ВЕСТНИК ТОМСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА
2021 Управление, вычислительная техника и информатика № 56

УДК 519.7
DOI: 10.17223/19988605/56/10

А.А. Солдатенко, Д.В. Семенова

О СВЯЗИ БИКЛИК ГИПЕРГРАФА И ВСЕХ МАКСИМАЛЬНО ПОЛНЫХ ПОДМАТРИЦ МАТРИЦЫ СМЕЖНОСТИ

Работа поддержана Красноярским математическим центром, финансируемым Минобрнауки РФ в рамках мероприятий по созданию и развитию региональных НОМЦ (Соглашение 075-02-2020-1534/1).

Исследуется связь между задачей поиска всех максимально полных подматриц (0, 1)-матрицы и задачей поиска всех максимальных биклик гиперграфов. Разработан алгоритм *HFindMCS* нахождения всех максимально полных подматриц (0, 1)-матрицы. Проведен теоретический анализ вычислительной сложности предложенного алгоритма и доказана его корректность. Вычислительные эксперименты показали эффективность алгоритма *HFindMCS* на гиперграфах с ограниченной степенью. Предложена процедура перехода от решения задачи поиска всех максимально полных подматриц (0, 1)-матрицы к решению задачи поиска всех максимальных биклик гиперграфов.

Ключевые слова: гиперграф; максимальные биклики; (0, 1)-матрица; максимально полные подматрицы.

Гиперграфы являются расширением классической теории графов на случай, когда ребро графа может содержать число вершин, отличное от двух. Изначально гиперграфы нашли практическое применение в моделировании реляционных баз данных и комбинаторной химии [1, 2]. Возможность объединения нескольких вершин одним ребром дает мощный инструмент для моделирования различных сетей и систем. В настоящее время гиперграфовые модели активно применяются при построении и исследовании дорожных и телекоммуникационных сетей [3, 4], для построения семантических сетей и представления лингвистических знаний [5, 6], при решении задач управления большими системами [7].

Значительная часть задач исследований указанных моделей сводится к проблемам определения различных конфигураций. Под конфигурацией понимается любая система подмножеств конечного множества [8]. Особый интерес вызывают задачи перечислительного типа [8], в которых существование интересующих исследователя конфигураций не вызывает сомнения, вопрос – лишь в количестве и способе их представления. Большинство задач, связанных с поиском различных конфигураций в данных, имеет высокую сложность, с которой ничего нельзя сделать, однако поиск новых моделей и алгоритмов может повысить производительность вычислений. При этом использование асимптотически более медленных алгоритмов может быть оправдано тем, что для наборов данных, организованных определенным образом, они будут эффективнее [9]. Задача может быть сложной в общем случае, но иметь эффективный алгоритм для некоторого частного случая или класса объектов. Если сети и системы представить гиперграфами, то интересующие конфигурации удобно описывать на языке (0, 1)-матриц [10]. Например, задача выделения всех максимальных биклик гиперграфа сводится к задаче построения следующих конфигураций – полных единичных подматриц (0, 1)-матрицы [11].

В настоящей работе рассматривается задача нахождения всех максимально полных подматриц (0, 1)-матрицы и схожая с ней задача поиска максимальных биклик гиперграфа. Поиск максимально полных единичных подматриц востребован в задаче нахождения всех формальных понятий, которая возникает при поиске скрытых взаимосвязей и построении онтологий предметных областей [5]. Поиск максимальных биклик связан с задачей адресации внутри коммуникационной сети [3]. В сетях, представленных графами и гиперграфами, выделение биклик способно повысить эффективность решения задачи адресации и маршрутизации [3, 12, 13]. Также максимальные биклики востребованы в анализе данных и при кластеризации, имеют широкое применение при исследовании генных структур и социальных сетей [14–16]. Заметим, что задачи поиска таких конфигураций являются $\#P$-полными [17].

В работе предлагаются новый алгоритм выделения максимально полных подматриц (0, 1)-матрицы, основанный на теории гиперграфов, и процедура выделения всех максимальных биклик на основе всех максимально полных подматриц.

В первом разделе статьи приводятся необходимые понятия и определения (0, 1)-матриц и теории гиперграфов. Второй раздел содержит описание предлагаемого алгоритма, доказательство его корректности и теоретический анализ вычислительной сложности. В третьем разделе представлены результаты вычислительных экспериментов.

1. Основные понятия и определения

Пусть задана (0, 1)-матрица I размера $n \times m$, где $n \geq 1$, $m \geq 1$ [8, 10].

Определение 1. Полной подматрицей (0, 1)-матрицы называется подматрица, все без исключения элементы которой равны 1 [18].

Определение 2. Максимально полной подматрицей называется полная подматрица, не содержащая никакой другой максимально полной матрицы [18].

Последнее можно трактовать следующим образом: максимально полной подматрицей называется полная подматрица, не входящая ни в какую другую полную подматрицу.

Матрица I по своей сути может представлять множество всевозможных комбинаторных объектов, примерами которых являются: матрицы инцидентности и смежности графов и гиперграфов, контекст в анализе формальных понятий, реализация конечного случайного множества и др. [1, 5, 8, 10].

Пусть задан (n, m)-гиперграф $H = (X, U)$, где X – конечное множество вершин, U – конечное семейство гиперребер гиперграфа, при этом $|X| \geq 1$, $|U| \geq 1$ и всякое гиперребро гиперграфа – некоторое подмножество множества X. Пусть $X(u)$ – множество всех вершин, инцидентных гиперребру $u \in U$, а $U(x)$ – множество гиперребер, инцидентных вершине $x \in X$. Двойственным к гиперграфу $H = (X, U)$ называют гиперграф $H^* = (X^*, U^*)$, где $X^* = U$ – множество вершин двойственного гиперграфа, $U^* = U(x)$ – семейство гиперребер двойственного гиперграфа для всякого $x \in X$. Одним из способов задания гиперграфа является (0, 1)-матрица инцидентности I, где 1 ставится в случае, когда гиперребро содержит вершину, и 0 в противном случае. Будем называть степенью гиперребра $u \in U$ мощность множества $|X(u)|$.

Приведем необходимые для дальнейшего изложения определения [19].

Определение 3. Подгиперграфом, индуцированным множеством вершин X', называется гиперграф $H' = (X', U')$, где $U' = \{u'\colon X(u') = X(u) \cap X' \neq \varnothing, u \in U\}$.

Заметим, что в рамках определения 3 для U' всегда справедливо высказывание: $|X(u) \cap X'| \geq 2$, или $|u'| = 1$.

Известно определение двудольности гиперграфа [20], которое аналогично 2-раскраске: гиперграф $H = (X, U)$ называется двудольным, когда множество вершин X может быть разделено на два множества S_0 и S_1 таким образом, что $S_0 \cup S_1 = X$, $S_0 \cap S_1 = \varnothing$, и для всякого гиперребра $u \in U$ выполнено $|X(u) \cap S_0| = 1$. В работе используется следующее определение двудольности подгиперафа.

Определение 4. Подгиперграф $H' = (X', U')$ является двудольным, если существует разбиение $S_0 \cup S_1 = X'$ такое, что $S_0 \cap S_1 = \varnothing$ и неравенства $|S_0 \cap X(u')| \leq 1$, $|S_1 \cap X(u')| \leq 1$ выполняются для любого гиперребра $u' \in U'$.

Приведем определение реализации гиперграфа в виде вершинного графа $L_2(H)$ [21].

Определение 5. Вершинным графом гиперграфа $H = (X, U)$ называется граф $L_2(H) = (X, E)$, множество вершин которого совпадает с множеством вершин X гиперграфа H, при этом две вершины графа $L_2(H)$ смежны тогда и только тогда, когда смежны соответствующие им вершины гиперграфа H.

Нетрудно показать, что подгиперграф $H' = (X', U')$ является двудольным тогда и только тогда, когда в вершинном графе $L_2(H)$ гиперграфа H существует двудольный подграф, индуцированный множеством вершин X'.

Определение 6. Двудольный граф называется полным двудольным графом (бикликой), если всякая вершина одной доли соединена со всеми вершинами второй доли [22].

Данное определение можно сформулировать иначе: если двудольный граф содержит все возможные ребра, не нарушающие условия двудольности, то такой граф называется полным двудольным графом. К поиску биклик сводится ряд теоретико-графовых задач [17], которые относятся к классу #*P*-полных или *NP*-полных задач, поскольку в общем случае число максимальных биклик экспоненциально зависит от размера графа [23].

Определение 7. Будем говорить, что подгиперграф $H' = (X', U')$, где $S_0 \cup S_1 = X'$, $S_0 \cap S_1 = \varnothing$ и $U' = \{u : s_0, s_1 \in X(u),\ s_0 \in S_0, s_1 \in S_1\}$, называется полным двудольным подгиперграфом (бикликой) исходного гиперграфа H.

Определение 8. Максимальная биклика – это биклика, которая не может быть расширена путем включения дополнительных смежных вершин, или, что эквивалентно, для максимальной биклики не существует другой биклики, которая полностью включает данную.

Определение 8 справедливо как для графов, так и для гиперграфов.

Задача поиска максимальной биклики хорошо известна в теории графов. При этом выделяют два варианта: задача поиска максимальной биклики с наибольшим числом вершин и задача поиска максимальной биклики с наибольшим числом ребер. Данные задачи возникают при обнаружении аномалий, анализе генных структур, анализе социальных структур [24]. Любой из вариантов данной задачи для связного графа произвольного вида является *NP*-трудной [25].

Помимо поиска одной максимальной биклики, не ослабевает интерес к задаче MBGP (Maximal Biclique Generation Problem), которая состоит в поиске всех возможных максимальных биклик для любого заданного графа. Известно, что MBGP не может быть решена за полиномиальное время относительно размера ввода, поскольку размер вывода может быть экспоненциально большим по n [25]. Сложность данной задачи не меньше, чем сложность задачи поиска одной максимальной биклики, которая является *NP*-трудной задачей [24].

Задача MBGP для графов может быть расширена на гиперграфы. В частности, для бигиперграфов задача MBGP исследована в [20]. Под бигиперграфом понимается гиперграф вида $H = (H^0, H^1)$, где всякое гиперребро гиперграфов H^0 и H^1 содержится в H. При этом вводится определение двудольного гиперграфа согласно устойчивости множества вершин. Множество вершин $S \subseteq V$ устойчиво в H^i, если S не содержит гиперребер из H^i, $i = 0, 1$. Гиперграф $H = (H^0, H^1)$ называется двудольным, если существует разбиение $S^0 \cup S^1 = V$, где S^i устойчиво в H^i, $i = 0, 1$.

В данной работе исследуется задача поиска всех возможных максимальных биклик для любого заданного гиперграфа.

Задача 1 (MBGP for Hypergraphs). Дан гиперграф $H = (X, U)$ без кратных гиперребер, требуется найти множество всех максимальных биклик.

Заметим, что задача нахождения биклик в графе тесно связана с поиском подматриц специального вида [11]. Проведем связь между задачей нахождения максимальных биклик в гиперграфе и задачей поиска максимально полных подматриц (0, 1)-матрицы.

Для двудольного подгиперграфа $H' = (X', U')$ запишем его эквивалентное представление в виде (0, 1)-матрицы смежности вершинного графа $L_2(H')$. Обозначим S_0, S_1 – доли подгиперграфа H', мощности которых равны c и d соответственно. Поскольку вершинный граф $L_2(H')$ тоже является двудольным, его матрица смежности представима в виде:

$$A' = \begin{pmatrix} O_c & B' \\ B'^{\mathrm{T}} & O_d \end{pmatrix}, \tag{1}$$

где O_c, O_d – нулевые матрицы порядка c и d соответственно, а B' – матрица размера $c \times d$, которая отражает смежность вершин между долями S_0 и S_1 [11]. Очевидно, что в случае, когда подгиперграф H'

является бикликой, матрица B является полной подматрицей матрицы A. Нетрудно показать, что проверка всякой подматрицы из множества всех максимально полных подматриц выполнима за линейное время относительно размера подматрицы. Таким образом, для поиска биклик в гиперграфе $H = (X, U)$ требуется найти такие полные подматрицы B' матрицы смежности A вершинного графа $L_2(H)$, для которых будет существовать подматрица вида (1), что приводит к задаче поиска всех максимально полных подматриц (maximally complete submatrices) исходной (0, 1)-матрицы.

Задача 2 (Maximally Complete Submatrices Problem, MCSP). Дана (0, 1)-матрица I, требуется найти множество всех максимально полных подматриц.

Заметим, что максимально полные подматрицы могут представлять различные комбинаторные объекты [8], а задача их поиска носит перечислительный характер и принадлежит классу сложности #*P*-полных задач [5, 8]. Применение алгоритмов решения задачи выделения максимально полных подматриц ограничивается высокой трудоемкостью на матрицах большой размерности.

2. Алгоритм нахождения всех максимально полных подматриц для заданной (0, 1)-матрицы

Основываясь на определениях теории гиперграфов и (0, 1)-матриц, опишем алгоритм для нахождения максимально полных подматриц исходной (0, 1)-матрицы. Для этого понадобится понятие l-уровня (0, 1)-матрицы.

Определение. Полным l-уровнем (0, 1)-матрицы будем называть все ее полные подматрицы с числом строк, равным l.

Всякую полную подматрицу инцидентности гиперграфа $H = (X, U)$ с числом строк, равным l, можно описывать как подгиперграф $H' = (X', U(X'))$, где $X' \subseteq X$ – множество вершин, при этом $|X'| = l$, а $U(X') \subseteq U$ – множество гиперребер. Следовательно, l-уровень в терминах гиперграфов описывается следующей системой множеств: $P_l = \{H' = (X', U(X')) : |X'| = l, X' \subseteq X\}$. Также отметим, что гиперграфу H соответствует (0, 1)-матрица инцидентности I с заданным лексикографическим порядком как для строк, так и для столбцов.

Заметим, что в общем случае всякую (0, 1)-матрицу можно представить в виде гиперграфа независимо от ее семантики, и наоборот. Обозначим множество всех максимально полных подматриц как *MCS*. Идея алгоритма заключается в генерации всех возможных l-уровней (0, 1)-матрицы и последующем выделении максимально полных подматриц. Иерархия уровней позволяет с легкостью проверять. является ли полная подматрица максимальной. Данная идея хорошо реализуется на языке гиперграфов. Приведем описание алгоритма *HFindMCS* и его вспомогательной функции *GenerateCombinations*(H, l) на псевдокоде.

Алгоритм *HFindMCS*
Вход: гиперграф $H = (X, U)$
Выход: множество *MCS*
Для $l := 1, \ldots, \Delta$ **выполнять**
$P_l := GenerateCombinations(H, l)$
Конец цикла
$P := \bigcup_{l=\Delta,\ldots,1} P_l$
$MCS := \{\varnothing\}$
Для $(Y, U(Y)) \in P$ **выполнять**
Если $U(Y) \neq U(Y') : \forall(Y', U(Y')) \in MCS$ **тогда**
$MCS := MCS \cup (Y, U(Y))$
Конец условия
Конец цикла
Возвратить *MCS*

Функция *GenerateCombinations* (H, l)
$P_l := \{\varnothing\}$
Для $u \in U$ **выполнять**
$C_u^l := \{X' : X' \subseteq X(u), \lvert X' \rvert = l\}$
Для $X' \in C_u^l$ **выполнять**
$P_l[X'] := P_l[X'] \cup u$
Конец цикла
Конец цикла
Возвратить P_l

Теорема 1. Алгоритм *HFindMCS* корректно решает задачу MCSP для (0, 1)-матрицы инцидентности I гиперграфа $H = (X, U)$ со степенью Δ за время, не превышающее $O\left[\log\left(|MCS|\right)|U|^2\,\Delta 2^\Delta \log\left(2^\Delta |U|\right)\right]$.

Доказательство. Алгоритм *HFindMCS* принимает на вход гиперграф $H = (X, U)$ без кратных гиперребер и вершин. Для простоты изложения будем считать, что максимальная степень гиперребра меньше максимальной степени вершины, т.е. $\max\limits_{u\in U}|X(u)| \le \max\limits_{x\in X}|U(x)|$. Это условие необходимо для быстродействия алгоритма, в общем случае алгоритм можно применять к любому гиперграфу без кратных гиперребер. Если условие не выполнено, то алгоритм *HFindMCS* применяется к двойственному гиперграфу $H^* = (X^*, U^*)$. Далее будем обозначать максимальную степень гиперребра как $\Delta = \max\limits_{u\in U}|X(u)|$ и называть степенью гиперграфа [19].

На шагах 1–3 последовательно формируются множества P_l, соответствующие полным l-уровням, где $l = 1, \ldots, \Delta$. Для каждого значения l, соответствующего требуемому уровню (0, 1)-матрицы, на шаге 2 вызывается функция *GenerateCombinations*(H, l). Вызываемая функция для каждого гиперребра $u \in U$ выполняет генерацию всех возможных подмножеств вершин $X' \subseteq X(u)$ таких, что $|X'| = l$. Множество всех возможных подмножеств l-уровня обозначим как C_u^l. Каждое сгенерированное множество X' индуцирует исходный гиперграф H в подгиперграф $H' = (X', U')$. Построение множества U' происходит последовательно путем просмотра всех гиперребер $u \in U$. Исходя из этого, множество U' индуцированного подгиперграфа формируется в лексикографическом порядке. Полученный подгиперграф H' однозначно сопоставим с полной подматрицей матрицы I, а множество всех таких подгиперграфов образует полный l-уровень (0, 1)-матрицы, т.е. множество P_l.

Выполняя данную процедуру для каждого уровня $l = 1, \ldots, \Delta$, получим набор множеств P_l, содержащих подгиперграфы, матрицы инцидентности которых соответствуют всем полным подматрицам матрицы инцидентности I гиперграфа. Очевидно, что полные подматрицы с наибольшим количеством строк, равным Δ, будут являться максимально полными, поскольку они гарантированно не вложены ни в какие другие полные подматрицы.

При формировании набора множеств P_l процесс генерации сочетаний для всех уровней не превосходит $\tau_1 = O(|U|2^\Delta)$. Действительно, поскольку на каждом уровне необходимо генерировать сочетания для каждого гиперребра и количество уровней $l = 1, \ldots, \Delta$, то

$$|U|\,C_u^1 + \ldots + |U|\,C_u^\Delta = |U|\left(C_u^1 + \ldots + C_u^\Delta\right) = |U|\left(2^\Delta - 1\right) = O\left(|U|2^\Delta\right).$$

Отметим, что всякое множество $X' \in C_u^l$ является лексикографически упорядоченным, и любые X_1', X_2' из C_u^l являются сравнимым. Эти два факта позволяют для хранения и обновления множеств P_l использовать древовидные структуры данных, в частности красно-черные деревья. Очевидно, что сравнение X_1' и X_2' выполнимо за $O(\Delta)$, так как мощность всякого X' не превосходит Δ. А поиск и добавление элементов в красно-черном дереве не превосходят $\tau_2 = O(\Delta\log(|U|2^\Delta))$ [26]. Следовательно, выполнение шагов 1–3 алгоритма *HFindMCS* требует не более

$$T_1 = \tau_1\tau_2 = O\left[|U|\Delta 2^\Delta \log\left(|U|2^\Delta\right)\right].$$

На шаге 4 алгоритма *HFindMCS* полученный на шагах 1–3 набор множеств P_l объединяется в упорядоченное множество P следующим образом: объединение осуществляется в порядке убывания значения уровня l с сохранением у каждого подгипреграфа лексикографического порядка на множестве вершин и на множестве гиперребер. Следовательно, формирование множества P выполнимо за время равное $T_2 = O(1)$.

Множество P является допустимым множеством решений задачи поиска всех максимально полных подматриц и соответствующих им гиперграфов. Формирование множества *MCS* осуществляется на шагах 5–10 путем просматривания множества P от подгиперграфов с наибольшей мощностью множества вершин до обладающих наименьшей мощностью. Изначально в *MCS* добавляются подгиперграфы с числом вершин, равным Δ, из множества *P*. Такие подгиперграфы всегда будут соответствовать максимально полным подматрицам. Далее последовательно просматриваются остальные элементы множества *P*, и в множество *MCS* добавляются только такие подгиперграфы, множества гиперребер которых еще не встречались. После этого множество MCS будет содержать все подгиперграфы, соответствующие максимально полным подматрицам матрицы инцидентности *I*. Следовательно, алгоритм корректно решает задачу MCSP.

Заметим, что сравнение подгиперграфов по множеству гиперребер требует не более $O(|U|)$ времени. Это следует из того, что максимальная мощность множества гиперребер подгиперграфа не превосходит $|U|$. Поскольку по построению все комбинации вершин для подгиперграфов в множестве *P* различны, то соответствующие им полные подматрицы могут содержаться в других только в случае, когда множество гиперребер совпадает. Это соответствует тому, что полная подматрица содержится в полной подматрице с большим числом строк. Для хранения и обработки множества *MCS* в работе используются красно-черные деревья, что облегчает поиск и добавление новых элементов. Последовательно просматривая множество *P*, для каждого элемента выполняется поиск в множестве *MCS* за время $T_3 = O\left[|U|\log\left(|MCS|\right)\right]$, аналогичное время требуется для добавления нового элемента. Таким образом построение множества *MCS* требует времени не более чем

$$T_1T_2T_3 = O\left[\log\left(|MCS|\right)|U|^2\,\Delta 2^{\Delta}\log\left(2^{\Delta}|U|\right)\right].$$

Теорема доказана.

Искомому множеству *MCS* будет соответствовать множество подгипреграфов, матрицы инцидентности которых максимально полные. Отметим, что время работы алгоритма *HFindMCS* существенно зависит от параметра Δ, который соответствует максимальной степени гиперребра или вершины.

Существует множество алгоритмов для нахождения максимально полных подматриц [6]. Всем известным алгоритмам нахождения комбинаторных объектов, связанных с максимально полными подматрицами, присущ комбинаторный взрыв на выходе в виде множества $|MCS|$. Наиболее универсальным является широко известный алгоритм нахождения всех формальных понятий формального контекста Close-by-One (*CbO*) [6]. Асимптотическая сложность данного алгоритма оценивается как $O(|MCS||U||X|^2)$. В общем случае предложенный алгоритм *HFindMCS* асимптотически медленнее алгоритма *CbO*, однако для наборов данных, организованных определенным образом [9], он будет эффективнее алгоритма *CbO*. В частности, для квадратных матриц алгоритм *HFindMCS* будет асимптотически быстрее алгоритма *CbO* при выполнении условия $\Delta 2^{\Delta}\log(2^{\Delta}|X|) \le |X|$.

Решением задачи MBGP for Hypergraphs является множество максимальных биклик гиперграфа $H = (X, U)$. Обозначим множество максимальных биклик как *BC*. Применение алгоритма *HFindMCS* к матрице смежности вершинного графа $L_2(H)$ дает на выходе множество всех максимально полных подматриц *MCS* в виде подгиперграфов. Поскольку алгоритм *HFindMCS* применяется к матрице смежности, то всякий подгиперграф $H' = (X', U')$ из множества *MCS* фактически отражает пару множеств вершин X' и U'. Кратко опишем процедуру нахождения множества *BC* на основе множества *MCS*. Если элемент множества *MCS* удовлетворяет условию (1), то он является бикликой, в противном случае происходит расщепление элемента на два по тем вершинам, которые смежны внутри доли X' или U', чтобы убрать смежность. После этого необходимо выбрать максимальные биклики из полученного множества биклик.

3. Вычислительные эксперименты и сравнение алгоритмов

В работе проводились две серии экспериментов. Первая серия экспериментов направлена на оценку результативности предложенного алгоритма *HFindMCS* решения задачи MCSP. Вторая серия экспериментов призвана оценить эффективность процедуры перехода от решения задачи MCSP к решению задачи MBGP for Hypergraphs. Вычислительные эксперименты проводились на ПК с процессором AMD Ryzen 5 3600 6-Core Processor 3,60 ГГц и ОЗУ объемом 16 Гбайт.

Эксперименты первой серии проводились на гиперграфах с различным числом вершин $|X|$ и гиперребер $|U|$, а также различной степенью гиперграфа Δ. Результаты вычислительных экспериментов представлены в табл. 1.

Таблица 1

Результаты вычислительных экспериментов алгоритма *HFindMCS*

Δ	$\|X\|$	$\|U\|$	$\|MCS\|$	t, с
5	100	100	276	0,002
	1 000	1 000	2 055	0,012
	10 000	9 904	19 609	0,145
	25 000	24 540	48 220	0,421
10	100	100	1 199	0,041
	1 000	1 000	3 786	0,422
	10 000	9 999	22 037	4,938
	25 000	24 989	52 501	13,571
13	100	100	2 570	0,339
	1 000	1 000	7 056	3,521
	10 000	10 000	26 102	38,091
	25 000	24 999	57 420	497,621

Таблица 1 хорошо демонстрирует сложность задачи поиска всех максимально полных подматриц. Число максимально полных подматриц соответствует столбцу $|MCS|$ и, как видно, даже при малом значении степени гиперграфа Δ оно достаточно велико. Учитывая, что все алгоритмы точного решения задачи поиска всех максимально полных подматриц зависят от мощности выходного множества *MCS*, алгоритм *HFindMCS* обладает приемлемым временем работы относительно размера исходного гиперграфа.

Для второй серии экспериментов были сгенерированы гиперграфы $H = (X, U)$ с фиксированной степенью вершин. Для гиперграфов строились вершинные графы $L_2(H)$. К матрицам смежности вершинного графа $L_2(H)$ применялся алгоритм *HFindMCS* для нахождения множества *MCS*. Далее к множеству *MCS* применялась процедура нахождения всех максимальных биклик. В табл. 2 приводятся параметры гиперграфа H, включая максимальную степень вершины Δ', а также время нахождения множества *BC*.

Таблица 2

Результаты вычислительных экспериментов расщепления множества *MCS*

Δ'	$\|X\|$	$\|U\|$	$\|MCS\|$	$\|BC\|$	t, с
3	100	90	202	191	0,040
	500	439	2 055	961	0,863
5	100	76	476	453	1,366
	500	362	2 506	2 208	40,848
7	100	73	1 234	757	30,132
	500	317	5 738	4 343	653,944

Из табл. 2 видно, что процедура нахождения всех максимальных биклик из множества *MCS* требует значительных временных затрат. Это связано с тем, что подгиперграфы, соответствующие максимально полным подматрицам, не обязательно являются бикликами. Из-за этого процедура расщепления может вызвать геометрический рост количества проверяемых подгиперграфов.

Заключение

В работе исследуются две задачи: поиск всех максимально полных подматриц (0, 1)-матрицы (MCSP) и поиск всех максимальных биклик гиперграфов (MBGP for Hypergraphs). Для решения задачи MCSP разработан гиперграфовый алгоритм *HFindMCS*. Установлено, что время формирования множества *MCS* алгоритмом *HFindMCS* существенно зависит от входного гиперграфа и составляет $O\left[\log\left(|MCS|\right)|U|^2\Delta 2^{\Delta}\log\left(2^{\Delta}|U|\right)\right]$. Разработанный алгоритм имеет высокую вычислительную сложность относительно известных алгоритмов. Однако использование данного алгоритма может быть оправдано тем, что для гиперграфов, обладающих малой степенью, он будет значительно эффективнее, что подтверждено вычислительными экспериментами для специального вида гиперграфов.

Предложена процедура перехода от решения задачи MCSP к решению задачи MBGP for Hypergraphs, идея которой заключается в расщеплении элементов множества *MCS* и проверке их на соответствие специальному виду. Вычислительные эксперименты показали высокую трудоемкость перехода от решения задачи MCSP к решению задачи MBGP for Hypergraphs. Для повышения производительности процедуры построения решения задачи MBGP for Hypergraphs предполагается применение описанной процедуры перехода к промежуточным множествам P_l, формируемым в процессе работы алгоритма *HFindMCS*. Такая модификация позволит сократить необходимое число расщеплений и тем самым повысить производительность алгоритма. Также стоить отметить, что для алгоритма *HFindMCS* перспективны исследования по применению технологий параллельных вычислений для улучшения времени его работы.

ЛИТЕРАТУРА

1. Мейер Д. Теория реляционных баз данных. М. : Мир, 1987. 608 с.
2. Konstantinova E.V., Skorobogatov V.A. Application of hypergraph theory in chemistry // Discrete Mathematics, 2001. V. 235, is. 1–3. P. 365–383.
3. Graham R.L., Pollak H.O. On the addressing problem for loop switching // The Bell System Technical Journal. 1971. V. 50, № 8. P. 2495–2519.
4. Попов В.Б. Экстремальная нумерация вершин гиперграфа и задача объектно-признаковой кластеризации // Динамические системы. 2010. № 28. С. 99–112.
5. Ganter B., Wille R. Formal concept analysis. Springer, 1999. 284 p.
6. Kuznetsov S., Obiedkov S. Comparing performance of algorithms for generating concept lattices // J. Exp. Theor. Artif. Intell. 2002. V. 14. P. 189–216.
7. Блюмин С.Л. Полные гиперграфы. Спектры лапласианов. Мультиагентные системы // Управление большими системами, 2010. № 30. С. 5–23.
8. Тараканов В.Е. Комбинаторные задачи и (0, 1)-матрицы. М. : Наука, 1985. 195 с.
9. Миллер Р., Боксер Л. Последовательные и параллельные алгоритмы: общий подход. М. : БИНОМ. Лаборатория знаний, 2009. 406 с.
10. Маркус М., Минк Х. Обзор по теории матриц и матричных неравенств. М. : Наука, 1972. 232 с.
11. Beineke L., Wilson R.J. Topics in Algebraic Graph Theory. Cambridge ; New York : Cambridge University Press, 2008. 276 p. (Mathematical Sciences Faculty Publications).
12. Jukna S., Kulikov A. On covering graphs by complete bipartite subgraphs // Discrete Mathematics. 2009. V. 309. P. 3399–3403.
13. Faure N., Chretienne P., Gourdin E., Sourd F. Biclique completion problems for multicast network design // Discrete Optimization. 2007. V. 4, is. 3–4. P. 360–377.
14. Hermelin D., Manoussakis G. Efficient enumeration of maximal induced bicliques // Discrete Applied Mathematics. 2020. DOI: 10.1016/j.dam.2020.04.034
15. Shaham E., Yu H., Li X. On finding the maximum edge biclique in a bipartite graph: a subspace clustering approach // Proc. of the 2016 SIAM International Conference on Data Mining (SDM). 2016. P. 315–323.
16. Damaschke P. Enumerating maximal bicliques in bipartite graphs with favorable degree sequences // Inf. Process. Lett. 2014. V. 114, is. 6. P. 317–321.
17. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. М. : Мир, 1982. 416 с.
18. Кофман А. Введение в прикладную комбинаторику. М. : Наука, 1975. 480 с.
19. Bretto A. Hypergraph theory: an introduction. Springer, 2013. 119 p.
20. Zverovich I., Zverovich I. Bipartite bihypergraphs: A survey and new results // Discrete Mathematics. 2006. V. 306. P. 801–811.
21. Емеличев В.А., Мельников О.И., Сарванов В.И., Тышкевич Р.И. Лекции по теории графов. М. : Наука, 1990. 384 с.
22. Харари Ф. Теория графов. М. : URSS, 2018. 304 с.
23. Prisner E. Bicliques in graphs I: bounds on their number // Combinatorica. 2000. V. 20. P.109–117.
24. Lyu B., Qin L., Lin X., Zhang Y., Qian Z., Zhou J. Maximum biclique search at billion scale // Proc. VLDB Endow. 2020. P. 1359–1372.
25. Alexe G., Alexe S., Crama Y., Foldes S., Hammer P.L., Simeone B. Consensus algorithms for the generation of all maximal bicliques // Discrete Applied Mathematics. 2004. V. 145. P. 11–21.

26. Кормен Т.Х., Лейзерсон Ч.И., Ривест Р.Л., Штайн К. Алгоритмы: построение и анализ. 3-е изд. М. : Вильямс, 2013. 1328 с.

Поступила в редакцию 10 апреля 2021 г.

Soldatenko A.A., Semenova D.V. (2021) ON CONNECTION BETWEEN BICLIQUES OF HYPERGRAPH AND ALL MAXIMALLY COMPLETE SUBMATRICES OF ADJACENCY MATRIX. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitelnaja tehnika i informatika* [Tomsk State University Journal of Control and Computer Science]. 56. pp. 90–99

DOI: 10.17223/19988605/56/10

Nowadays hypergraph models are actively used in modeling and researching road and telecommunication networks, design semantic networks and representing linguistic knowledge, when solving problems of control of large systems.

In this paper we investigate connection between problem of finding all maximally complete submatrices of (0, 1)-matrix and problem of finding all maximal bicliques of hypergraph. Problem of finding all maximally complete submatrices of (0, 1)-matrix is in demand in the analysis of formal context. This problem arises in searching for hidden relationships in subject areas and building ontologies of subject areas. Finding bicliques in networks represented by graph and hypergraph can increase the efficiency of solving the problem of addressing and routing. It is known that problem of finding all maximally complete submatrices of (0, 1)-matrix is #*P*-complete, and problem of finding all maximal bicliques at least hard as *NP*-hard problems. We propose new hypergraph algorithm *HFindMCS* for searching all maximally complete submatrices of (0, 1)-matrix. In the algorithm initial (0, 1)-matrix is represented as a hypergraph H. We introduce new definition of *l*-level for algorithm are given. By the complete *l*-level of (0, 1)-matrix we mean all its complete submatrices with the number of rows equal to *l*. Main idea of algorithm *HFindMCS* is to generate all possible *l*-levels of (0, 1)-matrix and then select maximally complete submatrices from them. Moreover, number of *l*-levels does not exceed the degree of the hypergraph. Level hierarchy makes it easy to check if complete submatrix is maximal. This idea is well implemented in the language of hypergraphs. We proved that algorithm *HFindMCS* finds all maximally complete submatrices of incidence (0, 1)-matrix *I* of hypergraph *H* in time $O\left[\log\left(|MCS|\right)|U|^2\Delta 2^{\Delta}\log\left(2^{\Delta}|U|\right)\right]$. The efficiency of the *HFindMCS* algorithm has been confirmed by computational experiments for hypergraphs with limited degree.

A procedure for transition from solution of the problem of finding all maximally complete submatrices of the (0, 1)-matrix (MCSP) to solution of the problem of finding the maximum bicliques of hypergraph (MBGP for Hypergraphs) is proposed. The idea of the procedure consists in splitting elements of the set of all maximally complete submatrices and checking for satisfaction to a special form of the matrix. A special form of matrix is associated with the adjacency matrix of the hypergraph. Computational experiments have shown the high complexity of the transition from solving the MCSP problem to solving the MBGP for Hypergraphs problem. To improve the performance of procedure for constructing a solution of MBGP for Hypergraphs problem, we assume to apply the described procedure to sets P_l, which are formed in *HFindMCS* algorithm. This modification can decrease a number of needed splits hence increase overall algorithm performance. It is also worth noting that researches on using of parallel computing technologies for the *HFindMCS* algorithm are perspective and will allow to reduce it operation time.

Keywords: hypergraph; maximal bicliques; (0, 1)-matrix; maximally complete submatrices.

SOLDATENKO Aleksandr Aleksandrovich (Post-graduate Student, Siberian Federal University, Krasnoyarsk, Russian Federation).
E-mail: ASoldatenko@sfu-kras.ru

SEMENOVA Darya Vladislavovna (Candidate of Physics and Mathematics, Associate Professor, Siberian Federal University, Krasnoyarsk, Russian Federation).
E-mail: DVSemenova@sfu-kras.ru

REFERENCES

1. Meyer, D. (1987) *Teoriya relyatsionnykh baz dannykh* [The Theory of Relational Databases]. Moscow: Mir.
2. Konstantinova, E.V. & Skorobogatov, V.A. (2001) Application of hypergraph theory in chemistry. *Discrete Mathematics*. 235(1–3). pp. 365–383. DOI: 10.1016/S0012-365X(00)00290-9
3. Graham, R.L. & Pollak, H.O. (1971) On the addressing problem for loop switching. *The Bell System Technical Journal*. 50(8). pp. 2495–2519. DOI: 10.1002/j.1538-7305.1971.tb02618.x
4. Popov, V.B. (2010) Ekstremal'naya numeratsiya vershin gipergrafa i zadacha ob"ektno-priznakovoy klasterizatsii [Extreme enumeration of the hypergraph vertex and the box clusterization problem]. *Dinamicheskie sistemy.* 28. pp. 99–112.
5. Ganter, B. & Wille, R. (1999) *Formal Concept Analysis*. Springer.
6. Kuznetsov, S. & Obiedkov, S. (2002) Comparing performance of algorithms for generating concept lattices. *Journal of Experimental & Theoretical Artificial Intelligence.* 14. pp. 189–216. DOI: 10.1080/09528130210164170

7. Blyumin, S.L. (2010) Complete Hypergraphs. Laplasian spectra. Multi-agent systems. *Upravlenie bol'shimi sistemami – Large-Scale Systems Control.* 30. pp. 5–23.
8. Tarakanov, V.E. (1985) *Kombinatornye zadachi i (0, 1)-matritsy* [Combinatorial problems and (0, 1) -matrices]. Moscow: Nauka.
9. Miller, R. & Boxer, L. (2009) *Posledovatel'nye i parallel'nye algoritmy: obshchiy podkhod* [Algorithms Sequential & Parallel. A Unified Approach]. Translated from English. Moscow: BINOM. Laboratoriya znaniy.
10. Marcus, M. & Minc, H. (1972) *Obzor po teorii matrits i matrichnykh neravenstv* [Matrix Theory and Matrix Inequalities]. Moscow: Nauka.
11. Beineke, L. & Wilson, R.J. (2008) *Topics in Algebraic Graph Theory*. Cambridge; New York : Cambridge University Press.
12. Jukna, S. & Kulikov, A. (2009) On covering graphs by complete bipartite subgraphs. *Discrete Mathematics*. 309. pp. 3399–3403. DOI: 10.1016/j.disc.2008.09.036
13. Faure, N., Chretienne, P., Gourdin, E. & Sourd, F. (2007) Biclique completion problems for multicast network design. *Discrete Optimization*. 4(3–4). pp. 360–377. DOI: 10.1016/j.disopt.2007.09.005
14. Hermelin, D. & Manoussakis, G. (2020) Efficient enumeration of maximal induced bicliques. *Discrete Applied Mathematics*. 303(1). DOI: 10.1016/j.dam.2020.04.034
15. Shaham, E., Yu, H. & Li, X. (2016) On finding the maximum edge biclique in a bipartite graph: a subspace clustering approach. *Proc. of the 2016 SIAM International Conference on Data Mining (SDM)*. pp. 315–323.
16. Damaschke, P. (2014) Enumerating maximal bicliques in bipartite graphs with favorable degree sequences. *Inf. Process. Lett.* 114(6). pp. 317–321. DOI: 10.1016/j.ipl.2014.02.001
17. Garey, M. & Johnson, D. (1982) *Vychislitel'nye mashiny i trudnoreshaemye zadachi* [Computers and Intractability]. Translated from English. Moscow: Mir.
18. Kaufmann, A. (1975) *Vvedenie v prikladnuyu kombinatoriku* [Introduction to applied combinatorics]. Translated from French. Moscow: Nauka.
19. Bretto, A. (2013) *Hypergraph Theory: An Introduction*. Springer.
20. Zverovich, I. & Zverovich, I. (2006) Bipartite bihypergraphs: A survey and new results. *Discrete Mathematics*. 306. pp. 801–811. DOI: 10.1016/j.disc.2005.10.026
21. Emelichev, V.A., Melnikov, O.I., Sarvanov, V.I. & Tyshkevich, R.I. (1990) *Lektsii po teorii grafov* [Lectures on Graph Theory]. Moscow: Nauka.
22. Harary, F. (2018) *Teoriya grafov* [Graph Theory]. Translated from English. Moscow: URSS.
23. Prisner, E. (2000) Bicliques in graphs I: bounds on their number. *Combinatorica*. 20. pp. 109–117.
24. Lyu, B., Qin, L., Lin, X., Zhang, Y., Qian, Z. & Zhou, J. (2020) Maximum biclique search at billion scale. *Proc. VLDB Endow*. pp. 1359–1372.
25. Alexe, G., Alexe, S., Crama, Y., Foldes, S., Hammer, P.L. & Simeone, B. (2004) Consensus algorithms for the generation of all maximal bicliques. *Discrete Applied Mathematics*. 145. pp. 11–21. DOI: 10.1016/j.dam.2003.09.004
26. Cormen, T.H., Leiserson, C.E., Rivest, R.L. & Stein, C. (2013) *Algoritmy: postroenie i analiz* [Introduction to Algorithms]. 3rd. ed. Translated from English. Moscow: Vil'yams.