УДК 519.683.6

# МЕТОД ПОСТРОЕНИЯ СЕМЕЙСТВА МАКСИМАЛЬНЫХ ТРАНЗИТИВНО НЕЗАВИСИМЫХ МНОЖЕСТВ АТРИБУТОВ

**И.В. Клименко**

*ФГБОУ ВПО «Ростовский государственный университет путей сообщения», г. Ростов-на-Дону*

*Рецензент д-р техн. наук, профессор С.М. Ковалев*

**Ключевые слова и фразы:** аксиомы вывода функциональных зависимостей; алгоритм Брона–Кэрбоша; задача о наименьшем покрытии; максимальные независимые транзитивные множества; нормализация отношений реляционной модели; нормальные формы; первичный ключ; функциональные зависимости.

**Аннотация:** Проведен анализ всех способов решения задачи нахождения максимальных транзитивно независимых множеств. Определен оптимальный способ решения данной задачи – метод перебора Брона–Кэрбоша, который в процессе решения задачи был модифицирован с целью устранения недостатков.

На первый взгляд, кажется, что нахождение всех максимальных транзитивно независимых множеств (**МТНМ**) – простая задача, которую можно решить простым перебором ТНМ с одновременной проверкой их на максимальность. Последнее действие обычно осуществляется путем добавления к исследуемому ТНМ дополнительного атрибута, не принадлежащего ему, и выяснения того, сохраняется ли независимость после применения аксиомы транзитивности F5.

Представление о простоте задачи справедливо только для небольших графов (в простых предметных областях) с числом атрибутов до 20. Однако с увеличением количества атрибутов этот метод поиска становится с вычислительной точки зрения громоздким. Такой вывод можно получить, если исследовать зависимость количества клик от количества вершин на графах Муна–Мозера [1]. Введенное понятие «максимального транзитивно независимого множества» является противоположным для понятия

Клименко Игорь Валерьевич – кандидат технических наук, доцент кафедры «Вычислительная техника и автоматизированные системы управления» e-mail: klimigor67@mail.ru, ФГБОУ ВПО «Ростовский государственный университет путей сообщения», г. Ростов-на-Дону.

«клика». Поэтому методы, применяемые для исследования клик на графах, применимы и для исследования ТНМ.

Число клик в графе может расти экспоненциально относительно числа вершин. Рассмотрим граф $M_n$ Муна–Мозера с $3n$ вершинами $\{1, 2, \ldots, 3n\}$, в котором вершины разбиты на триады $\{1, 2, 3\}$, $\{4, 5, 6\}$, …, $\{3n - 2, 3n - 1, 3n\}$; $M_n$ не имеет ребер внутри любой триады, но вне них каждая вершина связана с каждой из остальных. Графы $M_1$, $M_2$, $M_3$ показаны на рис. 1.

Легко доказать, что $M_n$ имеет $3^n$ клик, каждая из которых содержит $n$ вершин. Это верно для $M_1$, в котором кликами являются сами вершины. Если $M_{n-1}$ имеет $3^{n-1}$ клик, каждый из которых состоит из $(n - 1)$ вершин, то каждая из трех вершин, добавленных для построения $M_n$, формирует клику с каждой из $3^{n-1}$ клик $M_{n-1}$. Поскольку только они являются новыми кликами, $M_n$ имеет $3 \cdot 3^{n-1} = 3^n$ клик, каждая из которых состоит из $n$ вершин. Таким образом, число клик в $M_n$ растет экспоненциально относительно числа вершин. Отсюда можем сделать вывод, что количество МТНМ растет экспоненциально относительно числа атрибутов, если граф исследуемой предметной области представляет собой дополнение к графу Муна–Мозера.

Поскольку число МТНМ может быть очень велико, строить их следует аккуратно. Каждое МТНМ должно порождаться только один раз, чтобы не тратилось время на повторную работу.

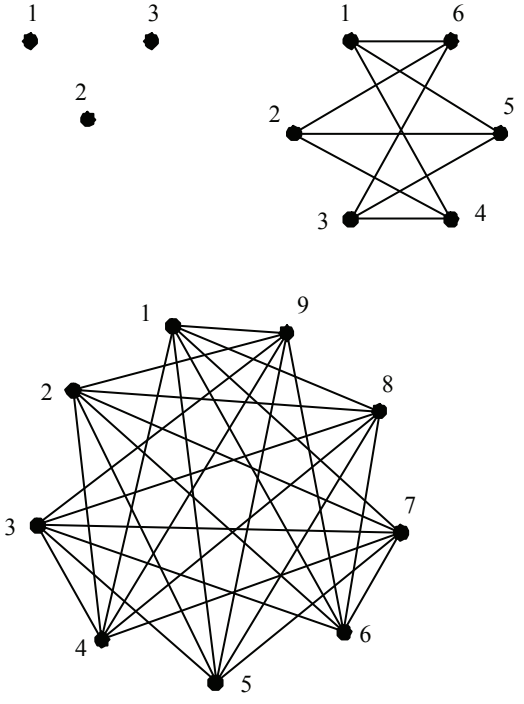


**Рис. 1. Первые три графа Муна–Мозера**

Вначале рассмотрим поиск максимальных транзитивно независимых множеств с возвращением, то есть поиск, в котором не делается никаких попыток упростить дерево поиска. Каждый узел в дереве поиска соответствует некоторому ТНМ, и каждое ребро соответствует вершине графа. Сын данного узла $C$ получается добавлением к $C$ вершины $x \notin C$, которая несмежна с каждой вершиной из $C$. Ребро, идущее от $C$ к сыну $C \cup \{x\}$, соответствует вершине $x$. На рис. 2 показаны некоторый граф $G$ и дерево поиска $T$, которое находится в процессе естественного поиска с возвращением. Заметим, что каждое максимальное ТНМ порождается много раз: каждое из подмножеств {1, 2, 3} и {3, 5, 6} порождается шесть раз, а подмножество {2, 4} порождается дважды; в общем случае максимальный ТНМ размера $k$ порождается $k!$ раз. На рис. 2 все ребра, изображенные тонкими линиями, можно оборвать, если воспользоваться теоремами 1 и 2 по аналогии с [1].

**Теорема 1.** Пусть $S$ – узел в дереве поиска $T$ (то есть $S$ есть подмножество вершин графа $G$, которое индуцирует ТНМ, представленные графом $G$), и пусть первый сын узла $S$ в дереве $T$, который надо исследовать, является множеством $S \cup \{x\}$ (то есть вершина $x$ несмежна с каждой вершиной из $S$). Предположим, что все поддеревья узла $S \cup \{x\}$ в дереве $T$ уже исследованы и порождены все максимальные ТНМ, включающие $S \cup \{x\}$. Тогда необходимо исследовать только те из сынов $S \cup \{v\}$, для которых $v \in \Gamma(x)$ (рис. 3).

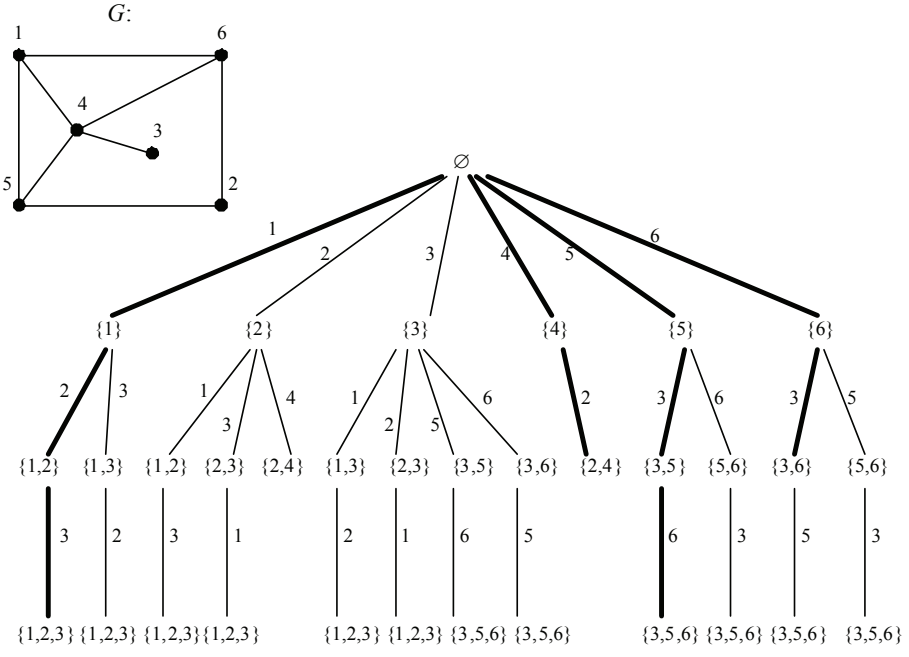


**Рис. 2. Граф $G$ и результат поиска с возвращением без ограничений ТНМ**
(ребра, которые можно оборвать, применяя теоремы 1 и 2, показаны тонкой линией)

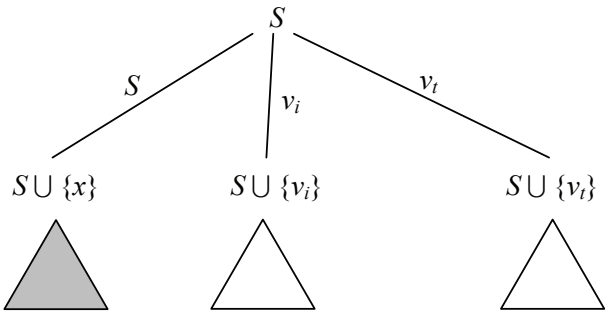


**Рис. 3. В соответствии с теоремой 1 поддерево с корнем в $S \cup \{v_i\}$ не нужно исследовать, если поддерево с корнем в $S \cup \{x\}$ уже исследовалось и $v_i \notin \Gamma(x)$**

*Доказательство.* Пусть $C$ – максимальное ТНМ, порожденное в процессе исследования поддерева с корнем $S \cup \{v_i\}$, где $v_i \notin \Gamma(x)$. Очевидно $S \subseteq C$, и если $C$ содержит вершину $v_j \in \Gamma(x)$, то $C$ будет найдено при исследовании поддерева, выходящего из $S \cup \{v_j\}$ ($S \subseteq \Gamma(x)$, $v_j \notin S$). Если транзитивно независимое множество $C$ не содержит таких вершин, то оно должно быть найдено при исследовании поддерева с корнем $S \cup \{x\}$.■

Следует отметить, что теорему 1 нельзя повторно применять к дереву, когда исследуются сыновья узла $S$, то есть после применения теоремы для обрывания дерева с корнем в $S \cup \{v_i\}$ (см. рис. 3) при условии $v_i \notin \Gamma(x)$ нельзя оборвать поддерево с корнем в $S \cup \{v_k\}$, когда $v_k \notin \Gamma(v_j)$ для некоторой вершины $v_j \in \Gamma(x)$. Другими словами, теорема 1 применяется только к первому исследуемому сыну $S$ и неприменима к оставшимся его сыновьям. Но, имеет место следующий результат [1].

**Теорема 2.** Пусть $S$ – узел в дереве поиска $T$, и пусть $\hat{S} \subset S$ – собственный предок $S$ в $T$. Если все поддеревья узла $\hat{S} \cup \{x\}$ уже исследованы, так что порождены все максимальные ТНМ, включающие $\hat{S} \cup \{x\}$, то все неисследованные поддеревья с корнями $S \cup \{x\}$ можно проигнорировать.

Задача нахождения ТНМ в множестве атрибутов предметной области аналогична задаче построения независимых множеств некоторого графа. Независимое множество вершин (известное также как внутренне устойчивое множество [2]) есть множество вершин неориентированного графа $G = (X, A)$, такое, что любые две вершины в нем несмежны, то есть никакая пара вершин не соединена ребром. Следовательно, любое множество $S \subset X$, которое удовлетворяет условию

$$S \cap \Gamma(S) = \varnothing, \tag{1}$$

является независимым множеством вершин.

**Теорема 3.** Задача определения того, содержит ли неориентированный граф $G = (X, A)$ независимое множество с $k$ вершинами, является *NP*-полной.

Для доказательства этой теоремы воспользуемся следующими определениями [3].

***Определение 1.*** Задача $P_1$ преобразуется в задачу $P_2$, если любой частный случай задачи $P_1$ можно преобразовать за полиномиальное время в некоторый частный случай задачи $P_2$, так что решение задачи $P_1$ можно получить за полиномиальное время из решения этого частного случая $P_2$.

Это понятие схематически проиллюстрировано на рис. 4. Ясно, что если $P_1$ преобразуется в $P_2$ и $P_2 \in \Pi$, то $P_1 \in \Pi$.

***Определение 2.*** Задача является *NP*-трудной, если каждая задача из *NP* преобразуется в нее, и задача является *NP*-полной, если она одновременно является *NP*-трудной и входит в класс *NP*.

Поэтому для доказательства того, что задача является *NP*-трудной, необходимо доказать, что некоторая *NP*-трудная задача преобразуется в нее. Трудность состоит в том, чтобы изначально установить, что некоторая частная задача является *NP*-трудной; затем можно использовать эту задачу для доказательства *NP*-трудности других задач.

*Доказательство*. Понятие, противоположное независимому множеству, есть полный подграф [2]. В противоположность независимому множеству, в котором не могут встретиться две смежные вершины, в полном подграфе все вершины попарно смежны. Очевидно, что независимое множество графа $G$ соответствует полному подграфу графа $\overline{G}$ и наоборот, где $\overline{G}$ – дополнение графа $G$. Следовательно, задача поиска независимых множеств графа $G$ линейно преобразуется в задачу поиска полных подграфов графа $\overline{G}$.

В [1, 3] доказано, что задача определения того, содержит ли неориентированный граф $G = (X, A)$ полный подграф с $k$ вершинами, является *NP*-полной.

Таким образом, на основании определений 1 и 2, получаем вывод, что задача определения того, содержит ли ориентированный граф $G = (X, A)$ независимое множество с $k$ вершинами, является *NP*-полной.

Множество $S$ является максимальным независимым множеством, если оно удовлетворяет условию (1) и еще такому условию:

$$H \cap \Gamma(H) \neq \varnothing; \quad \forall H \supset S. \tag{2}$$

В работе [3] отмечается, что лучшим из известных алгоритмов поиска семейства максимальных независимых множеств является метод перебора Брона–Кэрбоша. В процессе выполнения этого алгоритма число максимальных независимых множеств увеличивается, но большое число независимых множеств формируется, а затем отбрасывается, так как обнаруживается, что они содержатся в других, ранее полученных, множествах и поэтому не являются максимальными.

Основным недостатком метода Брона–Кэрбоша является необходимость сохранения во время работы алгоритма большого количества вспомогательной информации за все выполненные шаги.

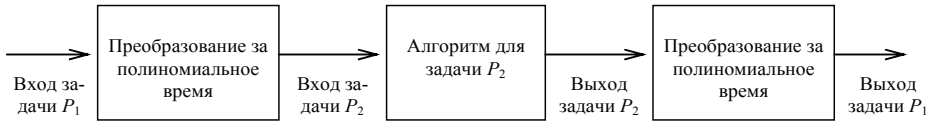


**Рис. 4. Диаграмма преобразования $P_1$ в $P_2$**

Рассмотрим метод перебора, который позволяет обходить указанные выше трудности. В этом методе не нужно запоминать генерируемые транзитивно независимые множества для проверки их на максимальность путем сравнения с ранее сформированными множествами. Формирование ТНМ производится целенаправленно, благодаря чему из рассмотрения исключаются неперспективные ветвления.

**Обоснование метода.** По существу метод является упрощенным перебором, использующим дерево поиска.

В процессе поиска – на некотором этапе $k$ – ТНМ $S_k$ расширяется путем добавления к нему подходящим образом выбранной вершины (чтобы получилось новое ТНМ $S_{k+1}$) на этапе $k + 1$, и так происходит до тех пор, пока добавление вершин станет невозможным, а порождаемое ТНМ не станет максимальным.

Пусть $Q_k$ будет на этапе $k$ наибольшим множеством вершин, для которого

$$S_k \cap Q_k = \varnothing,$$

то есть после добавления любой вершины из $Q_k$ к $S_k$ получается ТНМ $S_{k+1}$.

В некоторый произвольный момент работы алгоритма множество $Q_k$ состоит из вершин двух типов: подмножества $Q_k^-$ тех вершин, которые уже использовались в процессе поиска для расширения $S_k$, и подмножества $Q_k^+$ таких вершин, которые еще не использовались.

Тогда дальнейшее ветвление в дереве поиска включает процедуру выбора вершины $x_{i_k} \in Q_k^+$, добавление ее к $S_k$ для построения множества

$$S_{k+1} = S_k \cup \{ x_{i_k} \},$$

и порождение новых множеств:

$$Q_{k+1}^- = Q_k^- - F(x_{i_k}) \quad \text{и} \quad Q_{k+1}^+ = Q_k^+ - (F(x_{i_k}) \cup \{ x_{i_k} \}).$$

Шаг возвращения алгоритма состоит в удалении вершины $x_{i_k}$ из $S_{k+1}$, чтобы вернуться к $S_k$, изъятии $x_{i_k}$ из старого множества $Q_k^+$ и добавлении $x_{i_k}$ к старому множеству $Q_k^-$ для формирования новых множеств $Q_k^+$ и $Q_k^-$.

Легко заметить, что множество $S_k$ является МТНМ только тогда, когда невозможно его дальнейшее расширение, то есть когда $Q_k^+ = \varnothing$. Если $Q_k^- \neq \varnothing$, то немедленно заключают, что текущее множество $S_k$ было расширено на некотором предшествующем этапе работы алгоритма путем добавления вершины из $Q_k^-$, и поэтому оно не является МТНМ. Таким образом, необходимым и достаточным условием того, что $S_k$ – МТНМ, является выполнение равенств

$$Q_k^- = Q_k^+ = \varnothing.$$

Теперь совершенно очевидно, что если очередной этап работы алгоритма наступает тогда, когда существует некоторая вершина $x \in Q_k^-$ для

которой $F(x) \cap Q_k^+ = \varnothing$, то безразлично, какая из вершин, принадлежащих $Q_k^+$, используется для расширения $S_k$, и это справедливо при любом числе дальнейших ветвлений; вершина $x$ не может быть удалена из $Q_p^-$ на любом следующем шаге $p > k$. Таким образом, условие

$$\exists\, x \in Q_k^- \text{ такая, что } F(x) \cap Q_k^+ = \varnothing, \qquad (3)$$

является достаточным для осуществления шага возвращения, поскольку из $S_k$ при всяком дальнейшем ветвлении уже не получится МТНМ.

Как и во всяком методе, использующем дерево поиска, здесь выгодно стремиться начать шаги возвращения как можно раньше, поскольку это ограничит «размеры» «ненужной» части дерева поиска. Следовательно, целесообразно сосредоточить усилия на том, чтобы возможно раньше добиться выполнения условия (3) с помощью подходящего выбора вершин, используемых при расширении множеств $S_k$. На каждом следующем шаге процедуры можно выбирать для добавления к $S_k$ любую вершину $x_{i_k} \in Q_k^+$; на шаге возвращения $x_{i_k}$ будет удалена из $Q_k^+$ и включена в $Q_k^-$. Если вершину $x_k$ выбрать так, чтобы она принадлежала множеству $F(x)$ при некоторой вершине $x$ из $Q_k^-$, то на соответствующем шаге возвращения величина

$$\Delta(x) = |F(x) \cap Q_k^-|$$

уменьшится на единицу (по сравнению с тем значением, которое было до выполнения прямого шага и шага возвращения), так что условие (3) теперь станет выполняться раньше.

Таким образом, один из возможных способов выбора вершины $x_{i_k}$ для расширения множества $S_k$ состоит, во-первых, в нахождении вершины $x^* \in Q_k^-$ с возможно меньшим значением величины $\Delta(x^*)$ и, кроме того, в выборе вершины $x_{i_k}$ из множества $F(x^*) \cap Q_k^+$. Такой выбор вершины $x_{i_k}$ будет приводить на шаге возвращения к уменьшению величины $\Delta(x^*)$ – каждый раз на единицу – до тех пор, пока вершина $x^*$ не станет удовлетворять условию (3) при выполнении шага возвращения.

Следует отметить, что поскольку на шаге возвращения вершина $x_{i_k}$ попадает в $Q_k^-$, то может оказаться, что при этом новом входе значение величины $\Delta$ меньше, чем для ранее фиксированной вершины $x^*$. Значит, надо проверить, не ускорит ли эта новая вершина выполнение условия (3). Это особенно важно в начале ветвления, когда $Q_k^- = \varnothing$.

Описание алгоритма.

**Начальная установка.**

*Шаг 1.*

Пусть $S_0 = Q_0^- = \varnothing$, $Q = X$, $k = 0$.

**Прямой шаг.**

*Шаг 2.*

Выбрать вершину с индексом $i = \min[i|\ x_i \notin Q_k^-]$ и сформировать $S_{k+1},\ Q_{k+1}^-$:

$$Q_{k+1}^- = Q_k^- \cup Q_i^-,$$
$$S_{k+1} = S_k \cup \{x_{i_k}\}.$$

**Проверка.**

*Шаг 3.*

Если удовлетворяется условие $Q_k^- = Q$, то перейти к шагу 4, иначе – к шагу 2.

*Шаг 4.*

Напечатать максимальное ТНМ $S_k$ и перейти к шагу 5.

**Шаг возвращения.**

*Шаг 5.*

Положить $k = k - 1$. Удалить $x_{i_k}$ из $S_{k+1}$, чтобы получилось $S_k$.

(i) Если $S_k = \varnothing$, то присвоить $Q_k^- = \varnothing$ и перейти к шагу 2.

(ii) В противном случае перейти к шагу 6.

*Шаг 6.*

Обновить множества $Q_j^-$:

$$\forall x_j, x_j \in S_k \qquad Q_j^- = Q_j^- \cup \{x_{i_k}\}.$$

(i) Если $Q_j^- = Q$, то изъять $x_j$ из $Q$:

$$Q = Q - \{x_j\},$$

и проверить следующее условие:

$$Q = \varnothing.$$

Выполнение этого условия свидетельствует об исчерпании множества $X$ при построении семейства максимальных ТНМ. К этому моменту уже будут напечатаны все максимальные ТНМ. Перейти к шагу 9.

(ii) В противном случае (при невыполнении условия) перейти к шагу 7.

*Шаг 7.*

Сформировать новое множество $Q_k^-$:

$$Q_k^- = \bigcup_j Q_j^-, \quad \forall\ x_j, x_j \in S_k.$$

*Шаг 8.*

(i) Если $Q_j^- = Q$, то перейти к шагу 5.

(ii) В противном случае перейти к шагу 2.

*Шаг 9.*

**Остановка.**

Результаты работы разработанного алгоритма перебора удобно представить в виде матрицы $[t_{ij}]$, число строк которой соответствует количеству атрибутов исследуемой предметной области, а число столбцов – количеству найденных МТНМ.

Значения элементов $t_{ij}$ матрицы определяются по следующему правилу.

$$t_{ij} = \begin{cases} 1, \text{если } x_i \in S_j, \\ 0, \text{если } x_i \notin S_j. \end{cases}$$

Результатом применения указанного алгоритма является полное семейство МТНМ. Каждое МТНМ в интерпретации баз данных представляет собой схему отношения, находящуюся в третьей нормальной форме или выше.

*Список литературы*

1. Рейнгольд, Э. Комбинаторные алгоритмы. Теория и практика / Э. Рейнгольд, Ю. Нивергельт, Н. Део ; пер. с англ. Е.П. Липатова ; под ред. В.Б. Алексеева. – М. : Мир, 1980. – 480 с.

2. Кристофидес, Н. Теория графов. Алгоритмический подход : пер. с англ. / Н. Кристофидес. – М. : Мир, 1978. – 432 с.

3. Кофман, А. Введение в прикладную комбинаторику : пер. с англ. / А. Кофман. – М. : Наука, 1975. – 480 с.

---

## Method of Construction of the Family of Maximum Transitive Independent Sets of Attributes

**I.V. Klimenko**

*Rostov State University of Railways, Rostov-on-Don*

**Key words and phrases:** axioms of output functional relationships; Bron–Kerbosh algorithm; functional dependencies; maximum independent transitive sets; normalization of the relational model; normal forms; primary key; problem of minimum coverage.

**Abstract:** The analysis of all possible the ways to solve the problem of finding maximal transitive independent sets is conducted. The optimal way to solve this problem is identified; in the process of solution Bron-Kerbosh method has been modified to correct the deficiencies.

---