Министерство образования Республики Беларусь

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

**Факультет прикладной математики и информатики**

Кафедра компьютерных технологий и систем

ГОЛУБЕВ АЛЕКСАНДР НИКОЛАЕВИЧ

**ЗАДАЧА О ПОКРЫТИИ ГРАФА БИКЛИКАМИ**

Курсовой проект  
студента 3 курса 13 группы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| “Допустить к защите“  **Руководитель проекта**  Лепин Виктор Васильевич  ученый секретарь Института математики НАН Беларуси, доцент, канд. физ.-мат. наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  “\_\_\_” \_\_\_\_\_\_\_\_\_\_\_ 2013 г |  |  |  |  |

Минск 2013

**Белорусский государственный университет**

**Факультет прикладной математики и информатики**

Кафедра дискретной математики и алгоритмики

“**Утверждаю**”

Заведующий кафедрой

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_В.М. Котов

“\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2013 г.

**ЗАДАНИЕ**

**ПО ПОДГОТОВКЕ КУРСОВОГО ПРОЕКТА**

Студенту 3 курса Голубеву Александру Николаевичу

1. **Тема работы** ЗАДАЧА О ПОКРЫТИИ ГРАФА БИКЛИКАМИ

2. **Срок сдачи студентом законченной работы** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. **Исходные данные к работе**

1. Учебная литература и научные публикации в области теории графов.
2. Перечень требований к реализации алгоритмов решения поставленных задач.

4. **Перечень вопросов подлежащих разработке или краткое содержание работы**

1. Изучение литературы по теме.
2. Выяснить сложность решения задачи о бикликовом покрытии декартова произведения графов.
3. Провести исследование и реализовать алгоритмы решения задач о бикликовом покрытии.
4. Протестировать разработанный алгоритм.

5. **Перечень графического материала**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6. **Консультанты по работе** (с указанием относящихся к ним разделов работы)

Лепин В.В., ученый секретарь Института математики НАН Беларуси, доцент, кандидат физ.-мат. наук

7. **Дата выдачи задания** “\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2013г.

8. **Календарный график** работы на весь период (с указанием этапов работы и   
сроков их выполнения)

Сентябрь-октябрь: Изучение литературы. Анализ имеющихся на данный момент методов, решающих аналогичный круг задач. Выделение основных свойств бикликовых покрытий графа. Составление плана работы.

Ноябрь-январь: Анализ и изучение существующих методов, выбор наиболее подходящего метода для решения поставленных задач.

Февраль-март: Реализация алгоритмов.

Апрель: Тестирование. Устранение ошибок.

Май-июнь: Оформление и защита работы.

**Руководитель** \_\_\_\_\_\_\_\_\_\_\_\_\_\_ / В.В. Лепин / …. сентября 2006 г.

**Задание принял к исполнению** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ .… сентября 2006 г.

(подпись студента)

**Аннотация**

В данной работе рассмотрена задача о покрытии графов бикликами. В работе были установлены границы на число бикликового покрытия графа**,** найдено минимальное число бикликового покрытия для лестниц и квадратных решеток. Построен алгоритм для нахождения бикликового покрытия в графах, а также для нахождения минимального бикликового покрытия в последовательно-параллельных графах, а также реализованы данные алгоритмы языке программирования Java.

**Анатацыя**

У дадзенай працы разгледжана задача аб пакрыцці дэкартавага здабытку графаў. Были знойдзены абмежавання на мінімальны лік біклікавага пакрыцця графа**,** знойдзены мінімальны лік біклікавага пакрыцця для лесвіцы и квадратных рашотак. Знойдзены алгарытм біклікавага пакрыцця ў графах, а таксама алгарытм для знаходжання мінімальнага біклікавага пакрыцця ў паслядоўна-паралельных графах, а таксама рэалізаваны гэтыя алгарытмы на мове праграмавання Java.

**The annotation**

In this work biclique cover of Cartesian product graphs was considered. Found bounds for minimum biclique cover and exactly value for stairs and square lattice. Found algorithm for built biclique cover and algorithm for built minimum biclique cover for serial-parallel graphs and this algorithms was implemented on programming language Java.

**Реферат**

Курсовой проект, 13 с., 2 рис., 7 источников, 1 приложение.

Ключевые слова: число бикликового покрытия графов, бикликовое покрытие графов, максимальное паросочетание, декартово произведение графов, последовательно-параллельные графы.

Объект исследования – алгоритмы решения задачи о бикликовом покрытии графов.

Цель работы – освоение алгоритмов решения задачи о бикликовом покрытии и их реализация.

Содержание

[1. Введение 6](#_Toc387611757)

[2. Краткая теория вопроса 6](#_Toc387611758)

[3. Оценки числа бикликового покрытия 6](#_Toc387611759)

[3.1. Декартово произведение графов 6](#_Toc387611760)

[3.2. Лестницы и квадратные решетки. 8](#_Toc387611761)

[4. Алгоритмы построения бикликового покрытия 11](#_Toc387611762)

[4.1. Эвристический алгоритм 11](#_Toc387611763)

[4.2. Последовательно-параллельные графы 12](#_Toc387611764)

[4.2.1 Общие сведения 12](#_Toc387611765)

[4.2.2 Узлы бинарного sp -дерева и биклики последовательно-параллельного графа. 14](#_Toc387611766)

[4.2.3 Алгоритм для нахождения числа бикликового покрытия последовательно-параллельного графа 20](#_Toc387611767)

[5. Сравнение алгоритмов 24](#_Toc387611768)

[6. Заключение 24](#_Toc387611769)

[Список использованной литературы 25](#_Toc387611770)

# Введение

Бикликой графа G называется подграф графа G(необязательно порожденный), изоморфный полному двудольному графу. Числом бикликового покрытия графа G называется наименьшее число биклик графа G, которое понадобится чтобы покрыть все ребра графа. Число бикликового покрытия является важным графовым параметром. С практической точки зрения, задача покрытия наименьшим числом биклик имеет широкое применение в области биоинформатики, искусственного интеллекта, компактного представления графа. С теоретической точки зрения, число бикликового покрытия графа играет важную роль в теории коммуникационной сложности.

Задача нахождения числа бикликового покрытия графа является NP-трудной и остается NP-трудной в классе двудольных хордальных графов. Но в классах двудольных C4-свободных графов, двудольных домино-свободных графов, двудольных выпуклых графов, последовательно-параллельных графах и графов с ограниченной путевой шириной задача о нахождении бикликового покрытия является полиномиально разрешимой. Число бикликового покрытия известно для простой цепи, простого цикла, полного графа, шестиугольной решетки.

# Краткая теория вопроса

Рассматриваются только неориентированные графы без кратных ребер и петель с множеством вершин  и множеством ребер . Граф с пустым множеством ребер называется пустым графом. Граф G называется двудольным, если существует разбиение V(G)= U1 U2 такое, что концы каждого ребра G принадлежат разным множествам U1,U2. Двудольный граф  с разбиением V(G) = U1  U2  называется полным двудольным, если каждая вершина из U1 смежна с каждой вершиной из U2. Бикликой графа G называется полный двудольный подграф G. Биклика графа называется максимальной, если она не содержится в другой биклике с большим числом вершин. Биклика графа G не обязательно является порожденным подграфом G. Биклику с биразбиением U1,U2 будем обозначать как K(U1,U2). Биклика K(U1,U2) называется звездой с центром в вершине , если U1={u} или U2={u}. Множество биклик  графа  называется бикликовым покрытием графа , если каждое ребро графа  содержится, по крайней мере, в одной биклике из . Наименьшее число биклик в бикликовом покрытии графа  называется числом бикликового покрытия графа  и обозначается как .

# Оценки числа бикликового покрытия

## Декартово произведение графов

Декартовым произведением графов G1 и G2 называется граф G1 × G2 с множеством вершин V(G1) × V(G2) и две вершины (u1, u2); (v1, v2) смежны тогда и только тогда, когда или u1 = v1 и u2, v2 смежны в G2, или u2 = v2 и u1, v1 смежны в G1.

**Теорема 1.** Для любых двух непустых графов G1, G2 выполняется неравенство:

bc(G1 × G2) ≤ |V(G1)| · bc(G2) + |V(G2)| · bc(G1).

*Доказательство*. Для того чтобы доказать это неравенство, построим бикликовое покрытие графа G1 × G2, которое содержит не более чем |V(G1)| · bc(G2) + |V(G2)| · bc(G1) биклик.

Пусть S1 — бикликовое покрытие G1 мощности bc(G1) и S2 — бикликовое покрытие

G2 мощности bc(G2). Рассмотрим произвольную биклику K(A, B) ∈ S1 и произвольную

вершину v ∈ V(G2). В графе G1 ×G2 каждая вершина из множества A×{v} смежна с каждой

вершиной из B ×{v}: Поэтому существует биклика G1 ×G2 с биразбиением (A×{v}, B ×{v}).

Аналогично, если K(A, B) ∈ S2 и v ∈ V(G1), то существует биклика G1 × G2 с биразбиением

({v} × A; {v} × B). Тогда

S = {K(A × {v}, B × {v}) : K(A, B) ∈ S1, v ∈ V(G2)}∪

∪{K({v} × A, {v} × B) : K(A, B) ∈ S2, v ∈ V(G1)}

является бикликовым покрытием G1 × G2 и

|S| = |V(G1)| · bc(G2) + |V(G2)| · bc(G1).

Действительно, пусть e = {(u1, u2), (v1, v2)} — ребро G1 × G2. Возможны два случая:

(i) u1 = v1, {u2, v2} ∈ E(G2) и, если биклика K(A, B) ∈ S2 содержит ребро {u2, v2}, то

биклика K({u1} × A, {u1} × B) ∈ S содержит ребро e; (ii) u2 = v2, {u1, v1} ∈ E(G1) и, если

биклика K(A′, B′) ∈ S1 содержит ребро {u1, v1}, то биклика K(A′× {u2}, B′× {u2}) ∈ S содержит ребро e: Таким образом, каждое ребро графа G1 × G2 содержится, по крайней

мере, в одной биклике из S. Следовательно, множество S биклик графа G1 × G2 является

бикликовым покрытием G1 × G2. *Теорема доказана.*

Следующее утверждение является прямым следствием теоремы 1.

**Следствие 1**. Если G1,G2 — полные двудольные графы, то

bc(G1 × G2) ≤ |V(G1)| + |V(G2)|:

Используя следствие 1, можно получить оценку сверху на число бикликового покрытия декартова произведения двух графов, которая не зависит от количества вершин в этих графах.

**Теорема 2**. Для любых двух непустых графов G1, G2 без изолированных вершин выполняется bc(G1 × G2) ≤ 2 · (∆(G1) + ∆(G2)) · bc(G1) · bc(G2).

*Доказательство*. Рассмотрим бикликовые покрытия S1, S2 наименьшей мощности графов G1, G2 соответственно. Пусть S1 = {K(X1, Y1),…, K (Xr, Yr)}, где r = bc(G1) и S2 = {K(U1, V1),…, K (Us, Vs)} где s = bc(G2). Заметим, что K(Xi, Yi) × K(Uj, Vj) является подграфом графа G1 × G2 для любых i = 1, r, j = 1, s. Рассмотрим граф

.

Покажем, что множество вершин и множество ребер графов G и G1 × G2 совпадают. Легко видеть, что V (G) ⊆ V (G1 ×G2) и E(G) ⊆ E(G1 ×G2). Остается доказать, что а) V(G1 ×G2) ⊆ V (G) и б) E(G1 × G2) ⊆ E(G).

Для того чтобы доказать, что условие а) выполняется, рассмотрим произвольную верши-ну (u1, v1) ∈ V (G1 × G2). Так как графы G1, G2 не содержат изолированных вершин, то существует ребро e1 ∈ E(G1) инцидентное вершине u1 и ребро e2 ∈ E(G2) инцидентное вершине v1. Тогда существуют биклики K(Xi, Yi) ∈ S1 и K(Uj, Vj) ∈ S2 такие, что K(Xi, Yi)

содержит ребро e1, а K(Uj, Vj) содержит ребро e2, где i ∈ {1,…,r }, j ∈ {1,…,s}. Так как u1 ∈ Xi ∪ Yi и v1 ∈ Uj ∪ Vj. то граф K(Xi, Yi) × K(Uj, Vj) содержит вершину (u1, v1). Из определения графа G следует, что (u1, v1) ∈ V (G). Выполнимость условия а) доказано.

Покажем, что условие б) также выполняется. Пусть e = {(u1, u2), (v1, v2)} — произвольное ребро графа G1 × G2. Из определения декартова произведения графов следует, что выполняется одно из следующих двух условий: i) u1 = v1 и {u2. v2} ∈ E(G2) или ii) u2 = v2 и {u1, v1} ∈ E(G1). Без потери общности, предположим, что выполняется условие i) (в случае условия ii) можно провести аналогичные рассуждения). Так как граф G1 не содержит изолированных вершин, то существует ребро e1 ∈ E(G1) инцидентное вершине u1. Тогда, существуют биклики K(Xi, Yi) ∈ S1 и K(Uj, Vj) ∈ S2 такие, что K(Xi, Yi) содержит ребро e1, а K(Uj, Vj) содержит ребро {u2, v2}. Граф K(Xi, Yi) × K(Uj, Vj) содержит ребро e. Из определения графа G следует, что e ∈ E(G). Следовательно, условие б) выполняется.

Таким образом, множество вершин и множество ребер графов G и G1 × G2 совпадают, это означает, что bc(G1 × G2) = bc(G). Для того чтобы доказать теорему 2, построим бикликовое покрытие графа G, которое состоит из не более чем 2 · (∆(G1) + ∆(G2)) · bc(G1) · bc(G2) биклик. Согласно следствию 1, для любых биклик K(Xi, Yi) ∈ S1 и K(Uj, Vj) ∈ S2 существует бикликовое покрытие графа K(Xi, Yi) × K(Uj, Vj) мощности не более |Xi| + |Yi| + |Uj| + |Vj|. Заметим, что |Xi| +|Yi| ≤ 2 · ∆(G1) и |Uj| +|Vj| ≤ 2 · ∆(G2). Пусть Si,j — бикликовое покрытие графа K(Xi, Yi)×K(Uj, Vj) мощности не более 2· ∆(G1)+2 · ∆(G2) для всех i = 1,…, r и j = 1,…, s.

Из определения графа G следует, что множество биклик S = является бикликовым покрытием графа G мощности не более 2 · (∆(G1) + ∆(G2)) · r · s.

## Лестницы и квадратные решетки.

Два различных ребра графа  будем называть зависимыми, если они содержатся в некоторой биклике графа. Заметим, что если два ребра содержатся в одной биклике, то они смежны или содержатся в некотором 4-цикле графа. Два различных ребра графа  называются независимыми, если они не являются зависимыми. Подмножество  ребер графа  называется независимым, если любые два различных ребра из  независимы. Следующая лемма непосредственно следует из определений.

**Лемма 1**. Пусть  — граф и  — независимое подмножество ребер графа , тогда .

Используя лемму 1, можно установить число бикликового покрытия графа , который называется лестницей.

**Теорема 3**.  для всех n ≥ 3:

*Доказательство*. Пусть v0 и v1 — вершины графа K2, а  – вершинная простая цепь Pn имеет множество вершин {u, u2,…, un} и множество ребер {{ui; ui+1} : }. Граф  изображен на рис. 1.

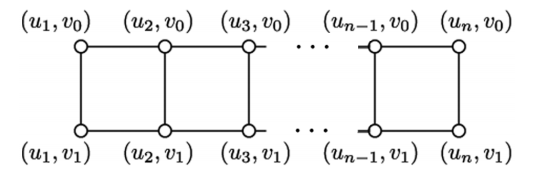


Рисунок 1 - Граф

Отметим, что подмножество ребер F = {{(ui, vi mod 2), (ui+1; vi mod 2)} : } графа  является независимым. Согласно лемме 1, =. С другой стороны, множество всех циклов длины 4 графа  является бикликовым покрытием  мощности . Следовательно, .

Решеткой называется двудольный граф Gm,n = (Vm,n, Em,n) такой, что

Vm,n = {(ui, vj) : 1 ≤ i ≤ m; 1 ≤ j ≤ n};

Em,n = {{(ui, vj), (uk; vl)} : (ui; vj) ∈ Vm,n, (uk, vl) ∈ Vm,n, }.

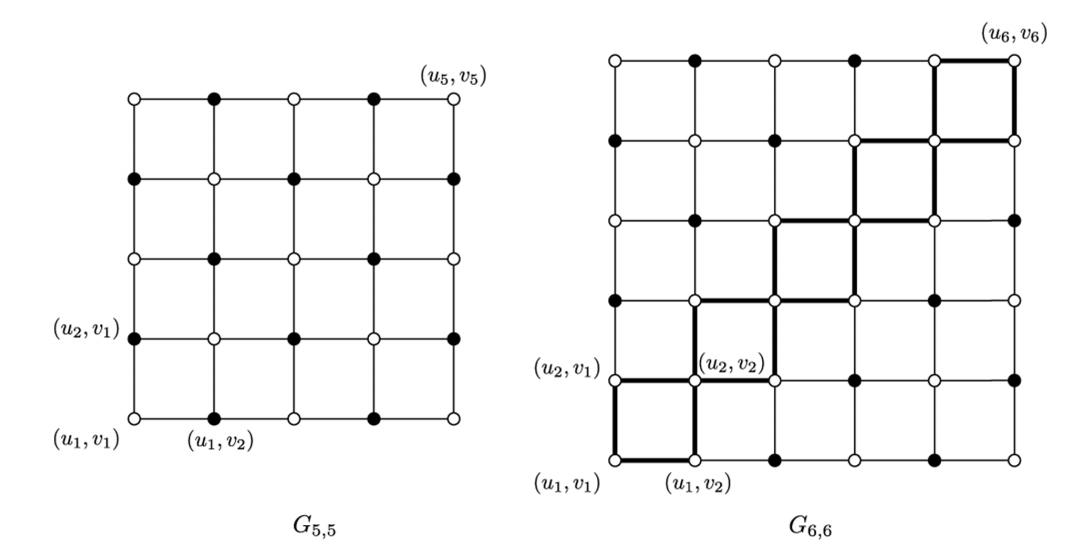
Заметим, что решетка Gm,n изоморфна графу . Решетку Gn,n будем называть квадратной решеткой.

|  |  |  |
| --- | --- | --- |
| **Теорема 4**. Для каждого n ≥ 2 выполняется | |  |
| n,n |

*Доказательство*. Покажем, что для квадратной решетки Gn,n существует бикликовое покрытие мощности в случае, когда n— нечетно и в случае, когда n— четно. Пусть — нечетное число. Решетка Gn,n представляет собой двудольныйграф с биразбиением , где V1 =  и

. При этом и .

Множество звезд с центрами в вершинах из множества V1 является бикликовым покрытием Gn,n мощности. На рис. 2 изображена квадратная решетка G5,5. В этом случае множество V1 состоит из всех черных вершин.



Пусть – четное. Тогда в качестве бикликового покрытия возьмем множество звезд, аналогичное для случая с нечетным и дополним его бикликами K2,2 на диагонали, тогда .

Результат полученный для декартова произведения двух простых цепей   позволяет сформулировать следующую гипотезу относительно значения числа бикликового покрытия произведения двух простых цепей  для m < n.

**Гипотеза 1**. Пусть m, n — натуральные числа, такие что . Если  — нечетное, то

   ,

где  — число вершинного покрытия графа . Если  — четное натуральное число, то

**Гипотеза 2.** Пусть =× , где , n, тогда

**Гипотеза 3.** Пусть, 2 -натуральные числа, тогда

Из гипотезы следует, что число бикликового покрытия данных графов совпадает с минимальным вершинным покрытием. Так как эти графы являются двудольными (т.к отсутствуют циклы нечетной длины), то для них справедлива теорема Кенига: в любом двудольном графе число рёбер в максимальном паросочетании равно числу вершин в минимальном вершинном покрытии. Но для двудольного графа задача нахождениея максимального паросочетания является полиномиальной, и может быть решена, например, при помощи алгоритма Хопкрофта-Карпа.

**Алгоритм Хопкрофта-Карпа**.

*Вход­*: Двудольный граф G(U V, E)

*Выход:* Наибольшее паросочетание

**begin**

**do**

*//строим максимальное по включению множество*

*//вершинно-неперсекающихся M-чередующихся цепей*

;

;

**while**

**end**

Трудоемкость данного алгоритма составляет .

Запустив программу, можем убедиться, что мощность минимального вершинного покрытия совпадает с заданными значениями в этих гипотезах для 40.

# Алгоритмы построения бикликового покрытия

## Эвристический алгоритм

Рассмотрим следующий эвристический алгоритм для задачи о бикликовом покрытии, основанный на эвристике Келлермана для задачи о покрытии графа кликами. На вход подается граф с множеством вершин V = {1, 2, ... , n}, имеющий m ребер.

Алгоритм начинает работу с пустого бикликового покрытия и последовательно для i = 1, 2, ... , n дополняет бикликовое покрытие так, чтобы покрыть все ребра {i, j} графа G такие, что j < i. В случае, когда нет ребра между текущей обрабатываемой вершиной i и вершинами множества W уже обработанных вершин смежных с i, то создается новая биклика, которая состоит из одной вершины i. Иначе пытаемся добавить вершину i в существующие биклики. После этого возможно, что еще остаются непокрытые ребра соединяющие вершину i и вершины W. Чтобы покрыть эти ребра, мы создаем новую биклику — звезду с центром в вершине i, которая включает все непокрытые ребра между i и W.

**Алгоритм 1**: Эвристика для решения задачи о покрытии графа бикликами.

*Вход* : Граф G = ({1, 2, ... , n}, E ) без изолированных вершин.

*Выход*: Бикликовое покрытие K(X1, Y1), K (X2, Y2), ... , K (Xk, Yk) графа G.

**begin**

k ← 0; //*число сгенерированных биклик*

**for** i = 1, ... , n **do**

//*инвариант цикла: K(X1, Y1), ... , K (Xk, Yk) покрывают все ребра графа G*

//*инцидентные вершинам u, w < i ;*

W ← {j : j < i, {i, j } ∈ E};

**if** W = ∅ **then**

k ← k + 1;

Xk ← {i};

Yk ← ∅;

**else**

//*пробуем добавить вершину i в существующие биклики;*

U ← ∅; //*множество вершин j смежных с i таких, что {i, j} уже покрыто*

**for** l = 1, 2, ... , k **do**

**if** Xl ⊆ W and Xl ≠ ∅ **then**

Yl ← Yl ∪ {i};

U ← U ∪ Xl;

**else** **if** Yl ⊆ W **and** Yl ≠ ∅ **then**

Xl ← Xl ∪ {i};

U ← U ∪ Yl;

**if** W = ∅ **then** break;

W ← W \ U ;

**if** W ≠ ∅ **then**

k ← k + 1;

Xk ← {i};

Yk ← W;

**end**

Выполнение алгоритма прерывается, если в ходе работы построено m или более биклик, так как в таком случае мы можем просто взять бикликовое покрытие, которое состоит из всех ребер графа. Время работы алгоритма 1 составляет O(nm2).

Эвристический алгоритм 1 может быть улучшен с помощью добавления шага, который исключает избыточные биклики из бикликового покрытия графа построенного алгоритмом 1.

Предположим, что K(X1, Y1), K (X2, Y2), ... , K (Xk, Yk) — бикликовое покрытие графа, которое построено алгоритмом 1. Рассматриваем, последовательно, каждую биклику K(Xi, Yi). Если каждое ребро содержащееся в K(Xi, Yi) покрывается остальными бикликами из покрытия, то такая биклика избыточна и ее можно исключить из покрытия.

## Последовательно-параллельные графы

### 4.2.1 Общие сведения

Граф называется последовательно-параллельным, если он не содержит полного графа K4 в качестве минора; эквивалентно, если он не имеет ни одного подграфа гомеоморфного графу K4. Поскольку оба графа K5 и K3,3 содержат подграфы гомеоморфные графу K4, то по теореме Куротовского последовательно-параллельные графы являются планарными.

Класс последовательно-параллельных графов можно определить как рекурсивно-порождаемый класс графов в множестве двухтерминальных графов. При формальном определении используется операция стягивания вершин.

Пусть G — граф и u ∈ V (G), v ∈ V (G). Операция стягивания вершин u и v по опре-делению эквивалентна следующей процедуре. Создается новая вершина x. Каждое ребро uw инцидентное u удаляется и добавляется новое ребро xw. Каждое ребро vy инцидентное v удаляется и добавляется новое ребро xy. Наконец удаляются изолированные вершины u и v. Операция стягивания обозначается следующим образом: x = u ◦ v.

Граф называется двухтерминальным, если в нем выделены две различные вершины, на-зываемые терминалами. Пара терминалов упорядоченная, поэтому часто первый терминал называют левым, а второй — правым. Двухтерминальный граф обозначают тройкой (G, s, t), где G — граф, s ∈ V (G) — левый терминал и t ∈ V (G) — правый терминал.

Последовательной композицией двух или более двухтерминальных графов называется операция, на вход которой подаются r ≥ 2 двухтерминальных графов (G1, s1, t1), ... , (Gr, sr, tr), а выходом является новый двухтерминальных граф (G, s, t), который получается в результате объединения графов G1, ... , Gr , стягивания вершины si+1 с вершиной ti для всех i, 1 ≤ i < r (все вершины zi = si+1 ◦ ti не имеют терминального статуса), и установки терминального статуса для терминалов: s = s1 и t = tr.

Параллельной композицией двух или более двухтерминальных графов называется опера-ция, на вход которой подаются r ≥ 2 двухтерминальных графов (G1, s1, t1), ... , (Gr, sr , tr), а выходом является новый двухтерминальный граф (G, s, t), который получается в результате объединения графов G1, ... , Gr , стягивания всех вершин s1, ... , sr в одну вершину s, которой присваивается статус левого терминала, и стягивания всех вершин t1, ... , tr в одну вершину t, которой присваивается статус правого терминала.

**Определение**. Двухтерминальный граф (G, s, t) называется последовательно-параллельным графом тогда и только тогда, когда выполняется одно из следующих утверждений:

* Граф (G, s, t) является базовым последовательно-параллельным графом, состоящим из двух вершин s и t и одного ребра между ними.
* Граф (G, s, t) образуется в результате последовательной или параллельной композиции r ≥ 2 последовательно-параллельных графов.

Отметим, что мы получим эквивалентный класс графов, если в определении использовать только двухоперандные последовательные и параллельные композиционные операции. Говорят, что граф G является последовательно-параллельным графом тогда и только тогда, когда существуют вершины s, t ∈ V (G) такие, что (G, s, t) — двухтерминальный последовательно-параллельный граф.

Декомпозиция последовательно-параллельного графа (G, s, t) на последовательные и параллельные композиции представляется sp-деревом TG. Sp-дерево является корневым деревом, в котором имеются узлы трех типов: p-узлы, s-узлы и листья. Каждый узел имеет метку в виде упорядоченной пары (u, v) вершин графа G. Каждому узлу α sp -дерева соответствует уникальный последовательно-параллельный граф (G(α), x, y), где G(α) — подграф графа G, и (x, y) — метка узла α. Корень дерева имеет метку (s, t) и соответствует графу (G, s, t). Листья дерева соответствуют базовым последовательно-параллельным графам, которые представляют ребра графа G (существует взаимно-однозначное соответствие между листьями sp -дерева TG и ребрами E(G)).

Внутренние узлы являются либо s-узлами (такой узел представляет последовательную

операцию), либо p-узлами (такой узел представляет параллельную операцию). Сыновья последовательных узлов упорядочены, в то время как сыновья параллельных узлов не являются упорядоченными. Последовательно-параллельный граф, ассоциируемый с s-узлом α — это граф, который получается в результате последовательной композиции последовательно-параллельных графов, ассоциированных с сыновьями узла α, причем порядок сынов задает такой порядок, в котором выполняются последовательные композиции. Последовательно-параллельный граф, ассоциируемый с p-узлом β — это граф, который получается в результате параллельной композиции последовательно-параллельных графов, ассоциированных с сыновьями узла β. Узел, который является сыном p-узла, имеет такую же метку, как и его родитель. Граф, ассоциируемый с узлом α, sp-дерева будем обозначать через G(α).

Отметим, что последовательно-параллельный граф может иметь различные sp-деревья. Sp-дерево называется бинарным sp-деревом, если каждый внутренний узел имеет двух сынов. Очевидно, что любой последовательно-параллельный граф имеет бинарное sp-дерево. Sp-дерево последовательно-параллельного графа (G, s, t) называется минимальным sp-деревом, если каждый его p-узел имеет в качестве сынов только s-узлы и листья, а каждый его s-узел имеет в качестве сынов только p-узлы и листья. Для каждого последовательно-параллельного графа (G, s, t) можно построить минимальное sp-дерево из любого sp-дерева для этого графа, выполнив следующий алгоритм.

Пока в дереве существует s-узел с сыном s-узлом или p-узел с сыном p-узлом выполнять следующее: если s-узел α имеет в качестве сына s-узел β, то стянуть ребро между ними и изменить метку; если p-узел имеет в качестве сына p-узел, то стянуть ребро между ними.

Ориентированные последовательно-параллельные графы определяются аналогично не ориентированным. Различие заключается в том, что в ориентированном случае базовым графом является граф с двумя вершинами s и t и ориентированным ребром от левого терминала s к правому — t. Ориентированные последовательно-параллельные графы являются ациклическими и каждая его вершина лежит на пути из левого терминала в правый.

Если sp-дерево последовательно-параллельного графа задано, то многие задачи на графах могут быть решены за линейное время (от числа ребер).

Двухтерминальные графы можно преобразовывать к такого же вида графам с меньшим

числом вершин и ребер, используя две операции — последовательную и параллельную редукции.

Последовательной редукцией двухтерминального графа (G, s, t) называется операция, которая удаляет вершину v ∈ V (G) степени два из G, v ≠ s, t, и добавляет ребро между соседями вершины v. Параллельной редукцией двухтерминального графа (G, s, t) называется операция, которая удаляет ребро e ∈ E(G) между вершинами u и v, которые являются концами двух или более кратных ребер.

### 4.2.2 Узлы бинарного sp -дерева и биклики последовательно-параллельного графа.

Два графа G и H называются изоморфными, обозначается это так: G ∼= H, если существует такая биекция φ : V (G) → V (H), что uv ∈ E(G) тогда и только тогда, когда

φ(u) φ(v) ∈ E(H).

Пусть G — последовательно-параллельный граф и пусть T — бинарное sp-дерево для G. Для каждого полного двудольного подграфа K(X, Y) графа G можно найти узел α в дереве T, для которого выполняется K(X, Y) ⊆ G(α), и который расположен на наибольшем расстоянии от корня. Следующая теорема устанавливает зависимость между типом узла α и классом полных двудольных графов, к которому принадлежит граф K(X, Y).

**Теорема 5**. Пусть G — последовательно-параллельный граф, и K(X, Y) — полный двудольный подграф графа G. Пусть T — бинарное sp-дерево для G. Пусть α — узел в дереве T, для которого выполняется K(X, Y) ⊆ G(α), и который расположен на наибольшем расстоянии от корня. В зависимости от типа узла α и класса полных двудольных графов, к которому принадлежит граф K(X, Y), возможны только следующие случаи:

**Случай 1.** α — лист и K(X, Y) K1,1. В этом случае

X = {u}, Y = {v}, (1)

где {u, v} ⊆ V (G) и (u, v) ∈ E(G).

**Случай 2.** α — s-узел и K(X, Y) K1,m , где m ≥ 2. Пусть β1— левый, а β2 — правый сын узла α. Пусть (G(α), s, t), (G(β1), s, z ) и (G(β2), z, t) — последовательно-параллельные графы, соответствующие узлам: α, β1 и β2, т.е. (G(α), s, t) = (G(β1), s, z ) ⊙s ⊙s (G(β2), z, t). Тогда

X = {z}, Y = {u1, ... , ua, v1, ... , vb} (2)

и существуют полные двудольные графы K1(X1, Y1) K1,a , K2(X2, Y2) K1,b, a ≥ 1, b ≥ 1, такие, что K1(X1, Y1) ⊆ G(β1), X1 = {z}, Y1 = {u1, ... , ua}, K2(X2, Y2) ⊆ G(β2), X2 = {z}, Y

2 = {v1, ... , vb}, m = a + b.

**Случай 3**. α — p-узел и K(X, Y) K1,m, где m ≥ 2. Пусть β1 и β2 — сыновья узла α. Пусть (G(α), s, t), (G(β1), s, t) и (G(β2), s, t) — последовательно-параллельные графы, соответствующие узлам α, β1 и β2, т.е. (G(α), s, t) = (G(β1), s, t) ⊙p (G(β2), s, t). Тогда

биклика K(X, Y) — это звезда, центром которой может быть либо вершина s ∈ V (G(α)),

либо вершина t ∈ V (G(α)). Точнее, возможно только следующее:

X = {z}, Y = {u1, ... , ua, v1, ... , vb}, (3)

где

z ∈ {s, t}, {z, u1, ... , ua, v1, ... , vb} ⊆ V (G)

и

E(K) = {(z, ui) : 1 ≤ i ≤ a} ∪ {(z, vi) : 1 ≤ i ≤ b} ⊆ E(G);

и существуют полные двудольные графы K1(X1, Y1) K1,a , K2(X2, Y2) K1,b , a ≥ 1, b ≥ 1, такие, что K1(X1, Y1) ⊆ G(β1), X1 = {z}, Y1 = {u1, ... , ua}, K2(X2, Y2) ⊆ G(β2), X2 = {z},Y2 = {v1, ... , vb}, m = a + b.

**Случай 4.** α — p-узел и K(X, Y) K2,m, где m ≥ 2. Пусть β1 и β2 — сыновья узла α. Пусть (G(α), s, t), (G(β1), s, t) и (G(β2), s, t) — последовательно-параллельные графы,

соответствующие узлам: α, β1 и β2, т.е. (G(α), s, t) = (G(β1), s, t) ⊙p (G(β2), s, t). Тогда

для подграфа K(X, Y) существуют три возможности:

**Случай 4.1.**

X = {s, t}, Y = {u1, ... , ua, v1, ... , vb}, (4)

где

{s, t, u1, ... , ua, v1, ... , vb} ⊆ V (G)

и

E(K) = {(s, ui) : 1 ≤ i ≤ a} ∪ {(t, ui) : 1 ≤ i ≤ a} ∪ {(s, vi) : 1 ≤ i ≤ b} ∪

∪ {(t, vi) : 1 ≤ i ≤ b} ⊆ E(G);

и существуют полные двудольные графы K1(X1, Y1) K2,a , K2(X2, Y2) K2,b,

a ≥ 1, b ≥ 1, такие, что

K1(X1, Y1) ⊆ G(β1), X1 = {s, t}, Y1 = {u1, ... , ua}, K2(X2, Y2) ⊆ G(β2),

X2 = {s, t}, Y2 = {v1, ... , vb}, m = a + b;

**Случай 4.2.**

X = {s, u}, Y = {t, v1, ... , vm−1}, (5)

где

{s, t, u, v1, ... , vm−1} ⊆ V (G)

и

E(K) = {(s, t)} ∪ {(t, u)} ∪ {(s, vi) : 1 ≤ i ≤ m − 1} ∪ {(u, vi) : 1 ≤ i ≤ m − 1} ⊆ E(G);

и существует подграф G1 = K({s, u}, {v1, ... , vm−1}) + (u, t) графа G(β1); в графе G(β2)

существует ребро (s, t) ∈ E(G(β2));

**Случай 4.3.**

X = {u, t}, Y = {s, v1, ... , vm−1}, (6)

где

{s, t, u, v1, ... , vm−1} ⊆ V (G)

и

E(K) = {(s, t)} ∪ {(s, u)} ∪ {(u, vi) : 1 ≤ i ≤ m − 1} ∪ {(t, vi) : 1 ≤ i ≤ m − 1} ⊆ E(G);

и существует подграф G2 = K({u, t}, {v1, ... , vm−1}) + (u, s) графа G(β1); в графе G(β2)

существует ребро (s, t) ∈ E(G(β2)).

**Доказательство.** Корню ρ любого sp-дерева последовательно-параллельного графа

(G, s, t) соответствует сам граф: G(ρ) = (G, s, t). Пусть T — бинарное sp-дерево для (G, s, t),

и пусть γ — лист этого дерева. Пусть γ = γ1, ... , γk = ρ — цепь, соединяющая лист γ с корнем ρ. Тогда выполняется условие:

G(γ1) ⊂ G(γ2) ⊂ ... ⊂ G(γk).

Следовательно, для любого подграфа K ⊆ G в дереве T всегда существует узел α, для

которого выполняется K ⊆ G(α), и который расположен на наибольшем расстоянии от корня.

Рассмотрим все случаи, приведенные в формулировке теоремы.

**Случай 1**. Равенство (1) следует из определения sp -дерева.

**Случай 2.** (G(α), s, t) = (G(β1), s, z ) ⊙s (G(β2), z, t). Отметим, что при выполнении этой операции вершина z ∈ V (G(β1)) стягивается с вершиной z ∈ V (G(β2)) и в результате получается вершина z ∈ V (G(α)). Надеемся, что в данном случае и в последующем идентификация стягиваемых вершин одной буквой не нарушает ясности изложения. Выполняется неравенство E(K) ∩ E(G(β1)) ≠ ∅, поскольку в противном случае граф K(X,Y) являлся бы подграфом графа G(β2), что противоречит условию: “ α — узел в дереве T, для которого выполняется K(X, Y ) ⊆ G(α), и который расположен на наибольшем расстоянии от корня”. Аналогично, по этой же причине, E(K) ∩ E(G(β2)) ≠ ∅. Пусть a = |E(K) ∩ E(G(β1))|, b = |E(K) ∩ E(G(β2))|, c = degG(β1)(z) и d = degG(β2)(z).

Звезда

S1 = ({z}, {u1, ... , ua})

является подграфом графа G(β1), а звезда

S2= ({z}, {v1, ... , vb})

является подграфом графа G(β2). Звезда K({z}, {u1, ... , ua, v1, ... , vb}) является подграфом графа G(α). Следовательно, m = a + b и (2) верно.

**Случай 3.** (G(α), s, t) = (G(β1), s, t) ⊙p (G(β2), s, t). Пусть

K(X, Y ) = K({y}, {x1, ... , xm}).

Тогда y /∈ V (G(β1)) \ {s, t}, поскольку в противном случае граф K(X, Y) являлся бы подграфом графа G(β1), что противоречит условию: “ α — узел в дереве T, для которого выполняется K(X, Y) ⊆ G(α), и который расположен на наибольшем расстоянии от корня”. Случаи y = s и y = t возможны. Рассмотрим только случай y = s, поскольку все выкладки для случая y = t аналогичны.

Звезда

S1 = K({s}, {u1, ... , ua})

является подграфом графа G(β1), а звезда

S2 = K({s}, {v1, ... , vb})

является подграфом графа G(β2). Звезда K(X, Y ) = K({s}, {u1, ... , ua, v1, ... , vb}) является

подграфом графа G(α). Следовательно, m = a + b и (3) верно.

**Случай 4.** (G(α), s, t) = (G(β1), s, t) ⊙p (G(β2), s, t). Пусть

K(X, Y ) = K({x, y}, {z1, ... , zm}).

Отметим, что следующие соотношения:

{x, y} ∩ {s, t} ≠ ∅, {z1, ... , zm} ̸⊆ V (G(β1)), {z1, ... , zm} ̸⊆ V (G(β2))

выполняются, поскольку в противном случае получаем противоречие условию: “ α — узел

в дереве T, для которого выполняется K ⊆ G(α), и который расположен на наибольшем

расстоянии от корня”.

В случае {x, y} ∩ {s, t} = {s, t} полный двудольный подграф

K(X, Y ) = K({s, t}, {u1, ... , ua, v1, ... , vb})

является подграфом графа G(α). Существуют не пустые полные двудольные подграфы:

K1(X1, Y1) = K({s, t}, {u1, ... , ua})

и

K2(X2, Y2) = K({s, t}, {v1, ... , vb})

такие, что K1(X1, Y1) ⊆ G(β1), K2(X2, Y2) ⊆ G(β2). Следовательно, m = a + b и (4) верно.

В случае {x, y} ∩ {s, t} = {s} полный двудольный подграф

K(X, Y ) = K({s, u}, {t, v1, ... , vm−1})

является подграфом графа G(α). Существует граф

K1(X1, Y1) = K({s, u}, {v1, ... , vm−1}) + (u, t),

являющийся подграфом графа G(β1). В графе G(β2) существует ребро (s, t) ∈ E(G(β2)).

Следовательно, (5) верно.

В случае {x, y} ∩ {s, t} = {t} полный двудольный подграф

K(X, Y ) = K({u, t}, {s, v1, ... , vm−1})

является подграфом графа G(α). Существует граф

K1(X1, Y1) = K({u, t}, {v1, ... , vm−1}) + (u, s),

являющийся подграфом графа G(β1). В графе G(β2) существует ребро (s, t) ∈ E(G(β2)).

Следовательно, (6) верно.

*Теорема 5 доказана.*

Пусть (G, s, t) — последовательно-параллельный граф с терминалами s и t. Пусть

C = {G1, ... , Gm} — покрытие графа G, т.е. G = G1 ∪ ... ∪ Gm. Рассмотрим шесть свойств,

которыми может обладать покрытие:

(P1) покрытие C содержит подграф Gi = K({s}, {t}) ∼= K1,1,

(P2) покрытие C содержит подграф Gi = K({s}, U) ∼= K1,|U|,

(P3) покрытие C содержит подграф Gi = K({t}, W ) ∼= K1,|W|,

(P4) покрытие C содержит подграф Gi = K({s, t}, Z) ∼= K2,|Z|,

(P5) покрытие C содержит подграф Gi = K({s, v}, A) + (v, t),

(P6) покрытие C содержит подграф Gi = K({u, t}, B ) + (u, s),

где U, W, Z, A и B — некоторые подмножества вершин графа G.

Свойства попарно совместимы, поэтому для покрытия C можно определить характеристический вектор c() = c1c2c3c4c5c6, где

ci =

Будем говорить, что покрытие обладает свойством c = c1c2c3c4c5c6, если оно обладает всеми свойствами (Pi), для которых ci = 1, и не обладает свойствами (Pj), для которых cj = 0, т.е., если c = c().

Для каждого последовательно-параллельного графа (G, s, t) определим 26

классов эквивалентности M(G, c) покрытий. Класс M(G, c) состоит из всех покрытий C = {G1, ... , Gm}, удовлетворяющих следующим условиям:

(а) для любого i = 1, ... , m либо существует такое целое число l ≥ 1, что Gi ∼= K1,l, либо существует такое целое число f ≥ 1, что Gi ∼= K2,f, либо существует такое подмножество вершин W ⊂ V (G), что Gi = K({s, u}, W) + (u, t), либо существует такое подмножество вершин W′ ⊂ V (G), что Gi = K({u, t}, W′) + (u, s);

(б) обладает свойством c;

(в) имеет наименьшее число подграфов среди покрытий графа (G, s, t), обладающих свойствами (а) и (б).

Пусть (G, s, t) — последовательно-параллельный граф, c ∈ {0, 1}6, M(G, c) ≠ ∅ и

∈ M(G, c), тогда:

если c1 = 1, то покрытие содержит только один подграф K({s}, {t}) ∼= K1,1,

если c2 = 1, то покрытие содержит только один подграф K({s}, U) ∼= K1,|U|,

если c3 = 1, то покрытие содержит только один подграф K({t}, W) ∼= K1,|W|,

если c4 = 1, то покрытие содержит только один подграф K({s, t}, Z) ∼= K2,|Z|,

если c5 = 1, то покрытие содержит f > 0 подграфов:

K({s, v1}, A1) + (v1, t), ... , K ({s, vf}, Af) + (vt , t),

если c6 = 1, то покрытие C содержит l > 0 подграфов:

K({u1, t}, B1) + (u1, s), ... , K ({ul , t}, Bl) + (ul, s).

Каждый такой подграф будем называть подграфом, реализующим (соответствующее)

свойство ci.

Пусть (G, s, t) — последовательно-параллельный граф, c ∈ {0, 1}6 и ∈ M(G, c), тогда возможны следующие преобразования, уменьшающие число подграфов в покрытии.

(П1) если c1 ∧ c2 = 1 и подграф G′ ∈ реализует свойство c1, а подграф G′′ ∈ реализует свойство c2, то R = \ {G′, G′′} ∪ {G′ ∪ G′′} — покрытие, имеющее свойство r(R) = r1r2r3r4r5r6, где r1 = 0, r2= 1, ri = ci, для i ∈ {3, 4, 5, 6}.

(П2) если c1 ∧ c3 = 1 и подграф G′ ∈ реализует свойство c1, а подграф G′′ ∈ реализует свойство c3, то R = \ {G′, G′′} ∪ {G′ ∪ G′′} — покрытие, имеющее свойство

r(R) = r1r2r3r4r5r6, где r1= 0, r3= 1, ri = ci, для i ∈ {2, 4, 5, 6}.

(П3) если c1 ∧ (c5 ∨ c6) = 1, то пусть G′ ∈ — подграф реализующий свойство c1, {, ... , } ⊂ — подграфы реализующие свойство c5 и {, ... , } ⊂ — подграфы

реализующие свойство c6. Тогда покрытие R = \ {G′} имеет свойство r(R) = r1r2r3r4r5r6, где r1= 0, ri = ci, для i ∈ {2, 3, 4, 5, 6}.

Ясно, что после любого из преобразований (П1) – (П3) |R| = || − 1.

Обозначим через h(G, c) число подграфов в покрытии из класса M(G, c). Для каждого

узла α sp-дерева числа h(G(α), c) можно хранить в массиве, имеющем 26 элементов.

Пусть a ∈ {0, 1}6 и b ∈ {0, 1}6. Пусть G, H, D — такие последовательно-параллельные графы, что либо G = H ⊙s D, либо G = H ⊙p D. Пусть CH = {H1, ..., Hl} и CD = {D1, ..., Dk} такие покрытия графов H и D соответственно, что CH ∈ M(H, a), CD ∈ M(D, b). Ясно, что CH CD — покрытие графа G. Часто из покрытий CH и CD можно построить покрытие CG = {G1, ..., Gm}, которое удовлетворяет условиям (а) и (б), и такое, что при соответствующих упорядочениях подграфов в покрытиях выполняется условие:

G1 = H1 ∪ D1, ..., Gt = Hr ∪ Dr {Gr+1, ..., Gm} = {Hr+1, ..., Hl} ∪ {Dr+1, ... , Dk}. (7)

Если такое покрытие существует, то m = l + k − r = h(H, a) + h(D, b) − r. Ясно, что в

таком случае нас интересует максимальное значение r.

Если G = H ⊙s D, то возможны следующие преобразования, которые можно использовать для построения покрытия вида (7), имеющего максимальное значение r.

(П4) если a1∧b1 = 1, то пусть H1 ∈ CH — подграф реализующий свойство a1, и D1 ∈ CD —

подграф реализующий свойство b1. Тогда покрытие {H1 ∪ D1} ∪ {H2, ..., Hl} ∪ {D2, ..., Dk} имеет свойство c(C) = c1c2c3c4c5c6, где c1= 0, c2 = a2, c3 = b3, c4= 1, c5 = c6 = 0.

(П5) если a1∧b2 = 1, то пусть H1∈ CH — подграф реализующий свойство a1, и D1∈CD — подграф реализующий свойство b2. Тогда покрытие {H1 ∪ D1} ∪ {H2, ..., Hl} ∪ {D2, ..., Dk} имеет свойство c(C) = c1c2c3c4c5c6, где c1= 0, c2 = a2, c3 = b3, c4 = c5 = c6 = 0.

(П6) если a1∧b4 = 1, то пусть H1∈ CH — подграф реализующий свойство a1, и D1∈ CD — подграф реализующий свойство b4. Тогда покрытие {H1 ∪ D1} ∪ {H2, ..., Hl} ∪ {D2, ..., Dk} имеет свойство c(C) = c1c2c3c4c5c6, где c1= 0, c2 = a2, c3 = b3, c4 = c5 = 0, c6 = 1.

(П7) если a3∧b1 = 1, то пусть H1∈ CH — подграф реализующий свойство a3, и D1∈ CD — подграф реализующий свойство b1. Тогда покрытие {H1 ∪ D1} ∪ {H2, ..., Hl} ∪ {D2, ..., Dk} имеет свойство c(C ) = c1c2c3c4c5c6, где c1= 0, c2 = a2, c3 = b3, c4 = c5 = c6 = 0.

(П8) если a3∧b2 = 1, то пусть H1∈ CH — подграф реализующий свойство a3, и D1∈ CD — подграф реализующий свойство b2. Тогда покрытие {H1 ∪ D1} ∪ {H2, ... , Hl} ∪ {D2, ... , Dk} имеет свойство c(C) = c1c2c3c4c5c6, где c1 = 0, c2 = a2, c3 = b3, c4 = c5 = c6 = 0.

(П9) если a4∧b2 = 1, то пусть H1∈ CH — подграф реализующий свойство a4, и D1∈ CD —

подграф реализующий свойство b2. Тогда покрытие {H1 ∪ D1}∪{H2, ... , Hl}∪{D2, ... , Dk} имеет свойство c(C) = c1c2c3c4c5c6, где c1 = 0, c2 = a2, c3 = b3, c4= 0, c5 = 1, c6 = 0.

Если G = H ⊙p D, то возможны следующие преобразования, которые можно использовать для построения покрытия вида (7), имеющего максимальное значение r.

(П10) если a2∧b2 = 1, то пусть H1 ∈ CH — подграф реализующий свойство a2, и D1∈CD— подграф реализующий свойство b2. Тогда покрытие {H1 ∪ D1}∪{H2, ... , Hl}∪{D2, ... , Dk} имеет свойство c(C) = c1c2c3c4c5c6, где c1 = a1 ∨ b1, c2 = 1, c3 = a3 ∨ b3, c4 = a4 ∨ b4, c5 = a5 ∨ b5, c6 = a6 ∨ b6.

(П11) если a3∧b3 = 1, то пусть H1 ∈ CH — подграф реализующий свойство a3, и D1∈CD— подграф реализующий свойство b3. Тогда покрытие {H1 ∪ D1}∪{H2, ... , Hl}∪{D2, ... , Dk} имеет свойство c(C) = c1c2c3c4c5c6, где c3 = 1 и ci = ai ∨ bi, для i ∈ {1, 2, 4, 5, 6}.

(П12) если a4∧b4 = 1, то пусть H1 ∈ CH — подграф реализующий свойство a4, и D1∈CD— подграф реализующий свойство b4. Тогда покрытие {H1 ∪ D1}∪{H2, ... , Hl}∪{D2, ... , Dk} имеет свойство c(C) = c1c2c3c4c5c6, где c4= 1 и ci = ai ∨ bi, для i ∈ {1, 2, 3, 5, 6}.

Ясно, что после выполнения любого из преобразований (П4) – (П12) получаем покрытие с числом подграфов: |C| = h(H, a) + h(D, b) − 1.

Некоторые из преобразований (П1) – (П12) можно комбинировать, чтобы получить максимальное значение параметра r в формуле (7). Проверка на корректность такого рода комбинаций реализована в виде двух функций: SC (a, b, c) и P C (a, b, c).

На множестве всех возможных троек (a, b, c) ∈ {0, 1}6× {0, 1}6× {0, 1}6 определим функцию SC (a, b, c) следующим образом: SC (a, b, c) = r, если существуют такие последовательно-параллельные графы G, H, D и их покрытия CG = {G1, ... , Gm}, CH = {H1, ... , Hl}, CD = {D1, ... , Dr}, что G = H ⊙s D, CG ∈ M(G, c), CH ∈ M(H, a), CD ∈ M(D, b) и r — максимальное значение, для которого G1 = H1 ∪ D1, ... , Gr = Hr ∪ Dr , {Gr+1, ... , Gm} = {Hr+1, ... , Hl} ∪ {Dr+1, ... , De}. Если из CH и CD невозможно построить покрытие вида (7) со свойством c, то полагаем SC(a, b, c) = −1.

Аналогично, на множестве всех возможных троек (a, b, c) ∈ {0, 1}6× {0, 1}6× {0, 1}6 определим функцию PC (a, b, c) следующим образом: PC (a, b, c) = r, если существуют такие последовательно-параллельные графы G, H, D и их покрытия CG = {G1, ... , Gm}, CH = {H1, ... , Hl}, CD = {D1, ... , Dr}, что G = H ⊙p D, CG ∈ M(G, c), CH ∈ M(H, a), CD ∈ M(D, b) и r — максимальное значение, для которого G1 = H1 ∪ D1, ... , Gr = Hr ∪ Dr , {Gr+1, ... , Gm} = {Hr+1, ... , Hl} ∪ {Dr+1, ... , De}. Если из CH и CD невозможно построить покрытие вида (7) со свойством c, то полагаем PC(a, b, c) = −1.

### 4.2.3 Алгоритм для нахождения числа бикликового покрытия последовательно-параллельного графа

Приведем рекурсивный алгоритм для нахождения числа бикликового покрытия последовательно-параллельного графа.

**Алгоритм 2**:

*Вход*: Простой последовательно-параллельный граф (G, s, t). (Вершины s и t являются

терминалами графа G, и его бинарное дерево декомпозиции.)

*Выход*: Число бикликового покрытия bc(G) и массив чисел h(G, 000000), ... , h(G, 111111), такой, что элемент h(G, c) содержит число подграфов в покрытии из класса M(G, c), если M(G, c) ≠ ∅, и h(G, c) = |E(G)| + 1, если M(G, c) = ∅.

**begin**

**if** |E| = 1 **then**

h(G, 100000) := h(G, 010000) := h(G, 001000) := 1;

**for** all c ∈ {0, 1}6 \ {100000, 010000, 001000} **do**

h(G, c) := m(G);

**else** {

**for** all c ∈ {0, 1}6 **do**

h(G, c) := m(G);

**if** G = H ⊙s D **then**

**for** all a ∈ {0, 1}6 **do**

**if** h(H, a) < m(H) **then**

**for** all b ∈ {0, 1}6 **do**

**if** h(D, b) < m(D) **then**

**for** all c ∈ {0, 1}6 **do**

**if** SC (a, b, c) ≥ 0 **then**

**if** h(G, c) > h(H, a) + h(D, b) − SC(a, b, c) **then**

h(G, c) := h(H, a) + h(D, b) − SC(a, b, c);

**else** **if** G = H ⊙p D **then**

**for** all a ∈ {0, 1}6 **do**

**if** h(H, a) < m(H) **then**

**for** all b ∈ {0, 1}6 **do**

**if** h(D, b) < m(D) **then**

**for** all c ∈ {0, 1}6 **do**

**if** PC(a, b, c) ≥ 0 **then**

**if** h(G, c) > h(H, a) + h(D, b) − PC(a, b, c) **then**

h(G, c) := h(H, a) + h(D, b) − PC(a, b, c);

}

**for** all c ∈ {0, 1}6 **do**

**if** ¬((c5 ∨ c6) ∧ ¬c1)) **then**

bc(G) = min{bc(G), h(G, c)};

**end**

В алгоритме используются функции: m(G), SC(a, b, c) и PC(a, b, c). Функция m(G) =

= |E(G)| + 1. Приведем описание функций SC(a, b, c) и PC(a, b, c).

**Функция** SC(a, b, c)

*Вход*: три характеристических векторы a, b, c свойств бикликовых покрытий.

*Выход*: наибольшее число r, такое, что при операции ⊙s из покрытия со свойством a и

покрытия со свойством b можно построить покрытие вида (7) со свойством c; −1 иначе.

**begin**

s := 0;

**if** a2 ∧ ¬c2 ∨ b3 ∧ ¬c3 ∨ a5 ∨ a6 ∨ b5 ∨ b6 ∨

c1 ∨ c2 ∧ ¬a2 ∨ c3 ∧ ¬b3 ∨ c4 ∧ ¬(a1 ∧ b1) **then**

**return** −1;

**if** a4 **then** s := s + 1;

**if** c5 **then** **if** a4 ∧ b1 **then** s := s − 1 **else** **return** −1;

**if** b4 **then** s := s + 1;

**if** c6 **then** **if** a1 ∧ b4 **then** s := s − 1 **else** **return** −1;

**if** a1 ∧ ¬(c2 ∨ c6 ∨ c4 ∧ b1) ∨ a3 ∨ b1 ∧ ¬(c3 ∨ c5 ∨ c4 ∧ a1) ∨ b2 **then**

s := s + 1;

**return** ω(a) + ω(b) − ω(c) − s;

**end**

**Функция** PC(a, b, c)

*Вход*: Три характеристических вектора a, b, c свойств бикликовых покрытий.

*Выход*: наибольшее число r, такое, что при операции ⊙p из покрытия со свойством a и

покрытия со свойством b можно построить покрытие вида (7) со свойством c; −1 иначе.

**begin**

**if** a1 ∧ b1 ∨ (a1 ∨ b1) ∧ ¬(c1 ∨ c2 ∨ c3 ∨ a5 ∨ a6 ∨ b5 ∨ b6) ∨

(a2 ∨ b2) ∧ ¬c2 ∨ (a3 ∨ b3) ∧ ¬c3 ∨

(a4 ∨ b4) ∧ ¬c4 ∨ ¬(a1 ∨ b1) ∧ ((a5 ∨ b5) ∧ ¬c5 ∨ (a6 ∨ b6) ∧ ¬c6) ∨

c1 ∧ ¬(a1 ∨ b1) ∨ c2 ∧ ¬(a1 ∨ b1 ∨ a2 ∨ b2) ∨ c3 ∧ ¬(a1 ∨ b1 ∨ a3 ∨ b3) ∨

c4 ∧ ¬(a4 ∨ b4) ∨ c5 ∧ ¬(a5 ∨ b5) ∨ c6 ∧ ¬(a6 ∨ b6) **then**

**return** -1;

**else**

s := **if** c1 ∧ (a5 ∨ a6 ∨ b5 ∨ b6) **then** 1 **else** 0;

**return** ω(a&111100) + ω(b&111100) − ω(c&111100) + s;

**end**

Здесь функция ω обозначает операцию взвешивания бинарного вектора, т.е. количество единиц в векторе.

**Теорема 6**. Данный алгоритм для любого последовательно-параллельного графа G вычисляет число бикликового покрытия bc(G) за время O(n), где n = |V (G)|.

*Доказательство*. Алгоритм основан на свойствах, установленных в теореме 4. Покажем корректность вычислений. Алгоритм является рекурсивным. В каждом узле α sp-дерева осуществляется рекурсивное обращение для вычисления числа бикликового покрытия bc(G(α)) и чисел h(G(α), c) для графа G(α), соответствующего узлу α. Корректность этих вычислений докажем структурной индукцией по порядку вычислений в узлах этого sp-дерева. Легко проверить, что операторы в строках 1–4 алгоритма корректно вычисляют числа h(G(α), c) для каждого листа α sp-дерева.

Пусть α — s-узел sp-дерева. Пусть β и γ его сыновья и (G, s, t) = G(α), (H, s1, t1) =

= G(β), (D, s2, t2) = G(γ). Предположим, что для узлов β и γ числа h(G(β), a) и h(G(γ ), b)

вычислены корректно. Покажем, что в строках 6–16 алгоритма корректно вычисляется

массив чисел h(G(α), c).

Отметим, что если h(H, a) < m(H), то класс покрытий M(H, a) не пуст. Следовательно, при обращении к функции SC(a, b, c) характеристические вектора a и b такие, что M(H, a) ≠ ∅ и M(D, b) ≠ ∅.

Пусть CH = {H1, ... , Hl } ∈ M(H, a) и CD = {D1, ... , Dk } ∈ M(D, b), тогда l = h(H, a) и k = h(D, b). Для некоторых значений характеристического вектора c из покрытий CH и CD можно построить покрытие CG = {G1, ... , Gm}, которое удовлетворяет условиям (а) и (б), и такое, что при соответствующих упорядочениях подграфов в покрытиях выполняется (7). Если такое покрытие существует, то m = l + k − t = h(H, a) + h(D, b) − t. Ясно, что в таком случае нас интересует максимальное значение t. Функция SC(a, b, c) отвечает на вопрос: существует ли покрытие CG = {G1, ... , Gm}, удовлетворяющее (7), и если да, то она возвращает максимальное значение t, в случае отрицательного ответа возвращается значение -1.

В условном операторе, записанном в строках 2, 3 функции SC(a, b, c), в качестве условия записана формула, задающая булеву функцию, принимающую значение ИСТИНА в случае не выполнения условий “передачи” свойств, которыми обладают графы H и D, графу G или не выполнения условий “приобретения” свойств графом G.

При выполнении операторов, записанных в строках 5–12, вычисляется значение переменной s, которое используется для корректного вычисления значения, возвращаемого функцией SC(a, b, c).

Таким образом, операторы в строках 14–16 алгоритма корректно вычисляют числа h(G(α), c). Корректность вычислений в строках 17–25 следует из определения параллельной композиции и предположения индукции. По индукции заключаем, что в корневом узле дерева декомпозиции массив чисел h(G, c) вычисляется корректно.

Определим трудоемкость рекурсивного алгоритма. Бинарное дерево декомпозиции (sp-дерево) последовательно-параллельного графа можно построить за линейное время. В каждом узле sp-дерева осуществляется рекурсивное обращение к алгоритму. В sp-дереве m - 1 внутренних узлов и m листьев, где m = |E(G)|. Трудоемкость вычислений, соответствующих одному узлу — O(1). Поскольку в простом последовательно-параллельном графе O(n) ребер, то трудоемкость алгоритма O(n).

*Теорема 6 доказана.*

Этот алгоритм может быть легко модифицирован для построения бикликового покрытия. Вычисляя для каждого узла вектор H, будем сохранять лучший результата h. Будем двигаться по дереву от листьев к корням. Тогда зная для сыновей узла лучшее бикликовое покрытие, легко строится лучшее бикликового покрытия для узла. Для реализации восстановления бикликового покрытия потребуется еще 1 проход по дереву, а сложность операций внутри узла O(1). Таким образом, трудоемкость алгоритма останется O(n).

# Сравнение алгоритмов

Первый алгоритм неточный – он находит бикликовое покрытие, но оно не обязательно будет минимальным. Второй же алгоритм находит минимальное бикликовое покрытие, но только для последовательно-параллельного графа. Поэтому вычислим, насколько хорошо эта эвристика работает на последовательно-параллельных графах.

Генерируем случайное SP-дерево и строим для него соответствующий граф. Далее вычисляем среднее относительное отклонение числа бикликового покрытия, полученного первым алгоритмом, от числа, полученного вторым алгоритмом.

Как можно убедиться (запустив программу), средняя относительная погрешность составляет 20-25%, (хотя в целом погрешность колеблется в пределах 10-40%).

А вот модифицированный алгоритм для данного типа графов нет смысла применять, он работает с такой же точностью, как и обычный эвристический алгоритм.

# Заключение

* Получена оценка сверху для числа бикликового покрытия
* Получены точные значения для графов лестница и квадратная решетка
* Построен приближенный алгоритмы для нахождения бикликового покрытия графа.
* Построен точный алгоритм для нахождения бикликового покрытия последовательно-параллельного графа.
* Написана программа для нахождения максимального паросочетания в двудольном графе, а также для нахождения бикликового покрытия графа.
* Практическим путем установлена точность приближенного алгоритма.

# 

# Список использованной литературы

1. Лепин В.В., Дугинов О.И. О числе бикликового покрытия декартова произведения графов //Тр. Ин-та математики 2013. Т.21. №1. С. 1-9.
2. Amilhastre J., Vilarem M.C., Janssen P. Complexity of minimum biclique cover and minimum biclique decomposition for bipartite domino-free graphs // Disc. App. Math. 1998. V. 86. P. 125–144.
3. Orlin J. Containment in Graph Theory: covering graphs with cliques // Nederl. Akad. Wetensch. Indag. Math. 1977. V. 39. P. 211–218.
4. Лепин В.В. Линейный алгоритм для вычисления числа бикликового покрытия последовательно-параллельного графа //Тр. Ин-та математики 2008. Т.16. №2. С. 1-12.
5. Лепин В.В., Дугинов О.И. Задачи и инварианты, связанные с бикликами и мультикликами графа //Тр. Ин-та математики 2013. Т.21. №2. С. 69-71.