмотря на это, как отмечено в [27], решение линейной релаксации задачи (11)–(15) даёт такую же нижнюю

оценку оптимального значения, как и линейная релаксация классической постановки.

Несмотря на зачастую существенный выигрыш в размерности, поиск решения в (11)–(15) с помощью коммерческих решателей может занять намного больше времени, что вызвано большой плотностью матрицы ограничений в сравнении с классической постановкой. Чтобы избежать этого недостатка и в то же время сохранить преимущества постановки (11)–(15), в [28] была предложена альтернативная формулировка. Заметим, что для всякого набора переменных $y$ значения $z_{kj}$ могут быть вычислены так:

$$z_{kj} = \prod_{i \in I \,|\, d_{ij} < \bar{d}_{kj}} (1 - y_i), \quad j = 1, \ldots, m, \ 2 \leqslant k \leqslant G_j.$$

Таким образом, $z_{kj}$ могут быть найдены по следующему рекуррентному правилу:

$$z_{1j} = \prod_{i \in I \,|\, d_{ij} = \bar{d}_{1j}} (1 - y_i), \quad z_{kj} = z_{k-1,j} \prod_{i \in I \,|\, d_{ij} = \bar{d}_{kj}} (1 - y_i).$$

С использованием введённых обозначений ограничения (12) могут быть записаны в следующей эквивалентной форме:

$$z_{1j} + \sum_{i \in I \,|\, d_{ij} = \bar{d}_{1j}} y_i \geqslant 1, \quad j = 1, \ldots, m,$$

$$z_{kj} + \sum_{i \in I \,|\, d_{ij} = \bar{d}_{kj}} y_i \geqslant z_{k-1,j}, \quad j = 1, \ldots, m, \ 2 \leqslant k \leqslant G_j.$$

Несмотря на то, что использование такого типа ограничений не позволяет улучшить значение линейной релаксации задачи, её допустимое множество включено в допустимое множество линейной релаксации (11)–(15). Такая улучшенная формулировка содержит то же число ограничений, однако матрица ограничений оказывается более разреженной, что позволяет коммерческому решателю находить решение намного быстрее, что было показано в ходе вычислительных экспериментов.

## 2. Алгоритмы кластеризации

Как отмечено выше, алгоритмы, направленные на поиск допустимых решений в задачах размещения, являются наиболее популярными и широко распространенными в настоящий момент алгоритмами кластеризации, реализованными в множестве специализированных программных библиотек по машинному обучению. Наиболее популярный — алгоритм $k$-средних — является алгоритмом локального поиска (альтернирующей эвристикой) для частного случая обобщённой задачи Вебера. В данном

разделе будут рассмотрены, возможно, не менее популярные и известные алгоритмы кластеризации, основанные на поиске решений в дискретном варианте той же самой обобщённой задачи Вебера (задачи о $p$-медиане). Однако прежде всего рассмотрим несколько классических эвристик, которые лежат в основе большинства описываемых в дальнейшем алгоритмов.

Одной из первых эвристик для задачи о $p$-медиане на графе был так называемый алгоритм Маранцаны [29], который фактически является вариантом алгоритма Купера (алгоритма $k$-средних) для случая постановки задачи на графе. Отметим, что хотя в оригинальной статье исследуемая задача не связывается с задачей о $p$-медиане Хакими, она предполагает размещение предприятий (медиан) в вершинах графа. Алгоритм начинает работу с некоторого начального множества медиан $p$ и присоединяет все остальные вершины к ближайшей медиане. В качестве расстояний используются взвешенные кратчайшие пути между вершинами. Для каждого получившегося подмножества вершин (кластера), присоединённых к одной и той же медиане, алгоритм находит новую медиану, взвешенная сумма расстояний до которой от всех остальных вершин подмножества минимальна, т. е. решает задачу о 1-медиане. Если расположение медиан в кластерах не изменилось, то алгоритм останавливается; в противном случае алгоритм переходит на следующую итерацию. Как было отмечено в оригинальной работе, алгоритм в общем случае сходится лишь к некоторому локально оптимальному решению задачи. Несмотря на схожесть с алгоритмом Ллойда, алгоритм Маранцаны имеет существенно более высокую вычислительную сложность, поскольку для поиска представителя кластера требует квадратичного от числа вершин в каждом кластере времени (см. алгоритм 1). Поиск же центроида в алгоритме $k$-средних требует линейного времени.

**Алгоритм 1.** Алгоритм Маранцаны

1: Выбрать начальное решение $S \subseteq I$, $|S| = p$.
2: Положить $J_i = \varnothing$, $i \in S$. Найти $J_s \cup \{j\}$: $s = \operatorname{argmin}\limits_{i \in S} d_{ij}$, $j \in I$, $j \notin S$.
3: Положить $S' = \varnothing$. Если $|J_i| > 0$, $i \in S$, то $S' \cup \{s'\}$: $s' = \operatorname{argmin}\limits_{s \in J_i} \sum\limits_{j \in J_i} d_{sj}$.
4: Если $S \neq S'$, то $S = S'$ и перейти на шаг 2, иначе stop.

Другой классической эвристикой для задачи о $p$-медиане является так называемый алгоритм Тейтца и Барт [30], известный также как алгоритм подстановки вершин, который представляет собой классический алгоритм локального поиска для задачи о $p$-медиане. Его основная идея состоит в том, чтобы, начиная с некоторого начального набора медиан,

последовательно исследовать соседние решения, получающиеся заменой одной медианы из этого набора на немедианную вершину. Если существует замена, улучшающая значение целевой функции, то она принимается и алгоритм исследует точки в окрестности нового решения. Алгоритм останавливается, если не существует возможных замен, улучшающих текущее лучшее значение целевой функции. Оригинальная статья содержит несколько неточностей, например, в вычислении значения целевой функции или в стратегии выбора следующего текущего решения из окрестности. Заметим, что схожий по идее алгоритм для задачи о $k$-средних известен как алгоритм Хартигана — Вонга [31].

Наиболее известным алгоритмом кластеризации, основанным на поиске решений в дискретной задаче о $p$-медиане, является алгоритм PAM (Partition Around Medoids) [14]. Он представляет собой алгоритм Тейтца и Барт, стартующий с решения, найденного с помощью классической реализации жадного алгоритма. В оригинальном описании PAM состоит из двух этапов: построение (BUILD), на котором ищется начальное жадное решение, и перестановка (SWAP), на котором применяется алгоритм перестановки вершин с целью улучшения жадного решения. Отметим, что на втором этапе PAM использует так называемую стратегию наискорейшего спуска, т. е. всегда ищет соседнее решение, приводящее к наилучшему уменьшению значения целевой функции. Поскольку алгоритм, сочетающий два таких компонента, был хорошо известен и широко использовался как тестовый алгоритм задолго до публикации PAM [14] (хотя в статье ссылки на известные работы не были представлены), авторы называют PAM программой, так как основной новизной работы была программная реализация такой эвристики на FORTRAN с использованием большого числа входных параметров, задаваемых пользователем. Отметим, что в качестве расстояний между объектами были использованы евклидова и манхэттенская метрики.

Отметим, что одно из первых описаний идеи классического жадного алгоритма для задачи о $p$-медиане встречается в [32]. В [33] исследовался классический жадный алгоритм и представлены теоретические оценки на качество получаемых решений в наихудшем случае. В статье также был рассмотрен обратный жадный алгоритм (greedy drop). Его основное отличие состоит в том, что на первой итерации все элементы данных выбраны в качестве представителей кластеров. Далее, на каждом шаге представитель, исключение которого приводит к наименьшему увеличению значения целевой функции, выбрасывается из решения. Этот процесс повторяется пока неотброшенными остаются $p$ элементов. Наконец, в [33] была также рассмотрена эвристика, аналогичная использованной намного позже в PAM, названная жадной перестановкой (greedy-interchage), в которой для найденного сначала жадного решения

применяется алгоритм Тейтца и Барт. В частности, было доказано, что в наихудшем случае такая комбинированная эвристика не может найти решение лучшее, чем решение жадного алгоритма.

Тем не менее, публикация PAM привлекла чрезвычайно широкое внимание сообщества машинного обучения и анализа данных, в те годы набиравшего силу. Это привело к существенному росту интереса к задачам размещения как к моделям кластеризации, так и в качестве ключевых компонентов для разработки других подходов интеллектуального анализа данных. Результатом этого стала публикация большого числа статей в данной области, рассматривающих такие аспекты, как анализ, приложения и модификации того же подхода PAM.

**Алгоритм 2.** Программа PAM, шаг BUILD

1: Инициализировать $S = \varnothing$, $S \cup \{i\}$: $i = \underset{s \in I}{\operatorname{argmin}} \sum_{j=1}^{m} \min d_{sj}$.
2: Пусть $i \in I$, $j \in I \setminus \{i\}$, $i, j \notin S$. Вычислить $C_{ji} = \max\{d_1(j) - d_{ij}, 0\}$ и $\sum_j C_{ji}$, где $d_1(j)$ — расстояние до ближайшей медианы из $S$.
3: Положить $S \cup \{i\}$: $i = \underset{s \in I,\, s \notin S}{\operatorname{argmax}} \sum_j C_{js}$.
4: Если $|S| = p$, то stop; иначе перейти на шаг 2.

В оригинальном описании эвристики, использованной в PAM, вводятся величины $d_1(j)$, $d_2(j)$, задающие расстояния до ближайшей и второй по близости медианы. Начальное жадное решение $S$ может быть найдено следующим образом на шаге BUILD (см. алгоритм 2). После того как начальное решение построено, выполняется второй шаг SWAP, направленный на его улучшение с помощью алгоритма Тейтца и Барт (см. алгоритм 3). Он рассматривает все возможные пары $(i, h) \in S \times I \setminus S$ и подсчитывает величину изменения значения целевой функции при перестановке $i$ и $h$.

Отметим, что поиск начального решения с помощью жадного алгоритма требует $O(pm^2)$ времени в худшем случае. На шаге SWAP поиск ближайшей и второй ближайшей медианы (представителя) требует $O(pm)$ операций. Далее, на каждой итерации требуется оценить изменение значения целевой функции для $p(m-p)$ возможных пар. Если предположить, что расстояния между элементами могут быть подсчитаны за время $O(1)$, т. е., например, матрица расстояний найдена и известна перед запуском алгоритма, то одна итерация SWAP требует $O(p(m-p)^2)$ времени.

Среди достоинств PAM в [14] отмечены его более высокая по сравнению с алгоритмом $k$-средних устойчивость к наличию выбросов и шума

**Алгоритм 3.** Программа PAM, шаг SWAP

Дано начальное решение $S \subseteq I$, $|S| = p$, подсчитать $d_1(j)$, $d_2(j)$.
**for** $(i, h) \in S \times I \setminus S$ **do**
    **if** $d_{ji} > d_1(j)$ **then**
        **if** $d_{jh} \geqslant d_1(j)$ **then** $C_{jih} = 0$
        **if** $d_{jh} < d_1(j)$ **then** $C_{jih} = d_{jh} - d_1(j)$
    **end if**
    **if** $d_{ji} = d_1(j)$ **then**
        **if** $d_{jh} < d_2(j)$ **then** $C_{jih} = d_{jh} - d_1(j)$
        **if** $d_{jh} \geqslant d_2(j)$ **then** $C_{jih} = d_2(j) - d_1(j)$
    **end if**
    Вычислить $T_{ih} = \sum_{j=1}^{m} C_{jih}$.
**end for**
Найти пару $(i, h)$, для которой $T_{ih}$ минимальна.
Если $T_{ih} < 0$, то $S \cup \{h\} \setminus \{i\}$ и перейти на шаг 2; иначе stop.

в данных, а также гибкость в выборе расстояний между объектами, т. е. последние могут быть заданы не только с помощью дивергенций Брэгмана. Более того, для PAM, вообще говоря, требуется только матрица расстояний, задающая отношения схожести между объектами. Так, несхожести могут быть найдены с помощью людей-аннотаторов или вычислительно сложных процедур, используемых, например, в биоинформатике для подсчёта схожести белков.

Данное обстоятельство часто отмечается как преимущество именно в сравнении с алгоритмом $k$-средних. Однако последний является вариантом алгоритма Купера для варианта обобщённой задачи Вебера, в которой расстояния заданы с помощью квадрата евклидовой метрики. Таким образом, алгоритм Купера может быть адаптирован и для других способов выбора расстояний, но требует решения задачи поиска представителя кластера (например, с помощью алгоритма Вейсфельда). Другим полезным преимущество PAM в сравнении с алгоритмом $k$-средних является то, что найденный представитель кластера является элементом данных, что существенно для интерпретируемости полученных результатов, например, при кластеризации изображений. Таким образом, представители могут быть использованы для дальнейшего анализа вместо исходных данных.

Среди недостатков PAM стоит выделить то обстоятельство, что он сходится лишь к некоторому локально оптимальному решению. Более того, существенным недостатком является его высокая вычислительная сложность, присущая как первому, так и второму шагу. Это значительно

затрудняет его применение для задач кластеризации большой размерности. Классическая реализация предполагает, что матрица попарных расстояний найдена и подаётся на вход алгоритма. Отметим, что подсчёт матрицы расстояний требует $O(m^2n)$ операций. Помимо вычислительной сложности при больших значениях $m$ и $n$, дополнительную трудность в случае больших данных представляет и необходимость хранения такой матрицы во время работы алгоритма. Например, хранение матрицы расстояний для 10000 элементов требует 800 МБ памяти, а для 100000 — уже 80 ГБ (в случае, если каждый элемент матрицы представлен числом с плавающей точкой двойной точности). Одним из возможных подходов является вычисление расстояний между объектами по мере необходимости, что, однако, приводит к увеличению числа необходимых операций для шага BUILD и одной итерации SWAP в $O(n)$ раз и существенно замедляет работу алгоритма для данных большой размерности.

В [34, 35] представлен первый анализ алгоритмов локального поиска, в частности, алгоритма Тейтца и Барт, для дискретных задач размещения. Авторами [34] было введено понятие так называемого локального разрыва (locality gap) для алгоритмов локального поиска — это максимальное отношение значения целевой функции в найденном локальном решении к глобальному решению. При этом показано, что локальный разрыв для алгоритма Тейтца и Барт равен 5, т. е. найденное с помощью него решение хуже глобально оптимального не более чем в пять раз. Если в локальном поиске заменяются одновременно $k$ медиан, то локальный разрыв составляет $3 + \frac{2}{k}$. В случае простейшей задачи размещения показано, что он составляет 3. Таким образом, PAM в метрическом случае является приближённым алгоритмом с константой 5. В [36,37] исследованы свойства локального поиска для задачи о $p$-медиане относительно полиномиально просматриваемых окрестностей. В частности, показано, что задача локального поиска относительно ряда окрестностей PLS-полна. Более того, авторами доказано, что локальный поиск в наихудшем случае находит локальное решение за экспоненциальное число итераций, вне зависимости от правила выбора следующего решения из окрестности.

Первые эффективные реализации эвристики, сочетающей жадный алгоритм и локальный поиск, были предложены в [38] задолго до публикации PAM. Основная идея быстрой реализации жадного алгоритма состоит в том, что, начиная со второй итерации, лишь часть элементов данных меняют своё назначение после добавления нового представителя. Таким образом, вычисление $\sum_j C_{ji}$ на шаге 2 (см. алгоритм 2) может быть произведено только относительно элементов, изменивших своё назначение к кластерам на предыдущей итерации. Такой подход позволяет избежать повторяющихся вычислений на каждой итерации. В основе предложенной в той же статье идеи быстрой реализации локального

поиска, получившей название процедуры Уайтакера, лежит наблюдение, что для каждого из $m - p$ кандидатов на включение в множество представителей (медиан), соответствующий кандидат на исключение может быть быстро найден за время $O(m)$, т. е. за один проход по элементам данных вместо $p$. Такой подход требует дополнительно $O(p)$ памяти, что не является критическим даже для практических задач большой размерности. С использованием процедуры Уайтакера одна итерация шага SWAP в PAM требует лишь $O((m-p)^2)$ операций, что в $p$ раз быстрее оригинальной процедуры. В статье была также предложена эффективная процедура пересчёта первой и второй ближайших медиан.

В [39] предложены многочисленные варианты модификаций жадных алгоритмов, а также вариант реализации алгоритма перестановки вершин. Его идея состоит в том, чтобы помимо единичной перестановки медианы и немедианы попытаться улучшить значение целевой функции за счёт применения алгоритма Маранцаны к получившемуся набору медиан. Схожий подход был использован и при реализации вариантов жадных алгоритмов, классического и обратного. В [40] была предложена ещё одна эвристика локального поиска, сочетающая алгоритмы Тейтца и Барт и Маранцаны, в которой для поиска кандидатов на включение и исключение из текущего решения используются жадный и обратный жадный алгоритмы соответственно. Если найденная пара не позволяет улучшить значение целевой функции, то запускается алгоритм Маранцаны.

К сожалению, реализация Уайтакера долгие годы оставалась незамеченной как исследователями задач размещения, так и сообществом машинного обучения. Например, оригинальная реализация PAM не использует быстрых вариантов жадного алгоритма и локального поиска. Подход Уайтакера для локального поиска, как полагают, был популяризирован Хансеном и Младеновичем [41], которые разработали классическую эффективную реализацию и применили её в своём алгоритме локального поиска с чередующимися окрестностями.

Совсем недавно очень похожая процедура была реализована для PAM в [42], а соответствующий вариант PAM получил название FastPAM1. В статье также предложен алгоритм FastPAM2, который состоит в поиске наилучшей перестановки в локальном поиске для нескольких медиан за одну итерацию. Однако представленные вычислительные результаты показали, что FastPAM2 уступает FastPAM1.

Поскольку жадный алгоритм, используемый для инициализации, имеет чрезвычайно высокую вычислительную сложность и занимает больше половины времени работы алгоритма PAM (даже с учётом времени, необходимого на вычисление матрицы попарных расстояний), авторами предложен его быстрый вариант, названный LAB (Linear Approximative

BUILD), который состоит в использовании выборки из $10 + \sqrt{m}$ элементов для поиска в ней новой медианы на каждой итерации алгоритма. Для улучшения результатов такую процедуру предлагается повторить $p$ раз.

В [43] предложена ещё одна быстрая реализация алгоритма Тейтца и Барт, сложность которой идентична реализации Уайтакера, однако на практике она демонстрирует трёхкратный выигрыш в скорости. Идея подхода состоит в использовании дополнительных вспомогательных структур данных для хранения информации, полученной на ранних итерациях, чтобы уменьшить объём вычислений на последующих.

Тем не менее быстрая реализация жадного алгоритма [38] и алгоритма локального поиска [38,43], а также эвристики из [39] (жадные алгоритмы и вариант алгоритма перестановки вершин) не получили распространения (последние — из-за сложности эффективной реализации) и остались в большинстве своём неизвестны в сообществе машинного обучения.

Другой вариант эвристики, сочетающей жадный алгоритм и локальный поиск, был предложен в [44] и получил название GRASP. Он представляет собой мультистартный алгоритм локального поиска, где начальное решение выбирается с помощью рандомизированного жадного алгоритма. В частности, для задачи о $p$-медиане в [45] разработана GRASP эвристика, в рамках которой предложены пять подходов к выбору начального жадного решения, использована быстрая реализация локального поиска [43], а также реализована так называемая процедура связывающих путей (path relinking). Отметим, что недавно предложенный FastPAM1 фактически является вариантом GRASP.

В [46] предложен стохастический жадный алгоритм, являющийся вариантом так называемого ленивого жадного алгоритма. Предложенный подход основан на представлении задачи о $p$-медиане как задачи максимизации субмодулярной функции. На каждой итерации он находит нового представителя не относительно всего множества элементов, а только из случайной выборки размера $\frac{m}{p} \log \frac{1}{\varepsilon}$. В статье показано, что такой алгоритм в среднем находит $\left(1 - \frac{1}{e} - \varepsilon\right)$-оптимальное решение и требует лишь $\mathcal{O}\left(m \log \frac{1}{\varepsilon}\right)$ вычислений значения целевой функции. Отметим, что описанный выше LAB следует той же идее.

Наконец, недавно предложен вероятностный вариант PAM, названный Bandit-PAM. Он основан на представлении задачи поиска жадного решения, а также лучшего решения из окрестности на каждой итерации локального поиска как задачи поиска наилучшей руки (многорукого бандита) [47]. Показано, что при некоторых специфических условиях, например, на распределение данных ($\sigma$-субгауссово относительно расстояний между элементами) с большой долей вероятности Bandit-PAM находит решение, эквивалентное PAM, причём математическое ожидание числа операций на одну итерацию составляет $O(m \log m)$, вместо $O(m^2)$ в PAM.

Поскольку Тейтц и Барт в [30] продемонстрировали преимущества своего алгоритма локального поиска над алгоритмом Маранцаны, последний оставался не слишком популярным и использовался в основном только в рамках гибридизации с другими подходами [39]. После публикации программы PAM и привлечения внимания к задачам размещения в области машинного обучения алгоритм Маранцаны был «открыт заново» как вариант алгоритма $k$-средних [48]. Алгоритм Маранцаны, который в случае задачи кластеризации ищет медианы в полном графе, в настоящее время часто называют алгоритмом поиска $k$-медоидов (или $k$-medoids).

Наиболее вычислительно сложной операцией алгоритма Маранцаны является поиск новой медианы каждого кластера (т. е. решение задачи об 1-медиане), который в наивной реализации может быть выполнен за время, квадратичное от размера кластера $N$. Тем не менее во многих случаях для этой цели могут быть использованы более быстрые подходы: например, точный алгоритм с ожидаемым временем работы $O(N^{\frac{3}{2}})$ при фиксированной размерности $n$ для метрического случая, неэффективный однако для задач при большом числе признаков [50]; рандомизированный алгоритм, находящий с высокой вероятностью медиану за $O(N \log N)$ операций [51]; или эвристический подход, основанный на поиске так называемых ключевых (якорных) точек для быстрого поиска приближённой медианы кластера [52].

Высокая вычислительная сложность первоначальной эвристики, реализованной в PAM, препятствует её применению в анализе данных большой размерности (содержащих большое число элементов данных, кластеров и признаков), в том числе и из-за невозможности хранения большой матрицы расстояний. Наиболее популярный подход в обобщении алгоритмов кластеризации для задач большой размерности состоит в отборе случайных выборок. Так, авторами PAM была предложена так называемая программа CLARA (Clustering Large Applications) для задач большой размерности, в которой PAM применяется не к первоначальным данным, а только к некоторой выборке [53]. После этого каждый элемент данных, не принадлежащий выборке, присоединяется к ближайшей найденной медиане для вычисления значения целевой функции. В первоначальной реализации авторами был предложен размер выборки $2p + 40$ (установленный эмпирически), а число выборок предполагалось равным пяти, после чего наилучший набор медиан и соответствующее значение целевой функции предъявлялось как окончательное решение. Недостатком CLARA является низкая эффективность для задач с большим числом кластеров.

Другой подход к отбору выборок применительно к алгоритму Тейтца и Барт был реализован в так называемом алгоритме CLARANS [54],

который представляет собой локальный поиск с мультистартом, где, в отличие от PAM, поиск осуществляется не по всей окрестности, а только по случайно выбранным соседним решениям, причём переход осуществляется в первое найденное улучшающее решение. В качестве критерия остановки используется число итераций локального поиска без улучшения значения целевой функции и количество перезапусков. В [55] проведены вычислительные эксперименты, демонстрирующие более высокую эффективность CLARANS по сравнению с алгоритмом Маранцаны, а также предложены некоторые техники повышения его эффективности.

Несмотря на то, что CLARA и CLARANS могут быть применены для задач очень большой размерности, качество получаемых решений оказывается намного ниже в сравнении с PAM. Очевидно, найденные решения лишь допустимы в задаче кластеризации, но не локально оптимальны.

Тем не менее PAM, CLARA, CLARANS и алгоритм Маранцаны (алгоритм $k$-medoids) являются одними из наиболее известных и популярных в настоящий момент алгоритмов кластеризации. Многочисленные исследования были посвящены различным аспектам улучшения данных алгоритмов таким, как уменьшение вычислительной сложности и/или разработка стратегий выбора начального решения. Например, в одной из ранних работ был предложен подход для кластеризации больших массивов пространственных данных, представляющий собой вариант CLARANS, который применяется к выборке данных, определяемой с помощью $R^*$-деревьев. Некоторые авторы демонстрировали преимущества применения алгоритма Тейтца и Барт для случаев, когда расстояния заданы с помощью так называемых «силуэтов» кластера [56]. В [57] предложен алгоритм CLATIN для кластеризации пространственных данных, который представляет собой алгоритм локального поиска, в котором для поиска наилучшей перестановки медианы и немедианы применяется нерегулярная триангуляционная сеть. В известной работе [49] был предложен вариант алгоритма Маранцаны, в котором начальные медианы выбираются как $p$ объектов $i \in I$, для которых $\sum\limits_{j=1}^{m} \frac{d_{ij}}{\sum\limits_{l=1}^{m} d_{jl}}$ минимальны. Такие объекты соответствуют по свидетельству авторов наиболее «центральным» элементам данных. В [58] предложен вариант алгоритма Маранцаны, в котором начальное решение ищется итерационно. Сначала алгоритм находит множество потенциальных медиан, т. е. множество точек, дисперсия которых меньше дисперсии данных, умноженной на регулирующий параметр. Начиная с поиска двух медиан как наиболее удалённых между собой элементов из множества кандидатов, каждая следующая медиана ищется как элемент кластеров, наиболее удалённый от уже выбранных медиан. Как и процедура инициализации в FastK [49], такой способ выбора начального решения имеет очень высокую вычислительную

сложность. Более быстрый вариант данного алгоритма, в котором для поиска представителя кластера используются центроид, геометрическая медиана и среднее значение, был предложен в [59]. В [60] приведена модификация алгоритма Маранцаны, где поиск новой медианы в каждом кластере основан на использовании информации о ближайших соседях, входящих в кластер объектов. Другой подход к поиску медоидов был предложен в [61], где последовательные выборки использовались для построения многодольного графа и поиска схожих объектов, принадлежащих одному и тому же кластеру. Получившийся в итоге граф состоит из $p$ деревьев, медианы которых и предъявляются в качестве решения. Наконец, некоторыми авторами предпринималась попытка улучшения жадного алгоритма для частных случаев выбора метрики, например, манхэттенской, так что сложность алгоритма становится линейно-логарифмической [62].

### 3. Точные и приближённые методы для задачи о $k$-медоидах

Описанные выше популярные алгоритмы кластеризации представляют собой по большей части эвристики локального поиска и/или версии жадного алгоритма для задачи о $k$-медоидах. Хотя авторы PAM напрямую связывали свой метод о $k$-медоидах с известной задачей о $p$-медиане и представили некоторый краткий обзор наиболее эффективных ранних точных методов, значительный прогресс 1990-х и 2000-х гг. в области разработки метаэвристик, точных методов и приближённых алгоритмов фактически не оказал существенного влияния на сообщество машинного обучения, даже несмотря на то, что многие практические задачи кластеризации могут быть решены точно или с некоторой гарантированной точностью. Обзоры точных и приближённых алгоритмов представлены в работах [63–65].

Один из наиболее эффективных точных алгоритмов предложен в [66] и представляет собой метод ветвей, отсечений и оценок. В качестве отсечений использованы три новых семейства правильных неравенств, в том числе предложены так называемые $W$-$q$-неравенства. Задача о $p$-медиане формулируется на полном простом орграфе. Основным компонентом подхода является метод генерации строк и столбцов для решения линейных релаксаций, в котором координирующая задача (мастер-задача) определяется на подграфе для некоторого подмножества дуг. Это подмножество увеличивается, если решение координирующей задачи не даёт решения линейной релаксации согласно установленным условиям оптимальности, и координирующая задача решается вновь.

В [67] предложен точный метод, основанный на так называемой постановке BEAMR. Её идея состоит в том, чтобы искать решение задачи о $p$-медиане только среди некоторого фиксированного числа ближайших

предприятий для каждого клиента, т. е.

$$\min \sum_{i=1}^{m} \sum_{j \in H_i} d_{ij} x_{ij} + \sum_{i=1}^{m} d_{is_i} g_i, \tag{16}$$

$$\sum_{j \in H_i} x_{ij} + g_i = 1, \quad i = 1, \ldots, m, \tag{17}$$

$$x_{ij} \leqslant y_i, \quad i = 1, \ldots, m,\ j \in H_i \tag{18}$$

$$\sum_{i=1}^{m} y_i = p, \tag{19}$$

$$y_i, x_{ij} \in \{0, 1\}, \quad i = 1, \ldots, m,\ j \in H_i, \tag{20}$$

где $H_i$ — множество $h_i$ ближайших предприятий для клиента $i$, а $s_i$ — индекс $(h_i + 1)$-го ближайшего предприятия. Переменная $g_i$ принимает значение 1, если потребитель $i$ должен быть присоединён к предприятию, более удалённому чем $h_i$. Очевидно, что любое решение такой редуцированной задачи даёт верхнюю оценку оптимального значения задачи о $p$-медиане, вне зависимости от выбора $h_i$. С другой стороны, если в оптимальном решении задачи (16)–(20) для некоторого $h_i$ все переменные $g_i$ принимают значение 0, то полученное решение оптимально и в исходной задаче. Таким образом, авторами были предложены точный и приближённый алгоритмы, близкие по идее к методу из [66]. Они состоят в последовательном решении задачи (16)–(20), начиная с заданных $h_i$. Если в оптимальном решении некоторые $g_i > 0$, то необходимо увеличить $h_i$ на заданную фиксированную константу $\Delta$. Алгоритм останавливается, когда все $g_i = 0$ или достигнута заданная точность решения.

Очень схожий по идее с описанными выше точный алгоритм, названный ZEBRA, предложен в [68], он основан на альтернативной формулировке (11)–(15). Заметим, что переменные $z_{kj}$ для некоторого фиксированного $j$ принимают значение 1 для всех первых $k$, для которых не существует медианы, находящейся на более близком расстоянии, и 0 для всех остальных $k$, т. е. $z_j = (1, \ldots, 1, 0, \ldots, 0)$. Тем самым если в оптимальном решении $(y^*, z^*_{kj})$ задачи (11)–(15), содержащей лишь часть переменных $z_{kj}$, $2 \leqslant k \leqslant t_j$, $j = 1, \ldots, m$, $t_j \leqslant G_j$, и соответствующих им ограничений (12), переменные $z^*_{t_j j}$ равны 0, то это решение оптимально и в исходной задаче. Это же свойство справедливо и для линейной релаксации. В работе предлагается метод генерации строк и столбцов для поиска решений в линейной релаксации задачи (11)–(15), состоящий в решении последовательности редуцированных релаксированных задач с последовательным увеличением числа переменных $z_{kj}$ и соответствующих им ограничений до тех пор, пока все $z_{t_j j}$ в оптимальном решении

не примут значение 0. Для поиска целочисленного решения данный метод был интегрирован в метод ветвей и границ с ветвлением по дробной переменой $y_i$. Предложенный алгоритм позволил найти точные решения для задач, содержащих до 90000 элементов данных, однако он показал эффективность только для случая относительно больших значений $p$.

Помимо точных методов для задачи о $p$-медиане были разработаны разнообразные прямо-двойственные эвристики, преимуществом которых перед алгоритмами, основанными на локальном поиске, такими, как PAM, является тот факт, что кроме допустимого решения они находят и двойственную оценку, которая может служить своего рода «сертификатом субоптимальности» найденного решения. Одна из первых таких эвристик применительно к области кластеризации была предложена в [69]. Она похожа на алгоритм из [33] и состоит в решении двойственной задачи Лагранжа, полученной за счёт ослабления ограничений (3). Алгоритм включает эвристики, основанные на иерархических алгоритмах кластеризации, для поиска начального допустимого решения, а информация, полученная в ходе решения двойственной задачи субградиентным алгоритмом, используется для поиска допустимых решений, дающих лучшее значение целевой функции.

Использование лагранжевых релаксаций для целочисленных задач имеет преимущество, поскольку полученная таким образом двойственная оценка в худшем случае совпадает с оценкой, получаемой с помощью линейной релаксации.

Помимо ослабления ограничений (3) возможен другой тип релаксации, в которой дополнительно ослабляются ограничения (5). Однако в обоих случаях имеет место так называемое свойство целочисленности [70], гарантирующее, что лагранжева релаксация не даёт двойственной оценки лучшей, чем получаемая с помощью линейной релаксации.

В [71] предложен метод частичных лагранжевых релаксаций, который предполагает помимо ослабления ограничений (3) и (5) добавлять в двойственную задачу неравенства $\sum\limits_{i=1}^{m} x_{ij} \leqslant 1$ и $\sum\limits_{i=1}^{m} y_i \leqslant p$ соответственно. Такие формы двойственных задач более трудны для решения, однако позволяют найти лучшую двойственную оценку. В статье был предложен алгоритм, состоящий в последовательном решении трёх релаксированных задач с ослабленными ограничениями (3), (5), в которых данные ограничения последовательно добавляются в виде неравенств, что в итоге позволяет найти оптимальные решения задачи.

Наиболее эффективными современными прямо-двойственными эвристиками являются прямо-двойственный декомпозиционный локальный поиск с чередующимися окрестностями (VNDS) [27] и алгоритм, предложенный в [72]. В первом применяется комбинация редуцированного

и декомпозиционного локального поиска с чередующимися окрестностями для нахождения хорошего допустимого решения задачи. Значение целевой функции в этой точке используется затем для подсчёта шага субградиентного алгоритма при решении двойственной задачи линейной релаксации. Это позволяет найти нижнюю оценку оптимального значения исходной задачи и оценить качество полученного решения. В [72] предложен подход, в котором сначала ищется решение лагранжевой релаксации задачи относительно ограничений (3), затем информация о найденных значениях двойственных переменных используется в так называемой ядровой эвристике, идея которой состоит в фиксировании части переменных. Для задач с небольшим числом кластеров $p$ предложенный алгоритм был дополнен агрегирующей эвристикой. Алгоритмы из [27,72] позволяют относительно быстро находить близкие к оптимальным решения в задачах кластеризации большой размерности, содержащих до 100000 элементов данных. В ходе вычислительных экспериментов была продемонстрирована малая относительная погрешность между верхней и нижней оценками оптимального значения в задачах такой размерности.

Наконец, поскольку задачи размещения (в частности, задача о $k$-медоидах) являются одними из базовых комбинаторных задач, для их решения реализованы, вероятно, все наиболее эффективные к настоящему времени метаэвристики [64]. Несмотря на то, что большинство из них основано на применении процедуры локального поиска, что делает их схожими с PAM и другими алгоритмами кластеризации, современные метаэвристики зачастую не используются для сравнения в рамках тестирования алгоритмов кластеризации. Как и алгоритм PAM, метаэвристики не позволяют получить оценку качества найденного решения, а также наследуют многие присущие ему недостатки, связанные, например, с вычислительной сложностью локального поиска для задач большой размерности (в особенности с большим числом элементов данных, кластеров и признаков), а также со сложностью вычисления и/или хранения матрицы попарных расстояний. Наиболее эффективная стратегия при реализации метаэвристик в задачах кластеризации большой размерности состоит в применении процедур агрегирования данных [73, 74].

Дискретные задачи размещения являются весьма популярным объектом исследования в области построения приближённых алгоритмов. Отметим, что в большинстве работ предполагается, что расстояния заданы метрикой, чаще всего евклидовой, либо есть другие дополнительные предположения относительно расстояний, например, пространство имеет фиксированную размерность. Первый приближённый алгоритм с константной оценкой точности 20/3 для задачи о $k$-медоидах, основанный на вероятностном округлении, был предложен в [15]. В широко известной работе [75] предложен 4-приближённый алгоритм, основанный

на использовании лагранжевой релаксации для ослабления ограничения на число кластеров. Таким образом, задача может быть сведена к простейшей задаче размещения, решение которой может быть найдено с помощью прямо-двойственного алгоритма.

Как отмечено выше, на протяжении более чем десяти лет лучшим приближённым алгоритмом был локальный поиск, в котором на каждом шаге заменяются $l$ медиан, с коэффициентом аппроксимации $3+\varepsilon$. Этот результат улучшен в [76], где был предложен алгоритм с коэффициентом $1+\sqrt{3}+\varepsilon$, который впоследствии был улучшен в [77], где предложен приближённый алгоритм с лучшим на сегодняшний день коэффициентом аппроксимации $2{,}675+\varepsilon$. В [78] показано, что не существует $\left(1+\frac{2}{e}-\varepsilon\right)$-приближённого алгоритма для задачи о $p$-медиане, если не NP $\subseteq$ DTIME$(m^{O(\log\log m)})$, где DTIME определяет класс, включающий множество задач, решаемых с помощью детерминированной машины Тьюринга за заданное время.

Как было отмечено в ранних исследованиях, оптимальное решение линейной релаксации задачи о $k$-медоидах в случае задач небольшой размерности (при использовании ограничений Балинского) во многих случаях является целочисленным. В работах [79, 80] исследовались условия, при которых оптимальные решения задачи кластеризации и её линейной релаксации совпадают. В частности, в [79] показано, что в случае если заданы $k$ единичных шаров в $n$-мерном евклидовом пространстве ($n>2$), причём попарные расстояния между центрами равны по меньшей мере 3,75 и в каждом шаре случайно и независимо выбраны $s$ точек согласно некоторому сферически симметричному распределению, то существуют такие значения $s$ и $k \geqslant 2$, что оптимальное решение линейной релаксации единственно и совпадает с оптимальным решением исходной задачи с вероятностью, превышающей $1-\frac{4k}{s}$, в случае если несхожесть между точками измеряется квадратом евклидова расстояния. Более того, точки каждого шара присоединяются к отдельному кластеру. В [80] этот результат был улучшен и было показано, что для любого $\varepsilon>0$ при условии, что центры шаров отдалены на расстояние $\Delta > 2+\varepsilon$, существует достаточно большое число $s$ такое, что оптимальные решения линейной релаксации и исходной задачи совпадают с высокой вероятностью. В частности, показано, что прямо-двойственный алгоритм из [75] при выполнении тех же условий гарантировано находит имеющиеся кластеры, а алгоритм PAM с высокой вероятностью не даёт правильного разбиения.

## 4. Параллельные и распределённые алгоритмы кластеризации

Развитие современной вычислительной техники и информационных технологий привели к накоплению больших объёмов разнородных данных, анализ которых представляет собой важную, но весьма сложную задачу. Отметим, что даже такой базовый инструмент анализа данных, как кластеризация, зачастую бывает осложнён или невозможен во многих подобных случаях. Поскольку сложность задачи кластеризации растёт быстро с ростом числа элементов данных, размерности пространства признаков и числа кластеров, даже такой общепризнано быстрый алгоритм, как $k$-средних, может требовать нескольких часов для выполнения одной итерации.

Вычислительная сложность жадного алгоритма и алгоритмов локального поиска в задаче о $k$-медоидах, а также большой объём потребляемой памяти делают их применение в случае больших данных затруднительным, в то время как подход, основанный на отборе выборок (CLARA, CLARANS), не позволяет найти хорошие допустимые решения.

В связи с этим в последние годы наметилось направление исследований, связанное с разработкой эффективных реализаций и адаптаций алгоритмов машинного обучения для задач большой размерности. Один из популярных подходов состоит в разработке параллельных и распределённых алгоритмов (с использованием MPI, MapReduce, GPU) как основанных на непосредственном распараллеливании последовательных алгоритмов, так и с применением различных вариантов стратегий декомпозиции области поиска и отбора выборок.

Что касается дискретных задач размещения, то многие начальные исследования в этом направлении были посвящены разработке параллельных реализаций различных метаэвристик для задачи о $p$-медиане. Так, например, в [81–83] предложены параллельные варианты локального поиска с чередующимися окрестностями (VNS) и распределённого поиска, предполагающие различные стратегии распараллеливания: параллельный мультистарт (всей метаэвристики), распараллеливание локального поиска путём разделения окрестности текущего решения между параллельными процессорами, мультистарт локального поиска (например, параллельный запуск нескольких локальных поисков с разных начальных решений текущей окрестности в VNS) и т. д. Другие работы, посвящённые распараллеливанию локального поиска, включают в себя GPU реализацию [84], а также параллельный кооперативный алгоритм, комбинирующий локальный поиск и алгоритм имитации отжига [85]. Большинство стратегий реализации параллельных метаэвристик подробно описаны в [86].

Отметим, что такие подходы во многом направлены на диверсификацию или интенсификацию локального поиска за счёт стратегии мультистарта или параллельного просмотра окрестности текущего решения. Целью большого числа таких работ является ускорение работы алгоритмов для относительно небольших задач кластеризации.

Другая стратегия реализации параллельного локального поиска была исследована в [87]. Она предполагает пропорциональное разделение элементов данных и искомых кластеров между параллельными процессорами с помощью так называемой кривой Гильберта. На каждом подмножестве элементов данных затем запускается локальный поиск с запретами. Найденные «частичные» решения объединяются за счёт поиска для каждого элемента ближайшей к нему медианы.

Для задачи о $p$-медиане был также предложен ряд параллельных приближённых алгоритмов, анализ и оценка эффективности которых зависит от выбранной модели параллельных вычислений. Так, например, в [88] для метрического случая предложены первые EREW-PRAM-алгоритмы. Напомним, что PRAM представляет собой параллельный компьютер с общей памятью, причём каждый процессор может получить доступ к любому биту за константное время. Таким образом, предложенные алгоритмы работают за полилогарифмическое время с использованием полиномиального числа процессоров. В работе предложены параллельные $(6+\varepsilon)$- и $(3+\varepsilon)$-приближённые алгоритмы, являющиеся параллельными вариантами жадного алгоритма из [78] и прямо-двойственного алгоритма из [75]. В [89] предложен параллельный $O(\log p)$-приближённый алгоритм, относящийся к классу RNC (рандомизированный эквивалент класса NC), для произвольного числа кластеров $p$.

Помимо параллельных алгоритмов для абстрактных моделей, имитирующих суперкомпьютеры с общей памятью, в литературе также представлены так называемые распределённые приближённые алгоритмы. Так, в [90] был разработан распределённый алгоритм для $k$-машинной модели ($k$-machine model), которая представляет собой абстрактную модель, имитирующую основные свойства таких систем, как Google Pregel и Apache Giraph, созданных для обработки графов большой размерности. Она представляет собой модель распределённых вычислений, в которой $k$ машин взаимодействуют за счёт синхронной передачи сообщений. Выполнение алгоритма состоит из раундов, в рамках которых каждая машина выполняет локальные вычисления и затем отправляет сообщения остальным машинам. Отметим, что каждое сообщение предполагается небольшим по размеру, состоящим из $O(\log m)$ битов. В начале работы все входные данные (элементы данных) случайно распределяются между машинами. В статье были получены нижние оценки на число

раундов для задачи о $p$-медиане: $\alpha$-приближённый алгоритм для $k$-машинной модели для любого $\alpha = \mathrm{poly}(m)$ может быть выполнен не менее чем за $\Omega\big(\frac{m}{k}\,\mathrm{poly}\,\big(\log\frac{m}{k}\big)\big)$ раундов. С использованием этого результата был предложен $(6+\varepsilon)$-приближённый вероятностный алгоритм с временем работы $O\big(\frac{m}{k}\,\mathrm{poly}\,\big(\log\frac{m}{k}\big)\big)$ раундов.

Поскольку размерность современных данных зачастую не позволяет обработать их с использованием одной ЭВМ, в последние годы особую популярность получили модели распределённых вычислений, из которых наиболее популярной является MapReduce, разработанная Google. Модель лежит в основе многих систем распределённых вычислений, например, упомянутой выше Apache Giraph. В рамках MapReduce предполагается наличие заданного числа вычислительных узлов, связанных между собой коммуникационной сетью. Каждый узел имеет ограниченный объём памяти. Выполнение алгоритма разбито на раунды, в рамках которых выполняется два шага Map и Reduce. На первом шаге главный узел распределяет входные данные между остальными узлами. На шаге Reduce каждый узел обрабатывает полученные данные независимо от других узлов. Результаты работы на данном шаге являются либо окончательными, либо подаются на вход следующего раунда.

В [91] был предложен класс сложности $\mathcal{MRC}^l$, целью которого было учесть особенности MapReduce и алгоритмов, реализованных с использованием этой модели. В рамках $\mathcal{MRC}^l$ предполагается, что шаги Map и Reduce выполняются за полиномиальное от длины первоначального входа время. Обозначим через $N$ длину входных данных. В этом случае алгоритм принадлежит $\mathcal{MRC}^l$, если для некоторого $\varepsilon > 0$ он использует не больше чем $N^{1-\varepsilon}$ узлов, размер памяти которых не превосходит $N^{1-\varepsilon}$, и возвращает правильный ответ с вероятностью не менее $3/4$ за $O(\log^l N)$ раундов.

В [92] предложен $\mathcal{MRC}^0$-алгоритм с константной оценкой точности, который использует не более чем $O(p^2 m^\delta)$ памяти каждого узла и выполняется за $1/\delta$ раундов, где $\delta > 0$ — заданная константа. Идея алгоритма состоит в отборе репрезентативной выборки элементов данных. Для каждого элемента выборки определяется вес, зависящий от того, для скольких элементов данных, не представленных в выборке, данный элемент будет ближайшим. Для получившейся взвешенной выборки применяется какой-либо алгоритм поиска решений в задаче о $p$-медиане (например, алгоритм Тейтца и Барт). Показано что полученный алгоритм кластеризации $(10\alpha+3)$-приближённый, где $\alpha$ — константа точности применяемого алгоритма для решения задачи о $p$-медиане на взвешенной выборке. Отметим, что такой алгоритм отбора выборок эффективен только для задач с относительно небольшим числом кластеров, в противном случае выборка совпадает с исходными данными.

Помимо приближённых распределённых алгоритмов в литературе также предложено множество распределённых реализаций базовых алгоритмов кластеризации, таких как CLARA, метод Маранцаны, жадный алгоритм и т. д. Отметим, что сама идея CLARA позволяет достаточно естественно реализовать его параллельно. Наиболее простой вариант состоит в одновременной обработке нескольких выборок, для чего могут быть использованы любые модели и интерфейсы параллельных вычислений, например, MapReduce [93] или MPI [94].

Ряд работ посвящён разработке распределённых версий алгоритма Маранцаны с использованием MapReduce и Spark. Например, в [95] реализована близкая к [93] стратегия, в которой алгоритм Маранцаны применяется параллельно к небольшим выборкам. Поскольку наиболее вычислительно затратным шагом алгоритма Маранцаны является поиск новой медианы в каждом кластере (решение задачи о 1-медиане), причём такой поиск осуществляется независимо для каждого кластера, наиболее естественная стратегия предполагает решение этой задачи параллельно [95–97]. В последней работе применялся описанный выше метод инициализации [52]. В [96] исследовался как прямой поиск медианы кластера, так и приближённый с использованием выборки элементов кластера. В [98] предложен алгоритм, в котором на начальном этапе происходит параллельный отбор выборок, каждая из которых разбивается на кластеры с помощью PAM. На втором этапе лучший из найденных по значению целевой функции наборов медиан используется как начальное решение для параллельного алгоритма Маранцаны, в котором поиск новой медианы происходит параллельно. Для быстрого поиска медианы кластера используется его случайное разбиение на подмножества. Наконец, в [99] задача о $p$-медиане представлена как задача максимизации субмодулярной функции, для которой предложен распределённый вариант жадного алгоритма. Его идея состоит в разбиении исходных данных на подмножества, в каждом из которых находятся $k$ представителей. Полученные в каждом подмножестве решения затем объединяются в одно.

Как можно видеть, разработка параллельных, распределённых версий приближённых и базовых алгоритмов кластеризации (в большинстве своём основанных на локальном поиске) зачастую приводит к существенному уменьшению качества получаемых решений. Несмотря на то, что представленные алгоритмы могут быть успешно применены для обработки данных, содержащих миллионы и даже сотни миллионов объектов, они в лучшем случае находят лишь локально оптимальное решение. Более того, большинство распределённых алгоритмов основаны на отборе случайных выборок, а потому часто становятся неэффективными или неприменимыми для задач с большим числом кластеров. Вычисление

попарных расстояний и/или необходимость хранения матрицы расстояний составляют основную вычислительную трудность алгоритмов.

В [94] предложен распределённый параллельный прямо-двойственный алгоритм для задачи о $p$-медиане, основанный на подходе из [72] и реализованный, в отличие от предыдущих работ, с помощью MPI-OpenMP. Ключевая особенность алгоритма состоит в том, что матрица расстояний аппроксимируется с помощью так называемого метода $l$-ближайших соседей и хранится распределённо. Это позволяет избежать пересчёта попарных расстояний между элементами данных в ходе работы алгоритма. Алгоритм предполагает вычисление двойственных оценок оптимального значения с помощью распределённого алгоритма генерации столбцов и субградиентного алгоритма. Прямые оценки оптимального значения затем находятся с помощью распределённой ядровой эвристики. В отличие от распределённых алгоритмов кластеризации, основанных на локальном поиске, предложенный подход позволяет найти двойственную оценку. В ходе экспериментов найдены близкие к оптимальным решения для задач кластеризации, содержащих до 12 миллионов объектов и тысяч кластеров.

В [100] реализована параллельная лагранжева эвристика для так называемой дискретной упорядоченной медианной задачи (discrete ordered median problem), частным случаем которой является задача о $p$-медиане. Наконец, распределённый грид-решатель для задачи о $p$-медиане, основанный на точном методе, предложенном в [66], был реализован в [101].

## 5. Некоторые обобщения базовых дискретных задач размещения и алгоритмов кластеризации

Этот раздел посвящён некоторым обобщениям базовых задач размещения и алгоритмам их решения, как непосредственно возникающим в области машинного обучения, так и имеющим вполне определённые приложения в этой области.

Основным требованием большинства алгоритмов кластеризации, основанных на поиске решений в задачах размещения (PAM, $k$-средних, CLARANS и т. д.), является фиксированное число кластеров, заданных в качестве входного параметра. Отметим, что задача определения числа кластеров сама по себе весьма нетривиальна и для её решения предложен ряд подходов (например, поиск так называемых силуэтов элементов данных).

Тем не менее, в литературе предложены алгоритмы кластеризации, предполагающие динамический поиск числа кластеров. Так, например, в [102] предложен вариант алгоритма Маранцаны, в котором число кластеров постепенно увеличивается до тех пор, пока расстояния между элементами данных и ближайшим медоидом не станет меньше заданного

порогового значения, для чего используется процедура динамического назначения веса для каждого элемента. Помимо этого для поиска кластера, к которому присоединяется каждый элемент, алгоритм использует информацию о дисперсии в направлении отрезка, соединяющего пару медиан. В [103] предложен вариант PAM, в котором число кластеров определяется с помощью средних значений силуэтов, вычисленных для разных значений кластеров.

Иной подход к поиску представителей кластеров был изучен в [104]. В работе ставилась задача поиска медоидов, причём их число предполагалось ограниченным регуляризующим параметром. Если предположить, что переменная $x_{ij} \in [0, 1]$ задаёт вероятность того, что элемент $i$ является представителем для $j$, то задача поиска представителей может быть записана как следующая задача (row-sparsity regularized trace minimization problem):

$$\begin{aligned} &\min \sum_{i=1}^{m} \sum_{j=1}^{m} d_{ij} x_{ij} + \lambda \sum_{i=1}^{m} I(\|x_i\|_q), \\ &\sum_{i=1}^{m} x_{ij} = 1, \quad x_{ij} \geqslant 0, \quad i, j = 1, \ldots, m, \end{aligned} \tag{21}$$

где $I(\cdot)$ — индикаторная функция, принимающая значение 0, если её аргумент равен нулю, и 1 в противном случае; $\|\cdot\|_q$ — $l_q$-норма, $x_i$ — строка матрицы переменных $X$, а $\lambda$ — регуляризующий параметр. Можно видеть, что второе слагаемое определяет число ненулевых строк в матрице $X$, т. е. число медоидов, а параметр $\lambda$ задаёт «компромисс» между числом медоидов и суммарным расстоянием между объектами и представителями.

Вместо исходной задачи (21) в работе [104] предлагается искать решения её выпуклой релаксации:

$$\begin{aligned} &\min \sum_{i=1}^{m} \sum_{j=1}^{m} d_{ij} x_{ij} + \lambda \sum_{i=1}^{m} \|x_i\|_q, \\ &\sum_{i=1}^{m} x_{ij} = 1, \quad x_{ij} \geqslant 0, \quad i, j = 1, \ldots, m. \end{aligned}$$

Авторами проведено теоретическое исследование влияния параметра $\lambda$ на число кластеров, а также представлены численные результаты.

Стоит отметить, что задача (21) представляет собой не что иное, как простейшую задачу размещения с одинаковой для всех предприятий стоимостью открытия $f_i$ (см. задачу (7)–(10)). Применение простейшей задачи размещения для задач кластеризации выглядит вполне естественным, поскольку она не предполагает, что число кластеров фиксировано.

Вместо этого их число может быть задано с помощью стоимости открытия предприятия в том или ином пункте. Помимо представленной выше работы, варианты простейшей задачи размещения для случая кластеризации при наличии выбросов были исследованы в [105].

В [106] предложен известный алгоритм кластеризации, названный affinity propagation, получивший чрезвычайно широкое распространение в прикладных областях, прежде всего в био- и химинформатике. Алгоритм направлен на поиск медоидов (представителей) элементов данных, однако также не предполагает, что их число является входным параметром. На вход алгоритм получает матрицу схожестей объектов (не расстояний), причём $d_{ii}$ задаёт для каждого объекта вес, определяющий его вероятность быть выбранным в качестве представителя. Идея алгоритма состоит в передаче двух типов сообщений между элементами данных. Например, сообщение $r(i,k)$ передаётся от $i$ к возможному представителю $k$ и отражает аккумулированное свидетельство того, насколько точка $k$ подходит в качестве представителя для $i$ (принимая во внимание других кандидатов). Сообщение $a(i,k)$ передаётся от возможного представителя $k$ к точке $i$ и отражает аккумулированное свидетельство того, насколько подходящим для $i$ будет выбрать $k$ в качестве представителя. Процедура обмена сообщениями останавливается после фиксированного числа итераций, либо когда изменения в сообщениях оказываются меньше порогового значения. Авторы протестировали свой алгоритм на задачах кластеризации изображений, текстов и детектировании генов в микроматричных данных.

Однако в [107] (комментариях к [106]) проведено численное сравнение алгоритма affinity propagation с алгоритмом Тейтца и Барт, в котором показано, что последний находит в целом лучшие решения. Наконец, в [108] показано, что алгоритм affinity propagation является эвристикой для простейшей задачи размещения, в которой расстояния заменены «схожестями» (т. е. задача определяется на максимум). Несмотря на то, что простейшая задача размещения может быть решена точно или приближённо для данных достаточно большой размерности, алгоритм affinity propagation остаётся популярным в приложениях. Более того, рядом авторов предложены его модификации [109].

Схожий тип обобщения задачи о $p$-медиане был предложен в [110], в котором число кластеров $p$ является ещё одной целочисленной переменной задачи, в то время как в целевой функции присутствует дополнительное слагаемое — выпуклая нелинейная функция от числа кластеров $\phi(p)$. Отметим, что в случае линейной функции $\phi(p)$ задача фактически аналогична задаче (21). Авторами была предложена эвристика, представляющая по сути метод бисекции по $p$, в котором на каждом шаге производится решение параметризованной простейшей задачи размещения

с помощью двойственного алгоритма DUALOC. Точный метод решения, основанный на представленной эвристике, предложен в [111]. Наконец, в [112] была приведена прямо-двойственная эвристика, основанная на релаксации Лагранжа, причём рассматривался случай как выпуклой, так и вогнутой монотонно возрастающей функции $\phi(p)$.

Во многих ситуациях все попарные расстояния (схожести) между объектами не могут быть вычислены ввиду высокой сложности, больших финансовых затрат или невозможности получить доступ ко всем парам объектов. Классическим примером является определение схожести белков, которое выполняется с помощью вычислительно сложного метода выравнивания последовательностей. Похожая ситуация возникает в случаях, когда схожесть объектов может быть определена только с привлечением человека, что, очевидно, также затратно.

В связи с этим в последние годы получили популярность так называемые активные алгоритмы кластеризации, в которых расстояния между объектами не известны заранее, но запрашиваются по мере необходимости для подмножества так называемых информативных элементов данных. Запрашиваться могут как расстояния между парой объектов, так и между элементом и всеми остальными объектами.

В [113] был предложен активный алгоритм кластеризации на основе задачи о $k$-медоидах, в котором расстояния между элементами задаются верхними оценками, полученными с помощью неравенства треугольника для элементов с уже известными попарными расстояниями. В статье предложена процедура выбора информативных элементов, основанная на иерархическом разделении объектов на подгруппы до тех пор, пока количество элементов в них не станет меньше заданного порога. Запрос на подсчёт попарных расстояний выполняется только для элементов в подгруппах самого нижнего уровня. После этого оценки расстояний обновляются и данные разбиваются на кластеры с помощью алгоритма Маранцаны.

Другим интересным вариантом задачи кластеризации является так называемая кластеризация с частичным привлечением учителя (semi-supervised clustering), в рамках которой имеется некоторая дополнительная информация об объектах, например, метки классов или информация об объектах, которые должны принадлежать к одному кластеру (must-link) или разным кластерам (cannot-link). В [114] предложен вариант задачи о $p$-медиане для кластеризации с частичным привлечением учителя, в которую интегрированы такого рода ограничения. Для поиска решений в такой задаче предложен алгоритм локального поиска с чередующимися окрестностями. Отметим, что must-link ограничения могут быть легко выполнены, если соответствующие пары рассматривать как один элемент. Схожие варианты простейшей задачи размещения были

предложены и исследованы в [115, 116]. В [115] рассматривалась задача, где имеются пары потребителей, которые не могут быть присоединены к одному предприятию (кластеру), в то время как в [116] предполагалось, что определённые пары клиентов должны присоединяться одновременно к нескольким предприятиям, причём для такой пары клиентов должно существовать хотя бы одно общее предприятие.

В прикладных задачах кластеризации, особенно возникающих в области вычислительной биологии и биоинформатики, для одних и тех же элементов данных имеется несколько разнородных наборов признаков, полученных из разных источников (multisource clustering). Один из подходов к анализу такого рода данных состоит в объединении всех признаков в один вектор. Однако такой подход не всегда позволяет правильно учесть разнородные признаки, специфические для каждого набора. Один из возможных подходов состоит в применении многокритериальных моделей кластеризации. В [117, 118] исследована бикритериальная задача о $p$-медиане в приложении к задаче кластеризации линий раковых клеток. Другой подход к кластеризации на основе двух наборов признаков основан на применении так называемых дискретных задач размещения с предпочтениями клиентов [119–121]. Такого рода задачи основаны на поиске представителей кластеров для минимизации суммарных расстояний по одному набору признаков, в то время как присоединение объектов к кластерам осуществляется на основе расстояний относительно другого набора. В [118] исследовано приложение варианта задачи о $p$-медиане с предпочтениями клиентов для задачи кластеризации линий раковых клеток и предложен точный метод её решения, основанный на алгоритме из [122].

В [123] исследовалось обобщение задачи о $p$-медиане, в котором вместе с наилучшим набором из $p$ представителей происходит одновременный поиск наилучшего подмножества из $q$ наиболее релевантных признаков. Таким образом, задача объединяет кластеризацию и отбор признаков. Для этого вводятся величины $d_{ijk}$, равные расстоянию между элементами $i$ и $j$ относительно признака $k$, и переменные $z_k$, принимающие значение 1, если признак $k$ выбран как релевантный. Упрощённый вариант представленной задачи, так называемая задача об отборе $q$ признаков, в котором представители кластеров предполагаются заданными, был исследован в [124]. Для поиска решений авторами предложены алгоритмы локального поиска по типу алгоритмов Маранцаны и Тейтца и Барт.

Оригинальные модели кластеризации, основанные на задачах размещения, не предполагают никаких ограничений на размеры кластеров, в связи с этим в ряде случаев найденные кластеры несбалансированны. Постановки, учитывающие подобного рода ограничения, известны как задачи размещения с ограничениями на мощности производства. Одна

из первых таких модификаций простейшей задачи размещения была исследована в [125] и с тех пор получила широкое внимание в области исследования операций. Как полагают, задача о $p$-медиане с ограничениями на мощность производства не стала столь же популярной. В [126] исследовалась подобного рода модификация задачи о $p$-медиане, постановка которой аналогична (2)–(6), за исключением дополнительных условий $\sum_{j=1}^{m} q_j x_{ij} \leqslant Q_i$, $i = 1, \ldots, m$, задающих ограничения на вес искомых кластеров, где $q_j$ — вес элемента данных, а $Q_i$ — размер кластера $i$. В статье предложены эвристики локального поиска, близкие к алгоритму Маранцаны, а также прямо-двойственная эвристика на основе лагранжевой релаксации.

В [127] исследовались как непрерывные, так и дискретные постановки задачи кластеризации с ограничением на размер кластеров, а также предложен ряд адаптаций классических эвристик. К настоящему моменту для таких вариантов задач кластеризации предложено огромное число как точных, так и эвристических алгоритмов [128–132], в том числе для прикладных задач, возникающих, например, в биоинформатике [133].

Поскольку данные большой размерности представляют существенную трудность для традиционных алгоритмов, помимо применения распределённых вычислений возможно также использование подхода, основанного на бинарном кодировании исходных данных и адаптации известных алгоритмов для их анализа. Так, например, в [134] gist-дескрипторы большого числа изображений были преобразованы в булевы векторы меньшей размерности с помощью методов бинарного хэширования LSBC или LSH (locality-sensitive hashing), что позволило существенно сократить размер исходных данных. Отметим, что идея того же LSH состоит в том, чтобы построить бинарные хэши элементов таким образом, чтобы схожие объекты оказались в одной «корзине». Далее, для их кластеризации в статье использовался алгоритм Маранцаны, реализованный на GPU, в котором расстояния между элементами данных и медоидами задавались расстоянием Хэмминга. Быстрая реализация алгоритма Маранцаны и алгоритма Ллойда для бинарных кодов также представлена в [135].

Задача о $p$-медиане и алгоритмы её решения могут быть использованы, как и алгоритм $k$-средних, в качестве методов векторного квантования — наиболее простого подхода к снижению размерности исходных данных. В этом случае каждый элемент данных может быть представлен или центром кластера, к которому он присоединён, или вектором расстояний до всех центров. Другой подход к снижению размерности с помощью задачи о $p$-медиане исследован в [136], где метод локального

поиска с чередующимися окрестностями применялся для поиска «разрезов» матрицы зависимых переменных в разрезанной обратной регрессии (sliced inverse regression).

Наконец, задачи размещения могут быть использованы в качестве инструментов в методах глубокого обучения, например, глубокого извлечения метрики (metric learning). Отметим, что стандартные способы определения «несхожестей» между объектами с помощью евклидовой, манхэттенской и др. метрик не всегда приемлемы для многих практических данных, поскольку не всегда адекватно оценивают схожие и различные элементы. В связи с этим возникает задача поиска подхода к определению расстояний для некоторого набора данных таким образом, чтобы похожие объекты были как можно более близки и в то же время несхожие между собой объекты — как можно более далеки. Одним из характерных примеров приложения методов извлечения метрики является кластеризация и классификация изображений, а также идентификация и верификация лиц. Отметим, что описанное выше хэширование данных также можно рассматривать как подход к извлечению расстояний.

Поскольку методы извлечения метрики часто предполагают, что исходные элементы данных отображаются в пространство меньшей размерности, их также можно рассматривать как методы снижения размерности задачи. Помимо классических подходов, например, основанных на обучении расстояний Махаланобиса [137], в последние годы особую популярность получили методы, использующие глубокие нейронные сети. Их идея состоит в обучении глубоких сетей так, чтобы отобразить исходные элементы в пространство меньшей размерности, чтобы минимизировать расстояния между точками (часто пары или тройки) в соответствии с некоторой функцией потерь. Один из наиболее эффективных методов глубокого извлечения метрики, предложенный в [138], предполагает использование функции потерь, которая направлена на одновременное уменьшение суммарных расстояний между точками и ближайшим к ним медоидом и увеличение расстояния между кластерами за счёт использования NMI-метрики (часто применяемой в кластерном анализе для оценки качества кластеризации).

Для вычисления градиента и реализации стохастического градиентного спуска в рамках обучения нейронной сети необходимо решить подзадачу, сводящуюся фактически к решению варианта задачи о $p$-медиане. Для её решения применяется жадный алгоритм, после чего полученное решение улучшается за счёт локального поиска внутри каждого кластера. С помощью сети Inception авторами продемонстрирована высокая эффективность предложенного подхода в задачах кластеризации и поиска ближайших соседей.

## Заключение

В данном обзоре было кратко прослежено, каким образом дискретные задачи размещения, зачастую наиболее базовые, нашли применение в области кластеризации и анализа данных. В частности, рассмотрено, каким образом классические эвристики для решения подобных задач получили приложения в современных методах машинного обучения. Помимо этого, был рассмотрен ряд обобщений классических дискретных задач размещения и алгоритмов, возникающих непосредственно в области машинного обучения, либо имеющих прямое приложение в этой области. Как отмечено в обзоре, огромное число современных алгоритмов кластеризации фактически основаны на классических эвристиках, предложенных для дискретных задач размещения более пятидесяти лет назад. Алгоритмы, рассмотренные в этом обзоре, были классифицированы как обобщения той или иной классической эвристики. Несмотря на большую популярность подобного рода моделей, многие результаты, полученные в различных сообществах, зачастую остаются малоизвестными другим исследователям, прежде всего применяющим полученные результаты на практике. Это в том числе связано с отсутствием доступных широкому кругу исследователей открытых программных реализаций многих наиболее эффективных алгоритмов.

Основные современные направления исследований в данной области связаны с разработкой различных обобщений задач размещения, в том числе учитывающих особенности задач, возникающих в области машинного обучения, а также разработкой эффективных точных и приближённых алгоритмов для данных большой размерности. Последнее представляется особенно актуальным, поскольку повышение эффективности многих современных алгоритмов зачастую связано с серьёзной потерей качества получаемых решений. Помимо применения параллельных и распределённых вычислений, а также отбора случайных выборок, в том числе гарантирующих получение приближённого оптимального решения, современным, хотя и не так широко представленным, направлением является разработка техник, объединяющих, например, алгоритмы кластеризации с техниками построения бинарных хэшей (binary hash learning) и извлечения метрики.

## ЛИТЕРАТУРА

1. **Cooper L.** Location-allocation problems // Oper. Res. 1963. Vol. 11, No. 3. P. 331–343.
2. **Cooper L.** Heuristic methods for location-allocation problems // SIAM Rev. 1964. Vol. 6, No. 1. P. 37–53.
3. **Plastria F.** The Weiszfeld algorithm: Proof, amendments, and extensions // Foundations of Location Analysis. New York: Springer, 2011. P. 357–389.

4. **Lloyd S.** Least squares quantization in PCM // IEEE Trans. Inf. Theory. 1982. Vol. 28, No. 2. P. 129–137.
5. **Forgy E. W.** Cluster analysis of multivariate data: Efficiency versus interpretability of classifications // Biometrics. 1965. Vol. 21, No. 3. P. 768—769.
6. **Banerjee A., Merugu S., Dhillon I. S., Ghosh J.** Clustering with Bregman divergences // J. Mach. Learn. Res. 2005. Vol. 6, No. 58. P. 1705–1749.
7. **MacQueen J.** Some methods for classification and analysis of multivariate observations // Proc. 5th Berkeley Symp. Math. Stat. Probab. (Berkeley, USA, June 21–July 18, 1965; Dec. 27, 1965–Jan. 7, 1966). Vol. 1. Berkeley: Univ. California Press, 1967. P. 281–297.
8. **Vinod H. D.** Integer programming and the theory of grouping // J. Am. Stat. Assoc. 1969. Vol. 64, No. 326. P. 506–519.
9. **Balinski M. L.** Integer programming: Methods, uses, computations // Manage. Sci. 1965. Vol. 12, No. 3. P. 253–313.
10. **Efroymson M. A., Ray T. L.** A branch-bound algorithm for plant location // Oper. Res. 1966. Vol. 14, No. 3. P. 361–368.
11. **ReVelle C. S., Swain R. W.** Central facilities location // Geogr. Anal. 1970. Vol. 2, No. 1. P. 30–42.
12. **Hakimi S. L.** Optimal location of switching centers and the absolute centers and medians of a graph // Oper. Res. 1964. Vol. 12, No. 3. P. 450–459.
13. **Hakimi S. L.** Optimum distribution of switching centers in a communication network and some related graph theoretic problems // Oper. Res. 1965. Vol. 13, No. 3. P. 462–475.
14. **Kaufman L., Rousseeuw P. J.** Clustering by means of medoids // Statistical Data Analysis Based on the $L_1$-Norm and Related Methods. Amsterdam: North-Holland, 1987. P. 405–416.
15. **Charikar M., Guha S., Tardos E., Shmoys D. B.** A constant-factor approximation algorithm for the $k$-median problem // J. Comput. Syst. Sci. 2002. Vol. 65, No. 1. P. 129–149.
16. **Balcan M.-F., Blum A., Gupta A.** Approximate clustering without the approximation // Proc. 20th Annu. ACM-SIAM Symp. Discrete Algorithms (New York, USA, Jan. 4–6, 2009). Philadelphia: SIAM, 2009. P. 1068–1077.
17. **Ahmadian S., Norouzi-Fard A., Svensson O., Ward J.** Better guarantees for $k$-means and Euclidean $k$-median by primal-dual algorithms // SIAM J. Comput. 2020. Vol. 49, No. 4. P. FOCS17-97–FOCS17-156.
18. **Kariv O., Hakimi S.** An algorithmic approach to network location problems. II: The $p$-medians // SIAM J. Appl. Math. 1979. Vol. 37, No. 3. P. 539–560.
19. **Megiddo N., Supowit K. J.** On the complexity of some common geometric location problems // SIAM J. Comput. 1984. Vol. 13, No. 1. P. 182–196.
20. **Mahajan M., Nimbhorkar P., Varadarajan K.** The planar $k$-means problem is NP-hard // Theor. Comput. Sci. 2012. Vol. 442. P. 13–21.
21. **Megiddo N., Zemel E., Hakimi S. L.** The maximum coverage location problem // SIAM J. Algebr. Discrete Methods. 1983. Vol. 4, No. 2. P. 253–261.

22. **Aloise D., Deshpande A., Hansen P., Popat P.** NP-hardness of Euclidean sum-of-squares clustering // Mach. Learn. 2009. Vol. 75, No. 2. P. 245–248.
23. **Papadimitriou C. H.** Worst-case and probabilistic analysis of a geometric location problem // SIAM J. Comput. 1981. Vol. 10, No. 3. P. 542–557.
24. **Rosing K. E., ReVelle C. S., Rosing-Vogelaar H.** The $p$-median and its linear programming relaxation: An approach to large problems // J. Oper. Res. Soc. 1979. Vol. 30, No. 9. P. 815–823.
25. **Church R. L.** COBRA: A new formulation of the classic $p$-median location problem // Ann. Oper. Res. 2003. Vol. 122, No. 1–4. P. 103–120.
26. **Cornuejols G., Nemhauser G. L., Wolsey L. A.** A canonical representation of simple plant location problems and its applications // SIAM J. Algebr. Discrete Methods. 1980. Vol. 1, No. 3. P. 261–272.
27. **Hansen P., Brimberg J., Urosević D., Mladenović N.** Solving large $p$-median clustering problems by primal-dual variable neighborhood search // Data Min. Knowl. Discov. 2009. Vol. 19, No. 3. P. 351–375.
28. **Elloumi S.** A tighter formulation of the $p$-median problem // J. Comb. Optim. 2010. Vol. 19, No. 1. P. 69–83.
29. **Maranzana F. E.** On the location of supply points to minimize transport costs // Oper. Res. Q. 1964. Vol. 15, No. 3. P. 261–270.
30. **Teitz M. B., Bart P.** Heuristic methods for estimating the generalized vertex median of a weighted graph // Oper. Res. 1968. Vol. 16, No. 5. P. 955–961.
31. **Hartigan J. A., Wong M. A.** Algorithm AS 136: A $k$-means clustering algorithm // J. R. Stat. Soc. Ser. C. 1979. Vol. 28, No. 1. P. 100–108.
32. **Church R. L., ReVelle C. S.** Theoretical and computational links between the $p$-median, location set-covering, and the maximal covering location problem // Geogr. Anal. 1976. Vol. 8, No. 4. P. 406–415.
33. **Cornuejols G., Fisher M. L., Nemhauser G. L.** Location of bank accounts to optimize float: An analytic study of exact and approximate algorithms // Manage. Sci. 1977. Vol. 23, No. 8. P. 789–810.
34. **Arya V., Garg N., Khandekar R., Meyerson A., Munagala K., Pandit V.** Local search heuristics for $k$-median and facility location problems // SIAM J. Comput. 2004. Vol. 33, No. 3. P. 544–562.
35. **Gupta A., Tangwongsan K.** Simpler analyses of local search algorithms for facility location. Ithaca, NY: Cornell Univ., 2008. (Cornell Univ. Libr. e-Print Archive; arXiv:0809.2554).
36. **Кочетов Ю. А., Пащенко М. Г., Плясунов А. В.** О сложности локального поиска в задаче о $p$-медиане // Дискрет. анализ и исслед. операций. Сер. 2. 2005. Т. 12, № 2. С. 44–71.
37. **Alekseeva E. V., Kochetov Yu. A., Plyasunov A. V.** Complexity of local search for the $p$-median problem // Eur. J. Oper. Res. 2008. Vol. 191, No. 3. P. 736–752.
38. **Whitaker R. A.** A fast algorithm for the greedy interchange for large-scale clustering and median location problems // INFOR. 1983. Vol. 21. P. 95–108.

**39. Whitaker R. A.** Some interchange algorithms for median location problems // Environ. Plann. Ser. B. 1982. Vol. 9, No. 2. P. 119–129.

**40. Densham P. J., Rushton G.** A more efficient heuristic for solving large $p$-median problems // Pap. Reg. Sci. 1992. Vol. 71, No. 3. P. 307–329.

**41. Hansen P., Mladenović N.** Variable neighborhood search for the $p$-median // Locat. Sci. 1997. Vol. 5, No. 4. P. 207–226.

**42. Schubert E., Rousseeuw P. J.** Faster $k$-medoids clustering: Improving the PAM, CLARA, and CLARANS algorithms // Similarity Search and Applications. Proc. 12th Int. Conf. (Newark, USA, Oct. 2–4, 2019). Cham: Springer, 2019. P. 171–187. (Lect. Notes Comput. Sci.; Vol. 11807).

**43. Resende M. G. C., Werneck R. F.** A fast swap-based local search procedure for location problems // Ann. Oper. Res. 2007. Vol. 150, No. 1. P. 205–230.

**44. Feo T. A., Resende M. G. C.** Greedy randomized adaptive search procedures // J. Glob. Optim. 1995. Vol. 6, No. 2. P. 109–133.

**45. Resende M. G. C., Werneck R. F.** A hybrid heuristic for the $p$-median problem // J. Heuristics. 2004. Vol. 10, No. 1. P. 59–88.

**46. Mirzasoleiman B., Badanidiyuru A., Karbasi A., Vondrák J., Krause A.** Lazier than lazy greedy // Proc. 29th AAAI Conf. Artificial Intelligence (Austin, USA, Jan. 25–30, 2015). Palo Alto: AAAI Press, 2015. P. 1812–1818.

**47. Tiwari M., Zhang M. J., Mayclin J., Thrun S.** Bandit-PAM: Almost linear time $k$-medoids clustering via multi-armed bandits // Proc. 34th Conf. Neural Information Processing Systems (Vancouver, Canada, Dec. 6–12, 2020). Red Hook: Curran Assoc., 2020. P. 10211–10222.

**48. Hastie T., Tibshirani R., Friedman J.** The elements of statistical learning: Data mining, inference, and prediction. New York: Springer, 2009.

**49. Park H.-S., Jun C.-H.** A simple and fast algorithm for $k$-medoids clustering // Expert Syst. Appl. 2009. Vol. 36, No. 2, Pt. 2. P. 3336–3341.

**50. Newling J., Fleuret F.** A sub-quadratic exact medoid algorithm // Proc. Mach. Learn. Res. 2017. Vol. 54. P. 185–193.

**51. Bagaria V., Kamath G., Ntranos V., Zhang M., Tse D.** Medoids in almost-linear time via multi-armed bandits // Proc. Mach. Learn. Res. 2018. Vol. 84. P. 500–509.

**52. Paterlini A. A., Nascimento M. A., Jr. C. T.** Using pivots to speed-up $k$-medoids clustering // J. Inf. Data Manage. 2011. Vol. 2, No. 2. P. 221–236.

**53. Kaufman L., Rousseeuw P. J.** Finding groups in data: An introduction to cluster analysis. Hoboken, NJ: Wiley, 2005.

**54. Ng R. T., Han J.** CLARANS: A method for clustering objects for spatial data mining // IEEE Trans. Knowl. Data Eng. 2002. Vol. 14, No. 5. P. 1003–1016.

**55. Newling J., Fleuret F.** $K$-medoids for $K$-means seeding // Proc. 31st Int. Conf. Neural Information Processing Systems (Long Beach, CA, USA, Dec. 4–9, 2017). Red Hook, NY: Curran Assoc., 2017. P. 5201–5209.

**56. Van der Laan M., Pollard K., Bryan J.** A new partitioning around medoids algorithm // J. Stat. Comput. Simul. 2003. Vol. 73, No. 8. P. 575–584.

**57. Zhang Q., Couloigner I.** A new and efficient $k$-medoid algorithm for spatial clustering // Computational Science and Its Applications. Proc. Int. Conf. (Singapore, May 9–12, 2005). Pt. 3. Heidelberg: Springer, 2005. P. 181–189. (Lect. Notes Comput. Sci.; Vol. 3482).

**58. Yu D., Liu G., Guo M., Liu X.** An improved $k$-medoids algorithm based on step increasing and optimizing medoids // Expert Syst. Appl. 2018. Vol. 92. P. 464–473.

**59. Wang X., Wang X., Wilkes D. M.** Machine learning-based natural scene recognition for mobile robot localization in an unknown environment. Singapore: Springer, 2020. P. 85–108.

**60. Zadegan S. M. R., Mirzaie M., Sadoughi F.** Ranked $k$-medoids: A fast and accurate rank-based partitioning algorithm for clustering large datasets // Knowl.-Based Syst. 2013. Vol. 39. P. 133–143.

**61. Rangel E. M., Hendrix W., Agrawal A., Liao W., Choudhary A.** AGORAS: A fast algorithm for estimating medoids in large datasets // Procedia Comput. Sci. 2016. Vol. 80. P. 1159–1169.

**62. Fushimi T., Saito K., Ikeda T., Kazama K.** Accelerating greedy $k$-medoids clustering algorithm with $L_1$ distance by pivot generation // Foundations of Intelligent Systems. Proc. 23rd Int. Symp. (Warsaw, Poland, June 26–29, 2017). Cham: Springer, 2017. P. 87–96. (Lect. Notes Comput. Sci.; Vol. 10352).

**63. An H.-C., Svensson O.** Recent developments in approximation algorithms for facility location and clustering problems // Combinatorial Optimization and Graph Algorithms. Singapore: Springer, 2017. P. 1–19.

**64. Mladenović N., Brimberg J., Hansen P., Moreno-Pérez J.** The $p$-median problem: A survey of metaheuristic approaches // Eur. J. Oper. Res. 2007. Vol. 179, No. 3. P. 927–939.

**65. Reese J.** Solution methods for the $p$-median problem: An annotated bibliography // Networks. 2006. Vol. 28, No. 3. P. 125–142.

**66. Avella P., Sassano A., Vasilyev I. L.** Computational study of large-scale $p$-median problems // Math. Program. 2007. Vol. 109, No. 1. P. 89–114.

**67. Church R. L.** BEAMR: An exact and approximate model for the $p$-median problem // Comput. Oper. Res. 2008. Vol. 35, No. 2. P. 417–426.

**68. García S., Labbé M., Marín A.** Solving large $p$-median problems with a radius formulation // INFORMS J. Comput. 2011. Vol. 23, No. 4. P. 546–556.

**69. Mulvey J. M., Crowder H. P.** Cluster analysis: An application of Lagrangian relaxation // Manage. Sci. 1979. Vol. 25, No. 4. P. 329–340.

**70. Geoffrion A. M.** Lagrangian relaxation for integer programming // Approaches to Integer Programming. Heidelberg: Springer, 1974. P. 82–114. (Math. Program. Study; Vol. 2).

**71. Beltran C., Tadonki C., Vial J.** Solving the $p$-median problem with a semi-Lagrangian relaxation // Comput. Optim. Appl. 2006. Vol. 35, No. 2. P. 239–260.

**72. Avella P., Boccia M., Salerno S., Vasilyev I. L.** An aggregation heuristic for large scale $p$-median problem // Comput. Oper. Res. 2012. Vol. 39, No. 7. P. 1625–1632.

**73. Irawan C. A., Salhi S.** Solving large $p$-median problems by a multistage hybrid approach using demand points aggregation and variable neighbourhood search // J. Glob. Optim. 2015. Vol. 63. P. 537–554.

**74. Cebecauer M., Buzna L.** A versatile adaptive aggregation framework for spatially large discrete location-allocation problems // Comput. Ind. Eng. 2017. Vol. 111. P. 364–380.

**75. Jain K., Vazirani V. V.** Approximation algorithms for metric facility location and $k$-median problems using the primal-dual schema and Lagrangian relaxation // J. ACM. 2001. Vol. 48, No. 2. P. 274–296.

**76. Li S., Svensson O.** Approximating $k$-median via pseudo-approximation // Proc. 45th Annu. ACM Symp. Theory of Computing (Palo Alto, USA, June 1–4, 2013). New York: ACM, 2013. P. 901–910.

**77. Byrka J., Pensyl T., Rybicki B., Srinivasan A., Trinh K.** An improved approximation for $k$-median and positive correlation in budgeted optimization // ACM Trans. Algorithms. 2017. Vol. 13, No. 2. P. 23:1–23:31.

**78. Jain K., Mahdian M., Markakis E., Saberi A., Vazirani V. V.** Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP // J. ACM. 2003. Vol. 50, No. 6. P. 795–824.

**79. Nellore A., Ward R.** Recovery guarantees for exemplar-based clustering // Inf. Comput. 2015. Vol. 245. P. 165–180.

**80. Awasthi P., Bandeira A. S., Charikar M., Krishnaswamy R., Villar S., Ward R.** Relax, no need to round: Integrality of clustering formulations // Proc. 2015 Conf. Innovations in Theoretical Computer Science (Rehovot, Israel, Jan. 11–13, 2015). New York: ACM, 2015. P. 191–200.

**81. Crainic T. G., Gendreau M., Hansen P., Mladenović N.** Cooperative parallel variable neighborhood search for the $p$-median // J. Heuristics. 2004. Vol. 10, No. 3. P. 293–314.

**82. Garcia-López F., Melián-Batista B., Moreno-Pérez J. A., Moreno-Vega J. M.** The parallel variable neighborhood search for the $p$-median problem // J. Heuristics. 2002. Vol. 8, No. 3. P. 375–388.

**83. Garcia-López F., Melián-Batista B., Moreno-Pérez J. A., Moreno-Vega J. M.** Parallelization of the scatter search for the $p$-median problem // Parallel Comput. 2003. Vol. 29, No. 5. P. 575–589.

**84. Crainic T. G., Toulouse M.** Parallel meta-heuristics // Handbook of Metaheuristics. New York: Springer, 2010. P. 497–541. (Int. Ser. Oper. Res. Manage. Sci.; Vol. 146).

**85. Ma L., Lim G. J.** GPU-based parallel vertex substitution algorithm for the $p$-median problem // Comput. Ind. Eng. 2013. Vol. 64, No. 1. P. 381–388.

**86. Xiao N.** A parallel cooperative hybridization approach to the $p$-median problem // Environ. Plann. Ser. B. 2012. Vol. 39, No. 4. P. 755–774.

**87. Arbelaez A., Quesada L.** Parallelising the $k$-medoids clustering problem using space-partitioning // Proc. 6th Annu. Symp. Combinatorial Search (Leavenworth, USA, July 11–13, 2013). Palo Alto: AAAI, 2013. P. 20–28.

**88. Blelloch G. E., Tangwongsan K.** Parallel approximation algorithms for facility-location problems // Proc. 22nd Annu. ACM Symp. Parallelism in Algorithms and Architectures (Thira Santorini, Greece, June 13–15, 2010). New York: ACM, 2010. P. 315–324.

**89. Blelloch G. E., Gupta A., Tangwongsan K.** Parallel probabilistic tree embeddings, $k$-median, and buy-at-bulk network design // Proc. 24th Annu. ACM Symp. Parallelism in Algorithms and Architectures (Pittsburgh, USA, June 25–27, 2012). New York: ACM, 2012. P. 205–213.

**90. Bandyapadhyay S., Inamdar T., Pai S., Pemmaraju S. V.** Near-optimal clustering in the $k$-machine model // Proc. 19th Int. Conf. Distributed Computing and Networking (Varanasi, India, Jan. 4–7, 2018). New York: ACM, 2018. P. 15:1–15:10.

**91. Karloff H. J., Suri S., Vassilvitskii S.** A model of computation for MapReduce // Proc. 21st Annu. ACM-SIAM Symp. Discrete Algorithms (Austin, USA, Jan. 17-19, 2010). Philadelphia, PA: SIAM, 2010. P. 938–948.

**92. Ene A., Im S., Moseley B.** Fast clustering using MapReduce // Proc. 17th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (San Diego, USA, Aug. 21–24, 2011). New York: ACM, 2011. P. 681–689.

**93. Jakovits P., Srirama S. N.** Clustering on the cloud: Reducing CLARA to MapReduce // Proc. 2nd Nordic Symp. Cloud Computing and Internet Technologies (Oslo, Norway, Sep. 2–3, 2013). New York: ACM, 2013. P. 64–71.

**94. Ushakov A. V., Vasilyev I. L.** Near-optimal large-scale $k$-medoids clustering // Inf. Sci. 2021. Vol. 545. P. 344–362.

**95. Yang X., Lian L.** A New data mining algorithm based on MapReduce and Hadoop // Int. J. Signal Process., Image Process., Pattern Recognit. 2014. Vol. 7, No. 2. P. 131–142.

**96. Martino A., Rizzi A., Frattale Mascioli F. M.** Efficient approaches for solving the large-scale $k$-medoids problem: Towards structured data // Computational Intelligence. Proc. 9th Int. Joint Conf. (Funchal-Madeira, Portugal, Nov. 1–3, 2017). Cham: Springer, 2019. P. 199–219.

**97. Zhu Y., Wang F., Shan X., Lv X.** $K$-medoids clustering based on MapReduce and optimal search of medoids // Proc. 9th Int. Conf. Comput. Sci. Education (Vancouver, Canada, Aug 22–24, 2014). Piscataway: IEEE, 2014. P. 573–577.

**98. Song H., Lee J.-G., Han W.-S.** PAMAE: Parallel $k$-medoids clustering with high accuracy and efficiency // Proc. 23rd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (Halifax, Canada, Aug. 13–17, 2017). New York: ACM, 2017. P. 1087–1096.

**99. Mirzasoleiman B., Karbasi A., Sarkar R., Krause A.** Distributed submodular maximization: Identifying representative elements in massive data // Proc. 26th Int. Conf. Neural Information Processing Systems (Lake Tahoe, USA, Dec. 5–10, 2013). Vol. 2. Red Hook, NY: Curran Assoc., 2013. P. 2049–2057.

**100. Redondo J. L., Marín A., Ortigosa P. M.** A parallelized Lagrangian relaxation approach for the discrete ordered median problem // Ann. Oper. Res. 2016. Vol. 246, No. 1. P. 253–272.

**101. Mancini E. P., Marcarelli S., Vasilyev I. L., Villano U.** A grid-aware MIP solver: Implementation and case studies // Futur. Gener. Comp. Syst. 2008. Vol. 24, No. 2. P. 133–41.

**102. Lai P.-S., Fu H.-C.** Variance enhanced $k$-medoid clustering // Expert Syst. Appl. 2011. Vol. 38, No. 1. P. 764–775.

**103. Ayyala D. N., Lin S.** GrammR: Graphical representation and modeling of count data with application in metagenomics // Bioinformatics. 2015. Vol. 31, No. 10. P. 1648–1654.

**104. Elhamifar E., Sapiro G., Vidal R.** Finding exemplars from pairwise dissimilarities via simultaneous sparse recovery // Proc. 25th Int. Conf. Neural Information Processing Systems (Lake Tahoe, USA, Dec. 3–8, 2012). Vol. 1. Red Hook, NY: Curran Assoc., 2012. P. 19–27.

**105. Charikar M., Khuller S., Mount D. M., Narasimhan G.** Algorithms for facility location problems with outliers // Proc. 12th Annu. ACM-SIAM Symp. Discrete Algorithms (Washington, USA, Jan. 7–9, 2001). Philadelphia, PA: SIAM, 2001. P. 642–651.

**106. Frey B. J., Dueck D.** Clustering by passing messages between data points // Science. 2007. Vol. 315, No. 5814. P. 972–976.

**107. Brusco M. J., Köhn H.-F.** Comment on "Clustering by passing messages between data points" // Science. 2008. Vol. 319, No. 5864. P. 726–726.

**108. Brusco M. J., Steinley D.** Affinity propagation and uncapacitated facility location problems // J. Classif. 2015. Vol. 32, No. 3. P. 443–480.

**109. Leone M., Sumedha, Weigt M.** Clustering by soft-constraint affinity propagation: Applications to gene-expression data // Bioinformatics. 2007. Vol. 23, No. 20. P. 2708–2715.

**110. Mirchandani P., Jagannathan R.** Discrete facility location with nonlinear diseconomies in fixed costs // Ann. Oper. Res. 1989. Vol. 18, No. 1. P. 213–224.

**111. Körkel M.** Discrete facility location with nonlinear facility costs // RAIRO-Oper. Res. 1991. Vol. 25, No. 1. P. 31–43.

**112. Carrizosa E., Ushakov A. V., Vasilyev I. L.** A computational study of a nonlinear minsum facility location problem // Comput. Oper. Res. 2012. Vol. 39, No. 11. P. 2625–2633.

**113. Aghaee A., Ghadiri M., Baghshah M. S.** Active distance-based clustering using $k$-medoids // Advances in Knowledge Discovery and Data Mining. Proc. 20th Pacific-Asia Conf. (Auckland, New Zealand, Apr. 19–22, 2016). Cham: Springer, 2016. P. 253–264. (Lect. Notes Comput. Sci.; Vol. 9651).

**114. Randel R., Aloise D., Mladenović N., Hansen P.** On the $k$-medoids model for semi-supervised clustering // Variable Neighborhood Search. Proc. 6th Int. Conf. (Sithonia, Greece, Oct. 4–7, 2018). Cham: Springer, 2019. P. 13–27. (Lect. Notes Comput. Sci.; Vol. 11328).

**115. Marín A., Pelegrín M.** Adding incompatibilities to the simple plant location problem: Formulation, facets and computational experience // Comput. Oper. Res. 2019. Vol. 104. P. 174–190.

**116. Marín A., Pelegrín M.** The double-assignment plant location problem with co-location // Comput. Oper. Res. 2021. Vol. 126. P. 105059.

**117. Fersini E., Messina E., Archetti F.** A $p$-median approach for predicting drug response in tumour cells // BMC Bioinform. 2014. Vol. 15, No. 1. P. 1–19.

**118. Ushakov A. V., Klimentova K. B., Vasilyev I. L.** Bi-level and bi-objective $p$-median type problems for integrative clustering: Application to analysis of cancer gene-expression and drug-response data // IEEE-ACM Trans. Comput. Biol. Bioinform. 2018. Vol. 15, No. 1. P. 46–59.

**119. Алексеева Е. А., Кочетов Ю. А.** Генетический локальный поиск для задачи о $p$-медиане с предпочтениями клиентов // Дискрет. анализ и исслед. операций. Сер. 2. 2007. Т. 14, № 1. С. 3–31.

**120. Cánovas L., García S., Labbé M., Marín A.** A strengthened formulation for the simple plant location problem with order // Oper. Res. Lett. 2007. Vol. 35, No. 2. P. 141–150.

**121. Vasilyev I. L., Klimentova K. B., Boccia M.** Polyhedral study of simple plant location problem with order // Oper. Res. Lett. 2013. Vol. 41, No. 2. P. 153–158.

**122. Васильев И. Л., Климентова К. Б.** Метод ветвей и отсечений для задачи размещения с предпочтениями клиентов // Дискрет. анализ и исслед. операций. 2009. Т. 16, № 2. С. 21–41.

**123. Benati S., García S.** A mixed integer linear model for clustering with variable selection // Comput. Oper. Res. 2014. Vol. 43. P. 280–285.

**124. Benati S., García S., Puerto J.** Mixed integer linear programming and heuristic methods for feature selection in clustering // J. Oper. Res. Soc. 2018. Vol. 69, No. 9. P. 1379–1395.

**125. Kuehn A. A., Hamburger M. J.** A heuristic program for locating warehouses // Manage. Sci. 1963. Vol. 9, No. 4. P. 643–666.

**126. Mulvey J. M., Beck M. P.** Solving capacitated clustering problems // Eur. J. Oper. Res. 2003. Vol. 18, No. 3. P. 339–348.

**127. Negreiros M., Palhano A.** The capacitated centred clustering problem // Comput. Oper. Res. 2006. Vol. 33, No. 6. P. 1639–1663.

**128. Boccia M., Sforza A., Sterle C., Vasilyev I. L.** A cut and branch approach for the capacitated $p$-median problem based on Fenchel cutting planes // J. Math. Model. Algorithms. 2008. Vol. 7. P. 43–58.

**129. Gnägi M., Baumann P.** A matheuristic for large-scale capacitated clustering // Comput. Oper. Res. 2021. Vol. 132. P. 105304.

**130. Lorena L. A. N., Senne E. L. F.** A column generation approach to capacitated $p$-median problems // Comput. Oper. Res. 2004. Vol. 31, No. 6. P. 863–876.

**131. Mai F., Fry M. J., Ohlmann J. W.** Model-based capacitated clustering with posterior regularization // Eur. J. Oper. Res. 2018. Vol. 271, No. 2. P. 594–605.

**132. Stefanello F., de Araújo O. C. B., Müller F. M.** Matheuristics for the capacitated $p$-median problem // Int. Trans. Oper. Res. 2015. Vol. 22, No. 1. P. 149–167.

**133. Chou C.-A., Chaovalitwongse W. A., Berger-Wolf T. Y., DasGupta B., Ashley M. V.** Capacitated clustering problem in computational biology: Combinatorial and statistical approach for sibling reconstruction // Comput. Oper. Res. 2012. Vol. 39, No. 3. P. 609–619.

**134. Frahm J.-M., Fite-Georgel P., Gallup D.** [et al.]. Building Rome on a cloudless day // Computer Vision. Proc. 11th Eur. Conf. (Heraklion, Greece, Sep. 5–11, 2010). Pt. 4. Heidelberg: Springer, 2010. P. 368–381. (Lect. Notes Comput. Sci.; Vol. 6314).

**135. Gong Y., Pawlowski M., Yang F., Brandy L., Boundev L., Fergus R.** Web scale photo hash clustering on a single machine // Proc. 2015 IEEE Conf. Computer Vision and Pattern Recognition (Boston, USA, June 7–12, 2015). Piscataway: IEEE, 2015. P. 19–27.

**136. Brusco M. J., Steinley D., Stevens J.** $K$-medoids inverse regression // Commun. Stat. Theory Methods. 2019. Vol. 48, No. 20. P. 4999–5011.

**137. Suárez J. L., García S., Herrera F.** A tutorial on distance metric learning: Mathematical foundations, algorithms, experimental analysis, prospects and challenges // Neurocomputing. 2021. Vol. 425. P. 300–322.

**138. Song H. O., Jegelka S., Rathod V., Murphy K.** Deep metric learning via facility location // Proc. 2017 IEEE Conf. Computer Vision and Pattern Recognition (Honolulu, USA, July 21–26, 2017). Piscataway: IEEE, 2017. P. 2206–2214.

*Васильев Игорь Леонидович*
*Ушаков Антон Владимирович*

Статья поступила
30 апреля 2021 г.
После доработки —
17 июня 2021 г.
Принята к публикации
21 июня 2021 г.

DISKRETNYI ANALIZ I ISSLEDOVANIE OPERATSII
/DISCRETE ANALYSIS AND OPERATIONS RESEARCH/
October–December 2021. Vol. 28, No. 4. P. 5–60

UDC 519.87 DOI 10.33048/daio.2021.28.714

# DISCRETE FACILITY LOCATION IN MACHINE LEARNING

*I. L. Vasilyev*[a] and *A. V. Ushakov*[b]

Matrosov Institute for System Dynamics and Control Theory,
134 Lermontov Street, 664033 Irkutsk, Russia
E-mail: [a] vil@icc.ru, [b] aushakov@icc.ru

**Abstract.** Facility location problems form a wide class of optimization problems, extremely popular in combinatorial optimization and operations research. In any facility location problem, one must locate a set of facilities in order to satisfy the demands of customers so as a certain objective function is optimized. Besides numerous applications in public and private sectors, the problems are widely used in machine learning. For example, clustering can be viewed as a facility location problem where one needs to partition a set of customers into clusters assigned to open facilities. In this survey we briefly look at how ideas and approaches arisen in the field of facility location led to modern, popular machine learning algorithms supported by many data mining and machine learning software packages. We also review the state-of-the-art exact methods and heuristics, as well as some extensions of basic problems and algorithms arisen in applied machine learning tasks. Note that the main emphasis here lies on discrete facility location problems, which, for example, underlie many widely used clustering algorithms (PAM, affinity propagation, etc.). Since the high computational complexity of conventional facility location-based clustering algorithms hinders their application to modern large-scale real-life datasets, we also survey some modern approaches to implementation of the algorithms for such large data collections. Bibliogr. 138.

**Keywords:** machine learning, facility location, clustering.

## REFERENCES

1. **L. Cooper,** Location-allocation problems, *Oper. Res.* **11** (3), 331–343 (1963).
2. **L. Cooper,** Heuristic methods for location-allocation problems, *SIAM Rev.* **6** (1), 37–53 (1964).

This research is supported by the Russian Foundation for Basic Research (Project 20–17–50233).

English version: Journal of Applied and Industrial Mathematics **15** (4) (2021).

3. **F. Plastria,** The Weiszfeld algorithm: Proof, amendments, and extensions, in *Foundations of Location Analysis* (Springer, New York, 2011), pp. 357–389.
4. **S. Lloyd,** Least squares quantization in PCM, *IEEE Trans. Inf. Theory.* **28** (2), 129–137 (1982).
5. **E. W. Forgy,** Cluster analysis of multivariate data: Efficiency versus interpretability of classifications, *Biometrics* **21** (3), 768–769 (1965).
6. **A. Banerjee, S. Merugu, I. S. Dhillon,** and **J. Ghosh,** Clustering with Bregman divergences, *J. Mach. Learn. Res.* **6** (58), 1705–1749 (2005).
7. **J. MacQueen,** Some methods for classification and analysis of multivariate observations, in *Proc. 5th Berkeley Symp. Math. Stat. Probab., Berkeley, USA, June 21–July 18, 1965; Dec. 27, 1965–Jan. 7, 1966,* Vol. 1 (Univ. California Press, Berkeley, 1967), pp. 281–297.
8. **H. D. Vinod,** Integer programming and the theory of grouping, *J. Am. Stat. Assoc.* **64** (326), 506–519 (1969).
9. **M. L. Balinski,** Integer programming: Methods, uses, computations, *Manage. Sci.* **12** (3), 253–313 (1965).
10. **M. A. Efroymson** and **T. L. Ray,** A branch-bound algorithm for plant location, *Oper. Res.* **14** (3), 361–368 (1966).
11. **C. S. ReVelle** and **R. W. Swain,** Central facilities location, *Geogr. Anal.* **2** (1), 30–42 (1970).
12. **S. L. Hakimi,** Optimal location of switching centers and the absolute centers and medians of a graph, *Oper. Res.* **12** (3), 450–459 (1964).
13. **S. L. Hakimi,** Optimum distribution of switching centers in a communication network and some related graph theoretic problems, *Oper. Res.* **13** (3), 462–475 (1965).
14. **L. Kaufman** and **P. J. Rousseeuw,** Clustering by means of medoids, in *Statistical Data Analysis Based on the $L_1$-Norm and Related Methods* (North-Holland, Amsterdam, 1987), pp. 405–416.
15. **M. Charikar, S. Guha, E. Tardos,** and **D. B. Shmoys,** A constant-factor approximation algorithm for the $k$-median problem, *J. Comput. Syst. Sci.* **65** (1), 129–149 (2002).
16. **M.-F. Balcan, A. Blum,** and **A. Gupta,** Approximate clustering without the approximation, in *Proc. 20th Annu. ACM-SIAM Symp. Discrete Algorithms, New York, USA, Jan. 4–6, 2009* (SIAM, Philadelphia, 2009), pp. 1068–1077.
17. **S. Ahmadian, A. Norouzi-Fard, O. Svensson,** and **J. Ward,** Better guarantees for $k$-means and Euclidean $k$-median by primal-dual algorithms, *SIAM J. Comput.* **49** (4), FOCS17-97–FOCS17-156 (2020).
18. **O. Kariv** and **S. Hakimi,** An algorithmic approach to network location problems. II: The $p$-medians, *SIAM J. Appl. Math.* **37** (3), 539–560 (1979).
19. **N. Megiddo** and **K. J. Supowit,** On the complexity of some common geometric location problems, *SIAM J. Comput.* **13** (1), 182–196 (1984).
20. **M. Mahajan, P. Nimbhorkar,** and **K. Varadarajan,** The planar $k$-means problem is NP-hard, *Theor. Comput. Sci.* **442**, 13–21 (2012).

21. **N. Megiddo, E. Zemel,** and **S. L. Hakimi,** The maximum coverage location problem, *SIAM J. Algebr. Discrete Methods* **4** (2), 253–261 (1983).
22. **D. Aloise, A. Deshpande, P. Hansen,** and **P. Popat,** NP-hardness of Euclidean sum-of-squares clustering, *Mach. Learn.* **75** (2), 245–248 (2009).
23. **C. H. Papadimitriou,** Worst-case and probabilistic analysis of a geometric location problem, *SIAM J. Comput.* **10** (3), 542–557 (1981).
24. **K. E. Rosing, C. S. ReVelle,** and **H. Rosing-Vogelaar,** The $p$-median and its linear programming relaxation: An approach to large problems, *J. Oper. Res. Soc.* **30** (9), 815–823 (1979).
25. **R. L. Church,** COBRA: A new formulation of the classic $p$-median location problem, *Ann. Oper. Res.* **122** (1–4), 103–120 (2003).
26. **G. Cornuejols, G. L. Nemhauser,** and **L. A. Wolsey,** A canonical representation of simple plant location problems and its applications, *SIAM J. Algebr. Discrete Methods* **1** (3), 261–272 (1980).
27. **P. Hansen, J. Brimberg, D. Urosević,** and **N. Mladenović,** Solving large $p$-median clustering problems by primal-dual variable neighborhood search, *Data Min. Knowl. Discov.* **19** (3), 351–375 (2009).
28. **S. Elloumi,** A tighter formulation of the $p$-median problem, *J. Comb. Optim.* **19** (1), 69–83 (2010).
29. **F. E. Maranzana,** On the location of supply points to minimize transport costs, *Oper. Res. Q.* **15** (3), 261–270 (1964).
30. **M. B. Teitz** and **P. Bart,** Heuristic methods for estimating the generalized vertex median of a weighted graph, *Oper. Res.* **16** (5), 955–961 (1968).
31. **J. A. Hartigan** and **M. A. Wong,** Algorithm AS 136: A $k$-means clustering algorithm, *J. R. Stat. Soc., Ser. C,* **28** (1), 100–108 (1979).
32. **R. L. Church** and **C. S. ReVelle,** Theoretical and computational links between the $p$-median, location set-covering, and the maximal covering location problem, *Geogr. Anal.* **8** (4), 406–415 (1976).
33. **G. Cornuejols, M. L. Fisher,** and **G. L. Nemhauser,** Location of bank accounts to optimize float: An analytic study of exact and approximate algorithms, *Manage. Sci.* **23** (8), 789–810 (1977).
34. **V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala,** and **V. Pandit,** Local search heuristics for $k$-median and facility location problems, *SIAM J. Comput.* **33** (3), 544–562 (2004).
35. **A. Gupta** and **K. Tangwongsan,** Simpler analyses of local search algorithms for facility location (Cornell Univ., Ithaca, NY, 2008) (Cornell Univ. Libr. e-Print Archive; arXiv:0809.2554).
36. **Yu. A. Kochetov, M. G. Pashchenko,** and **A. V. Plyasunov,** On the complexity of local search in the $p$-median problem, *Diskretn. Anal. Issled. Oper., Ser. 2,* **12** (2), 44–71 (2005) [Russian].
37. **E. V. Alekseeva, Yu. A. Kochetov,** and **A. V. Plyasunov,** Complexity of local search for the $p$-median problem, *Eur. J. Oper. Res.* **191** (3), 736–752 (2008).
38. **R. A. Whitaker,** A fast algorithm for the greedy interchange for large-scale clustering and median location problems, *INFOR* **21**, 95–108 (1983).

**39. R. A. Whitaker,** Some interchange algorithms for median location problems, *Environ. Plann., Ser. B,* **9** (2), 119–129 (1982).

**40. P. J. Densham** and **G. Rushton,** A more efficient heuristic for solving large $p$-median problems, *Pap. Reg. Sci.* **71** (3), 307–329 (1992).

**41. P. Hansen** and **N. Mladenović,** Variable neighborhood search for the $p$-median, *Locat. Sci.* **5** (4), 207–226 (1997).

**42. E. Schubert** and **P. J. Rousseeuw,** Faster $k$-medoids clustering: Improving the PAM, CLARA, and CLARANS algorithms, in *Similarity Search and Applications* (Proc. 12th Int. Conf., Newark, USA, Oct. 2–4, 2019) (Springer, Cham, 2019), pp. 171–187 (Lect. Notes Comput. Sci., Vol. 11807).

**43. M. G. C. Resende** and **R. F. Werneck,** A fast swap-based local search procedure for location problems, *Ann. Oper. Res.* **150** (1), 205–230 (2007).

**44. T. A. Feo** and **M. G. C. Resende,** Greedy randomized adaptive search procedures, *J. Glob. Optim.* **6** (2), 109–133 (1995).

**45. M. G. C. Resende** and **R. F. Werneck,** A hybrid heuristic for the $p$-median problem, *J. Heuristics* **10** (1), 59–88 (2004).

**46. B. Mirzasoleiman, A. Badanidiyuru, A. Karbasi, J. Vondrák,** and **A. Krause,** Lazier than lazy greedy, in *Proc. 29th AAAI Conf. Artificial Intelligence, Austin, USA, Jan. 25–30, 2015* (AAAI Press, Palo Alto, 2015), pp. 1812–1818.

**47. M. Tiwari, M. J. Zhang, J. Mayclin,** and **S. Thrun,** Bandit-PAM: Almost linear time $k$-medoids clustering via multi-armed bandits, in *Proc. 34th Conf. Neural Information Processing Systems, Vancouver, Canada, Dec. 6–12, 2020* (Curran Assoc., Red Hook, 2020), pp. 10211–10222.

**48. T. Hastie, R. Tibshirani,** and **J. Friedman,** *The elements of statistical learning: Data mining, inference, and prediction* (Springer, New York, 2009).

**49. H.-S. Park** and **C.-H. Jun,** A simple and fast algorithm for $k$-medoids clustering, *Expert Syst. Appl.* **36** (2, Pt. 2), 3336–3341 (2009).

**50. J. Newling** and **F. Fleuret,** A sub-quadratic exact medoid algorithm, *Proc. Mach. Learn. Res.* **54**, 185–193 (2017).

**51. V. Bagaria, G. Kamath, V. Ntranos, M. Zhang,** and **D. Tse,** Medoids in almost-linear time via multi-armed bandits, *Proc. Mach. Learn. Res.* **84**, 500–509 (2018).

**52. A. A. Paterlini, M. A. Nascimento,** and **C. T. Jr.,** Using pivots to speed-up $k$-medoids clustering, *J. Inf. Data Manage.* **2** (2), 221–236 (2011).

**53. L. Kaufman** and **P. J. Rousseeuw,** *Finding Groups in Data: An Introduction to Cluster Analysis* (Wiley, Hoboken, NJ, 2005).

**54. R. T. Ng** and **J. Han,** CLARANS: A method for clustering objects for spatial data mining, *IEEE Trans. Knowl. Data Eng.* **14** (5), 1003–1016 (2002).

**55. J. Newling** and **F. Fleuret,** $K$-medoids for $K$-means seeding, in *Proc. 31st Int. Conf. Neural Information Processing Systems, Long Beach, CA, USA, Dec. 4–9, 2017* (Curran Assoc., Red Hook, NY, 2017), pp. 5201–5209.

**56. M. Van der Laan, K. Pollard,** and **J. Bryan,** A new partitioning around medoids algorithm, *J. Stat. Comput. Simul.* **73** (8), 575–584 (2003).

**57. Q. Zhang** and **I. Couloigner,** A new and efficient $k$-medoid algorithm for spatial clustering, in *Computational Science and Its Applications* (Proc. Int. Conf., Singapore, May 9–12, 2005), Pt. 3 (Springer, Heidelberg, 2005), pp. 181–189 (Lect. Notes Comput. Sci., Vol. 3482).

**58. D. Yu, G. Liu, M. Guo,** and **X. Liu,** An improved $k$-medoids algorithm based on step increasing and optimizing medoids, *Expert Syst. Appl.* **92**, 464–473 (2018).

**59. X. Wang, X. Wang,** and **D. M. Wilkes,** *Machine Learning-Based Natural Scene Recognition for Mobile Robot Localization in an Unknown Environment* (Springer, Singapore, 2020), pp. 85–108.

**60. S. M. R. Zadegan, M. Mirzaie,** and **F. Sadoughi,** Ranked $k$-medoids: A fast and accurate rank-based partitioning algorithm for clustering large datasets, *Knowl.-Based Syst.* **39**, 133–143 (2013).

**61. E. M. Rangel, W. Hendrix, A. Agrawal, W. Liao,** and **A. Choudhary,** AGORAS: A fast algorithm for estimating medoids in large datasets, *Procedia Comput. Sci.* **80**, 1159–1169 (2016).

**62. T. Fushimi, K. Saito, T. Ikeda,** and **K. Kazama,** Accelerating greedy $k$-medoids clustering algorithm with $L_1$ distance by pivot generation, in *Foundations of Intelligent Systems* (Proc. 23rd Int. Symp., Warsaw, Poland, June 26–29, 2017) (Springer, Cham, 2017), pp. 87–96 (Lect. Notes Comput. Sci., Vol. 10352).

**63. H.-C. An** and **O. Svensson,** Recent developments in approximation algorithms for facility location and clustering problems, in *Combinatorial Optimization and Graph Algorithms* (Springer, Singapore, 2017), pp. 1–19.

**64. N. Mladenović, J. Brimberg, P. Hansen,** and **J. Moreno-Pérez,** The $p$-median problem: A survey of metaheuristic approaches, *Eur. J. Oper. Res.* **179** (3), 927–939 (2007).

**65. J. Reese,** Solution methods for the $p$-median problem: An annotated bibliography, *Networks* **28** (3), 125–142 (2006).

**66. P. Avella, A. Sassano,** and **I. L. Vasilyev,** Computational study of large-scale $p$-median problems, *Math. Program.* **109** (1), 89–114 (2007).

**67. R. L. Church,** BEAMR: An exact and approximate model for the $p$-median problem, *Comput. Oper. Res.* **35** (2), 417–426 (2008).

**68. García S., Labbé M., Marín A.** Solving large $p$-median problems with a radius formulation, *INFORMS J. Comput.* **23** (4), 546–556 (2011).

**69. J. M. Mulvey** and **H. P. Crowder,** Cluster analysis: An application of Lagrangian relaxation, *Manage. Sci.* **25** (4), 329–340 (1979).

**70. A. M. Geoffrion,** Lagrangian relaxation for integer programming, in *Approaches to Integer Programming,* (Springer, Heidelberg, 1974), pp. 82–114 (Math. Program. Study, Vol. 2).

**71. C. Beltran, C. Tadonki,** and **J. Vial,** Solving the $p$-median problem with a semi-Lagrangian relaxation, *Comput. Optim. Appl.* **35** (2), 239–260 (2006).

**72. P. Avella, M. Boccia, S. Salerno,** and **I. L. Vasilyev,** An aggregation heuristic for large scale $p$-median problem, *Comput. Oper. Res.* **39** (7), 1625–1632 (2012).

**73. C. A. Irawan** and **S. Salhi,** Solving large $p$-median problems by a multistage hybrid approach using demand points aggregation and variable neighbourhood search, *J. Glob. Optim.* **63**, 537–554 (2015).

**74. M. Cebecauer** and **L. Buzna,** A versatile adaptive aggregation framework for spatially large discrete location-allocation problems, *Comput. Ind. Eng.* **111**, 364–380 (2017).

**75. K. Jain** and **V. V. Vazirani,** Approximation algorithms for metric facility location and $k$-median problems using the primal-dual schema and Lagrangian relaxation, *J. ACM* **48** (2), 274–296 (2001).

**76. S. Li** and **O. Svensson,** Approximating $k$-median via pseudo-approximation, in *Proc. 45th Annu. ACM Symp. Theory of Computing, Palo Alto, USA, June 1–4, 2013* (ACM, New York, 2013), pp. 901–910.

**77. J. Byrka, T. Pensyl, B. Rybicki, A. Srinivasan,** and **K. Trinh,** An improved approximation for $k$-median and positive correlation in budgeted optimization, *ACM Trans. Algorithms* **13** (2), 23:1–23:31 (2017).

**78. K. Jain, M. Mahdian, E. Markakis, A. Saberi,** and **V. V. Vazirani,** Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP, *J. ACM* **50** (6), 795–824 (2003).

**79. A. Nellore** and **R. Ward,** Recovery guarantees for exemplar-based clustering, *Inf. Comput.* **245**, 165–180 (2015).

**80. P. Awasthi, A. S. Bandeira, M. Charikar, R. Krishnaswamy, S. Villar,** and **R. Ward,** Relax, no need to round: Integrality of clustering formulations, in *Proc. 2015 Conf. Innovations in Theoretical Computer Science, Rehovot, Israel, Jan. 11–13, 2015* (ACM, New York, 2015), pp. 191–200.

**81. T. G. Crainic, M. Gendreau, P. Hansen,** and **N. Mladenović,** Cooperative parallel variable neighborhood search for the $p$-median, *J. Heuristics* **10** (3), 293–314 (2004).

**82. F. Garcia-López, B. Melián-Batista, J. A. Moreno-Pérez,** and **J. M. Moreno-Vega,** The parallel variable neighborhood search for the $p$-median problem, *J. Heuristics* **8** (3), 375–388 (2002).

**83. F. Garcia-López, B. Melián-Batista, J. A. Moreno-Pérez,** and **J. M. Moreno-Vega,** Parallelization of the scatter search for the $p$-median problem, *Parallel Comput.* **29** (5), 575–589 (2003).

**84. T. G. Crainic** and **M. Toulouse,** Parallel meta-heuristics, in *Handbook of Metaheuristics* (Springer, New York, 2010), pp. 497–541 (Int. Ser. Oper. Res. Manage. Sci., Vol. 146).

**85. L. Ma** and **G. J. Lim,** GPU-based parallel vertex substitution algorithm for the $p$-median problem, *Comput. Ind. Eng.* **64** (1), 381–388 (2013).

**86. N. Xiao,** A parallel cooperative hybridization approach to the $p$-median problem, *Environ. Plann., Ser. B,* **39** (4), 755–774 (2012).

**87. A. Arbelaez** and **L. Quesada,** Parallelising the $k$-medoids clustering problem using space-partitioning, in *Proc. 6th Annu. Symp. Combinatorial Search, Leavenworth, USA, July 11–13, 2013* (AAAI, Palo Alto, 2013), pp. 20–28.

**88.** **G. E. Blelloch** and **K. Tangwongsan,** Parallel approximation algorithms for facility-location problems, in *Proc. 22nd Annu. ACM Symp. Parallelism in Algorithms and Architectures, Thira Santorini, Greece, June 13–15, 2010* (ACM, New York, 2010), pp. 315–324.

**89.** **G. E. Blelloch, A. Gupta,** and **K. Tangwongsan,** Parallel probabilistic tree embeddings, $k$-median, and buy-at-bulk network design, in *Proc. 24th Annu. ACM Symp. Parallelism in Algorithms and Architectures, Pittsburgh, USA, June 25–27, 2012* (ACM, New York, 2012), pp. 205–213.

**90.** **S. Bandyapadhyay, T. Inamdar, S. Pai,** and **S. V. Pemmaraju,** Near-optimal clustering in the $k$-machine model, in *Proc. 19th Int. Conf. Distributed Computing and Networking, Varanasi, India, Jan. 4–7, 2018* (ACM, New York, 2018), pp. 15:1–15:10.

**91.** **H. J. Karloff, S. Suri,** and **S. Vassilvitskii,** A model of computation for MapReduce, in *Proc. 21st Annu. ACM-SIAM Symp. Discrete Algorithms, Austin, USA, Jan. 17-19, 2010* (SIAM, Philadelphia, PA, 2010), pp. 938–948.

**92.** **A. Ene, S. Im,** and **B. Moseley,** Fast clustering using MapReduce, in *Proc. 17th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, San Diego, USA, Aug. 21–24, 2011* (ACM, New York, 2011), pp. 681–689.

**93.** **P. Jakovits** and **S. N. Srirama,** Clustering on the cloud: Reducing CLARA to MapReduce, in *Proc. 2nd Nordic Symp. Cloud Computing and Internet Technologies, Oslo, Norway, Sep. 2–3, 2013* (ACM, New York, 2013), pp. 64–71.

**94.** **A. V. Ushakov** and **I. L. Vasilyev,** Near-optimal large-scale $k$-medoids clustering, *Inf. Sci.* **545**, 344–362 (2021).

**95.** **X. Yang** and **L. Lian,** A New data mining algorithm based on MapReduce and Hadoop, *Int. J. Signal Process., Image Process., Pattern Recognit.* **7** (2), 131–142 (2014).

**96.** **A. Martino, A. Rizzi,** and **F. M. Frattale Mascioli,** Efficient approaches for solving the large-scale $k$-medoids problem: Towards structured data, in *Computational Intelligence* (Proc. 9th Int. Joint Conf., Funchal-Madeira, Portugal, Nov. 1–3, 2017) (Springer, Cham, 2019), pp. 199–219.

**97.** **Y. Zhu, F. Wang, X. Shan,** and **X. Lv,** $K$-medoids clustering based on MapReduce and optimal search of medoids, in *Proc. 9th Int. Conf. Comput. Sci. Education, Vancouver, Canada, Aug 22–24, 2014* (IEEE, Piscataway, 2014), pp. 573–577.

**98.** **H. Song, J.-G. Lee,** and **W.-S. Han,** PAMAE: Parallel $k$-medoids clustering with high accuracy and efficiency, in *Proc. 23rd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, Halifax, Canada, Aug. 13–17, 2017* (ACM, New York, 2017), pp. 1087–1096.

**99.** **B. Mirzasoleiman, A. Karbasi, R. Sarkar,** and **A. Krause,** Distributed submodular maximization: Identifying representative elements in massive data, in *Proc. 26th Int. Conf. Neural Information Processing Systems, Lake Tahoe, USA, Dec. 5–10, 2013,* Vol. 2 (Curran Assoc., Red Hook, NY, 2013), pp. 2049–2057.

**100. J. L. Redondo, A. Marín,** and **P. M. Ortigosa,** A parallelized Lagrangian relaxation approach for the discrete ordered median problem, *Ann. Oper. Res.* **246** (1), 253–272 (2016).

**101. E. P. Mancini, S. Marcarelli, I. L. Vasilyev,** and **U. Villano,** A grid-aware MIP solver: Implementation and case studies, *Futur. Gener. Comp. Syst.* **24** (2), 133–41 (2008).

**102. P.-S. Lai** and **H.-C. Fu,** Variance enhanced $k$-medoid clustering, *Expert Syst. Appl.* **38** (1), 764–775 (2011).

**103. D. N. Ayyala** and **S. Lin,** GrammR: Graphical representation and modeling of count data with application in metagenomics, *Bioinformatics* **31** (10), 1648–1654 (2015).

**104. E. Elhamifar, G. Sapiro,** and **R. Vidal,** Finding exemplars from pairwise dissimilarities via simultaneous sparse recovery, in *Proc. 25th Int. Conf. Neural Information Processing Systems, Lake Tahoe, USA, Dec. 3–8, 2012* (Curran Assoc., Vol. 1. Red Hook, NY, 2012), pp. 19–27.

**105. M. Charikar, S. Khuller, D. M. Mount,** and **G. Narasimhan,** Algorithms for facility location problems with outliers, in *Proc. 12th Annu. ACM-SIAM Symp. Discrete Algorithms, Washington, USA, Jan. 7–9, 2001* (SIAM, Philadelphia, PA, 2001), pp. 642–651.

**106. B. J. Frey** and **D. Dueck,** Clustering by passing messages between data points, *Science* **315** (5814), 972–976 (2007).

**107. M. J. Brusco** and **H.-F. Köhn,** Comment on "Clustering by passing messages between data points", *Science* **319** (5864), 726–726 (2008).

**108. M. J. Brusco** and **D. Steinley,** Affinity propagation and uncapacitated facility location problems, *J. Classif.* **32** (3), 443–480 (2015).

**109. M. Leone, Sumedha,** and **M. Weigt,** Clustering by soft-constraint affinity propagation: Applications to gene-expression data, *Bioinformatics* **23** (20), 2708–2715 (2007).

**110. P. Mirchandani** and **R. Jagannathan,** Discrete facility location with nonlinear diseconomies in fixed costs, *Ann. Oper. Res.* **18** (1), 213–224 (1989).

**111. M. Körkel,** Discrete facility location with nonlinear facility costs, *RAIRO-Oper. Res.* **25** (1), 31–43 (1991).

**112. E. Carrizosa, A. V. Ushakov,** and **I. L. Vasilyev,** A computational study of a nonlinear minsum facility location problem, *Comput. Oper. Res.* **39** (11), 2625–2633 (2012).

**113. A. Aghaee, M. Ghadiri,** and **M. S. Baghshah,** Active distance-based clustering using $k$-medoids, in *Advances in Knowledge Discovery and Data Mining* (Proc. 20th Pacific-Asia Conf., Auckland, New Zealand, Apr. 19–22, 2016) (Springer, Cham, 2016), pp. 253–264 (Lect. Notes Comput. Sci., Vol. 9651).

**114. R. Randel, D. Aloise, N. Mladenović,** and **P. Hansen,** On the $k$-medoids model for semi-supervised clustering, in *Variable Neighborhood Search* (Proc. 6th Int. Conf., Sithonia, Greece, Oct. 4–7, 2018) (Springer, Cham, 2019), pp. 13–27 (Lect. Notes Comput. Sci., Vol. 11328).

**115.** **A. Marín** and **M. Pelegrín,** Adding incompatibilities to the simple plant location problem: Formulation, facets and computational experience, *Comput. Oper. Res.* **104**, 174–190 (2019).

**116.** **A. Marín** and **M. Pelegrín,** The double-assignment plant location problem with co-location, *Comput. Oper. Res.* **126**, 105059 (2021).

**117.** **E. Fersini, E. Messina,** and **F. Archetti,** A $p$-median approach for predicting drug response in tumour cells, *BMC Bioinform.* **15** (1), 1–19 (2014).

**118.** **A. V. Ushakov, K. B. Klimentova,** and **I. L. Vasilyev,** Bi-level and bi-objective $p$-median type problems for integrative clustering: Application to analysis of cancer gene-expression and drug-response data, *IEEE-ACM Trans. Comput. Biol. Bioinform.* **15** (1), 46–59 (2018).

**119.** **E. A. Alekseeva** and **Yu. A. Kochetov,** Genetic local search for the $p$-median problem with customer preferences, *Diskretn. Anal. Issled. Oper., Ser. 2,* **14** (1), 3–31 (2007) [Russian].

**120.** **L. Cánovas, S. García, M. Labbé,** and **A. Marín,** A strengthened formulation for the simple plant location problem with order, *Oper. Res. Lett.* **35** (2), 141–150 (2007).

**121.** **I. L. Vasilyev, K. B. Klimentova,** and **M. Boccia,** Polyhedral study of simple plant location problem with order, *Oper. Res. Lett.* **41** (2), 153–158 (2013).

**122.** **I. L. Vasilyev** and **K. B. Klimentova,** A branch and bound method for the facility location problem with customer preferences, *Diskretn. Anal. Issled. Oper.* **16** (2), 21–41 (2009) [Russian] [*J. Appl. Ind. Math.* **4** (3), 441–454 (2010)].

**123.** **S. Benati** and **S. García,** A mixed integer linear model for clustering with variable selection, *Comput. Oper. Res.* **43**, 280–285 (2014).

**124.** **S. Benati, S. García,** and **J. Puerto,** Mixed integer linear programming and heuristic methods for feature selection in clustering, *J. Oper. Res. Soc.* **69** (9), 1379–1395 (2018).

**125.** **A. A. Kuehn** and **M. J. Hamburger,** A heuristic program for locating warehouses, *Manage. Sci.* **9** (4), 643–666 (1963).

**126.** **J. M. Mulvey** and **M. P. Beck,** Solving capacitated clustering problems, *Eur. J. Oper. Res.* **18** (3), 339–348 (2003).

**127.** **M. Negreiros** and **A. Palhano,** The capacitated centred clustering problem, *Comput. Oper. Res.* **33** (6), 1639–1663 (2006).

**128.** **M. Boccia, A. Sforza, C. Sterle,** and **I. L. Vasilyev,** A cut and branch approach for the capacitated $p$-median problem based on Fenchel cutting planes, *J. Math. Model. Algorithms* **7**, 43–58 (2008).

**129.** **M. Gnägi** and **P. Baumann,** A matheuristic for large-scale capacitated clustering, *Comput. Oper. Res.* **132**, 105304 (2021).

**130.** **L. A. N. Lorena** and **E. L. F. Senne,** A column generation approach to capacitated $p$-median problems, *Comput. Oper. Res.* **31** (6), 863–876 (2004).

**131.** **F. Mai, M. J. Fry,** and **J. W. Ohlmann,** Model-based capacitated clustering with posterior regularization, *Eur. J. Oper. Res.* **271** (2), 594–605 (2018).

**132. F. Stefanello, O. C. B. de Araújo,** and **F. M. Müller,** Matheuristics for the capacitated $p$-median problem, *Int. Trans. Oper. Res.* **22** (1), 149–167 (2015).

**133. C.-A. Chou, W. A. Chaovalitwongse, T. Y. Berger-Wolf, B. DasGupta,** and **M. V. Ashley,** Capacitated clustering problem in computational biology: Combinatorial and statistical approach for sibling reconstruction, *Comput. Oper. Res.* **39** (3), 609–619 (2012).

**134. J.-M. Frahm, P. Fite-Georgel, D. Gallup** [et al.], Building Rome on a cloudless day, in *Computer Vision* (Proc. 11th Eur. Conf., Heraklion, Greece, Sep. 5–11, 2010), Pt. 4 (Springer, Heidelberg, 2010), pp. 368–381 (Lect. Notes Comput. Sci., Vol. 6314).

**135. Y. Gong, M. Pawlowski, F. Yang, L. Brandy, L. Boundev,** and **R. Fergus,** Web scale photo hash clustering on a single machine, in *Proc. 2015 IEEE Conf. Computer Vision and Pattern Recognition, Boston, USA, June 7–12, 2015* (IEEE, Piscataway, 2015), pp. 19–27.

**136. M. J. Brusco, D. Steinley,** and **J. Stevens,** $K$-medoids inverse regression, *Commun. Stat. Theory Methods* **48** (20), 4999–5011 (2019).

**137. J. L. Suárez, S. García,** and **F. Herrera,** A tutorial on distance metric learning: Mathematical foundations, algorithms, experimental analysis, prospects and challenges, *Neurocomputing* **425**, 300–322 (2021).

**138. H. O. Song, S. Jegelka, V. Rathod,** and **K. Murphy,** Deep metric learning via facility location, in *Proc. 2017 IEEE Conf. Computer Vision and Pattern Recognition, Honolulu, USA, July 21–26, 2017* (IEEE, Piscataway, 2017), pp. 2206–2214.

*Igor L. Vasilyev*
*Anton V. Ushakov*

Received April 30, 2021
Revised June 17, 2021
Accepted June 21, 2021