Н.Н. Петров, Ф.В. Фомин

ПОИСК НА ГРАФАХ,
и
ближайшие родственники

Санкт-Петербург
Май 1999

Оглавление

Предисловие

Предисловие, как это обычно и бывает, еще не написано. Также «почти» не написаны комментарии и ссылки к главам. Быть может имеет смысл написать краткое введение к каждой из глав. Из глобальных изменений — я выкинул доказательство теоремы Ловаса о совершенных графах поскольку она не имеет непосредственного отношения к нашим делам.

То, чего в этой книжке нет, но о чем хотелось бы написать когда-нибудь:

- Поиск на деревьях.
- Алгоритмы на деревьях.
- Pebbling
- Поиск с ограничением на скорости.
- Шахматный поиск.
- Сложность, NP-полнота, алгоритмы.
- Связь разделителей и наименьших триангуляций.
- Дополнение до хордального графа с наименьшим числом ребер.
- Затраты на поиск.
- ε- и l-поиск.
- Решение модельных задач на графах правильных многогранников.
- Астероидальные тройки.
- Теорема Меринга о наименьших триангуляциях.
- Доминирующие пути и доминирующий поиск.
- Миноры; древесная и путевая ширина в теории Робертсона и Сеймура.

Основные понятия теории графов

Граф G,по определению, состоит из конечного непустого множества вершин $V(G)$ и множества ребер $E(G) \subseteq V^{(2)}(G)$ (символом $V^{(2)}$ обозначается множество всех неупорядоченных пар различных вершин из $V(G)$). Каждую пару $e = \{u, v\}$ (в фигурных скобках мы будем писать неупорядоченные множества, а в круглых — упорядоченные) из $E(G)$ называют *ребром* графа а вершины u, v *концами ребра* e. Говорят, что вершины u и v *смежны*, если пара $e = \{u, v\}$ является ребром (вершины u и v *соединяются* ребром e); вершина u и ребро e *инцидентны*, так же как v и e. Ребра называются смежными, если у них есть общая инцидентная вершина. Графы удобно изображать рисунками (потому их и называют графами). На рисунках вершины графа обозначаются кружочками или точками, а соединяющие их рёбра линиями. На рисунке 1 изображены графы пяти правильных многогранников (такие графы часто называют платоновыми).

Иногда нам придется рассматривать более общие объекты, в которых две вершины могут соединяться более чем одним ребром, т.е.в множестве E некоторые элементы $V^{(2)}$ могут повторяться. Такие повторяющиеся элементы называют *кратными* ребрами.Если к тому же допускаются рёбра,инцидентные одной вершине,т.е. петли,то мы будем называть такое обобщение графа *псевдографом.*

Для определения ориентированного графа в определении графа подмножество $V^{(2)}$ нужно заменить на $V \times V$. Итак, *ориентированный граф* D[1] состоит из конечного непустого множества вершин $V(D)$ и множества дуг $A(D) \subseteq V(D) \times V(D)$[2] (напомним, что $\times$ обозначает декартово произведение). Таким образом, дугами графа являются упорядоченные

[1]Обозначать неориентированные графы символом G от первой буквы слова graph уже устоявшаяся традиция. По этой же причине ориентированные графы обычно обозначают D от слова digraph, сокращения directed graph.

[2]Обозначения $E(G)$, $V(G)$ и $A(G)$ от первых букв слов edges,vertices и arcs.

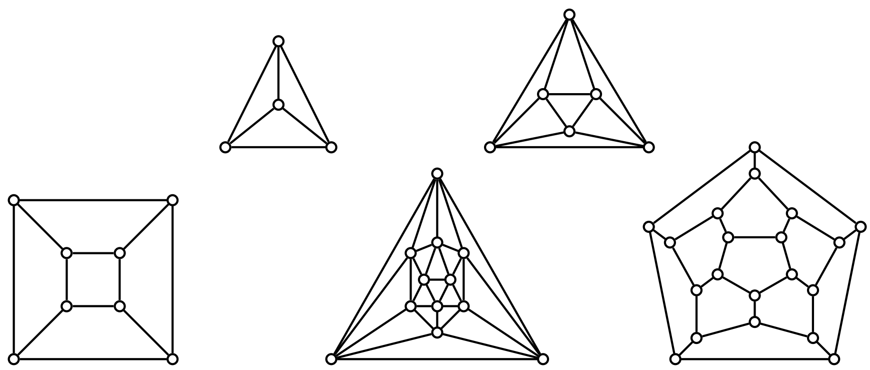


Рис. 1: Графы правильных многогранников: тетраэдра, октаэдра, куба, икосаэдра и додекаэдра

пары его вершин. Если $a = (u, v)$ — дуга, то u — начало, а v — конец a.

Граф G называется *подграфом* графа H, если $V(G) \subseteq V(H)$, $E(G) \subseteq E(H)$. Пусть $U = \{v_1, \ldots, v_k\}$ — произвольное множество вершин графа G.Будем обозначать через $G[U]$ или $G[v_1, \ldots, v_k]$ граф с множеством вершин U, у которого множество рёбер состоит из тех и только тех рёбер графа G, концы которых принадлежат U.Будем говорить, что $G[U]$ является *порождённым* (вершинами из U) подграфом графа G. Граф H называется *надграфом* графа G, если G является подграфом H.

Чередующаяся последовательность

$$v_1, e_1, v_2, e_2, \ldots, e_k, v_{k+1} \qquad (*)$$

вершин и рёбер графа, такая, что $e_i = \{v_i, v_{i+1}\}$, $i \in \{1, \ldots, k\}$, называется *маршрутом, соединяющим вершины v_1 и v_{k+1}* (или (v_1, v_{k+1})-маршрутом). Число k называется длиной (v_1, v_{k+1})-маршрута. Маршрут называется *путём* [3], если все его вершины различны. *Циклом* называется (v_1, v_{k+1})-маршрут длины ≥ 3, в котором вершины $v_1, \ldots v_k$ попарно различны, а $v_{k+1} = v_1$.Аналогичное понятие вводится и для псевдографов, при этом в них могут существовать циклы длины 2 (состоящие

[3] В русскоязычной литературе путь в неориентированном графе часто называют цепью.

из кратных рёбер) и циклы длины 1(петли). Для ориентированных графов путь (иногда его называют *ориентированным путём* или *орпутём*) определяется как последовательность (∗), в которой $e_i = (v_i, v_{i+1})$. Цепью называют последовательность (∗), в которой $e_i = (v_i, v_{i+1})$ или $e_i = (v_{i+1}, v_i)$. Другими словами, цепь получается из пути в неориентированном графе приписыванием каждому из его рёбер произвольной ориентации. Аналогично определяется цикл и для ориентированного графа (*орцикл*).

Граф называется *связным*, если любые две его вершины можно соединить путем.

Несколько специальных классов графов будут встречаться в этой книге. Граф называется *полным*, если все его вершины попарно смежны. Удобно полагать, что пустой граф (граф, множество вершин которого $\emptyset$ — пустое множество) и граф K_1, имеющий одну вершину, являются полными графами. Полный граф с n вершинами обычно обозначается K_n. Связный граф без циклов называется *деревом.* Граф G называется *двудольным*, если множество вершин графа $V(G)$ можно разбить на две доли V_1 и V_2 так, что любые две вершины из одной доли несмежны. Заметим, что дерево является двудольным графом. Двудольный граф G называется *полным двудольным* графом, если всякая вершина одной доли смежна всем вершинам другой доли. Полный двудольный граф с долями размера n и m обозначается через $K_{m,n}$.

Понятие клики будет нами часто использоваться. *k-Кликой* или просто *кликой* в графе G называется множество, состоящее из $k \geq 1$ попарно смежных вершин графа G. Таким образом, всякая вершина графа является 1-кликой. В принятых нами обозначениях k-клика порождает полный граф K_k.

Множество всех вершин графа G, смежных с некоторой вершиной v, называется *окружением вершины v* и обозначается через $N_G(v)$ или просто $N(v)$. *Замкнутым окружением* вершины v называется $N[v] \triangleq N(v) \cup \{v\}$. Степень вершины v в графе G определяется как $deg_G(v) \triangleq |N_G(v)|$. Наибольшую из степеней вершин графа G будем обозначать через $\Delta(G)$.

Для подмножества S вершин графа G через $G \setminus S$ обозначается подграф, порождаемый множеством вершин $V(G) \setminus S$, т.е. $G[V(G) \setminus S]$. Если множество удаляемых вершин S состоит из одной вершины v, то вместо $G \setminus \{v\}$ мы будем часто пишсать $G \setminus v$.

Все остальные термины теории графов будут определяться по мере их появления.

Глава 1

Монотонность поиска

1 Введение

Задачи поиска, рассматриваемые в этой главе, возникают при моделировании следующей чрезвычайной ситуации. В пещере, представляющей собой систему туннелей (граф), потерялся спелеолог. Группа спасателей, располагающая картой этой пещеры, должна разработать план поиска, гарантирующий его спасение. Задача решается просто (достаточно одного спасателя), если незадачливый спелеолог стоит на месте. Но уверенности в этом нет, и группа поиска должна принять все меры, чтобы обеспечить встречу со спелеологом при любом его поведении. Интуитивно ясно, что если на скорость перемещения спелеолога не накладывается никаких ограничений,то необходимое число спасателей будет тем больше, чем сложнее "топология"пещеры. Задача состоит в том, чтобы определить наименьшую численность группы спасателей, способной успешно завершить поиск.

Поскольку речь идёт о гарантированном поиске,то задачу "спасения"можно трактовать как задачу поимки уклоняющегося от встречи убегающего, которому заранее известна программа действий "спасателей". В дальнейшем, следуя традиции, мы будем называть спелеолога убегающим, а спасателей — преследователями.

Целью группы преследователей является построение программы действий, приводящей к поимке (в том или ином смысле) убегающего. На каждом этапе этой программы преследователи определяют множество, в котором может находиться убегающий, избежавший поимки на предыдущих этапах. Это множество называют "загрязнённым", а его дополнение — "очищенным". Используемая терминология имеет простое объяснение: процесс поиска может быть интерпретирован как очистка вершин и рёбер от некой вредной субстанции(например, от газа или пыли). Дело в том, что количество убегающих в задачах подобного рода не имеет значения: выигрывающая стратегия должна приводить к поимке любого из них. Поэтому удобнее следить не за отдельными траекториями убегающего, а за тем множеством, где он может находиться.

С понятием очистки графа связана ещё одна интересная трактовка задач поиска. Предположим, что в нашей сети обнаружен компьютерный вирус. Не имея никакой информации о масштабах заражения,мы принимаем решение "очистить"все узлы. Для очистки используются так называемые программы-вакцины. Во многих случаях важно знать, какое количество этих программ необходимо для "гарантированной"очистки сети. Недостаточное число программ-вакцин или неудачная стратегия очистки может повлечь за собой повторное заражение узлов.

Переходим теперь к точной постановке задачи. Процесс поиска состоит из конечного числа шагов, на каждом из которых разрешается либо "ставить"некоторое число преследователей на свободные вершины графа, либо "удалять"нескольких преследователей с уже занятых вершин (предполагая использовать их на следующих шагах). *Программой поиска* П на графе G называется такая последовательность

$$(Z_0, Z_1, \ldots, Z_n),$$

подмножеств $V(G)$, что $Z_0 = \emptyset$ и для каждого $i \in \{0, \ldots, n-1\}$ справедливо одно из включений: $Z_i \subseteq Z_{i+1}$ или $Z_{i+1} \subseteq Z_i$ (множество Z_i есть множество вершин, занимаемых преследователями на i-ом шаге поиска).

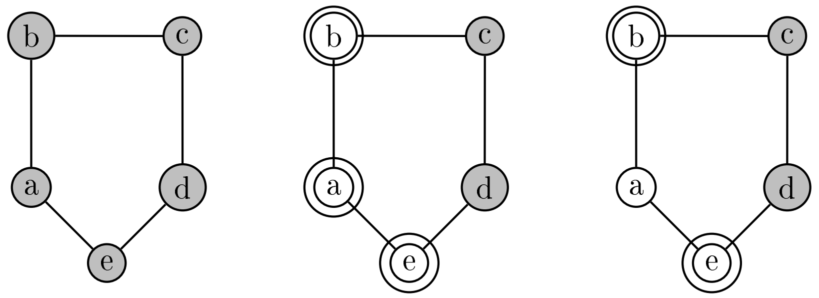
В начальный момент все вершины и рёбра графа предполагаются "загрязнёнными". Ребро e графа G становится "очищенным"на некотором шаге, если его концы оказываются занятыми преследователями. Очищенное ребро e может быть снова загрязнено в момент удаления преследователя, если в этот момент существует не содержащий преследователей путь, соединяющий e с каким либо загрязнённым ребром. Вершина v считается *очищенной* в момент i в двух случаях:

(i) либо она занята преследователем,т.е.$v \in Z_i$,

(ii) либо в момент i все инцидентные ей рёбра являются очищенными.

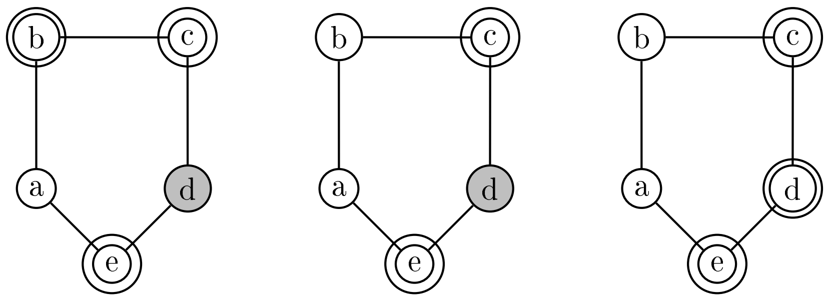
Во всех остальных случаях вершина v считается загрязнённой.

Программа поиска П называется *выигрывающей*, если $A_n = V(G)$, т.е. по окончании поиска все вершины (а, следовательно, и рёбра) оказались очищенными. На рисунке 1.1 приведен пример выигрывающей программы поиска.



$Z_0 = \emptyset$ $\quad$ $Z_1 = \{a, b, e\}$ $\quad$ $Z_2 = \{b, e\}$

$A_0 = \emptyset$ $\quad$ $A_1 = \{a, b, e\}$ $\quad$ $A_2 = \{a, b, e\}$



$Z_3 = \{b, e, c\}$ $\quad$ $Z_4 = \{c, e\}$ $\quad$ $Z_5 = \{c, e, d\}$

$A_3 = \{a, b, c, e\}$ $\quad$ $A_4 = \{a, b, c, e\}$ $\quad$ $A_5 = \{a, b, c, d, e\}$

Рис. 1.1: Пример выигрывающей программы поиска трех преследователей на цикле C_5. Вершины, занимаемые преследователями, обведены окружностями. Загрязненные вершины окрашены серым.

Число преследователей, необходимое для реализации выигрывающей программы П, т.е. максимальное число преследователей, одновременно находящихся на графе, равно

$$\alpha(\Pi) \doteq \max_{i\in\{1\dots n\}} |Z_i|.$$

Минимум величины $\alpha(\Pi)$ по всевозможным выигрывающим программам П на графе G называется *вершинно-поисковым числом* графа G и обозначается через $ns(G)$.[1]

Заметим, что аналогичную задачу поиска можно ставить и для псевдографов. Однако нетрудно видеть, что добавление кратных рёбер и петель приводит к псевдографу с тем же вершинно-поисковым числом.

Таким образом, вершинно-поисковое число графа G есть минимальное число преследователей, необходимое для успешного завершения поиска на графе G в рамках описанной выше формализации. Этой задаче поиска и посвящена настоящая глава.

Задача 1.1. Найдите $ns(K_n)$ для полного графа K_n, $n \geq 1$.

Задача 1.2. Пусть H — подграф графа G. Докажите, что

$$ns(H) \leq ns(G).$$

2 Монотонность

Будем говорить, что программа поиска

$$(Z_0, Z_1, \dots, Z_n)$$

является *монотонной*, если

$$A_0 \subseteq A_1 \subseteq \dots \subseteq A_n,$$

т.е. очищенные вершины повторно не загрязняются.

Мы начнем изучение поставленной задачи с доказательства одного из основополагающих фактов в теории поиска — теоремы о монотонности выигрывающих программ. Этот факт на английском языке звучит как recontamination does not help to search a graph, т.е. повторное загрязнение не помогает поиску. Нами будет доказано, что на всяком графе G с вершинно-поисковым числом k существует монотонная выигрывающая программа k для преследователей.

[1]от английского node search number.

Для подмножества ребер $X \subseteq E(G)$ графа G мы определяем $\delta(X)$ как множество вершин, одновременно инцидентных как ребрам из множества X, так и ребрам из $E(G) - X$. Очевидно, что

$$\delta(X) = \delta(\overline{X}) \doteq \delta(E(G) - X)$$

.Кроме того, имеет место

Лемма 1.3 ***Лемма о "субмодулярности"функции*** $|\delta|$**..** *Для всех* $X, Y \subseteq E(G)$ *справедливо неравенство*

$$|\delta(X \cap Y)| + |\delta(X \cup Y)| \leq |\delta(X)| + |\delta(Y)|. \qquad (*)$$

Доказательство. Замечая, что $\delta(X \cup Y) = \delta(\overline{X} \cap \overline{Y})$,обозначим через $A \doteq \delta(X \cap Y), B \doteq \delta(\overline{X} \cap \overline{Y})$.Если некоторая вершина v принадлежит , например, множеству A, то найдутся такие инцидентные ей рёбра e_1, e_2,что $e_1 \in X \cap Y, e_2 \in \overline{X} \cup \overline{Y}$. Отсюда заключаем, что v принадлежит либо $\delta(X)$,либо $\delta(Y)$. Если же вершина $v \in A \cap B$, то найдутся такие два инцидентные ей ребра e_1, e_2,что $e_1 \in X \cap Y, e_2 \in \overline{X} \cap \overline{Y}$. В этом случае $v \in \delta(X) \cap (Y)$. Таким образом, всякая вершина в левой части неравенства (*) как минимум столько же раз считается и в правой.

Подмножество вершин графа G, являющихся концами ребер множества $X \subseteq E(G)$, будем обозначать через $V(X)$.

Следующее важное понятие клубка вводится для псевдографов. Обозначим через G^0 псевдограф,полученный из графа G добавлением петли к каждой вершине.

k-*Клубком* в G^0, где k — целое неотрицательное число, будем называть такую последовательность $(X_0, X_1, \ldots, X_n)$ подмножеств ребер из $E(G^0)$, что

1. $X_0 = \emptyset$ и $X_n = E(G^0)$;
2. $|V(X_i) - V(X_{i-1})| \leq 1,\ i \in \{1, \ldots, n\}$;
3. если $v \in V(X_i),\ i \in \{1, \ldots, n\}$, то петля с концом в v также принадлежит множеству X_i;
4. $|\delta(X_i)| \leq k$ для всех $i \in \{0, \ldots, n\}$.

Использование клубков оказалось удобным средством для доказательства теорем о монотонноси поиска.

k-Клубок $(X_0, X_1, \ldots, X_n)$ называется *увеличивающимся,* если $X_0 \subseteq X_1 \subseteq \cdots \subseteq X_n$ и $|V(X_i) - V(X_{i-1})| = 1$ для всех $i \in \{1, \ldots, n\}$. Заметим, что если клубок является увеличивающимся, то $n = |V(G)|$.

Следующая теорема устанавливает существование монотонной выигрывающей программы во всех случаях,когда вершиный поиск может быть завершён успешно.

Теорема 1.4 *Для всякого графа G и целого числа $k \geq 0$ следующие утверждения эквивалентны:*

(i) *На графе G существует выигрывающая программа вершинного поиска $(k+1)$ преследователей.*

(ii) *В G^0 существует k-клубок.*

(iii) *В G^0 существует увеличивающийся k-клубок;*

(iv) *На графе G существует монотонная программа вершинного поиска $(k+1)$ преследователей.*

Доказательство. $(i) \Rightarrow (ii)$. Как уже отмечалось, задачи поиска можно рассматривать и на псевдографах, при этом добавление петель и кратных ребер не влияет на поисковое число. Поэтому $ns(G) = ns(G^0)$. Пусть

$$\Pi = (Z_0, Z_1, \ldots, Z_n)$$

— выигрывающая программа поиска $(k+1)$ преследователей на G^0.

Эта программа задает последовательность очищенных вершин

$$(A_0, A_1, \ldots, A_n)$$

в которой $A_0 = \emptyset$ и $A_n = V(G)$. Пусть $S \doteq \{i : |Z_i| = k+1\}$. т.е. множество S состоит из всех номеров шагов с максимальным числом преследователей. Обозначим через

$$(Z'_0, Z'_1, \ldots, Z'_m),$$

где $m = n - |S|$, последовательность, получаемую из Π удалением всех множеств с номерами из S,а через

$$(A'_0, A'_1, \ldots, A'_m),$$

соответствующую последовательность очищенных вершин. Построим теперь искомый клубок.Для всякого $i \in \{0, 1, \ldots, m\}$ определим X_i как множество всех рёбер, порождаемое вершинами из A'_i, так что $V(X_i) = A'_i$. Покажем теперь, что последовательность $(X_0, X_1, \ldots, X_m, X_{m+1})$, где $X_{m+1} = E(G^0)$ является k-клубком в G_0. Свойства 1 и 3 выполняются очевидным образом.Переходя к проверке свойства 2, заметим. что

$V(X_i) - V(X_{i-1}) = A_i - A_{i-1}$ для $i \in \{1, \ldots, m+1\}$, где $A'_{m+1} = V(G)$. Если множества A'_i и A'_{i-1} — соседние члены последовательности A, то требуемая оценка легко следует из способа построения множества A_i.В противном случае при $i \leq m$ существует такой номер j, что $A'_{i-1} = A_j$ и $A'_i = A_{j+2}$.Так как на шаге с номером $(j+1)$ число преследователей, находящихся на графе, равно $(k+1)$ и, следовательно, один из преследователей снимается с вершины,то

$$A_{j+2} - A_j \subseteq A_{j+1} - A_j$$

Таким образом, требуемая оценка справедлива и в этой ситуации.Осталось рассмотреть случай $i = m+1$. Если множество A_n является членом последовательности A' (в этом случае оно совпадает с A'_m),то доказываемое утверждение очевидно.В противном случае $A'_m = A_{n-1}$ и

$$A'_{m+1} - A'_m = A_n - A_{n-1}.$$

так как $A'_{m+1} = A_n = V(G)$. Таким образом, требуемая оценка справедлива для любого $i \in \{1, \ldots, m+1\}$.

Проверим теперь свойство 4. Для этого фиксируем некоторый номер $i \in \{0, 1, \ldots, m+1\}$ и рассмотрим произвольную вершину $v \in \delta(X_i)$ (если $\delta(X_i) = \emptyset$ требуемое неравенство очевидно). Этой вершине инцидентны два ребра, одно из которых $x \in X_i$, а другое $y \in E(G^0) - X_i$.Оба конца ребра x принадлежат A'_i, а один из концов ребра y не принадлежит этому множеству. Следовательно, на i-ом шаге ребро x очищено, а ребро y загрязнено. Поэтому вершина v должна содержать преследователя. Таким образом,$\delta(X_i) \subseteq Z'_i$,и, следовательно,$|\delta(X_i)| \leq k$.

$(ii) \to (iii)$.Для каждого k-клубка $Y = (Y_0, Y_1, \ldots, Y_m)$ псевдографа G^0 введём в рассмотрение величины

$$C(Y) \doteq \sum_{i=0}^{m} |\delta(Y_i)|, D(Y) \doteq \sum_{i=0}^{m}(|Y_i| + 1).$$

Заметим, что добавление 1 в i-ом слагаемом суммы $D(Y)$ существенно, так как число m зависит от k-клубка.Выберем теперь такой k-клубок $X = (X_0, X_1, \ldots, X_m)$,который минимизирует величину $C(Y)$ на множестве всех k-клубков и доставляет минимум сумме $D(Y)$ на множестве тех k-клубков, которые минимизируют величину $C(Y)$.Покажем, что выбранный таким образом k-клубок является увеличивающимся.Докажем сначала,что для каждого $j \in \{1, \ldots, n\}$ справедливо включение $X_{j-1} \subseteq X_j$.

Нетрудно видеть, что $|\delta(X_{j-1} \cup X_j)| \geq |\delta(X_j)|$. Действительно, в противном случае в силу соотношений

$$V(X_{j-1} \cup X_j) - V(X_{j-1} = V(X_j) - V(X_{j-1})$$

и

$$V(X_{j+1} - V(X_{j-1} \cup X_j) \subseteq V(X_{j+1}) - V(X_j)$$

(если $j \leq n-1$) можно было бы заключить, что последовательность

$$(X_0, X_1, \ldots, X_{j-1}, X_{j-1} \cup X_j, X_{j+1}, \ldots, X_n),$$

т.е. последовательность X, в которой X_j заменено на $X_{j-1} \cup X_j$, является k-клубком. Существование такого клубка,однако, противоречит минимальности X.Используя полученную оценку в "субмодулярном "неравенстве,

$$|\delta(X_{j-1} \cup X_j)| + |\delta(X_{j-1} \cap X_j)| \leq |\delta(X_{j-1}| + |\delta(X_j)|$$

получим

$$|\delta(X_{j-1} \cap X_j)| \leq |\delta(X_{j-1}|. \qquad (*)$$

Покажем теперь, что последовательность

$$X^j = (X_0, X_1, \ldots, X_{j-2}, X_{j-1} \cap X_j, X_j, X_{j+1}, \ldots, X_n),$$

т.е. последовательность X, в которой X_{j-1} заменено на $X_{j-1} \cap X_j$, также является k-клубком.Если вершина $v \in V(X_{j-1}) \cap V(X_j)$,то петля с концом в v принадлежит $X_{j-1} \cap X_j$ и, следовательно, $v \in V(X_{j-1} \cap X_j)$. Поэтому

$$V(X_j) - V(X_{j-1} \cap X_J) \subseteq V(X_j) - [V(X_{j-1} \cap V(X_j)] = V(X_j) - V(X_{j+1}).$$

Это соотношение вместе с очевидным включением

$$V(X_{j-1} \cap X_j) - V(X_{j-2}) \subseteq V(X_{j-1}) - V(X_{j-2})$$

(если $j > 1$) гарантирует выполнение свойства 2 для последовательности X^j. Свойство 4 немедленно следует из неравенства (*).Свойства 1 и 3 очевидны. В силу минимальности k-клубка X имеем $|\delta(X_{j-1} \cap X_j)| \geq |\delta(X_{j-1})|$, откуда с учётом (*) получаем $|\delta(X_{j-1} \cap X_j)| = |\delta(X_{j-1})|$. Так как $C(X^j) = C(X)$, то $D(X^j) \geq D(X)$ и, следовательно,$|X_{j-1} \cap X_j| \geq |X_{j-1}|$, что эквивалентно включению $X_{j-1} \subseteq X_j$.Осталось доказать, что для любого $j \in \{1, \ldots, n\}$

$$|V(X_j) - V(X_{j-1})| = 1.$$

Если $|V(X_j)-V(X_{j-1})|=0$, то $V(X_{j+1})-V(X_{j-1})=V(X_{j+1})-V(X_j)$ и последовательность

$$Y^j=(X_0,X_1,\dots,X_{j-1},X_{j+1},\dots,X_n)$$

т.е. последовательность X с удалённым j-ым членом, оказывается k-клубком. В силу минимальности X величина $|\delta(X_j)|$ должна равняться нулю.Но тогда $C(X)=C(Y^j)$ и, следовательно,$D(Y^j)\geq D(X)$,что немедленно приводит к противоречию.На этом доказательство импликации $(ii)\to(iii)$ завершается.

$(iii)\to(iv)$.Пусть $(X_0,X_1,\dots,X_n)$ — увеличивающийся k-клубок в G^0.Обозначим через v_{i+1} единственную вершину из $V(X_{i+1})-V(X_i)$ и введём в рассмотрение множество

$$\Delta_{i+1}\doteq\delta(X_i)\cup\{v_{i+1}\}\subseteq V(X_{i+1}).$$

Нетрудно убедиться, что

$$\delta(X_{i+1})\subseteq\Delta_{i+1}. \qquad (**)$$

В самом деле, если $w\in\delta(X_{i+1})$ и $w\neq v_{i+1}$, то $w\in V(X_i)$ и, стало быть, существует ребро из X_i, инцидентное w.С другой стороны,вершина w инцидентна некоторому ребру из $\bar{X}_{i+1}\subseteq\bar{X}_i$ и, следовательно,$w\in\delta(X_i)$.

Определим программу П, в которой каждому $i\in\{0,1,\dots,n-1\}$ соответствует два шага:

$$Z_{2i}=\delta(X_i)$$

и

$$Z_{2i+1}=\Delta_{i+1}.$$

Таким образом, на шаге с номером 2i+1 преследователь ставится на вершину v_{i+1},в то время как на следующем шаге "лишние"преследователи, занимавшие вершины из $\Delta_{i+1}-\delta(X_{i+1})$, снимаются (см.(**)). Здесь нам удобно считать, что на одном шаге могут быть удалены несколько преследователей.

Покажем теперь, что П является монотонной выигрывающей программой. Пусть

$$A=(A_0,A_1,\dots,A_{2n-1})$$

последовательность подмножеств очищенных вершин,соответствующая программе П. Имеем

$$A_0=V(X_0)=\emptyset, A_1=\{v_1\}=V(X_1), A_2=V(X_1)$$

Пусть для некоторого i множество $A_{2i} = V(X_i)$.Тогда ясно, что $A_{2i+1} = V(X_{i+1})$.Если $i < n-1$,то и $A_{2i+2} = V(X_{i+1})$. В самом деле,на $(2i+2)$-ом шаге множество очищенных вершин не может уменьшиться, так как при снятии преследователя с вершины v повторного загрязнения не происходит ($v \in V(X_{i+1}) - \delta(X_{i+1})$ и, следовательно,все рёбра, инцидентные вершине v на этом шаге очищены). Поскольку $A_{2n-1} = V(X_n) = V(G)$,программа Π является монотонной и выигрывающей.Наибольшее число преследователей, необходимое для реализации программы Π, равно

$$max_{i\in\{0,\ldots,n-1\}}|Z_{2i+1}| = max_{i\in\{0,\ldots,n-1\}}|\delta(X_i)| + 1 \leq k+1.$$

Эта оценка и завершает доказательство импликации $(iii) \to (iv)$. Импликация $(iv) \Rightarrow (i)$ очевидна.

$\square$

Комментарии и ссылки

Первая из задач поиска, рассматриваемых в литературе, является задача рёберного поиска [37].

Гипотеза о монотонности рёберного поиска была высказана Megiddo et al. в [33]. Первое (технически очень сложное) доказательство монотонности рёберного поиска было получено LaPaugh в 1983 году (опубликовано значительно позже [28]). Используя монотонность рёберного поиска Kirousis и Papadimitriou [27] доказали монотонность вершинного поиска. Приводимое доказательство монотонности основано на элегантной теореме Bienstock и Seymour [7] о монотонности смешанного поиска. Определяемые нами клубки являются «родственниками» crusades (крестоносцев). Структуры с таким странным названием использовались Bienstock и Seymour в теореме о монотонности смешанного поиска. Теорема 6.16 заимствована нами из работы [42] (см. также [43]), доказательство монотонности рёберного поиска из [7].

Связь вершинного поиска с задачей нахождения надграфа интервалов с наименьшим кликовым числом была замечена Kirousis и Papadimitriou в [26].

Глава 2

Графы интервалов

1 Графы пересечений

Многие приложения теории графов основываются на следующей конструкции. Пусть A — произвольное множество и $A_1, \ldots, A_n$ — конечная совокупность его непустых подмножеств. Некоторую информацию об их взаимном расположении дает так называемый граф «пересечений», «вершинами» которого являются подмножества $A_1, \ldots, A_n$, а «ребрами» — такие пары (A_i, A_j), $i \neq j$, что $A_i \cap A_j \neq \emptyset$.

Абстрактный граф называется *графом пересечений*, если каждая его вершина может быть «ассоциирована» с одним из подмножеств некоторого множества A, причем вершины графа смежны тогда и только тогда, когда соответствующие подмножества пересекаются. В этом случае говорят, что упомянутые подмножества определяют «реализацию» графа G. Заметим, что «вершины» A_i и A_j при $i \neq j$ могут совпадать как подмножества множества A. Таким образом, полный граф K_n, допускающий очевидную реализацию в произвольном множестве A ($A_1 = \cdots = A_n = A$), для любого натурального n является графом пересечений.

На самом деле нетрудно убедиться, что любой граф G является графом пересечений. Действительно, пусть $A = E(G)$, а реализация $\{A_v, v \in V(G)\}$ определяется следующим образом: A_v есть множество всех рёбер, инцидентных вершине v. Тогда для произвольной пары несовпадающих вершин u и v условие $A_u \cap A_v \neq \emptyset$ эквивалентно смежности вершин u и v.

Далее мы будем часто иметь дело с графами пересечений, для которых множеством A является множество вершин $V(H)$ некоторого графа H, а подмножествами $\{A_i\}$ — подмножества $V(H)$. Такой граф H в совокупности с наборами подмножеств $V(H)$ называют *графическим пред-*

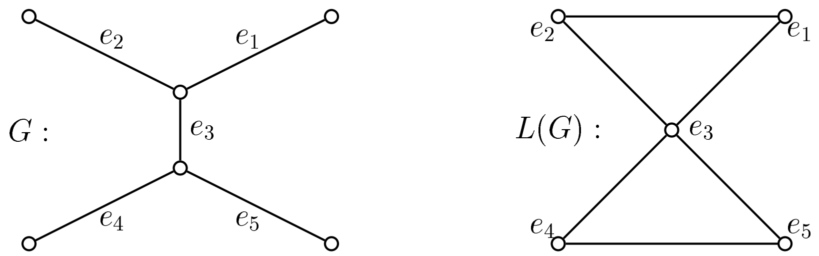


Рис. 2.1: Графы G и $L(G)$.

ставлением графа пересечений. В дальнейшем мы часто будем отождествлять подмножества вершин графического представления H и порождаемый этим множеством подграф графа H.

Задача 2.1. Докажите, что всякий граф имеет графическое представление. Другими словами, для всякого графа G существует граф H такой, что G является графом пересечений некоторого набора подмножеств $V(H)$.

Интересным классом графов пересечений, имеющих графическое представление, является класс «рёберных» графов. *Рёберным графом* графа G называется граф $L(G)$, вершинами которого служат ребра графа G, т.е. пары смежных вершин, причем вершины e_1 и e_2 графа $L(G)$ смежны тогда и только тогда, когда эти рёбра смежны в G. Рёберные графы являются графами пересечений, поскольку несовпадающие ребра $e_1 = \{u_1, v_1\}$, $e_2 = \{u_2, v_2\}$ смежны тогда и только тогда, когда $\{u_1, v_1\} \cap \{u_2, v_2\} \neq \emptyset$. Пример реберного графа изображен на рис. 2.1.

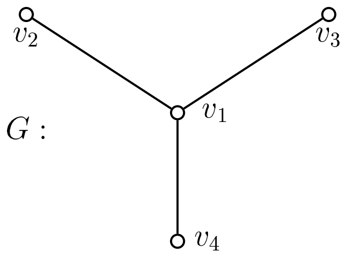
Заметим, что рёберный граф не может быть произвольным.

Задача 2.2. Докажите, что рёберный граф не может иметь в качестве порождённого подграфа полный двудольный граф $K_{1,3}$.

Полный список «запрещенных» порождённых подграфов для рёберного графа можно найти, например, в книге Харари [].

Другим важным классом графов пересечений являются графы интервалов. *Интервалом* мы называем произвольный «открытый» отрезок $]a, b[$, концы которого являются точками числовой прямой $\mathbb{R}^1$. *Графом интервалов* называется граф, множество вершин которого ассоциируется с некоторым набором интервалов, причем две вершины графа смежны тогда и только тогда, когда соответствующие интервалы пересекаются. Такой набор называется *интервальной реализацией графа.* (См. пример на рисунке 2.2).

Задача 2.3. Будет ли дополнение к графу интервалов графом интервалов?



$$\begin{aligned} v_1 &\sim (1,4)\\ v_2 &\sim (1,2)\\ v_3 &\sim (2,3)\\ v_4 &\sim (3,4) \end{aligned}$$

Рис. 2.2: Граф интервалов G и одна из его реализаций $\mathcal{I}$.

Граф G называют *графом пересечений подпутей пути* P, если G является графом пересечений подмножеств $V(P)$, каждое из которых порождает связный подграф графа P, т.е. подпуть.

Задача 2.4. Докажите, что следующие утверждения эквивалентны:

i) Граф G является графом интервалов.

ii) Граф G является графом пересечений сегментов (т.е. замкнутых отрезков) на прямой.

iii) Граф G является графом пересечений подпутей некоторого пути.

Задача 2.5. Докажите, что граф G является графом интервалов тогда и только тогда, когда вершины графа можно пронумеровать различными числами от 1 до $|V(G)|$ таким образом, что для любых трех вершин u, v, w условия:

1. номер вершины v больше номера вершины u и меньше номера w,

2. $(u, w) \in E(G)$

влекут $(u, v) \in E(G)$.

Дополнением к графу G называется граф $\overline{G}$, $V(\overline{G}) = V(G)$, любые две вершины которого u, v смежны тогда и только тогда, когда u и v не смежны в G. Выясним теперь, что представляет собой дополнение к графу интервалов.

Ориентированный граф называется *транзитивным*, если для любых вершин a, b, c этого графа верно следующее: если (a, b) и (b, c) — дуги, то и (a, c) — дуга. Напомним, что дугой (a, b) ориентированного графа называется его ребро с началом в вершине a и концом в вершине b. Граф, из которого ориентацией ребер можно получить транзитивный граф, называют графом *сравнений*.

Задача 2.6. Привести пример графа, не являющегося графом сравнений.

Справедливо следующее утверждение.

Лемма 2.7. *Дополнение к графу интервалов является графом сравнений.*

Доказательство. Пусть G — граф интервалов и $\{I_v | v \in V(G)\}$ — его интервальная реализация. Будем писать $I_u < I_v$, если и только если для любых точек $x \in I_u$ и $y \in I_v$ выполняется неравенство $x < y$. Таким образом, если $I_u \cap I_v = \emptyset$, то либо $I_u < I_v$, либо $I_v < I_u$. Это отношение порождает на графе $\overline{G}$, дополнении графа G, естественную ориентацию: для любого ребра $(u, v) \in E(\overline{G})$ полагаем

$$(u, v) \text{ — дуга } \iff I_u < I_v.$$

Легко проверяется, что ориентированный таким образом граф $\overline{G}$ является транзитивным. $\square$

Напомним, что *кликовым числом* $\omega(G)$ графа G называется число вершин наибольшего полного подграфа (наибольшей клики) в G, иными словами, наибольшее количество попарно смежных вершин. Наименьшее число цветов, в которые можно раскрасить вершины графа G так, что любые две смежные вершины окрашены в разные цвета, (см. также более формальное определение на стр. 35) называется хроматическим числом графа G и обозначается через $\chi(G)$.

Задача 2.8. Докажите, что хроматическое число графа интервалов G совпадает с его кликовым числом, т.е.

$$\chi(G) = \omega(G).$$

Графы интервалов естественным образом возникают во многих прикладных задачах.

Пример 2.9 Задача составления расписания. Необходимо распределить аудитории для чтения n лекций $C_1, \ldots, C_n$ так, чтобы никакие две лекции не читались одновременно в одной аудитории. Пусть $\Delta_i \triangleq [t_i, t^i]$ — промежуток времени, в течении которого должна быть прочитана лекция C_i. Рассмотрим граф G, вершинами которого являются лекции $C_1, \ldots, C_n$, а рёбрами — такие пары C_i, C_j, $i \neq j$, что $\Delta_i \cap \Delta_i \neq \emptyset$. Ясно, что граф G является графом интервалов. Каждая аудитория может быть представлена для чтения лишь тех лекций, среди которых нет ни одной пары «смежных». Нетрудно видеть, что если в наличии имеется достаточно много аудиторий, то их всегда можно распределить так, как требуется. При этом искомое распределение естественным образом порождает «правильную» раскраску вершин графа G, т.е. раскраску, при которой любые две смежные вершины раскрашены в разные цвета (каждая используемая аудитория определяет «цвет» тех вершин-лекций, для которых эта аудитория отведена), а наименьшее число аудиторий, необходимое для такого распределения, равно хроматическому числу графа G. Хроматическое же число графа интервалов совпадает с его кликовым числом. $\square$

Пример 2.10 Задача о хранении продуктов. Имеется n продуктов $c_1, \ldots, c_n$, причём продукт c_i должен храниться при температуре от t_i до t^i. Требуется найти наименьшее число холодильников для хранения этих продуктов. Предполагается, что все холодильники достаточно вместительные и в каждом из них может поддерживаться любая температура в пределах от $\min_i t_i$ до $\max_i t^i$. Набор отрезков $[t_i, t^i]$, $i \in \{1, \ldots, n\}$, является интервальной реализацией некоторого графа интервалов G, вершинами которого являются продукты $c_1, \ldots, c_n$. Так как все продукты из произвольной клики могут храниться в одном холодильнике, то поставленная задача сводится к нахождению наименьшего числа клик, объединение которых совпадает с множеством всех вершин графа G. Эта известная задача называется задачей о *наименьшем покрытии кликами.* $\square$

2 Хордальные графы

Важным обобщением графов интервалов являются *хордальные* (или триангулированные) графы. Классическое определение таково: граф называется хордальным, если ни один из его порожденных подграфов не является простым циклом длины ≥ 4. Другими словами, для всякого простого цикла длины ≥ 4 есть ребро (называемое *хордой*), соединяющее две «несоседние» вершины этого цикла. Из определения хордального графа немедленно следует

Лемма 2.11. *Всякий граф интервалов является хордальным.*

Задача 2.12. Привести пример хордального графа, не являющегося графом интервалов.

Далее нам понадобится следующее определение. Клика C графа называется *максимальной (по включению) кликой*, если она не содержится в клике с бо́льшим числом вершин.

Структуру графа интервалов в значительной мере выясняет

Теорема 2.13 *(Gilmore-Hoffman [1964]) Следующие утверждения эквивалентны:*

1. *G — граф интервалов.*
2. *G — хордальный граф и его дополнение $\overline{G}$ является графом сравнений.*
3. *Максимальные клики графа G можно перенумеровать так, чтобы выполнялось следующее утверждение: для всякой вершины $v \in V(G)$ и любых двух клик с номерами $i < j$, содержащих v, всякая максимальная клика с номером k, $i < k < j$, тоже содержит v.*

Доказательство. 3)$\Longrightarrow$ 1). Пусть L — нумерация клик, удовлетворяющая условию 3). Сопоставим каждой вершине v графа G сегмент $I(v)$, левым концом $I(v)$ является наименьший из номеров максимальных клик содержащих v, а правым — наибольший. По свойству нумерации L всякое целое число из отрезка $I(v)$ является номером клики, содержащей v, и, следовательно, несовпадающие вершины u, v смежны тогда и только тогда, когда они принадлежат одной и той же максимальной клике. Отсюда немедленно следует, что смежность вершин u, v эквивалентна непустоте пересечения сегментов $I(u)$ и $I(v)$.

1)$\Longrightarrow$2). Эта импликация немедленно следует из леммы 2.7 и леммы 2.11.

Для завершения доказательства теоремы осталось доказать истинность импликации 2)$\Longrightarrow$3).

Пусть F — транзитивная ориентация графа $\overline{G}$. Имеет место

Лемма 2.14. *Для любых несовпадающих максимальных клик C_1, C_2 графа G существует дуга из F, начало которой принадлежит C_1, а конец — C_2 (или наоборот). Если таких дуг несколько, то все они имеют «одинаковое направление» (из C_1 в C_2 или наоборот).*

Доказательство леммы. Существование по крайней мере одной дуги, соединяющей C_1 и C_2, очевидно, так как в противном случае множество $C_1 \cup C_2$ оказывается кликой, что противоречит максимальности клик C_1 и C_2.

Докажем, что все такие дуги имеют «одинаковое направление». Предположим противное, тогда найдутся такие дуги (a, b), (d, c) из F, что $a, c \in C_1$, $b, d \in C_2$, при этом ясно, что $a \neq d$ и $b \neq c$. Если $a = c$, то из транзитивности F следует, что (d, b) является дугой в F, но это противоречит тому, что C_2 является кликой. Поэтому $a \neq c$. По аналогичной причине $b \neq d$. Т.о. все вершины a, b, c, d различны.

Если $(a, d) \in E(G)$ и $(b, c) \in E(G)$, то граф G содержит цикл длины 4, порожденный вершинами a, b, c, d и, следовательно, не является хордальным. Предположим для определённости, что вершины a, d не являются смежными. Тогда, если (a, d) является дугой в F, то по транзитивности F и (a, c) должна быть дугой, что невозможно, так как C_1 — клика. По аналогичной причине и (d, a) не может быть дугой в F. Таким образом, во всех случаях мы приходим к противоречию. $\triangle$

Продолжим доказательство теоремы. Для максимальных клик графа G определим отношение $<$: $C_1 < C_2$ тогда и только тогда, когда в графе F есть дуга из C_1 в C_2. Корректность такого определения гарантируется леммой. Докажем, что введенное нами отношение является транзитивным, иначе говоря, соотношения $C_1 < C_2$ и $C_2 < C_3$ влекут за собой $C_1 < C_3$.

Пусть (w, x), (y, z) суть дуги F, причём $w \in C_1$, $y, x \in C_2$ и $z \in C_3$. Заметим, что $w \neq z$, ибо в противном случае (в силу транзитивности) дуга $(y, x) \in F$, что противоречит смежности вершин x и y в клике C_2. Доказываемое утверждение немедленно вытекает из транзитивной ориентации F, если $x = y$ или одна из пар $(x, z), (w, y)$ является ребром $\overline{G}$. Осталось рассмотреть следующий случай: все вершины w, x, y, z различны и $(x, z), (w, y) \in E(G)$. Если $(w, z) \in E(G)$, то вершины w, x, y, z порождают цикл длины 4, что противоречит хордальности графа G. Поэтому дуга $(w, z) \in F$ и $C_1 < C_3$.

Введённое отношение позволяет перенумеровать все максимальные клики графа G «в порядке возрастания». Покажем, что при такой нумерации выполнено условие, сформулированное в утверждении 3). Предположим противное: тогда найдутся такие максимальные клики $C_i < C_k < C_j$ и вершина x, что $x \in C_i \cap C_j$, но $x \notin C_k$. Так как C_k является максимальной кликой, то найдется такая вершина $y \in C_k$, что $(x, y) \notin E(G)$. Поскольку $C_i < C_k$, то $(x, y) \in F$. С другой стороны, поскольку $C_k < C_j$ и $x \in C_j$, то и $(y, x) \in F$. Противоречие. $\square$

Всякому графу G можно сопоставить *матрицу клик*, в которой вершинам соответствуют столбцы, а максимальным кликам — строки. Пусть $(C_1, \ldots, C_q)$ — некоторое упорядочение максимальных клик, а $(v_1, \ldots, v_n)$ — некоторое упорядочение вершин графа G. Определим матрицу клик $C(G) = \{c_{ij}\}_{i \in \{1,\ldots,q\}, j \in \{1,\ldots,n\}}$ графа G, где

$$c_{ij} = \begin{cases} 1, & \text{если } v_j \in C_i, \\ 0, & \text{в противном случае.} \end{cases}$$

Очевидно, что матрица клик определяется с точностью до перестановки строк и столбцов.

Из доказанной теоремы получаем

Следствие 2.15. *Следующие утверждения эквивалентны:*

1. *G — граф интервалов.*

2. *Существует упорядочение максимальных клик графа G такое, что в матрице клик, соответствующей данному упорядочению, в каждом столбце единицы расположены «последовательно», т.е., если $c_{ji} = 1$ и $c_{ki} = 1$, $j < k$, то для всех $l \in \{j, \dots, k\}$ $c_{li} = 1$.*

3 Связь с поиском

Пусть $\omega(G)$, как и раньше, — кликовое число графа G, т.е. наибольшее число попарно смежных вершин в графе G. Связь вершинно-поискового числа с размером наибольшей клики в графе интервалов устанавливает следующая лемма.

Лемма 2.16. *Для всякого графа интервалов I верно равенство $\omega(I) = ns(I)$.*

Доказательство. Известно (теорема 2.13), что для всякого графа интервалов I существует упорядочение

$$(C_1, C_2, \dots, C_m)$$

его максимальных клик графа I такое, что для всякой вершины $v \in V(G)$ и любых двух клик с номерами $i < j$, содержащих v, всякая максимальная клика с номером k, $i \le k \le j$, тоже содержит v. Это свойство можно переформулировать следующим образом: для всех индексов $i \le k \le j$ выполняется условие

$$C_i \cap C_j \subseteq C_k. \qquad (*)$$

Для каждого $k \in \{1, \dots, m-1\}$ обозначим через I_k объединение первых k клик, т.е.

$$I_k \doteq \cup_{i=1}^{k} C_i$$

. Покажем, что любой путь, соединяющий произвольную вершину $v \in I_k$ с некоторой вершиной из $V(G) - I_k$, содержит вершину из $C_k \cap C_{k+1}$. В самом деле, каждый такой путь содержит ребро $\{x, y\}$, один конец которого (скажем, x) принадлежит I_k, а другой — множеству $V(G) - I_k$. Концы этого ребра принадлежат некоторой максимальной клики C_j, номер которой , очевидно, больше k. Так как $x \in I_k \cap C_j$, то в силу (*) вершина $x \in C_k \cap C_{k+1}$.

Опишем теперь следующую программу поиска. Сначала преследователи занимают вершины клики C_1. На следующем шаге снимаются все преследователи, занимающие вершины из $C_1 - C_2$ (если они есть). Оставшиеся преследователи, занимающие вершины из $C_1 \cap C_2$, в силу доказанного выше свойства множества I_1, препятствуют повторному загрязнению. Далее преследователи занимают все вершины клики C_2, освобождают вершины из $C_2 - C_1$ и т.д. В силу доказанного выше ни на одном этапе повторного загрязнения не происходит и по окончании действия программы очищенным окажется весь граф I. Число преследователей, задействованных в описанной программе, равно размеру наибольшей (по числу вершин) клики графа I, т.е. кликовому числу $\omega(I)$. Поэтому $ns(I) \le \omega(I)$.

Докажем теперь противоположное неравенство. Пусть I_0 — подграф графа I, порождённый вершинами той клики, мощность которой равна $\omega(I)$. Тогда (см. задачи 1.1 и 1.2)

$$\omega(I) = ns(I_0) \le ns(I). \qquad \square$$

Как нетрудно убедиться (см. задачу 2.8), кликовое число графа интервалов совпадает с его хроматическим числом. Таким образом, для всякого графа интервалов I

$$ns(I) = \chi(I).$$

Лемму 2.16 можно обобщить следующим образом. Определим $iw(G)$ — *интервальную ширину* (interval-width) графа G как

$$iw(G) \doteq min w(I),$$

где минимум берется по всем графам интервалов I, содержащих граф G в качестве подграфа, или, как иногда говорят, являющихся надграфами графа G.

Теорема 2.17 (Kirousis-Papadimitriou, 85) *Для всякого графа G верно равенство $iw(G) = ns(G)$.*

Доказательство. Пусть

$$(Z_0, Z_1, \dots, Z_n)$$

— монотонная программа вершинного поиска для $ns(G)$ преследователей на графе G. Для вершин $u \in V(G)$ определим номера

$$l_u = \min\{i \colon u \in Z_i\}$$

$$a \sim [1,1]$$
$$b \sim [1,4]$$
$$c \sim [3,5]$$
$$d \sim [5,5]$$
$$c \sim [1,5]$$

$I:$

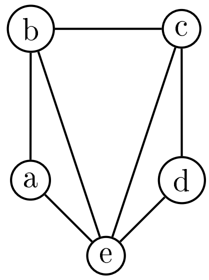


Рис. 2.3: Граф интервалов I и его представление, построенное по выигрывающей программе поиска

$$r_u = \max\{i \colon u \in Z_i\}.$$

Поскольку программа является монотонной, нетрудно убедиться, что для всякого $j \in \{l_u, \ldots, r_u\}$ выполняется условие $u \in Z_j$.

Построим граф интервалов I, сопоставив вершине $u \in V(G)$ сегмент $[l_v, r_v]$ числовой оси. В результате поиска все ребра графа G очищаются, поэтому для всякого ребра с концами u, v существует индекс $i \in \{1, \ldots, n\}$ такой, что $u, v \in Z_i$. Поэтому если $\{v, u\} \in E(G)$, то $[l_v, r_v] \cap [l_u, r_u] \neq \emptyset$, и, следовательно, граф G является подграфом графа интервалов I. Поскольку кликовое число графа интервалов I равно максимальному числу интервалов, имеющих общую точку, что, в свою очередь, равно максимальному числу преследователей, одновременно находящихся на графе G,то

$$\omega(I) = \max_{i \in \{1,\ldots,n\}} |Z_i| = ns(G).$$

Таким образом,

$$ns(G) = \omega(I) \geq iw(G).$$

На рисунке 2.3 изображен граф интервалов и его представление, соответствующее программе поиска с рисунка 1.1 на стр. 3.

Для доказательства обратного неравенства $ns(G) \leq iw(G)$ достаточно заметить, что поисковое число подграфа не превосходит поискового число самого графа (задача 7.30) и воспользоваться леммой 2.16. Для графа интервалов I, содержащего G в качестве подграфа, и такого, что $\omega(I) = iw(G)$ имеет место следующее неравенство

$$ns(G) \leq ns(I) = \omega(I) = iw(G).$$

□

Отметим, что доказательство теоремы является «конструктивным». По известной поисковой программе легко строится надграф интервалов с наименьшим кликовым числом.

4 Матрицы сетей-ключей

Необходимость определения надграфа интервалов с наименьшим кликовым числом возникает не только в задачах поиска. Проблема нахождения укладки матрицы ключей (Gate Matrix Layout) возникла при решении задач о нахождении *линейных укладок* СБИС (сверхбольших интегральных схем). Термин линейные укладки подчеркивает тот факт, что наиболее важной особенностью данной СБИС-архитектуры является линейное (т.е. размерности один) расположение соответствующих физических объектов, называемых ключами.

Задача формулируется для $n \times m$ матрицы $M = m_{ij}$, состоящей из 0 и 1 и называемой *матрицей сетей-ключей* (net-gate matrix). Столбцы этой матрицы представляют ключи $\{G_1, \ldots, G_m\}$, а строки — сети (соединители) $\{N_1, \ldots, N_n\}$. Ключи соответствуют электронным устройствам, расположенных на одной дорожке, а сети — связям между этими устройствами. Сеть N_i соединяет все ключи G_j, для которых выполняется равенство $m_{ij} = 1$. Соединение производится по дорожкам. Для заданного упорядочения ключей (столбцов) каждому соединению отводится часть дорожки от самого левого до самого правого ключа, между которыми должно быть установлено соединение.

Мы ограничимся лишь математической постановкой задачи, отсылая заинтересованного практическими применениями читателя к обзору Мёринга [34].

Для упорядочения $\delta = (G_1, \ldots, G_m)$ ключей (столбцов) матрицы M определим расширенную матрицу $M_\delta = \overline{m}_{ij}$ следующим образом:

$$\overline{m}_{ij} = \begin{cases} 1, & \text{если найдутся } k \le j \le l \text{ для которых } m_{ik} = m_{il} = 1, \\ 0, & \text{в противном случае} \end{cases}$$

Сети в расширенной матрице (на практике это металлические линии) могут быть уложены на одну дорожку, если у них нет общих ключей (т.е. если металлические линии не пересекаются). Более формально, будем говорить, что сети N_i, N_j расширенной матрицы пересекаются, если у этих сетей имеется общий ключ, т.е. найдется индекс k такой, что $\overline{m}_{ik} = \overline{m}_{jk} = 1$. Говорят, что сети $\{N_1, \ldots, N_n\}$ расширенной матрицы M_δ

можно уложить на t дорожек, если существует разбиение $T_1, T_2, \ldots, T_t$ множества индексов $\{1, 2, \ldots, n\}$ такое, что для всякого $j \in \{1, 2, \ldots, t\}$ любые две строки из T_j не пересекаются. Наименьшее число дорожек, на которые можно уложить сети матрицы M_δ, обозначим через $t(M, \delta)$. Определим число

$$t(M) \stackrel{\triangle}{=} \min\{t(M, \delta)\colon \delta \text{ — упорядочение ключей в } M\}.$$

Укладка сетей в наименьшее число дорожек важна при разработке линейных укладок СБИС, поскольку площадь электронной платы, занимаемой m ключами, пропорциональна mt, где t — число дорожек, на которые укладываются сети соответствующей расширенной матрицы сетей-ключей.

Первое, что должно привлечь наше внимание, это очевидная связь расширенных матриц сетей-ключей с графами интервалов. В самом деле, каждая строчка матрицы задает интервал и соответствующий граф пересечений строчек является графом интервалов.

По матрице сетей-ключей M зададим граф G с вершинами $V(G) = \{v_1, v_2, \ldots, v_n\}$ следующим образом. Вершины v_i, v_j, $i \neq j$, графа G смежны тогда и только тогда, когда соответствующие сети N_i и N_j имеют общий ключ (найдется k такое, что $m_{ik} = m_{jk} = 1$). Такой граф часто называют *графом несовместимости,* из-за того, что его смежные (в этом графе) сети не могут быть уложены на одну дорожку. Для упорядочения ключей δ граф интервалов I_δ, определяемый как граф пересечений сетей $\{N_1, \ldots, N_n\}$, очевидным образом содержит граф несовместимости G в качестве подграфа. Поскольку укладка сетей матрицы M_δ на дорожке соответствует разбиению вершин графа I_δ на множества попарно не смежных вершин, то для хроматического числа графа I_δ получаем следующее неравенство:

$$\chi(I_\delta) \leq t(M, \delta).$$

Таким образом,

$$t(M) \geq \min\{\chi(I)\colon I \text{ — граф интервалов и } E(G) \subseteq E(I)\}.$$

Нетрудно убедиться в том, что верно и обратное неравенство. Пусть I — граф интервалов, содержащий граф несравнимости G в качестве подграфа. Интервальная реализация $\{I_v | v \in V(G)\}$ графа I задает естественным образом частичное упорядочение вершин графа G (и сетей матрицы M) $P = (V(G), <)$: $u < v$ тогда и только тогда когда $I_u < I_v$ (т.е. для любых точек $x \in I_u$ и $y \in I_v$ выполняется неравенство $x < y$). Частичное упорядочение P определяет (неоднозначно) следующее упорядочение $\delta = (G_1, G_2, \ldots, G_m)$ множества ключей: $i < j$, если найдутся такие сети $N_r < N_s$, что $G_i \in N_r, G_j \in N_s$ (т.е. $m_{ri} = m_{sj} = 1$) и $N_r < N_s$ в упорядочении P. Нетрудно убедиться, что это определение корректно.

Рассмотрим теперь минимальную раскраску графа I (в $\chi(I)$ цветов)и какие-нибудь две его одинаково раскрашенные вершины. Сети в матрице M, соответсвующие этим вершинам, не должны пересекаться. Более того, эти сети не могут пересекаться и в матрице M_δ в силу выбора δ. Следовательно, все сети матрицы M_δ, соответствующие одинаково окрашенным вершинам, можно уложить на одной дорожке и, стало быть,

$$\chi(I) \geq t(M, \delta) \geq t(M).$$

Окончательно имеем

$$t(M) = \min\{\chi(I)\colon I \text{ — граф интервалов и } E(G) \subseteq E(I)\}.$$

Принимая во внимание задачу 2.8 и теорему **??**, получаем следующее утверждение.

Теорема 2.18 *Для всякой матрицы сетей-ключей M и её графа несовместимости G имеет место равенство*

$$ns(G) = t(M).$$

Мы еще встретимся с параметрами, возникающими при проектировании СБИС.

Комментарии и ссылки

Материал первой главы достаточно традиционной и её содержание частично покрывается книгами по теории графов Веста [47] и Емеличева et al. [48]. Совершенным графам, и, в частности, хордальным и графам интервалов посвящена книга Голамбика [20].

Книга Зыкова [49]. Разнообразные примеры применения графов интервалов и других графов пересечений приводятся в [40]. Классической книгой о графах пересечений является [20]. Новая книга, [32].

Обзоры: [45], [10].

Доказательство того, что проверку хордальности графа можно осуществить за линейное время можно найти в книге [20], а также в курсе лекций Шамира [41] по алгоритмическим аспектам теории совершенных графов, доступному в Интернет по адресу:

`http://www.math.tau.ac.il/ shamir/atga/atga.html`

Обзоры про свойства хордальных графов [8, 31].
Доказательство теоремы 2.13 приводится в [17].
Теорема 5.3: [16]
Теорема 3.3: [11, 15].
Теорема 3.26 [6]
[21]
Теорема 4.2: [36]. Доказательство взято из работы Айгнера и Фрома [1].

Глава 3

PE-упорядочения

1 Характеризация хордальных графов

Обратимся теперь к проблеме характеризации хордальных графов.

Задача 3.1. Доказать, что хордальность является наследственным свойством. Другими словами, всякий порожденный подграф хордального графа является хордальным.

Вершина графа G называется *симплициальной* (simplicial), если её окружение (т.е. множество $N_{G[v_i,\ldots,v_n]}[v_i]$) пусто или является кликой. Мы докажем, что всякий хордальный граф имеет симплициальную вершину. Поскольку хордальность является наследственным свойством (см. предыдущую задачу), это дает итерационную проверку хордальности: ищем и удаляем симплициальную вершину (вместе с инцидентными ей рёбрами, если они существуют). Если такой вершины нет, то граф не является хордальным. Если в результате последовательных удалений симплициальных вершин получается пустой граф, то (это тоже будет доказано) исходный граф является хордальным.

Дадим некоторые важные определения. Упорядочение $(v_1, \ldots, v_n)$ всех вершин графа G называется *PE-упорядочением* (от английского perfect elimination ordering), если для всякого $i \in \{1, \ldots, n\}$ вершина v_i является симплициальной в графе, порождаемом вершинами $\{v_i, \ldots, v_n\}$.

Говорят, что подмножество вершин $S \subset V(G)$ связного графа G *разделяет* несмежные вершины a и b, если в подграфе, порождённом множеством вершин $V(G) - S$, вершины a и b принадлежат различным компонентам связности. Такое множество S мы будем называть (a, b)-*разделителем* или просто *разделителем* (vertex separation, иногда переводится как разделяющее множество вершин, или сепаратор). Нетрудно видеть, что для вершин a и b разделитель существует в том и только в том случае, если они несмежные.

Если никакое собственное подмножество (a, b)-разделителя S не является (a, b)-разделителем, то S называется *минимальным (a, b)-разделителем.*

Задача 3.2. Пусть S — минимальный (a, b)-разделитель в графе G, G_a и G_b — компоненты связности графа $G \setminus S$, содержащие a и b соответственно. Докажите, что каждая вершина множества S смежна по крайней мере одной вершине из $V(G_a)$ и по крайней мере одной вершине из $V(G_b)$.

Введённые выше понятия позволяют сформулировать удобные критерии хордальности.

Теорема 3.3 *(Dirac 61, Fulkerson and Gross 65). Следующие утверждения эквивалентны:*

1. *Граф G хордален.*
2. *Для графа G существует PE-упорядочение. Более того, в качестве первой вершины в этом упорядочении может быть взята любая симплициальная вершина.*
3. *В графе G всякий минимальный разделитель произвольной компоненты связности этого графа является кликой.*

Доказательство.

3)$\Longrightarrow$ 1). Предположим, что в графе есть цикл длины ≥ 4 без хорды вида

$$a, x_1, \ldots, x_k, b, y_1, \ldots, y_l, a,$$

где $k, l \geq 1$ и $x_i \neq y_j$ для всех i, j. Тогда, как нетрудно убедиться, существует минимальный (a, b)-разделитель, который должен содержать хотя бы одну вершину x_i и хотя бы одну вершину y_j. Поскольку этот разделитель по условию является кликой, вершины x_i и y_j должны быть смежными, что противоречит основному свойству рассматриваемого цикла.

1)$\Longrightarrow$ 3). Если G является полным графом, то разделителей в нем нет. Пусть a и b — произвольные несмежные вершины некоторой компоненты связности хордального графа G и S — минимальный (a, b)-разделитель в G. Покажем, что множество S является кликой, т.е. любые две различные вершины s_1, s_2 из S смежны (случай, когда S состоит из одной вершины, тривиален). Обозначим через G_a и G_b компоненты связности подграфа графа $G \setminus S$, содержащие a и b соответственно. Поскольку S

является минимальным разделителем, то каждая из вершин $s_1, s_2 \in S$ имеет «соседей» в G_a и G_b (см. упражнение 3.2). Поэтому из s_1 в s_2 ведет путь, все «внутренние» вершины которого принадлежат $V(G_a)$, и путь, все «внутренние» вершины которого принадлежат $V(G_b)$. Объединение кратчайших путей

$$(s_1, a_1, \ldots, a_k, s_2), \quad a_i \in V(G_a),$$
$$(s_1, b_1, \ldots, b_l, s_2), \quad b_i \in V(G_a)$$

является простым циклом длины ≥ 4. Поскольку G является хордальным графом, в полученном цикле должны быть хорды. Хорд типа (a_i, b_j) быть не может, так как G_a, G_b — различные компоненты связности. Поскольку пути выбирались кратчайшими, то хорд типа (a_i, a_j), (b_i, b_j), (s_1, a_i), (s_1, b_j) (s_2, a_i), (s_2, b_j) в цикле тоже нет. Следовательно, смежными вершинами могут быть только вершины s_1 и s_2.

Для завершения доказательства теоремы нам понадобится лемма.

Лемма 3.4. *(Dirac 61) Всякий хордальный граф G имеет симплициальную вершину. Если к тому же граф не является полным, то он имеет две несмежные симплициальные вершины.*

Докажем лемму индукцией по числу вершин графа. Без труда проверяется, что утверждение леммы справедливо для любого графа G, для которого $|V(G)| \leq 3$. Предположим, что это утверждение имеет место для всякого графа с числом вершин $\leq n-1$, и докажем его справедливость для произвольного хордального графа G с n вершинами. В двух случаях это утверждение очевидно.

1. Граф G — полный. Тогда все его вершины являются симплициальные.
2. Граф G — несвязный. Тогда требуемое утверждение немедленно следует из индукционного предположения.

Таким образом, осталось рассмотреть случай, когда в графе G имеются две несмежные вершины v_1 и v_2, «разделяемые» некоторым минимальным разделителем S. Обозначим через G_i компоненту связности, содержащую вершину v_i в подграфе, порождённом множеством вершин $V(G) \setminus S$, а через A_i — подграф, порождённый множеством вершин $V(G_i) \cup S$ $(i = 1, 2)$. Если граф A_i — полный, то всякая вершина графа G_i является симплициальной в A_i. В противном случае по индукционному предположению в A_i существуют две несмежные вершины, симплициальные в A_i. Обе эти вершины не могут принадлежать S, поскольку по доказанному выше (импликация 1)$\Longrightarrow$ 3)) S является кликой. Следовательно, одна из них принадлежит G_i. Таким образом, в графе G_i всегда существует вершина u_i, симплициальная в A_i. Более того, поскольку G_1,

G_2 — различные компоненты связности, эта вершина является симплициальной и в графе G. По той же причине вершины u_1, u_2 не являются смежными. $\triangle$

Продолжим доказательство теоремы.

1)$\Rightarrow$ 2) Пусть G — хордальный граф с n вершинами. Используя индукцию по числу вершин, без труда убеждаемся, что база ($n = 1$) очевидна, а индукционный переход легко следует из леммы 3.4. В самом деле, по этой лемме в графе G существует симплициальная вершина x. Подграф, порождённый множеством вершин $V(G) \setminus \{x\}$, является хордальным (хордальность наследуется), и для него по индукционному предположению существует PE-упорядочение $(v_1, \ldots, v_{n-1})$. Поскольку вершина x является симплициальной, то $(x, v_1, \ldots, v_{n-1})$ образует PE-упорядочение в G.

2)$\Rightarrow$ 1) Пусть δ — PE-упорядочение в G и C — произвольный цикл в этом графе длины ≥ 4. Обозначим через x первую (при упорядочении δ) вершину этого цикла, а через y, z смежные ей вершины, лежащие на C. Вершины y, z должны быть смежными, поскольку их номера в δ бо́льшие, чем у x, Таким образом, всякий цикл в G длины ≥ 4 имеет хорду. $\square$

Пример 3.5 Разреженные системы линейных уравнений. Рассмотрим систему линейных уравнений

$$Ax = y,$$

где $A = \{a_{ij}\}_{i,j=1}^{n}$ — некоторая разреженная матрица, то есть матрица, «большинство» элементов которой является нулями. По матрице A определим граф $G[A]$ с вершинами $v_1, \ldots, v_n$ и ребрами

$$\{(v_i, v_j)\colon i \neq j, a_{ij} \neq 0 \text{ или } a_{ji} \neq 0\}.$$

Одним из «классических» способов решения систем линейных уравнений является метод последовательного исключения неизвестных, называемый *методом Гаусса.* Напомним, что на каждом «шаге» метода Гаусса исключаемый элемент x_i (предполагается, что $a_{ii} \neq 0$) представляется в виде

$$x_i = -\frac{1}{a_{ii}} \sum_{k \neq i} a_{ik} x_k + \frac{y_i}{a_{ii}}$$

и подставляется во все содержащие его уравнения. В результате исключения неизвестного x_i получается новая матрица A'. Посмотрим, как будет выглядеть граф $G[A']$. Вершины графа $G[A']$ суть вершины $V(G[A]) \setminus v_i$. Если вершины v_k и v_j были смежны вершине v_i в $G[A]$, то после исключения x_i из системы, x_j присутствует в k-ом уравнении, а x_k в j-ом уравнении. Поэтому рёбрами графа $G[A']$ являются рёбра подграфа, порождаемого вершинами $V(G[A]) \setminus v_i$, а также «новые» ребра типа

$$\{(v_j, v_k)\colon v_j, v_k \in N_{G[A]}(v_i)\}.$$

Таким образом, число ненулевых элементов в матрицах A' может сильно зависеть от порядка исключения элементов. В целях экономии компьютерной памяти и повышения эффективности вычислений разреженные матрицы удобно задавать списками ненулевых элементов. В этом случае появление в матрице A' «новых» ненулевых элементов крайне нежелательно. Но если граф $G[A]$ является хордальным, то существует упорядочение δ его вершин такое, что при исключении вершин в порядке δ «новых» рёбер не возникает и для нахождения «оптимальной» последовательности исключения неизвестных в системе уравнений с матрицей A достаточно найти PE-упорядочение вершин графа $G[A]$. Если же граф $G[A]$ не является хордальным, то задача нахождения «наилучшего» порядка исключения неизвестных значительно усложняется. □

Итак, для выяснения того, является ли граф G хордальным, достаточно найти PE-упорядочение вершин этого графа. «Наивным» способом поиска PE-упорядочения является поиск и удаление симплициальных вершин (такая проверка потребует $\mathcal{O}(|V(G)|^3)$ времени). Известно несколько более «изощрённых» и «быстрых» способов распознавания хордальности графа. Одним из них является поиск по наибольшей размерности.

Будем говорить, что упорядочение вершин $\delta = (v_1, \ldots, v_n)$ обладает *MC-свойством* (от английского maximum cardinality), если для всех $i \in \{1, \ldots, n-1\}$ и $j \in \{1, \ldots, i\}$ выполняются неравенства:

$$|N_G(v_i) \cap \{v_{i+1}, v_{i+2}, \ldots, v_n\}| \geq |N_G(v_j) \cap \{v_{i+1}, v_{i+2}, \ldots, v_n\}|.$$

Упорядочение вершин, обладающее MC-свойством находится по следующему простому алгоритму. В этом алгоритме вершина v_n выбирается произвольным образом, а при уже выбранных вершинах $v_{i+1}, v_{i+2}, \ldots, v_n$ вершиной v_i становится вершина, смежная наибольшему числу вершин из уже «упорядоченных» вершин, т.е. вершин $v_{i+1}, v_{i+2}, \ldots, v_n$. Этот алгоритм часто называют *MC-поиском* (maximum cardinality search). Пример работы алгоритма MC-поиска приведён на рисунке 3.1 .

MC-свойство оказывается чрезвычайно полезным при нахождения PE-упорядочений и, следовательно, распознавания хордальности графов. Следующее свойство MC-поиска было обнаружено Tarjan в 1976 году [44].

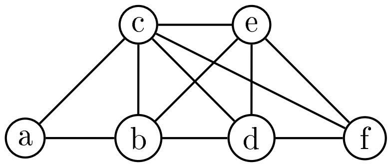


Рис. 3.1: Последняя вершина MC-упорядочения (первая выбираемая MC-поиском вершина) выбирается произвольно. Выберем a. На следующем шаге поиска должна быть выбрана вершина b или c. Выберем c. Тогда далее однозначно выбирается b. Теперь мы опять можем выбрать одну из двух вершин: e или d. Если мы выбираем e, то далее выбор однозначен: d и f. В результате получаем MC-упорядочение $\delta = (f, d, e, b, c, a)$. Обратите, пожалуйста, внимание на то, что δ является и PE-упорядочением. Такое совпадение не случайно.

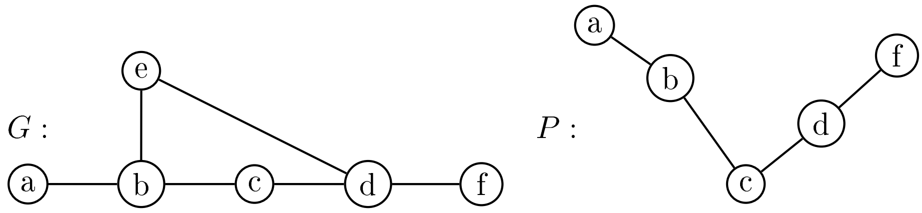


Рис. 3.2: Для упорядочения $\delta = (c, d, b, f, e, a)$ путь $P = (a, b, c, d, f)$ является рвом.

Теорема 3.6 *Граф G является хордальным тогда и только тогда, когда всякое упорядочение $\delta = (v_1, \ldots, v_n)$ вершин графа G, получаемое MC-поиском, является PE-упорядочением.*

Доказательство. Если $\delta = (v_1, \ldots, v_n)$ является PE-упорядочением, то в силу теоремы 3.3 G является хордальным.

Пусть G — хордальный граф, а $\delta = (v_1, \ldots, v_n)$ — произвольное упорядочение вершин графа G, получаемое MC-поиском. Убедимся в том, что δ — PE-упорядочение.

Будем называть *рвом* (при упорядочении δ) путь P длины ≥ 2 без хорд (т.е. путь, вершины которого порождают граф без циклов) такой, что из всех вершин P наибольшими номерами (при упорядочении δ) обладают начало и конец P. Название ров возникло из-за картинки, получаемой при «нанесении» вершин пути P на плоскость. Каждая вершина имеет вертикальную координату равную порядковому номеру в δ и горизонтальную равную порядковому номеру этой вершины в пути P (см. пример на рисунке 3.2). Докажем сначала, что при упорядочении δ в G нет рвов. Предположив противное, из всех возможных рвов выберем (v_i, v_j)-путь P для концов которого достигается максимум $\min\{i, j\}$. Не умаляя общности можно считать, что $i < j$. По определению рва, все «внутренние» вершины $v \in V(P)$ имеют порядковые номера $\delta(v) < i$. Обозначим через x «внутреннюю» вершину пути P, смежную вершине v_j. По предположению, $\delta(x) < i < j$, а потому вершина v_i в MC-алгоритме выбирается перед x и вершина v_j к этому моменту уже «упорядочена». Но то, что v_i выбирается перед x означает существование вершины $y \in N(v_i) \setminus N(x)$ для которой $\delta(y) > i$. В подграфе, порождаемом вершинами $V(P) \cup \{y\}$ есть кратчайший путь P_1 с концами v_i и y. Учитывая хордальность графа G, нетрудно убедиться в том, что этот же путь является кратчайшим путём, связывающим v_i с y и в G. Более того, P_1 является рвом и $\min\{\delta(y), j\} > \min\{i, j\}$. Но последнее противоречит выбору вершин v_i, v_j.

Таким образом, вершина v_1 является симплициальной вершиной, поскольку в противном случае в G существовал бы кратчайший путь v_i, v_1, v_j для некоторых несмежных «соседей» вершины v_1. Теперь теорема легко доказывается индукцией по числу вершин n. Действительно, если v_1 является симплициальной вершиной, то применение алгоритма к графу $G \setminus \{v_1\}$ даёт упорядочение $(v_2, \ldots, v_n)$ и по индукционному предположению $(v_2, \ldots, v_n)$ является PE-упорядочением вершин графа $G \setminus \{v_1\}$. Вершина v_1 — симплициальна, следовательно, $(v_1, v_2, \ldots, v_n)$ — PE-упорядочение графа G. □

Можно показать (мы этого делать не будем, отсылая читателя к работам, ссылки на которые даны в конце главы), что при разумном представлении данных время работы алгоритма поиска по наибольшей размерности есть $\mathcal{O}(|E(G)| + |V(G)|)$. Оказывается, что проверку того, является ли найденное алгоритмом упорядочение PE-упорядочение (а, следовательно, и того, является ли граф хордальным), тоже можно выполнить за линейное время.

В заключение несколько задач.

Задача 3.7. Верно ли, что всякое упорядочение, «обратное» к PE-упорядочению обладает свойством наибольшей размерности?

Задача 3.8. Пусть $\delta = (v_1, \ldots, v_n)$ — PE-упорядочение хордального графа G. Докажите, что

$$\omega(G) - 1 = \max_{i \in \{1,\ldots,n\}} deg_{G[v_i,\ldots,v_n]}(v_i)$$

и

$$|E(G)| = \sum_{i=1}^{n} deg_{G[v_i,\ldots,v_n]}(v_i).$$

(Напомним, что $deg_{G[v_i,\ldots,v_n]}(v_i)$ обозначает степень вершины v_i в подграфе, порождаемом вершинами $\{v_i, \ldots, v_n\}$, а ω — кликовое число.)

Задача 3.9. Докажите, что если хордальный граф G не является полным и размер наибольшей клики $\omega(G) = k$, то в G найдутся две несмежные вершины степени $\leq k - 1$.

Задача 3.10. Пусть $\delta = (v_1, v_2, \ldots, v_n)$ — PE-упорядочение вершин хордального графа G. Докажите, что всякая максимальная клика графа G совпадает с одним из множеств

$$K_i = N_G[v_i] \cap \{v_i, v_{i+1}, \ldots, v_n\}.$$

Задача 3.11. Докажите, что для всякого хордального графа G с m ребрами, n вершинами и кликовым числом k выполняется неравенство $m \leq n(k-1) - \frac{1}{2}k(k-1)$.

Задача 3.12. Докажите, что число максимальных клик хордального графа G не превосходит $|V(G)|$, причем равенство достигается только тогда, когда в графе нет ребер.

Задача 3.13. Доказать, что граф является хордальным тогда и только тогда, когда всякий его порожденный связный подграф с $p \geq 2$ вершинами имеет не более $p - 1$ максимальных клик.

Задача 3.14. Докажите, что рёберный граф $L(G)$ графа G является хордальным только если граф G хордален. Покажите, что обратное утверждение не верно.

Задача 3.15. Докажите, что всякая не содержащая симплициальных вершин максимальная клика хордального графа является разделителем.

2 k-деревья

По теореме 3.3 всякий хордальный граф можно уразобратьф, последовательно удаляя симплициальные вершины. Эту же теорему можно трактовать и как уруководство по сборкеф хордальных графов, поскольку всякий хордальный граф можно получить из хордального графа на единицу меньшего порядка, уприклеиваяф к нему определённым образом вершину. Действительно, пусть $(v_1, \ldots, v_n)$ — PE-упорядочение вершин хордального графа G. Для всякого $i \in \{1, \ldots, n-1\}$ граф G_{i+1}, порождаемый вершинами $v_{i+1}, \ldots, v_n$ является хордальным. Тогда граф G_i получается из G_{i+1} следующим образом: в G_{i+1} выделяется некоторая клика C и новая вершина v_i объявляется смежной каждой вершине из C и несмежной остальным вершинам из G_{i+1}.

Важным подклассом хордальных графов являются *k-деревъя.* Неформально этот класс можно определить как класс хордальных графов, которые можно получить из $(k+1)$-клики, уприклеиваяф вершины к k-кликам.

Для каждого натурального числа k множества T_n^k всех k-деревьев, имеющих n вершин, определяется индуктивно для $n \geq k+1$. Множество T_{k+1}^k состоит, по определению, из одного элемента, который представляет собой полный граф с $k+1$ вершиной. Предположим, что множества T_m^k построены для всех $m < n$ и каждое из них содержит k-клику. Тогда множество T_n^k определяется следующим образом. Каждый его элемент сопоставляется паре (G, G_1), где G — некоторый граф из T_{n-1}^k, а G_1 — одна из его k-клик. Этот элемент имеет следующий вид. К графу G удобавляетсяф новая вершина, которая объявляется смежной каждой вершине из G_1 и несмежной остальным $n-k$ вершинам из G.

Следующая задача оправдывает принятую терминолгию.

Задача 3.16. Доказать, что деревья являются 1-деревьями.

Покажем теперь, что k-деревья являются хордальными графами.

Лемма 3.17. *Граф G с n вершинами является k-деревом тогда и только тогда, когда $n > k$ и существует такое PE-упорядочение $\delta = \{v_1, \ldots, v_n\}$ его вершин, что для всякого $i \leq n-k$ вершина v_i смежна каждой вершине некоторой k-клики в подграфе, порождённом вершинами $v_i, \ldots, v_n$, и несмежна остальным вершинам этого подграфа, т.е. $deg_{G[v_i,\ldots,v_n]}(v_i) = k$.*

Доказательство. Пусть G — произвольное k-дерево с n вершинами. Если $n = k+1$, то G является полным графом и всякое упорядочение вершин обладает требуемым свойством. Предположим, что для всех k-деревьев с числом вершин $< n$ требуемое упорядочение существует. По определению k-дерева существует такая вершина x, что граф $G \setminus x$ является k-деревом с $n-1$ вершиной, содержащим k-клику, каждая вершина которой смежна вершине x. Пусть $\{v_2, \ldots, v_n\}$ — PE-упорядочение вершин графа $G \setminus x$, существующее в силу индукционного предположения. Тогда упорядочение $\{x, v_2, \ldots, v_n\}$ будет искомым упорядочением для графа G.

Утверждение ув обратную сторонуф является очевидным следствием определения k-дерева и того факта, что вершины $v_{n-k+1}, \ldots, v_n$ в PE-упорядочении δ порождают k-клику. $\square$

Из леммы 3.17 и теоремы 3.3 вытекает

Теорема 3.18 *Для произвольного натурального k любое k-дерево является хордальным графом.*

С учётом леммы 3.17 следующие две леммы очевидны.

Лемма 3.19. *Кликовое число k-дерева равно $k+1$.*

Лемма 3.20. *Число t-клик в k-дереве, $1 \le t \le k+1$, равно $C_k^t + (n-k)C_k^{t-1}$.*

Следующая теорем содержит удобную характеризацию k-дерева.

Теорема 3.21 *Граф G, $|V(G)| = n$, является k-деревом тогда и только тогда, когда*

1. *G является хордальным графом;*
2. *кликовое число G равно $k+1$;*
3. *$|E(G)| \ge nk - \frac{1}{2}k(k+1)$.*

Доказательство. Если граф G является k-деревом, то в силу теоремы 3.18 и леммы 3.19 он является хордальным графом с кликовым числом $k+1$. По лемме 3.20 число ребер в G (2-клик) равно $nk - \frac{1}{2}k(k+1)$.

Предположим теперь, что граф G удовлетворяет условиям 1–3. Покажем, что в этом случае он является k-деревом. В самом деле, так как граф G является хордальным, то $|E(G)| \le nk - \frac{1}{2}k(k+1)$ (см. задачу 3.11), а потому

$$|E(G)| = nk - \frac{1}{2}k(k+1). \qquad (*)$$

С другой стороны, хордальность графа G гарантирует существование РЕ-упорядочения $\delta = \{v_1, \ldots, v_n\}$ его вершин. Поскольку

$$deg_{G[v_i,\ldots,v_n]}(v_i) \le k$$

и

$$|E(G)| = \sum_{i=1}^{n} deg_{G[v_i,\ldots,v_n]}(v_i)$$

(см. задачу 3.8), то, используя $(*)$ и очевидное неравенство

$$\sum_{i=n-k+1}^{n} deg_{G[v_i,\ldots,v_n]}(v_i) \le \frac{k(k-1)}{2},$$

получаем, что для всякого $i \in \{1, \dots, n-k\}$

$$deg_{G[v_i,\dots,v_n]}(v_i) = k.$$

Таким образом, оказывается, что упорядочение δ удовлетворяет условию леммы 3.17 и, следовательно, граф G является k-деревом. $\square$

Задача 3.22. Доказать, что для всякого графа G с n вершинами условия 1)–3) теоремы 3.21 эквивалентны следующим трем условиям:

1. G связный граф;

2. кликовое число G равно $k+1$;

3. Всякий минимальный разделитель в G является k-кликой.

Мы еще вернемся к k-деревьям при изучении путевых и древесных декомпозиций графов.

3 Совершенные графы

Напомним, что правильной k-раскраской графа G (в дальнейшем мы часто будем опускать слово правильная) называется такое разбиение (раскраска) вершин $V(G)$ на k непересекающихся классов, что никакие две вершины из одного класса не смежны. Напомним, что множество вершин графа называется *независимым*, если никакие две вершины этого множества не смежны. Таким образом, вершины одного класса образуют независимое множество. В этом случае будем говорить также, что граф G может быть (правильно) раскрашен в k цветов. Удобно считать, что эти цвета пронумерованы натуральными числами от 1 до k. Наименьшее k, для которого существует k-раскраска графа G называется хроматическим числом графа G и обозначается через $\chi(G)$. Правильная раскраска графа G в $\chi(G)$ цветов называется *минимальной раскраской*. Поскольку для полного графа его хроматическое число совпадает с числом вершин, то для всякого графа G выполняется неравенство $\omega(G) \leq \chi(G)$ (через $\omega(G)$ мы условились обозначать размер наибольшей (по числу элементов) клики графа G). Понятие совершенного графа было введено К. Бержем. Граф G называется *совершенным* (perfect graph), если у G и всех его порожденных подграфов кликовые и хроматические числа совпадают. К примеру, всякий двудольный граф является совершенным, поскольку если в двудольном графе G есть хоть одно ребро, то $\omega(G) = \chi(G) = 2$, а если ребер нет, то $\omega(G) = \chi(G) = 1$.

Число вершин в наибольшем независимом множестве графа G называется *числом независимости* этого графа и обозначается через $\alpha(G)$.

Пример 3.23. Целочисленное программирование. Рассмотрим совокупность всех максимальных независимых множеств $(A_1, A_2, \ldots, A_m)$ графа G. Построим $(n \times m)$ матрицу $A(G) = \{a_{ij}\}_{i \in \{1,\ldots,n\}, j \in \{1,\ldots,m\}}$ графа G, где

$$a_{ij} = \begin{cases} 1, & \text{если } v_i \in A_j, \\ 0, & \text{в противном случае.} \end{cases}$$

Тогда сумма $\sum_{j=1}^{m} a_{ij}$ равняется числу независимых множеств, содержащих вершину v_i. Пусть $\mathbf{x}$ — вектор $\mathbb{R}^m$, каждая из координат которого равна 0 или 1, а $\mathbf{1}_n$ есть вектор из $\mathbb{R}^n$ все координаты которого равны 1. Тогда ненулевые координаты вектора $\mathbf{x}$, удовлетворяющего условию

$$A\mathbf{x} \geq \mathbf{1}_n,$$

«порождают» покрытие вершин графа G максимальными независимыми множествами. А поскольку раскраска графа есть разбиение вершин графа на независимые множества, то всякое решение $\mathbf{x}^0$ следующей задачи целочисленного линейного программирования

$$\begin{aligned} &\min < \mathbf{1}_m, \mathbf{x} > \\ &A\mathbf{x} \geq \mathbf{1}_n, \\ &\mathbf{x}_i \in \{0,1\}, \qquad i \in \{1,\ldots,m\}, \end{aligned}$$

удовлетворяет равенству

$$< \mathbf{1}_m, \mathbf{x}^0 >= \chi(G).$$

Двойственной к этой задаче будет задача

$$\begin{aligned} &\max < \mathbf{1}_n, \mathbf{p} > \\ &A^*\mathbf{p} \leq \mathbf{1}_m, \\ &\mathbf{p}_i \in \{0,1\}, \qquad i \in \{1,\ldots,n\}. \end{aligned}$$

Всякое решение $\mathbf{p}^0$ двойственной задачи удовлетворяет равенству

$$< \mathbf{1}_n, \mathbf{p}^0 >= \omega(G).$$

Решения прямой и двойственных задач связывает отношение слабой двойственности:

$$< \mathbf{1}_m, \mathbf{x}^0 > \geq < A^*\mathbf{p}^0, \mathbf{x}^0 > = < \mathbf{p}^0, A\mathbf{x}^0 > \geq < \mathbf{p}^0, \mathbf{1}_n > .$$

Переформулируя это неравенство в терминах хроматического и кликового числа получаем $\chi(G) \geq \omega(G)$. Если же граф является совершенным, то должно выполняться равенство

$$\chi(G) = < \mathbf{1}_m, \mathbf{x}^0 > = < \mathbf{p}^0, \mathbf{1}_n > = \omega(G).$$

Из общей теории линейного программирования известно, что при отсутствии в условии задачи требования на целочисленность значения целевых функционалов в прямой и двойственной задаче совпадают. Неприятность в том, что решение задачи линейного программирования может и не быть целым. Идея формулировки задач теории графов как задач линейного программирования послужили базисом создания очень интересной области теории графов — дробной теории графов (fractional graph theory). В этой теории хроматическое число графа может и не быть целым, но обязано совпадать с кликовым числом!

$\square$

Задача 3.24. Сформулируйте задачи нахождения чисел $\alpha(G)$ и $\theta(G)$ (размера наибольшего независимого множества и наименьшего числа клик, покрывающих вершины G) как задачи целочисленного линейного программирования.

Задача 3.25. Найдите дробное хроматическое и дробное кликовое число цикла длины пять.

Покажем, что хордальные графы совершенны.

Теорема 3.26 *(Berge 61) Хордальные графы совершенны.*

Доказательство. Индукция по n — числу вершин графа. База индукции: $n = 1$ — очевидна. Будем считать, что для всех хордальных графов с числом вершин $\leq n$ теорема верна. Выберем произвольный хордальный граф G с $n+1$ вершиной. Для обоснования индукционного перехода достаточно показать, что $\omega(G) = \chi(G)$. Если G является полным, то равенство выполняется. В противном случае в G есть несмежные вершины, а потому в G есть и минимальный разделитель S, который по теореме 3.3 является кликой.

Пусть $G_1, \ldots, G_k$ — компоненты связности графа $G \setminus S$. Обозначим через A_i подграф G, порождаемый вершинами $V(G_i) \cup S$, $i \in \{1, \ldots, k\}$. Покажем, что $\chi(G) = \max\{\chi(A_1), \ldots, \chi(A_k)\}$. Пусть для определённости $\max\{\chi(A_1), \ldots, \chi(A_k)\} = \chi(A_1)$. Непосредственно из определения хроматического числа следует, что $\chi(A_1) \leq \chi(G)$. С другой стороны, нетрудно убедиться, что граф G можно правильно раскрасить с помощью $\chi(A_1)$

цветов. В самом деле, в минимальной правильной раскраске подграфа A_1 $|S|$ цветов использованы для раскраски вершин клики S. Легко видеть, что с помощью остальных $\chi(A_1) - |S|$ цветов можно раскрасить вершины подграфа A_i, не принадлежащие S, так, чтобы раскраска A_i была правильной ($i \in \{1, \ldots, k\}$). Полученная таким образом раскраска всех вершин графа G является, очевидно, правильной. Пусть K — некоторая максимальная клика графа G. Поскольку S является разделителем, то для некоторого $i \in \{1, \ldots, k\}$ K является подмножеством $V(A_i)$. Поэтому $\omega(G) = \max\{\omega(A_1), \ldots, \omega(A_k)\}$.

Осталось заметить, что всякий граф A_i, порождаемый множеством вершин $V(G_i) \cup S$, $i \in \{1, \ldots, k\}$, является хордальным и, стало быть, для него индукционное предположение выполняется. Таким образом, для всех $i \in \{1, \ldots, k\}$ $\omega(A_i) = \chi(A_i)$ и, следовательно, $\omega(G) = \chi(G)$. □

Теорема 3.3 характеризует хордальные графы в терминах PE-упорядочений и в терминах минимальных разделителей. Используя характеризацию хордального графа, как графа, у которого всякий минимальный разделитель является кликой, мы доказали, что хордальные графы совершенны. С помощью PE-упорядочений легко доказывается

Теорема 3.27 *(Haknal-Suranyi 58, Dirac 61). Если G — хордальный граф, то $\overline{G}$ — совершенный граф.*

Доказательство. Если в G одна вершина, то утверждение очевидно. Предположим, что для всех хордальных графов с числом вершин $< n$ утверждение доказано. Для доказательства теоремы достаточно убедиться, что для произвольного хордального графа G с n вершинами справедливо неравенство $\chi(\overline{G}) \leq \omega(\overline{G})$. Пусть x — одна из симплициальных вершин графа G и K — максимальная клика графа $\overline{G}$. Поскольку K является максимальной кликой, то она должна содержать вершину множества $N_G[x]$ — замкнутого окружения x в G, так как в противном случае $K \subset N_{\overline{G}}[x]$ (в силу симплициальности вершины x в G). Поэтому при удалении множества вершин $N_G[x]$ из $\overline{G}$ кликовое число должно уменьшиться по крайней мере на 1, т.е.

$$\omega(\overline{G}) \geq \omega(\overline{G} \setminus N_G[x]) + 1. \qquad (*)$$

При этом хроматическое число не может уменьшиться более, чем на 1, т.е.

$$\chi(\overline{G} \setminus N_G[x]) + 1 \geq \chi(\overline{G}). \qquad (**)$$

В самомо деле, правильную раскраску можно получить следующим образом: выбрать минимальную правильную раскраску графа $G \setminus N_G[x]$ и использовать ещё один цвет для окраски вершин из множества $N_G[x]$, которое в силу симплициальности вершины x является независимым в $\overline{G}$. Учитывая индукционное предположение и неравенства $(*)$, $(**)$, получаем требуемую оценку: $\chi(\overline{G}) \leq \omega(\overline{G})$. □

На самом деле, доказанный факт является следствием теоремы 3.26 и нижеследующего более сильного утверждения. Это утверждение в виде гипотезы сформулировано К. Бержем в 1961 и было доказано двадцатидвухлетним Л. Ловасом. Доказательство этой красивой теоремы можно найти во многих учебниках по теории графов и мы его опускаем.

Теорема 3.28 *(Lovász 72) Граф G является совершенным тогда и только тогда, когда совершенным является граф $\overline{G}$.*

Задача 3.29. Докажите, что граф G является совершенным тогда и только тогда, когда для всякого порожденного подграфа графа G кликовое покрытие и число независимости $\alpha(G)$ совпадают.

Задача 3.30. Пусть $\delta = (v_1, v_2, \dots, v_n)$ — PE-упорядочение вершин хордального графа G. Для $i \in \{1, \dots, n\}$ определим

$$K_i = N_G[v_i] \cap \{v_i, v_{i+1}, \dots, v_n\}.$$

Зададим последовательность вершин

$$y_1, y_2, \dots, y_t$$

индуктивно. Положив $y_1 = v_1$, определим y_i, $i \in \{2, \dots, n\}$ как вершину с наименьшим номером $\delta(y_i) = j$ таким, что

1. $\delta(y_i) > \delta(y_{i-1})$ и
2. $y_i \notin K_{\delta(y_1)} \cup K_{\delta(y_2)} \cup \dots \cup K_{\delta(y_{i-1})}$.

Докажите, что множество $y_1, y_2, \dots, y_t$ является наибольшим независимым множеством в G, т.е. $t = \alpha(G)$.

Совершенные графы можно охарактеризовать и следующим образом (теорема приводится без доказательства):

Теорема 3.31 *(Lovász 72) Граф G является совершенным тогда и только тогда, когда для всякого порожденного подграфа H*

$$\alpha(H)\omega(H) \geq |V(H)|.$$

4 Сцепление и ширина графа

Всякое упорядочение $\delta = (v_1, \ldots, v_n)$ вершин графа G «порождает» (не обязательно оптимальную) раскраску. Окрасим вершину v_1 в первый цвет. Когда вершины $v_1, \ldots, v_i$ уже окрашены, то вершина v_{i+1} красится в наименьший возможный цвет. На такую раскраску потребуется не более

$$max_{i \in \{1,\ldots,n\}} deg_{G[v_1,\ldots,v_i]}(v_i) + 1$$

цветов, поскольку для вершины v_i число смежных ей и окрашенных до нее вершин равно степени этой вершины в графе, порождаемым вершинами $\{v_1, \ldots, v_i\}$.

Шириной вершины $v \in V(G)$ при упорядочении δ называется число смежных v вершин, предшествующих v в упорядочении δ. Иными словами, при упорядочении $\delta = (v_1, v_2, \ldots, v_n)$ ширина вершины v_i равна

$$|N_G(v_i)| \cap \{v_1, \ldots, v_{i-1}\}$$

— мощности окружения этой вершины в графе $G[v_1, \ldots, v_i]$. *Шириной графа, соответствующей упорядочению* δ называется наибольшая из ширин вершин при этом упорядочении. *Шириной графа* G, $w(G)$, (width, другое название — цветное число (colouring number)) называется наименьшая из ширин графа, соответствующих всевозможным упорядочениям вершин. Очевидно, что для всякого графа G

$$\chi(G) \leq w(G) + 1 \leq \Delta(G) + 1.$$

Мы часто будем встречаться с инвариантами графов (ширина ленты, разреза, величина вершинного разделения и т.д.), определяемых как минимум некоторого параметра по $n!$ возможным упорядочениям вершин графа. Как правило, задачи нахождения таких инвариантов NP-трудны. На первый взгляд ширина графа является одним из таких трудно-вычислимых инвариантов, но это не так.

Известен простой и быстрый способ нахождения ширины графа, использующий следующее минимаксное утверждение для $w(G)$. Обозначим через $d(G)$ наименьшую из степеней вершин графа G. Сцеплением графа G называется

$$linkage(G) = \max\{d(H) \colon H \text{ является подграфом } G\}.$$

Лемма 3.32. *Для всякого графа G, $w(G) = linkage(G)$.*

Доказательство. Докажем сперва, что $w(G) \leq linkage(G)$. Пусть $linkage(G) = k$. Тогда всякий подграф графа G имеет вершину степени $\leq k$. В графе

$G_0 \stackrel{\triangle}{=} G$ выберем вершину v_1 степени $\leq k$. В графе $G_1 \stackrel{\triangle}{=} G \setminus v_1$ тоже есть вершина v_2, степень которой $\leq k$ и т.д. Степень каждой из вершин v_i в графе G_{i-1} не превосходит k, поэтому ширина графа, соответствующая упорядочению $\delta = (v_n, v_{n-1}, \ldots, v_1)$, не превосходит k. Следовательно, $w(G) \leq k$.

В другую сторону. Выберем подграф H графа G для которого $d(H) = linkage(G) = k$. Тогда $w(H) \geq k$. С другой стороны, очевидно, что ширина графа является наследуемым свойством, т.е. ширина всякого подграфа графа G не превосходит $w(G)$. Таким образом, $w(G) \geq w(H) \geq k$. □

Опишем «жадный» алгоритм нахождения оптимального (для ширины графа) упорядочения вершин. В этом алгоритме вершиной v_n выбирается вершина наименьшей степени в графе G. При уже выбранных вершинах $v_{i+1}, v_{i+2}, \ldots, v_n$ вершиной v_i становится вершина, смежная наименьшему числу вершин из ещё «неупорядоченных» вершин, т.е. вершин $V(G) \setminus \{v_{i+1}, v_{i+2}, \ldots, v_n\}$. Например, применяя алгоритм к графу на рисунке 3.1, можно получить (одно из возможных) упорядочение $\{d, f, c, e, b, a\}$. Данный алгоритм полезно сравнить с алгоритмом МС-поиска, приводимого на стр. 29.

Теорема 3.33 *Упорядочение вершин графа G $\delta = (v_1, \ldots, v_n)$, получаемое в результате работы алгоритма является оптимальным (для ширины графа).*

Доказательство. В самом деле, при выборе вершины v_i, $i \in \{1, \ldots, n\}$, выполняется условие

$$deg_{G[v_1, v_2, \ldots, v_i]}(v_i) = d(G[v_1, , v_2, \ldots, v_i]).$$

Из определений ширины и сцепления графа получаем:

$$\begin{aligned} w(G) &\leq max_{i \in \{1,\ldots,n\}} deg_{G[v_1,v_2,\ldots,v_i]}(v_i) = \\ &= max_{i \in \{1,\ldots,n\}} d(G[v_1, v_2, \ldots, v_i]) \leq \\ &\quad \leq linkage(G). \end{aligned}$$

По лемме 3.32 ширина и сцепление графа совпадают, следовательно, упорядочение δ является оптимальным. □

Задача 3.34. Пусть $\delta = (v_1, v_2, \ldots, v_n)$ есть РЕ-упорядочение вершин хордального графа G. Докажите, что $\delta^{-1} = (v_n, \ldots, v_2, v_1)$ является оптимальным (для ширины) упорядочением вершин графа G.

Задача 3.35. Является ли верным утверждение, обратное к утверждению задачи 3.34? Иными словами, верно ли, что всякое обратное к оптимальному (для ширины) упорядочение вершин хордального графа является PE-упорядочением?

Задача 3.36. Докажите, что для всякого хордального графа G $w(G) = \chi(G) - 1$.

Задача 3.37. Докажите, что для всякого графа G $w(G) \leq ns(G) - 1$.

Комментарии и ссылки

Теорема 3.3 является объединением результатов работ [11, 15].

Доказательство того, что проверку хордальности графа можно осуществить за линейное время можно найти в книге [20], а также в курсе лекций Шамира [41] по алгоритмическим аспектам теории совершенных графов, доступному в Интернет по адресу:

http://www.math.tau.ac.il/ shamir/atga/atga.html

Вместо MC-поиска для нахождения PE-упорядочений можно использовать лексикографический поиск в ширину (LexBFS).

Начальные факты о совершенных графах можно найти в книгах по теории графов Веста [47] и Емеличева et al. [48]. Изучению «алгоритмических» свойств совершенных графов посвящена книга Голамбика [20]. Для дальнейшего изучения совершенных графов и раскрасок мы можем порекомендовать обзор [45], книги [10] и [24].

Теорема 3.26 [6] [21]

Глава 4

Игра «полицейские и грабитель»

1 Постановка задачи

Традиционно именно эти персонажи фигурируют в моделях преследования, изучаемых в зарубежной литературе. Два игрока, называемые уполицейскийф и уграбительф, играют на связном неориентированном конечном графе G. В распряжении полицейского имеется k фишек, у грабителя лишь одна фишка. Правила игры таковы. Сначала полицейский расставляет свои фишки на некоторые вершины графа G, после чего вершину выбирает грабитель и ставит на нее свою фишку. Далее игроки по очереди передвигают свои фишки вдоль рёбер графа. При своём ходе грабитель и полицейский могут переставить каждую из своих фишек в смежную вершину или оставить её на месте. При этом каждый из них делает свой выбор, обладая полной информацией о местоположении соперника (и, разумеется, своём собственном).

Назовём *позицией* упорядоченный набор вершин $(p_1, p_2, \ldots, p_k, r)$ в котором первые k составляющих соответствуют положению k полицейских, а последняя составляющая — положению грабителя (полицейским не запрещается занимать одну вершину одновременно и для некоторых $i \neq j$ возможно совпадение вершин p_i, p_j). *Позиционной стратегией полицейских* назовём пару, состоящую из набора вершин $(p_1^1, p_2^1, \ldots, p_k^1)$ графа G (занимаемых полицейскими на первом ходу) и отображения

$$\delta\colon \underbrace{V(G) \times \cdots \times V(G)}_{k+1} \to \underbrace{V(G) \times \cdots \times V(G)}_{k},$$

определяющего выбор полицейского начиная со второго хода и сопоставляющего каждой позиции $(p_1, p_2, \ldots, p_k, r)$ набор вершин

$$\delta(p_1, p_2, \ldots, p_k, r) = (p'_1, p'_2, \ldots, p'_k)$$

такой, что для всякого индекса $i \in \{1, \ldots, k\}$

$$p'_i \in N[p_i].$$

Позиционной стратегией грабителя называется пара, состоящая из отображения

$$\sigma\colon \underbrace{V(G) \times \cdots \times V(G)}_{k} \to V(G),$$

(определяющее выбор грабителя на первом ходу в ответ на первый ход полицейского) и отображения

$$\gamma\colon \underbrace{V(G) \times \cdots \times V(G)}_{k+1} \to V(G),$$

сопоставляющее каждой позиции $(p_1, p_2, \ldots, p_k, r)$ вершину

$$\gamma(p_1, p_2, \ldots, p_k, r) \in N[r]$$

(которое определяет выбор грабителя начиная со второго хода).

Полицейский побеждает, если ему удается поместить одну из своих фишек на одну вершину с фишкой грабителя (поймать его). Соответствующую стратегию будем называть выигрывающей. Наименьшее число фишек у игрока-полицейского (число k), необходимое ему для победы на графе G, называется *полицейским числом* (cop-number) графа G и обозначается через $cop(G)$.

2 Один полицейский

Изучим сперва случай, когда в распоряжении полицейского имеется только одна фишка. Будем говорить, что граф G принадлежит классу $\mathcal{C}$, если полицейское число этого графа равно одному (т.е. один полицейский обладает на этом графе выигрывающей стратегией).

Рассмотрим граф $G \in \mathcal{C}$. Какая ситуация может возникнуть перед последним ходом грабителя? Пусть грабитель находится в вершине r, а полицейский в вершине p. Тогда, как нетрудно видеть,

$$N(r) \cup \{r\} \subseteq N(p).$$

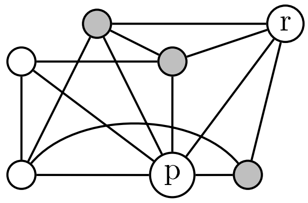


Рис. 4.1: Ловушка r и ее доминирующая вершина p.

Вершину r будем называть *ловушкой*, если для неё существует такая вершина p, что справедливо предыдущее включение. В том случае p называется вершиной, *доминирующей* ловушку r (см. рис. 4.1). Отметим, что если в графе G нет ловушек, то $G \notin \mathcal{C}$. Например, на графе октаэдра, изображенного на рисунке 5.3 стр. 68, полицейский не сможет поймать грабителя, поскольку в этом графе нет ловушек. Во всяком цикле длины большей трех тоже нет ловушек.

Лемма 4.1. *Если вершина r является ловушкой в графе G, то $G \in \mathcal{C}$ $\Leftrightarrow G' \triangleq G \setminus r \in \mathcal{C}$.*

Доказательство. Пусть p — вершина, доминирующая ловушку r в графе G. Если $G' \in \mathcal{C}$, то выигрывающую стратегию полицейского в G' можно урасширитьф до выигрывающей стратегии в G следующим образом. Когда грабитель встает в вершину r, то полицейский реагирует так, будто грабитель встает в вершину p в G'. На все остальные движения грабителя полицейский реагирует в графах G' и G одинаково.

Если $G' \notin \mathcal{C}$, то грабитель, в свою очередь, тоже может урасширитьф свою выигрывающую стратегию так, как это делал полицейский в предыдущем случае. $\square$

Теорема 4.2 *(Nowakowski-Winkler, 1983) Граф G принадлежит классу $\mathcal{C}$ тогда и только тогда, когда последовательным удалением ловушек (в любом порядке) граф преобразуется в одну вершину.*

Доказательство. По предыдущей лемме, принадлежность классу $\mathcal{C}$ не может измениться при удалении ловушки. Таким образом, удаляя ловушки, мы в конце концов получим либо граф, состоящий из одной вершины (в этом случае $G \in \mathcal{C}$), либо граф с бо́льшим чем единица числом вершин и без ловушек (в этом случае $G \notin \mathcal{C}$). $\square$

Пример нетривиального графа из $\mathcal{C}$ приведён в верхней части рисунка 4.2.

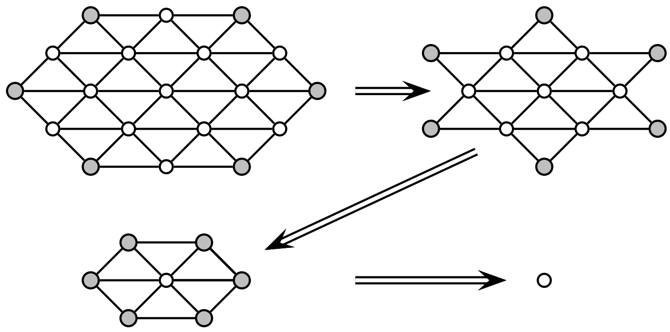


Рис. 4.2: Последовательное удаление ловушек. Удаляя на каждом шаге закрашенные ловушки, получаем точку.

Доказанная характеризация графов из $\mathcal{C}$ позволяет установить связь этого класса с хордальными графами.

Теорема 4.3 *Следующие утверждения эквивалентны.*

1. *G — хордальный граф.*

2. *Всякий связный порожденный подграф графа G принадлежит классу $\mathcal{C}$.*

Доказательство.

1)⇒ 2) Поскольку хордальность наследуется, всякий связный порождённый подграф графа G хордален, и, следовательно, для него существует совершенное исключающее упорядочение $\{v_1, v_2, \ldots, v_n\}$. Так как всякая симплициальная вершина, имеющая непустое окружение, является ловушкой, увыбрасываниеф вершин $v_1, v_2, \ldots, v_{n-1}$ не нарушает принадлежности классу C и приводит к единственной вершине v_n.

2)⇒ 1) Если G не является хордальным графом, то найдется порожденный цикл длины ≥ 4 без хорд, на котором полицейский не может поймать грабителя. □

Задача 4.4. Приведите пример не хордального графа из $\mathcal{C}$.

3 Полицейское число

Увеличим ушансы на выигрышф игрока-полицейского, добавив ему фишек. Аналога теоремы 4.2, позволяющего охарактеризовать (при помощи упорядочений, запрещенных подграфов или каких-нибудь других уинструментовф теории графов) графы с полицейским числом $k > 1$ нам неизвестно и мы вынуждены ограничиться лишь приведением нескольких (но очень красивых) оценок полицейского числа.

Первый вопрос, на который мы дадим ответ, это вопрос об ограниченности полицейского числа на множестве всех графов. Иными словами, верно ли, что для некоторой константы c на всяком графе c полицейских обладают выигрывающей стратегией? Следующие две теоремы дают отрицательный ответ на поставленный вопрос.

Напомним, что через $d(G)$ обозначается наименьшая из степеней вершин графа G. При изучении ширины графа (см. стр. 40) мы определяли сцепление графа G как

$$linkage(G) = \max\{d(H)\colon H \text{ является подграфом } G\}.$$

Теорема 4.5 *(Aigner-Fromme 84) Если в графе G нет порожденных циклов длины три или четыре, то $cop(G) \geq linkage(G)$.*

Доказательство. Выберем порожденный подграф H графа G для которого $d(H) = linkage(G) = n$ и покажем, что грабитель, оставаясь на подграфе H может уклониться от встречи с $n-1$ полицейскими.

Для доказательства теоремы достаточно показать, что для всякого множества $C \subseteq V(G)$ мощности $\leq n-1$ найдутся вершина $w \in V(H) \setminus C$ и вершина $w' \in N_H[w]$ такие, что $N_G[w'] \cap C = \emptyset$. Действительно, если такие вершины существуют, то после выбора полицейскими начальной позиции, грабитель занимает вершину, не смежную ни одной из занятых полицейскими вершин (вершину w). Если после какого-то хода один из полицейских оказался в смежной w вершине, то грабитель переходит в вершину, не смежную ни одной из занятых противником вершин (вершину w') и т.д.

Пусть $l \leq n-1$ и $C = \{v_1, \ldots, v_l\}$ — какое-то подмножество множества вершин графа G. Выберем произвольную вершину $w \in V(H) \setminus C$. Такая вершина всегда найдётся, поскольку $d(H) = n$. Если $N_G(w) \cap C = \emptyset$, то положим $w = w'$. В противном случае, пусть

$$\{v_1, \ldots, v_k\} = N_G(w) \cap C.$$

В H вершина w смежна по крайней мере n вершинам и как минимум $n-k$ из этих n вершин не принадлежат множеству C. Обозначим множество таких вершин через

$$W = \{w_1, \ldots, w_{n-k}\} \triangleq N_H(w) \setminus C.$$

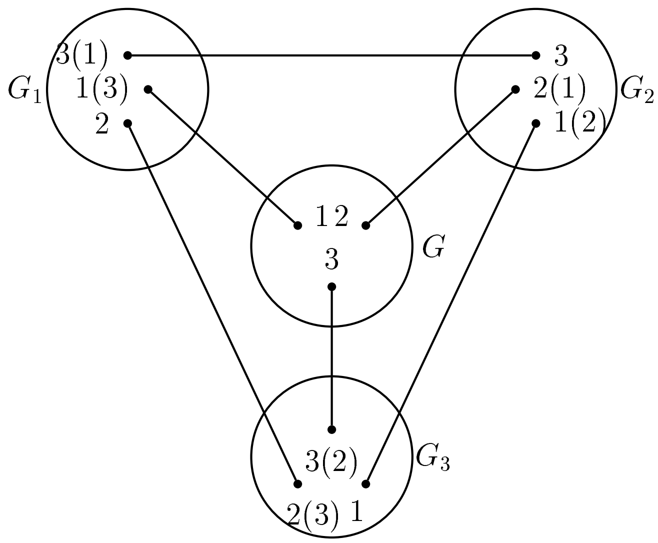


Рис. 4.3: Построение $(n+1)$-регулярного графа и его раскраска

По условию в G нет 3-, 4-циклов, а потому у любых двух вершин $a, b \in N_G(w)$ имеется только один общий усоседф — вершина w, т.е. $N_G(a) \cap N_G(b) = \{w\}$. Таким образом, при $i \leq k$ вершины v_i не смежны вершинам из W. Для $i > k$ вершина v_i может быть смежна не более, чем одной вершине из W.

Учитывая то, что $l \leq n-1$ и $|W| = n-k$ мы заключаем, что одна из вершин $w' \in W$ не имеет смежных вершин в C. $\square$

Следующая теорема утверждает, что для всякого n существуют графы, удовлетворяющие условию теоремы 4.5 и со сцеплением $\geq n$.

Теорема 4.6 *(Aigner-Fromme 84) Для всякого n существует граф без 3- и 4-циклов со степенями всех вершин равными n (n-регулярный граф). Таким образом, для всякого натурального n найдется граф G у которого $cop(G) \geq n$.*

Доказательство. Докажем теорему при помощи индукции по n. Для $n = 1$ и $n = 2$ полный граф K_2 и цикл с пятью вершинами C_5 удовлетворяют условию теоремы. Кроме того, C_5 можно раскрасить в три цвета.

Предположим, что нам удалось построить n-регулярный граф G без порожденных циклов длины ≤ 4 и который можно раскрасить в три цвета. Возьмем три копии G_1, G_2, G_3 графа G и раскрасим каждый из четырех графов одинаковым (изоморфным) образом в три цвета. Построим $(n+1)$-регулярный граф по схеме, приведенной на рис. 4.3. Смысл этой схемы таков: если, к примеру, вершина графа G_1 окрашена цветом 3, то мы соединяем ее с соответствующей (изоморфной) вершиной графа G_2. После того, как все изоморфные вершины соединены согласно схеме (к каждой вершине добавилось ровно по одному инцидентному ребру), поменяем цвета следующим образом: 3 с 1 в G_1, 2 с 1 в G_2 и 3 с 2 в G_3. Построеный граф является $(n+1)$-регулярным, 3-раскрашиваемым и в нём нет порожденных циклов длины 3 и 4. □

Следующая теорема дополняет результаты теоремы 4.6.

Теорема 4.7 *(Aigner-Fromme 84) Пусть максимальная степень графа G $\Delta(G) \leq 3$. Если для всякой пары смежных ребер графа G найдется цикл длины ≤ 5, содержащий эти два ребра, то $cop(G) \leq 3$.*

Доказательство. Предположим, что после некоторого хода игрока-полицейского полицейские C_1, C_2 и C_3 располагаются в вершинах c_1, c_2 и c_3, а грабитель — в вершине r. Выберем пути P_i, $i \in \{1, \dots, 3\}$, из c_i в r такие, что все инцидентные вершине r ребра содержатся в этих путях (существование таких путей вытекает из условий теоремы) и сумма длин этих путей $l = l_1 + l_2 + l_3$ минимальна (l_i — длина пути P_i). Отметим, что эти пути могут иметь общие ребра. Покажем, что на любой ход грабителя у полицейского есть ответный ход из позиции (c_1, c_2, c_3) в позицию (c_1', c_2', c_3'), для которой $l' < l$. Поэтому после конечного числа шагов суммарная длина l будет меньше трех и грабитель окажется пойманным.

Предположим, что вершина r смежна трем вершинам a_1, a_2 и a_3 (в случае $deg(r) \leq 2$ рассуждения такие же). Будем считать, что $P_i = (c_i, \dots, a_i, r)$, $i \in \{1, \dots, 3\}$. Если грабитель остается на месте, то каждый полицейский переходит по своему пути по направлению к r и тогда $l' \leq l - 3 < l$. Пусть грабитель переходит в вершину a_1. Если $l_1 = 1$ (в частности, $deg(a_1) = 1$), то полицейский C_1 ловит его в a_1. Если степень вершины a_1 равна двум и $l_1 \geq 2$, то опять все полицейские переходят по своим путям в направлении r и

$$l' \leq (l_1 - 2) + l_2 + l_3 = l - 2 < l.$$

Если $deg(a_1) = 3$ и $l_1 \geq 2$, то обозначим через u смежную a_1 вершину, не лежащую на пути P_1. По условию теоремы, путь (r, a_1, u) является участьюф цикла длины ≤ 5. Степень r не превосходит трех, поэтому одно из ребер $(a_2, r), (a_3, r)$ принадлежит этому циклу. Будем считать, что циклу принадлежит ребро (a_2, r). Таким образом, вершинами цикла являются вершины a_2, r, a_1, u. Обозначим пятую вершину цикла (если она есть) через v (см. рис. 4.4). Если полицейские продвинутся в сторону v, то для

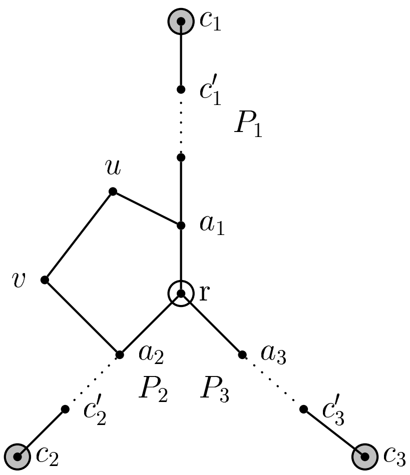


Рис. 4.4: Положение трех полицейских

путей $P_1' = (c_1', \ldots, a_1)$, $P_2' = (c_2', \ldots, a_2, v, u, a_1)$ и $P_3' = (c_3', \ldots, r, a_1)$ имеем

$$l' \leq l_1' + l_2' + l_3' = (l_1 - 2) + (l_2 + 1) + l_3 = l - 1 < l.$$

$\square$

Задача 4.8. Найдите полицейские числа графов правильных многогранников: тетраэдра, октаэдра, куба, икосаэдра и додекаэдра.

$\triangle$ Для тетраэдра достаточно одного полицейского. В графах октаэдра, икосаэдра и куба нет ловушек, поэтому у одного полицейского не существует выигрывающей стратегии. Два же полицейских выигрывают на этих графах после первого хода (у этих графов имеются две вершины, доминирующие все оставшиеся вершины). Полицейское число графа додекаэдра равно трём. Для доказательства этого факта достаточно воспользоваться теоремами 4.5, 4.7. $\triangledown$

Комментарии и ссылки

Задача с одним полицейским была решена в работе [36]. Основой для данной главы послужила статья Айгнера и Фромма [1]. В этой же статье доказывается, что полицейское число планарного графа не превосходит трех. Обобщение этого утверждения приводится в [39] Куиллиотом, доказавшим, что полицейское число графа рода g не превосходит $3 + 2g$. Ряд интересных результатов, касающихся задачи о полицейских и разбойниках на графах получен в работах [2, 3, 4, 5, 14, 13, 18, 22, 29, 46],

Глава 5

Древесные декомпозиции

1 Хордальные графы как графы пересечений

Ранее мы выяснили, что всякий граф интервалов является хордальным. Поскольку графы интервалов являются графами пересечений подпутей некоторого пути, то возникает вполне естественный вопрос характеризации хордальных графов как графов пересечений. Граф G называют *графом пересечений поддеревьев дерева* T, если G является графом пересечений подмножеств вершин $V(T)$, каждое из которых порождает связный подграф дерева T, т.е. поддерево. Мы докажем, что граф является хордальным тогда и только тогда, когда он является графом пересечений поддеревьев некоторого дерева (некоторые поддеревья могут встречаться в этом семействе несколько раз). Такое дерево будем называть *принимающим деревом* (host tree). Если принимающее дерево является путём, то соответствующий граф является графом интервалов (см. задачу 2.4). В дальнейшем, для краткости, мы часто будем отождествлять подмножества вершин принимающего дерева с подграфами, порождаемыми этими вершинами. Так, например, под пересечением поддеревьев $T_1, T_2, \dots, T_m$ принимающего дерева мы будем подразумевать пересечение подмножеств вершин $V(T_1), V(T_2), \dots, V(T_m)$ принимающего дерева T, каждое из которых порождает поддерево.

Напомним важное определение. Говорят, что семейство подмножеств $\{T_i\}_{i\in I}$ множества T обладает *свойством Хелли*, если для всякого индексного подмножества $J \subseteq I$ верно следующее: если $T_i \cap T_j \neq \emptyset$ для всех $i, j \in J$, то $\cap_{j\in J} T_j \neq \emptyset$.

Лемма 5.1. *Пусть $\{T_i\}_{i\in I}$ есть некоторое множество поддеревьев дерева T. Если существуют такие три вершины a, b, c дерева T, что для* *всякого $i \in I$, по крайней мере две из них являются вершинами дерева T_i, т.е. $|V(T_i) \cap \{a,b,c\}| \geq 2$, то $\cap_{j \in I} T_j \neq \emptyset$.*

Доказательство. Для пары вершин u, v дерева T введём в рассмотрение множество P_{uv}, являющееся путём с концами u, v, если эти вершины различны, или вершиной u, если $u = v$. Нетрудно убедиться, что множество $V(P_{ab}) \cap V(P_{bc}) \cap V(P_{ca})$ является вершиной[1] дерева T. Поскольку всякое поддерево T_i, $i \in I$, по условию содержит по крайней мере две вершины из совокупности $\{a,b,c\}$, то оно должно содержать и одно из множеств P_{ab},P_{bc}, P_{ca}. Следовательно,

$$\bigcap_{i \in I} T_i \supseteq P_{ab} \cap P_{bc} \cap P_{ca} \neq \emptyset.$$

□

Лемма 5.2. *Семейство поддеревьев принимающего дерева обладает свойством Хелли.*

Доказательство. Рассмотрим произвольное семейство поддеревьев $\{T_i\}_{i \in I}$ принимающего дерева T. Для доказательства леммы достаточно показать, что для всякого индексного подмножества $J \subseteq I$ верно следующее: если $T_i \cap T_j \neq \emptyset$ для всех $i, j \in J$, то $\cap_{i \in I} T_i \neq \emptyset$.

Это утверждение очевидно, если множество J состоит из одного или двух элементов. Продолжая рассуждать по индукции, предположим, что оно справедливо для любого индексного подмножества с числом элементов $\leq k$. Рассмотрим теперь семейство

$$\{T_i\}_{i \in J}, \qquad |J| = k+1,$$

состоящее из поддеревьев $\{T^1, \ldots, T^{k+1}\}$ и обладающее следующим свойством: $T^i \cap T^j \neq \emptyset$ для всех $i, j \in \{1, \ldots, k+1\}$. Поскольку, в частности, $T^1 \cap T^{k+1} \neq \emptyset$, то существует вершина $a \in T^1 \cap T^{k+1}$. Кроме того, по индукционному предположению множества

$$\bigcap_{i=1}^{k} T^i, \qquad \bigcap_{i=2}^{k+1} T^i$$

непусты, и, следовательно, существуют вершины

$$b \in \bigcap_{i=1}^{k} T^i \quad \text{и} \quad c \in \bigcap_{i=2}^{k+1} T^i.$$

[1]эта вершина является «произведением» вершин a, b, c в некоторой тернарной алгебре.

Нетрудно убедиться, что вершины a, b, c удовлетворяют условию леммы 5.1 для семейства

$$\{T_i\}_{i \in I} = \{T^1, \dots, T^{k+1}\}$$

и, стало быть,

$$\bigcap_{i \in J} T_i \neq \emptyset.$$

□

Теорема 5.3 *(Walter 72, Gavril 74, Buneman 74) Для всякого графа G эквивалентны следующие утверждения:*

1. *G — хордальный граф.*

2. *G — граф пересечений поддеревьев некоторого дерева.*

Доказательство. Импликацию из (1) в (2) докажем, применив индукцию по числу вершин. Для графов с одной и двумя вершинами утверждение очевидно. Предположим, что для всякого хордального графа с числом вершин $< n$ импликация верна. Рассмотрим произвольный хордальный граф G с n вершинами. По теореме 3.3 существует PE-упорядочение $\delta = (v_1, \dots, v_n)$ вершин графа G. Тогда упорядочение $\delta' = (v_2, \dots, v_n)$ является PE-упорядочением вершин графа $G' \stackrel{\triangle}{=} G \setminus \{v_1\}$. По индукционному предположению граф G' является графом пересечений поддеревьев некоторого принимающего дерева T'. Поэтому найдётся семейство поддеревьев $\{T'_2, \dots, T'_n\}$ дерева T' такое, что для всех $i, j \in \{2, \dots, n\}$ $T'_i \cap T'_j \neq \emptyset$ тогда и только тогда, когда вершины v_i, v_j смежны или совпадают.

Вершина v_1 является симплициальной в графе G, поэтому множество вершин $N_G(v_1)$ образует клику K в графе G'. По лемме 5.2 деревья семейства $\{T'_2, \dots, T'_n\}$, соответствующие вершинам из K имеют общую вершину x. Построим дерево T, добавив к дереву T' новую висячую вершину y смежную вершине x. Вершине v_1 поставим в соответствие подграф T_1 дерева T состоящий из единственной вершины y. Для всех вершин $v_i \in K$ определим $T_i \stackrel{\triangle}{=} T'_i \cup \{x, y\}$. Для оставшихся вершин графа G зададим $T_i \stackrel{\triangle}{=} T'_i$. Мы указали семейство поддеревьев $\{T_1, \dots, T_n\}$ дерева T такое, что $T_i \cap T_j \neq \emptyset$, $i, j \in \{1, \dots, n\}$, тогда и только тогда, когда вершины v_i, v_j смежны или совпадают.

(2)$\Rightarrow$ (1). Пусть T — принимающее дерево графа G с наименьшим числом вершин. По определению принимающего дерева, каждой вершине

v графа G можно сопоставить поддерево T_v дерева T таким образом, что для всех $v, u \in V(G)$ $T_u \cap T_v \neq \emptyset$ тогда и только тогда, когда вершины v, u смежны или совпадают. Докажем, что граф G является хордальным, показав существование PE-упорядочения вершин G.

Из минимальности дерева T следует, что для всякой пары смежных в T вершин x, y, в графе G найдётся такая вершина v, что $x \in V(T_v)$, но $y \notin V(T_v)$ (в противном случае, при стягивании ребра (x, y) получается принимающее дерево с меньшим числом вершин). Поэтому для висячей вершины x дерева T можно выбрать вершину $u \in V(G)$ такую, что $V(T_u) = \{x\}$. Для всякой вершины v, смежной (в графе G) вершине u, поддерево T_v обязано пересекаться с T_u, и, стало быть, содержит вершину x. Таким образом, множество

$$\bigcap_{v \in N_G(u)} V(T_v) \neq \emptyset,$$

а значит, вершина u является симплициальной в графе G. Очевидно, что дерево

$$T^1 \triangleq T \setminus \{x\}$$

является принимающим деревом (с наименьшим числом вершин) графа

$$G^1 \triangleq G \setminus \{u\},$$

а потому в G^1 тоже есть симплициальная вершина и т.д.

Мы доказали, что в графе G есть PE-упорядочение, а потому этот граф является хордальным. □

Замечание. У хордального графа может существовать несколько (неизоморфных) принимающих деревьев с минимальным числом вершин. Но как следует из доказательства теоремы 5.3 (импликация из (2) в (1)), всякая висячая вершина минимального принимающего дерева графа G соответствует симплициальной вершине этого графа. Таким образом, принимающее дерево с семейством поддеревьев, графом пересечений которых является граф G, «порождает» PE-упорядочение вершин G. Используя эту же теорему (импликация из (1) в (2)) по заданному PE-упорядочению можно построить принимающее дерево с соответствующим семейством поддеревьев.

Задача 5.4. Докажите, что всякий граф является графом пересечений поддеревьев некоторого графа.

Задача 5.5. Докажите, что если хордальный граф не является графом интервалов, то у него имеется как минимум три симплициальные вершины.

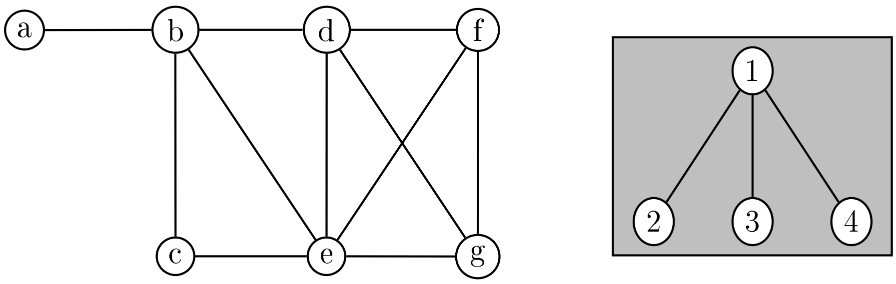


Рис. 5.1: Для графа G (слева) определим древесную декомпозицию $(T, \mathcal{X})$, где дерево T изображено на рисунке справа, $X_1 = \{b, d, e\}$, $X_2 = \{a, b\}$, $X_3 = \{b, c, e\}$ и $X_4 = \{d, e, f, g\}$. Тогда клики, содержащие, к примеру, вершину e порождают в T поддерево с вершинами $1, 3, 4$.

Задача 5.6. Докажите, что у всякого хордального графа есть принимающее дерево, максимальная из степеней вершин которого не превосходит трёх.

Древесной декомпозицией клик графа G называется пара $TD = (T, \mathcal{X})$, состоящая из дерева T, $V(T) = I$, и множеств $\mathcal{X} = (X_i : i \in I)$, являющихся кликами графа G и удовлетворяющих следующим свойствам:

(T1) $\bigcup_{i \in I} X_i = V(G)$;

(T2) Для каждого ребра $\{u, v\} \in E(G)$ найдется вершина i дерева T такая, что $u, v \in X_i$;

(T3) Для любых вершин i, j, k дерева T, если j лежит на пути с концами i и k, то $X_i \cap X_k \subseteq X_j$.

Чтобы не путать вершины древесной декомпозиции с вершинами графа, мы будем называть их *узлами*. В дальнейшем, мы часто будем отождествлять узлы дерева T с набором клик $\mathcal{X}$. Пример древесной декомпозиции клик изображен на рисунке 5.1.

Задача 5.7. Пусть $(T, \mathcal{X})$ — древесная декомпозиция клик графа G. Докажите, что для всякой вершины $v \in V(G)$ множество узлов дерева T, содержащих v, т.е. узлов $\{i \in V(T) : v \in X_i\}$, порождает связный подграф в T.

Древесную декомпозицию клик можно построить не для всякого графа.

Задача 5.8. Докажите, что у циклов длины ≥ 4 не существует древесной декомпозиции клик.

При изучении древесных декомпозиций можно ограничиться рассмотрением *древесными декомпозициями максимальных клик.*

Лемма 5.9. *Если у графа G существует древесная декомпозиция клик, то у G существует и древесная декомпозиция максимальных клик, т.е. такая декомпозиция $(T, \mathcal{X})$ графа G, в которой каждый узел X_i является максимальной кликой.*

Доказательство леммы ватекает из следующих задач.

Задача 5.10. Пусть $(T, \mathcal{X})$ — древесная декомпозиция клик графа G. Докажите, что всякая клика C графа G содержится в одном из узлов дерева T.

Задача 5.11. Пусть $i \neq j$ — узлы дерева T древесной декомпозиции $(T, \mathcal{X})$ клик графа G. Предположим, что для клик, соответствующих этим узлам, выполнено условие $X_i \subseteq X_j$. Определим дерево T' как дерево, получаемое из T, стягиванием первого ребра пути из i в j. Докажите, что пара $(T', \mathcal{X} \setminus X_i)$ тоже является древесной декомпозицией клик графа G.

Древесная декомпозиция клик $(T, \mathcal{X})$, в которой дерево T является путем, называется *путевой декомпозицией клик.*

Используя теорему 2.13 можно доказать следующее утверждение

Задача 5.12. Граф G является графом интервалов тогда и только тогда, когда существует путевая декомпозиция клик графа G.

Задача 5.13. Докажите, что если граф интервалов не является полным, то существует по крайней мере две различные путевые декомпозиции максимальных клик этого графа.

Используя теорему 5.3 докажем следующую лемму о древесных декомпозициях клик.

Лемма 5.14. *Для всякого графа G эквивалентны следующие утверждения:*

1. *G — хордальный граф.*

2. *У графа G существует древесная декомпозиция клик.*

Доказательство. (2)$\Rightarrow$ (1). Пусть $(T, \mathcal{X})$ — древесная декомпозиция клик. Покажем, что дерево T является одновременно и принимающим деревом. Для вершины $v \in V(G)$ множество узлов дерева T, содержащих v, т.е. узлов $\{i \in V(T) \colon v \in X_i\}$, порождает поддерево T_v в T (см. задачу 5.7). По свойству (T2) для любых смежных или совпадающих вершин $u, v \in V(G)$ выполняется условие $T_u \cap T_v \neq \emptyset$. С другой стороны, каждый из узлов дерева T является кликой, поэтому из условия $T_u \cap T_v \neq \emptyset$ следует, что по крайней мере одна их клик содержит одновременно вершины u и v. Итак, мы указали набор поддеревьев $\{T_v\}_{v \in V(G)}$ такой, что для любых вершин u, v вершина $u \in N_G[v]$ тогда и только тогда, когда $T_u \cap T_v \neq \emptyset$.

(1)$\Rightarrow$ (2). Пусть T — принимающее дерево графа G с минимальным числом вершин. Тогда каждой вершине v графа G можно сопоставить поддерево T_v дерева T таким образом, что для всех несовпадающих $v, u \in V(G)$ $T_u \cap T_v \neq \emptyset$ тогда и только тогда, когда вершины v, u смежны. Зададим отображение ϕ, действующее из $V(T)$ в множество клик графа G следующим образом: каждому узлу x отображение ϕ сопоставляет клику в G, состоящую из всех вершин v таких, что $x \in V(T_v)$. Убедимся, что ϕ является взаимно-однозначным отображением из $V(T)$ в множество максимальных клик графа G.

По лемме 5.2 поддеревья, соответствующие максимальной клике в графе G, имеют общий узел в T, поэтому всякая максимальная клика X в G имеет прообраз x в $V(T)$. Если клика $X' \subseteq X$, оказалась сопоставленной некоторому узлу $x' \neq x$, то поддеревья, содержащие узлы x и x', содержат весь путь с концами x', x. Но тогда при стягивании первого ребра этого пути получаем принимающее дерево с числом узлов меньшим, чем в T. Поэтому узлы дерева отображаются только в максимальные клики графа G и у всякой максимальной клики существует только один прообраз.

Проверим, что пара $(T, \mathcal{X})$, где $\mathcal{X} = \{\phi(x) \colon x \in V(T)\}$, удовлетворяет свойствам древесной декомпозиции. Выполнение свойств (T1) и (T2) очевидно, поскольку всякая вершина и всякое ребро содержатся в одной из максимальных клик графа. Для узлов x, y дерева T и вершины $u \in V(G)$, условие $u \in \phi(x) \cap \phi(y)$ означает, что $x, y \in V(T_u)$. Тогда всякий узел z (x, y)-пути в T является узлом дерева T_u. Поэтому $\phi(x) \cap \phi(y) \subseteq \phi(z)$ и условие (T3) тоже выполняется. $\square$

В заключении, несколько задач.

Задача 5.15. Покажите, что семейство максимальных клик хордального графа может и не обладать свойством Хелли.

Задача 5.16. Пусть T — дерево в древесной декомпозиции максимальных клик связного хордального графа G. Докажите, что всякий минимальный разделитель в графе G имеет вид $C_i \cap C_j$, где C_i, C_j — смежные узлы дерева T.

Задача 5.17. Пусть C и C' — максимальные клики хордального графа G, $k \geq 0$ — целое число. Докажите, что из C в C' ведет k непересекающихся путей тогда и только тогда, когда существует последовательность клик $C = C_1, C_2, \ldots, C_q = C'$ такая, что $|C_i \cap C_{i+1}| \geq k$ для всех $i \in \{1, \ldots, q-1\}$.

Задача 5.18. Пусть T — остовное дерево взвешенного связного графа $G = (V, E, w)$. Тогда для всякого ребра $e \in E(G) \setminus E(T)$ в графе $T \cup \{e\}$ имеется единственный цикл $C(e,T)$, называемый *фундаментальным циклом* e и T. Докажите, что T является остовным деревом максимального веса тогда и только тогда, когда для всех ребер $e \in E(G) \setminus E(T)$ и всех ребер f фундаментального цикла $C(e,T)$ выполняется неравенство $w(e) \leq w(f)$.

Задача 5.19. Обозначим через K полный взвешенный граф, вершинами которого являются максимальные клики некоторого хордального графа G, а вес ребра с концами C_1, C_2 равен $|C_1 \cap C_2|$. Пусть T — некоторое остовное дерево графа K. Докажите, что дерево T является остовным деревом максимального веса графа K тогда и только тогда, когда дерево T является деревом в древесной декомпозиции максимальных клик графа G.

2 Древесная ширина

Как мы выяснили в предыдущем параграфе, у всякого хордального граф есть древесная декомпозиция клик. Древесная декомпозиция является обобщением этого понятия для произвольных графов. *Древесной декомпозицией* (tree decomposition) графа G называется пара $TD = (T, \mathcal{X})$, состоящая из дерева T, $V(T) = I$, и множеств $\mathcal{X} = (X_i \colon i \in I)$, являющихся подмножествами $V(G)$ и удовлетворяющих следующим свойствам:

(T1) $\bigcup_{i \in I} X_i = V(G)$;

(T2) Для каждого ребра $\{u, v\} \in E(G)$ найдется вершина i дерева T такая, что $u, v \in X_i$;

(T3) Для всех вершин i, j, k дерева T, если j лежит на пути с концами i и k, то $X_i \cap X_k \subseteq X_j$.

Чтобы не путать вершины древесной декомпозиции с вершинами графа, мы будем называть их *узлами*. Единственное отличие древесной декомпозиции от древесной декомпозиции клик заключается в том, что множества X_i не обязаны являться кликами. Очевидно, что древесная

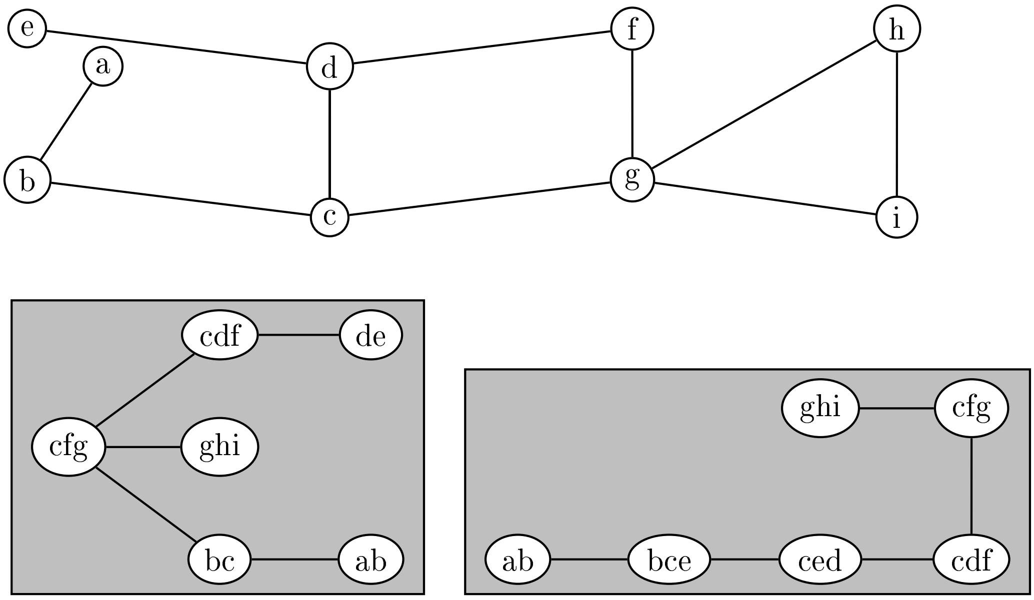


Рис. 5.2: Пример древесной и путевой декомпозиций ширины два.

декомпозиция существует (и не одна) у любого графа. В качестве примера декомпозиции, которая есть у всякого графа G, можно выбрать пару $(T, \mathcal{X})$, где T — дерево с одной вершиной, а множество $\mathcal{X}$ состоит из одного элемента $X_1 = V(G)$.

Выберем следующую меру древесной декомпозиции графа. Шириной древесной декомпозиции графа G является $\max_{i\in I} |X_i| - 1$, а *древесная ширина* (treewidth) графа G, обозначается $tw(G)$, есть минимальная ширина древесной декомпозиции, где минимум берется по всевозможным древесным декомпозициям.

По аналогии с путевой декомпозиции клик определяется путевая декомпозиция. *Путевая декомпозиция* (path decomposition) графа это древесная декомпозиция, в которой дерево T является путём. Шириной путевой декомпозиции $(X_1, \dots, X_r)$ графа G является $\max_{i\in I} |X_i| - 1$, а *путевая ширина* (pathwidth) графа G есть минимальная ширина путевой декомпозиции, где минимум берется по всевозможным путевым декомпозициям.

Пример графа с древесной и путевой шириной равной двум изображен на рисунке 5.2

Мы часто будем пользоваться тем, что свойство (T3) древесной декомпозиции эквивалентно следующему свойству (см. также задачу 5.7):

(T3′) Для всякой вершины $v \in V(G)$ множество узлов, содержащих v, порождает связный подграф дерева T.

Действительно, возьмем два узла i и k древесной декомпозиции некоторого графа G, имеющих общую вершину $v \in V(G)$. По свойству (T3) всякий узел j, лежащий на пути с концами i и k, содержит вершину v. Поэтому, для всякой вершины $v \in V(G)$ множество узлов, содержащих v, порождает связный подграф дерева T древесной декомпозиции TD. Таким образом, (T3) влечет (T3′).

В другую сторону. Рассмотрим дерево T, узлами которого являются подмножества X_i вершин графа G, удовлетворяющие условию (T3′). Если некоторая вершина $v \in V(G)$ принадлежит обоим узлам i и k, то всякий узел j, лежащий на пути с концами i и k, должен содержать вершину v, иначе подграф, порождаемый узлами с v не является связным. Но тогда $X_i \cap X_k \subseteq X_j$. □

Задача 5.20. Докажите, что для всякого хордального графа G верно тождество $tw(G) = \omega(G) - 1$.

Задача 5.21. Докажите, что для всякого графа интервалов G верно тождество $pw(G) = \omega(G) - 1$.

Докажем некоторые простые свойства древесной и путевой ширины.

Лемма 5.22. *Для всякого графа G выполняются следующие свойства:*

1. *Древесная (путевая) ширина всякого подграфа графа G не больше $tw(G)$ ($pw(G)$).*

2. *Древесная (путевая) ширина G есть максимальная из древесных (путевых) ширин его компонент связности.*

Доказательство.

1). Пусть T — древесная декомпозиция графа G, а G' — подграф графа G. В древесную декомпозицию графа G' T превращается при удалении из узлов T вершин $V(G) \setminus V(G')$.

2). Как мы уже доказали, древесная ширина всякой компоненты не больше древесной ширины графа. Построим для каждой компоненты древесную декомпозицию T_i. Добавив необходимое число ребер, превратим объединение деревьев T_i в одно дерево T. Дерево T является древесной декомпозицией графа G, а ширина такой древесной декомпозиции есть максимальная из декомпозиций T_i. Для путевой ширины доказательство аналогично. □

Задача 5.23. Пусть $TD = (T, \mathcal{X})$ — древесная декомпозиция графа G, u и v — произвольные вершины графа G, а $i, j \in I$ такие индексы, что $u \in X_i$, $v \in X_j$. Докажите, что каждый из узлов пути с концами i, j в T содержит по крайней мере по одной вершине от каждого из путей в G с концами u и v. Другими словами, если узел k (i, j)-пути, не содержит u и v, то множество вершин X_k является (u, v)-разделителем в G.

Задача 5.24. Пусть $TD = (T, \mathcal{X})$ — древесная декомпозиция графа G. Докажите, что для всякого связного подграфа G' графа G множество узлов дерева T, содержащие вершины G' порождают связный граф.

Лемма 5.25. *(Лемма о содержащихся кликах). Пусть $TD = (T, \mathcal{X})$ — древесная декомпозиция графа G, $T = (I, F)$ и $\mathcal{X} = \{X_i \colon i \in I\}$. Для всякой клики C графа G найдется такой индекс $i \in I$, что $C \subseteq X_i$.*

Доказательство. Для вершины $v \in V(G)$ договоримся обозначать через T_v подграф T, порождаемый узлами, содержащими v. По свойству (T3′) T_v является поддеревом дерева T. Выберем произвольные вершины u, v клики C. По свойству (T2) найдется узел, содержащий одновременно эти две вершины. Таким образом, любые два дерева семейства $\{T_v\}_{v \in C}$ имеют общие узлы. Но семейство поддеревьев обладает свойством Хелли (лемма 5.2), а потому $\cap_{v \in C}\{T_v\} \neq \emptyset$. Следовательно, в дереве T есть узел, содержащий все вершины клики C. □

3 Триангуляции

Триангуляцией графа G мы будем называть хордальный граф H такой, что $V(H) = V(G)$ и $E(G) \subseteq E(H)$. Название «триангуляция» возникло от еще одного названия хордальных графов — триангулированные графы. Логичнее было бы называть триангуляции «хордализациями», но тут мы вынуждены придерживаться уже установившихся традиций.

Напомним, что кликовым числом $\omega(G)$ графа G называется размер наибольшей клики в G. Обозначим через $ct(G)$ минимум кликового числа графа H, где минимум берется по всем триангуляциям H графа G.

Связь триангуляций с древесными декомпозициями выявляет следующая лемма.

Лемма 5.26. *Пусть $TD = (T, \mathcal{X})$ древесная декомпозиция графа G, $V(T) = I$ и $\mathcal{X} = \{X_i \colon i \in I\}$. Определим надграф H графа G следующим образом:*

1. $V(H) = V(G)$

2. $\{u, v\} \in E(H)$ *тогда и только тогда, когда найдется индекс* $i \in I$ *такой, что* $u, v \in X_i$.

Тогда H *является триангуляцией графа* G.

Доказательство. Для доказательства леммы достаточно показать, что H является хордальным. Поскольку каждое из множеств X_i является кликой в H, то древесная декомпозиция TD является древесной декомпозицией клик графа H. Тогда по лемме 5.14 H является хордальным. □

Теорема 5.27 *Для всякого графа* G *верно равенство* $tw(G) = ct(G) - 1$.

Доказательство. Неравенство $tw(G) \geq ct(G) - 1$ вытекает из доказанной леммы. Действительно, пусть $TD = (T, \mathcal{X})$ — древесная декомпозиция графа G. Рассмотрим граф H, удовлетворяющий условиям леммы. Тогда H является триангуляцией G. По лемме о содержащихся кликах (лемма 5.25) наибольшая клика графа H содержится в одном из узлов дерева T, поэтому

$$ct(G) \leq \omega(H) \leq \max_{i \in I} |X_i|.$$

Поскольку последнее неравенство верно для всякой древесной декомпозиции графа G, то $ct(G) \leq tw(G) + 1$.

Докажем $tw(G) \leq ct(G) - 1$. Пусть H — триангуляция графа G. По лемме 5.14 существует древесная декомпозиция максимальных клик $(T, \mathcal{X})$ графа H. Но, как нетрудно заметить, эта же декомпозиция является древесной декомпозицией графа G. Поэтому для всякой триангуляции H существует древесная декомпозиция $(T, \mathcal{X})$ такая, что

$$tw(G) \leq \max_{X_i \in \mathcal{X}} |X_i| - 1 = \omega(H) - 1.$$

□

Задача 5.28. Докажите, что для графа G с n вершинами и m ребрами имеет место неравенство $tw(G) \geq \frac{2n-1-\sqrt{(2n-1)^2-8m}}{2}$.

Задача 5.29. Докажите, что для графа G с n вершинами $tw(G) = n - 1$ тогда и только тогда, когда G — полный граф.

Для графа G с n вершинами определим

$$\varrho(G) = \begin{cases} n - 1, & \text{если } G \text{ полный граф,} \\ min_{(u,v) \notin E(G)}\{max\{deg(u), deg(v)\}\}, & \text{в противном случае.} \end{cases}$$

Задача 5.30. Докажите, что для всякого графа G $tw(G) \geq \varrho(G)$.

Задача 5.31. Докажите, что для всякого графа G $tw(G) = n-1 \Leftrightarrow \varrho(G) = n-1$.

Задача 5.32. Докажите, что для всякого графа G $tw(G) = n-2 \Leftrightarrow \varrho(G) = n-2$.

Поскольку хордальные графы являются (как графы пересечений) естественным обобщением графов интервалов, естественно предполагать, что путевая ширина должна быть связана с графами интервалов.

Лемма 5.33. *Пусть $PD = (X_1, \ldots, X_r)$ — путевая декомпозиция графа G. Зададим граф I следующим образом: $V(I) = V(G)$, $E(I) = \{\{u, v\}\colon$ найдётся такой номер $1 \leq i \leq r$, что $u, v \in X_i\}$. Тогда I является*

1). графом интервалов;

2). триангуляцией графа G.

Доказательство. Каждой вершине u графа I сопоставим замкнутый отрезок $[l_u, r_u]$, где l_u — наименьший, а r_u — наибольший из номеров $i \in \{1, \ldots, r\}$ таких, что $u \in X_i$. Вершины u, v смежны в I тогда и только тогда, когда отрезки $[l_u, r_u]$ и $[l_v, r_v]$ пересекаются. А потому I — граф интервалов. Всякий граф интервалов является хордальным и для доказательства второго пункта леммы нужно показать, что если вершины u, v смежны в G, то они смежны и в графе I. Если вершины u,v смежны в G, то найдется узел путевой декомпозиции, содержащий u и v (свойство (T2) древесных декомпозиций). Тогда пересечение отрезков $[l_u, r_u]$ и $[l_v, r_v]$ непусто, а потому вершины u,v смежны и в I. □

Напомним, что мы определяли $iw(G)$ — *интервальную ширину* (interval-width) графа G, как минимум кликового числа графа I, где минимум берется по всем графам интервалов I, являющихся триангуляциями графа G.

Теорема 5.34 *Для всякого графа G верно равенство $pw(G) = iw(G) - 1$.*

Доказательство. Убедимся в истинности неравенства $pw(G) \geq iw(G) - 1$. Пусть $PD = (X_1, \ldots, X_r)$ — путевая декомпозиция графа G ширины k. Тогда граф I такой, что $V(I) = V(G)$, $E(I) = \{(u, v)\colon$ найдётся такой номер $1 \leq i \leq r$, что$u, v \in X_i\}$, является графом интервалов (предыдущая лемма) и триангуляцией графа G. Кликовое число графа I равняется $k + 1$.

Докажем, что $pw(G) \leq iw(G) - 1$. Пусть I — граф интервалов с кликовым числом k, содержащий в качестве подграфа граф G. По теореме 2.13

на стр. 16 максимальные (по включению) клики графа G' можно перенумеровать так, что для всякой вершины $v \in V$ и любых двух клик X_i и X_j, $i < j$, содержащих v, всякая клика X_k, $i < k < j$, тоже содержит v. Тогда $PD = (X_1, \ldots, X_r)$ является путевой декомпозицией графа G, а ширина этой путевой декомпозиции равна k. $\square$

Из доказанной теоремы и теоремы 2.17 получаем следующее тождество для вершинно-поискового числа.

Следствие 5.35. *Для всякого графа G $pw(G) = ns(G) - 1$.*

Иногда теоретико-игровая интерпретация путевой ширины бывает удобна при доказательтве свойств путевых декомпозиций.

Приведем пример. Будем называть *двудольным разделением* графа G разделение $V(G) = V_1 \cup V_2$ вершин графа G такое, что $|V_i| \geq 2$ $(i = 1, 2)$ и ребра графа G, соединяющие вершины из V_1 и V_2 порождают полный двудольный граф.

Задача 5.36. Пусть $V(G) = V_1 \cup V_2$ — двудольное разделение графа G, а $A_i \subseteq V_i$, $i = 1, 2$, — вершины соответствующего полного двудольного графа. Тогда во всякой монотонной программе поиска есть шаг, на котором все вершины A_1 или A_2 заняты преследователями. В частности, $ns(G) \geq \min\{|A_1|, |A_2|\} + 1$.

Задача 5.37. Покажите, что для полного двудольного графа G с долями V_1, V_2 $pw(G) = \min\{|V_1|, |V_2|\}$.

Соединением графов G_1, G_2, $V(G_1) \cap V(G_2) = \emptyset$, называется граф $G_1 \times G_2$, множество вершин

$$V(G_1 \times G_2) = V(G_1) \cup V(G_2)$$

и множеством ребер

$$E(G_1 \times G_2) = E(G_1) \cup E(G_2) \cup \{(u, v) : u \in V(G_1), v \in V(G_2)\}.$$

Нестрого говоря, граф $G_1 \times G_2$ получается из G_1 и G_2 добавлением всевозможных ребер с концами в разных графах.

Задача 5.38. Докажите, что

$$\begin{aligned} pw(G_1 \times G_2) &= \min\{pw(G_1) + |V(G_2)|, pw(G_2) + |V(G_1)|\}, \\ tw(G_1 \times G_2) &= \min\{tw(G_1) + |V(G_2)|, tw(G_2) + |V(G_1)|\}. \end{aligned}$$

Задача 5.39. Докажите, что для всякого графа G с n вершинами $tw(G) = n-2$ $\Leftrightarrow pw(G) = n - 2$.

В § 2 мы изучали свойства важного класса хордальных графов k-деревьев. Граф G называется *частичным k-деревом* (partial k-tree), если он является подграфом некоторого k-дерева и не является подграфом никакого $k-1$-дерева.

Задача 5.40. Докажите, что граф G является частичным k-деревом тогда и только тогда, когда $tw(G) = k$.

Пример 5.41. Динамическое программирование на графах с небольшой древесной шириной. Предположим, что мы хотим найти наибольшее независимое множество вершин (т.е. множество попарно несмежных вершин) графа G древесная ширина которого не превосходит некоторого числа k.

Предположим также, что нам известна древесная декомпозиция $(T, \mathcal{X})$ этого графа. Не умаляя общности можно считать, что степень каждого узла дерева T не превосходит трех. (Доказательство того, что у графа с древесной шириной k всегда существует такая декомпозиция ширины k является легким упражнением, см. также задачу 5.6). Выберем произвольный узел r дерева T в качестве корня. Тогда для каждого узла $i \in V(T)$ можно ввести множество узлов-потомков. Узел j называется *потомком* узла i, если i лежит на (r, j)-пути в дереве T (как ни странно, но i тоже удобно считать своим потомком). Каждый узел имеет не более двух смежных ему потомков, называемых его *сыновьями.*

Обозначим через Y_i множество вершин графа G, состоящее из вершин узла $i \in V(T)$ и его потомков. То есть, более формально,

$$Y_i = \bigcup \{X_k \colon k \in V(T) \text{ и } k \text{ — потомок } i\}.$$

Для нахождения наибольшего независимого множества можно воспользоваться общим методом, называемым *динамическим программированием.* Мы будем для каждого i, начиная с листьев дерева T, вычислять независимые множества графа $G[Y_i]$ — порождаемого множеством вершин Y_i. При «обработке» i-го узла будет использоваться только уже полученная информация о независимых множествах в графах, «порождаемых» сыновьями узла i и множеством X_i. Пусть j, k — сыновья узла i. Возможность применения динамического программирования основана на свойстве (T3) древесных декомпозиций. Пусть I_j и I_k некоторые независимые множества в графах $G[Y_j]$ и $G[Y_k]$ соответственно. Независимые множества в $G[Y_i]$ можно построить, перебирая допустимые (т.е. независимые) объединения независимых множеств в $G[Y_j]$ и $G[Y_k]$ и добавляя вершины из $X_i \setminus (Y_j \cup Y_k) = X_i \setminus (X_j \cup X_k)$. Из свойства (T3) следует, что для проверки того, что множество вершин $I_j \cup I_k$ будет независимо, достаточно проверить независимость множества $(I_j \cup I_k) \cap X_i$. Поэтому при конструировании независимого множества в $G[Y_i]$ можно перебирать только вершины из X_i.

Итак, для подмножества вершин $Z \subseteq X_i$, $i \in V(T)$, определим число $is_i(Z)$ как наибольшую мощность независимого в графе $G[Y_i]$ множества вершин $W \subseteq Y_i$ такого, что $W \cap X_i = Z$ (если такого множества W не существует, то мы полагаем $is_i(Z) = -\infty$). Отметим, что размер наибольшего независимого множества в графе G равен $\max_{Z \subset X_r} is_r(Z)$. Если узел i является листом (висячей вершиной отличной от корня), мы вычисляем для всех $2^{|X_i|}$ подмножеств Z множества X_i значения is_i по формуле

$$is_i(Z) = \begin{cases} |Z|, & \text{если } Z \text{ независимое множество} \\ -\infty, & \text{в противном случае.} \end{cases}$$

Для каждого подмножества $Z \subseteq X_i$ внутреннего узла i с сыновьями j и k положим $is_i(Z) = -\infty$, если Z не является независимым в G. Если же Z независимо, то определим

$$is_i(Z) = \max\{is_j(Z_1) + is_k(Z_2) + |Z \cap (X_i \setminus (X_j \cup X_k)| - |Z \cap X_j \cap X_k|\},$$

где максимум берется по всевозможным $Z_1 \subseteq X_j$ и $Z_2 \subseteq X_k$ таким, что $Z \cap X_j = Z_1$ и $Z \cap X_k = Z_2$. (Число $|Z \cap X_j \cap X_k|$ вычитается, поскольку элементы множества $Z \cap X_j \cap X_k$ при суммировании $is_j(Z_1) + is_k(Z_2)$ считаются два раза.) Вычислив все is_r мы можем легко «двигаясь обратно» найти наибольшее независимое множество.

Если $tw(G) = k$, то на обработку каждого внутреннего узла мы затрачиваем «время» $\mathcal{O}(2^{3(k+1)})$ и длительность работы алгоритма $\mathcal{O}(2^{3(k+1)}|V(G)|)$ (при помощи некоторых ухищрений время работы алгоритма можно уменьшить).

Таким образом, на графах, с ограниченой древесной шириной независимое множество можно найти за полиномиальное время (в общем же случае задача NP-полна). Подобные идеи срабатывают и для обширного класса NP-трудных и даже для некоторых PSPACE-полных задач.

□

4 Поиск видимого убегающего

Рассмотрим задачу поиска, очень похожую на задачу вершинного поиска. Единственным отличием от вершинного поиска является то, что преследователи видят убегающего. Преследователи переставляются из вершины в вершину, а убегающий не может покидать граф, зато может передвигаться сколь угодно быстро. Конечно же условие видимости убегающего дает сильное преимущество преследователем. Так, например, на дереве видимый убегающий ловится двумя преследователями следующим образом. Один преследователь встает в невисячую вершину v дерева. Поскольку убегающий не может пробежать через вершину v, он находится ровно в одной из компонент графа $G \setminus v$, называемой загрязненной. Тогда второй преследователь встает в ближайшую к v вершину загрязненной компоненты (поскольку преследователи видят убегающего, они знают какая компонента загрязнена). Действуя таким образом, в конце концов преследователи «зажмут» убегающего в угол.

Дадим формальную постановку задачи. Для подмножества вершин X графа G компоненту связности графа, порождаемого вершинами $V(G) \setminus X$, будем называть X-компонентой.[2] Назовём *позицией* k преследователей упорядоченную пару (X, Y), где $X, Y \subseteq V(G)$, $|X| \le k$, и Y является X-компонентой (X интерпретируется как множество занятых преследователями вершин, а Y — множество загрязненных вершин). Полагается, что первоначальная позиция $(\emptyset, V(G))$. *Позиционной стратегией преследователей* называется отображение

$$\mathcal{F}\colon 2^{V(G)} \to 2^{V(G)}$$

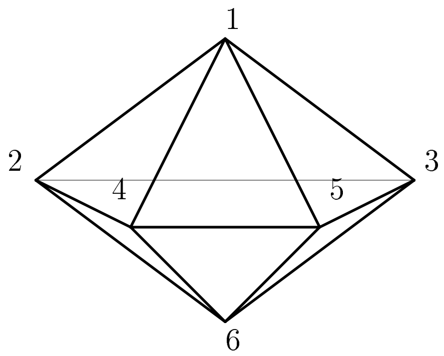
(напомним, что 2^X обозначает множество подмножеств X), такое, что либо $\mathcal{S}(X) \subseteq X$ (часть преследователей снимается с вершин), либо $\mathcal{S}(X) \supseteq X$ (часть преследователей занимает новые вершины). *Позиционной стратегией убегающего* называется отображение $\mathcal{E}$, сопоставляющее каждой тройке $(X, \mathcal{S}(X), Y)$, где Y есть некоторая X-компонента, $\mathcal{S}(X)$-компоненту имеющую непустое пересечение с Y. Если же такой $\mathcal{S}(X)$-компоненты нет, то полагаем $\mathcal{E}(X, \mathcal{S}(X), Y) = \emptyset$. Стратегия преследователей называется выигрывающей, если она приводит к позиции (X, Y) такой, что $\mathcal{E}(X, \mathcal{S}(X), Y) = \emptyset$. Наименьшее число k при котором k преследователей обладают выигрывающей стратегией на графе G, называется t-поисковым числом G и обозначается через $ts(G)$.

Задача 5.42. Докажите, что для всякого дерева T $ts(T) \le 2$.

Лемма 5.43. *Для всякого графа G $ts(G) \le tw(G) + 1$.*

Доказательство. Пусть $TD = (T, \mathcal{X})$ — древесная декомпозиция графа G ширины k. Выигрывающая стратегия преследователей легко строится

[2] Thomas и Seymour называли такие компоненты X-flap



$H_1 = \{1, 5\}$
$H_2 = \{1, 3\}$
$H_3 = \{1, 2\}$
$H_4 = \{2\}$
$H_5 = \{4\}$
$H_6 = \{6\}$
$H_7 = \{3, 5\}$

Рис. 5.3: Экран толщины ≥ 5 в графе октаэдра

обобщением случая поиска двумя преследователями на дереве. Пусть $tw(G) = k$.

Возьмем узел v дерева T и поставим $|X_v|$ преследователей на соответствующие вершины (вершины множества X_v) в G. Мы можем это сделать, поскольку $|X_v| \leq k + 1$. Убегающий находится в одной из X_v-компонент графа G, назовем ее Y, и перебежать в другую компоненту не может. Отметим, что по свойству (T3) (см. также задачу 5.23) вершины из Y могут принадлежать узлам только одной из v-комонент дерева. Пусть T_1 такая компонента. Выберем узел v_1 дерева T_1 смежный узлу v (если такого узла нет, то убегающий, очевидно, уже пойман). Удалим преследователей с вершин $X_v \setminus X_{v_1}$. Повторного загрязнения при таком удалении по свойству (T3) не произойдёт. Далее поставим преследователей в еще не занятые вершины $X_{v_1} \setminus X_v$. В результате, все вершины X_{v_1} окажутся занятыми преследователями. Число узлов в поддереве, соответствующему загрязненной компоненте, уменьшилось. Поэтому «передвигаясь» описанным образом по этому поддереву, преследователи в конце концов поймают убегающего. $\square$

Множества вершин H_1, H_2 графа G *касаются*, если $H_1 \cap H_2 \neq \emptyset$ или существует ребро $e \in E(G)$ с концами в H_1 и H_2. *Экраном* (screen) S в графе G называется набор $H_1, \dots, H_r$ непустых попарно касающихся подмножеств вершин графа, каждое из которых порождает связный подграф. Говорят, что *толщина* экрана $S \geq k$ тогда и только тогда, когда для всякого подмножества $X \subseteq V(G)$, $|X| < k$, найдется подмножество $H_i \in S$ не имеющее с X общих вершин. Отметим, что множества X не обязаны порождать связный подграфы, а объединение множеств H_i не обязано совпадать с V (см. пример экрана на рисунке 5.3).

Оказывается, что задача о нахождении наибольшего экрана является «двойственной» к задаче о нахождении древесной ширины.

Теорема 5.44 *(Thomas и Seymour, 1993). Следующие утверждения эквивалентны*

1. *Древесная ширина графа $G \geq k$;*

2. *В G есть экран толщины $k+1$.*

Доказательство. Мы докажем только лёгкую часть теоремы — импликацию из 2) в 1). Доказательство обратной импликации гораздо сложнее и мы его не приводим.

Пусть $H_1, \dots, H_r$ — экран толщины $\geq k+1$ в графе G. Очевидно, что набор множеств $H_1, \dots, H_r$ является экраном толщины $\geq k+1$ и во всяком надграфе графа G. Выберем триангуляцию G^* (хордальный надграф) графа G такую, что кликовое число этого графа удовлетворяет тождеству

$$\omega(G^*) = tw(G) + 1.$$

По теореме 5.27 такая триангуляция существует.

Рассмотрим некоторую древесную декомпозицию клик $(T, \mathcal{X})$ графа G^*. Каждое из множеств H_i, $i \in \{1, \dots, r\}$, является связным и содержащие вершины из H_i клики порождают в T поддерево T_i. Множества H_i попарно пересекаются или касаются, а так как всякое ребро графа G^* содержится в одной из клик, то для всех $i, j \in \{1, \dots, r\}$

$$T_i \cap T_j \neq \emptyset.$$

По свойству Хелли (см. лемму 5.2) у деревьев T_i имеется общий узел x. Узлу x дерева T соответствует клика K_x графа G^*. Поскольку для всякого множества H_i одна из его вершин содержится в K_x, то

$$tw(G) + 1 = \omega(G^*) \geq |K_x| \geq k+1.$$

□

Импликация из 2) в 1) в теореме 5.44 также следует и из лемм 5.43, 5.45

Лемма 5.45. *Если в G есть экран толщины $k+1$, то $ts(G) \geq k+1$.*

Идея доказательство леммы напоминает идею, использовавшуюся при доказательстве теоремы 4.5 о связи исцепления и полицейского числа. В этой теореме доказывалось, что у грабителя всегда есть возможность перебежать из одной вершины в другую. У нас еще будет возможность вернуться к сцеплениям и экранам при изучении инерционного поиска, а сейчас приступим к доказательству.

Доказательство. Пусть $H_1, \ldots, H_r$ — экран толщины $\geq k+1$ в графе G. Допустим, что лемма не верна и k преследователей могут поймать убегающего. Пусть на некотором шаге преследователи занимают множество вершин X. Поскольку $|X| \leq k$, то найдется элемент экрана H_i на вершинах которого нет преследователей. Предположим, что убегающий находится в одной из вершин множества H_i (первоначально загрязнены все вершины и предположение выполняется). Тогда при занятии преследователями на следующем ходу множества вершин Y, убегающий имеет возможность перебежать в вершину элемент экрана H_j (элементы экрана попарно касаются или пересекаются) такого, что $H_j \cap Y = \emptyset$.

Таким образом, k преследователей недостаточно для поимки убегающего. □

Из теоремы 5.44 и лемм 5.43, 5.45 вытекает

Следствие 5.46. *Для всякого графа G $tw(G) = ts(G) - 1$.*

Комментарии и ссылки

Глава 6

Упорядочения вершин и ребер

Линейным упорядочением вершин (или просто *упорядочением*) графа G (linear ordering) называется последовательность вершин $\sigma = (v_1, \dots, v_n)$, $n = |V(G)|$. Часто мы будем интерпретировать упорядочение как взаимно-однозначное отображение

$$L\colon V(G) \longrightarrow \{1, \dots, |V(G)|\}.$$

Линейное упорядочение можно трактовать как укладку вершин графа на целочисленные точки прямой. Отсюда возникло еще одно название упорядочение вершин — *линейная укладка*. Проблемы нахождения оптимального по некоторому критерию упорядочения вершин графа возникают в ряде практических приложений, например при проектировании СБИС. В этой главе мы обсудим некоторые критерии оптимальности и различные связи укладок с задачами поиска.

1 Поиск инерционного убегающего

Мы уже выяснили (теорема 3.3 на стр. 26), что граф является хордальным тогда и только тогда, когда для его вершин существует PE-упорядочение. Пусть $(v_1, \dots, v_n)$ является PE-упорядочением вершин графа G. Тогда для всякого $i \in \{1, \dots, n\}$ вершина v_i является симплициальной вершиной в графе G_i, порожденном вершинами $(v_i, \dots, v_n)$. Следовательно, кликовое число (размер максимальной клики) хордального графа G равняется наибольшей из степеней вершин $v_i + 1$ в графах G_i, $i \in \{1, \dots, n\}$.

Для произвольного графа G определим операцию *исключения вершины* $v \in V(G)$. Результатом этой операции является граф $G * v$, получаемый из G сперва преобразованием всех смежных v вершин в клику, а потом удалением v. *kPE-упорядочением* называется такое упорядочение вершин $(v_1, \ldots, v_n)$, что для всякого $i \in \{0, \ldots, n-1\}$ степень вершины v_i в графе $G_i \stackrel{\triangle}{=} G * v_1 * \cdots * v_{i-1}$ не превосходит k.

Теорема 6.1 *В графе G существует kPE-упорядочение тогда и только тогда, когда древесная ширина графа G не больше k.*

Доказательство. Если в графе G существует kPE-упорядочение $\sigma = (v_1, \ldots, v_n)$, то для графа G_C, получаемого из графа G добавлением всевозможных ребер графов G_i, $i \in \{0, \ldots, n-1\}$, σ является kPE-упорядочением. Таким образом, G_C является триангуляцией графа G, а кликовое число графа G_C не превосходит $k+1$. Но мы уже доказали (теорема 5.27), что древесная ширина графа плюс один не превосходит размера максимальной клики его триангуляции. Поэтому $tw(G) \leq k$.

В другую сторону. Если $tw(G) \leq k$, то по теореме 5.27 найдется триангуляция G_C графа G, с кликовым числом $k+1$. Тогда PE-упорядочение графа G_C является kPE-упорядочением для G. □

Как мы уже доказали (следствие **??**), наименьшее число преследователей, необходимое для поимки видимого убегающего совпадает с древесной шириной плюс один.

Следствие 6.2. *В графе G существует kPE-упорядочение тогда и только тогда, когда $ts(G) \leq k+1$.*

С древесной шириной связана еще одна модификация вершинного поиска — инерционный поиск.

В задаче поиска инерционного убегающего действия преследователей определяются так же, как и в вершинном поиске. Каждым ходом программы поиска преследователь может быть поставлен или удален с вершины. Отличие заключается в поведении убегающего. Убегающий, действия которого заключаются в переходе из вершины в вершину, может начать движение только в тот момент, когда на занимаемую им вершину ставится преследователь.

Более формально. *Программой поиска* Π на графе G называется последовательность
$$(Z_0, Z_1, \ldots, Z_n),$$
подмножеств $V(G)$ такая, что $Z_0 = \emptyset$ и для всех $i \in \{0, \ldots, n-1\}$ $Z_i \subseteq Z_{i+1}$ или $Z_{i+1} \subseteq Z_i$ (множество Z_i есть множество вершин, занимаемых преследователями на i-ом шаге поиска). В дальнейшем для простоты мы будем считать (очевидно, что это не меняет сути задачи), что на каждом шаге ставится не более одного преследователя, т.е.

$|Z_{i+1} \setminus Z_i| \le 1$. На каждым шаге с номером i программы поиска задается множество очищенных вершин A_i. Множество A_0 полагается пустым. Если $Z_i \supseteq Z_{i+1}$ (на $i+1$-ом шаге часть преследователей снимается с графа), то $A_{i+1} = A_i$, т.е. при удалении части преследователей с вершин графа повторного загрязнения не происходит. Пусть на $i+1$-ом шаге преследователь ставится на вершину v. Если $v \in A_i$, то $A_{i+1} = A_i$. Если $v \notin A_i$, то определим A как множество вершин $u \in A_i$ таких, что всякий (v, u)-путь содержит вершину из Z_i, т.е. $A \subseteq A_i$ это множество вершин, до которых убегающий не может добраться из v. Тогда $A_{i+1} = A \cup \{v\}$. Программа называется *выигрывающей*, если $A_n = V(G)$.

Как и ранее, требуется найти наименьшее из чисел $\max_{0 \le i \le n} |Z_i|$ (число преследователей на i-ом шаге), где минимум берется по всевозможным выигрывающим программам (наименьшее число преследователей достаточное для очистки графа). Будем обозначать это число через $is(G)$. Программа поиска инерционного убегающего $(Z_0, Z_1, \dots, Z_n)$ называется монотонной, если для всякого $i \in \{1, \dots, n\}$ $A_{i-1} \setminus A_i = \emptyset$, т.е. вершины повторно не загрязняются. Наименьшее число преследователей, достаточное для монотонной очистки графа обозначим через $mis(G)$.

Мы докажем, что наименьшее число преследователей, необходимое для поимки инерционного убегающего равно древесной ширине графа плюс один. Сперва мы убедимся, что необходимое число преследователей в монотонной программе равно древесной ширине графа плюс один.

Для упорядочения $\delta = \{v_1, \dots, v_n\}$ вершин графа G определим *опору* вершины v_i, как множество вершин v_j, $j > i$, для каждой из которых найдётся путь P с концами v_i, v_j, все внутренние вершины которого (если они есть) предшествуют v_i в δ.

Лемма 6.3. *Для всякого исключающего упорядочения $\delta = \{v_1, \dots, v_n\}$ опора вершины v_i, $1 \le i \le n$, равна степени вершины v_i в графе $G_i \triangleq G * v_1 * \dots * v_{i-1}$.*

Доказательство. Пусть v_j принадлежит опоре вершины v_i в графе G. При исключении вершины u, предшедствующей v_i в δ, v_j принадлежит опоре вершины v_i и в графе $G * u$. Таким образом, в графе G_i вершины v_i и v_j смежны. В другую сторону, если вершины v_i и v_j, $i < j$, смежны в G_i, то в графе G_{i-1} v_j и v_i являются концами пути, который может иметь только одну внутреннюю вершину — v_{i-1}. Повторяя такое рассуждение от i до 1, мы приходим к тому, что v_j принадлежит опоре вершины v_i в графе G. $\square$

Лемма 6.4. *Для всякого графа G $tw(G) + 1 = mis(G)$.*

Доказательство. Докажем сперва неравенство $tw(G) + 1 \geq mis(G)$. Пусть $tw(G) = k$. По теореме 6.1 существует kPE- упорядочение $\delta = \{v_1, \ldots, v_n\}$ вершин графа G. Построим программу инерционного поиска следующим образом. Сперва один преследователь встает в v_n. Тогда убегающий (если он стоял в v_n) может перебежать в любую вершину, отличную от v_n. Предположим, что все вершины $\{v_{i+1}, \ldots, v_n\}$ уже посещались преследователями и убегающий находится в одной из вершин $\{v_1, \ldots, v_i\}$. Уберем всех преследователей с графа (убегающий при этом неподвижен), после чего сперва займем преследователями все вершины опоры к вершине v_i, а потом поставим преследователя на v_i. На вершинах опоры убегающего не было, а потому начать движение убегающий может лишь когда преследователь ставится на v_i. Но перебежать из вершины v_i в вершину v_j, $j > i$, убегающий не может, поскольку всякий путь из v_i в v_j содержит вершину из опоры вершины v_i. Таким образом, после постановки преследователя на v_i убегающий может находиться лишь на вершинах $\{v_1, \ldots, v_{i-1}\}$. Определяемая таким образом программа преследователей является монотонной, а по лемме 6.3 в программе используется не более $k + 1$ реследователя.

Для доказательства неравенства $tw(G) + 1 \leq mis(G)$ предположим, что для графа имеется монотонная программа поиска $k + 1$ преследователя. Всякая монотонная программа задает естественное упорядочение вершин по очередности посещения их преследователями. Рассмотрим обратное к естественному упорядочение $\delta = \{v_1, \ldots, v_n\}$ ($j > i$ тогда и только тогда, когда вершина v_i посещается преследователями впервые только после того, как была занята вершина v_j). Покажем, что δ является k-исключающим упорядочением. Действительно, если для некоторого i степень вершины v_i в графе G_i превосходит k, то согласно лемме 6.3 опора вершины v_i тоже больше k. Тогда при постановке преследователя на вершину v_i по крайней мере одна из вершин опоры к v_i не охраняется, что приводит к повторному загрязнению. □

Для доказательства теоремы о монотонности поиска инерционного убегающего чрезвычайно полезным оказывается понятие экрана (см. стр. 68).

Лемма 6.5. *Если в G есть экран толщины $\geq k + 1$, то $is(G) > k$.*

Доказательство. Рассмотрим произвольную программу поиска k преследователями инерционного убегающего на графе G. Поскольку число преследователей меньше толщины экрана, на каждом шаге поиска имеется элемент экрана на вершинах которого нет преследователей. Все элементы экрана попарно касаются, а поскольку первоначально все вершины загрязнены, то нетрудно убедиться, что на каждом шаге поиска имеется загрязненный элемент экрана. Поэтому программа не является выигрывающей. □

Как следствие из лемм 6.4 и 6.5 получаем

Теорема 6.6 *(Dendris, Kirousis, Thilikos 97) Для всякого графа G $tw(G)+1 = is(G)$.*

Доказательство. Пусть древесная ширина графа G равна k. Тогда по лемме 6.4 $mis(G) \leq k+1$. По лемме 5.44 в G есть экран толщины $\geq k+1$, тогда лемма 6.5 влечет $is(G) \geq k+1$. Поскольку неравенство $is(G) \leq mis(G)$ очевидно выполняется, то $k+1 \leq is(G) \leq mis(G) \leq k+1$. □

Лемму 6.5 легко усилить, показав, что если в графе есть экран ширины $\geq k+1$, то k преследователей не смогут поймать *в*идимого инерционного убегающего. Поэтому при поиске инерционного убегающего результат поиска не зависит, как это ни удивительно, от располагаемой преследователями информацией.

Другой интересной задачей поиска инерционного убегающего является случай, когда расстояние, на которое может отбежать убегающий, ограничено. Рассмотрим вариант задачи инерционного поиска, в котором убегающий может перебегать лишь в смежную вершину. Мы выясним, что в этом случае поисковое число совпадает с другим, уже встречавшимся нам инвариантом, шириной графа.

Итак, пусть

$$(Z_0, Z_1, \dots, Z_n$$

— программа поиска. Множества очищенных вершин задаются следующим образом. Если $Z_i \supseteq Z_{i+1}$ (на $i+1$-ом шаге часть преследователей снимается с графа), то $A_{i+1} = A_i$. Если на $i+1$-ом шаге преследователь ставится на вершину v и $v \in A_i$, то $A_{i+1} = A_i$. Если $v \notin A_i$, то определим $A = N_G[v] \cap Z_{i+1}$ (множество смежных v вершин занятых преследователями вместе с вершиной v). Тогда $A_{i+1} = (A_i \setminus N_G[v]) \cup A$, т.е. загрязняются лишь смежные v вершины незанятые преследователями.

Как и ранее, требуется найти наименьшее из чисел $\max_{0 \leq i \leq n} |Z_i|$ (число преследователей на i-ом шаге), где минимум берется по всевозможным выигрывающим программам (наименьшее число преследователей достаточное для очистки графа). Будем обозначать это число через $is_1(G)$. Наименьшее число преследователей, достаточное для монотонной очистки графа в задаче поиска инерционного убегающего со скоростью 1, обозначим через $mis(G)$.

При изучении «жадных» раскрасок графа нами было введено понятие ширины графа G (см. §4) $w(G)$.

Лемма 6.7. *Для всякого графа G $mis_1(G) = w(G) + 1$.*

Доказательство. Пусть $\delta = \{v_1, \dots, v_n\}$ — упорядочение вершин графа G наименьшей ширины k. Построим монотонную программу $k+1$ преследователей следующим образом. Первоначально игрок ставится на вершину v_1. Предположим теперь, что в вершинах $v_1, \dots, v_i$ убегающего нет. Перед первой постановкой преследователя в вершину v_{i+1} удалим из вершин графа всех преследователей. Далее, займем все предшествующие (при упорядочении δ) и смежные вершине v_{i+1} вершины. По определению ширины таких вершин не больше k и по индукционному предположению в этих вершинах нет убегающего. Тогда при постановке преследователя на вершину v_{i+1} убегаюший, если он находился в этой вершине, может перебежать только в вершину, идущую после v_{i+1}. Таким образом, $mis_1(G) \leq w(G) + 1$.

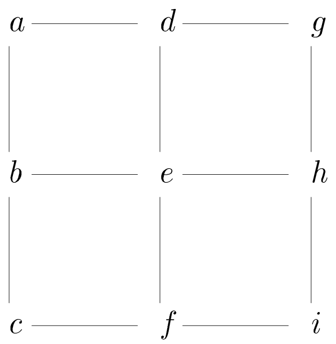
Для доказательства неравенства $mis_1(G) \geq w(G)+1$ отметим, что монотонная программа поиска $k+1$ преследователя естественным образом задает упорядочение (по порядку посещения вершин преследователями) $\delta = \{v_1, \dots, v_n\}$ вершин графа G. Ширина графа, соответствующая упорядочению δ не превосходит k. Действительно, если ширина некоторой вершины $v \in V(G)$ при упорядочении δ больше k, то при постановке преследователя на v произойдет повторное загрязнение. □

Для выявления связи задачи нахождения ширины графа с немонотонной задачей инерционного поиска, удобно рассматривать ее в паре с «двойственной» задачей. Обозначим через $d(G)$ наименьшую из степеней вершин графа G. Напомним, что сцеплением графа G называется $linkage(G) = \max\{d(H)\colon\ H$ является подграфом $G\}$. Мы уже доказывали (лемма 3.32), что для всякого графа его сцепление совпадает с шириной.

Лемма 6.8. *Если в графе G есть подграф H, наименьшая из степеней вершин которого равна k, то $is_1(G) > k$. Следовательно, $is_1(G) > linkage(G)$.*

Доказательство. Если убегающий стоит в некоторой вершине $v \in V(H)$, то всякая попытка поймать его k преследователями будет безуспешна. Когда один из преследователей становится в вершину v, то как минимум одна из смежных вершине v в графе H вершина u свободна от преследователей и убегающий переходит в u. □

Теорема 6.9 *(Dendris, Kirousis и Thilikos, 97) Для всякого графа G $mis_1(G) = is_1(G) = w(G) + 1$.*



$$L = \{a, b, c, f, e, d, g, h\}$$

$$\partial L_1 = \{a\},\ \partial L_2 = \{a, b\},\ \partial L_3 = \{a, b, c\},$$

$$\partial L_4 = \{a, b, f\},\ \partial L_5 = \{a, e, f\},\ \partial L_6 = \{d, e, f\},$$

$$\partial L_7 = \{e, f, g\},\ \partial L_8 = \{f, h\},\ \partial L_9 = \emptyset.$$

Рис. 6.1: Решетка 3×3 и множества ∂L_i.

Доказательство. Неравенство $mis_1(G) \geq is_1(G)$ очевидно. Неравенство $is_1(G) \geq linkage(G) + 1 = w(G) + 1$ следует из лемм 6.8 и 3.32, а неравенство $w(G) + 1 \geq mis_1(G)$ из леммы 6.7. Таким образом,

$$w(G) + 1 \geq mis_1(G) \geq is_1(G) \geq linkage(G) + 1 = w(G) + 1.$$

□

Задача 6.10. Докажите, что $pw(G) \geq w(G)$.

Обобщая постановку задачи инерционного поиска можно определить числа $is_k(G)$. Но для всех конечных $k > 1$ вопрос о монотонности поиска, т.е. существуют ли $k > 1$ и граф G такие, что $is_k(G) < mis_k(G)$, остается открытым.

2 Величина вершинного разделения

Для упорядочения $\sigma = (v_1, \ldots, v_n)$ вершин графа G определим величину $\partial\sigma_i$ как множество вершин смежных одновременно вершинам с номерами $\leq i$ и вершинам с номерами большими i. Всякий путь из вершины с номером, не превосходящим i, в вершину с номером большим i содержит по крайней мере одну вершину множества $\partial\sigma_i$. Вершины множества $\partial\sigma_i$ удобно представлять себе как «граничные точки» множества вершин $\{v_1, \ldots, v_i\}$. На рисунке 6.1 изображены множества $\partial\sigma_i$ упорядочения σ вершин решетки размера 3×3.

Величиной вершинного разделения для упорядочения σ, обозначается $vs(G, \sigma)$, называется наибольшее из чисел $|\partial\sigma_i|$, $i \in \{1, \ldots, n\}$. *Величина* *вершинного разделения графа* G (vertex separation number) определяется как

$$vs(G) = \min\{vs(G,\sigma) \colon\ \sigma \text{ — упорядочение вершин графа } G\}.$$

Следующая теорема выявляет связь вершинного поиска и величины вершинного разделения. Ничего удивительного в этом нет, поскольку программа монотонного поиска задает упорядочение вершин по мере посещения их преследователями и на каждом ходе, преследователи «разделяют» очищенные вершины от загрязненных.

Теорема 6.11 *Для всякого графа* G $vs(G) = ns(G) - 1$.

Доказательство. Пусть $\sigma = (v_1, \ldots, v_n)$ оптимальное для величины вершинного разделения упорядочение вершин графа G. Опишем программу вершинного поиска $k = vs(G) + 1$ преследователей. Преследователи ставятся на вершины G в порядке, задаваемом σ. Преследователь удаляется из вершины сразу после того, как на всех соседних с ней вершинах побывали преследователи. Отметим, что повторного загрязнения при таких действиях игроков не происходит. На каждом шаге программы на вершинах графа находится не более, чем $\max_{i\in\{1,\ldots,n-1\}} \partial\sigma_i \cup \{v_{i+1}\} = k$ преследователей.

Для доказательства неравенства $ns(G) \geq vs(G)+1$ рассмотрим монотонную программу $k + 1 = ns(G)$ игроков. Порядок посещения вершин графа преследователями естественным образом задает упорядочение σ. Если найдется номер $1 \leq i \leq n-1$ такой, что $|\partial\sigma_i| > k+1$, то при постановке преследователя на вершину v_{i+1} происходит повторное загрязнение — что противоречит монотонности программы. □

3 Реберный и смешанный поиск

Рассмотрим две модификации задачи вершинного поиска невидимого убегающего. Эти задачи отличаются лишь динамикой преследователей и способом поимки убегающего (или очисткой ребер). В первом варианте поиска, называемом *реберным поиском*, преследователи обладают дополнительной особенностью передвигаться по ребрам. Преследователь ловит убегающего, если «сталкивается» с ним при прохождении ребра. В такой постановке задача поиска более естественна и в первых работах по поиску [37, 38] изучалась именно эта задача. Во второй модификации, называемой задачей *смешанного поиска*, преследователи тоже могут переходить по ребрам, а ловят убегающего либо «столкнувшись» с ним при прохождении ребра, либо оккупировав концы ребра, на котором находится убегающий. Таким образом, эта задача является «смесью» задач вершинного и реберного поиска.

В задачах реберного и смешанного поиска действия преследователей описываются последовательностью ходов. Эту последовательность мы будем называть *программой поиска*. Каждым ходом преследователь может быть

- поставлен на вершину,
- удален с вершины,
- передвинут из вершины в вершину вдоль ребра.

В терминах очистки эти задачи определяются следующим образом. Первоначально все ребра графа предполагаются загрязненными. Ребро e с концами $\{x, y\}$ становится очищенным:

В задаче реберного поиска. Если либо один из преследователей стоит в x, а второй идет по e из x в y, либо все инцидентные x ребра, за исключением e, очищены, и преследователь идет по e из x в y.

В задаче смешанного поиска. Если один из преследователей идет по e из x в y и все инцидентные x ребра, за исключением e, очищены, либо если вершины x и y заняты преследователями.

Очищенное ребро e может быть опять загрязнено в момент удаления или передвижения преследователя, если в этот момент существует не содержащий преследователей путь, соединяющий e с каким-то загрязненным ребром. Программа поиска называется *выигрывающей*, если по окончании ее действия все ребра графа очищены. В обеих задачах требуется найти минимальное число преследователей, необходимое для существования выигрывающей программы (для очистки всех ребер графа G). Эти числа называются реберно- и смешанно-поисковыми числами графа G. Будем обозначать их $es(G)$ и $ms(G)$.[1] Если в результате действий преследователей ни одно из уже очищенных ребер не становится вновь загрязненным, то такую программу преследователей мы будем называть *монотонной*.

На рисунке 6.2 показаны способы очистки одномерного остова тетраэдра четырьмя преследователями в задачах вершинного и реберного поиска и тремя преследователями в задаче смешанного поиска. Отметим, что задачи поиска можно рассматривать и на псевдографах.

[1] От английского edge search и mixed search numbers.

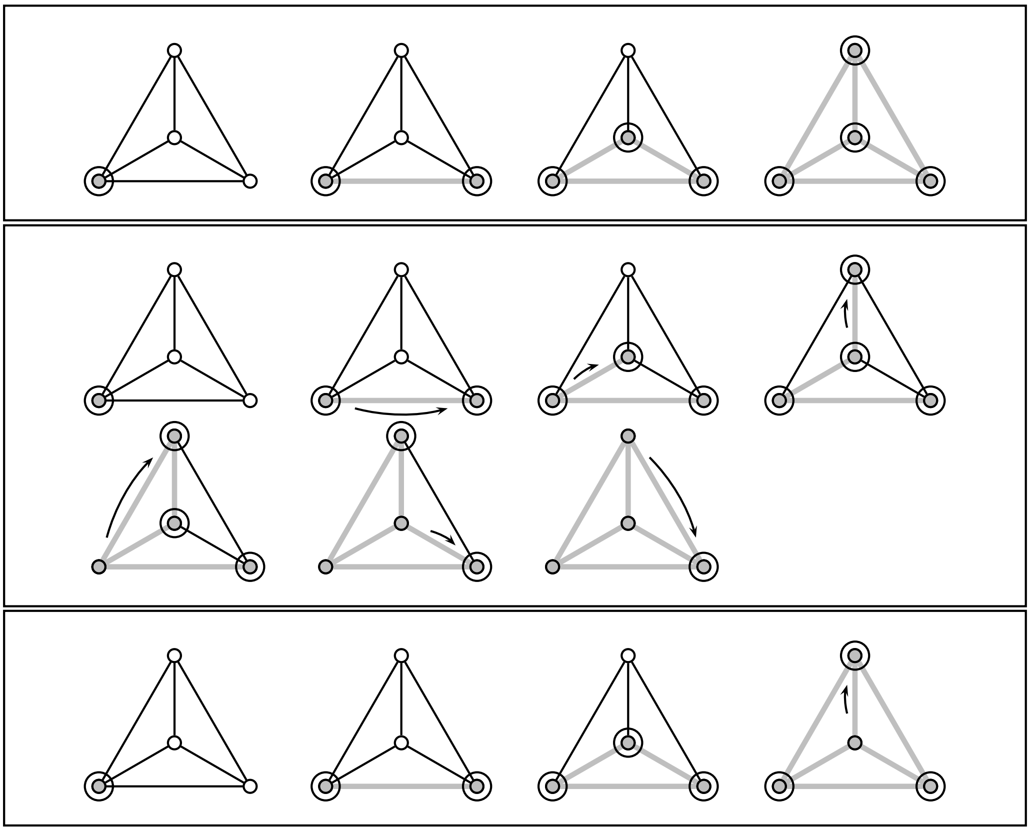


Рис. 6.2: Очистка одномерного остова тетраэдра в задачах (сверху вниз) вершинного, рёберного и смешанного поиска. Серым цветом окрашены очищенные рёбра и вершины. Вершины, занимаемые преследователями, обведены окружностями. Стрелка, нарисованная рядом с ребром, означает, что ребро было очищено прохождением по нему (в направлении стрелки).

Задача 6.12. Пусть граф G^l получен из графа G добавлением нескольких петель. Докажите, что $ms(G) = ms(G^l)$, $ns(G) = ns(G^l)$ и $es(G) \le es(G^l) \le es(G) + 1$.

Задача 6.13. Пусть граф G^m получен из графа G добавлением нескольких кратных рёбер. Докажите, что $ns(G) = ns(G^m)$, $es(G) \le es(G^m) \le es(G) + 1$ и $ms(G) \le ms(G^m) \le ms(G) + 1$.

Задача 6.14. Доказать, что $ms(G) \le \min\{es(G), ns(G)\} \le \max\{es(G), ns(G)\} \le ms(G) + 1$.

Задача 6.15. Для полного графа K_n, $n \ge 1$, найдите числа $ms(K_n)$ и $es(K_n)$.

Изучение вершинного поиска мы начали с доказательства трудного факта о свойствах монотонности выигрывающих программ (теорема 1.4). При помощи некоторых ухищрений, из этого факта можно установить наличие монотонности и для задач реберного и смешанного поиска.

Теорема 6.16 *На всяком графе G, $ms(G) = k$, существует выигрывающая монотонная программа смешанного поиска k преследователей.*

Доказательство. Обозначим через G_m граф, получаемый помещением на каждое ребро вершины степени два (заменой ребра на путь длины два). Вершины $V(G_m) \subset V(G)$ будем называть *старыми* вершинами, а вершины $V(G_m) \setminus V(G)$ — *искусственными*. Мы докажем, что $ns(G_m) = ms(G) + 1$ и покажем как построить монотонную программу смешанного поиска $ns(G_m) - 1$ преследователей на G.

Неравенство $ns(G_m) \le ms(G) + 1$ почти очевидно. Действительно, обладая дополнительным игроком легко указать программу вершинного поиска на G_m «имитирующую» смешанный поиск на G.

Докажем $ns(G_m) \ge ms(G) + 1$. Рассмотрим монотонную программу Π вершинного поиска на G_m. Поскольку программа монотонная, мы считаем, что на каждую из вершин преследователь ставится ровно один раз и всякий преследователь удаляется из вершины как только все инцидентные ей рёбра очищены.

Не умаляя общности, можно полагать, что в этой программе преследователь занимает искусственную вершину только после того, как была занята по крайней мере одна из смежных ей старых вершин. Построим по этой программе программу Π^* смешанного поиска на G, действуя на каждом шаге по следующим правилам:

1. Если в программе Π преследователь занимает старую вершину v, то в программе Π^* эта вершина (если все инцидентные этой вершине рёбра ещё не очищены) занимается преследователем;

2. Всякий преследователь в Π^* удаляется из вершины как только все инцидентные ей рёбра очищены. Заметим, что если в программе Π преследователь снимается со старой вершины v, то в программе Π^* эта вершина (если она еще не освобождена) тоже освобождается.

3. Если в программе Π преследователь ставится на искусственную вершину ребра $\{u, v\}$, то в программе Π^* очищается ребро $\{u, v\}$ (если оно еще не очищено) следующим образом. Будем считать для определённости, что вершина u занята преследователем, а вершина v свободна (одновременно эти вершины не могут быть заняты, иначе ребро было бы уже очищено). Если есть возможность очистить ребро $\{u, v\}$ прохождением по нему не вызвав повторного загрязнения (такое возможно если все рёбра инцидентные u за исключением $\{u, v\}$ уже очищены), то очищаем ребро прохождением из u в v. В противном случае, очищаем это ребро постановкой преследователя в v.

4. Во всех остальных случаях ничего делать не надо.

Очевидно Π^* является монотонной программой поиска. Для завершения доказательства нам осталось убедиться в том, что наибольшее число занятых в этой программе игроков не больше $ns(G_m) - 1$. Во-первых, на каждом шаге программы Π^* игроков задействовано не больше, чем в программе Π. Пусть на некотором шаге программы Π используется максимальное число преследователей $ns(G_m)$. Будем считать, что на этом шаге преследователь занимает вершину v. Поскольку число преследователей на G_m достигло предела, то на этом же шаге должно быть убрано несколько игроков. Мы предполагаем, что в программе поиска всякий преследователь снимается с вершины как только все инцидентные этой вершине рёбра очищены. Поэтому один из преследователей снимается с v или с одной из смежной v вершин. Если хотя бы на одном из путей в G_m, соответствующему некоторому ребру в G находится три преследователя, то на этом шаге в программе Π^* задействовано $\leq ns(G_m) - 1$ игроков. Поэтому если преследователь снимается с искусственной вершины, то на соответствующем пути длины два находилось три преследователя и теорема доказана.

Поэтому мы можем ограничиться случаями, когда

- v является старой вершиной. Тогда преследователь удаляется из v. Это означает, что все инцидентные в G_m вершине v рёбра были загрязнены перед этим ходом и все смежные этой вершине искусственные вершины заняты преследователями. Но тогда на этом же

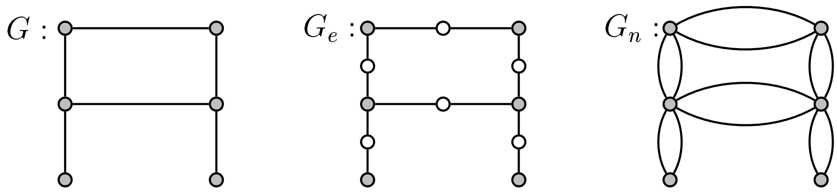


Рис. 6.3: Граф G и графы G_e, G_n.

шаге в программе Π^* уже нет преследователя (он должен был снят сразу после того, как в программе Π была занята последняя смежная v искусственная вершина).

- v является искусственной вершиной. Тогда преследователь снимается со смежной v старой вершине u. При этом вторая смежная v вершина w уже посещалась преследователем. Поэтому в программе Π^* постановке игрока на искусственную вершину соответствует очистка прохождением из w в u.

И в том, и в другом случаях число игроков задействованных на данном шаге $\leq ns(G_m) - 1$ и теорема доказана. □

Обратимся теперь к доказательству монотонности в задаче реберного поиска. Обозначим через G_e граф, получаемый из графа G заменой каждого ребра на путь, состоящий из двух ребер (См. Рис. 6.3). Как нетрудно убедиться, при такой замене реберное поисковое число графа не меняется и $es(G_e) = es(G)$. Неравенство $ms(G) \leq es(G)$ выполняется для всякого графа, а потому $ms(G_e) \leq es(G_e) = es(G)$. Далее порядок наших рассуждений таков: сперва мы докажем, что монотонную программу смешанного поиска на графе G_e с k преследователями можно преобразовать в монотонную программу реберного поиска с тем же числом преследователей (Лемма 6.17 является чуть более сильным утверждением). Далее мы покажем как всякую монотонную программу реберного поиска k преследователей на G_e переделать в монотонную программу реберного поиска k преследователей на G (Лемма 6.18).

Лемма 6.17. *Пусть в графе G всякое ребро имеет конец инцидентный еще ровно одному ребру. Тогда монотонная программа смешанного поиска k преследователей на G может быть преобразована в монотонную программу реберного поиска k преследователей на G.*

Отметим, что граф G_e удовлетворяет условиям леммы.

Доказательство. Рассмотрим произвольную монотонную программу смешанного поиска. Если все ребра в этой программе очищаются прохождением преследователя, то эта программа является также и программой реберного поиска и лемма доказана. Предположим, что на i-ом шаге программы некоторое ребро f очищается не прохождением игрока. Тогда сразу перед шагом i двое преследователей стоят в концах u, v ребра f. Пусть вершина v инцидентна еще ровно одному ребру g.

Мы заменим шаг с номером i тремя шагами. Если ребро g загрязнено, то преследователь убирается из вершины v, ставится в вершину u и очищает ребро f проходом из u в v. Если ребро g очищено, то преследователь очищает ребро f проходом из v в u, убирается из вершины u и ставится в вершину v. Заменяя указанным способом все необходимые шаги, получаем монотонную программу реберного поиска. □

Лемма 6.18. *Всякую монотонную программу реберного поиска k преследователей на G_e можно переделать в монотонную программу реберного поиска k преследователей на G.*

Доказательство. Обозначим через f' и f'' ребра графа G_e на которые было заменено ребро f графа G. Определим граф H как граф получаемый из G_e стягиванием всех ребер f'', $f \in E(G)$. Таким образом H изоморфен G. Каждой вершине v графа H естественным образом можно сопоставить множество C_v вершин графа G_e — прообразов вершины v при стягивании. Программу поиска на G_e преобразуем в программу поиска на H следующим образом: если на i-ом шаге преследователь находится в вершине $u \in V(G_e)$, то в новой программе он находится в вершине $v \in V(H)$, $u \in C_v$. Очевидно, что новая программа является монотонной. □

Из доказанных лемм вытекает следующая теорема о монотонности реберного поиска.

Теорема 6.19 *На всяком графе G, $es(G) = k$, существует выигрывающая монотонная программа реберного поиска k преследователей.*

На самом деле, научившись доказывать монотонность одной задачи поиска, можно доказать монотонность во всех задачах поиска. К примеру, из монотонности смешанного поиска получить монотонность вершинного поиска следующим образом. Обозначим через G_n граф, получаемый из графа G заменой каждого ребра двумя кратными ребрами (см. Рис. 6.3). Читателю рекомендуется самостоятельно доказать следующие утверждения:

Задача 6.20. $ms(G_n) = ns(G)$.

Задача 6.21. Всякую монотонную программу вершинного поиска k преследователей на G_n можно преобразовать в монотонную программу вершинного поиска k преследователей на G.

Задача 6.22. Пусть в графе G всякое ребро имеет кратное ребро. Тогда монотонная программа смешанного поиска k преследователей на G может быть преобразована в монотонную программу вершинного поиска k преследователей на G.

4 Ширина разреза

С упорядочением вершин графа связан еще один интересный параметр. Пусть $L = \{v_1, \dots, v_n\}$ — некоторое упорядочение вершин графа G. *i-м разрезом* будем называть множество ребер

$$\{\{v_k, v_j\} \in E(G) \colon k \le i < j\}$$

Шириной i-го разреза $cw_i(G, L)$ называется число ребер в этом разрезе. Название возникло из-за того, что если «уложить» вершины графа на прямой в порядке L, то перпендикуляр к этой прямой, проходящий между вершинами v_i и v_{i+1}, пересекает $cw_i(G, f)$ ребер. Зададим $cw(G, L) = \max_{i \in \{1,\dots,n\}} cw_i(G, L)$. Тогда *шириной разреза* (cutwidth) графа G называется

$$cw(G) \stackrel{\triangle}{=} \min\{cw(G, L) \colon L \text{ — упорядочение вершин } G\}.$$

На рисунке 6.4 изображено упорядочение вершин графа куба. Ширина разреза данного упорядочения равна пяти.

Задача нахождения упорядочения ширины вершин графа с наименьшей шириной разреза возникает при проектировании СБИС — сверхбольших интегральных схем.[2] Электронные приборы (вершины графа) бывает нужно расположить на краю печатной платы. Плату изготавливают из изоляционного материала, а соединительные провода (ребра графа) наносят печатным способом на поверхность платы. Печатные проводники не изолированы и поэтому не могут пересекаться. Поэтому число плоскостей (толщина схемы), на которых производится печать, пропорционально ширине разреза, соответствующая данной линейной укладке.

[2]VLSI—Very Large Scale Integrated systems

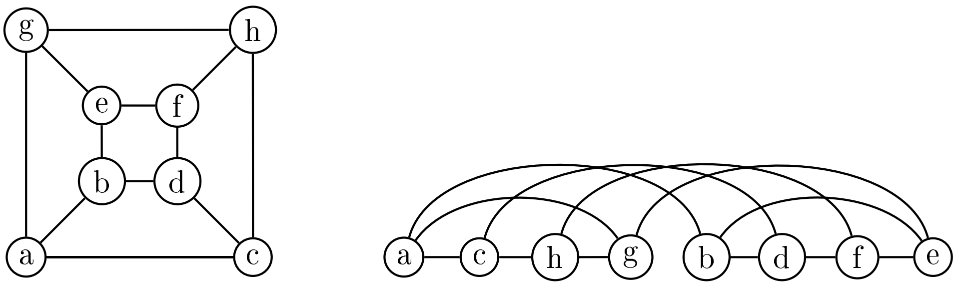


Рис. 6.4: Упорядочение вершин куба с шириной разреза пять

Задача 6.23. Докажите, что для всякого графа G $cw(G) \geq \left\lceil \frac{\Delta(G)}{2} \right\rceil$.

Пусть $L = \{v_1, \ldots, v_n\}$ есть некоторое упорядочение (или линейная укладка) вершин графа G. Для всякого $i \in \{1, \ldots, |V(G)|\}$ всякая вершина множества ∂L_i (подмножества вершин с номерами $\leq i$ и смежных вершинам с номерами $> i$) инцидентна ребру, «проходящему» через i-й разрез. Поэтому

$$ns(G) - 1 = vs(G) \leq cw(G).$$

Аналогичное неравенство справедливо и для реберно-поискового числа.

Лемма 6.24. *Для всякого графа G $es(G) \leq cw(G)$.*

Доказательство. Для доказательства леммы покажем, как по упорядочению $L = \{v_1, \ldots, v_n\}$ построить выигрывающую программу поиска с числом преследователей $\leq k = cw(G, L)$.

Предположим, что для некоторого $i \in \{1, \ldots, |V(G)| - 1\}$ нам удалось построить программу k преследователей такую, что преследователи на некотором шаге стоят в вершинах ∂L_i и все ребра подграфа, порождаемого множеством вершин $\{v_1, \ldots, v_i\}$ на этом шаге очищены. Отметим, что для $i = 1$ существование такой программы очевидно.

Покажем, что тогда такая программа есть и для $i + 1$. Если $|\partial L_i| \leq k - 2$, то утверждение очевидно — нужно одного преследователя поставить в v_{i+1}, после чего вторым преследователем очистить все ребра $\{v_{i+1}, u\}$, где $u \in \partial L_i$. Если $|\partial L_i| = k$, то все вершины множества ∂L_i инцидентны ровно одному ребру из i-го разреза. Тогда для всякой вершины множества $w \in \partial L_i$ смежной v_{i+1} мы переводим преследователя, стоявшего в этой вершине, по ребру $\{w, v_{i+1}\}$ в v_{i+1}. Если $|\partial L_i| = k - 1$, то в этом случае все вершины множества ∂L_i, за исключением, быть может, одной вершины u, инцидентны ровно одному ребру из i-го разреза. Если вершина u существует и смежна v_{i+1}, то мы ставим «имеющегося в запасе» единственного преследователя и проводим его по ребру $\{u, v_{i+1}\}$, после чего для всякой вершины множества $w \in (\partial L_i \setminus u)$ смежной v_{i+1} мы переводим преследователя по ребру $\{w, v_{i+1}\}$ в v_{i+1}. Если же такой вершины u нет, или она есть, но не смежна v_{i+1}, то мы поступаем как в случае $|\partial L_i| = k$, т.е. переводим преследователей в v_{i+1} из вершин, инцидентных этой вершине.

Такие действия не влекут повторного загрязнения, а потому когда закончится переход преследователей в вершину v_{i+1}, очищенными окажутся ребра подграфа, порождаемые вершинами с номерами $\{v_1 \dots, v_i\}$ плюс ребра, ведущие из вершин с номерами $< i+1$ в вершину v_{i+1}. Но все эти ребра в совокупности и являются ребрами подграфа, порождаемого вершинами $\{v_1 \dots, v_{i+1}\}$. □

Оценка, приводимая в лемме 6.24 не является точной. Так, к примеру, реберно-поисковое число любой звезды $K_{1,n}$ равно двум, в то время как $cw(K_{1,n}) = \left\lceil \frac{n}{2} \right\rceil$ (см. задачу 6.23). Оказывается, что ширина разреза равна реберно-поисковому числу для всех графов, максимальная степень вершин которых не превосходит три. Makedon-Sudborough [30] доказали более сильное утверждение (напомним, что $\Delta(G)$ обозначает наибольшую из степеней вершин графа G).

Теорема 6.25 *Для всякого графа G $cw(G) \le \left\lfloor \frac{\Delta(G)}{2} \right\rfloor (es(G) - 1) + 1$.*

Доказательство. Мы хотим по монотонной выигрывающей программе реберного поиска Π с k преследователями «построить» упорядочение вершин L, ширина которого не превосходит $\left\lfloor \frac{\Delta(G)}{2} \right\rfloor (k-1) + 1$. Стандартный прием, уже использованный при доказательстве связи вершинного поиска с упорядочением вершин графа: упорядочить вершины по мере посещения их преследователями в данной ситуации не применим. В этом можно убедиться, рассмотрев поиск двумя игроками на $K_{1,n}$. Мы докажем теорему только для случая, когда в графе G нет вершин степени два. Теорема верна и для общего случая, но присутствие вершин степени два приводит к необходимости рассмотрения ряда технических случаев, «раздувающих» доказательство.

Для каждой вершины $v \in V(G)$ обозначим через $f(v)$ наименьший номер шага в программе Π, на котором как минимум половина инцидентных вершине v ребер очищена. Отметим, что на $f(v)$-м шаге преследователь либо заходит, либо выходит из v. Функция f может принимать одинаковые значения не более чем на двух вершинах. Рассмотрим упорядочение $L = (v_1, \dots, v_n)$, удовлетворяющее следующему условию: если $f(v_i) < f(v_j)$, то $i < j$.

По определению, в i-м разрезе находится $cw_i(G, L)$ ребер. Мы докажем, что для всякого i в i-м разрезе находится не более $\left\lfloor \frac{\Delta(G)}{2} \right\rfloor (k-1)+1$ ребер.

Выберем произвольное i. На $f(v_i)$-м шаге программы Π один преследователь либо заходит, либо выходит из v_i, а остальные преследователи (их не больше $k-1$) стоят в вершинах графа. Для всякого ребра e из i-го разреза возможна одна из трех ситуаций:

- $e \in C_i$, где C_i — множество очищенных к $f(v_i)$-му шагу ребер i-го разреза,
- $e \in N_i$, где N_i — множество загрязненных после $f(v_i)$-го шага ребер i-го разреза,
- $e = e_i$, e_i — ребро, очищаемое на $f(v_i)$-м шаге.

Всякое ребро $e = \{v_l, v_m\}$ из i-го разреза имеет один конец v_l «левее», а один конец «правее» вершины v_i, т.е. $l \leq i < m$.

Предположим, что $e \in C_i$. Поскольку $i < m$, то после $(f(v_i)-1)$-го шага вершине v_m инцидентно как минимум два загрязненных ребра, а потому на $f(v_i)$-м шаге в v_m должен находиться преследователь. С другой стороны, вершина v_m на $f(v_i)$-м шаге инцидентна не более, чем $\left\lfloor \frac{deg(v_m)}{2} \right\rfloor$ очищенным ребрам, следовательно,

$$|C_i| \leq \left\lfloor \frac{\Delta(G)}{2} \right\rfloor k_1, \qquad (*)$$

где k_1 — число вершин, «лежащих» правее v_i в которых на $f(v_i)$-м шаге стоят преследователи.

Предположим, что $e \in N_i$. Тогда после $f(v)$-го шага в вершине u либо должен находиться «стационарный» преследователь (т.е. преследователь, простоявший этот ход в вершине). (Заметим, что если степень вершины u равна двум и $u = v$, то данное утверждение может и не выполняться. Именно из-за этого случая требуется более аккуратная работа с вершинами степени два.)

Каждая стоящая перед v_{i+1} вершина (в порядке, задаваемым L) инцидентна не более, чем $\frac{\Delta}{2}$ ребрам из множества N_i. Таким образом,

$$|N_i| \leq \left\lfloor \frac{\Delta}{2} \right\rfloor k_2, \qquad (**)$$

где k_2 — число вершин с номерами $< i+1$ и занятых на $f(v_i)$-м шаге «стационарными» преследователями. Поскольку $k_1 + k_2 \leq k - 1$ (преследователь, проходящий по ребру e_i, не является стационарным), то используя неравенства (*) и (**), получаем

$$cw_i(G,L) \le |C_i| + |N_i| + 1 \le \left\lfloor \frac{\Delta}{2} \right\rfloor (k-1) + 1$$

($cw_i(G,L) \le |C_i| + |N_i| + 1$ обращается в равенство, если ребро e_i принадлежит разрезу). □

Из доказанной теоремы и предыдущей леммы вытекает следствие.

Следствие 6.26. *Если максимальная степень вершины графа G не превосходит трех, то $es(G) = cw(G)$.*

Интересная связь между шириной разреза и величиной вершинного разделения реберного графа была подмечена Головачем [19]. Доказательство теоремы Головача мы опускаем.

Теорема 6.27 *Для всякого связного и имеющего по крайней мере два ребра графа G верны следующие неравенства:*

$$cw(G) \le vs(L(G)) \le cw(G) + \left\lfloor \frac{\Delta(G)}{2} \right\rfloor - 1$$

С учетом следствия 6.26 и теоремы 6.11 для поисковых чисел графов с максимальной степенью вершин три получаем следующий факт.

Следствие 6.28. *Пусть связный граф G имеет более одного ребра и $\Delta(G) \le 3$. Тогда $cw(G) = vs(L(G))$, и, следовательно, $es(G) = ns(L(G)) - 1$.*

5 Линейная ширина

Для подмножества ребер $X \subseteq E(G)$ графа G мы определяем $\delta(X)$ как множество вершин, одновременно инцидентных ребрам из множеств X и $E(G) \setminus X$. Пусть

$$L = (e_1, \ldots, e_m)$$

— некоторое упорядочение ребер графа G. Для $i \in \{1, \ldots, m\}$ определим множество ребер

$$E[i,L] = \bigcup_{j=1} ie_i.$$

Линейная ширина графа G, соответствующая упорядочению L, определяется как

$$lw(G,L) \stackrel{\triangle}{=} \max_{i \in \{1,\ldots,|E(G)|\}} |\delta(E[i,L])|,$$

а *линейная ширина* (linear width) графа G как

$$lw(G) \stackrel{\triangle}{=} \min\{lw(G,L)\colon L \text{ упорядочение ребер } G\}.$$

Задача 6.29. Докажите, что линейная ширина и величина вершинного разделения графа G связаны следующим соотношением:

$$vs(G) \leq lw(G) \leq vs(G) + 1.$$

Совпадение линейной ширины со смешанно-поисковым числом для графов без вершин степени, доказывается в следующей теорема.

Теорема 6.30 *Для всякого графа G $lw(G) \leq ms(G)$. Если в графе G всякая вершина имеет степень ≥ 2, то $lw(G) = ms(G)$.*

Доказательство. Рассмотрим монотонную программу смешанного поиска k преследователей. Программа поиска задает упорядочение ребер графа по мере их очистки (если одновременно очищается несколько ребер, то упорядочиваем их произвольно). Для порождаемого таким образом упорядочения ребер

$$L = (e_1, \ldots, e_m)$$

выполняется $|\delta(E[i,L])| \leq k$ поскольку на каждом шаге поиска множество очищенных ребер «отделяется» преследователями от загрязнённых. Поэтому $lw(G) \leq ms(G)$.

Предположим, что всякая вершина графа G имеет степень ≥ 2. Для упорядочения ребер графа G

$$L = (e_1, \ldots, e_m)$$

определим $k = lw(G,L)$ и покажем, что на G существует выигрывающая программа смешанного поиска k преследователей.

Предположим, что для некоторого $i \in \{1, \ldots, n\}$ есть такая программа поиска k преследователей, что на некотором шаге этой программы преследователям удалось очистить множество ребер $P_i \supseteq E_i$ и на этом шаге вершины $\delta(E_i)$ заняты преследователями (для $i = 1$ такая программа очевидно имеется). Мы докажем теорему, показав, что преследователи могут занять вершины множества $\delta(E_{i+1})$, очистив при этом ребра E_{i+1}.

Если ребро e_{i+1} уже очищено, то преследователи просто удаляются из вершин $\delta(E_i) \setminus \delta(E_{i+1})$.

Предположим, что ребро e_{i+1} загрязнено. Если ребро e_{i+1} не инцидентно ни одной вершине из $\delta(E_i)$, то неравенства $|\delta(E_{i+1})| \leq k$ мы заключаем, что $|\delta(E_{i+1})| \leq k-2$ и тогда ребро e_{i+1} можно очистить, поставив на его концы двух преследователей. Если e_{i+1} инцидентно некоторой вершине $v \in \delta(E_{i+1})$ (двум вершинам из этого множества загрязненное ребро не может быть инцидентно) и $|\delta(E_{i+1})| \leq k-1$, то при постановке преследователя на еще не занятый конец ребра e_{i+1} это ребро очищается.

Единственный оставшийся неизученным случай когда все преследователи уже стоят на вершинах графа, т.е. $|\delta(E_i)| = k$ и ребро e_{i+1} инцидентно ровно одной вершине множества $\delta(E_i)$. Будем считать, для определенности, что $e_{i+1} = \{u, v\}$ и $v \in \delta(E_i)$. Тогда вершине u не инцидентно ни одного ребра из E_i и так как каждой вершине инцидентно как минимум два ребра, то $u \in \delta(E_{i+1})$. Таким образом, $v \notin \delta(E_{i+1})$, поскольку в противном случае $|\delta(E_{i+1})| = k+1$. Но последнее означает, что вершине v инцидентно единственное загрязненное ребро e_{i+1} и стоящий в этой вершине преследователь очищает e_{i+1} прохождением по нему из v в u.

После очистки ребра e_{i+1} снимаем преследователей с вершин $\delta(E_i) \setminus \delta(E_{i+1})$. Как нетрудно убедиться, повторного загрязнения при описанных действиях преследователей не происходит. В результате, все ребра из E_{i+1} оказались очищенными, а преследователи занимают вершины $\delta(E_{i+1})$. $\square$

Комментарии и ссылки

Глава 7

Поиск на ориентированных графах

1 Пути в орграфах

Хроматическое число $\chi(D)$ ориентированного графа D определяется как хроматическое число неориентрированного графа G, ориентацией которого является орграф D. Обозначим через $\lambda(D)$ число вершин в самом длинном ориентированном пути в орграфе D. Связь параметров $\chi(D)$ и $\lambda(D)$ выявляется в следующей теореме.

Теорема 7.1 (Gallai 68, Roy 67) *Для всякого ориентированного графа D имеет место неравенство*

$$\chi(D) \leq \lambda(D).$$

Доказательство. Пусть хроматическое число D равно k. Докажем, что D содержит путь длины $\geq k-1$.

Выберем в D максимальный (по включению) ациклический подграф A. Тогда число

$$k' \triangleq \lambda(A)$$

не превосходит $\lambda(D)$. Покажем как можно раскрасить D в k' цветов.

Каждой вершине v сопоставим число $c(v)$ равное числу вершин в самом длинном из путей, имеющих начало в v. Отметим, что $V(D) = V(A)$ и отображение

$$c\colon V(D) \to \{1, \ldots, k'\}$$

можно трактовать как раскраску вершин графа D. Убедимся, что данная раскраска является правильной.

Из ацикличности орграфа A следует, что если в A из вершины u в вершину v ведёт путь, то $c(u) > c(v)$. В частности, если (v, y) является дугой A, то $c(u) > c(v)$. Если же дуга $(u, v) \in E(D) \setminus E(A)$, то в A из вершины u в вершину v ведёт путь (в противном случае A не был бы максимальным ациклическим подграфом) и $c(u) > c(v)$. □

Доказанная теорема позволяет выразить хроматическое число неориентированного графа через длины путей в его ориентациях.

Следствие 7.2. *Для всякого графа G*

$$\chi(G) = \min\{\lambda(D) \colon D \text{ — ориентация } G\}.$$

Доказательство. Для всякой ориентации D графа G неравенство $\chi(G) \leq \lambda(D)$ следует из теоремы 7.1. Обратное неравенство выполняется, когда в качестве ориентации выбрана ориентация, приписывающая каждому ребру направление от вершины с меньшим цветом к вершине с большим цветом. Действительно, если в такой ориентации есть путь длины k, то $\chi(G) \geq k$ поскольку все вершины пути должны быть окрашены разными цветами. □

Будем называть *разбиением* графа G множество непересекающихся подграфов, множество вершин которых является разбиением вершин графа G. *Разбиением на пути* называется разбиение графа в котором каждый из подграфом является путём. Наименьшее число путей в разбиении графа G на пути обозначим через $\pi(G)$. Аналогичным образом определяется разбиение орграфа на ориентированные пути.

Напомним, что числом *числом независимости* $\alpha(G)$ графа G называется число вершин в наибольшем независимом множестве (т.е. множестве попарно несмежных вершин) этого графа. По аналогии с хроматическим числом ориентированного графа, число независимости $\alpha(D)$ ориентированного графа D определяется как число независимости неориентрированного графа G, ориентацией которого является орграф D.

Следующая теорема связывает число независимости орграфа D с числом $\pi(D)$ и является «двойственной» к теореме 7.1.

Теорема 7.3 (Gallai-Milgram, 60) *Для всякого ориентированного графа D имеет место неравенство*

$$\alpha(D) \geq \pi(D).$$

Доказательство. Пусть L — разбиение на пути в орграфе D. Введем обозначения. Через $\alpha(L)$ обозначим мощность наибольшего (по числу элементов) независимого множества в D имеющего в каждом пути разбиения L не более одной вершины. Через $begin(L)$ будем обозначать начала, а через $end(L)$ концы путей из L. В принятых обозначениях доказываемая теорема сразу следует из следующей леммы.

Лемма 7.4. *Если для разбиения на пути L орграфа D выполняется неравенство $|L| > \alpha(L)$, то существует разбиение на пути P такое, что*

1. $|P| = |L| - 1$;
2. $begin(P) \subset begin(L)$;
3. $end(P) \subset end(L)$.

Действительно, если разбиение на пути L является оптимальным, то оно не может удовлетворять условиям леммы. Поэтому $|L| \leq \alpha(L)$. А так как неравенство $\alpha(L) \leq \alpha(D)$ выполняется всегда, то

$$\pi(D) = |L| \leq \alpha(D).$$

Доказательство леммы. Докажем лемму используя индукцию по числу вершин n в орграфе. Для $n = 1$ лемма справедлива. Предположим, что лемма справедлива для всех орграфов с числом вершин $\leq n$. Предположим, что в орграфе D, $|V(D)| = n + 1$, имеется разбиение на пути L удовлетворяющее $|L| > \alpha(L)$. Тогда множество $begin(L)$ не может быть независимым множеством и найдётся пара вершин $y, z \in begin(L)$, образующих дугу (y, z). Если вершина y является путём из L (путём, состоящим из одной вершины), то искомое разбиение P получается удалением из L пути y и добавлением к пути, ведущему из z, дуги (y, z). Поэтому будем полагать, что путь с началом в y содержит как мимнимум две вершины.

Обозначим через x вершину, идущую за y в пути из L. Определим $D' \stackrel{\triangle}{=} D \setminus \{y\}$ и L' как ограничение L на D'. Тогда

$$\begin{aligned}
&|L'| = |L| > \alpha(L) \geq \alpha(L'),\\
&begin(L') = begin(L) \setminus \{y\} \cup \{x\} \text{ и}\\
&end(L') = end(L).
\end{aligned}$$

По индукционному предположению для D' существует разбиение на пути P' такое, что

$$\begin{aligned}
&|P'| = |L'| - 1,\\
&begin(P') \subset begin(L') \text{ и}\\
&end(P') \subset end(L').
\end{aligned}$$

Если $x \in begin(P')$, то добавив к пути с началом в x дугу (y, x) получаем искомое разбиение P. Если $x \notin begin(P')$, то должно выполняться условие $z \in begin(P')$ (число путей в P' на единицу меньше числа путей в L'). В этом случае добавим дугу (y, z) к пути из P' с началом в z. В обоих случаях получаем разбиение на пути P, удовлетворяющее всем трём условиям леммы. □

Как и в случае с хроматическим числом, доказанная теорема влечет

Следствие 7.5. *Для всякого графа G*

$$\alpha(G) = \max\{\pi(D)\colon D \text{ — ориентация } G\}.$$

Доказательство. По доказанной теореме неравенство $\alpha(G) \geq \pi(D)$ выполняется для всякой ориентации графа G. Докажем обратное неравенство. Выберем произвольное независимое множество A в графе G. Рассмотрим ориентацию D графа G такую, что всякая вершина $v \in A$ не является началом дуги D. Очевидно, что такая ориентация существует. Тогда для всякого разбиения на пути орграфа D всякая вершина из A содержится ровно в одном пути и $|A| \leq \pi(D)$. □

Следующий известный факт из теории упорядоченных множеств является следствием теоремы 7.3. Напомним, что отношение $\prec$ на множестве X называется *отношением частичного порядка* (partial order), если оно рефлексивно, антисимметрично и транзитивно. Если $\prec$ — отношение порядка на X, то пару $P = (X, \prec)$ называют частично упорядоченным множеством (в англоязычной литературе используется термин poset от partial ordered set). Цепью называется набор попарно сравнимых, а антицепью — попарно несравнимых элементов упорядоченного множества.

Задача 7.6. [Теорема Дилворта] Пусть $P = (X, \prec)$ есть частично упорядоченное множество. Докажите, что наименьшее число цепей в P, на которое можно разбить X, равняется наибольшему числу элементов антицепи в P.

2 Поиск на ориентированных графах

Задачи поиска можно определять и на ориентированных графах. Рассмотрим следующий ориентированный вариант задачи реберного поиска на ациклическом графе D[1], с множеством вершин $V(D)$ и дуг $A(D)$. Как и в поиске на неориентированном графе на движение убегающего не накладывается никаких ограничений. Преследователи же, напротив,

[1]от английского digraph, орграф

могут передвигаться в направлении дуг. Преследователи не имеют возможности покидать граф (в отличии от неориентированного случая). Первоначально преследователи занимают все вершины с нулевой степенью захода (т.е вершины, которые не являются концами ни одной из дуг). Будем обозначать множество таких вершин через $V_0(D)$.

Дадим точную постановку ориентированного *дугового поиска* (название дуговой поиск для того, чтобы подчеркнуть, что поиск ведется на орграфе). Мы ограничимся рассмотрением только связных ациклических графов (т.е. связнвх орграфов, не содержащих ориентированых циклов). В задаче дугового поиска действия преследователей описываются последовательностью ходов. Эту последовательность мы будем называть *программой поиска.* Первоначально преследователи занимают вершины $V_0(D)$. Каждым ходом преследователь может быть передвинут из вершины в смежную вершину по направлению дуги.

Первоначально все дуги графа предполагаются загрязненными. Дуга $e = (x, y)$ очищается либо если либо один из преследователей стоит в x, а второй идет по e из x в y, либо все инцидентные x дуги, за исключением e, очищены, и преследователь идет по e из x в y. Очищенная дуга e может быть опять загрязнена в момент передвижения преследователя, если в этот момент существует не содержащая преследователей цепь, соединяющая e с одной из загрязненнх дуг. Программа поиска называется *выигрывающей*, если по окончании ее действия все дуги графа очищены. Как всегда, требуется найти минимальное число преследователей, необходимое для существования выигрывающей программы (очистки всех дуг графа D). Будем обозначать это число через $as(D)$.

Напомним, что в предыдущем параграфе мы определяли параметр $\pi(D)$, как наименьшее число путей в разбиении орграфа D. Введем новый параметр $\gamma(D)$, равный наибольшему числу вершин, попарно не содержащихся ни в одном из путей графа D. Нам понадобится следующее следствие теоремы 7.3.

Лемма 7.7. *Для всякого ациклического графа D, $\gamma(D) = \pi(D)$.*

Доказательство. Неравенство $\gamma(D) \leq \pi(D)$ очевидно.

Для доказательства обратного неравенства воспользуемся теоремой 7.3. Рассмотрим транзитивное замыкание T орграфа D, т.е. орграф с тем же самым множеством вершин, что и D, причем $(u, v) \in A(H)$ тогда и только тогда, когда существует (u, v)-путь в D. Тогда размер наибольшего независимого множества $\alpha(H)$ в H совпадает с $\gamma(H)$. С другой стороны, $\alpha(H) \leq \alpha(G)$ и $\gamma(H) = \gamma(G)$. По теореме 7.3 $\alpha(H) \geq \pi(H)$ и, окончательно, $\alpha(G) \geq \alpha(H) \geq \gamma(H) = \gamma(G)$. □

Для ациклического орграфа D его реберный орграф $L(D)$ определяется как граф, вершинами которого являются дуги $A(D)$ и пары $((a,b),(c,d))$, $(a,b),(c,d) \in A(D)$ образуют дугу в $L(D)$ тогда и только тогда, когда $c = d$.

Теорема 7.8 *Для всякого ациклического графа D без изолированных вершин $as(D) = \gamma(L(D))$.*

Доказательство. Если набор дуг M образует множество, элементы которого попарно не содержатся ни в одном из путей орграфа $L(D)$, то никакие две дуги из этого множества не могут быть пройдены одним и тем же преследователем. Поэтому $as(D) \geq \gamma(L(D))$.

Пусть $p = \gamma(L(D))$. Тогда по лемме 7.7 существует разбиение $L(D)$ на p путей $L_1, L_2, \ldots, L_p$. Этим путям естественным образом соответствуют пути $D_1, D_2, \ldots, D_p$ в D. Заметим, что эти пути задают разбиение дуг орграфа D, причем начало и конец всякой дуги из $A(D)$ являются последовательными вершинами в некотором пути D_j.

Для вершины v определим $l(v)$ как наибольшую из длин путей в D ведущих в v. Стратегия поиска p преследователей такова: первоначально преследователи $\{S_1, S_2, \ldots, S_p\}$, расставляются по началам путей $\{D_1, D_2, \ldots, D_p\}$. Пусть на i-ом шаге программы, $i \geq 1$, преследователи занимают множество вершин Z_i. Обозначим через Z_i' подмножество вершин Z_i, каждая из которых является началом некоторой дуги графа D. Выберем из множества вершин Z_i' вершину v с наименьшим числом $l(v)$. По определению, существует дуга $e = (v, u)$. Дуга e должна принадлежать одному из путей D_j. Тогда в v должен находиться преследователь S_j (если это не так, то вершина, занимаемая игроком S_j имеет меньший номер $l(v)$). На $i+1$ шаге преследователь S_j передвигается по дуге e из v в u. В результате действия такой программы каждый преследователь пройдет по своему пути от начала до конца и все дуги графа будут пройдены. Покажем, что при таком поведении преследователей не произойдет загрязнения уже очищенных дуг, и следовательно, по окончании программы все дуги будут очищены.

Предположим, что до i-го шага повторного загрязнения не произошло (для $i = 1$ утверждение очевидно выполняется). Пусть на i-ом шаге программы преследователь S_j передвигается из вершины a с минимальным «весом» $l(a)$ в некоторую вершину b. Если в результате этого хода какие-то дуги повторно загрязняются, то по определению поиска, в a не остается преследователей и имеется инцидентная вершине a и загрязненная на $i-1$ шаге дуга e. По предположению, дугу e преследователи еще не проходили. Поэтому если $e = (c, a)$, то $l(a) > l(c)$ и преследователь должен идти из c. Если же $e = (a, c)$, то $c \neq b$ и дуга (a, c) содержится в некотором пути D_k, $k \neq j$. Эта дуга тоже еще не проходилась преследователями, следовательно в a после i-го шага должен находиться еще и преследователь S_k. Противоречие, завершающее доказательство теоремы, достигнуто. □

Комментарии и ссылки

Первый параграф основан на обзоре [9].

При написании параграфа о поиске на ориентированных графах была использовалась статья [35]. В этой же работе обсуждается вершинный вариант поиска на орграфах.

Поиск на вертолете

3 Ширина ленты

Первоначально задача о ширине ленты возникла в теории матриц. Пусть $A = \{a_{ij}\}_{i,j=1}^{n}$ — некоторая разреженная матрица, то есть матрица, большинство элементов которой является нулями. Подобные матрицы возникают в практике достаточно часто. При выполнении таких операций как умножение или обращение, приходится много раз перемножать нулевые элементы. При хранении в компьютере матрицы мы должны держать в памяти n^2 записей, что явно не экономно, когда большинство из элементов матрицы нулевые. Естественно для более эффективной работы и экономного хранения разреженных матриц «сконцетрироваться» на ненулевых элементах. Следить за ненулевыми элементами матрицы легче, если все они находятся в «ленте», т.е. в небольшом числе диагоналей над и под главной диагональю. Подобное расположение ненулевых элементов может значительно ускорить вычисление произведения матриц, нахождение обратной матрицы и решение систем уравнений.

Поставим следующую задачу: с помощью одинаковых перестановок столбцов и строк (если меняются местами i-й и j-й столбцы, то меняются местами и i-я и j-я строки) привести симметрическую матрицу A к такому виду, чтобы все ненулевые элементы находились на диагоналях матрицы, ближайших к главной, т.е. ищется матрица A, получаемая из исходной одновременной перестановкой столбцов и строк, с наименьшим значением $\max\{|i-j| \colon a_{ij} \neq 0\}$. *Матрицей перестановок* называется $(0,1)$-матрица в каждой строке и столбце которой ровно одна 1. *Ширина ленты матрицы* A равна наименьшему из чисел k для которого существует матрица перестановок P такая, что матрица $P \cdot A \cdot P^T$ имеет

$$A = \begin{pmatrix} \boxed{0} & \boxed{1} & \boxed{1} & \boxed{1} & \boxed{1} & \boxed{1} \\ \boxed{1} & 0 & \boxed{0} & 0 & 0 & 0 \\ \boxed{1} & \boxed{0} & \boxed{0} & \boxed{0} & \boxed{0} & \boxed{0} \\ \boxed{1} & 0 & \boxed{0} & 0 & 0 & 0 \\ \boxed{1} & 0 & \boxed{0} & 0 & 0 & 0 \\ \boxed{1} & 0 & \boxed{0} & 0 & 0 & 0 \end{pmatrix} \quad B = \begin{pmatrix} \boxed{0} & \boxed{0} & \boxed{1} & \boxed{0} & 0 & 0 \\ \boxed{0} & \boxed{0} & \boxed{1} & \boxed{0} & \boxed{0} & 0 \\ \boxed{1} & \boxed{1} & \boxed{0} & \boxed{1} & \boxed{1} & \boxed{1} \\ \boxed{0} & \boxed{0} & \boxed{1} & \boxed{0} & \boxed{0} & \boxed{0} \\ 0 & \boxed{0} & \boxed{1} & \boxed{0} & \boxed{0} & \boxed{0} \\ 0 & 0 & \boxed{1} & \boxed{0} & \boxed{0} & \boxed{0} \end{pmatrix}$$

Рис. 7.1: Матрицы A и B.

$2k + 1$-диагональную форму. Таким образом, минимизируется ширина ленты матрицы. К примеру, ширина ленты матрицы A на рисунке 7.1 равна пяти, а лента матрицы B, получаемой из A перестановкой первого и третьего столбца и первой и третьей строки, имеет ширину три.

Задачу минимизации ширины ленты можно сформулировать и на языке теории графов. Пусть $L\colon V(G) \longrightarrow \{1, \ldots, |V(G)|\}$ — некоторое линейное упорядочение вершин графа G. Ширина ленты графа G, соответствующая упорядочению L, обозначаемая через $bw(G, L)$, определяется как

$$\max\{|L(u) - L(v)|\colon (u, v) \in E(G)\},$$

а *шириной ленты* (bandwidth) графа G называется

$$bw(G) \stackrel{\triangle}{=} \min\{bw(G, L)\colon L \text{ упорядочение } G\}.$$

Между ширинами лент графа и матрицы имеется простая связь. Для квадратной матрицы $A = \{a_{ij}\}_{i,j=1}^{n}$ определим граф G_A, каждой вершине которого сопоставлен столбец матрицы A и вершины $i \neq j$ в графе G_A смежны тогда и только тогда, когда элемент a_{ij} или a_{ji} не равен нулю. Тогда ширина ленты матрицы A равна ширине ленты графа G_A.

В электронике, при планировании сверхбольших интегральных схем часто требуется расположить связанных между собой узлы на одной линии (т.е. упорядочить вершины графа) так, чтобы наибольшая из длин связей между узлами была минимальной (т.е. ширина ленты графа)

Напомним, что расстоянием $d_G(u, v)$ между несовпадающими вершинами u и v в графе G называется длина кратчайшего пути, соединяющего эти вершины. Будем говорит, что граф G^n является n-й степенью графа G, если $V(G^n) = V(G)$ и $(u, v) \in E(G^n)$ тогда и только тогда, когда $d_G(u, v) \leq n$.

Обозначим через P_n^k k-ю степень пути с n вершинами. На рисунке 7.2 изображен граф P_{10}^4.

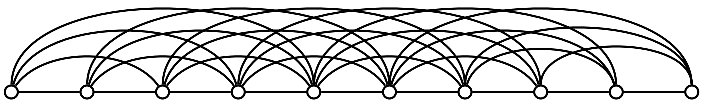


Рис. 7.2: P_{10}^4: четвертая степень пути с десятью вершинами.

Задача 7.9. Докажите, что граф с n вершинами имеет ширину ленты $\leq k$ тогда и только тогда, когда он является подграфом P_n^k.

Правильный граф интервалов (proper interval graph) это такой граф интервалов, для которого существует интервальная реализация в которой ни один из интервалов не содержится целиком в другом.

Задача 7.10. Докажите, что граф интервалов является правильным тогда и только тогда, когда не содержит $K_{1,3}$ в качестве порожденного подграфа.

Задача 7.11. Докажите, что граф интервалов является правильным тогда и только тогда, когда он является графом пересечений путей одинаковой длины.

Граф P_n^k является правильным графом интервалов, так как является графом пересечений сегментов $[i, i+k]$, $i \in \{1, \dots, n\}$. Определим $piw(G)$ как минимум кликового числа графа I, где минимум берется по всем правильным графам интервалов I, являющихся триангуляциями графа G. Поскольку, размер максимальной клики в P_n^k равен $k+1$, то верна следующая лемма:

Лемма 7.12. *Для всякого графа G верно равенство $bw(G) = piw(G) - 1$.*

Напоследок несколько задач о ширине ленты. Некоторые из них являются достаточно трудными.

Задача 7.13. Докажите, что $bw(G) \geq \left\lceil \frac{\Delta(G)}{2} \right\rceil$, где $\Delta(G)$ — наибольшая из степеней вершин графа G.

Оценку ширины ленты из предыдущей задачи можно уточнить. В работе [12] определялась интервальная степень графа G

$$id(G) \overset{\triangle}{=} \min\{\Delta(I) \colon I \text{ — граф интервалов, являющийся триангуляцией графа } G\}.$$

Задача 7.14. Докажите, что $bw(G) \leq id(G) \leq 2bw(G)$.

Задача 7.15. Докажите, что $bw(G) \geq \frac{vs(G^2)}{2}$, где $vs(G^2)$ — величина вершинного разделения графа второй степени графа G.

Для множества вершин $S \subseteq V(G)$ обозначим через ∂S множество вершин из S смежных одновременно вершинам S и $V(G) \setminus S$. Данное неравенство было замечено Harper [23].

Задача 7.16. Докажите, что $bw(G) \geq \max_k \min_{|S|=k} |\partial S|$.

Задача 7.17. Докажите, что $bw(G) \geq n - \frac{1+\sqrt{(2n-1)^2-8m}}{2}$, где $n = |V(G)|$, $m = |E(G)|$.

Задача 7.18. Докажите, что $bw(G) \geq \kappa(G)$, где $\kappa(G)$ — число связности графа G.

Задача 7.19. Докажите, что $bw(G) \geq \frac{n}{\alpha(G)} - 1$, где $\alpha(G)$ — число независимости графа G.

Плотностью (density) связного графа G называется

$$\max\left\{\left\lceil\frac{|V(H)|-1}{D(H)}\right\rceil : H \text{ — связный подграф } G\right\},$$

где $D(G)$ — диаметр графа G.

Задача 7.20. Докажите, что для всякого графа ширина ленты не меньше плотности.

Ширина ленты реберных графов изучалась в [25].

Задача 7.21. Докажите, что для всякого графа G с наименьшей степенью вершины ≥ 2 $bw(G) \leq bw(L(G))$, где $L(G)$ — реберный граф графа G.

Задача 7.22. Докажите, что для всякого дерева G $bw(G) \leq bw(L(G))$.

Задача 7.23. Докажите, что для всякого графа G $bw(G) \leq bw(L(G))$.

Иногда упорядочение L удобно трактовать как взаимно-однозначную функцию, отображающую вершины графа на некоторое множество целых чисел (не обязательно от 1 до $|V(G)|$). Следующая задача взята из работы

Задача 7.24. Пусть T — с k висячими вершинами. Докажите, что $bw(T) \leq \left\lceil k/2 \right\rceil$.

4 Поиск на вертолете

Будем называть *топологическим графом* граф, вершины которого суть точки в $\mathbb{R}^3$, а ребра — непересекающиеся конечнозвенные ломаные с концами в соответствующих вершинах. Пусть множество G, на котором рассматривается задача поиска, представляет собой конечный связный топологический граф (в дальнейшем просто граф) без петель и кратных ребер. Мы ограничимся лишь рассмотрением графов с ребрами единичной длины, имеющих по крайней мере две вершины. Множество таких топологических графов обозначим через $\mathcal{E}$.

На графе $G \in \mathcal{E}$ находятся двое игроков: преследователь и убегающий. Целью преследователя является обнаружение убегающего, убегающий старается уклониться от обнаружения. Действия преследователя задаются конечной последовательностью ходов, называемой программой поиска П. Каждым ходом преследователь ставится (перелетает на вертолете) из вершины в вершину (вершины не обязаны быть смежными). Таким образом, программу поиска П можно трактовать как отображение

$$\Pi\colon \{1, 2, \ldots, T\} \longrightarrow V(G),$$

где $\Pi(i)$, $i \in \{1, \ldots, T\}$, — вершина, занимаемая преследователем на i-м шаге.

Непрерывная функция

$$y\colon [1, T] \longrightarrow G$$

трактуется как траектория убегающего. Будем предполагать, что скорость убегающего ограничена некоторой константой μ, т.е. для всех $t_1, t_2 \in [1, T]$, $t_1 \neq t_2$,

$$\left| \frac{\rho(y(t_1), y(t_2))}{t_1 - t_2} \right| \leq \mu,$$

где $\rho(y(t_1), y(t_2))$ — длина (по евклидовой норме) кратчайшего пути с концами $y(t_1)$, $y(t_2)$, лежащего в G. Таким образом, убегающий не имеет возможности покидать G и на протяжении одного хода преследователя может пройти расстояние, не превосходящее μ.

Преследователь *обнаруживает* убегающего на i-м ходу, если $\rho(\Pi(i), y(i)) < 1$. При предположении, что ребра графа — отрезки, преследователь, вставая в вершину, "просматривает" все инцидентные ребра и "видит" убегающего, находящегося на одном из них, и в этом случае мы имеем дело с задачей типа «увидел — поймал». Программа поиска $\Pi(i)$, $i \in \{1, \ldots, T\}$, называется *выигрывающей*, если для всякой траектории убегающего $y(t)$,

$t \in [1, T]$, существует такое $i \in \{1, \ldots, T\}$, что на i-м ходу убегающий, совершающий движение по траектории y, обнаружен.

Отметим, что поставленную задачу поиска можно интерпретировать как задачу очистки ребер графа от «распыленного» убегающего. Будем говорить, что в программе Π точка $x \in G$ является загрязненной в момент времени $t^* \geq 1$, если существует траектория $y(t)$, $t \in [1, t^*]$, такая что $y(t^*) = x$, и, двигаясь по этой траектории, убегающий обеспечивает уклонение от обнаружения преследователем до $\lfloor t^* \rfloor$-го шага включительно. Множество $F(\Pi, G, t^*)$, состоящее из всех загрязненных в момент t^* точек графа G, будем называть загрязненным в момент t^* множеством, а множество $C(\Pi, G, t^*) = G \setminus F(\Pi, G, t^*)$ — очищенным множеством. Тогда программа $\Pi(i)$, $i \in \{0, \ldots, T\}$, является выигрывающей, если для некоторого $C(\Pi, G, T) = G$.

Существование выигрывающей программы преследователя в этой задаче зависит только от константы μ. Если эта константа мала, например, меньше $\frac{1}{n}$, где n — число вершин графа, то преследователь сможет поймать убегающего. Для графа G определим параметр $\mu(G)$ следующим образом:

$$\begin{aligned}\inf\{\mu\colon\ &\text{при скорости убегающего } \mu \text{ у преследователя на } G\\ &\text{не существует выигрывающей программы}\}.\end{aligned}$$

Нетрудно убедиться в истинности следующего простого, но очень важного для дальнейших рассуждений, факта:

Лемма 7.25. *Пусть* $\Pi(i)$, $i \in \{1, \ldots, T\}$, *— программа преследователя на графе* G, v *— вершина графа* G. *Если существует момент времени* $t^* \in [1, T]$ *такой, что* $v \in F(\Pi, G, t^*)$, *и для всех* $i \in \{\lceil t^* \rceil, \ldots, T\}$ $\Pi(i) \neq v$, *то программа* Π *не является выигрывающей.*

Предположим, что по каким-то причинам преследователь может посетить каждую вершину графа не более одного раза. Обозначим в этом случае наименьшее μ, для которого не существует выигрывающей программы на графе G, через $\mu_1(G)$.

Будем говорить, что подмножество ребер M графа G является k-запрещающим набором, если для всякого упорядочения L вершин графа G найдется ребро $e = (u, v) \in M$ такое, что $|L(u) - L(v)| \geq k$. Задача нахождения наибольшего k при котором в графе есть запрещающий набор рёбер «двойственна» задаче минимизации ширины ленты.

Лемма 7.26. *Для всякого графа* G $bw(G) \geq k$ *тогда и только тогда, когда в* G *есть* k*-запрещающий набор рёбер.*

Доказательство. Если $bw(G) \geq k$, то в качестве k-запрещающего набора можно взять множество $E(G)$. Если подмножество M ребер графа G является k-запрещающим набором, то порожденный этим подмножеством подграф $G[M]$ имеет ширину ленты $\geq k$, а потому $bw(G) \geq bw(G[M]) \geq k$. □

Лемма 7.27. *Если в графе $G \in \mathcal{E}$ есть k-запрещающий набор ребер, то $\mu_1 \leq \frac{1}{k}$.*

Доказательство. Для доказательства леммы нужно показать, что если убегающий передвигается со скоростью $\frac{1}{k}$, то он может уклониться от встречи с преследователем.

Сославшись на лемму 7.25 можно утверждать, что по окончании действия программы преследователь побывал в каждой вершина графа. Поскольку по условию преследователь посещает каждую вершину не более одного раза, то $T = |V(G)|$, и отображение

$$\Pi^{-1}\colon V(G) \longrightarrow \{1, \ldots, |V(G)|\}$$

является упорядочением вершин графа G. Пусть $e = \{u, v\}$ — ребро из k-запрещающего набора такое, что $|\Pi^{-1}(v) - \Pi^{-1}(u)| \geq k$. Будем считать, что $\Pi^{-1}(v) > \Pi^{-1}(u)$. Тогда действия убегающего, позволяющие ему уклониться, таковы: до момента $\Pi^{-1}(u)$ он стоит в вершине v, с момента $\Pi^{-1}(u)$ до момента $\Pi^{-1}(v)$ бежит со скоростью $\frac{1}{k}$ по ребру e из v в u и до момента T стоит в u. С $\Pi^{-1}(u)$-го до $\Pi^{-1}(v)$-го шага преследователь не «просматривает» ребро e и, потому, убегающий остается «незамеченным». □

Лемма 7.28. *Если $G \in \mathcal{E}$ и $bw(G) \leq k$, то $\mu_1(G) \geq \frac{1}{k}$.*

Доказательство. Пусть L — линейное упорядочение, такое что $bw(G, L) \leq k$. Определим программу преследователя $\Pi(i) = L^{-1}(i), i \in \{1, \ldots, |V(G)|\}$ (преследователь посещает вершины в задаваемом L порядке). Покажем, что если максимальная скорость убегающего строго меньше $\frac{1}{k}$, то программа Π будет выигрывающей.

Обозначим через $G[i]$ подграф графа G, порождаемый вершинами u, $L(u) \leq i$. Убедимся, что для всякого $i \in \{0, \ldots, |V(G)|\}$ выполнено условие:

$$G[i] \cap F(\Pi, G, i) = \emptyset, \tag{*}$$

т.е. в момент i подграф $G[i]$ очищен. Поскольку при $i = |VG|$ выполняется равенство $G[i] = G$, то выполнение (*) завершит доказательство теоремы.

Для $i = 0$ условие (*) выполняется (множество вершин u, $L(u) \leq 0$, пусто). Предположим, что после i-го шага условие (*) перестало выполняться. Это означает, что убегающему удалось пройти в одну из вершин u, $L(u) = j < i$, из некоторой смежной вершине u вершины v. До i-го шага условие (*) выполнялось, следовательно, вершина v посещается преследователем не раньше i, т.е. $L(v) = l \geq i$. На j-ом шаге преследователь встает в u и «просматривает« все ребра инцидентные u, так что убегающий может начать переходить из v в u только после момента j. В момент l преследователь встает в v, и убегающий должен попасть в u не позже l. По определению упорядочения L, $l - j \leq k$, а потому скорость убегающего должна быть $\geq \frac{1}{k}$. Противоречие, доказывающее утверждение (*), достигнуто. □

Как следствие из доказанных лемм получаем следующее утверждение:

Теорема 7.29 *Для всякого графа $G \in \mathcal{E}$ справедливо равенство $\frac{1}{\mu_1(G)} = bw(G)$.*

Доказательство. Пусть $bw(G) = k$. Тогда по лемме 7.26 в G есть k-захват и по лемме 7.27 $\mu_1(G) \leq \frac{1}{k}$. С другой стороны, по лемме 7.28 $\mu_1(G) \geq \frac{1}{k}$. □

5 Ширина ленты расщепления графа

Естественным обобщением ширины ленты графа является *ширина ленты расщепления* графа.

Рассмотрим операцию *расщепления вершины*. Пусть v — одна из вершин графа G. Разобьем ее окружение (т.е. множество $N(v)$) произвольным образом на две части M и N (отметим, что M и N могут быть пустыми). Выполним следующее преобразование графа G: удалим вершину v вместе с инцидентными ей ребрами, добавим новые вершины u и w и соединяющее их ребро $\{u, w\}$, вершину u соединим ребром с каждой вершиной из множества M, а вершину w — с каждой вершиной из множества N. Полученный в результате граф обозначим символом G_v. Будем говорить, что граф G_v получается из графа G *расщеплением вершины* v (рис. 7.3). Будем называть граф G^* *расщеплением* графа G, если G^* получается из G последовательным применением операции расщепления вершин. Стягивание ребра является «обратной» операцией к расщеплению вершины. Пусть u,w смежные вершины графа G. Определим граф H, как граф, получаемый из G удалением вершин u, w, добавлением

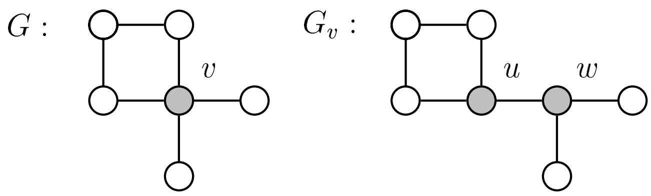


Рис. 7.3: Граф G_v получается из графа G расщеплением вершины v, а граф G из G_v стягиванием ребра $\{u, w\}$/

новой вершины v, смежной всем вершинам множества $N(u) \cup N(w)$, и удалением возможных кратных ребер. Будем говорить, что граф H получен из графа G *стягиванием* ребра $\{u, w\}$ (см. рис. 7.3). Граф G^* называется *сжатием* графа G, если G^* получается из G последовательным применением операции стягивания ребер. В некотором смысле операция расщепления обратна операции сжатия, поскольку всякий граф является сжатием своего расщепления и наоборот.

При доказательстве основной теоремы данного параграфа мы будем использовать следующий факт.

Задача 7.30. Докажите, что для всякого ребра e и вершины v графа G рёберно-(смешанно-, вершинно-) поисковые числа графов получаемых из G удалением e или v, а также стягиванием e, не превосходят рёберно-(смешанно, вершинно) поискового числа графа G.

Задача 7.31. Докажите, что смешанно-поисковое число графа P_n^k (k-ой степени n вершинного пути P_n) не превосходит k, т.е. $ms(P_n^k) \leq k$

Определим ширину ленты расщепления графа G, обозначаемую через $sbw(G)$, как

$$\min\{bw(G^*) \colon G^* \text{ является расщеплением } G\}.$$

Предположим, что преследователь может посещать вершины графа повторно, но не может допускать повторного загрязнения уже посещенных вершин. Другими словами, будем говорить, что программа поиска преследователя Π, $i \in \{1, \ldots, T\}$, на графе G является *монотонной*, если для всяких $i^* \in \{1, \ldots, T\}$ и $u \in G$ условие $u \in F(\Pi, G, i^*)$ влечет $u \in F(\Pi, G, i)$ для всех $i < i^*$. Обозначим через $\mu_m(G)$ наименьшее $\mu > 0$, при котором на графе G у преследователя не существует монотонной выигрывающей программы.

Мы покажем, что ширина ленты расщепления совпадает с $\frac{1}{\mu_m}$ и со смешанно-поисковым числом графа.

Теорема 7.32 *Для всякого графа $G \in \mathcal{E}$ следующие утверждения эквивалентны:*

i) $\frac{1}{\mu_m(G)} \leq k$*;*

ii) $sbw(G) \leq k$*;*

iii) $ms(G) \leq k$*.*

Доказательство. $i) \Rightarrow ii)$.

Доказательство. $i) \Rightarrow ii)$. Пусть $\frac{1}{\mu(G)} \leq k$. Тогда, если максимальная скорость убегающего не превосходит k, на графе G существует монотонная выигрывающая программа преследователя Π. Будем считать, что в этой программе $n \geq |V(G)|$ шагов и что монотонной выигрывающей программы с меньшим числом шагов не существует. Отметим следующие свойства программы Π.

Свойство 1. Если преследователь посещает вершину v в моменты t_1 и t_2, $t_1 < t_2$, то найдется момент $t \in \{t_1, t_1 + 1, \ldots, t_1 + k\}$ такой, что $\Pi(t) = v$.

$\triangle$ Поскольку число шагов в монотонной программе Π минимально, повторное посещение вершины может быть вызвано только тем, что в момент t_2 (а, следовательно, и в t_1) есть смежная вершине v вершина u, еще ни разу не посещаемая преследователем. Поэтому если преследователь на протяжении k шагов не посетит v, то в момент $t_1 + k$ произойдет повторное загрязнение этой вершины (из вершины u). $\bigtriangledown$

Свойство 2. Пусть $e = \{u, v\} \in E(G)$. Предположим, что к моменту t_1 — моменту первого посещения преследователем вершины v, вершина u уже посещалась преследователем. Тогда найдется момент времени $t < t_1$ такой, что $t_1 - t \leq k$ и $\Pi(t) = u$.

$\triangle$ Если на протяжении $t_1 - k, t_1 - k, \ldots, t_1 - 1$ вершина v не будет посещаться преследователем, то в момент $t_1 - 1$ произойдет повторное загрязнение этой вершины (из вершины u). $\bigtriangledown$

Обозначим через $\Pi^{-1}(v)$ множество всех моментов времени, в которые преследователь находился в вершине v. Построим расщепление графа G с шириной ленты $\leq k$. Рассмотрим граф H с $|V(G)|$ компонентами связности. Каждая компонента связности H_i графа H является путем, причем существует взаимно-однозначное отображение

$$P\colon V(G) \longrightarrow \{H_1, \ldots, H_{|V(G)|}\},$$

сопоставляющее вершине v путь с $|\Pi^{-1}(v)|$ вершинами. Зададим упорядочение вершин L графа H следующим образом. Для всех $v \in V(G)$ вершины пути $P(v) = (x_1, \ldots, x_l)$ получают номера из множества $\Pi^{-1}(v)$, т.е. $\{L(x_1), \dots, L(x_l)\} = \Pi^{-1}(v)$, и $L(x_1) < L(x_2) < \dots < L(x_l)$. Отметим, что по первому свойству программы Π для любых смежных вершин $x, y \in P(v)$ выполняется неравенство $|L(x) - L(y)| \le k$.

Для преобразования графа H в расщепление графа G нужно для каждого ребра $\{u, v\} \in E(G)$ одну из вершин пути $P(u)$ объявить смежной одной из вершин пути $P(v)$. По второму свойству программы Π для всякого ребра $\{u, v\} \in E(G)$ найдутся такие моменты $i_u \in \Pi^{-1}(u)$ и $i_v \in \Pi^{-1}(v)$, что $|i_u - i_v| \le k$. Добавляя для всякого ребра $\{u, v\} \in E(G)$ к графу H ребро $\{L^{-1}(i_u), L^{-1}(i_v)\}$, получаем расщепление графа G. По построению ширина ленты этого графа не превосходит k.
ii) $\Rightarrow$ *iii)*. При стягивании ребра смешанно-поисковое число не увеличивается (см. задачу 7.30). Поскольку всякое расщепление графа G можно, последовательно стягивая ребра, свести обратно к G, то для доказательства истинности импликации достаточно убедиться, что для всякого графа G $bw(G) \ge ms(G)$. Доказывать последнее неравенство можно многими способами. Например, показать, что $ms(P_n^k) \le k$ (см. задачу 7.31).
iii) $\Rightarrow$ *i)*. Будем считать, что граф G связен. Пусть $ms(G) \le k$. Обозначим через G' псевдограф, получаемый из G добавлением к каждой вершине петли. Очевидно, что $ms(G') = ms(G) \le k$. Поскольку в псевдографе G' все вершины имеют степень ≥ 2, то по теореме 6.30 смешанно-поисковое число совпадает с линейной шириной, т.е. $ms(G') = lw(G')$.

Таким образом, существует упорядочение L рёбер графа G^0 такое, что

$$\max_{i \in \{1, \dots, |E(G^0)|\}} \{\delta(E[i, L])\} \le k.$$

Не умаляя общности можно считать, что для всякого ребра $e \in E(G)$ смежные этому ребру петли имеют номера (при упорядочении L) меньшие $L(e)$. Поэтому для всякого $i \in \{2, \dots, |E(G^0)|\}$

$$|\delta(E[i, L]) - \delta(E[i-1, L])| \le 1.$$

Опишем программу поиска одного преследователя. Программа состоит из $n-1$ частей. i-я часть программы, $i \in \{1, \dots, n-1\}$, содержит $|\delta(E[i, L])|$ шагов, за которые преследователь обходит все вершины $\delta(E[i, L])$. Обход вершин преследователь совершает, добиваясь выполнения следующего условия: если вершина v посещается преследователем в $i-1$ и i-ой частях программы в моменты t_{i-1} и t_i соответственно, то $t_i - t_{i-1} \le k$. Добиться выполнения условия преследователь может всегда, поскольку $|\delta(E[i, L])| \le k$. Заметим, что для всех $i \le j \le k$ включение $v \in \delta(E[i, L]) \cap \delta(E[k, L])$ влечет $v \in \delta(E[j, L])$. А поскольку каждой вершине инцидентна петля, то всякая вершина содержится в одном из множеств $\delta(E[i,L])$. Таким образом, преследователь к концу программы посетит все вершины, и для доказательства того, что программа является монотонной и выигрывающей (при скорости убегающего $< k$), достаточно показать, что ни одна из посещенных вершин не окажется вновь загрязненной.

Предположим, что на j-ом шаге программы впервые произошло загрязнение уже очищенной вершины. Обозначим через v эту вершину, а через i — номер части программы, во время выполнения которой произошло загрязнение. Поскольку это первое загрязнение, вершина v смежна еще не посещаемой до j-го шага преследователем вершине u. Тогда $(u,v) \notin E[i,L]$, а потому $v \in \delta(E[i,L])$. Так как скорость убегающего $< k$ и $|\delta(E[i,L])| \leq k$, вершина v на протяжении i-ой части программы к моменту j еще не посещалась преследователем. Но ребро (u,v) упросматривалосьф преследователем в некоторый момент j' $(i-1)$-ой части программы. Как мы уже выяснили, в i-ой части программы вершина v должна быть пройдена преследователем после момента j. Но по построению программы поиска $j - j' < k$, и убегающий не успевает перебежать из u в v. Достигнутое противоречие завершает доказательство теоремы. $\square$

6 Поиск без ограничений

Рассмотрим теперь задачу поиска с одним преследователем без всяких ограничений на его действия. Преследователь может посещать вершины графа повторно и может допускать повторное загрязнения уже посещенных вершин. Обозначим через $\mu(G)$ наименьшее $\mu > 0$, при котором на графе G у преследователя не существует выигрывающая программа.

Величина $\mu(G)$ оценивается сверху через ширину $w(G)$ графа G.

Теорема 7.33 *Для всякого графа G, $\mu(G) \leq \left\lceil \frac{w(G)}{2} \right\rceil^{-1}$.*

Доказательство. Для доказательства неравенства удобнее воспользоваться задачей «двойственной» к задаче о ширине (см. лемму 3.32 на стр. 40). Нами доказано, что для всякого графа его ширина совпадает с его сцеплением. Пусть $w(G) = d$. Тогда существует подграф H графа G наименьшая из степеней вершин которого равна d. Пусть $\Pi(i)$, $i \in \{1,\ldots,T\}$, — некоторая программа поиска. Покажем как убегающий со скоростью $\left\lceil \frac{d}{2} \right\rceil^{-1}$ может уклониться от преследователя. Траектория убегающего строится индуктивно. Предположим, что для некоторого $i \in \{1,\ldots,T - \left\lceil \frac{d}{2} \right\rceil + 1\}$, $y(i) = u \in V(H)$, и для всякого $j \in$

$\{i, \ldots, \min(i + \left\lceil \frac{d}{2} \right\rceil - 1, T)\}$, $\Pi(j) \neq u$. При $i = 1$ такая вершина очевидно и найдется. Докажем существование момента $i' > i$ такого, что

1. $y(i') = v \in V(H)$;

2. для всех $j \in \{i', \ldots, \min(i' + \left\lceil \frac{d}{2} \right\rceil - 1, T)\}$, $\Pi(j) \neq v$;

3. для всех $j \in \{i, \ldots, i'\}$, $\rho(\Pi(j), y(j)) \geq 1$.

Если u не посещается преследователем после i-го шага, то доказательство очевидно. Пусть k — наименьший момент времени $\geq i + \left\lceil \frac{d}{2} \right\rceil$ для которого $\Pi(k) = u$. Если $k > i + \left\lceil \frac{d}{2} \right\rceil$, зададим $i' = k - \left\lceil \frac{d}{2} \right\rceil$. В этом случае просто стоит в u с i до i'.

Предположим, что $k = i + \left\lceil \frac{d}{2} \right\rceil$. Вершина u инцидентна как минимум d ребрам графа H, поэтому найдется ребро $(u, v) \in E(H)$ для которого $\Pi(j) \neq v$ при всех $j \in \{i + 1, \ldots, \min(i + d, T)\}$. В этом случае убегающий начинает бежать из u в v со скоростью $\left\lceil \frac{d}{2} \right\rceil^{-1}$ сразу после момента i. В момент $i' = k$ он достигает вершину v и для всех моментов $j \in \{i, \ldots, i'\}$ выполняется неравенство $\rho(\Pi(j), y(j)) \geq 1$. А так как $2\left\lceil \frac{d}{2} \right\rceil - 1 \leq d$, то $\Pi(j) \neq v$ для всех моментов $j \in \{i', \ldots, \min(i' + \left\lceil \frac{d}{2} \right\rceil - 1, T)\}$. □

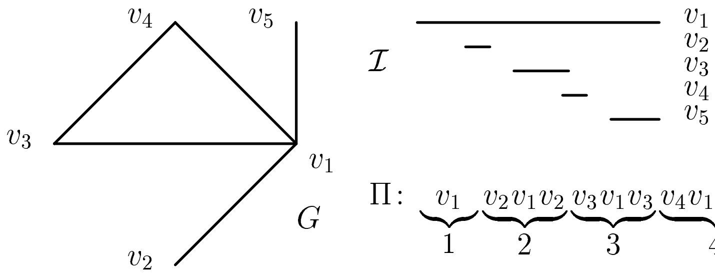
Снизу величину $\mu(G)$ можно оценить через путевую ширину $pw(G)$ графа G.

Для доказательства теоремы 7.35 нам понадобится следующее понятие. *Канонической реализацией* графа интервалов с n вершинами называется реализация, в которой левыми концами отрезков являются попарно различные целые числа от 1 до n.

Задача 7.34. Докажите, что для всякого графа интервалов существует каноническая реализация.

Теорема 7.35 *Для всякого графа G выполняется неравенство $\mu(G) \geq \frac{2}{pw(G)+1}$.*

Доказательство. Докажем сперва неравенство для случая, когда граф G является графом интервалов с n вершинами. Тогда $pw(G) = \omega(G) - 1$, где $\omega(G)$ обозначает кликовое число графа G (см. теорему 5.34 на стр. 63). Обозначим через $\mathcal{I}$ интервальную реализацию графа G. а через v_i — вершину, соответствующую интервалу $[i, r(i)] \in \mathcal{I}$, $i \in \{1, \ldots, n\}$. Определим $\delta(i)$ как множество всех вершин v_j , $j \leq i$, для которых $r(j) \geq$



$$\Pi\colon\ \underbrace{v_1}_{1}\ \underbrace{v_2v_1v_2}_{2}\underbrace{v_3v_1v_3}_{3}\underbrace{v_4v_1v_3v_4}_{4}\underbrace{v_5v_1v_5}_{5}$$

Рис. 7.4: Граф G, его каноническая реализация $\mathcal{I}$ и ссответствующая программа Π.

i. Поскольку все вершины множества $\delta(i)$ попарно смежны, то $|\delta(i)| \leq \omega(G)$, $i \in \{1, \dots, n\}$.

Опишем программу поиска $\Pi(i)$, $i \in \{1, \dots, T\}$, где

$$T = \sum_{i \in \{1,\dots,n\}} (|\delta(i)| + 1) - 1.$$

Π состоит из n частей. В начале i-ой части программы преследователь сперва встает в вершину v_i, а затем посещает остальные вершины (если они есть) множества $\delta(i)$ в порядке возрастания (см. рисунок 7.4). Таким образом, i-я часть состоит из $|\delta(i)| + 1$ шагов.

Покажем, что в случае, когда скорость убегающего меньше $\frac{2}{\omega(G)+1}$ программа Π является выигрывающей. Предположим противное — существование траектории убегающего $y(t)$, $t \in [1, T]$, такой, что для всех $i \in \{1, \dots, T\}$, $\rho(\Pi(i), y(i)) \geq 1$. Преследователь проходит все вершины графа G, поэтому должен существовать момент t — момент первого захода убегающего в вершину, уже посещенную преследователем. Обозначим соответствующую вершину через v_i. Определим $k \leq t$ как наибольшее из чисел для которых $\Pi(k) = v_i$. Отметим, что $t - k > \frac{pw(G)+1}{2}$. Если преследователь посещает v_i после k, то по определению программы Π имеется момент $l \in \{k+1, \dots, k+\omega(G)\}$ такой, что $\Pi(l) = v_i$. Поскольку $l - t < \frac{\omega(G)}{2}$, то зайдя в момент l в вершину v_i преследователь «обнаружит» убегающего, а потому мы заключаем, что после k-го шага преследователь не посещает v_i. Обозначим через v_j вершину, из которой убегающий попадает в v_i. Отметим, что вершины v_j и v_i смежны, а также $j > i$ и $v_i \in \delta(j)$. Поэтому для некоторого $m \in \{k+1, \dots, k+\omega(G)-1\}$, $\Pi(m) = v_j$. Итак, убегающий может начать бежать из v_j в v_i только после k, и, следовательно, преследователь видит его в момент m. Противоречие достигнуто.

Мы доказали, что для всякого графа интервалов G выполняется неравенство $\mu(G) \geq \frac{2}{\omega(G)}$. Если G — произвольный граф с путевой шириной равной $\theta - 1$, то найдется граф интервалов I, содержащий G в качестве подграфа, для которого $\omega(I) = \theta$. Нетрудно убедиться, что $\mu(G) \geq \mu(I)$ и, следовательно, $\mu(G) \geq \frac{2}{\theta}$. □

Поскольку максимальная клика в графе G порождает подграф, все вершины которого имеют степень $\omega(G) - 1$, то для всякого графа G ширина $w(G)$ не меньше $\omega(G) - 1$. Поэтому для всякого графа интервалов I верно неравенство $w(I) \geq pw(I)$. С другой стороны, для всякого графа G $w(G) \leq pw(G)$ (см. задачу 6.10). Таким образом, из теорем 7.33 и 7.35 получаем

Следствие 7.36. *Если кликовое число графа интервалов I четное, то* $\mu(I) = \left\lceil \frac{w(I)}{2} \right\rceil^{-1} = \frac{2}{pw(I)+1}$.

Вычислим $\mu(G)$ для полных графов с n вершинами.

Теорема 7.37 $\mu(K_n) = \left\lfloor \frac{n}{2} \right\rfloor^{-1}$.

Доказательство. Граф K_n является графом интервалов и если n четное, то теорема является частным случаем следствия 7.36.

Предположим , что $n = 2\theta + 1$. Пусть $(v_1, v_2, \ldots, v_n)$ — некоторое упорядочение вершин графа K_n. Определим программу преследователя $\Pi(t), t \in [0, n(\theta{+}1)]$ следующим образом: для всякого $t \in \{1, \ldots, n(\theta + 1)\}$, $\Pi(t) = v_s$, где $t \equiv s (\mathrm{mod}\ n)$. Таким образом, Π представима в виде следующей последовательности шагов:

$$\underbrace{v_1, v_2, \ldots, v_n, v_1, v_2, \ldots, v_n, \ldots, v_1, v_2, \ldots, v_n}_{\theta+1 \text{ раз}}.$$

Для вершин v_i, v_j определим «ориентированное расстояние»

$$\vec{\rho}(v_i, v_j) = \begin{cases} j - i, & \text{if } i \leq j; \\ n - i + j, & \text{if } i > j. \end{cases}$$

Иными словами, $\vec{\rho}(v_i, v_j)$ есть число ребер в ориентированном пути, соединяющего v_i и v_j в ориентированном цикле $(v_1, v_2, \ldots, v_n, v_1)$. Заметим, что $\vec{\rho}(v_i, v_j)$ $+\vec{\rho}(v_j, v_i)$=n.

Обозначим через r_k, $k \in \{1, \ldots, T - 2\theta\}$, наименьший момент $t \in [k, T]$ для которого $y(t) \in V(G)$.

Убедимся, что при скорости убегающего меньшей θ^{-1} программа П является выигрывающей. То, что программа П является выигрывающей, вытекает из следующих утверждений:

1. Если $\vec{\rho}(\Pi(k), y(r_k)) \leq \theta$, то преследователь обнаруживает убегающего в момент $k + \theta$.

2. Если $\vec{\rho}(\Pi(k), y(r_k)) = M > \theta$, то преследователь либо обнаруживает убегающего в момент $k + \theta$, либо существует $i \in \{k, \ldots, k + \theta\}$ для которого $\vec{\rho}(\Pi(i + \theta), y(r_{i+\theta})) \leq M - 1$.

Первое утверждение очевидно (скорость убегающего меньше θ^{-1}).

Приступим к доказательству второго утверждения. Положим $y(r_k) = u$ и $y(r_{k+\theta}) = v$. Если для всякого $i \in \{k, \ldots, k + \theta\}$ имеет место $v \neq \Pi(i)$, то $\vec{\rho}(\Pi(k + \theta), y(r_{k+\theta})) \leq \theta < M$. Предположим, что для некоторого $i \in \{k, \ldots, k + \theta\}$, $\Pi(i) = v$. Убегающий может начать двигаться из u в v только после момента i. Его скорость меньше θ^{-1} и добраться до v убегающий может только после $i + \theta$. Это означает, что $y(r_{i+\theta}) = v$. В момент $j \in \{k, \ldots, k + 2\theta\}$, для которого $\Pi(j) = u$, преследователь, встающий в u «просматривает» ребро (u, v). Следовательно, $j - i = \vec{\rho}(v, u) > \theta$ (в противном случае убегающий обнаружен на j-м шаге). Таким образом, $\vec{\rho}(u, v) = n - \vec{\rho}(v, u) < n - \theta = \theta + 1 \leq M - 1$.

Итак, мы доказали, что программа П является выигрывающей, а потому $\mu(K_n) \geq \left\lfloor \frac{n}{2} \right\rfloor^{-1}$. Ширина графа K_n равна $n - 1$ и по теореме 7.33 $\mu(K_n) \leq \left\lceil \frac{n-1}{2} \right\rceil^{-1} = \left\lfloor \frac{n}{2} \right\rfloor^{-1}$. □

Комментарии и ссылки

Первый параграф основан на обзоре [9].

При написании параграфа о поиске на ориентированных графах была использовалась статья [35]. В этой же работе обсуждается вершинный вариант поиска на орграфах.

Решения и указания

Задача 2.1. Рассмотрим набор клик $\{Q_1, Q_2, \ldots, Q_m\}$ такой, что всякая пара смежных вершин содержится в одной из клике этого набора. Определим набор множеств индексов $\{S_1, S_2, \ldots, S_n\}$, где $S_i = \{j \colon v_i \in Q_j\}$, т.е. S_i состоит из индексов клик, содержащих вершину i. Поскольку для всех $j \neq j$ $S_i \cap S_j$ тогда и только тогда, когда $\{v_i, v_j\} \in E(G)$, то G является графом пересечений множеств S_i. Определим граф H с множеством вершин $\{1, 2, \ldots, m\}$ и множеством ребер $\{i, j\} \in E(H)$ тогда и только тогда, когда индексы $\{i, j\}$ содержатся одновременно в одном из множеств S_k. Тогда каждое множество S_k является кликой в H и G является графом пересечений набора клик графа H.

Задача 3.2. Предположим противное: существует вершина $x \in S$, не смежная ни одной вершине, скажем, из $V(G_a)$. Тогда всякий путь с концами a и b в графе G содержит вершину из $S \setminus x$, а потому $S \setminus x$ тоже является a,b-разделителем, что противоречит минимальности S.

Задача 3.7. Нет.

Задача 3.9. Если граф не является полным, то в нем есть две симплициальные вершины. Но степени этих вершин не могут превосходить $k - 1$.

Задача 5.4. Возьмем произвольный граф G и «поставим» на каждое его ребро вершину степени два. Тогда G является графом пересечений «звезд» графа H (см. также стр. 11).

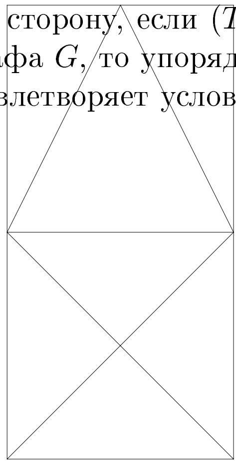
Задача 5.5. Если G не является графом интервалов, то никакое минимальное принимающее дерево этого графа не является путём.

Задача 5.6. Если в принимающем дереве есть вершина v степени > 3, то «расщепим» (формальное определение операции расщепления можно найти на странице 106) эту вершину так, чтобы у двух новых вершин степени были меньше степени v.

Задача 5.7. Решение задачи следует из свойства (T3) древесных декомпозиций. Более того, свойство (T3) может быть заменено в определении декомпозиции на условие данной задачи.

Задача 5.12. Пусть $\mathcal{X} = (X_1, X_2, \ldots, X_m)$ — набор максимальных максимальных клик графа интервалов G, удовлетворяющий условию теоремы 2.13, т.е. для всякой вершины $v \in V(G)$ и индексов $1 \le i \le j \le k \le m$ условие $v \in X_i \cap X_k$ влечёт $v \in X_j$. Тогда определив T как путь из m вершин, получаем путевую декомпозицию $(T, \mathcal{X})$ графа G.

В другую сторону, если $(T, \mathcal{X})$ — путевая декомпозиция максимальных клик графа G, то упорядочение максимальных клик, порождаемое путем T, удовлетворяет условиям теоремы 2.13.

Задача 5.15. 

Задача 3.10. Каждое из множеств K_i является кликой. Для произвольной максимальной клики K выберем вершину $v \in K$ с наименьшим номером i в упорядочении δ. Тогда $K_i = K$.

Задача 3.11. Пусть $\delta = (v_1, \ldots, v_n)$ — PE-упорядочение. Тогда $deg_{G[v_i,\ldots,v_n]}(v_i) \le k-1$, а $m = \sum_{i=1}^{n} deg_{G[v_i,\ldots,v_n]}(v_i)$. Поскольку $\sum_{i=n-k+1}^{n} deg_{G[v_i,\ldots,v_n]}(v_i) \le \frac{k(k-1)}{2}$, то $m \le (k-1)(n-k) + \frac{k(k-1)}{2}$.

Задача 3.12. Указание: рассмотреть PE-упорядочение.

Задача 3.24. Заметим, что $\alpha(G) = \omega(\overline{G})$ и $\theta(G) = \chi(\overline{G})$. Так как задача нахождения этих чисел есть ни что иное, как нахождение хроматического и кликовго числа в $\overline{G}$, то матрицу максимальных независимых множеств (в задаче о хроматическом числе) надо заменить на матрицу максимальных клик.

Задача 3.25. 5/2.

Задача 3.34. Для всякого графа имеет место $linkage(G) \geq \omega(G) - 1$. Поскольку вершина v_i является симплициальной в графе $G[v_i, v_{i+1}, \ldots, v_n]$, то

$$\omega(G) - 1 = \max_{i \in \{1,\ldots,n\}} deg_{G[v_i, v_{i+1}, \ldots, v_n]}(v_i).$$

Как мы уже доказали, $linkage(G) = w(G)$, поэтому

$$\max_{i \in \{1,\ldots,n\}} deg_{G[v_i, v_{i+1}, \ldots, v_n]}(v_i) = \max_{i \in \{1,\ldots,n\}} deg_{G[v_n, v_{n-1}, \ldots, v_i]}(v_i) \leq w(G).$$

Задача 3.35. Нет. Из алгоритма нахождения оптимального для ширины графа упорядочения следует, что всегда можно выбрать оптимальное упорядочение в котором вершина наименьшей степени стоит на последнем месте. Но не во всяком хордальном графе такая вершина является симплициальной.

Задача 3.36. Неравенство $\chi(G) \leq w(G) + 1$ вытекает из определения ширины графа. Поскольку хордальные графы совершенны, достаточно доказать $\omega(G) \geq w(G) + 1$. У хордального графа существует PE-упорядочение $\delta = (v_1, \ldots, v_n)$. Поскольку вершина v_i является симплициальной в графе $G[v_i, v_{i+1}, \ldots, v_n]$, то

$$\omega(G) - 1 = \max_{i \in \{1,\ldots,n\}} deg_{G[v_i, v_{i+1}, \ldots, v_n]}(v_i) \geq w(G).$$

Задача 3.29. Число кликового покрытия графа G совпадает с хроматическим числом графа $\overline{G}$, а поскольку $\alpha(G) = \omega(\overline{G})$, то утверждения упражнения вытекает из теоремы о совершенных графах.

Задача 3.30. Множество $y_1, y_2, \ldots, y_t$ является независимым множеством, поскольку если $\{y_i, y_j\} \in E(G)$, $i < j$, то $y_j \in K_{\delta(y_i)}$. Поэтому $\alpha(G) \geq t$ и $\alpha(G) = \omega(\overline{G}) \geq t$. С другой стороны, множество $\{K_{\delta(y_1)}, K_{\delta(y_2)}, \ldots, K_{\delta(y_t)}\}$ является кликовым покрытием графа G (см. задачу 3.10), а потому $\chi(\overline{G}) \leq t$. Дополнение хордального графа является совершенным графом, и, следовательно, $\chi(\overline{G}) = \omega(\overline{G}) = \alpha(G) = t$.

Задача 5.28. Пусть $tw(G) = k$. Воспользуемся неравенством $m \leq nk - \frac{1}{2}k(k+1)$ (см. упражнение 3.11). Решениями неравенства $k^2 - k(2k-1) +$

$2m \le 0$ являются числа $\frac{2n-1\pm\sqrt{(2n-1)^2-8m}}{2}$. Поскольку $k \le n-1$, получаем $k \ge \frac{2n-1-\sqrt{(2n-1)^2-8m}}{2}$.

Задача 5.29. Если граф полный, то его древесная ширина равна $n-1$. Если граф не является полным, то в G есть несмежные вершины x, y. Рассмотрим триангуляцию H графа G, в которой всего две клики: $V(G) \setminus \{x\}$ и $V(G) \setminus \{x\}$. Поскольку $\varphi(H) = n-1$, то $tw(G) \le n-2$.

Задача 5.30. См. задачу 3.9.

Задача 5.31. См. задачу 5.29

Задача 5.32 Если $\varrho(G) = n-2$, то (см. предыдущие упражнения) $n-2 \le tw(G) < n-1$. Если $tw(G) = n-2$, то $\varrho(G) \le n-2$. Если $\varrho(G) < n-2$, то найдутся две несмежные вершины x, y со степенями $\le n-3$. В графе G имеется n вершин, поэтому существует вершина $x_1 \ne y$, не смежная x, и вершина $y_1 \ne x$ (случай $x_1 = y_1$ не исключается), не смежная y. Тогда граф H, в котором три максимальные клики: $V \setminus (\{y\} \cup \{x_1\})$, $V \setminus (\{x\} \cup \{y\})$ и $V \setminus (\{x\} \cup \{y_1\})$, является триангуляцией графа G. Но $\varphi(H) - 1 = n-3 \ge tw(G)$ — противоречие.

Задача 5.36. Рассмотрим первый шаг на котором преследователь впервые удаляется из вершины v, инцидентной уже очищенному ребру (v, u). Предположим, что как в A_1, так и A_2, есть вершины еще не посещаемые преследователями. Пусть x_1 и x_2 вершины из разных долей еще не посещаемые преследователями. Тогда ребро (x_1, x_2) загрязнено. Вершина v смежна x_1, либо x_2, поэтому удаление преследователя из v влечет повторное загрязнение ребра (u, v). Таким образом, на некотором шаге все вершины одной из долей (для определенности A_1) должны быть заняты преследователями. Снять преследователя из вершины доли A_1 можно лишь после того, как будут пройдены все вершины доли A_2, а потому $ns(G) \ge \min\{|A_1|, |A_2|\} + 1$.

Задача 5.37. Указание: использовать теоретико игровую интерпретацию путевой ширины.

Задача 5.38. Путевую и древесную декомпозиции требуемой ширины построить просто. Пусть

$$k = \min\{pw(G_1) + |V(G_2)|, pw(G_2) + |V(G_1)|\} = pw(G_1) + |V(G_2)|.$$

В качестве путевой декомпозиции ширины $\leq k$ можно взять старую путевую декомпозицию графа G_1 и «добавить» к каждому узлу множество вершин $V(G_2)$. Такие же рассуждения годятся и для древесной ширины.

Для доказательства $k \leq pw(G_1 \times G_2)$ можно воспользоваться, как и в задаче 5.36, рассуждениями о поисковом числе. Рассмотрим некоторую программу поиска и выберем шаг поиска на котором число преследователей, стоящих в вершинах $V(G_1)$ максимально. Это число не меньше $ns(G_1)$. Когда после этого шага преследователь удаляется из $V(G_1)$, то все вершины $V(G_2)$ должны быть заняты преследователями (иначе удаление преследователя из $v \in V(G_1)$ стало бы причиной повторного загрязнения v). Таким образом,

$$pw(G_1 \times G_2) = ns(G_1 \times G_2) - 1 \geq ns(G_1) + |V(G_2)| - 1 = k.$$

Доказательство неравенства для древесной ширины, можно осуществить, используя следующий факт о триангуляциях: во всякой триангуляции H соединения графов G_1, G_2 по крайней мере одно из множеств $V(G_1)$, $V(G_2)$ является кликой. В самом деле, если $x_1, y_1 \in V(G_1)$, $x_2, y_2 \in V(G_2)$ и $\{x_1, y_1\}, \{x_2, y_2\} \notin E(H)$, то вершины x_1, x_2, y_1, y_2 порождают цикл длины четыре. Будем считать, что $V(G_1)$ является кликой в H. Пусть K — клика наибольшей мощности в H, все вершины которой принадлежат $V(G_2)$. По теореме 5.27 о триангуляциях $|K| - 1 \geq tw(G_2)$. Множество вершин $V(G_1) \cup K$ является кликой в H. Окончательно,

$$tw(G_2) + |V(G_1)| \leq |V(G_1)| + |K| \leq \omega(H).$$

Поскольку триангуляция H выбиралась произвольно, то

$$\min\{tw(G_1) + |V(G_2)|, tw(G_2) + |V(G_1)|\} \leq tw(G_1 \times G_2).$$

Задача 5.39. Триангуляция H графа G, определяемая в упражнении 5.32, является графом интервалов.

Задача 7.6. Обозначим через $D(P)$ орграф, вершинами которого являются элементы из X, а дугами упорядоченные пары (x, y) такие, что $x \prec y$ в P. Цепи в P соответствуют путям в $D(P)$, а антицепи, в силу транзититвности отношения, — независимым множествам. По теореме 7.3 существует разбиение элементов P на цепи мощности не превосходящей числу элементов в максимальной антицепи в P. Обратное неравенство очевидно, поскольку по транзитивности отношения $\prec$ никакие два элемента антицепи не могут принадлежать одной цепи.

Задача 7.17 Пусть $bw(G) = k$. Тогда $|E(P_n^k)| = \frac{k(2n-k-1)}{2}$, а потому $m \leq \frac{k(2n-k-1)}{2}$. Решая задачу другим способом, можно воспользоваться неравенством $bw(G) \geq pw(G) \geq tw(G)$ и сослаться на упражнение 5.28.

Задача 7.18 $\kappa(P_n^k) = k$.

Задача 7.19 $\alpha(P_n^k) = \left\lceil \frac{n}{k} \right\rceil$. Если $V(G) = V(H)$ и $E(H) \subseteq E(G) \Longrightarrow \alpha(G) \leq \alpha(H)$.

Задача 7.21 Пусть f — упорядочение ребер графа G, $bw(L(G), f) = k$. Опишем пошаговую процедуру построения нумерации g вершин графа. В начале каждого шага выбирается произвольная еще не имеющая номер вершина, называемая активной. Далее на каждом шаге нумерации выбирается ребро e с наименьшим (среди всех инцидентных активной вершине ребер с неиспользованными номерами) номером $f(e)$. Активной вершине присваивается номер $f(e)$, после чего ребро e становится использованным, а новой активной вершиной становится второй конец e. Шаг продолжается до тех пор, пока новая активная вершина не имеет номер.

Осталось показать, что для смежных вершин u, v $|g(u) - g(v)| \leq k$. Каждая вершина имеет номер одного из инцидентных ей ребер. Пусть $g(u) = a = f(e) < g(v) = b = f(e')$. Если e, e' смежны, то выполняется $|f(e) - f(e')| \leq k$. Если эти ребра не смежны, то ребро u, v имеет номер $c = f(u, v)$. Если $c > b$, то $b - a < c - a \leq k$. Если $c < b$, то вершина v могла получить номер b только потому, что номер c уже был занят. Но это означает, что вершина u не может иметь номер меньший c.

Задача 7.22 Пусть f — упорядочение ребер графа G, $bw(L(G), f) = k$. Выберем ребро (u, v) с номером 1. Зададим (временно) $g(u) = g(v) = 1$. Всякая вершина $x \neq u, v$ получает номер $g(x)$ равный номеру инцидентного ребра, лежащего на пути из x в (u, v). Тогда для всех $(x, y) \in E(G)$ $|g(x) - g(y)| \leq k$. Вершина с номером $k+1$ не может быть смежна одновременно вершинам u и v, а потому одной из этих вершин можно присвоить номер 0.

Задача 7.23 Пусть f — нумерация ребер G. Если G — дерево, то это задача 7.22. Если в G есть цикл, то удалив все висячие ребра, получим граф из задачи 7.21. Перенумеруем вершины этого графа, после чего будем добавлять удаленные висячие ребра. Добавляемая вершина получает номер инцидентного ребра.

Задача 7.24 Доказательство основано на следующем факте: дерево T является объединением $m = \lceil k/2 \rceil$ попарно пересекающихся путей. Таким набором путей может служить, например, множество путей $P_1, \ldots, P_m$, покрывающих висячие вершины дерева T и имеющих наибольшую суммарную длину. Действительно, если какое-то ребро дерева не покрывается этими путями, то найдутся непересекающиеся пути P_i и P_j. Последнее же противоречит выбору всего набора путей, поскольку P_i и P_j можно было бы заменить на пути с бо́льшей суммарной длиной. Поэтому для всякого j $T_j = \cup_{i=1}^{j} P_i$ является поддеревом и $T = \cup_{i=1}^{m} P_i$.

Упорядочение вершин L для которого $bw(T, L) \leq m$ можно построить следующим образом. Сперва присвоим вершинам пути P_1 (в порядке прохождения из одного конца в другой) порядковые номера $1, 1+m, 1+2m, 1+3m$ и т.д. Предположим теперь, что для некоторого j вершины поддерева $T_{j-1} = \cup_{i=1}^{j-1} P_i$ пронумерованы числами сравнимыми по модулю m с числами $\{1, 2, \ldots, j-1\}$ так, что $bw(T_{j-1}, L) \leq m$. Выберем вершины $u, v \in V(P_j) \cap T_{j-1}$, ближайшие к концам пути P_j (эти вершины могут и совпадать). Будем считать, что $L(u) \leq L(v)$. Пусть a — наибольшее целое число вида $j + km$ меньшее $L(u)$, а b — наименьшее целое число $j + km$ большее $L(v)$. Еще не упорядоченные вершины пути P_j порождают одну или две (взависимости от четности k) компоненты связности. Вершины этих компонент упорядочиваются следующим образом. Сперва присвоим номера $a - m, a - 2m, a - 3m$ и т.д. вершинам подпути идущего из u (некоторым вершинам могут быть приписаны отрицательные номера). Далее от v нумеруем вершины начиная с b «шагом» m.

Литература

[1] M. Aigner and M. Fromme, *A game of cops and robbers*, Discrete Appl. Math., 8 (1984), pp. 1–12.

[2] T. Andrea, *Note on a pursuit game played on graphs*, Discrete Appl. Math., 9 (1984), pp. 111–115.

[3] ——, *On a pursuit game played on graphs for which a minor is excluded*, J. Combin. Theory Ser. B., 41 (1986), pp. 37–47.

[4] R. Anstee and M. Farber, *On a bridged graphs and cop-win graphs*, J. Comb. Theory Series B, 44 (1988), pp. 22–28.

[5] A. Berarducci and B. Intrigila, *On the cop number of a graph*, Adv. in Appl. Math., 14 (1993), pp. 389–403.

[6] C. Berge, *Les problemes de colorations en theorie des graphs*, Publ. Inst Statist. Univ. Paris 9, (1960), pp. 123–160.

[7] D. Bienstock and P. Seymour, *Monotonicity in graph searching*, J. Algorithms, 12 (1991), pp. 239 – 245.

[8] J. R. S. Blair and B. Peyton, *An introduction to chordal graphs and clique trees*, in Graph theory and sparse matrix computation, IMA Vol. Math. Appl., A. George, J. R. Gilbert, and J. W. H. Liu, eds., vol. 56, 1993, pp. 1–29.

[9] J. A. Bondy, *Basic graph theory: Paths and circuits*, in Handbook of Combinatorics, Vol. 1, R. L. Graham, M. Grötschel, and L. Lovász, eds., Elsevier Science B.V., 1995, pp. 3–110.

[10] V. Chvatal, *Perfectly ordered graphs*, in Topics on Perfect Graphs, Ann. Discrete Math., **21**, C. Berge and V. Chvatal, eds., North Holland, 1984.

[11] G. Dirac, *On rigid circuit graphs*, Abhandl. Math. Sem. d. Univ. Hamburg, 25 (1961), pp. 71–76.

[12] F. V. Fomin and P. A. Golovach, *Interval completion with the smallest max-degree*, in Proceedings 24th International Workshop on Graph Theoretic Concepts in Computer Science WG'98, J. Hromkovič and O. Sýkora, eds., Springer Verlag, Lecture Notes in Computer Science, vol. 1517, 1998, pp. 359–372.

[13] P. Frankl, *Cops and robbers in graphs with large girth and Cayley graphs*, Discrete Appl. Math., 17 (1987), pp. 301–305.

[14] ——, *On a pursuit game on Cayley graphs*, Combinatorica, 7 (1987), pp. 67–70.

[15] D. Fulkerson and O. Gross, *Incidence matrices and interval graphs*, Pacif. J. Math, 15 (1965), pp. 835–855.

[16] F. Gavril, *The intersection graphs of subtrees in trees are exactly the chordal graphs*, J. Comb. Theory Series B, 16 (1974), pp. 47–56.

[17] P. C. Gilmore and A. J. Hoffman, *A characterization of comparability graphs and of interval graphs*, Canad. J. Math., 16 (1964), pp. 539–548.

[18] A. S. Goldstein and E. M. Reingold, *The complexity of pursuit on a graph*, tcs, 143 (1995), pp. 93–112.

[19] P. A. Golovach, *The cutwidth of a graph and the vertex separation number of the line graph*, Discrete Math. Appl., 3 (1993), pp. 517–521. *translation from Diskretn. Mat. 5, No.3, 76-80 (1993).*

[20] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.

[21] A. Haknal and J. Suranyi, *Uber die auflosung von graphen in vollstandige teilgraphen*, Ann. Univ. Budapest, Eotvos Sect. Math 1, (1958), pp. 113–121.

[22] Y. O. Hamidoune, *On a pursuit game on Cayley digraphs*, European J. Combin., 8 (1987), pp. 289–295.

[23] L. J. Harper, *Optimal numberings and isoperimetric problem on graphs*, J. Comb. Th., 8 (1966), pp. 385–393.

[24] T. R. Jensen and B. Toft, *Graph Coloring Problems*, Wiley-Interscience Series in Discrete Mathematics and Optimization, John Wiley & Sons, Inc., 1995.

[25] T. Jiang, D. Mubayi, A. Shastri, and D. B. West, *Edge bandwidth of graphs*, 1998. manuscript.

[26] L. M. Kirousis and C. H. Papadimitriou, *Interval graphs and searching*, Disc. Math., 55 (1985), pp. 181–184.

[27] ——, *Searching and pebbling*, Theor. Comp. Sc., 47 (1986), pp. 205–218.

[28] A. S. LaPaugh, *Recontamination does not help to search a graph*, J. ACM, 40 (1993), pp. 224–245.

[29] M. Maamoun and H. Meyniel, *On a game of policemen and robber*, Discrete Appl. Math., 17 (1987), pp. 307–309.

[30] F. S. Makedon and I. H. Sudborough, *On minimizing width in linear layouts*, Discrete Appl. Math., 23 (1989), pp. 243–265.

[31] T. A. McKee, *How chordal graphs work*, Bull. Inst. Comb. Appl., 9 (1993), pp. 27–39.

[32] T. A. McKee and F. R. McMorris, *Topics in Intersection Graph Theory*, SIAM Monographs on Discrete Mathematics and Applications, SIAM, Philadelphia, PA, 1999.

[33] N. Megiddo, S. L. Hakimi, M. R. Garey, D. S. Johnson, and C. H. Papadimitriou, *The complexity of searching a graph*, J. ACM, 35 (1988), pp. 18–44.

[34] R. H. Möhring, *Graph problems related to gate matrix layout and PLA folding*, in Computational Graph Theory, Comuting Suppl. 7, E. Mayr, H. Noltemeier, and M. Sysło, eds., Springer Verlag, 1990, pp. 17–51.

[35] R. J. Nowakowski, *Search and sweep numbers of finite directed acyclic graphs*, Discrete Appl. Math., 41 (1993), pp. 1–11.

[36] R. J. Nowakowski and P. Winkler, *Vertex-to-vertex pursuit in a graph*, Discrete Math., 43 (1983), pp. 325–329.

[37] T. D. Parsons, *Pursuit-evasion in a graph*, in Theory and Application of Graphs, Y. Alavi and D. R. Lick, eds., Berlin, 1976, Springer Verlag, pp. 426–441.

[38] N. N. Petrov, *Some extremal search problems on graphs*, Differ. Equations, 18 (1982), pp. 591–595. *Translation from Differ. Uravn 18, 821–827 (1982).*

[39] A. Quilliot, *A short note about pursuit games played on a graph with a given genus*, J. Comb. Theory B., 38 (1985), pp. 89–92.

[40] F. S. Roberts, *Discrete mathematical models, with applications to social, biological and environmental problems*, Prentice Hall, Englewoods Cliffs, NJ, 1976.

[41] R. Shamir, *Advanced topics in graph algorithms. lecture notes*, technical report, Tel-Aviv University, 1994.

[42] A. Takahashi, S. Ueno, and Y. Kajitani, *Mixed-searching and proper-path-width*, tech. rep., Tokyo Institute of Technology, 1991.

[43] ——, *Mixed-searching and proper-path-width*, Theor. Comp. Sc., 137 (1995), pp. 253–268.

[44] R. E. Tarjan and M. Yannakakis, *Simple linear time algorithms to test chordiality of graphs, test acyclicity of graphs, and selectively reduce acyclic hypergraphs*, SIAM J. Comput., 13 (1984), pp. 566–579.

[45] B. Toft, *Colouring, stable sets and perfect graphs*, in Handbook of Combinatorics, Vol. 1, R. L. Graham, M. Grötschel, and L. Lovász, eds., Elsevier Science B.V., 1995, pp. 233–288.

[46] R. Tošić, *On cops and robber game*, Studia Sci. Math. Hungar., 23 (1988), pp. 225–229.

[47] D. B. West, *Introduction to Graph Theory*, Prentice Hall, Upper Saddle River, NJ, 1996.

[48] В. А. Емеличев, О. И. Мельников, В. И. Сарванов, and Р. И. Тышкевич, *Лекции по теории графов*, М.: Наука, 1990.

[49] А. А. Зыков, *Основы теории графов*, М.: Наука, 1987.

Предметный указатель