УДК 519.717 Вестник СПбГУ. Сер. 1, 2001, вып. 4 (№ 25)

Ф. В. Фомин

ОБ ОДНОМ ОБОБЩЕНИИ ШИРИНЫ ЛЕНТЫ ГРАФА*

1. Постановка задачи. Первоначально задача о ширине ленты возникла в теории разреженных матриц (т. е. матриц, большинство элементов которых является нулями). Один из распространенных способов, повышающих эффективность работы с разреженными матрицами, — преобразование заданной матрицы A к ленточной форме (см., например, [1]) с минимальной шириной ленты: требуется найти матрицу перестановок P, такую, что в матрице $P \cdot A \cdot P^T$ все ненулевые элементы находятся «недалеко» от главной диагонали.

Задачу минимизации ширины ленты можно сформулировать и на языке теории графов. Под графом мы будем понимать простой (без петель и кратных ребер) конечный неориентированный граф. Обозначим через $V(G)$ множество вершин графа G и через $E(G)$ множество его ребер. Пусть $L\colon V(G) \longrightarrow \{1, \ldots, |V(G)|\}$ — некоторое упорядочение вершин графа G.

Определим *ширину ленты графа G, соответствующую упорядочению L*, как

$$bw(G, L) \triangleq \max\{|L(u) - L(v)|\colon (u, v) \in E(G)\},$$

а *ширину ленты (bandwidth) графа G* как

$$bw(G) \triangleq \min\{bw(G, L)\colon L \text{ — упорядочение вершин } G\}.$$

Подробную информацию о ширине ленты графа и ее разнообразных применениях можно найти в работах [2, 3], где приводится обширный обзор литературы. Отметим только, что задача нахождения ширины ленты остается NP-трудной даже на очень узком классе деревьев — так называемых гусеницах с волосами длины три [4].

Известно, что помещение вершин степени два на ребра графа может существенно уменьшить ширину ленты графа. Граф G' называется гомеоморфным образом графа G, если граф, изоморфный графу G', может быть получен из графа G помещением на ребра G некоторого числа вершин степени два.

Топологическая ширина ленты графа G определяется как $tbw(G) = \min\{bw(G')\colon G'$ — гомеоморфный образ $G\}$. Одним из интересных отличий топологической ширины ленты от ширины ленты является то, что вычисление топологической ширины деревьев осуществимо за полиномиальное время [5, 6].

Естественным обобщением топологической 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 вершин графа G, смежных вершине 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), соединим ребром вершину u с каждой вершиной из

*Работа выполнена при финансовой поддержке РФФИ (проект № 01-01-00235).

множества M, а вершину w — с каждой вершиной из множества N. Полученный граф обозначим символом G_v. Будем говорить, что граф G_v получается из графа G расщеплением вершины v.

Назовем граф G^* *расщеплением графа* G, если G^* получается из G последовательным применением операции расщепления вершин. В некотором смысле операция расщепления обратна операции сжатия (т. е. последовательного стягивания ребер), поскольку всякий граф является сжатием своего расщепления и наоборот.

Определим ширину ленты расщепления графа G как

$$sbw(G) \triangleq \min\{bw(G^*) \colon G^* \text{ — расщепление } G\}.$$

Поскольку всякий гомеоморфный образ графа является и его расщеплением, то для всякого графа G выполняются неравенства $sbw(G) \le tbw(G) \le bw(G)$. Также нетрудно убедиться, что если максимальная из степеней вершин графа G не превосходит трех, то $sbw(G) = tbw(G)$.

Известно, что нахождение топологической ширины является NP-трудной задачей (см. [5]) даже для графов с максимальной степенью вершин три. Поэтому задача нахождения ширины ленты расщепления графа также NP-трудна.

В данной работе мы докажем, что ширина ленты расщепления графа совпадает с собственно путевой шириной графа (proper pathwidth).

Понятие путевой ширины было введено Н. Робертсоном и П. Сеймуром в их первой работе [7] по теории миноров графов.

Путевой декомпозицией графа G называется последовательность подмножеств вершин графа $\{X_i\}_{1\le i\le r}$, обладающая следующими свойствами:

(P1) $\bigcup_{1\le i\le r} X_i = V(G)$;

(P2) для всякого ребра $(u, v) \in E(G)$ существует $1 \le i \le r$, такое, что $u, v \in X_i$;

(P3) $X_i \cap X_k \subseteq X_j$ для всех $1 \le i \le j \le k \le r$.

Шириной путевой декомпозиции $\{X_i\}_{1\le i\le r}$ называется величина

$$\max_{1\le i\le r} |X_i| - 1.$$

Путевая ширина $pw(G)$ графа G определяется как минимальная ширина путевой декомпозиции, причем минимум берется по всевозможным путевым декомпозициям графа G.

А. Тахакаши, С. Уено и В. Кажитани [8] определили собственно путевую декомпозицию как путевую декомпозицию, удовлетворяющую следующим дополнительным свойствам:

(P4) для всех $1 \le i < j \le r$, $X_i \not\subseteq X_j$;

(P5) для всех $1 \le i < j < k \le r$, $|X_i \cap X_k| \le |X_j| - 2$.

Собственно путевой шириной $ppw(G)$ графа G называется минимальная ширина путевой декомпозиции, причем минимум берется по всем собственно путевым декомпозициям графа G.

Основным результатом настоящей работы является доказательство равенства $sbw(G) = ppw(G)$.

Приведем ряд вспомогательных утверждений и определений.

Будем использовать понятие линейной ширины графа, введенное Р. Томасом [9] и являющееся «родственным» понятию «crusaders», определенному Биенстоком и Сеймуром [10]. Для подмножества ребер $X \subseteq E(G)$ графа G определяем $\delta(X)$ как множество вершин, одновременно инцидентных ребрам из множеств X и $E(G) \setminus X$.

Пусть $L\colon E(G) \longrightarrow \{1, \dots, |E(G)|\}$ — некоторое упорядочение ребер графа G. Для $i \in \{1, \dots, |E(G)|\}$ определим множество ребер $E[i, L] = \{e \in E(G)\colon L(e) \leq i\}$.

Линейная ширина графа G, соответствующая упорядочению L, определяется как

$$lw(G, L) \triangleq \max_{i \in \{1, \dots, |E(G)|\}} |\delta(E[i, L])|,$$

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

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

В терминах линейной ширины переформулируем утверждение, доказанное в [8].

Теорема 1 [8]. *Если наименьшая из степеней вершин графа G не меньше двух, то $lw(G) = ppw(G)$.*

Как линейную, так и путевую ширину, можно определять и для псевдографов. Заметим, что добавление петель не меняет собственно путевую ширину графа. Мы будем использовать следующий факт, вытекающий из теоремы 1.

Следствие 1. *Пусть псевдограф G^0 получен из графа G добавлением петли к каждой вершине. Тогда $lw(G^0) = ppw(G)$.*

2. Задача поиска. Для доказательства основного утверждения работы введем вспомогательный параметр $\mu(G)$, имеющий следующую теоретико-игровую интерпретацию. Обозначим через $\mathcal{E}$ множество конечных связных топологических графов, вложенных в некоторое конечномерное пространство $\mathbf{R}^n$ и с ребрами единичной длины. Рассмотрим следующую задачу поиска на графе $G \in \mathcal{E}$. На G 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 Π. Первым ходом преследователь выбирает одну из вершин графа G. Каждым последующим ходом преследователь переставляется (например, перелетает на вертолете) из вершины в вершину (вершины не обязаны быть смежными).

Таким образом, программу поиска Π можно трактовать как отображение

$$\Pi\colon \{1, 2, \dots, T\} \longrightarrow V(G),$$

где $\Pi(i)$, $i \in \{1, \dots, T\}$, — вершина, занимаемая преследователем на i-м шаге.

Непрерывная (в топологии $\mathbf{R}^n$) функция $y\colon [0, T] \longrightarrow G$ трактуется как траектория убегающего. Будем предполагать, что скорость убегающего ограничена некоторой константой μ, т. е. для всех t_1, $t_2 \in [0, T]$, $t_1 \neq t_2$,

$$\left| \frac{\rho(y(t_1), y(t_2))}{t_1 - t_2} \right| \leq \mu,$$

где $\rho(y(t_1), y(t_2))$ — длина (по евклидовой норме) кратчайшего пути, лежащего в G, с концами $y(t_1)$, $y(t_2)$. Таким образом, убегающий не имеет возможности покидать G и во время 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 $\Pi(i)$, $i \in \{1, \dots, T\}$, называется *выигрывающей*, если для всякой траектории убегающего $y(t)$, $t \in [0, T]$, существует такое $i \in \{1, \dots, T\}$, 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 такая траектория $y(t)$, $t \in [0, t^*]$, что $y(t^*) = x$, и убегающий, двигаясь по этой траектории, 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.

Естественно предполагать, что $G = F(\Pi, G, t)$ для всех $t \in [0, 1)$. Тогда программа $\Pi(i)$, $i \in \{0, \dots, T\}$, является выигрывающей, если $C(\Pi, G, i^*) = G$ для некоторого $i^* \in \{1, \dots, T\}$.

Существование выигрывающей программы преследователя на заданном графе зависит только от константы μ. Если эта константа мала, например меньше $1/n$, где n — число вершин графа, то преследователь сможет поймать убегающего.

Предположим, что преследователь не может допускать повторного загрязнения уже посещенных вершин. Другими словами, будем говорить, что программа поиска преследователя $\Pi(i)$, $i \in \{1, \dots, T\}$, на графе G является *монотонной*, если для всяких $i^* \in \{0, \dots, T\}$ и $u \in G$ условие $u \in F(\Pi, G, i^*)$ влечет $u \in F(\Pi, G, i)$ для всех $i < i^*$.

Для графа G определим параметр $\mu(G)$ следующим образом:

$$\inf\{\mu\colon \text{ при скорости убегающего } \mu \text{ у преследователя на } G \\ \text{ не существует монотонной выигрывающей программы}\}.$$

В работе [11] доказано, что для всякого графа G с числом ребер > 1 параметры $sbw(G)$ и $1/\mu(G)$ совпадают. В настоящей работе мы приведем краткое доказательство этого факта и, более того, докажем, что $sbw(G) = ppw(G) = 1/\mu(G)$.

3. Основной результат.

Теорема 2. *Для всякого графа G с непустым множеством ребер следующие утверждения эквивалентны:*

i) $\frac{1}{\mu(G)} \leq k$;

ii) $sbw(G) \leq k$;

iii) $ppw(G) \leq k$.

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 $t \in \{t_1, t_1 + 1, \dots, t_1 + k\}$, такой, что $\Pi(t) = v$. 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 вершина u, смежная вершине v, еще ни разу не посещаемая преследователем. Поэтому, если преследователь во время k шагов не посетит v, то в момент $t_1 + k$ произойдет повторное загрязнение этой вершины (из вершины u).

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 $\Pi(t) = u$.

Если в моменты времени $t_1 - k, t_1 - k, \ldots, t_1 - 1$ вершина v не будет посещаться преследователем, то в момент $t_1 - 1$ произойдет повторное загрязнение этой вершины (из вершины u).

Обозначим через $\Pi^{-1}(v)$ множество всех моментов времени, в которые преследователь находился в вершине v. Построим расщепление графа G с шириной ленты, не превосходящей 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. $\{L(x_1), \ldots, L(x_l)\} = \Pi^{-1}(v)$, и $L(x_1) < L(x_2) < \ldots < L(x_l)$.

Отметим, что по первому свойству программы Π для любых смежных вершин $x, y \in P(v)$ выполняется неравенство $|L(x) - L(y)| \leq k$. 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 $|i_u - i_v| \leq k$. 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)*. Нетрудно убедиться, что при стягивании ребра собственно путевая ширина графа не увеличивается. Поскольку всякое расщепление графа G можно свести обратно к G, последовательно стягивая ребра, то для доказательства истинности импликации достаточно убедиться, что $bw(G) \geq ppw(G)$ для всякого графа G.

Пусть L — оптимальное (для ширины ленты) упорядочение вершин графа G, $bw(G, L) = p$. Тогда последовательность подмножеств вершин графа

$$X_i \triangleq \{L^{-1}(i), L^{-1}(i+1), \ldots, L^{-1}(i+p-1)\},$$

$i \in \{1, \ldots, n-p+1\}$, является собственной путевой декомпозицией графа G ширины k.

iii) $\Rightarrow$ *i)*. Пусть $ppw(G) \leq k$. Обозначим через G^0 псевдограф, получаемый из G добавлением петли к каждой вершине. Как отмечалось в следствии 1 для ребер графа G^0 существует упорядочение L, такое, что

$$\max_{i \in \{1, \ldots, |E(G^0)|\}} \{\delta(E[i, L])\} \leq k.$$

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 $i \in \{2, \ldots, |E(G^0)|\}$

$$|\delta(E[i, L]) - \delta(E[i-1, L])| \leq 1.$$

Опишем программу поиска одного преследователя. Программа состоит из $n-1$ частей, таких, что каждая 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-й частях программы в моменты t_{i-1} и t_i соответственно, то $t_i - t_{i-1} \le k$, что всегда выполняется, 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, к концу программы преследователь 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-м 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 $|\delta(E[i,L])| \le k$, вершина v на протяжении i-й части программы к моменту j еще не посещалась преследователем. Но ребро (u,v) «просматривалось» преследователем в некоторый момент j' выполнения $(i-1)$-й части программы. Как мы уже выяснили, в i-й 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.

Summary

Fomin F.V. On a generalization of the graph bandwidth.

We establish an interesting relation between two graph parameters: bandwidth and pathwidth. In particular, we show that split bandwidth is equal to proper pathwidth.

Литература

1. *Тьюарсон Р.* Разреженные матрицы. М., 1977. **2.** *Chinn P.Z., Chvátalová J., Dewdney A.K., Gibbs N.E.* The bandwidth problem for graphs and matrices — a survey // J. Graph Theory. 1982. Vol. 6. P. 223–254. **3.** *Chung F.* Labelings of graphs // Selected Topics in Graph Theory. New York, 1988. P. 151–168. **4.** *Monien B.* The Bandwidth Minimization Problem for Caterpillars with Hair Length 3 is NP-complete // SIAM J. Alg. Disc. Meth. 1986. Vol. 7. P. 505–512. **5.** *Makedon F.S., Papadimitriou C.H., Sudborough I.H.* Topological bandwidth // SIAM J. Alg. Disc. Meth. 1985. Vol. 6. P. 418–444. **6.** *Miller Z.* A linear algorithm for topological bandwidth in degree-three trees // SIAM J. Comput. 1988. Vol. 17. P. 1018–1035. **7.** *Robertson N., Seymour P.D.* Graph minors. I. Excluding a forest // J. Comb. Theory. Series B. 1983. Vol. 35. P. 39–61. **8.** *Takahashi A., Ueno S., Kajitani Y.* Mixed-searching and proper-path-width // Theor. Comp. Sc. 1995. Vol. 137. P. 253–268. **9.** *Thomas R.* Tree decompositions of graphs. Lecture notes. Georgia Institute of Technology, Atlanta, Georgia, 30332, USA, 1996. **10.** *Bienstock D., Seymour P.D.* Monotonicity in graph searching // J. Algorithms. 1991. Vol. 12. P. 239–245. **11.** *Fomin F.V.* Helicopter search problems, bandwidth and pathwidth // Disc. Appl. Math. 1998. Vol. 85. P. 59–71.

Статья поступила в редакцию 16 марта 2001 г.