Санкт-Петербургский государственный университет
Прикладная математика и информатика
Исследование операций и принятие решений в задачах оптимизации, управления и экономики

Ржевская Екатерина Эдуардовна

Проблема связности в некоторых задачах поиска на графах

Бакалаврская работа

Научный руководитель:
к. ф.-м. н., старший преподаватель
Т. В. Абрамовская

Рецензент:
к. ф.-м. н., доцент Н. И. Наумова

Санкт-Петербург
2016

Saint Petersburg State University
Applied Mathematics and Computer Science
Operation Research and Decision Making in Optimisation, Control and Economics Problems

Rzhevskaia Ekaterina Eduardovna

Problem of connectivity in some search problems on graphs

Bachelor's Thesis

Scientific Supervisor:
Senior Lecturer T. V. Abramovskaia

Reviewer:
Associate Professor N. I. Naumova

Saint Petersburg
2016

Содержание

1. Введение

В данной работе рассматривается задача поиска с дискретным временем мобильного объекта на графе. На связном неориентированном графе G находится невидимый объект, перемещения которого невозможно предсказать. К его поимке стремится множество *игроков* S, которые также могут передвигаться на G. Условия, при которых искомый объект пойман игроками, а также некоторые другие параметры задачи, задают вид поиска. Тем не менее, в каждой задаче гарантированного поиска требуется найти минимальное количество игроков, которые ловят объект независимо от его действий — эта величина называется *поисковое число графа* G.

В обзоре [1] Т. В. Абрамовской и Н. Н. Петрова приводится следующая интерпретация задачи поиска на графах. Пусть в пещере, состоящей из лазов и ходов, потерялся спелеолог. У спасателей есть карта пещеры в виде графа, но перемещения исследователя непредсказуемы, а передвигаться он может с любой скоростью. Более того, следует предположить, что он действует себе во вред и самостоятельно не может выбраться из пещеры. Таким образом, спелеолог никак не может помочь спасателям, которым нужен алгоритм действий, гарантирующий, что хоть один из команды спасателей наткнётся на блуждающего спелеолога. П. А. Головач показал в [2], что задача о поиске спелеолога сводится к дискретной задаче поиска на графе.

Часто дискретную задачу поиска удобно формулировать как задачу *очищения* графа. Будем говорить, что ребро *очищено*, если на нём гарантированно нет объекта, иначе ребро *загрязнено*. Считается, что изначально все рёбра графа загрязнены, а игроки стремятся обеспечить момент, когда все рёбра графа единовременно очищены. Далее в работе будем придерживаться именно такой терминологии.

Для описания возможных действий игроков введём следующие *шаги* *поиска*:

1. поставить игрока в вершину графа;

2. игрок скользит вдоль ребра из одного из концов, где он ранее находился;

3. снять игрока с вершины.

Очищенное ребро $e = (u, v)$ защищено от загрязнения в данный момент, если для каждой из вершин u, v верно, что в вершине стоит игрок или все рёбра, инцидентные ей, очищены. Иными словами, ребро повторно загрязняется, если в вершинах на пути от него до загрязнённого ребра нет ни одного игрока. Можно считать, что в начальный момент на графе нет ни одного игрока, а дальше выполняется последовательность шагов поиска, которая называется *стратегией поиска*. Стратегию называют *монотонной (monotone)*, если при её использовании не допускается повторное загрязнение рёбер. Если на каждом шаге множество очищенных рёбер индуцирует связный подграф, то такая стратегия является *связной*. Использование связной стратегии моделирует ситуацию, когда игрокам необходимо поддерживать безопасный канал связи. Например, спасатели в пещере могут посылать сообщения друг другу и быть уверенными, что сообщение не будет перехвачено спелеологом.

Существуют несколько видов поиска, которые различаются условиями очищения рёбер и множеством допустимых стратегий. В *стандартном поиске* разрешены все шаги поиска, перечисленные выше, а чтобы очистить ребро $e = (u, v)$, игроку необходимо проскользить вдоль ребра e от одного его конца к другому. Стандартный поиск был впервые поставлен в [3] Н.Н. Петровым и в [4] Т.Д. Парсонсом. Поисковое число стандартного поиска обозначается $s(G)$. Если потребовать, чтобы все стратегии в таком поиске были связными, то получим *связный поиск (connected search)*, поисковое число которого обозначается $cs(G)$.

Поиск, в котором запрещён шаг 2, а ребро $e = (u, v)$ считается очищенным, если для в вершинах u, v стоят игроки, называется *вершинным поиском (node search)*, тогда $ns(G)$ – поисковое число в данном случае. Вершинный поиск был поставлен в [5] Кироусисом и Пападимитриу, там же была показана связь с «игрой в камни».

В *смешанном поиске (mixed search)*, сформулированном Такахаши, Уено и Каджитани в [6], разрешены все шаги поиска, а ребро считается очищенным, если оно очищено по условиям вершинного или стандартного поисков, $mixs(G)$ — поисковое число смешанного поиска.

В случае, когда множество очищенных вершин смешанного поиска индуцирует связный подграф на каждом шаге, говорят, что поиск *связный смешанный (connected mixed search)*, а его поисковое число обозначают $cmixs(G)$. Для каждого поиска можно определить монотонную стратегию и поисковые числа $mmixs(G)$, $mns(G)$, $mcmixs(G)$. Нас интересуют *стандратный*, *смешанный (mixed)* и *связный смешанный (connected mixed search)* поиски. В следующей главе будет введён *блочный поиск* для специального класса графов, сформированного на основе примера в статье [7], среди авторов которой — Д.М. Тиликос и Н. Санторо. Для некоторых классов блочных графов будут найдены поисковые числа интересующих нас поисков и построены оптимальные стратегии.

Для любого связного графа между поисковыми числами известны следующие соотношения, описанные в [8] Б.Йангом:

1. $ns(G) - 1 \leqslant s(G) \leqslant ns(G) + 1$;

2. $mixs(G) - 1 \leqslant s(G) \leqslant mixs(G) + 1$;

3. $mixs(G) - 1 \leqslant ns(G) \leqslant mixs(G) + 1$;

Соотношения (1)–(3) показывают связь между стандартным, вершинным и смешанным поиском.

4. $mixs(G) \leqslant cmixs(G) \leqslant cs(G)$;

5. Если G_0 — минор G, то $s(G) \geqslant s(G_0)$, $ns(G) \geqslant ns(G_0)$, $mixs(G) \geqslant mixs(G_0)$;

6. Если граф T — дерево, то $s(T) = ms(T) \leqslant cs(T) = mcs(T) \leqslant 2s(T) - 2$;

Неравенства 4, 6 объясняют, как можно получить оценку для поискового числа связного поиска.

7. $mixs(G) - 1 \leqslant pw(G) \leqslant mixs(G)$;

8. $s(G) = ms(G)$, $ns(G) = mns(G)$, $mixs(G) = mmixs(G)$;

9. $s(G) \leqslant cs(G) \leqslant mcs(G)$;

10. $mixs(G) \leqslant cmixs(G) \leqslant mcmixs(G)$.

Соотношение (7) демонстрирует связь поискового числа с путевой шириной — важным инвариантом графа, о чём более подробно рассказано в главе 8. Равенство (8) позволяет рассматривать только монотонные стратегии для нахождения поисковых чисел стандартного, смешанного и вершинного поисков, а пункты 9,10 показывают, что для связных поисков это неверно.

Особого внимания заслуживает соотношение (5). *Минором* графа G называется граф G_0, полученный из графа G с помощью последовательного удаления некоторых ребёр, вершин и стягивания некоторых ребёр. Теория миноров началась с цикла статей Робертсона и Сеймура. На рисунках 2 показаны примеры графов, которые являются или не являются минорами для исходного графа, изображённого на 1. Сам граф тоже является своим минором, как и граф, содержащий только одну вершину. В главе 7 проиллюстрирован пример, в котором для нахождения интересующих нас поисковых чисел достаточно удачно взять минор исходного графа, для которого искомые величины уже сосчитаны, а далее воспользоваться соотношением (5). Оказывается, что класс таких графов G, что $s(G) \leqslant k$, замкнут относительно операции взятия минора. То же самое можно сказать про смешанный и вершинный поиски. В главе 6 доказывается, что связный поиск таким свойством не обладает.

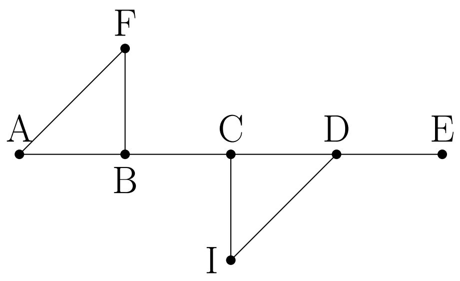

Рис. 1. Граф G

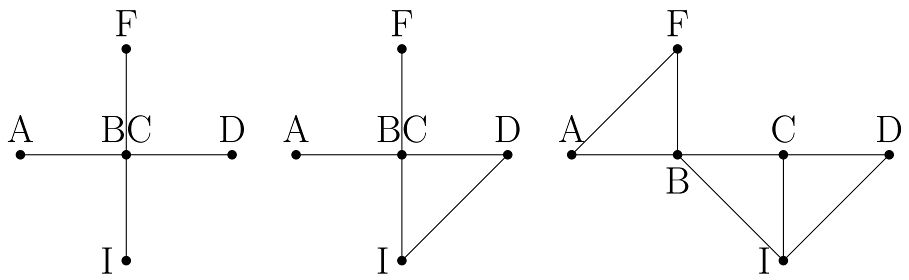

Рис. 2. Графы G_0, G_1, G_2

На рисунке 2 граф G_1 — минор G, полученный удалением вершины E, удалением ребра (A, F) и стягиванием ребра (B, C), а граф G_0 — минор G_1, в котором удалено ребро (I, D), следовательно, G_0 — минор G. Граф G_2 не является минором G, так как ребро (B, I) не могло появиться в миноре графа без стягивания ребра (B, C).

2. Блоки и графы блочной структуры

Рассмотрим класс графов, которые можно изобразить на плоскости как сетку размера $m \times n$, где пересечения строк и столбцов — вершины нашего графа. Такие графы назовем *блоками* размерами $m \times n$. В блоках mn вершин, $m(n-1)+n(m-1)$ рёбер. Для блоков можно определить *границу*, как подграф, порожденный вершинами степени меньше 4. Граница каждого блока состоит из четырех частей: верхней, нижней, правой и левой. На рисунке 3 пример блока, где $m = 11,\ n = 41$, граница которого выделена.

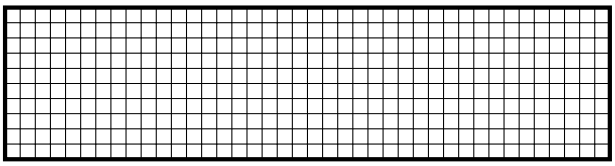

Рис. 3. Граф–блок

Определим процедуру *склейки* двух блоков $B_1 \sqcup B_2$ размеров $m_1 \times n_1, m_2 \times n_2$ так: *все* вершины одной из частей границ B_1 отождествляются с вершинами одной из частей границ B_2, при этом кратные рёбра не появляются. Фактически, блок B_1 «приклеивается» к границе B_2. *Границей* полученного графа назовём множество $\Gamma = \Gamma_1 \bigcup \Gamma_2 \setminus \gamma$, где Γ_1 — граница B_1, Γ_2 — граница B_2, а γ— множество рёбер между «склеивающимися» вершинами границ. Понятно, что склейка блоков определяется неоднозначно, что видно на рисунке 4, а на 5 показан пример графа, не являющегося склейкой двух блоков.

Процедура склейки графа $B_1 \sqcup B_2$ и блока B_3 определяется практически так же, как и склейка блоков, с той лишь разницей, что теперь одна из границ B_3 склеивается с вершинами границы $B_1 \sqcup B_2$, по-прежнему у одного из графов существует такая часть границы, все вершины которой отождествляются с вершинами другого графа. Этот граф будем называть

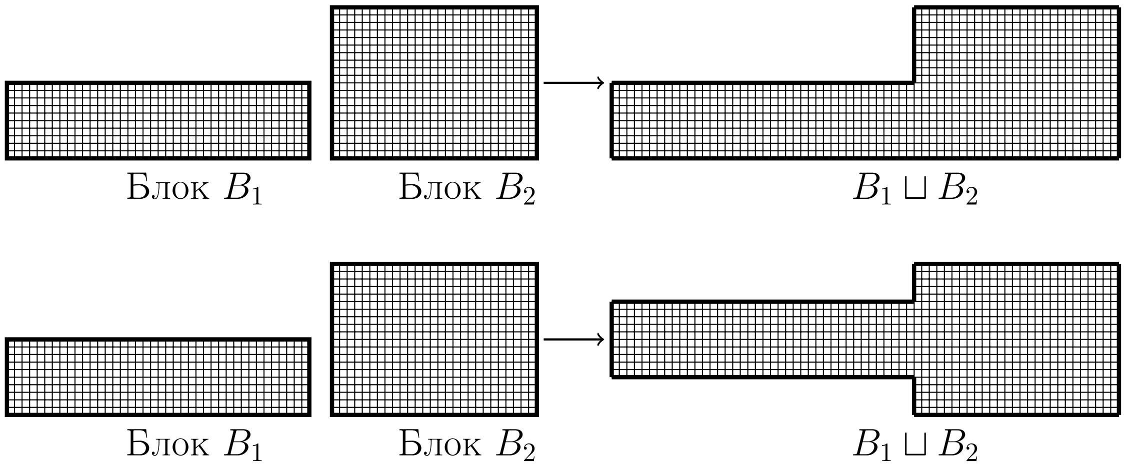

Рис. 4

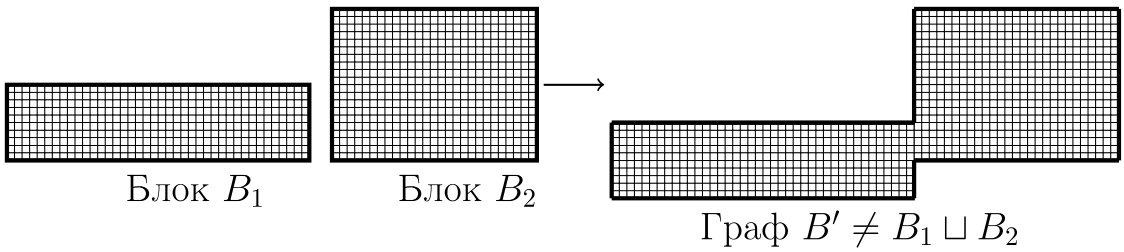

Рис. 5

результатом склейки трёх блоков. *Границей* полученного графа будет множество $\Gamma = \Gamma_{12} \cup \Gamma_3 \setminus \gamma$, где Γ_{12} — граница $B_1 \sqcup B_2$, Γ_3 — граница B_3, а γ— множество рёбер между «склеивающимися» вершинами границ. Склейка трёх блоков проиллюстрирована на рисунке 6. Аналогично определяется склейка k блоков и граница полученного графа, которая естественным образом делится на части. Графы, получающиеся в результате склейки, будем называть *графами блочной структуры.*

Назовем стратегию очищения графа $G = B_1 \sqcup B_1 \sqcup \ldots \sqcup B_k$ блочной структуры *поблочной*, если на каждом шаге существует не более одного блока B_i, где $i \in 1 : k$, все рёбра которого, за исключением рёбер границы, не являются одновременно ни очищенными, ни загрязненными, $bs(G)$ — минимальное количество игроков, для которого существует поблочная стратегия, очищающая граф G блочной структуры. Условия очищения рёбер для блочного поиска совпадают с условиями смешанного поиска. Поблочную стратегию очищения графа $G = B_1 \sqcup B_2 \sqcup \ldots \sqcup B_m$ блочной структуры назовём *поблочно-связной*, если на каждом шаге множество очищенных ребер связно, а поисковое число поблочно-связного поиска — $cbs(G)$. Заметим, что тогда $mixs(G) \leqslant bs(G) \leqslant cbs(G)$.

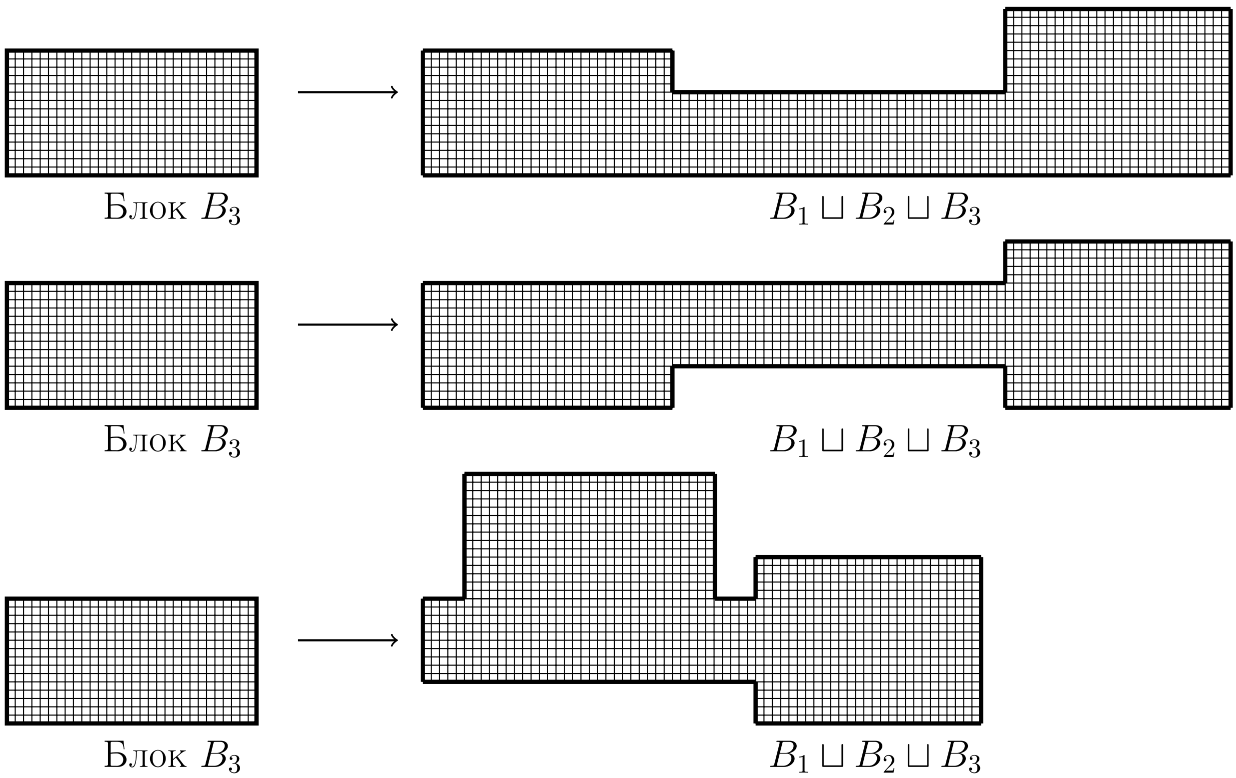

Рис. 6

3. Горизонтальная склейка блоков по одной из частей границы

Понятно, что склейка блоков определяется не единственным образом. Рассмотрим графы, которые могли получиться в результате склейки блоков, удовлетворяющие следующим условиям на каждом шаге склейки:

1. блоки «приклеиваются» по правой или левой границе;
2. после каждого шага полученный граф $B_1 \sqcup \ldots \sqcup B_k$, B_i размера $m_i \times$

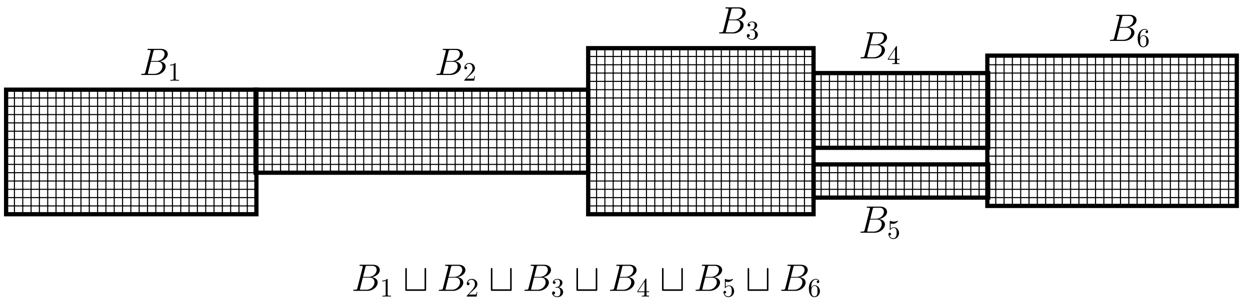

Рис. 7. Пример горизонтальной склейки, удовлетворяющей условиям 1–3

n_i, $m_i \leqslant n_i$ где $i \in 1:k$ покрывается блоком B размера $\max\limits_{i\in 1:k} m_i$, такой блок будем называть *минимальным покрывающим*;

3. результат склейки нельзя получить склейкой меньшего числа блоков, удовлетворяющей условичм 1–2.

Для таких графов блочной структуры верно следующее утверждение:

Утверждение 3.1. *Пусть $G = B_1 \sqcup \ldots \sqcup B_k$, где B_1 размера $m_1 \times n_1, \ldots, B_k$ — $m_k \times n_k$ – граф блочной структуры, удовлетворяющим условиям 1–3, где $m_i \leqslant n_i$, $i \in 1:k$, тогда $s(G) = \max\limits_{i\in 1:k} m_i + 1$, $bs(G) = mixs(G) = \max\limits_{i\in 1:k} m_i$.*

Доказательство. Известно, что $s(G) = ms(G)$ и если стратегия A очищает граф G с использованием k игроков, то для любого G_0 — подграфа G — сужение стратегии A на G_0 заведомо очищает G_0. Таким образом, получаем, что $s(G) \geqslant s(G_0)$ для любого G_0 — подграфа G.

Для графа G блочной структуры получаем, что если $G = B_1 \sqcup \ldots \sqcup B_k$, то $s(G) \geqslant \max\{s(B_1), \ldots, s(B_k)\}$.

Покажем, что для некоторого блока B размера $m \times n$ выполняется равенство $s(B) = m + 1$. Для этого рассмотрим оптимальную стратегию и первый очистившийся столбец вершин $v_1, \ldots, v_m$ графа B. Все рёбра между вершинами этого столбца являются очищенными. Ребро $e = (v_i, v_{i+1})$

очищено, следовательно, либо в вершине v_i стоит игрок, либо все ребра, инцидентные v_i очищены.

В случае выполнения второго варианта получаем, что в i–ой строке вершин блока B есть очищенное ребро. Если эта строка не полностью очищена, в ней есть игрок. Если же строка полностью очищена, то в каждом столбце стоит игрок, откуда получается, что $s(B) \geqslant n$, но $m+1 < n$, иначе рассматриваемый очищенный столбец не является первым. Итак, $s(B) \geqslant m$. Рассмотрим шаг, когда очищалось последнее загрязнённое ребро e в столбце. Заметим, по-прежнему верно, что в любой строке стоит игрок, при этом на последнем шаге игрок скользил вдоль ребра e. Значит, в одной из строк на предпоследнем шаге стояло 2 игрока. Тогда $s(B) \geqslant m+1$.

Приведем пример стратегии, очищающей блок B с использованием $m+1$ игрока. Для этого пронумеруем столбцы блока от 1 до n, а строки — от 1 до m, а вершину, находящуюся в i–ой строке и j–ом столбце, обозначим $v_{i,j}$. Формально опишем стратегию очищения блока *слева направо*. Пусть S_i, где $i \in 1 : m+1$ — игроки, участвующие в очищении блока.

for $i = 1, \ldots, m$ **do**

  поставить S_i в $v_{i,1}$;

  S_{m+1} скользить вдоль ребра $e = (v_{i,1}, v_{i+1,1})$;

$\{$первый столбец очищен$\}$

$j = 2$;

while $j \leqslant n$ **do**

  S_{m+1} поставить в вершину $v_{1,j}$;

  for $i = 1, \ldots, m$ **do**

    S_i скользит вдоль ребра $e = (v_{i,j-1}, v_{i,j})$;

    S_{m+1} скользить вдоль ребра $e = (v_{i,j}, v_{i+1,j})$;

  $\{j$–ый столбец очищен$\}$

  $j++$;

Видно, что такая стратегия очищает блок B. Аналогично, для смешанного поиска получаем $mixs(B) = m$, примером будет следующая стратегия:

for $i = 1, \ldots, m$ **do**

поставить S_i в $v_{i,1}$;

$\{$первый столбец очищен$\}$

$j = 2$;

while $j \leqslant n$ **do**

for $i = 1, \ldots, m$ **do**

S_i скользит вдоль ребра $e = (v_{i,j-1}, v_{i,j})$;

$\{j$–ый столбец очищен$\}$

$j + +$;

Аналогичным образом определяется стратегия очищения *справа налево.* Теперь покажем, как с использованием $m + 1$ игрока очистить граф G блочной структуры, где m — высота минимального покрывающего покрытия. Рассмотрим блок, левая граница которого пересекается с левой границей графа G (если таких блоков несколько, то очистим их поочередно описанным способом). Заметим, что мы сможем это сделать, так как $mixs(G) = m$. Когда эти блоки очищены, все игроки в последнем столбце каждого блока. Заметим, что они все стоят в одном ряду по определению графа блочной структуры. Дальше продолжается тот же процесс на блоках, имеющих общую границу с очищенными блоками, на каждом шаге выставлено не более, чем m, а во время очищения блока шириной m выставлен m игрок, стратегия является поблочной. Следовательно, $bs(G) = mixs(G)$. Итак, утверждение доказано.

4. Поисковые числа графов, получающихся в результате склейки двух блоков

Пусть склеиваются два блока B_1, B_2. Найдём поисковые числа для всех способов склейки и предъявим оптимальные стратегии. Выделим два класса полученных графов с точностью до поворота.

4.1. Склейка двух блоков с отождествлением угловых вершин

Начнём со склейки блоков, в результате которой хотя бы одна пара угловых вершин отождествляется.

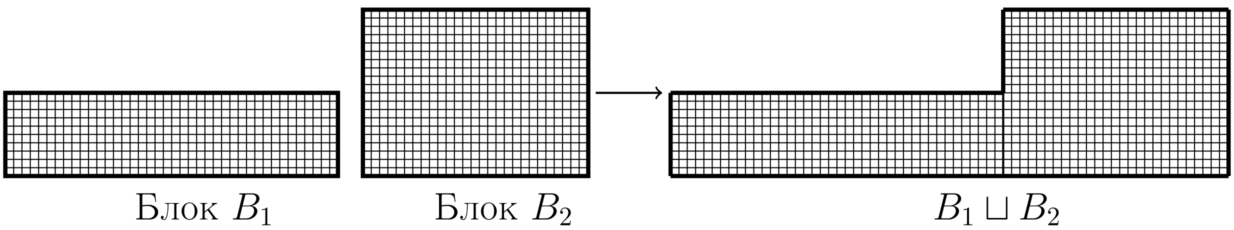

Рис. 8

Полученный граф назовём B и будем искать для него $mixs(B)$, $cmixs(B)$, $bs(B)$, $cbs(B)$.

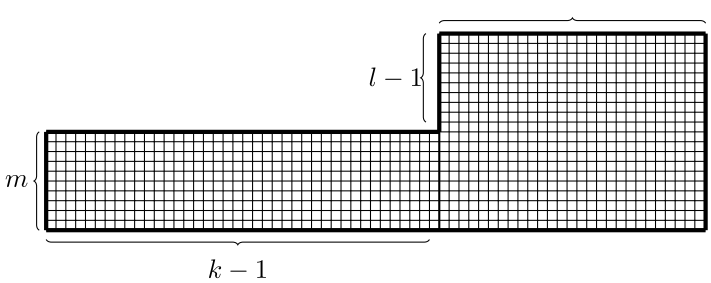

Рис. 9. Граф B

Начнём исследования поисковых чисел с нахождения $mixs(G)$. Имеют место следующие неравенства:

$$mixs(B) \geqslant \min\{\min\{m+l-1, m+n\}, \max\{m, n\}, m+l-1, n+k-1\}$$

$$mixs(B) \geqslant \min\{\max\{m, n\}, m + l - 1, n + k - 1\}$$

Для удобства чтения доказательства и вывод неравенств приводятся в приложении А к данной работе. Обобщая полученные неравенства, видим окончательную оценку

$$mixs(B) \geqslant \min\{\max\{m, n\}, m + l - 1, n + k - 1\},$$

которую можно переписать в следующем виде

$$mixs(B) \geqslant \min\{m, n+k-1\},\ m \geqslant n,\ mixs(B) \geqslant \min\{n, m+l-1\},\ n > m$$

Предъявим стратегии, при которых неравенство превращается в равенство. Для этого пронумеруем столбцы блока от 1 до n, а строки — от 1 до m, а вершину, находящуюся в i–ой строке и j–ом столбце, обозначим $v_{i,j}$:

- $m \geqslant n, m \geqslant n + k - 1$, следовательно $mixs(B) = n + k - 1$. Оптимальной стратегий тогда будет являться стратегия «снизу вверх», когда все игроки ставятся на вершины самой нижней строки, а затем по очереди скользят по рёбрам вверх, последовательно очищая строки. Описанная стратегия будет ещё и блочной, если рассмотреть полученный граф как склейку двух блоков размеров $m \times (n + k - 1)$ и $l \times n$.

- $m \geqslant n, m \leqslant n + k - 1$, следовательно $mixs(B) = m$. Оптимальной будет следующая стратегия:

  – очищаются столбцы высотой m «слева направо» или «справа налево» в зависимости от положения границы графа, мы будем считать, что используется стратегия «слева направо», после чего скользят по рёбрам до столбца k;

  – затем все игроки, кроме игрока в вершине $v_{m,k}$, поочередно скользят по рёбрам;

– теперь игрок в вершине $v_{m-1,k+1}$ тоже стоит на месте, а другие скользят по рёбрам своей строки;

– продолжая таким образом, получаем игрока в вершине $v_{1,n+k-1}$;

– игроки поочерёдно скользят вверх по рёбрам своих столбцов и очищают граф;

Такую стратегию назовём *диагональной*. Диагональная стратегия является поблочной для такого класса графов.

- $n > m, n \geqslant m+l-1$, следовательно $mixs(B) = m+l-1$. Оптимальной стратегией будет поблочная стратегия «слева направо», когда игроки последовательно очищают столбцы, двигаясь при этом по строкам.

- $n > m, n \leqslant m+l-1$, следовательно $mixs(B) = n$. Стратегия, аналогичная случаю $m \geqslant n, m \leqslant n+k-1$.

Все стратегии получились связными и поблочными, так что мы нашли и $cmixs(B)$, $bs(B)$, $cbs(B)$ для графов B из этого класса. Таким образом, доказано утверждение

Утверждение 4.1. *Для любого графа B, полученного в результате склейки двух блоков B_1 и B_2 размеров $m \times k$ и $(m+l-1) \times n$, $mixs(B) = cmixs(B) = bs(B) = cbs(B) = \min\{m, n+k-1\}$, $m \geqslant n$ и $mixs(B) = cmixs(B) = bs(B) = cbs(B) = \min\{n, m+l-1\}$, $n > m$*

4.2. Склейка двух блоков без отождествления угловых вершин

Теперь рассмотрим ещё один класс графов, которые могут получаться в результате склейки двух блоков.

Снова начнём исследования поисковых чисел с нахождения $mixs(G)$. Имеют место следующее неравенство:

$$mixs(B) \geqslant \min\{\max\{2m, n\}, \max\{m, n+k-1\}, n+p+k-2, m+l-1, m+n\}$$

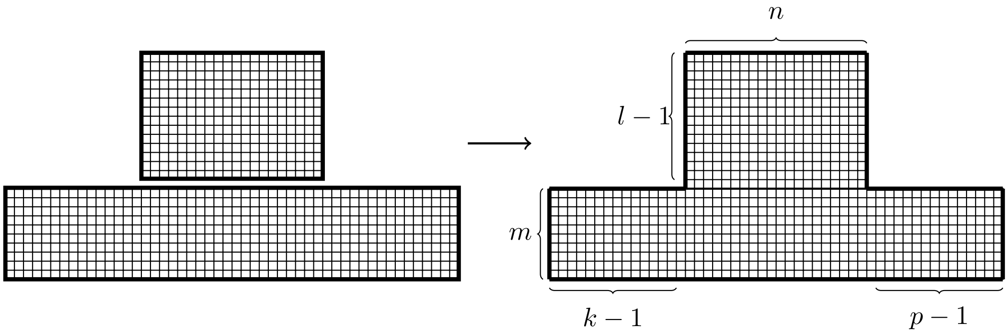

Рис. 10

Подробный вывод неравенства приводится в приложении B к данной работе. Теперь построим стратегии для каждого возможного значения минимума.

- $\min\{n+m, \max\{n+k-1, m\}, n+p+k-2, m+l-1, \max\{2m, n\}\} = n+m$

  Стратегия следующая: очистить $k-1$ подряд идущих столбцов «слева направо» с использованием m игроков, затем поставить n игроков в вершины $v_{m,k} \dots v_{m,n+k-1}$, продолжить очищение m игроками «слева направо», после чего n игроков очищают остаток графа «снизу вверх». Стратегия получилось поблочной;

- $\min\{n+m, \max\{n+k-1, m\}, n+p+k-2, m+l-1, \max\{2m, n\}\} = n+p+k-2$ — стратегия «снизу вверх», является связной и блочной;

- $\min\{n+m, \max\{n+k-1, m\}, n+p+k-2, m+l-1, \max\{2m, n\}\} = m+l-1$ — стратегия «справа налево». Если рассматривать граф как результат склейки двух блоков, то такая стратегия не будет поблочной, но будет таковой, если рассматривать граф, полученный в результате склейки трёх блоков размеров $m \times k$, $(m+l-1) \times n$, $m \times p$;

- $\min\{n+m, \max\{n+k-1, m\}, n+p+k-2, m+l-1, \max\{2m, n\}\} = \max\{n+k-1, m\}$

  В этом случае стратегия строится так:

- n игроков очищают «длинные» столбцы «сверху вниз», пока не окажутся в вершинах $v_{m,k} \dots v_{m,k+n-1}$;
- $k-1$ игроков занимают вершины $v_{m,1} \dots v_{m,k-1}$
- Затем игроки становятся в диагональ: игрок в вершине $v_{m,k+n-1}$ остаётся на месте, остальные скользят вниз по рёбрам своих столбцов;
- Теперь игрок в вершине $v_{m-1,k+n-2}$ тоже стоит на месте, а остальные скользят вниз по рёбрам, то есть на каждом шаге в строке остаётся игрок, у которого справа в строке нет игроков;
- Когда в каждой строке с номером, меньшим $m+1$ есть игрок, игроки последовательно скользят по рёбрам своих строк.

Стратегия оказалась поблочной.

- $\min\{n+m, \max\{n+k-1, m\}, n+p+k-2, m+l-1, \max\{2m, n\}\} = \max\{2m, n\}$

Полученная в этом случае стратегия тоже будет поблочной.

- «Сверху вниз» очищаются короткие строки по столбцам n игроками;
- Затем в строке m выставляем оставшихся игроков, если $2m \geqslant n$;
- Крайние игроки с двух сторон остаются на месте, остальные последовательно скользят по рёбрам столбцов;
- Повторяем предыдущий шаг, пока не дойдём до первой строки;
- Так как игроков не меньше $2m$, то в каждой «длинной» строке стоит хотя бы по два игрока, все рёбра между которыми очищены, так что теперь каждый игрок скользит по рёбрам строки в сторону ближней границы графа.

В итоге доказали следующее утверждение:

Утверждение 4.2. *Для графа B, являющегося результатом склейки двух блоков размеров $m \times (n+p+k-2)$ и $l \times n$ без отождествления вершин, верно соотношение* $mixs(B) = bs(B) = cmixs(B) = cbs(B) = \min\{m+l-1,\ \max\{n, 2m\}m+n,\ \max\{m, n+k-1\},\ n+k+p-2\}$

Действительно, минимум может принимать любое из перечисленных значений, для каждого из них в таблице приведён пример на $m,\ n,\ p,\ k,\ l$.

	$m+l-1$	$\max\{n, 2m\}$	$m+n$	$\max\{m, n+k-1\}$	$n+k+p-2$
m	5	4	5	5	10
n	6	7	4	5	3
p	5	5	9	6	3
k	4	3	8	3	4
l	3	5	6	4	10

5. Блочный граф «крест»

Одним из самых распространённых приёмов для нахождения поисковых чисел является операция взятия минора графа. В том случае, если поиск обладает свойством замкнутости класса графов, поисковое число которых меньше k, относительно взятия минора, это позволяет найти нижние оценки для поискового числа. Сейчас покажем, как результаты предыдущей главы помогают найти поисковые числа графа следующего вида: к каждой из частей границ блока $m \times m$ приклеиваются блоки $m \times k$, $m \times l$, $m \times p$, $m \times q$, что изображено на рисунке 11.

У полученного графа G выделим четыре минора, каждый из которых получен склейкой блока $m \times m$ с тремя из блоков $m \times k$, $m \times l$, $m \times p$, $m \times q$, а затем воспользуемся тем, что мы знаем поисковые числа для блочных графов такого вида. Получаем, что при $k \leqslant p,\ l \leqslant q$

$$mixs(G) \geqslant \max\{M_1,\ M_2,\ M_3,\ M_4\}$$

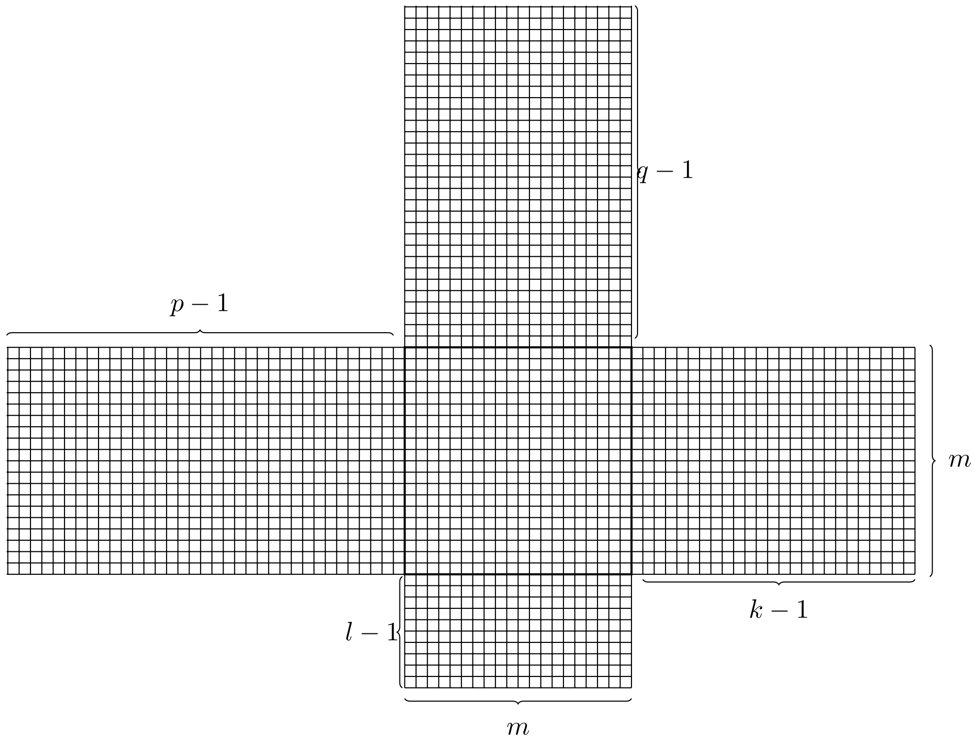

Рис. 11. «Граф-крест»

$$M_1 = \min\{2m, m+l-1, m+k-1\},$$

$$M_2 = \min\{2m, m+q-1, m+k-1\},$$

$$M_3 = \min\{2m, m+q-1, m+p-1\},$$

$$M_4 = \min\{2m, m+l-1, m+p-1\}.$$

Пусть $l \leqslant k$, тогда $M_1 = M_4$ и

$$mixs(G) \geqslant \max\{M_1,\ M_2,\ M_3\}$$

$$M_1 = \min\{2m, m+l-1\},$$

$$M_2 = \min\{2m, m+q-1, m+k-1\},$$

$$M_3 = \min\{2m, m+q-1, m+p-1\}.$$

Теперь заметим, что $\max\{M_1,\ M_2,\ M_3\} \neq m+l-1$, так как в этом случае $m+l-1 \leqslant 2m$, но тогда $M_2, M_4 \geqslant m+l-1$, потому что $l = \min\{l, k, p, q\}$. Таким образом, всего три варианта, для каждого из который существует стратегия, причём связная. Снова нумеруем строки сверху вниз, столбы слева направо, соответственно вершина $v_{i,j}$ находится на пересечении i–ой строки и j–го столбца.

1. $\max\{M_1,\ M_2,\ M_3\} = 2m$. Тогда с помощью $2m$ игроков можно очистить следующим образом:

   - m игроков $S_1, \ldots, S_m$ очищают «снизу вверх» блок $m \times l$, и стоят в вершинах $v_{m+q-1,p-1}, \ldots, v_{m+q-1,m+p-1}$;
   - оставшиеся m игроков $S_{m+1}, \ldots, S_{2m}$ ставятся в вершины $v_{m+q-1,m+p-1}$, $\ldots$, $v_{q-1,m+p-1}$;
   - игроки $S_1, \ldots, S_m$ скользят вдоль рёбер столбца и выстраиваются по диагонали в вершинах $v_{p-1,m+p-1}$, $\ldots$, $v_{q-1,m+p-1}$;
   - игроки $S_{m+1}, \ldots, S_{2m}$ скользят по строкам и очищают блок $m \times (k-1)$ «слева направо»;
   - игроки $S_{m+1}, \ldots, S_{2m}$ ставятся в вершины $v_{m+q-1,p}, \ldots, v_{q-1,p}$;
   - игроки $S_1, \ldots, S_m$ скользят вверх по рёбрам столбца и очищают столбцы из $m+q+l-2$ вершин;
   - игроки $S_{m+1}, \ldots, S_{2m}$ скользят по строкам и очищают блок $m \times (p-1)$ «справо налево»;

2. $\max\{M_1,\ M_2,\ M_3\} = m+q-1$, тогда $m+q-1$ игроков смогут очистить граф так:

   - m игроков $S_1, \ldots, S_m$ очищают «слева направо» блок $m \times k$, и стоят в вершинах $v_{m+q-1,m+p-1}$,$\ldots$, $v_{q-1,m+p-1}$;
   - оставшиеся $q-1$ игроков $S_{m+1}, \ldots, S_{m+q-1}$ ставятся в вершины $v_{m+q-1,m+p-1}$, $\ldots$, $v_{1,m+p-1}$;
   - игроки $S_1, \ldots, S_m$ скользят вдоль рёбер столбца и выстраиваются по диагонали в вершинах $v_{q,p}$, $\ldots$, $v_{m+q-1,m+p-1}$, игроки S_{m+1},

$\ldots, S_{2m}$ скользят по строкам и очищают блок $m\times(q-1)$ «справа налево»;

- игроки $S_{m+1}, \ldots, S_{2m}$ становятся в вершины $v_{m+q+p-2,m+p-1}, \ldots, v_{m+q-1,m+p-1}$;
- все игроки последовательно скользят по рёбрам по своих строк «справа налево» и очищают граф.

3. $\max\{M_1,\ M_2,\ M_3\} = m+p-1$, тогда $m+p-1$ игроков смогут очистить аналогично пункту 2;
4. $\max\{M_1,\ M_2,\ M_3\} = m+k-1$, тогда $m+k-1$ игроков смогут очистить аналогично пункту 2, начиная с очищения «сверху вниз» блока $(q-1)\times m$.

Заметим, что приведённые стратегии получились связными и блочными. Таким образом, получим, что

$$mixs(G) = cmixs(G) = bs(G) = cbs(G) = \max\{M_1,\ M_2,\ M_3\}$$

$$M_1 = \min\{2m, m+l-1\},$$

$$M_2 = \min\{2m, m+q-1, m+k-1\},$$

$$M_3 = \min\{2m, m+q-1, m+p-1\}.$$

6. Связный поиск

Докажем утверждение, показывающее одну из основных особенностей связного поиска, сформулированную в [7]. Мы приведём строгое доказательство следующего факта.

Утверждение 6.1. *Существует класс графов, для которых* связный смешанный поиск *не замкнут относительно взятия минора, то есть существует пара (G, H), где G — минор H и $cmixs(G) > cmixs(H)$.*

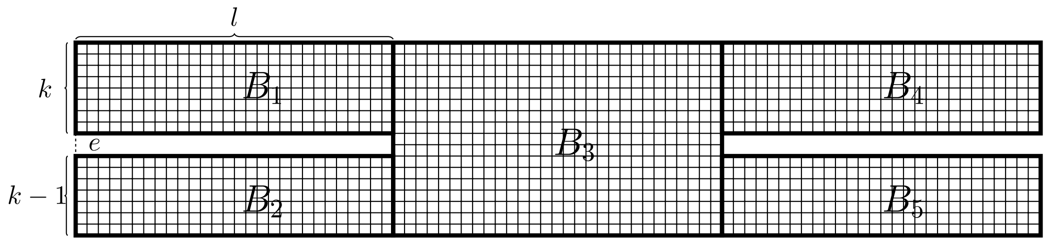

Рис. 12. Граф H

Доказательство. Граф $H = G \cup \{e\}$ (рисунок 12), $mixs(H) = 2k = cmixs(H)$ (оптимальная стратегия для смешанного поиска «слева направо» является связной). Блоки графа G назовём B_1, B_2, B_3, B_4, B_5, B_3 имеет общие вершины с B_1, B_2, B_4, B_5, B_1 размера $k \times l$, B_2 — $(k-1) \times l$, B_3 — $2k \times l$, B_4 — $k \times l$, B_5 — $(k-1) \times l$, $l \gg k > 3$. Если в оптимальной стратегии граф очищается поблочно-связно, то возможны следующие варианты:

- Первым очищается B_3, после чего B_3 защищен на шаге после очищения последнего ребра блока. Значит, на всех вершинах блока, которым инцидентны ребра другого блока, поставлены игроки. Таким образом, $cbs(G) \geqslant 4k - 2$.

- Первым очищается B_1. Тогда в силу требования связности затем очищается B_3. Рассуждая, аналогично предыдущему пункту, получаем, что $cbs(G) \geqslant 3k - 2$.

- Первым очищается B_2. Тогда в силу требования связности затем очищается B_3. Рассуждая, аналогично предыдущему пункту, получаем, что $cbs(G) \geqslant 3k - 1$.

В силу симметрии графа получаем так же оценки, когда первым очищается блок B_4 или B_5, то есть в любом случае $cbs(G) \geqslant 3k - 2 > 2k$.

Введем «систему координат»: присвоим всем столбцам вершин номера от 1 до $3l$ слева направо, а всем строкам — номера от 1 до $2k$ сверху вниз, тогда $v_{i,j} = (i, j)$. Нас интересует связный смешанный поиск.

Существует стратегия, для которой требуется $3k-1$ игрок, то есть $cmisx(G) \leqslant 3k-1$: очищается блок B_1 слева направо, после чего k игроков стоят в вершинах $v_{i,l},\ l \in 1:k$. Теперь поставим еще $2k-1$ игрока таким образом: одного игрока в вершину $v_{k+1,l}$, по два в $v_{i,l},\ l \in k+2:2k$. Теперь по одному игроку из вершин $v_{i,l},\ l \in k+2:2k$ очищают блок B_2 справа налево, после чего игроки из l–го столбца очищают блок B_3 «слева направо», а затем блоки $B_4,\ B_5$.

Для графа блочной структуры можно определить *ячейку* – полный двудольный подграф, порожденный четырьмя вершинами, по две в каждой доле. Заметим, что если в ячейке очищены два противоположных ребра, то и оставшиеся два ребра тоже очищены. Пусть это не так: среди этих ребер есть грязное $e=(u,v)$. Тогда в u,v стоят игроки. Но в таком случае по определению смешанного поиска ребро очищено, противоречие. Из этого следует, что если в блоке все строки очищены, то блок полностью очищен.

Итак, пусть первым в **оптимальной стратегии** очищается B_3. Тогда рассмотрим, для примера, блок B_1 и оценим, сколько в нем игроков. Так как B_3 чист, то для любой строки блока B_1 верно, что либо все ребра строки очищены, либо в строке стоит хотя бы один игрок. Если для всех строк выполняется второй вариант, то получаем оценку снизу на количество игроков $k = mixs(B_1)$. Покажем, что в первом варианте игроков потребуется больше.

Пусть есть полностью очищенная строка i. Тогда уже для любого столбца верно, что либо все его ребра очищены, либо в нем есть хотя бы один игрок. Если для всех столбцов выполняется второй вариант, то получаем оценку снизу на количество игроков $l >> k$, то есть при достаточно большом k, например, $k = 10l$, подобная стратегия не может быть оптимальной. Таким образом, существует столбец j, все ребра которого очищены (в столбце нет игрока). Рассмотрим его. Так как ни в какой вершине $v_{i,j}, i \in 1:k$ не стоит игрок, то для неё верно, что все инцидентные ей ребра очищены. Получаем полностью очищенный столбец ячеек (у ячейки есть пара очищенных ребер). Теперь рассмотрим строки с номерами от 1 до k и столбцы с номерами от 1 до $j-1$ (считаем, что $j-1>1$). Рассмотрим такой алгоритм:

$p = 1, q = j - 1;$

while $p < k + 1$ **do**

  while $q > 0$ **do**

    if $e = (v_{p,q}, v_{p,q-1})$ очищено **then**

      $p++$; {опускаемся на строку вниз}

    else

      S скользит вдоль $e = (v_{p,q}, v_{p,q-1})$; $p++$; {в вершине $v_{p,q-1}$ есть игрок S, а все инцидентные ей ребра, кроме e, уже очищены};

  $q--$; {просматриваем следующий столбец}

{$e = (v_{p,q}, v_{p+1,q})$ очищено как ребро в ячейке парой очищенных противоположных ребер в обоих случаях}

Заметим, что в результате работы алгоритма мы очистим все вершины $v_{p,q}, p \in 1 : i-1, q \in 1 : k$. Аналогично очищаем оставшиеся вершины. Мы показали, как в таком случае с помощью игроков, находящихся в вершинах блока B_1, преобразовать стратегию так, чтобы в случае наличия очищенной строки и очищенного столбца можно очистить весь блок до очищения блока B_3, что противоречит предположению.

Таким образом, получаем, что для каждого $B_j,\ j = 1, 2, 4, 5$ верно, что в нем находится не меньше, чем $mixs(B_j)$ игроков. Тогда $cmix(G) = 4k - 2 > 3k - 1$.

Итак, получили, что для любой стратегии, использующей m игроков, где первым очищается блок B_3, существует стратегия, очищающая граф не более чем m игроками, в которой B_3 не является первым очищенным блоком.

Пусть тогда первым очищается B_1. Рассмотрим момент, когда в блоке B_1 есть очищенное множество и начинает очищаться B_4 или B_5. В таком случае существует путь из B_1 в один из этих блоков. Этот путь целиком лежит в B_3 и проходит по всем столбцам ячеек. Тогда для каждого столбца верно, что либо в нем есть игрок, либо он полностью очищен. Если для всех верно первое, то уже есть l игроков, что очень много. Значит, есть очищенный столбец, в котором нет игрока. Тогда повторяем алгоритм, описанный выше, и получаем очищенный B_3. Доказали, что в оптимальной стратегии как только начинается очищение блоков B_4, B_5, блок B_3 очищен.

Рассматриваем шаг, когда в B_3 впервые появился очищенный столбец j. Уже знаем, что в этот момент очищенные множества располагаются в блоках по одну сторону от B_3.

Случай 1 Очищенное множество есть только в блоке B_3, в остальных блоках, возможно, лишь очищенная граница B_3. Тогда в каждой строке есть хотя бы один игрок для защиты от загрязнения на данном шаге, за исключением строки с номером $k+1$. Если там нет игрока, то она является полностью очищенной, откуда получаем в силу выбора столбца необходимость $l-1 \gg 3k-1$ игрока. Значит, в блоке хотя бы $2k$ игроков. Теперь отметим, что загрязненные ребра есть с обеих сторон от очищенного столбца, значит, если в строке очищено хотя бы 1 ребро, то необходимо строго больше $2k$ игроков. Следовательно, в строке чистых ребер нет, то есть в блоке B_3 очищен только один столбец, причем на каждой вершине столбца стоит игрок. На следующем шаге либо появляется новый игрок, откуда $cmixs(H) < cmixs(G)$, либо переставляется один из игроков. Из-за связности мы можем удалять только игроков из крайних ячеек, но мы могли получить такую конфигурацию последовательным выставлением игроков, то есть очищение столбца было лишним шагом, а стратегию, в которой достигается этот случай, можно преобразовать таким образом, что к моменту такой расстановки игроков и расположения очищенных ребёр ещё не встречался шаг, на котором бы был очищенный столбец.

Случай 2 Есть очищенный связный подграф W хотя бы в одном из блоков, пусть B_1, причем W не является границей с B_3. Аналогично предыдущему случаю получаем, что в блоке хотя бы $2k$ игроков. Так как B_4, B_5 загрязнены, то все $2k$ игроков находятся не левее очищенного столбца. Покажем, что $j = l$, то есть первый очищенный столбец блока B_3 — левая граница B_3. Пусть не так, тогда от W существует очищенный путь до столбца j, проходящий через все столбцы B_3 с номерами, меньшими j, а j — первый очищенный столбец, то есть во всех столбцах с меньшими номерами есть неочищенные ребра, следовательно, в них есть игроки, таким образом, $cmixs(G) > 2k$.

1. B_1 очищен не полностью. Тогда есть строки, в которых есть и очищенное, и загрязненное рёбра, то есть там заведомо есть игрок. Если их нет, то все строки либо полностью очищены, а значит и блок очищен, либо есть полностью загрязненная строка, а значит для защиты W есть игрок внутри блока и $cmixs(G) > 2k$.

2. B_1 полностью очищен, в B_2 есть очищенный связный подграф, описанный выше. Аналогично предыдущему подслучаю получаем, что $cmixs(G) > 2k$.

3. B_1, B_2 полностью очищены. Рассмотрим шаг, когда впервые есть очищенные ребра в строках 1 и $2k$. Из-за связности на этом шаге заведомо выставлено $2k$ игроков, по одному в каждой строке. Покажем, что есть строка, в которой более одного игрока. Видно, что на каждом шаге в строке очищается не более двух ребер: скольжение по ребру увеличивает количество очищенных ребер в строке максимум на 1, а постановка игрока — на два. Пусть на данном шаге очисти-

лось ребро e в B_1 (если очистилось 2 ребра, рассмотрим самое левое). Если $e = (v_{1,i}, v_{1,i+1}), i \neq 1$, то в строке два игрока для защиты от загрязнения на данном шаге, а всего не менее $2k+1$. Иначе, чтобы выйти на границу блока B_1 и попасть в B_3 необходимо пройти через $l-2 > 3k-1$ столбца, в каждом из которых есть игрок, так как все ребра $e = (v_{1,i}, v_{1,i+1}), i \neq 1, 2$, загрязнены.

4. B_1 полностью очищен, в B_2 очищен лишь столбец l. Тогда в вершинах $v_{l,i}, i \in k+2, \ldots, 2k$ стоят игроки. Пусть больше не требуется новых игроков. Для строк с номерами, меньшими $k+2$, верно следующее: если в вершине $v_{i,y},\ y \in l : 2l-1$ стоит игрок, то в $(i-1)$–ой строке в блоке B_3 игрок стоит в вершине $v_{i-1,z}$, где $z \leqslant y+1$. Для $i = k+1$ это очевидно, а дальше это верно, так как в столбцах с номерами, большими $y+1$ еще нет игроков. Если новых игроков не требуется, то

   - один из игроков из вершин $v_{l,i}, i \in k+2, \ldots, 2k$ покидает вершину, а из-за связности это могут сделать только игроки с двух нижних вершин, и столбец перестает быть полностью очищенным. Тогда такое же положение игроков и очищенных ребер можно получить, не очищая столбца: очистим B_1 «слева направо», выставим нужное количество игроков (как в оптимальной стратегии после ухода игрока из вершины стобца) в l–ый столбец, а затем последовательно проскользим каждым игроком вдоль ребра до нужной позиции, после чего поставим удаленного игрока на место, которое ему соответствовало на следующем шаге исходной стратегии;
   - перемещается игрок из вершин $v_{l,i}$, где $i \in 1, \ldots, k+1$, при этом по доказанному выше в i–ой строке игрок стоит не дальше $(l+k+2-i)$–го столбца. Значит, в какой-то момент получим,

что новых ребер больше не очистить. Если на следующем шаге покидает вершину игрок из B_2, то верно предыдущее рассуждение. Иначе либо игрок скользит влево по ребру строки, и мы можем преобразовать стратегию описанным выше способом (уберем ненужные шаги), либо загрязняется блок B_1 и очищенный столбец, так как игрок больше не защищает строку от загрязненных столбцов, а значит, мы можем преобразовать стратегию, последовательно выставляя оставшихся игроков в l–ом столбце.

То есть, если не добавлять новых игроков, то стратегия преобразуется таким образом, что получается та же конфигурация игроков и очищенных ребер, но при этом еще нет очищенного столбца. Тогда продолжаем дальше преобразованную стратегию и ждем следующего появления очищенного столбца в блоке B_3, после чего попадаем в один из случаев и так далее. В итоге, на каком-то шаге будет выставлено строго больше $2k$ игроков. Таким образом, утверждение доказано.

7. Удаление блока

Минором графа G называется граф H, полученный из графа G с помощью последовательного удаления ребёр, вершин и стягивания ребёр. Известно, что для любого минора H графа G верно, что $mixs(H) \leqslant mixs(G)$. С помощью этого свойства найдём поисковые числа следующего класса блочных графов. Рассмотрим блок $m \times m$, из которого удалён блок $k \times k$, где $k < m$ (рисунок 13). Такие графы блочной структуры можно получить в результате склейки минимум четырёх блоков, но в этом случае удобнее рассматривать новую процедуру *удаления блока.*

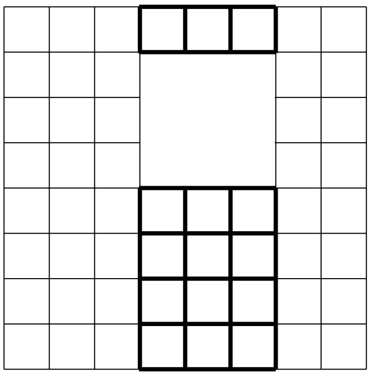

Рис. 13

Удалим в выделенных блоках все вертикальные рёбра, не входящие в границу:

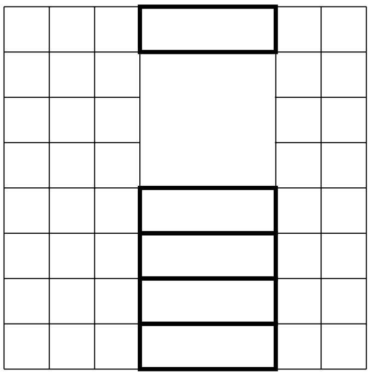

Рис. 14

Будем стягивать выделенные горизонтальные рёбра поэтапно справа налево $k-1$ раз:

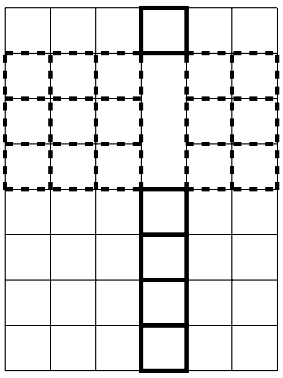

Рис. 15

Проделаем ту же самую процедуру с блоками, выделенными пунктиром и получим:

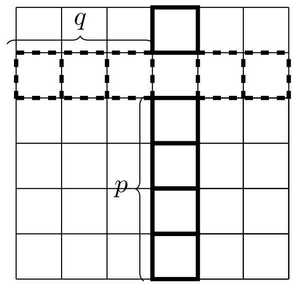

Рис. 16

Для таких графов $mixs(H) = m - k + 2$, то есть совпадает с размером блока. Предъявим стратегию, которая очищает исходный граф с помощью $m - k + 2$ игроков. Снова занумеруем все вершины $v_{i,j},\ i, j \in 1 : m$ слева направа и снизу вверх. Обозначим p и q размеры блоков $k \times q,\ p \times k$, над рёбрами которых и проводились операции удаления и стягивания, причём $p \geqslant m - p - k + 2,\ q \geqslant m - q - k + 2, p \geqslant q$, иначе можем переобозначить. Стратегия будет выглядеть так:

- Поставить по 2 игрока в вершины $v_{q,p+k-2}, \ldots, v_{q,m}$, игроки $S'_1, \ldots, S'_{m-p-k+2}$ во время поиска остаются на этих местах, остальных игроков будем использовать для очищения;

- Последовательно скользить игроками$S_1, \ldots, S_{m-p-k+2}$ вдоль ребёр соответствующих строк до вершин $v_{q,p+k-2} \ldots v_{q+p+k-2-m,m}$, то есть игроки выстраиваются в диагональ;

- При необходимости доставить в вершины вида $v_{i,m}$ игроков так, чтобы в каждом столбце с номерами от 1 до q стояли игроки. Это возможно, так как мы используем $p \geqslant q$ игроков для очищения;

- Снова скользим активными игроками вниз по рёбрам столбца, пока это возможно, выстраивая по завершении игроков по диагонали. При необходимости доставляем игроков и скользим по рёбрам строк и так далее.

Полученная стратегия связная, так что $cmixs(G) = m - k + 2$. Для этого частного случая удаления блока оказалось достаточно удачно взять минор, однако в общем случае нахождение поисковых чисел является более сложной процедурой.

Рассмотрим теперь класс блочных графов, состоящих из блоков размера $m \times m$, в которых удален блок $1 \times k$, где $k < m$. Разобьём полученные графы на два класса: в первом случае удалёный блок расположен горизонтально (рисунок 17), а во втором — вертикально (рисунок 18).

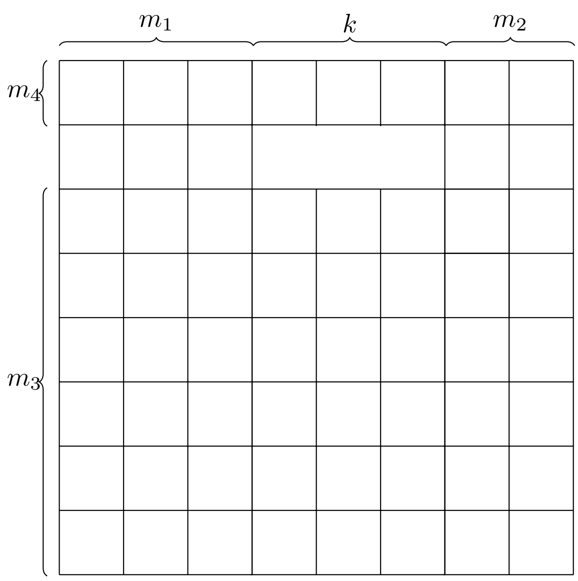

Рис. 17

Пусть $m_3 = \max\limits_{i \in 1:n} m_i$. Обозначим через $a = \min\limits_{i \in 1:n} m_i$. Так же заметим, что $mixs(G) \leqslant mixs(B)$, где B — блок размера $m \times m$. Предположим, что $mixs(G) < m$, а значит в любой момент времени есть столбец и строка без игроков. Зафиксируем шаг, когда появились первый очищенный столбец или строка. Если это столбец, то существует столбец без игроков, в нём нет очищенных рёбер, откуда получаем, что в каждой строке стоит игрок и оценку $mixs(G) \geqslant m$. В случае строки возможны варианты

- Очищенная строка i находится в блоке $m_3 \times m$. По предположению у нас есть строка и столбец без игроков. Заметим, что столбец не может быть ни полностью очищенным, ни полностью загрязнённым, откуда получается, что столбец без игроков j, где $j \in m_1+1 : m_1+k-1$, причём все его рёбра до строки m_3 включительно очищены. Если строка без игроков расположена по одну сторону от удалённого блока с i, то получаем $mixs(G) \geqslant m$. Значит, она находится среди строк с номерами, большими m_3. Теперь смотрим на шаг, когда впервые появилась описанная конструкция столбца, то есть мы можем убрать игрока со столбца j. На предыдущем шаге в столбце j стоял хотя бы один игрок, значит существовал столбец j', в котором не было игроков, при этом все его рёбра до строки m_3 загрязнены, следовательно между столбцами $j,\ j'$ стояли m_3 игроков. Пусть $j' > j$. Если во всех строках блока $m_\times m$ стоят хотя бы два игрока, то $mixs(G) \geqslant 2m_3$. Иначе есть строка $i',\ i' \leqslant m_3$, все рёбра которой до столбца m_1 включительно очищены, а в вершинах $v_{1,i'}, \ldots, v_{m_1,i'}$ нет игроков. Но существует полностью загрязнённая строка, откуда получаем что в первых $m_1 \geqslant a$ столбцах стоят игроки, откуда оценка $mixs(G) \geqslant m_3 + a$. В итоге, $mixs(G) \geqslant \min\{m, m_3 + a\}$.

- Очищенная строка i находится в блоке $m_4 \times m$. Аналогично получаем наличие столбца описанного j с той лишь разницей, что теперь загрязнены вершины столбца в строках, строго меньших $m_3 + 1$, возможно, такой столбец не один. Покажем, что если $mixs(G) \leqslant \min\{m, m_3{+}a\}$, то пока такой столбец существует, в любом столбце из m вершин должен стоять игрок. Рассмотрим первый шаг, на котором снимается последний игрок с одного из столбцов длины m. Получается, в этот момент по одну из сторон от столбца j стоят m_3 столбцов. Так во всех столбцах по другую сторону стоит игрок, то заведомо выставле-

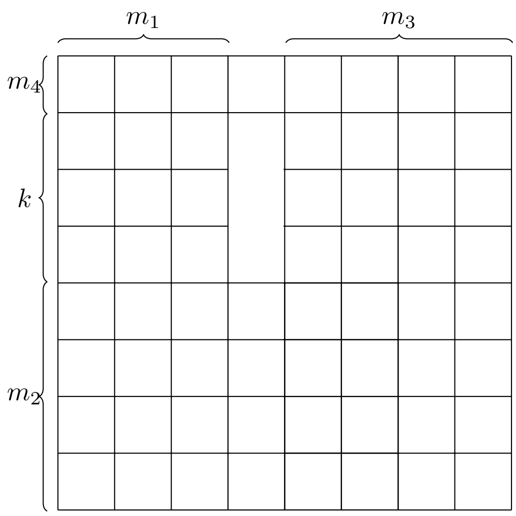

Рис. 18

но $m_3 + a$ игроков.

Теперь покажем, что $mixs(G) = \min\{m, m_3 + a\}$. Для m игроков подойдёт связная стратегия «слева направо», например. Чтобы очистить граф за $m_3 + a$ игроков, воспользуемся стратегией, описанной для квадрата: a игроков стоят на месте в той части графа, где достигается минимум m_i а m_3 очищает граф так же, как и в случае с квадратными удалёнными блоками.

Теперь пусть $m_2 = \max\limits_{i \in 1:n} m_i$. Обозначим через $a = \min\limits_{i \in 1:n} m_i$.

В этом случае вертикального расположения удалённого блока заметим, что блок размера $m_2 + m_4 \times m$ будет минором исходного графа, а так как $(m_2 + m_4) < m$, то $mixs(G) \geqslant m_2 + m_4$. С использованием $m_2 + m_4$ игроков исходный граф можно очистить такой же стратегией, как и в случае, когда $m_3 = \max\limits_{i \in 1:n} m_i$.

Давайте рассмотрим блоки размера $m \times n$, где $m < n$, с удалённым блоком $1 \times k$. Возможно несколько вариантов расположения (рисунки 19, 20), рассмотрим их.

Теперь будем поочередно удалять рёбра между вершинами столбца $m_1 + 1$, стянем рёбра между вершинами v_{i,m_1+1} и v_{i,m_1+2}. Если получен-

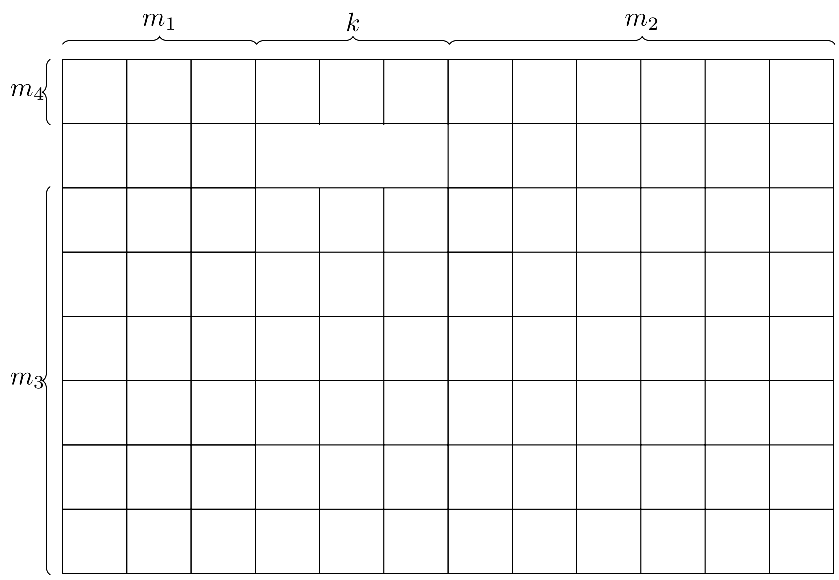

Рис. 19

ный блок B_1 имеет размер $m \times m$, то знаем $mixs(B_1)$, иначе продолжаем процедуру. В итоге либо в какой-то момент получим блок B_l размером $m \times m_1 + m_2$, причём $m_1 + m_2 > m$, а значит $mixs(B_l) = m$, либо получим квадрат с удалённым блоком $1 \times k$, тогда $mixs(B_l) = \min\{m, a + b\}$, где $a = \max\limits_{i \in 1:4} m_i$, $b = \min\limits_{i \in 1:4} m_i$. Таким образом, $mixs(G) \geqslant \min\{m, a + b\}$. На самом деле, $mixs(G) = \min\{m, a + b\}$, стратегии строятся по тому же принципу, что и в случае квадрата.

Пусть $b = m_3$, тогда есть несколько вариантов:

1. $a + m_3 = m$

   В этом случае рассмотрим подграф B исходного графа, B является блоком $m \times m$, из которого удалён блок $1 \times k$, а $mixs(B) = m$, так как $m_3 + a = m$. Существует стратегия с использованием m игроков очищает граф и является связной, например, «слева направо».

2. $a + m_3 > m$

   Рассмотрим подграф B, для которого $m_3 = m_3'$, $m_1 = a$ такое, что $m_3' + a = m$. Заметим, что для полученного блочного графа $b = m_3'$,

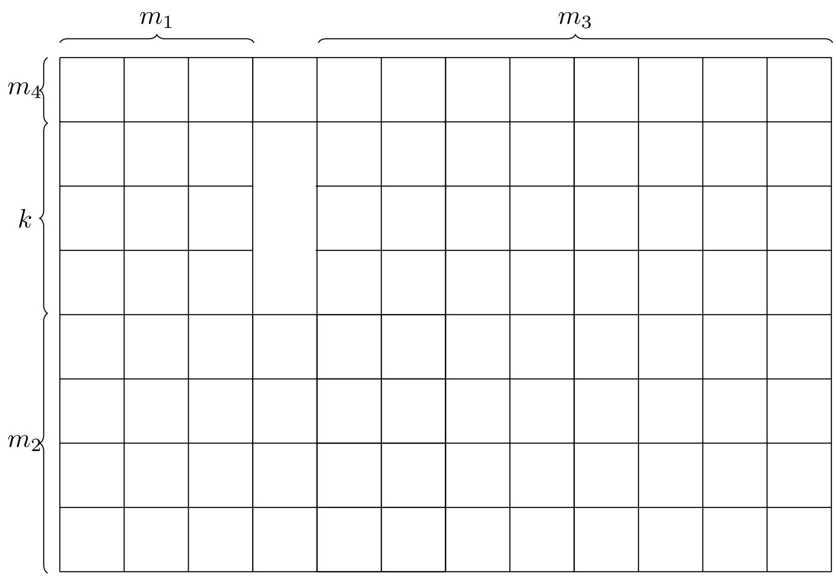

Рис. 20

так как $a + m_4 \leqslant m_2 + m_4 < m$, а $a + m'_3 = m$, следовательно $m'_3 > m_4 \geqslant a$, $m'_3 > m_2 \geqslant a$. Получаем, что $mixs(G) = m$, как и в предыдущем случае.

3. $a + m_3 < m$

   Заметим, что тогда $a \neq m_1$, т.к. $m_1 + m_3 = n > m$. Тогда рассмотрим такой подграф B, что $m_1 = m'_1 > a$ такое, что $m'_1 + m_3 = m$, то есть удалим из графа лишние столбцы. Снова получили, что B является блочным графа нужной нам структуры, для которого мы знаем, что $mixs(B) = \min\{m_3 + a, m\}$. Связные выигрывающие стратегии строятся уже описанными способами.

Если же $b = m_2$, то блок B размера $(m_2 + m_4) \times n$ — минор графа G, $mixs(B) = m_2 + m_4$. Докажем, что $a = m_4$. Из того, что $m_2 + m_4 < m < n$, $m_1 + m_3 = n$, $m_2 \geqslant m_3, m_1$, получаем $m_4 < m_3$, $m_4 < m_1$, а значит $a = m_4$. С помощью $(a + b)$ игроков исходный граф очищается, причём стратегия связная.

Теперь рассмотрим общий случай: есть блок $m \times n$, из которого удален блок размера $k_1 \times k_2$, пример показан на рисунке 21.

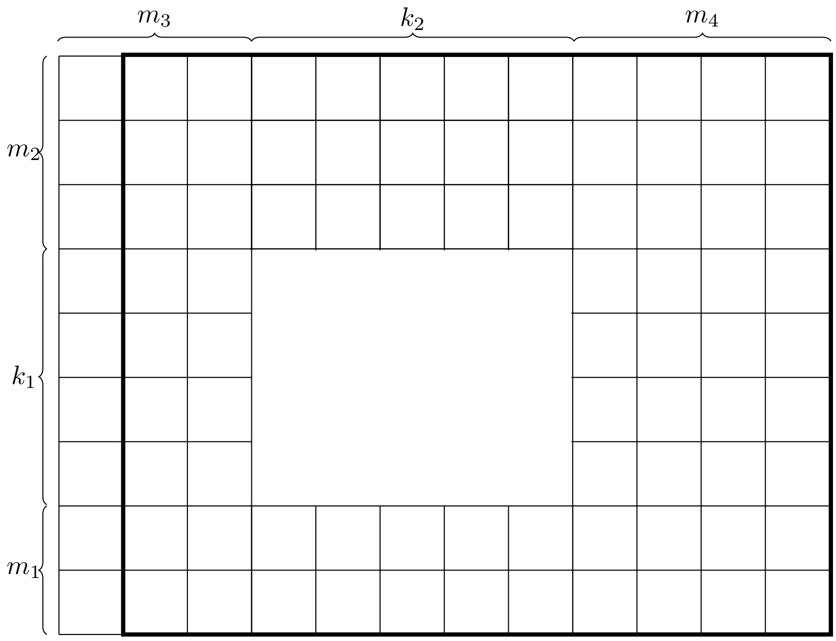

Рис. 21

Пусть $b = m_4 = \max\limits_{i \in 1:4} m_i$, $a = \max\limits_{i \in 1:4} m_i$. Рассмотрим подграф исходного графа, где $m_3 = a$ (граница полученного подграфа выделена). Заметим, что граф B, полученный удалением из блока $m \times a + b$ удалением блока $k_1 \times 1$ является минором исходного графа, а $mixs(B) = \min\{m, a + b\}$. Понятно, что исходный граф можно очистить этим количеством игроков уже описанными стратегиями.

8. Путевая ширина графа

Поисковые числа связаны соотношениями с такими инвариантами графа, как древесная и путевая ширина. В этом разделе рассматриваются некоторые свойства путевой ширины, помогающие при построении стратегий.

Древесной декомпозицией графа $G =< V, E >$ называется такое дерево T, вершины которого $V_1, \ldots, V_n$, $V_i \subset V$ для любого $i \in 1 : n$, удовлетворяют следующим свойствам:

1. $\bigcup\limits_{i=1}^{n} V_i = V$, то есть каждая вершина исходного графа G входит хотя бы в одну вершину древесной декомпозиции;

2. Если вершина v исходного графа содержится в $V_i \cap V_j$, то v содержится во всех вершинах древесной декомпозиции на пути из V_i в V_j;

3. Если существует ребро (v, w) в исходном графе, то существует вершина древесной декомпозиции V_k, что вершины $v, w \in V_k$, то есть смежные вершины входят вместе в хотя бы одну вершину древесной декомпозиции.

Шириной $w(T)$ древесной декомпозиции T графа G называется мощность максимальной её вершины минус один, то есть $\max\limits_{i \in 1:n} |V_i| - 1$. *Древесной шириной $tw(G)$* графа G называется минимальная ширина среди всех возможных древесных декомпозиций графа. Если рассматривать только те древесные декомпозиции, где полученное дерево является путём, то аналогично определяется *путевая декомпозиция* и *путевая ширина $pw(G)$* графа G. Понятно, что $pw(G) \geqslant tw(G)$.

Сформулируем несколько очевидных свойств путевой декомпозиции связного графа:

Утверждение 8.1. *Пусть $V_1, \ldots, V_n$ какая-то путевая декомпозиция P_1 графа G, где $V_i \subseteq V_{i+1}$ для какого-то i. Тогда $V_1, \ldots, V_{i-1}, V_{i+1}, \ldots V_n$ — тоже путевая декомпозиция P_2 графа G, при этом $w(P_1) = w(P_2)$.*

Доказательство. Покажем, что $V_1, \ldots, V_{i-1}, V_{i+1}, \ldots V_n$ — путевая декомпозиция графа G. Так как $V_i \subset V_{i+1}$, то $\bigcup\limits_{k=1,\ k \neq i}^{n} V_k = \bigcup\limits_{k=1}^{n} V_k = V$, то есть свойство 1 выполняется. Свойство 3 верно, так как все смежные вершины из V_i учитываются V_{i+1}, а свойство 2 по той же причине, что и свойство 1, а так как $|V_i| \leqslant |V_{i+1}|$, то $w(P_1) = w(P_2)$.

Аналогично доказывается следующее утверждение:

Утверждение 8.2. *Пусть* $V_1, \dots, V_n$ — *какая-то путевая декомпозиция* P_1 *графа* G, *где* $V_{i+1} \subseteq V_i$ *для какого-то* i. *Тогда* $V_1, \dots, V_i, V_{i+2}, \dots V_n$ — *тоже путевая декомпозиция* P_2 *графа* G, *при этом* $w(P_1) = w(P_2)$.

Утверждение 8.3. *Пусть* $V_1, \dots, V_n$ — *какая-то путевая декомпозиция* P_1 *связного графа* $G = (V, E)$, $|V| > 1$, *причём подграф, индуцированный* V_i *для какого-то* i, *содержит только изолированные вершины. Тогда* $V_1, \dots, V_{i-1}, V_{i+1}, \dots V_n$ — *тоже путевая декомпозиция* P_2 *связного графа* G, *при этом* $w(P_1) \geqslant w(P_2)$.

Доказательство. Так как граф связный, изолированных вершин нет, значит, для любой вершины существует вершина, смежная с ней, так что любая вершина $v \in V_i$ входит в ещё какую-то вершину путевой декомпозиции P_1, так что свойства 1 и 3 выполняются для P_2 тоже. Так как P_1 — путевая декомпозиция, то для неё выполняется свойство 2, тогда и для P_2 оно выполняется: если $v \in \ V_i \bigcap V_j, \ i < j$, то $v \in V_k$, где $k \in i+1 : j-1$. Таким образом, доказано, что P_2 — путевая декомпозиция. Из того, что все вершины P_2 содержатся в P_1, следует $w(P_1) \geqslant w(P_2)$.

Утверждение 8.4. *Пусть* $V_1, \dots, V_n$ — *какая-то путевая декомпозиция* P_1 *связного графа* G, $v \in V_i, \ v \notin V_{i+1}$, *где* v — *вершина исходного графа* G, *причём* v *не смежна ни с одной вершиной* $v' \in V_i$. *Пусть* $V_i' = V_i \setminus v$. *Тогда* $V_1, \dots, V_{i-1}, V_i', V_{i+1}, \dots V_n$ — *тоже путевая декомпозиция* P_2 *связного графа* G, *при этом* $w(P_1) \geqslant w(P_2)$.

Доказательство. Из связности графа следует, что $v \in V_k$ для какого-то $k \neq i, \ k \in 1 : n$, тогда свойство 1 выполнено. Так как $v \in V_i, \ v \notin V_{i+1}$, то свойство 2 тоже выполнено, так как для любой вершины $v' \in V, \ v' \neq v$ верно из того, что P_1 — путевая декомпозиция, а вершина v удалена только из последней вершины путевой декомпозиции, содержащей v. Свойство 3 выполняется, так как v — изолированная вершина в подграфе, индуцированном вершинами из V_i.

Заметим, что если вершина $v \in V_i$, $v \notin V_{i+1}$, при этом для любой вершины $v' \in V$, v' смежна с v, верно, что существует $k(v') < i$ такой, что $v, v' \in V_{k(v')}$, то можно построить новую путевую декомпозицию, где на месте V_i будет находиться $V_i' = V_i \setminus v$, причём ширина путевой декомпозиции не увеличится. Доказательство окончено.

Эти утверждения помогают выделить класс путевых декомпозиций, которые достаточно рассмотреть, чтобы найти путевую ширину графа. Далее мы ограничимся только теми путевыми декомпозициями, к которым нельзя применить утверждения, указанные выше. Теперь сформулируем свойство путевой декомпозиции графов блочной структуры, на которой достигается путевая ширина. Такую путевую декомпозицию назовём *минимальной.*

Утверждение 8.5. *Для любой путевой декомпозиции $V_1, \ldots, V_n$ графа блочной структуры G верно, что если вершина $a \in V_k$, $a \notin V_{k+1}$, то в V_k содержатся вершины a, b, c, которые находятся в одной ячейке.*

Доказательство. Предположим противное. Рассмотрим первую вершину V_k путевой декомпозиции, не содержащую такой тройки вершин исходного графа a, b, c. Из утверждений 8.1–8.4 следует, что существует вершина a такая, что $a \in V_k$, $a \notin V_{k+1}$, причём a смежна с какой-то вершиной $b \in V_k$, при этом $b \notin V_{k-1}$, иначе бы мы могли удалить a на предыдущем шаге. Так как $a \notin V_{k+1}$ и степень любой вершины графа G больше 1, то в исходном графе есть ребро (c, a), а значит, $c \in V_l$, $c \notin V_k$, $l < k$, так как иначе мы нашли искомую тройку вершин и получили противоречие. Рассмотрим ячейку вершин a, b, c, d. Так как $c \notin V_k$, то существует m такое, что $c, d \in V_m$. Если $d \in V_k$, то получим противоречие, так как тогда мы нашли тройку a, b, d. Таким образом, существует r, когда $b, d \in V_r$, $r < k$. Получаем, что из свойства 2 путевой декомпозиции $a, b \in V_{k-1}$. Тогда вершину a можно удалить из V_k, а мы рассматриваем путевые декомпозиции, к которым подобное действие уже нельзя применить. Таким образом, утверждение доказано.

Теперь заметим, что если $V_1, \ldots, V_n$ — минимальная путевая декомпозиция, то и $V_n, \ldots, V_1$ — тоже минимальная путевая декомпозиция, так как свойства путевой декомпозиции 1–3 по-прежнему выполняются, а мощность максимальной вершины декомпозиции не изменилась. Таким образом, утверждение 8.5 верно для $V_n, \ldots, V_1$, а значит, для $V_1, \ldots, V_n$ верно следующее:

Утверждение 8.6. *Для любой путевой декомпозиции $V_1, \ldots, V_n$ графа блочной структуры G верно, что если вершина $a \in V_k,\ \ a \notin V_{k-1}$, то в V_k содержатся вершины a, b, c, которые находятся в одной ячейке.*

Доказательство. Применим к путевой декомпозиции $V_n, \ldots, V_1$ утверждение 6, получим, что если вершина $a \in V_{k+1},\ \ a \notin V_k$, то в V_{k+1} содержатся вершины a, b, c, которые находятся в одной ячейке.

Теперь попробуем найти путевую ширина блока B размера $m \times n$. Выделим общее свойство путевой декомпозиции блоков:

Утверждение 8.7. *Обозначим $v_{i,j}$, где $j \in 1 : n$, — вершины i–ой строки блока B размера $m \times n$. Тогда все вершины V_k путевой декомпозиции графа, содержащие вершины i–ой строки, где $i \in 1 : m$ образуют путь, то есть все описанные вершины путевой декомпозиции имеют последовательные номера.*

Доказательство. Предположим противное, тогда существует такая пара вершин $v_{i,l},\ \ v_{i,l+1}$ и такие числа $p_1 < p_2$, что $v_{i,l} \in V_{p_1},\ \ v_{i,l} \notin V_k$ при $k > p_1$, а $v_{i,l+1} \in V_{p_2},\ \ v_{i,l+1} \notin V_s$ при $s < p_2$. Но по свойству путевой декомпозиции вершины $v_{i,l},\ \ v_{i,l+1}$ принадлежат какой-то вершине V_k, так в блоке B эти вершины смежны.

Воспользовавшись этими утверждениями, мы можем доказать следующее:

Утверждение 8.8. *Если B — блок размера $m \times n$, то $pw(B) = \min\{m, n\}$.*

Доказательство. Покажем, что в путевой декомпозиции блока существует вершина, содержащая вершин всех строк или столбцов блока. После этого воспользуемся утверждением 8.5 и получим, что $pw(G) = \min\{m, n\}$.

Рассуждения будут похожи на нахождение поискового числа блока. Более того, пример путевой декомпозиции, на которой достигается путевая ширина графа, строится с помощью стратегии очищения блока «слева направо» смешанного поиска: в V_i попадают те вершины блока, в которых перед шагом i и после него стоят игроки, а затем проводятся преобразования, описанные в утверждениях 8.1–8.4. По построению стратегии выполняются свойства 1 и 3 путевой декомпозиции, а свойство верно, так как стратегия монотонная.

Теперь покажем, что в путевой декомпозиции существует вершина, содержащая вершины всех строк или всех столбцов. Для это рассмотрим номер i такой, что $\bigcup\limits_{k=1}^{i} V_k$ содержит вершины всех строк или всех столбцов, а для любого $j < i$ это неверно. Рассмотрим вершину V_i. Пусть в ней нет вершин, например, строки j. Так как какая-то вершина этой строки уже была ранее, то по утверждению 8.7 любая вершина этой строки уже была в V_l, где $l < i$ (l для каждой вершины может быть своим). Но тогда существует $i' < i$, что $\bigcup\limits_{k=1}^{i'} V_k$ содержит вершины всех столбцов, откуда получаем противоречие.

Заключение

В работе найдены поисковые числа $mixs(G)$, $cmixs(G)$, $bs(G)$, $cbs(G)$ для следующих графов G блочных структуры:

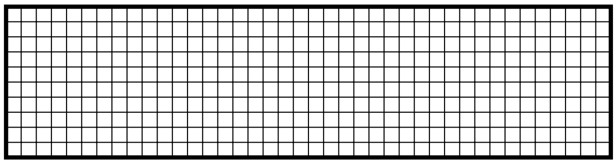

Рис. 22. Граф-блок

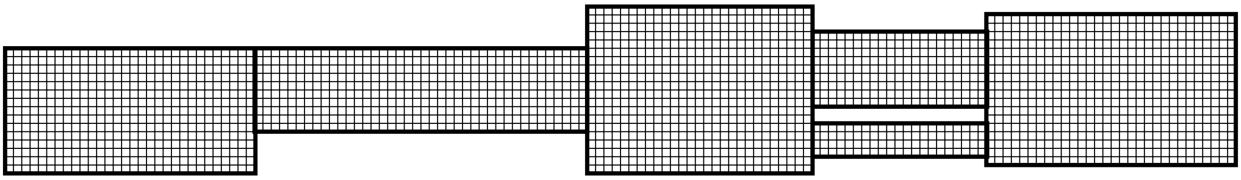

Рис. 23. Граф, полученный горизонтальной склейкой блоков B_i размера $m_i \times n_i$, где $m_i \leqslant n_i$

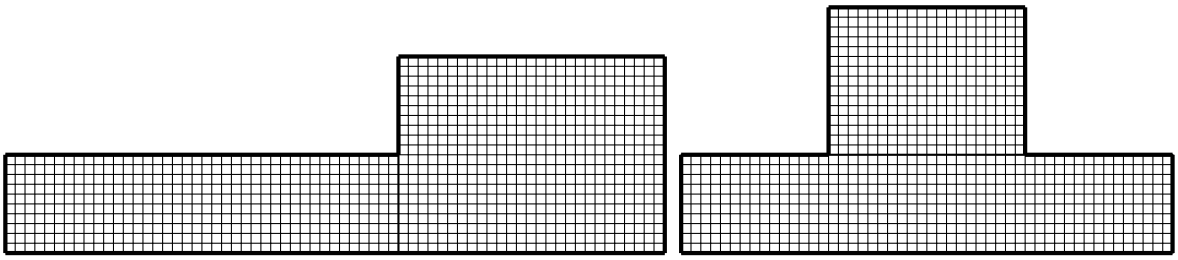

Рис. 24. Графы, полученные склейкой двух блоков

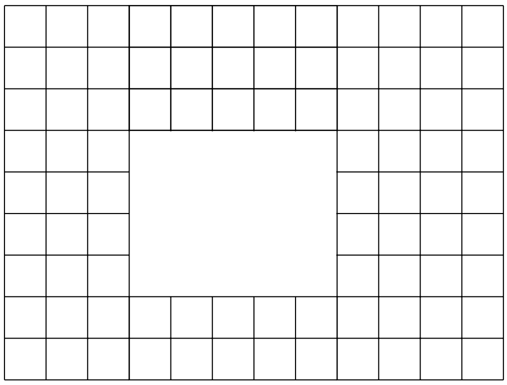

Рис. 25. Граф, полученный удалением одного блока из другого

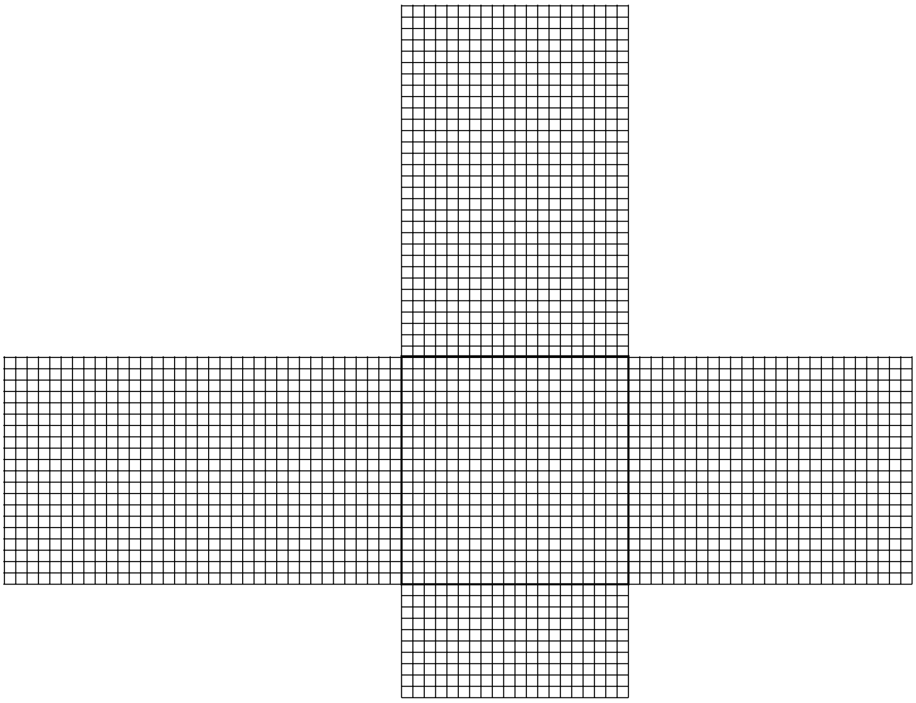

Рис. 26. «Граф–крест»

Для графа–блока найдена путевая ширина и сформулированы некоторые свойства путевой декомпозиции графов блочной структуры.

Приложение А

Рассматриваются графы из 4.1, полученные в результате склейки двух блоков (рисунок 27) с отождествлением угловых вершин. Для таких графов блочной структуры хочется найти некоторые поисковые числа. Вначале попробуем найти $mixs(B)$, для этого строим для описанных графов оптимальную монотонную стратегию смешанного поиска.

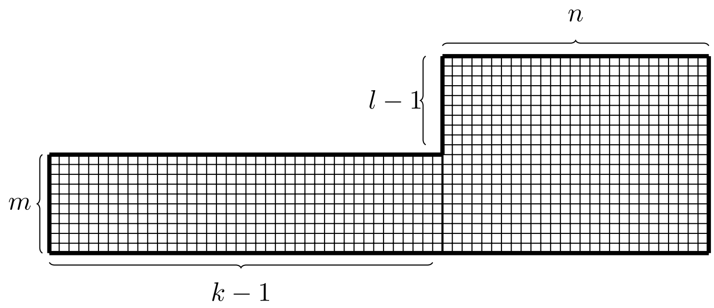

Рис. 27

Теперь рассматриваем первый шаг поиска, когда появляется полностью очищенный столбец или полностью очищенная строка:

1. Пусть нужная нам строка и столбец появились одновременно, а так как на предыдущем шаге ни очищенного столбца, ни очищенной строки не было, то и в строке, и в столбце есть игроки. Более того, рассматриваемые строка и столбец пересекаются (иначе невозможно очистить рёбра строки и столбца за один шаг). Тогда есть несколько вариантов:

   - Очищена строка из $n+k-1$ вершин и столбец из $m+l-1$. В таком случае $mixs(B) \geqslant \max\{n+k-1, m+l-1\}$. Действительно, если очищен такой столбец, то в каждой строке должен стоять игрок, так как полностью очищенных строк без игроков ещё нет, откуда получаем, что $mixs(B) \geqslant m + l - 1$. Аналогично рассуждая, получаем, что $mixs(B) \geqslant n + k - 1$. Тогда $mixs(B) \geqslant \max\{n + k - 1, m + l - 1\}$.

- Очищена строка из $n + k - 1$ вершин и столбец из m вершин. Тогда в m строках из $n + k - 1$ вершины есть игроки, то есть $mixs(B) \geqslant m$. Таким же образом получаем, что $mixs(B) \geqslant n + k - 1$, в итоге $mixs(B) \geqslant \max\{n + k - 1, m\}$.
- Очищена строка из n вершин и столбец из $m + l - 1$. Рассуждая так же, как и в предыдущем пункте, делаем вывод, что $mixs(B) \geqslant \max\{n, m + l - 1\}$.

Три полученных неравенства позволяют сделать оценку, что для стратегий такого типа $mixs(B) \geqslant \min\{\max\{n+k-1, m+l-1\}, \max\{n, m+l-1\}, \max\{n + k - 1, m\}\} \geqslant \min\{n + k - 1, m + l - 1\}$.

2. Очищена только строка из $n + k - 1$ вершин или только столбец из $m+l-1$. Тогда $mixs(B) \geqslant m+l-1$ или, соответственно, $mixs(B) \geqslant n + k - 1$.

3. Очищен только столбец из m вершин. Если при этом в каждом столбце стоит игрок, тогда $mixs(B) \geqslant n+k-1$. Если есть столбец без игроков, все рёбра которого загрязнены, тогда в каждой строке, с которой пересекается столбец, стоит игрок, то есть выставлено как минимум m игроков. Эти строки из $n + k - 1$ вершины будем называть «длинными», а из n — «короткими». Если в каждой строке стоит игрок, то $mixs(B) \geqslant m + l - 1$, иначе есть полностью загрязнённая строка из n вершин. Пусть после появления такого столбца на каждом шаге оптимальной стратегии в «длинных» строках всегда есть хотя бы по одному игроку, пока есть полностью загрязнённая «короткая» строка. Далее рассмотрим, сколько нужно дополнительно игроков, что

избавиться от таких строк. Для этого необходимо, чтобы в каждой такой строке стоял игрок, тогда снова оценка $mixs(B) \geqslant m + l - 1$, иначе всегда есть строка без игроков.

Рассмотрим тогда шаг, когда в графе осталась только одна грязная «короткая» строка без игроков. Если есть полностью очищенная строка, то количество игроков в «коротких» строках n, тогда $mixs(B) \geqslant m + n$. В противном случае в остальных строках есть игроки. Если в строке были очищенные рёбра и один игрок, то мы не можем его снять со строки в силу монотонности стратегии, значит, либо была строка с двумя игроками, тогда $mixs(B) \geqslant m+l-1$, либо мы сняли игрока с вершины строки без очищенных рёбер и снова получили такую строку. Таким образом, $mixs(B) \geqslant \min\{m+l-1, m+n\}$.

Теперь посмотрим, когда в «длинных» строках может быть меньше m игроков, то есть, когда мы можем убрать игрока оттуда при наличии полностью загрязнённой «короткой» строки. В силу монотонности стратегии это возможно только и только тогда, когда есть очищенная строка. Теперь рассмотрим шаг, на котором появилась первая очищенная «длинная» строка. С одной стороны, стоит хотя бы m игроков в длинных строках, с другой — стоят игроки в n «длинных» столбцах, то есть $mixs(B) \geqslant \max\{m, n\}$.

Таким образом, можно сказать, что

$$mixs(B) \geqslant \min\{\min\{m+l-1, m+n\}, \max\{m, n\}, m+l-1, n+k-1\}$$

$$mixs(B) \geqslant \min\{\max\{m, n\}, m + l - 1, n + k - 1\}.$$

4. Очищена только строка из n вершин. Аналогично рассуждая, получаем, что

$$mixs(B) \geqslant \min\{\max\{m, n\}, m + l - 1, n + k - 1\}.$$

Обобщая полученные неравенства, видим окончательную оценку

$$mixs(B) \geqslant \min\{\max\{m,n\}, m+l-1, n+k-1\},$$

которую можно переписать в следующем виде

$$mixs(B) \geqslant \min\{m, n+k-1\},\ m \geqslant n,\ mixs(B) \geqslant \min\{n, m+l-1\},\ n > m$$

.

Приложение B

Ищется $cmixs(B)$ для графов B блочной структуры, являющихся результатом склейки двух блоков без отождествления угловых вершин.

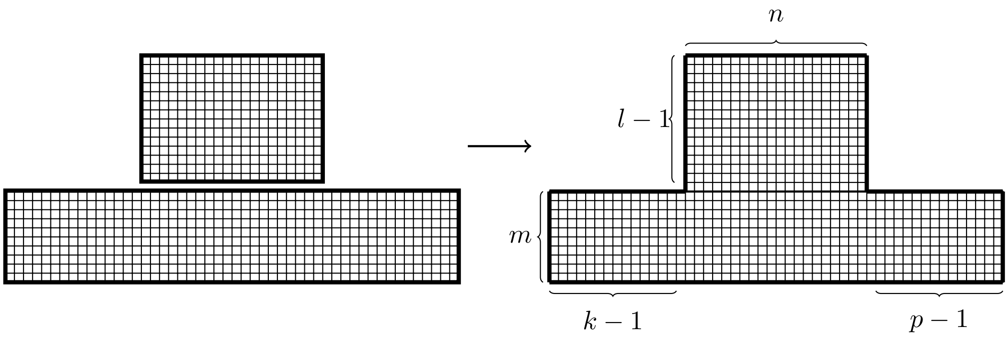

Рис. 28

Рассматривается оптимальная монотонная стратегия смешанного поиска и первый шаг, когда появляется полностью очищенный столбец или полностью очищенная строка, пусть $k \leqslant p$, то есть в рассуждениях мы при выборе между p и k выбираем k:

1. Пусть нужная нам строка и столбец появились одновременно, а так как на предыдущем шаге ни очищенного столбца, ни очищенной строки не было, то и в строке, и в столбце есть игроки. Более того, рассматриваемые строка и столбец пересекаются. Тогда есть несколько вариантов (так же, как и в склейке выше):

   - Очищена строка из $n + p + k - 2$ вершин и столбец из $m + l - 1$. В таком случае $mixs(B) \geqslant \max\{n + k + p - 2, m + l - 1\}$. Действительно, если очищен такой столбец, то в каждой строке должен стоять игрок, так как полностью очищенных строк без игроков ещё нет, откуда получаем, что $mixs(B) \geqslant m + l - 1$. Аналогично рассуждая, получаем, что $mixs(B) \geqslant n + p + k - 2$. Тогда $mixs(B) \geqslant \max\{n + p + k - 2, m + l - 1\}$.

- Очищена строка из $n+p+k-2$ вершин и столбец из m вершин. Тогда в m строках из $n+2k-2$ вершины есть игроки, то есть $mixs(B) \geqslant m$. Таким же образом получаем, что $mixs(B) \geqslant n+2k-2$, в итоге $mixs(B) \geqslant \max\{n+p+k-2, m\}$.

- Очищена строка из n вершин и столбец из $m+l-1$. Рассуждая так же, как и в предыдущем пункте, делаем вывод, что $mixs(B) \geqslant \max\{n, m+l-1\}$.

Три полученных неравенства позволяют сделать оценку, что для стратегий такого типа $mixs(B) \geqslant \min\{\max\{n+p+k-2, m+l-1\}, \max\{n, m+l-1\}, \max\{n+p+k-2, m\}\}$. Заметим, что в таком случае стратегия заведомо не будет блочной.

2. Очищена только строка из $n+p+k-2$ вершины или только столбец из $m+l-1$. Тогда $mixs(B) \geqslant m+l-1$ или $mixs(B) \geqslant n+p+k-2$.

3. Очищен только столбец из m вершин, тогда либо в каждом столбце стоит игрок и $mixs(B) \geqslant n+p+k-2$, либо есть столбец без игроков, всё рёбра которого загрязнены, откуда получаем, что в каждой строке, с которой пересекается столбец, стоит игрок, то есть выставлено как минимум m игроков. Эти строки из $n+p+k-2$ вершины будем называть «длинными», а из n — «короткими». Если в каждой строке стоит игрок, то $mixs(B) \geqslant m+l-1$, иначе есть полностью загрязнённая строка из n вершин. Пусть после появления такого столбца на каждом шаге оптимальной стратегии в «длинных» строках всегда есть хотя бы по одному игроку, пока есть полностью загрязнённая «короткая» строка. Далее оценим, сколько нужно дополнительно игроков, чтобы избавиться от таких строк. Для этого необходимо, чтобы в каждой такой строке стоял игрок, тогда снова оценка $mixs(B) \geqslant m+l-1$, иначе всегда есть строка без игроков.

Рассмотрим тогда шаг, когда в графе осталась только одна грязная «короткая» строка без игроков. Если есть полностью очищенная строка, то количество игроков в «коротких» строках n, тогда $mixs(B) \geqslant m + n$. В противном случае в остальных строках есть игроки. Если в строке были очищенные рёбра и один игрок, то мы не можем его снять со строки в силу монотонности стратегии, значит, либо была строка с двумя игроками, тогда $mixs(B) \geqslant m+l-1$, либо мы сняли игрока с вершины строки без очищенных рёбер и снова получили такую строку. Таким образом, $mixs(B) \geqslant \min\{m+l-1, m+n\}$.

Теперь посмотрим, когда в «длинных» строках может быть меньше m игроков, то есть, когда мы можем убрать игрока с этих вершин при наличии полностью загрязнённой «короткой строки». В силу монотонности стратегии это возможно только и только тогда, когда есть очищенная «длинная» строка. Давайте рассмотрим шаг, на котором появилась первая очищенная «длинная» строка. Заметим, что в каждом из n длинных столбцов стоит хотя бы один игрок. Если в каждом столбце стоит игрок, тогда $mixs(B) \geqslant n+p+k-2$. Пусть есть очищенные столбцы без игроков. Если все очищенные столбцы находятся по одну сторону от длинных столбцов, то $mixs(B) \geqslant \max\{n + k - 1, m\}$ (n в «длинных» столбцах и $k - 1$ в «коротких» столбцах по другую сторону). В противном случае, есть очищенные столбцы по обе стороны от «длинных» столбцов. Рассмотрим теперь первый момент, когда по обе стороны есть очищенные столбцы. Если с одной из сторон есть грязный «короткий» столбец, то оценка $mixs(B) \geqslant \max\{n + m, m\}$, так как между столбцами в каждой строке для защиты от загрязнения должен быть хотя бы один игрок. Иначе есть точно с одной стороны от «длинных» столбцов $k - 1$ столбец с хотя бы одним игроком, откуда оценка $mixs(B) \geqslant \max\{n + k - 1, m\}$.

Итак, получили, что $mixs(B) \geqslant \min\{\max\{n+m, m\}, \max\{n+k-1, m\}, n+p+k-2, m+l-1, m+n\} = \min\{n+m, \max\{n+k-1, m\}, n+p+k-2, m+l-1\}$.

4. Очищена только «короткая» строка. Если в каждой строке стоит игрок, то $mixs(B) \geqslant m+l-1$, иначе есть строка без игроков и очищенных рёбер, то есть в «длинных» столбцах есть n игроков. Далее, оценим количество игроков, если все «короткие» столбцы очищаются новыми игроками, отличными от n уже выставленных. Изначально у нас нет очищенных «коротких» столбцов. Рассмотрим первый момент, когда появляется первый очищенный столбец. Снова, либо в каждом столбце стоит игрок, тогда $mixs(B) \geqslant n+k+p-2$, либо есть полностью грязный столбец без игроков. В случае выполнения второго посмотрим, как расположены очищенный и полностью грязный столбец. Если они по одну сторону от «длинных» столбцов, то между ними в каждой строке стоит игрок, тогда $mixs(B) \geqslant n+m$, если по разные, то в $k-1$ «коротком» столбце есть игроки, то есть всего игроков не меньше $n+k-1$. Так как по другую сторону от длинных столбцов есть полностью грязный столбец, то общее количество игроков в силу монотонности оптимальной стратегии хотя бы m, ведь очищенный столбец не теряется. Тогда в этом случае оценка $mixs(B) \geqslant \max\{m, n+k-1\}$.

   Аналогично предыдущему пункту, можно показать, что для того, чтобы с одной стороны от «длинных» столбцов не было полностью загрязнённых столбцов, нужен заведомо $\min\{k-1, m\}$ игроков.

   Теперь посмотрим, когда можно убрать с «длинного» столбца одного из игроков. Понятно, что для этого столбец должен быть полностью очищен. Рассмотрим шаг, когда с полностью очищенного столбца снимается последний игрок. Если на этом шаге полностью неочищенный столбец находится от него только с одной стороны, тогда $mixs(B) \geqslant \max\{m, n+k-1\}$ (m игроков должны защищать от загрязнения, а $k-1$ необходимы, чтобы в одном из блоков не было полностью загрязнённых столбцов, так как в таком случае либо в каждом столбце есть игрок с одной сторон, либо есть очищенный). Если же такие столбцы находятся с двух сторон, то $mixs(B) \geqslant \max\{2m, n\}$ (n заведомо стоят, а $2m$ защищают от загрязнения, с каждой стороны по m).

В итоге, имеет место неравенство

$$mixs(B) \geqslant \min\{n+m, \max\{n+k-1, m\}, n+p+k-2, m+l-1, \max\{2m, n\}\}.$$

Литература

1. Абрамовская Т.В., Петров Н.Н. Теория гарантированного поиска на графах // Дифференциальные уравнения и процессы управления. 2012. № 2. С. 9–65.

2. Головач П.А. Об одном топологическом инварианте в задачах преследования // Дифференциальные уравнения. 1989. Т. 2, № 6. С. 923–929.

3. Петров Н.Н. Задачи преследования при отсутствии информации об убегающем // Дифференциальные уравнения. 1982. Т. 18, № 8. С. 1345–1352.

4. Parsons T.D. Pursuit–evasion in a graph // Theory and applications of graphs. Berlin: Springer. 1978. P. 426–441.

5. Kirousis L.M., Papadimitriou C.H. Searching and pebbling // Theoretical Computer Science. 1986. V. 47, № 1. P. 205–218.

6. Takahashi A., Ueno S., Kajitani Y. Mixed searching and proper-path-width // Theoretical Computer Science. 1995. V. 137, № 2. P. 253–268.

7. Barriere L., Fraigniaud P., Santoro N., Thilikos D.M. Connected and Internal Graph Searching // In 29th Workshop on Graph Theoretic Concepts (WG). Springer-Verlag. 2003. P. 34–45.

8. Boting Yang Strong-mixed Searching and Pathwidth // Journal of Combinatorial Optimization. 2007. V. 13, № 1. P. 47–59.