Алгоритм плоской укладки графов

Иринёв Антон, Каширин Виктор

Оглавление

1. Введение 2

2. Основные определения.. 2

3. Задача о плоской укладке 3

4. Теорема (Понтрягин-Куратовский) 3

5. Гамма-алгоритм 3

6. Корректность гамма-алгоритма 6

Литература 10

1. Введение

Среди большого множества задач на графах можно встретить следующую: как изобразить некий граф на плоскости так, чтобы никакие два его ребра не пересекались, т.е. нарисовать исходный граф как граф без самопересечений. Такую задачу принято называть **задачей о плоской укладке графа**.

Область применения алгоритма для решения этой задачи весьма обширна. Хорошим примером может послужить проблема изготовления электронных микросхем. Электрические цепи печатным способом наносятся на плату из изолирующего материала. Так как наносимые цепи не изолированы, то они не должны пересекаться. Отсюда вытекает вопрос, как расположить контакты на схеме, чтобы можно было без пересечений нанести цепи на плату. А если так сделать нельзя, то каким минимальным числом плат (*слоев* графа) можно обойтись.

Мы рассмотрим ответ на первую часть вопроса, т.е. когда и как можно нарисовать граф без самопересечений.

2. Основные определения

Плоский граф это граф, изображенный на плоскости так, что никакие два ребра не имеют общих точек, кроме инцидентной им обоим вершины.

Любой граф, изоморфный плоскому, называется планарным. Говорят, что граф **допускает плоскую укладку**, если его можно изобразить как плоский. Например, граф, показанный на рис.1, плоский.

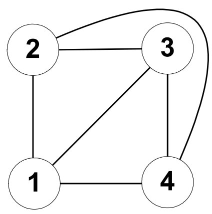

Рис. 1

Существуют также и непланарные графы. На рис. 2 показаны два таких графа: полный пятивершинник и полный двудольный граф. Для них есть специальные обозначения: K_5 и $K_{3,3}$ соответственно.

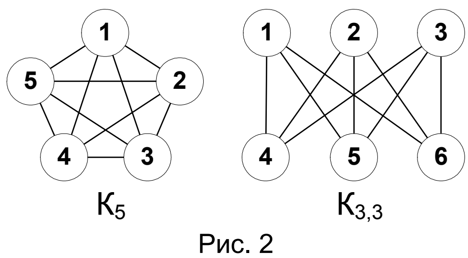

Рис. 2

В планарном графе можно выделять не только вершины и ребра, но и грани.

Грань — это геометрическое место точек, каждые две из которых могут быть соединены жордановой кривой, не пересекающей ребра графа. Всякий плоский граф имеет одну, и притом единственную, неограниченную грань, называемую **внешней гранью**; остальные грани называются **внутренними**.

3. Задача о плоской укладке

Постановка задачи: Определить, является ли граф планарным, и, если да, произвести его плоскую укладку.

Замечание. Если на ребра планарного графа нанести произвольное число вершин степени 2, то он останется планарным. С другой стороны, если на ребра непланарного графа нанести вершины степени 2, то он планарным не станет.

В конце 1920-х годов К. Куратовский (K. Kuratowski) в Польше и Л.С. Понтрягин в СССР независимо доказали интересную теорему, показывающую, что мешает графу быть планарным.

4. Теорема (Понтрягин-Куратовский)

Граф планарен тогда и только тогда, когда он не содержит в качестве частей графы K_5 и $K_{3,3}$ (быть может с добавочными вершинами степени 2).

Мы опустим достаточно сложное доказательство этой теоремы. Заинтересованный читатель сможет найти его в [2] или [3]. Однако не следует думать, что раз подобных проблемных графов всего два, то непланарных графов мало: при росте числа вершин непланарны почти все графы.

Критерий планарности, основанный на теореме Понтрягина-Куратовского не слишком практичен, потому что найти в графе подграфы K_5 и/или $K_{3,3}$ не так просто. Например так называемый граф Петерсена не является планарным (см. рис. 3). Какой граф содержится в нем в качестве части: K_5 или $K_{3,3}$? На поставленный вопрос можно ответить, если убрать ребра (3, 4) и (7, 10), назначить вершины 1, 8 и 9 в одну долю, и 2, 5, 6 в другую, после чего можно увидеть $K_{3,3}$.

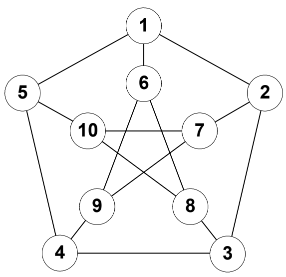

Рис. 3

Рассмотренный критерий очень трудоемок для проверки и представляет лишь теоретический интерес, поэтому для практической работы нужны более конструктивные критерии.

5. Гамма-алгоритм

Для плоской укладки графа и попутной проверки, планарен ли он, удобно пользоваться **гамма-алгоритмом**. Для корректной работы алгоритма необходимо несколько ограничить свойства графов, подающихся на вход. Как мы покажем, это не лишит нас возможности с помощью гамма-алгоритма укладывать графы произвольной конфигурации.

Итак, на вход подаются графы, обладающие следующими свойствами:

1. граф связный;
2. граф имеет хотя бы один цикл;
3. граф не имеет мостов, т. е. ребер, после удаления которых, граф распадается на две компоненты связности.

Если нарушено свойство (1), то граф нужно укладывать отдельно по компонентам связности.

Если нарушено свойство (2), то граф — дерево и нарисовать его плоскую укладку тривиально.

Случай нарушения свойства (3) рассмотрим более подробно. Если в графе есть мосты, то их нужно разрезать, провести отдельно плоскую укладку каждой компоненты связности, а затем соединить их мостами. Здесь может возникнуть трудность: в процессе укладки концевые вершины моста могут оказаться внутри плоского графа. Нарисуем одну компоненту связности, и будем присоединять к ней другие последовательно. Каждую новую компоненту связности будем рисовать в той грани, в которой лежит концевая вершина соответствующего моста. Так как граф связности мостами компонент связности является деревом, мы сумеем получить плоскую укладку.

Замечание. Если концевая вершина, принадлежащая новой (только что нарисованной) компоненте связности, также оказалась внутри, необходимо выполнить повторную укладку этой компоненты так, чтобы ребро, содержащее концевую вершину, принадлежало внешней грани. Теорему, гарантирующую то, что для любого ребра можно произвести плоскую укладку так, что оно будет принадлежать внешней грани, можно найти в [3].

Сначала изложим алгоритм на конкретном примере. Пусть дан граф *G* (см. рис. 4).

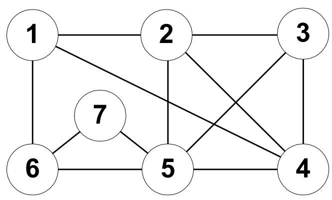

Рис. 4

Первый шаг - инициализация алгоритма: выбираем любой простой цикл в G, пусть это будет {1, 2, 3, 4, 5, 6}, укладываем его на плоскости, и получаем две грани: Γ_1 — внешнюю и Γ_2 — внутреннюю (см. рис. 5).

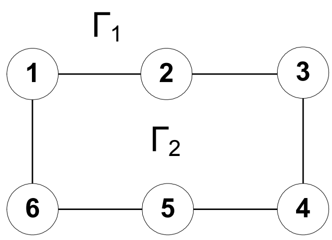

Рис. 5

Уже уложенную часть исходного графа будем обозначать как *G′*, тогда после первого шага *G′* представляет собой цикл {1, 2, 3, 4, 5, 6}.

На каждом шаге будем строить множество **сегментов**. Каждый сегмент *S* относительно уже построенного графа *G′* представляет собой одно из двух:

- ребро, оба конца которого принадлежат *G′*, но само оно не принадлежит *G′*;

- связную компоненту графа $G – G'$, дополненную всеми ребрами графа G, один из концов которых принадлежит связной компоненте, а второй из графа G'.

Вершины, которые одновременно принадлежат G' и какому-то сегменту, назовем **контактными вершинами**. Для нашего примера сегменты изображены на рис. 6. Контактные вершины обведены в квадрат.

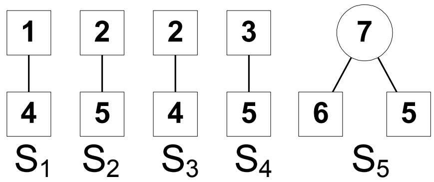

Рис. 6

Если бы в каком-нибудь сегменте не было ни одной контактной вершины, то граф до построения множества сегментов был бы несвязный; если бы была только одна контактная вершина, то граф имел бы мост. Эти возможности заранее исключены, так что каждый сегмент имеет не менее двух контактных вершин. Поэтому в каждом сегменте имеется цепь между любой парой таких вершин.

Если все контактные вершины сегмента S имеют номера вершин какой-то грани Г, то мы будем говорить, что грань Г *вмещает* этот сегмент, в этом случае будем использовать следующее обозначение: S □ Г. Однако, может быть так, что не одна грань вмещает в себя сегмент S, а несколько. Множество таких граней обозначим Г(S), а их число |Г(S)|.

Общий шаг алгоритма следующий: выделяются все сегменты S_i и определяются числа |Г(S_i)|. Если хоть одно из них равно 0, то граф не планарен, конец. Иначе, выбираем сегмент, для которого число |Г(S)| минимально, или любой из них, если таких сегментов несколько. В этом сегменте найдем произвольную цепь между двумя контактными вершинами и уложим ее в любую из граней множества Г(S). При этом данная грань разобьется на две. Уже уложенная часть графа G' после укладки цепи увеличится, а сегмент, из которого вынута цепь, исчезнет или развалится на меньшие с новыми контактными вершинами, ведущими к вершинам G'.

В результате повторения общего шага будет либо получена плоская укладка, когда множество сегментов станет пустым, либо будет получено, что граф G не является планарным.

Вернемся к нашему примеру. Пока для любого i: S_i □ {$Г_1$, $Г_2$}, |Г(S_i)| = 2. Поэтому возьмем первый по номеру сегмент S_i и в нем цепь {1, 4}; вставим эту цепь в грань $Г_2$. После укладки цепи G' увеличится и произойдут изменения в структуре сегментов (см. рис. 7 a, b).

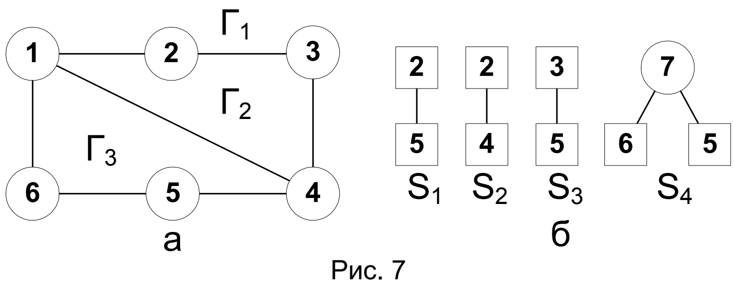

Рис. 7

Определим, какие грани вмещают новые сегменты. Теперь сегменты S_1 и S_3 можно уложить только в одну грань $Г_1$, в то время как сегменты S_2 и S_4 можно уложить в две грани (для S_2 это грани $Г_1$ и $Г_{2,}$ для S_4 - $Г_1$ и $Г_3$). Поэтому берем S_1. Возьмем в нем цепь {2, 5} и уложим ее в $Г_1$. Получим увеличенный граф G' и уменьшенную систему сегментов (см. рис. 8 a, b).

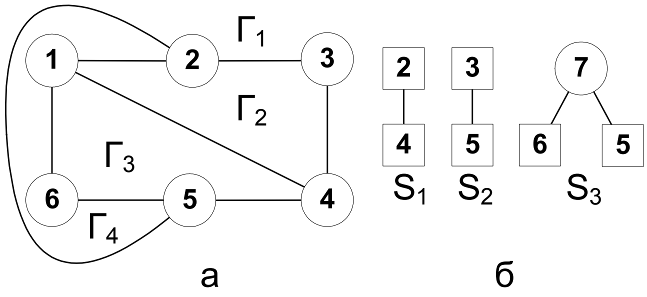

Рис. 8

Продолжая таким образом, в итоге получим плоскую укладку исходного графа G (см. рис. 9).

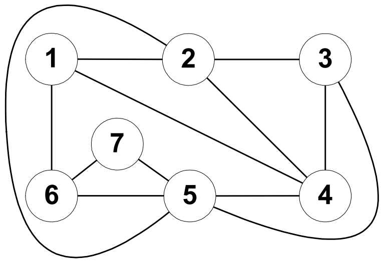

Рис. 9

Еще раз опишем гамма-алгоритм компактно и займемся его обоснованием.

Гамма-алгоритм

1. *Инициализация*. Выберем любой простой цикл C исходного графа G; изобразим его на плоскости в виде грани, которую примем за уже уложенную часть G'; сформируем сегменты S_i; если множество сегментов пусто, то перейти к п. 3. В противном случае перейти к п.2.

2. *Общий шаг*:

   a. Для каждого сегмента S найти множество $\Gamma(S)$. Если существует сегмент S, для которого $|\Gamma(S)| = 0$, то граф не планарный, конец.

   b. Выбираем один из сегментов с минимальным числом, вмещающих его граней.

   c. Выбираем одну из подходящих граней для выбранного сегмента.

   d. В данном сегменте выбираем цепь между двумя контактными вершинами и укладываем ее в выбранной грани. Учтем изменения в структуре сегментов, и если множество образовавшихся сегментов не пусто, перейдем к п. a). В противном случае перейдем к п.3.

3. *Завершение*. Построена плоская укладка G' исходного графа G, конец.

6. Корректность гамма-алгоритма

Вначале приведем ряд вспомогательных утверждений, чтобы с их помощью доказать главную теорему о корректности гамма-алгоритма.

Назовем сегменты S_1 и S_2 **конфликтующими относительно уже уложенной части**, если:

- существует грань, которая вмещает каждый из сегментов;

- в сегментах S_1 и S_2 есть две цепи (между контактными вершинами) L_1 и L_2 соответственно, такие, что их невозможно уложить в одну грань без пересечения.

Лемма 1

Конфликтующие сегменты S_1 и S_2 обладают следующим свойством: если $|\Gamma(S_1)| \geq 2$ и $|\Gamma(S_2)| \geq 2$, то $\Gamma(S_1) = \Gamma(S_2) = 2$.

Доказательство. Действительно, в противном случае, имея по определению одну общую вмещающую грань Γ_3, они бы имели еще по собственной вмещающей грани Γ_1 и Γ_2 соответственно. Тогда любые цепи из S_1 и S_2 могли бы разместиться в Γ_1 и Γ_2 соответственно, а значит и в Γ_3, причем без пересечения; следовательно, S_1 и S_2 не были бы конфликтующими. Противоречие. Что и требовалось доказать.

Замечание. Из доказанной леммы следует, что, если имеется сегмент S_1, и еще сегмент S_2, конфликтующий с S_1, затем сегмент S_3, конфликтующий с S_2 (но не с S_1) и т. д., причем каждый вмещается в две грани, то эти грани являются общими для всех сегментов последовательности, и можно укладывать цепь L_1 из S_1 в первую грань Γ_1, L_2 из S_2 в Γ_2, L_3 из S_3 снова в Γ_1 и т. д. до конца последовательности. Если цепочка сегментов замыкается в цикл четной длины, то проблем не будет; если в нечетный цикл, то в конце окажется, что два конфликтующих сегмента нужно разместить без пересечений в общую грань, что невозможно.

Сейчас нам понадобится в качестве вспомогательного утверждения теорема Д. Кёнига (D. König), которая полезна при решении и многих других задач.

Теорема (Кёнига, 1936)

В графе все циклы четные тогда и только тогда, когда граф является двудольным.

Доказательство.

Достаточность. Рассмотрим двудольный граф. Начнем цикл в верхней доле. Нужно пройти по четному числу ребер, чтобы подняться снова в верхнюю долю. Следовательно, при замыкании цикла число ребер будет четным.

Необходимость. Если граф несвязный, то проведем доказательство отдельно для каждой компоненты. Пусть граф связный и все циклы в нем четные. Выделим произвольную вершину v_0 и найдем произвольные цепи между v_0 и всеми остальными вершинами (например, самые короткие, с помощью алгоритма Дейкстры). Если одна цепь (v_0, v_i) нечетной длины, то и любая цепь (v_0, v_i) нечетная, иначе бы эти цепи образовали нечетный цикл. Аналогично, если (v_0, v_i) — четная, то и любая (v_0, v_i) — четная. Разобьем вершины на две доли: в одну войдет вершина v_0 и все, находящиеся от v_0 на четном расстоянии; в другую долю поместим все вершины, находящиеся от v_0 на нечетном расстоянии. Если вершины u_1 и u_2 принадлежат одной доле, то между ними не может быть ребра, иначе это ребро вместе с цепями (v_0, u_1) и (v_0, u_2) образовали бы нечетный цикл. Ч. т. д.

Двудольным графом называют граф, множество вершин которого можно разбить на две части (доли) таким образом, что концы любого ребра окажутся в разных частях.

Частичной укладкой G' планарного графа G называется граф, который можно получить из какой-нибудь укладки графа G на плоскости удалением каких-то ребер и вершин.

В частичной укладке G' сопоставим каждому сегменту вершину в некотором постороннем служебном графе $A(G')$, где вершины соединяются ребрами, если соответствующие сегменты являются конфликтующими.

Лемма 2

Если результатом некоторого шага работы гамма-алгоритма является частичная укладка G' планарного графа G такая, что $|\Gamma(S)| \geq 2$ для любого сегмента S относительно G', то $A(G')$ — двудольный граф.

Доказательство. Докажем от противного. Пусть $A(G')$ — не двудольный, тогда по теореме Кенига в нем есть r - цикл нечетной длины. Этому циклу соответствует последовательность P сегментов $S_1, S_2,\ldots, S_r, S_1$ относительно G', в которой каждые соседние сегменты конфликтующие. Поэтому на основании Леммы 1 $\Gamma(S_i) = \{\Gamma_1, \Gamma_2\}$ (i = 1..r). Так как G' - частичная укладка графа, то все сегменты $S_1, S_2,\ldots, S_r$ могут быть уложены, а так как соседние сегменты последовательности P конфликтующие, то они должны быть уложены в различные грани (Γ_1, Γ_2). Но это невозможно в силу нечетности числа сегментов r. Противоречие. Что и требовалось доказать.

Наконец, стало возможным доказать главную теорему.

Теорема (о корректности гамма-алгоритма)

Гамма-алгоритм корректен, то есть, если G — планарный граф, то результатом каждого шага гамма-алгоритма является частичная укладка G'.

Доказательство. Докажем индукцией по числу шагов.

База индукции очевидна: результат инициализации есть плоская укладка, так как уложенный цикл будет в любой укладке.

Пусть граф G'_{k-1}, полученный на (k−1)-м шаге, является частичной укладкой. На текущем шаге к нему присоединится цепь L: $G'_k = G'_{k-1} \cup L$. Докажем, что граф G'_k — тоже частичная укладка. Среди сегментов на текущем шаге нет такого S, что $|\Gamma(S)| = 0$, иначе G'_{k-1} не был бы частичной укладкой. Значит, либо существует S такой, что $|\Gamma(S)| = 1$, либо все S таковы, что $|\Gamma(S)| \geq 2$.

Первый случай означает, что укладка S в Γ неизбежна, так что граф G'_k после добавления цепи из S останется частичной укладкой. Во втором случае построим граф $A(G'_{k-1})$, по Лемме 2 он двудольный. Возьмем его связную компоненту A', этот граф тоже двудольный. Для сегментов из A' имеются ровно две грани Γ_1 и Γ_2, вмещающие их. Раскидаем сегменты A' по граням Γ_1 и Γ_2 попеременно. В итоге будет получена частичная укладка.

Фактически, основой последней части доказательства было, что если граф $A(G'_{k-1})$ двудольный, то после удаления части ребер и вершин граф $A(G'_k)$ тоже двудольный. Таким образом, в результате каждой итерации для планарного графа частичная укладка увеличивается, в конце мы получим плоскую укладку. Что и требовалось доказать.

Следствие

Если граф G планарный, то гамма-алгоритм строит его плоскую укладку.

Замечание

В источнике [1] перед доказательством теоремы о корректности приведена лемма, утверждающая, что для любого сегмента S верно $|\Gamma(S)| \le 2$. Это утверждение ошибочно, так как нетрудно найти пример показывающий ситуацию, где $|\Gamma(S)| > 2$. Рассмотрим следующий пример (см. рис. 10).

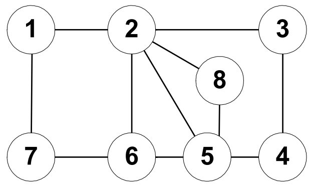

Рис. 10

На шаге инициализации выберем простой цикл {1, 2, 3, 4, 5, 6, 7} , это будет уже уложенная часть графа, G'. Затем, к получившемуся графу G' добавим сегмент 2 – 6. На следующем шаге мы уложим цепь 2 – 5 в одну из граней (2-3-4-5-6): мы видим, что теперь оставшуюся цепь 2 - 8 - 5 мы можем уложить в грани Γ_2, Γ_3 и Γ_4 (см. рис. 11).

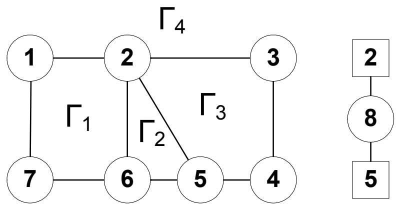

Рис. 11

Литература

1. Бондарев В.М., Рублинецкий В.И., Качко Е.Г. Основы программирования. — Харьков: Фолио; Ростов н/Д: Феникс, 1997.

2. Емеличев Р.И., Мельников О.И., Сарванов В.И., Тышкевич Р.И. Лекции по теории графов. — М.: Наука, 1990.

3. Харари Ф. Теория графов. – М.: Мир, 1973.