

ПРИКЛАДНАЯ ТЕОРИЯ ГРАФОВ

DOI 10.17223/20710410/19/6

УДК 519.1

ИССЛЕДОВАНИЕ КЛИКОВЫХ ПОКРЫТИЙ РЁБЕР ГРАФА

И. А. Бадеха

*Российский государственный социальный университет, г. Москва, Россия***E-mail:** ivanbadekha@gmail.com

Исследуются кликовые покрытия рёбер графа. Описаны графы, имеющие единственное кликовое покрытие рёбер, рассмотрена возможность снижения порядка (сжатия) графа с сохранением структуры покрытия, приведены методы, упрощающие процедуру поиска минимального кликового покрытия рёбер, исследована взаимосвязь между изоморфизмами графов и соответствующих сжатых графов.

Ключевые слова: *графы, покрытия графов кликами, кликовое покрытие рёбер, изоморфизм графов, NP-полные задачи.*

Введение

Кликовые покрытия рёбер графа и порождаемое представление графа в виде объединения клик изучается в теории графов достаточно давно. Одно из первых исследований проведено в работе L. Kou, L. Stockmeyer и C. Wong [1], в которой показана взаимосвязь данной задачи с другими задачами (поиск кликового покрытия вершин, проблема конфликта ключей), доказан ряд утверждений о полиномиальных алгоритмах поиска покрытия, приведён простейший эвристический алгоритм поиска покрытия, достигающий результата за полиномиальное время. Обзор вопросов, связанных с покрытиями и разбиениями графов, приводит J. Orlin в работе [2], которая также вышла в 70-е годы. Современные результаты по данной проблеме получены J. Gramm, J. Guo, F. Huffner, R. Niedermeier и отражены в работе [3], в которой авторы предлагают и исследуют метод редукции графа, позволяющий восстановить кликовое покрытие графа рёбрами из покрытия меньшего графа. Обзор имеющихся современных результатов приводит в тезисах к своей диссертации на соискание степени Master of science M. Cavers [4].

Введём основные используемые понятия. Базовые понятия теории графов можно найти в монографии Ф. Харари [5].

Примем следующие обозначения: $G = (V, E)$, где V — множество вершин, $|V| = n$; E — множество рёбер графа G . Для любого графа T через V_T, E_T обозначим соответственно множества его вершин и рёбер.

Кликой графа G называется любой его максимальный полный подграф, то есть подграф, все вершины которого попарно смежны [5]. При этом *максимальным* (по включению) полным подграфом клика называется в том смысле, что никакой другой подграф, содержащий клику и не совпадающий с ней, не является полным. Полный подграф из трёх вершин будем далее называть треугольником. Легко видеть, что для любого полного подграфа существует клика, его содержащая.

Под *окружением вершины* v понимается множество $N(v)$, состоящее из вершин графа G , смежных с вершиной v . Расширим понятие окружения.

Определение 1. *Окружением ребра e* будем называть множество вершин $N(e)$ графа G , смежных одновременно с обоими концами ребра e .

По аналогии с данным определением, окружением множества вершин $M \subseteq V_G$ будем считать множество вершин $N(M)$, смежных одновременно с каждой вершиной из M .

В некоторых случаях будем рассматривать *невыколотое окружение* вершины, т. е. множество, состоящее из самой вершины и всех смежных с ней вершин. Аналогичным образом определяется невыколотое окружение ребра: это множество, состоящее из концов u и v ребра (u, v) и всех вершин его окружения. Невыколотым окружением множества вершин M будем называть множество $H(M) = M \cup N(M)$. Невыколотое окружение ребра e будем обозначать $H(e)$, невыколотое окружение вершины v — $H(v)$. Граф, порождённый множеством вершин M графа G , будем обозначать $\overline{M}_G$.

Определение 2. *Кликовым покрытием рёбер (КПР)* графа G называется множество клик, такое, что каждое ребро графа G входит, как минимум, в одну из клик данного множества.

Определение 3. КПР графа назовём *тупиковым*, если при исключении из него любой клики оно перестаёт покрывать все рёбра графа.

Каждая изолированная вершина в КПР входит в единственную клику, состоящую из одной этой вершины. Так как случай наличия изолированной вершины тривиален, далее не будем уделять ему особого внимания.

Определение 4. *Минимальным КПР (МКПР)* графа G назовем КПР графа G , содержащее наименьшее из всех возможных КПР число клик.

Мощность МКПР графа G иногда называют его кликовой характеристикой. Отметим, что задача поиска МКПР графа G является NP-полной (см., например, [2]).

Далее в п. 1 и 2 рассматриваются методы упрощения поиска МКПР. В п. 3 исследуется класс графов, обладающих единственным КПР.

1. Упрощение задачи построения МКПР: использование собственных рёбер и полных наборов баз

1.1. Собственные рёбра графа

Рассмотрим особый вид рёбер графа G , наличие которых упрощает процедуру поиска МКПР.

Определение 5. Ребро e назовем *собственным ребром клики K* , если оно принадлежит клике K и не принадлежит никакой другой клике графа G .

Определение 6. Клику K назовем *зафиксированной*, если она содержит хотя бы одно собственное ребро либо содержит только одну вершину.

Укажем простой способ определения собственного ребра.

Утверждение 1. Ребро e является собственным ребром графа G тогда и только тогда, когда $\overline{N(e)}_G$ представляет собой клику в графе G .

Доказательство. Пусть ребро $e = (a, b)$ является собственным и клика K — единственная клика, содержащая ребро e . Пусть существует вершина x , такая, что $x \in N(e)$ и $x \notin K$. Тогда вершины a, b, x образуют треугольник, который не содержится внутри клики K . Это значит, что существует клика K' , содержащая данное ребро и не совпадающая с кликой K , следовательно, ребро e не является собственным, что противоречит предположению.

Пусть теперь $\overline{N(e)}_G = K$, где K — клика. Так как все вершины клики, в которую входит ребро, принадлежат его окружению, клика K — зафиксированная клика, содержащая собственное ребро e . ■

Если граф имеет зафиксированные клики, то процедура поиска МКПР для него упрощается следующим образом: сначала определяются зафиксированные клики, которые входят в любое КПР и соответственно в МКПР, а затем уже производится поиск оставшихся клик МКПР.

1.2. Использование полных наборов баз

Введём ряд понятий, которые будут использованы в дальнейшем для ускорения поиска МКПР. Будем говорить, что *полный подграф H графа G базируется на ребре e* , если любая клика K , содержащая ребро e , содержит H в качестве своего подграфа.

Под пересечением графов далее будем понимать граф, множество вершин которого получено пересечением множеств вершин пересекаемых графов, а множество рёбер — пересечением множеств их рёбер. Очевидно, что пересечением полных подграфов некоторого графа является также полный подграф этого графа.

Определение 7. *Базой b_e ребра e в графе G назовём полный подграф графа G , являющийся пересечением всех полных подграфов, базирующихся на ребре e . При этом ребро e назовём порождающим ребром базы b_e .*

Докажем утверждение, которое позволяет получить характеристику базы ребра.

Утверждение 2. Пусть $e = (a, b)$ — ребро графа G , $N(e)_d$ — множество, состоящее из доминирующих вершин окружения ребра e . Подграф графа G , порождённый множеством вершин $\{a, b\} \cup N(e)_d$, является базой ребра e .

Доказательство. Несложно убедиться, что граф $\{a, b\} \cup N(e)_d$ является полным подграфом графа G . Для этого необходимо отметить, что любая доминирующая вершина из окружения ребра смежна с каждой доминирующей вершиной окружения и обоими концами ребра, а каждый конец ребра смежен с каждой вершиной своего окружения, в том числе с любой доминирующей вершиной окружения. Пусть ребро e принадлежит клике K . Любая вершина клики K смежна с обоими концами ребра e , а значит, принадлежит невыколтому окружению ребра. Предположим, что вершина d , являющаяся доминирующей вершиной в окружении ребра e , не принадлежит клике K . Тогда вершина d смежна с каждой вершиной окружения и с обоими концами ребра. Значит, подграф, порождённый множеством вершин $V_K \cup \{d\}$, является полным. Так как указанный полный граф содержит подграф K , то последний в этом случае не является максимальным по включению полным графом, то есть кликой. Данное противоречие подтверждает верность доказываемого утверждения. ■

Таким образом, поиск базы данного ребра $e = (a, b)$ может производиться по следующему алгоритму: перебираем вершины графа и определяем количество k вершин, смежных с a и b ($O(n)$ элементарных операций). Для каждой вершины u , смежной с обоими концами ребра, снова перебираем все вершины графа и ведём подсчёт количества $k(u)$ вершин, смежных одновременно с a , b и u ; если $k(u) = k$, то вершину u помечаем как доминирующую ($O(n)$ элементарных операций).

Получаем, что трудоёмкость нахождения базы для конкретного ребра составляет $O(n^2)$ элементарных операций.

Определение 8. Множество $B = \{b_1, \dots, b_k\}$, где каждый элемент b_i является базой какого-то ребра, назовём *полным набором баз графа G* , если каждое ребро графа G входит в состав хотя бы одной из этих баз.

Определение 9. *Кликовым покрытием баз* (сокращенно КПБ) графа G для некоторого полного набора баз B назовем множество клик графа G , такое, что каждая база из набора B входит как минимум в одну из клик этого множества.

Докажем утверждение о взаимосвязи между КПР и КПБ.

Утверждение 3. Любое КПБ графа G является его КПР, и любое КПР является КПБ.

Доказательство. Пусть P — КПР графа G . Если ребро e входит в некоторую клику K покрытия P , то и порождённая этим ребром база b_e также входит в эту клику. Так как все рёбра покрыты кликами из КПР P , а любая база порождается каким-то ребром, то в любом полном наборе баз B все базы покрыты кликами из P . Следовательно, P — КПБ графа G .

Теперь предположим, что P — КПБ графа G . Тогда любая база b_e из некоторого полного набора баз B входит в какую-то клику K из P . Поскольку B — полный набор баз, каждое ребро e графа G входит в некоторую базу b_e из B , а следовательно, и в какую-то клику K из P . Следовательно, P — КПР графа G . ■

Таким образом, для получения КПР достаточно получить КПБ.

Отметим, что для того чтобы найти кликовое покрытие баз из некоторого полного набора, достаточно найти множество клик, покрывающих все порождающие рёбра для баз, входящих в этот полный набор. Действительно, если порождающее ребро e для базы b_e покрыто некоторой кликой K , то, согласно определению базы, b_e также покрывается кликой K .

Введём понятие минимального КПБ (по аналогии с минимальным КПР).

Определение 10. *Минимальным КПБ* (сокращенно МКПБ) графа G назовем КПБ графа G , содержащее наименьшее из всех возможных КПБ число клик.

Центральный результат, являющийся следствием из приведённых выше рассуждений и обуславливающий перспективность использования баз в задаче поиска МКПР, сформулирован в следующем утверждении.

Утверждение 4. Пусть P — МКПБ графа G для некоторого полного набора баз B . Тогда P является МКПР графа G .

Доказательство. Пусть P' — МКПР графа G и $|P'| = r$. Из утверждения 3 следует, что любое МКПР графа G является его КПБ. Предположим, существует P'' — МКПБ графа G , такое, что $|P''| < r$. Тогда P'' является КПР графа G и $|P''| < |P'|$, что противоречит тому, что P' является минимальным КПР. Значит, P' является МКПБ графа G . Следовательно, $|P'| = |P|$. Значит, P является МКПР графа G . ■

Таким образом, можно строить покрытие только тех рёбер, которые являются порождающими для баз, входящих в полный набор, не заботясь об остальных рёбрах.

Далее опишем простейший метод построения некоторого полного набора баз в графе G . Изначально считая множество B пустым, для каждого ребра $e \in E_G$ последовательно проведем следующие действия:

1. Найдём множество доминирующих вершин окружения ребра e , обозначим его как b_e .
2. Добавим элемент b_e в множество B .
3. Рёбра, входящие в состав b_e , исключим из дальнейшего рассмотрения.

Когда закончатся рассматриваемые рёбра, будем считать множество B сформированным.

С учётом трудоёмкости $O(n^2)$ нахождения базы ребра и оценки сверху n^2 на количество рёбер, трудоёмкость предложенного алгоритма составляет $O(n^4)$.

Утверждение 5. Множество B , полученное по вышеописанному алгоритму, является полным набором баз графа G .

Доказательство. Так как любой доминирующей вершиной невыколотого окружения ребра e можно расширить любой полный подграф, содержащий ребро e , множество $H(e)_D$ является базой ребра e . Так как ребро исключается из рассмотрения только в том случае, если оно покрывается базой, порождённой другим ребром, то множество B покрывает все рёбра. Таким образом, B является полным набором баз графа G . ■

подавляющее большинство существующих алгоритмов построения МКПР строят покрытие последовательно, добавляя в уже построенное покрытие новые клики. Утверждение 5 позволяет останавливать задачу поиска МКПР в случае, когда в уже найденный набор клик входят все порождающие рёбра некоторого полного набора баз. При таком подходе возникают следующие аспекты.

1) К трудоёмкости поиска МКПР добавляется трудоёмкость построения полного набора баз. Выше предложен алгоритм такого построения с трудоёмкостью $O(n^4)$. Однако полный набор из малого числа баз снижает количество рёбер, которые необходимо покрыть в процессе построения МКПР. Соответственно перед поиском МКПР необходимо найти полный набор из как можно меньшего числа баз; можно рассмотреть другие, более трудоёмкие алгоритмы построения полного набора баз как можно меньшей мощности. Вместе с тем неясно, какой выигрыш при построении МКПР даст выделение порождающих рёбер, которые необходимо покрыть при построении, поэтому такое увеличение трудоёмкости может оказаться нецелесообразным.

2) Используемый алгоритм (эвристический или точный) поиска МКПР можно модернизировать следующим образом: если на каком-то шаге возникает возможность выбрать произвольным образом следующее ребро для добавления к уже покрытым рёбрам, в первую очередь необходимо выбирать порождающие рёбра баз.

Итак, основным достоинством предложенного метода является тот факт, что поиск полного набора баз добавляет к трудоёмкости алгоритма построения МКПР небольшую величину (ограниченную полиномом от числа вершин графа), в то время как при его применении трудоёмкость построения МКПР может существенно уменьшиться за счёт того, что на некотором шаге алгоритма будут покрыты все порождающие рёбра баз (но ещё не все рёбра графа).

2. Редукция графа в задаче построения МКПР: сжатие графа

2.1. Основные понятия

Пусть дано покрытие графа кликами $K_1, \dots, K_s$. Для удобства примем обозначение $I = \{1, \dots, s\}$. Далее буквой U будем обозначать некоторое подмножество I : $U \subseteq I$.

Определение 11. В принятых выше обозначениях для графа $G = (V, E)$ *классом, порождённым множеством U* , назовём множество

$$M_U = \bigcap_{i \in U} V_{K_i} \setminus \bigcup_{j \in I \setminus U} V_{K_j}.$$

В дальнейшем будем рассматривать в качестве U непустые множества, поскольку в случае $U = \emptyset$ множество M_U также пусто. На рис. 1 изображён граф G и все классы

сы M_U для него. В графе G пять непустых классов M_U , три из которых содержат по две вершины, а два — по одной.

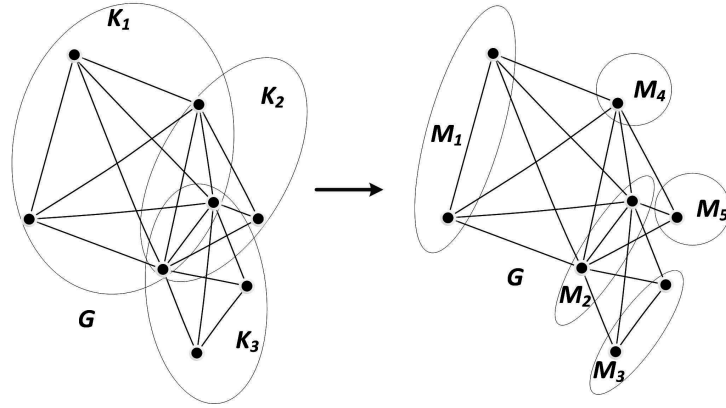


Рис. 1. Покрытие графа G (слева); множества $M_1 = M_{\{1\}}$, $M_2 = M_{\{1,3\}}$, $M_3 = M_{\{3\}}$, $M_4 = M_{\{1,2\}}$, $M_5 = M_{\{2\}}$ (справа)

Отметим очевидный факт: все вершины, принадлежащие одному классу M_U , образуют полный подграф. Это следует из того, что любые две вершины, принадлежащие одному классу, принадлежат одной клике с номером $i \in U$ и, следовательно, соединены ребром в графе G .

Утверждение 6. Две вершины $v, w \in V$ входят в множество M_U для некоторого U тогда и только тогда, когда их невыколотые окружения совпадают.

Доказательство. Необходимость. Пусть вершины v и w находятся в одном множестве M_U . Отметим, что вершины v и w соединены ребром, так как M_U — полный подграф. Пусть существует вершина $u \in V$, соединённая ребром с вершиной v . Значит, вершины u и v входят в общую клику, следовательно, $u \in V_{K_p}$ для некоторого $p \in U$. Так как в силу $p \in U$ и $w \in M_U$ клика K_p содержит и вершину w , то $(w, u) \in E$. Аналогично из смежности произвольной вершины с вершиной w следует её смежность с вершиной v . Таким образом, невыколотые окружения вершин v и w совпадают.

Достаточность. Пусть вершины v и w соединены ребром и $H(v) = H(w)$. Тогда любая клика либо содержит обе эти вершины, либо не содержит ни одной из них, ведь если K — полный подграф, $v \in V_K$ и $w \notin V_K$, то K не является кликой, так как существует полный подграф $K' = K \cup \{w\}$, содержащий K . Это значит, что для любого $p \in I$ выполнено: из $v \in V_{K_p}$ следует $w \in V_{K_p}$ и из $w \in V_{K_p}$ следует $v \in V_{K_p}$, то есть $v \in M_U \iff w \in M_U$. В качестве U возьмём множество номеров тех клик, в которые входят обе вершины v и w . Тогда $v, w \in M_U$. ■

Следствие 1. Отношение принадлежности двух вершин одному классу M_U является отношением эквивалентности.

Таким образом, покрытие графа кликами задает группировку вершин в классы с одинаковыми невыколотыми окружениями. В дальнейшем будем называть классы вершин M_U *классами эквивалентности*. Введём следующее обозначение: для вершины v под U_v будем понимать такое подмножество I , для которого $v \in M_{U_v}$.

Определение 12. Множеством классов вершин графа G для некоторого покрытия назовём множество $\mathcal{M} = \{M \subset V_G : M \neq \emptyset \text{ \& } \exists U \subseteq I (M = M_U)\}$.

Так как эквивалентные между собой вершины, формирующие единый класс, определяются исходя из совпадающих невыколотых окружений, не зависящих от выбора КПР, то верно

Утверждение 7. Для любых покрытий графа кликами множества классов вершин совпадают.

Так как любая вершина извне конкретного класса либо входит в невыколотое окружение каждой вершины класса, либо не смежна ни с одной из них, для классов эквивалентности вершин также выполнено следующее

Утверждение 8. Для любых двух классов эквивалентности M_1 и M_2 либо все вершины класса M_1 смежны со всеми вершинами класса M_2 , либо ни одна из вершин класса M_1 не смежна ни с одной вершиной класса M_2 .

Определение 13. Назовём два класса вершин M_1 и M_2 *соседними*, если все вершины класса M_1 смежны со всеми вершинами класса M_2 .

В случае, когда требуется указать, о каком графе идет речь, будем использовать следующие обозначения. Под множеством U^G для некоторого покрытия $\{K_1, \dots, K_s\}$ графа G будем понимать подмножество множества $I = \{1, \dots, s\}$. Класс эквивалентности в графе G , порождённый множеством U^G , обозначается M_{U^G} .

Для обозначения класса эквивалентности вершин, в который входит вершина u , будем иногда прибегать к такой упрощённой записи: M_u (или M_u^G , если необходимо указать, что рассматривается класс эквивалентности вершин графа G).

2.2. Основные результаты

Пусть дан простой граф $G = (V, E)$ и некоторое его покрытие $\{K_1, \dots, K_s\}$. Рассмотрим взвешенный граф $T(G) = (V_{T(G)}, E_{T(G)})$, полученный следующим образом: каждому непустому классу эквивалентности M графа G соответствует одна вершина $t(M)$ графа $T(G)$, вес которой равен $|M|$, и других вершин в графе $T(G)$ нет. Две вершины $t(M_1)$ и $t(M_2)$ графа $T(G)$ соединены ребром в графе $T(G)$ в том и только в том случае, если классы M_1 и M_2 соседние. Класс всевозможных графов, каждый из которых может быть получен описанным выше образом из какого-то графа, назовём классом $\mathcal{G}$.

На рис. 2 изображён граф G и соответствующий ему граф $T(G)$.

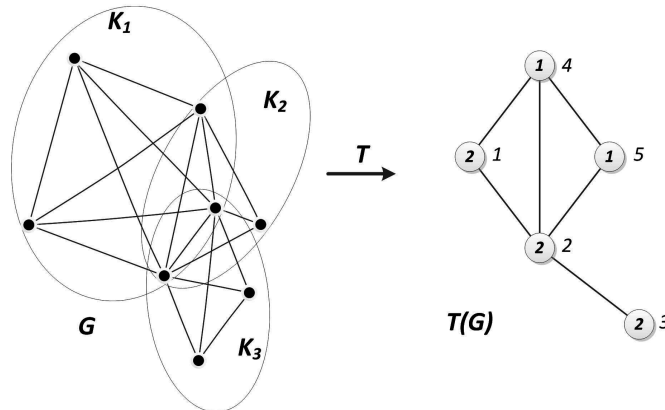


Рис. 2. Покрытие графа G (слева); взвешенный граф $T(G)$ (справа)

Исследуем некоторые свойства, которыми обладает класс $\mathcal{G}$.

Легко видеть, что по данному графу $T(G)$ можно с точностью до изоморфизма восстановить граф G следующим способом:

1. Каждой вершине v графа $T(G)$ поставить в соответствие полный граф K_v размера, равного весу этой вершины.
2. Для каждой пары смежных вершин v, w графа $T(G)$ все вершины графа K_v попарно соединить со всеми вершинами графа K_w .

Согласно построению, для каждого полного подграфа графа G получается полный подграф графа $T(G)$, и наоборот. При этом двум разным полным подграфам графа G соответствуют два разных полных подграфа графа $T(G)$, и также наоборот. Это следует из того, что вершины каждого класса эквивалентности графа G либо все входят в некоторый полный подграф, либо все в него не входят.

Отметим, что в графе $T(G)$ может присутствовать изолированная вершина, даже если в графе G изолированных вершин нет. Это возможно в том и только в том случае, если в графе G есть отдельная компонента связности, являющаяся кликой. Случай наличия изолированной вершины не приводит ни к каким сложностям, поскольку любая такая вершина образует клику, которая входит в любое покрытие графа.

Лемма 1. Если в графе G вершины a и b принадлежат разным классам M_a и M_b , то в графе $T = T(G)$ вершины $t(M_a)$ и $t(M_b)$ принадлежат разным классам $M_{t(M_a)}^T$ и $M_{t(M_b)}^T$.

Доказательство. Из того, что вершины a и b принадлежат разным классам M_a и M_b , в силу утверждения 6 и определения невыколотового окружения вершины следует, что существует вершина $u \in V$, такая, что $(u, a) \in E$ и $(u, b) \notin E$.

С л у ч а й 1. Пусть вершины a и b не соединены ребром в графе G . Тогда $t(M_a)$ и $t(M_b)$ не соединены ребром в графе T , так как вершины класса M_a не смежны с вершинами класса M_b . Значит, невыколотовые окружения вершин $t(M_a)$ и $t(M_b)$ не совпадают и эти вершины принадлежат разным классам эквивалентности $M_{t(M_a)}^T$ и $M_{t(M_b)}^T$.

С л у ч а й 2. Пусть вершины a и b соединены ребром в графе G . Тогда, согласно утверждению 8, все вершины класса M_a смежны с вершинами класса M_b . Значит, вершины $t(M_a)$ и $t(M_b)$ соединены ребром в графе T . По аналогичным соображениям, вершины $t(M_u)$ и $t(M_a)$ соединены ребром в графе T . По соображениям, аналогичным приведённым в п. 1, вершины $t(M_u)$ и $t(M_b)$ не соединены ребром в графе T . Из этого следует, что $M_{t(M_a)}^T$ и $M_{t(M_b)}^T$ различны. ■

Теорема 1 (критерий принадлежности графа классу $\mathcal{G}$). Граф Γ принадлежит классу $\mathcal{G}$ тогда и только тогда, когда мощность любого класса M_u^Γ не превосходит 1.

Доказательство. Необходимость. Предположим, что в графе $\Gamma \in \mathcal{G}$ мощность некоторого класса M_u^Γ больше 1. Рассмотрим граф $G = T^{-1}(\Gamma)$ (как показано ранее, граф G однозначно с точностью до изоморфизма восстанавливается из графа $T(G)$). Из сделанного предположения следует, что в графе $\Gamma = T(G)$ найдутся две различные вершины α и β , принадлежащие одному классу $M_\alpha^\Gamma = M_\beta^\Gamma$. Пусть вершины $a, b \in E_G$ таковы, что $t(M_a) = \alpha$, $t(M_b) = \beta$. Согласно лемме 1, если бы вершины a и b принадлежали разным классам эквивалентности в графе G , то и вершины α и β также принадлежали бы разным классам эквивалентности в графе Γ . Следовательно, вершины a и b принадлежат одному классу $M_a = M_b$. Значит, по построению графа $T(G) = \Gamma$ вершины $\alpha = t(M_a)$ и $\beta = t(M_b)$ совпадают. Приходим к противоречию с тем, что вершины α и β различны. Значит, мощность любого класса M^Γ не превосходит 1.

Достаточность. Пусть в графе $\Gamma \in \mathcal{G}$ мощность каждого класса эквивалентности M_u^Γ равна 1 и в графе Γ содержится ровно s вершин. Построим граф $G = T^{-1}(\Gamma)$ по

тому алгоритму, который предложен в начале п. 2.2. Получим s полных подграфов, порожденных множествами вершин $M_1, \dots, M_s$, где каждое множество M_i соответствует вершине $v_i \in G_\Gamma$. При этом вершины из разных множеств M_i и M_j смежны в том и только в том случае, если смежны вершины v_i и v_j в графе Γ , т. е. если $u \notin M_i$, то либо все вершины множества M_i смежны с u , либо все они не смежны с u . Таким образом, для двух вершин $v, w \in M_i$ выполнено равенство $H(v) = H(w)$.

Рассмотрим теперь две вершины v, w из разных классов: $v \in M_i, w \in M_j, i \neq j$. Предположим, что $H(v) = H(w)$. Все вершины M_i и M_j смежны со всеми вершинами одних и тех же множеств $M_{k_1}, \dots, M_{k_m}$, среди которых есть и M_i , и M_j . Это значит, что в графе Γ соответствующие данным множествам вершины соединены рёбрами с одними и теми же вершинами и между собой, то есть их невыколотые окружения совпадают. Согласно утверждению 6, они принадлежат одному классу эквивалентности $M_{u_i}^\Gamma$, что противоречит условию. Значит, $H(v) \neq H(w)$.

Итак, $H(v) = H(w)$ тогда и только тогда, когда v и w принадлежат одному множеству M_i . Согласно утверждению 6, множества M_i есть классы эквивалентности вершин. По построению графа G граф $T(G)$ есть Γ , откуда получаем, что $\Gamma \in \mathcal{G}$. ■

Таким образом, в любом графе $\Gamma \in \mathcal{G}$ каждая вершина составляет отдельный класс эквивалентности и все такие графы входят в $\mathcal{G}$.

Следствие 2. Пусть граф Γ входит в $\mathcal{G}$. Тогда граф $T(\Gamma)$ совпадает с графом Γ .

Доказательство. В графе Γ каждая вершина α составляет отдельный класс M_α , сумма весов вершин в котором равна весу α . Две вершины α и β графа Γ , соединённые ребром, составляют два соседних класса вершин. Таким образом, при построении графа $T(\Gamma)$ каждая вершина этого графа переводится в отдельную вершину графа Γ , имеющую такой же вес, а каждое ребро — в ребро между соответствующими вершинами графа Γ . ■

Из доказанного выше, в частности, следует, что множество $\mathcal{G}$ является замкнутым относительно операции построения графа на классах вершин. Далее граф, входящий в множество $\mathcal{G}$, будем называть сжатым графом. Докажем теорему о связи между изоморфизмом двух графов и изоморфизмом соответствующих им сжатых графов.

Теорема 2. Графы G_1 и G_2 изоморфны в том и только в том случае, если взвешенные графы $T_1 = T(G_1)$ и $T_2 = T(G_2)$ изоморфны и веса их соответствующих вершин совпадают.

Доказательство. Обозначим вершины $t(M_a^{G_1})$ и $t(M_b^{G_1})$ как α и β соответственно, будем считать, что $M_a \neq M_b$.

Необходимость. Пусть графы G_1 и G_2 изоморфны, а $f : V_{G_1} \rightarrow V_{G_2}$ — соответствующая функция изоморфизма. Это значит, что функция f является биекцией и $(u, v) \in E_{G_1}$ тогда и только тогда, когда $(f(u), f(v)) \in E_{G_2}$. Рассмотрим вершину a графа G_1 и вершину $f(a)$ графа G_2 . Пусть $H(a) = \{a, h_1, h_2, \dots, h_k\}$ — невыколотое окружение вершины a . В силу того, что функция f смежные вершины переводит в смежные, а несмежные — в несмежные, $H(f(a)) = \{b, f(h_1), f(h_2), \dots, f(h_k)\}$. Поэтому для двух вершин a и b графа G_1 выполнено $H(a) = H(b)$ тогда и только тогда, когда для вершин $f(a)$ и $f(b)$ графа G_2 выполнено $H(f(a)) = H(f(b))$.

Обозначим множество вершин графа G_2 , в которые переводится множество M_a функцией f , как $F(M_a)$. Из того, что равенство $H(a) = H(b)$ равносильно $H(f(a)) = H(f(b))$, и из утверждения 6 следует, что множества $F(M_a)$ и $F(M_b)$ являются классами эквивалентности вершин. Поскольку $f(a) \in F(M_a)$ и $f(b) \in F(M_b)$, выполнены

следующие равенства: $F(M_a) = M_{f(a)}$, $F(M_b) = M_{f(b)}$. Значит, каждый класс эквивалентности в графе G_1 функцией f поэлементно переводится в класс эквивалентности в графе G_2 .

Определим функцию $g : V_{T_1} \rightarrow V_{T_2}$ следующим образом: $g(\alpha) = t(M_{f(a)}^{G_2})$. Так как f — функция изоморфизма графов G_1 и G_2 , $|M_a^{G_1}| = |M_{f(a)}^{G_2}|$. Следовательно, вес каждой вершины α в графе T_1 совпадает с весом вершины $g(\alpha)$ в графе T_2 .

Таким образом, функция g переводит каждую вершину графа T_1 в вершину графа T_2 , при этом все вершины графа T_2 покрыты, так как в T_2 каждая вершина соответствует классу эквивалентности графа G_2 , порождённому функцией f из некоторого класса эквивалентности в графе G_1 .

Осталось показать, что вершины α и β соединены ребром в графе T_1 в том и только в том случае, если вершины $g(\alpha)$ и $g(\beta)$ соединены ребром в графе T_2 .

С л у ч а й 1. Пусть вершины α и β соединены ребром в графе T_1 . Это значит, что соответствующие классы вершин M_a и M_b являются соседними в графе G_1 . Так как функция f смежные вершины переводит в смежные, вершины $f(a) \in M_{f(a)}$ и $f(b) \in M_{f(b)}$ соединены ребром в графе G_2 и, следовательно, все вершины класса $M_{f(a)}$ соединены со всеми вершинами класса $M_{f(b)}$, т. е. эти классы соседние. Значит, $g(\alpha) = t(M_{f(a)}^{G_2})$ и $g(\beta) = t(M_{f(b)}^{G_2})$ соединены ребром в графе T_2 .

С л у ч а й 2. Пусть вершины α и β не соединены ребром в графе T_1 , т. е. классы вершин M_a и M_b не являются соседними в графе G_1 . Так как функция f несмежные вершины переводит в несмежные, вершины $f(a) \in M_{f(a)}$ и $f(b) \in M_{f(b)}$ не соединены ребром в графе G_2 и, следовательно, ни одна вершина класса $M_{f(a)}$ не соединена ни с одной из вершин класса $M_{f(b)}$, т. е. эти классы не являются соседними. Значит, $g(\alpha) = t(M_{f(a)}^{G_2})$ и $g(\beta) = t(M_{f(b)}^{G_2})$ не соединены ребром в графе T_2 .

Таким образом, g — функция изоморфизма для графов T_1 и T_2 .

Достаточность. Пусть графы T_1 и T_2 изоморфны, $g : V_{T_1} \rightarrow V_{T_2}$ — функция изоморфизма и веса соответствующих вершин совпадают.

Рассмотрим класс эквивалентности $t^{-1}(g(\alpha))$ графа G_2 . В графе T_2 ему соответствует вершина $g(\alpha)$ с весом, равным мощности рассматриваемого класса. Поскольку вес вершины $g(\alpha)$ равен весу вершины α , мощность класса $t^{-1}(g(\alpha))$ равна мощности класса $M_a^{G_1}$.

Определим функцию $f : V_{G_1} \rightarrow V_{G_2}$ следующим образом. Каждой вершине a графа G_1 ставится в соответствие одна вершина $f(a)$ графа G_2 , принадлежащая классу эквивалентности $t^{-1}(g(\alpha))$. При этом если две вершины a и b графа G_1 принадлежат одному классу эквивалентности $M_a = M_b$, то $f(a) \neq f(b)$. Поставленное требование выполнимо в силу того, что классы эквивалентности M_a и $t^{-1}(g(\alpha))$ равномощны. Других условий на функцию не налагается. Покажем, что f является функцией изоморфизма.

Во-первых, данная функция каждой вершине графа G_1 ставит в соответствие одну вершину графа G_2 . Во-вторых, для каждой вершины графа G_2 найдётся её вершина-прообраз в графе G_1 , так как функция f каждому элементу из множества M_a ставит в соответствие уникальный элемент из $t^{-1}(g(\alpha))$ и такими классами покрываются все вершины графа.

Наконец, покажем, что вершины a и b графа G_1 смежны в том и только в том случае, если смежны вершины $f(a)$ и $f(b)$ графа G_2 .

С л у ч а й 1. Пусть вершины a и b графа G_1 соединены ребром. В этом случае возможны два варианта. Первый вариант: $M_a = M_b$, в этом случае $t(M_a)$ и $t(M_b)$ — одна вершина в графе T_1 , которая функцией g переводится в вершину $g(t(M_a))$ графа T_2 , вес

которой совпадает с весом вершины $t(M_a)$ и с мощностью класса эквивалентности M_a . Согласно определению функции f , вершины a и b переводятся в различные вершины $f(a)$ и $f(b)$, принадлежащие одному классу эквивалентности. Значит, они соединены ребром. Второй вариант: $M_a \neq M_b$, вершины $t(M_a)$ и $t(M_b)$ различны и соединены ребром в графе T_1 . В этом случае вершины $g(t(M_a))$ и $g(t(M_b))$ различны и соединены ребром в графе T_2 ; значит, классы эквивалентности $t^{-1}(g(\alpha))$ и $t^{-1}(g(\beta))$ — соседние в графе G_2 . Отсюда следует, что все элементы $t^{-1}(g(\alpha))$ соединены со всеми элементами $t^{-1}(g(\beta))$, т. е. вершина $f(a)$ соединена с $f(b)$.

С л у ч а й 2. Пусть вершины a и b графа G_1 не соединены ребром. Тогда множества M_a и M_b не являются соседними, и вершины $t(M_a)$ и $t(M_b)$ не соединены ребром в графе T_1 , а значит, и вершины $g(t(M_a))$ и $g(t(M_b))$ не соединены ребром в графе T_2 в силу того, что g — функция изоморфизма. Согласно описанию функции f , отсюда следует, что $f(a)$ и $f(b)$ не соединены ребром в графе G_2 .

Таким образом, f — функция изоморфизма для графов G_1 и G_2 . ■

Доказанная теорема говорит о том, что при изучении свойств кликовых покрытий рёбер можно ограничиться рассмотрением графов, входящих в класс $\mathcal{G}$.

3. Графы с единственным тупиковым КНР

Графы, имеющие единственное тупиковое покрытие, интересны с той точки зрения, что для них процедура поиска минимального покрытия очень проста: необходимо всего лишь породить какое-нибудь тупиковое покрытие, оно и будет единственным и минимальным. Далее приведён алгоритм порождения всех графов, обладающих единственным тупиковым покрытием.

Теорема 3. Клика K присутствует в любом покрытии графа G тогда и только тогда, когда она является зафиксированной.

Доказательство. Пусть клика K присутствует в любом покрытии графа. Рассмотрим некоторое покрытие P графа G . Предположим, что клика K не является зафиксированной, т. е. все рёбра клики K не являются собственными. Но тогда можно в P заменить K на множество клик, каждая из которых, не совпадая с K , покрывает своё ребро клики K . Получили противоречие — построенное покрытие не содержит K .

В обратную сторону утверждение очевидно. ■

Следствие 3. Граф $G = (V, E)$ обладает единственным тупиковым КНР тогда и только тогда, когда все его рёбра покрываются зафиксированными кликами.

Полученный критерий может быть сформулирован в несколько иной форме.

Теорема 4. Покрытие $\{K_1, \dots, K_s\}$ графа G является единственным тупиковым покрытием этого графа тогда и только тогда, когда для любой клики K_i найдутся такие вершины a и b графа G , что $U_a \cap U_b = \{i\}$.

Доказательство. Согласно следствию из теоремы 3, единственность покрытия графа равносильна тому, что покрытие $\{K_1, \dots, K_s\}$ содержит только зафиксированные клики. Рассмотрим клику K_i , $i = 1, \dots, s$. В ней существуют две вершины a и b , не принадлежащие одновременно ни одной клике, кроме K_i , что равносильно условию $U_a \cap U_b = \{i\}$. ■

Полученное утверждение поможет построить алгоритм порождения всех графов, имеющих единственное покрытие кликами.

Утверждение 9. Граф G имеет единственное тупиковое покрытие тогда и только тогда, когда граф $T(G)$ имеет единственное тупиковое покрытие.

Доказательство. Необходимость. Пусть граф G обладает единственным тупиковым покрытием $\{K_1^{(G)}, \dots, K_s^{(G)}\}$. Очевидно, вершины, содержащиеся в одном классе эквивалентности графа G , целиком принадлежат любой клике, которая содержит хотя бы одну из этих вершин.

Рассмотрим множество $\{K_1^{(T(G))}, \dots, K_s^{(T(G))}\}$ полных подграфов графа $T(G)$, где каждый полный подграф $K_i^{(T(G))}$ получается из клики $K_i^{(G)}$ заменой в ней каждого набора эквивалентных вершин одной вершиной, соответствующей в графе $T(G)$ данному классу эквивалентности.

Каждая клика $K_i^{(G)}$ из покрытия графа G является зафиксированной, т. е. содержит собственное ребро. Каждый полный подграф $K_i^{(T(G))}$ является кликой, так как если бы была возможность достраивания его до клики, то клику $K_i^{(G)}$ также можно было бы достроить до большей клики, что противоречит определению клики. Для рассматриваемой клики $K_i^{(G)}$ возможны два варианта.

1. Каждое собственное ребро клики $K_i^{(G)}$ соединяет вершины из одного класса эквивалентности. В этом случае данная клика является отдельной компонентой связности графа G , потому что иначе существует другой класс эквивалентности $M_U^G \subseteq V_{K_i^{(G)}}$, такой, что $i \in U$ и $\{i\} \neq U$, и в этом случае $U \cap \{i\} = \{i\}$, то есть любое ребро (w, u) , где $w \in U$, $u \in \{i\}$, является собственным, что противоречит рассматриваемому условию. Значит, $K_i^{(G)}$ — отдельная компонента связности. В этом случае клика $K_i^{(T(G))}$ — изолированная вершина в графе $T(G)$, таким образом, она входит в любое покрытие.

2. В клике $K_i^{(G)}$ существует собственное ребро (u, v) , соединяющее вершины из разных классов эквивалентности, т. е. $M_u^{(G)} \neq M_v^{(G)}$. Рассмотрим в графе $T(G)$ ребро $(t(M_u), t(M_v))$. Оно является собственным для клики $K_i^{(T(G))}$, так как если существует другая клика, содержащая данное ребро, то соответствующее множество вершин графа G , очевидно, является кликой, отличной от $K_i^{(G)}$, и содержит ребро (u, v) . Таким образом, клика $K_i^{(T(G))}$ входит в любое покрытие графа $T(G)$.

Наконец, осталось заметить, что любое ребро графа $T(G)$ покрыто одной из клик набора $\{K_1^{(T(G))}, \dots, K_s^{(T(G))}\}$ по построению. Значит, данный набор является тупиковым покрытием графа $T(G)$, а из сказанного ранее следует, что данное тупиковое покрытие единственно.

Достаточность. Пусть граф $T(G)$ обладает единственным тупиковым покрытием. Это значит, что все клики данного покрытия являются зафиксированными (либо содержат единственную изолированную вершину).

Пусть $\{K_1^{(T(G))}, \dots, K_s^{(T(G))}\}$ — единственное покрытие графа $T(G)$. Рассмотрим множество клик $\{K_1^{(G)}, \dots, K_s^{(G)}\}$, где каждую клику $K_i^{(G)}$ получим из клики $K_i^{(T(G))}$, заменив в ней каждую вершину на все вершины из соответствующего класса эквивалентности графа G . Как нетрудно убедиться, полученные полные подграфы будут кликами.

Очевидно, данное множество покрывает все рёбра графа G , так как любое ребро либо принадлежит классу эквивалентности, либо соединяет между собой вершины из соседних классов. Все рёбра первого типа покрыты благодаря тому, что все вершины одного класса эквивалентности принадлежат одной клике. Все рёбра второго типа покрыты по построению графа $T^{-1}(G)$.

Рассмотрим клику $K_i^{(T(G))}$. Если она состоит из одной вершины, то клика $K_i^{(G)}$ является отдельной компонентой связности графа G и поэтому входит в любое его покрытие. Если она состоит из более чем одной вершины, то она содержит собствен-

ное ребро $(t(M_u), t(M_v))$ (при этом классы $M_u^{(G)}$ и $M_v^{(G)}$ не пересекаются). Рассмотрим вершины $u \in M_u^{(G)}$ и $v \in M_v^{(G)}$ графа G . Они смежны, так как принадлежат соседним классам. Ребро (u, v) является собственным ребром клики $K_i^{(G)}$, так как если бы существовала другая клика $\overline{K_i^{(G)}} \neq K_i^{(G)}$, содержащая это ребро, то существовала бы соответствующая клика $\overline{K_i^{(T(G))}} \neq K_i^{(T(G))}$ графа $T(G)$, содержащая ребро $(t(M_u), t(M_v))$.

Таким образом, любая клика данного тупикового покрытия входит в любое покрытие графа G и, следовательно, данное покрытие единственно. ■

Породить все графы на n вершинах, имеющие единственное тупиковое КПП, можно с помощью алгоритма, существенно более простого, чем полный перебор графов порядка n . Далее приводим описание такого алгоритма.

Алгоритм состоит из трёх стадий, применяемых для всех возможных количеств клик в покрытии.

На первой стадии для фиксированного числа s клик в покрытии работает рекурсивная процедура $P(s, i)$. На i -м уровне рекурсии в процедуре перебираются функции $f_i : I \setminus \{i\} \rightarrow \{0, 1, 2\}$, для каждой из которых вызывается копия процедуры $P(s, i + 1)$. Глубина рекурсии равна s , вектора значений всех функций f_i хранятся в глобальной памяти и доступны с любого уровня рекурсии. Когда сформирован набор векторов функций f_i для всех $i = 1, \dots, s$, он сохраняется в памяти для дальнейшего использования.

Алгоритм 1. Процедура $P(s, i)$

- 1: Выбрать функцию f_i с нулевым вектором значений на области её определения: $(0, \dots, 0)$.
 - 2: **Пока** вектор значений функции f_i не равен $(2, \dots, 2)$, выполнять:
 - 3: **Если** $i < s$, **то**
 - 4: вызвать процедуру $P(s, i + 1)$,
 - 5: **иначе**
 - 6: сохранить текущий набор векторов функций f_i для $i = 1, \dots, s$ в памяти.
 - 7: Изменить функцию f_i , в качестве вектора значений выбрав следующий в лексикографическом порядке вектор.
 - 8: **Выход** из процедуры.
-

На второй стадии из каждого набора функций, порождённого в процедуре $P(s, 1)$, строится некоторый взвешенный граф, вес каждой вершины в котором равен 1. При этом рассматриваются множества вершин вида L_V , каждое из которых определяется множеством $V \subseteq I$. Каждое такое множество изначально пустое, однако в процессе работы алгоритма в некоторые из них добавляются вершины.

На третьей стадии строятся все возможные варианты весовой функции данного графа, дающие в сумме значений n , и по каждому из них строится граф, обладающий единственным тупиковым КПП.

Вопрос о корректности алгоритма 2 возникает в связи с построением графа $T^{-1}(\Gamma)$. Для графа Γ множества L_V являются классами эквивалентности вершин и каждый из них содержит ровно одну вершину. Значит, согласно теореме 1, взвешенный граф Γ (независимо от весовой функции) принадлежит классу $\mathcal{G}$. Поэтому граф $T^{-1}(\Gamma)$ может быть построен.

Утверждение 10. Алгоритм 2 порождает с точностью до изоморфизма все графы на n вершинах, обладающие единственным тупиковым КПП, и только их.

Алгоритм 2. Построение графов, обладающих единственным тупиковым КНР

-
- 1: Для всех s от 1 до n^2
 - 2: вызвать процедуру $P(s, 1)$;
 - 3: каждый набор векторов значений функций преобразовать во взвешенный граф следующим образом:
 - 4: Для всех функций f_i
 - 5: выбрать из множеств вида L_V два множества L_{V_1} и L_{V_2} , для которых $V_1 = \{k \in I : f_i(k) = 1\}$ и $V_2 = \{k \in I : f_i(k) = 2\}$;
 - 6: если множество L_{V_1} пустое, то добавить к нему вершину;
 - 7: если множество L_{V_2} пустое, то добавить к нему вершину;
 - 8: если общее число добавленных вершин больше чем n , то данный набор векторов не порождает граф — перейти к п. 3 и рассмотреть следующий набор.
 - 9: Построить взвешенный граф Γ следующим образом:
каждому непустому множеству L_V поставить в соответствие вершину v ;
две вершины v_1 и v_2 соединить ребром в том и только в том случае, если соответствующие множества V_1 и V_2 пересекаются;
каждой вершине полученного графа поставить в соответствие вес 1.
 - 10: Преобразовать полученный взвешенный граф размера k в множество графов на $n \geq k$ вершинах следующим образом:
 - 11: Для всех размещений из $n - k$ по k
в соответствии с размещением добавить к весу каждой из k вершин графа Γ соответствующее число;
построить граф $G = T^{-1}(\Gamma)$ и выдать пользователю.
 - 12: **Выход** из алгоритма.
-

Доказательство. Во-первых, покажем, что любой граф, порожденный алгоритмом 2, обладает единственным покрытием. В графе Γ для каждой i -й клики непустыми оказываются некоторые множества $L_{V_1^{(i)}}$ и $L_{V_2^{(i)}}$ (по построению графа являющиеся его классами эквивалентности), такие, что $V_1^{(i)} \cap V_2^{(i)} = \{i\}$. Согласно теореме 4, из этого следует, что граф Γ обладает единственным покрытием. При увеличении веса некоторых вершин графа Γ множество его покрытий остаётся неизменным. Из утверждения 9 следует, что граф G также обладает единственным покрытием.

Во-вторых, покажем, что любой граф, обладающий единственным покрытием, порождается алгоритмом 2 с точностью до изоморфизма. Пусть граф G обладает единственным покрытием. Тогда, согласно утверждению 9, граф $\Gamma = T(G)$ также обладает единственным покрытием. Так как множество покрытий не зависит от весовой функции, граф Γ , у которого каждой вершине приписан вес 1, обладает единственным покрытием. Согласно теореме 4, из этого следует, что для любой клики K_i , входящей в покрытие, найдутся две вершины u и v , соединённые ребром, для которых $U_u \cap U_v = \{i\}$. Таким образом, множество $I \setminus \{i\}$ разбивается на три подмножества: $U_u \setminus \{i\}$, $U_v \setminus \{i\}$ и $I \setminus (\{i\} \cup U_u \cup U_v)$. Любое такое деление однозначно задается функцией $f_i : I \setminus \{i\} \rightarrow \{0, 1, 2\}$. Поскольку в процессе работы алгоритма перебираются все такие функции (и все возможные количества клик в покрытии), можно сделать вывод, что рассматриваются все возможные графы $\Gamma \in \mathcal{G}$. При переходе от наборов векторов к графу Γ отбрасываются только те наборы, которые не порождают графа на n

вершинах. Значит, любой граф, обладающий единственным КПР, строится данным алгоритмом. ■

ЛИТЕРАТУРА

1. *Kou L. T., Stockmeyer L. J., and Wong C. K.* Cliques with regard to keyword conflicts and intersection graphs // *Comm. ACM.* 1978. V. 21. No. 2. P. 135–139.
2. *Orlin J.* Contentment in graph theory: covering graphs with cliques // *Indagat. Math.* 1977. No. 39. P. 406–424.
3. *Gramm J., Guo J., Huffner F., and Niedermeier R.* Data reduction, exact, and heuristic algorithms for clique cover // *Proc. 8th Workshop on Algorithm Engineering and Experiments*, Miami, Fl. January 21, 2006. P. 86–94.
4. *Cavers M. S.* Clique partitions and coverings of graphs. Masters thesis, University of Waterloo, 2005. <http://www.math.uwaterloo.ca/co/graduate-students/files/mmath/Mike-Cavers.pdf>
5. *Харари Ф.* Теория графов М.: Мир, 1973. 301 с.