УДК 519.172
DOI https://doi.org/10.32782/tnv-tech.2024.1.10

МОДИФІКАЦІЯ МЕТОДУ НИТЕЙ С ВИКОРИСТАННЯМ ГАМА АЛГОРИТМУ ДЛЯ ПОБУДОВИ ТОПОЛОГИЧНОГО МАЛЮНКУ ГРАФУ

***Сгадов С. О.** – старший викладач кафедри комп’ютерних систем та мереж Національного університету «Запорізька політехніка»*
ORCID ID: 0000-0002-7994-6530

В цій роботі розглядається модифікований алгоритм методу ниток, призначений перевірки графа на планарність з одночасним побудовою топологічного малюнка плоского графа як системи ізометричних циклів і списків обертання вершин графа. Отримання обертання вершин графа одразу вирішує дві найважливіші задачі: завдання перевірки графа на планарність та завдання побудови топологічного малюнка плоского графа. Отримана в результаті роботи алгоритму система ізометричних циклів графа індукує обертання вершин опису топологічного малюнка плоского графа з метою подальшої візуалізації. Топологічний малюнок плоскої частини графа дозволяє описувати процес планаризації методами алгебри, не виробляючи ніяких геометричних побудов на площині. Представлений алгоритм заснований на розбудові опорного циклу та побудові списку зворотних маршрутів. Основою розрахунку є побудова матриці фундаментальних циклів на підставі побудови остовного дерева графа методом пошуку в глибину. З матриці фундаментальних циклів проводися виділення ланцюгів, які утворюють маршрути. Для отримання системи ізометричних циклів використовують вбудовування маршрутів за допомогою модифікованого гамма алгоритму. Адаптована версія гамма алгоритму у додатку методу ниток дозволяє вбудовувати в цикли графа утворюють маршрути ланцюга та спрощує завдання пошуку відповідних для цього циклів. Адаптація гамма алгоритму включає додатковий прогностичний індекс визначення можливості вбудовування в даний цикл наступних шляхів. Планаризация графів є найважливішим підзавданням під час вирішення безлічі актуальних прикладних завдань, як-то проектування складних виробів і систем, плоских конструктивів, аналіз соціальних мереж та інших.

***Ключові слова:** граф, перевірка планарности, візуалізація графів, топологічний малюнок графа, діаграма обертання вершин, ізометричні цикли, метод нитей, обертання вершин графа, алгоритм.*

Sgadov S. A. Modification of the thread method with using the gamma algorithmfor graph topological drawing building

In this paper, one consider a modified algorithm of the thread method, which is designed to check a graph for planarity with the simultaneous building a topological drawing of a planar graph in the form of a system of isometric cycles and lists of rotation of graph vertices. Obtaining the rotation of the graph vertices immediately solves two important problems: the problem of checking the graph for planarity and the problem of building a topological drawing of a planar graph. The system of graph isometric cycles obtained as a result of the algorithm induces the rotation of vertices to describe the topological pattern of a planar graph for the purposes of subsequent visualization. The topological drawing of the flat part of the graph makes it possible to describe the process of planarization by algebraic methods without making any geometric constructions on the plane. The presented algorithm is based on the rearrangement of the reference cycle and the construction of a list of reverse routes. The basis of the calculation is the construction of a matrix of fundamental cycles based on the construction of the spanning tree of the graph by the depth-first search method. Based on the matrix of fundamental cycles, the chains that form the routes are selected. To obtain a system of isometric cycles, routes are embedded using a modified gamma algorithm. An adapted version of the gamma algorithm in the application of the thread method makes it possible to embed path-forming chains into graph cycles and simplifies the task of finding suitable cycles for this. Adaptation of the gamma algorithm includes an additional predictive index to determine the possibility of embedding subsequent paths in a given cycle. Graph planarization is the most important subtask in solving many relevant applied problems, such as designing complex products and systems, flat constructs, analyzing social networks, etc.

***Key words:** graph, planarity check, graph visualization, graph topological drawing, vertex rotation diagram, isometric cycles, thread method, graph vertex rotation, algorithm.*

Вступ. Побудова плоских графів і візуалізація їх зображень [1–7] – важлива подзадача, що виникає при розв'язку прикладних задач у багатьох областях. Прикладами можуть служити різні задачі планаризации в сучасному виробництві [3; 4] і побудова автоматизованих систем для технічні обладнань, у яких неперечні зв'язки між елементами обладнання й самі елементи розташовані на паралельних площинах. До них ставляться друковані плати, інтегральні схеми, ВІС і СВІС. В [8] представлений метод контролю планарности графів шляхом одночасної побудови математичних структур, що описують топологічні паттерны графів для цілей візуалізації. Метод заснований на побудові ізометричних циклів, обертанні вершин графа і визначенні операції перетинання ребер як перетинань проекцій на координатнобазисную систему, у якості якої виступає опорний цикл DFS-дерева графа [8]. Варто відзначити, що в цей час існують ефективні алгоритми для визначення того, чи є граф планарным чи ні, і їхня складність характеризується лінійною залежністю від кількості вершин у графі. Однак метод в [8] має високу обчислювальну складність $0(m^2)$, де m – кількість вершин у графі) на відміну від алгоритмів з лінійною складністю. Цей метод не тільки визначає, чи є граф планарным чи ні, але й дає топологічну діаграму, яка може бути використана для візуалізації графа, однак висока обчислювальна складність уважається істотним недоліком. У роботі був досліджений модифікований метод ниток для визначення планарности й побудови списку обертання вершин з обчислювальною складністю, порівнянної з алгоритмом Хопкрофта-Тарьяна [7]. Він заснований на способі переформування DFS-дерева й зворотніх шляхів, а також на рекурсивному способі побудови обручів. У даній роботі запропонована модифікація методу ниток за допомогою адаптованого для цілей вбудовування ланцюгів гама-алгоритму [9].

Гамма-алгоритм. Дамо короткий опис цього відомого алгоритму для планаризации графа [9]. На вхід подаються граф, що володіє наступними властивостями:

граф зв'язний;

граф має хоча б один цикл;

граф не має містків, тобто ребер після видалення яких, граф розпадається на дві компоненти связности. Далі виконуються кроки:

1. Ініціалізація. Вибирається простий цикл у вихідному графі й зображується на площині.

2. Загальний крок. Цей крок повторюється доти, поки граф не буде укладений або поки не буде отримано, що граф не планарен:

будується множина сегментів;

для кожного сегмента обчислюється величина г(s), рівна кількості граней (циклів), у яких лежать усі вершини сегмента s. Якщо існує і: г(s) = 0, то граф не планарен, алгоритм завершує роботу;

обирается сегмент із мінімальним числом r(s) > 0;

у цьому сегменті вибирається ланцюг між двома контактними вершинами;

цей ланцюг укладається в будь-яку грань, що вміщає даний сегмент.

Або отримане плоске укладання графа, або граф виявився не планарен.

Недоліком гама-алгоритму в такому виді служить, мабуть, той факт, що у відмінності від методу ниток, сегментом може бути не тільки ланцюг, але й значно більший компонент связности, який також окремо випливає планаризовать. Крім того, залишається проблема вибору циклу на кроці (1).

Постановка задачі. Нехай заданий довільний граф G. Послідовно переглядаючи всі вершини графа, вилучимо петлі, «висячі» вершини й кратні ребра (якщо такі будуть знайдені). Потім вилучимо мости й крапки зчленування, тим самим одержавши трохи компонент связности, які можна розглядати окремо. Два ребра, з’єднаних однієї вершиною, що має локальний ступінь рівну двом, можна замінити одним ребром. Це перетворення потрібно запам’ятати для наступного відновлення первісного виду графа G. Із цією метою дамо поняття несепарабельного графа.

Визначення 1. Несепарабельним графом G будемо називати зв’язний неорієнтований граф без петель і кратних ребер, без мостів і крапок зчленування, без вершин з локальним ступенем меншої або рівної двом.

Надалі будемо розглядати несепарабельные неорієнтовані графи. До таких несепарабельным графам для визначення планарности можна застосувати критерій планарності Маклейна і використовувати операцію кільцевого підсумовування суграфов у підпросторі циклів.

Визначення 2. Ізометричним циклом у графі називається простий цикл, для якого найкоротший шлях між будь-якими двома його вершинами складається з ребер цього циклу.

Визначення 3. Топологічним малюнком планарного графа будемо називати однаково спрямоване обертання всіх його вершин, индуцирующее множина ізометричних циклів графа, яке задовольняє критерію планарності Маклейна.

Таким чином, задача полягає в тому щоб одержати алгоритм, який перевірить заданий несепарабельный граф G і побудує його топологічний малюнок, спочатку у вигляді системи ізометричних циклів, а потім по них можна буде одержати список обертань вершин.

Опис алгоритму. За основу поберемо алгоритм методу ниток даного в [14] і модифікуємо його таким чином, як показано нижче.

1. Побудова DFS-дерева графа й формування матриці фундаментальних циклів (МФЦ).

2. Вибираємо із МФЦ цикл максимальної довжини – опорного циклу й прозводим його подовження.

3. Побудова зворотних маршрутів. Виписуємо із МФЦ ребра циклів, які не входять в опорний цикл і відповідні хорди. Залишаємо самі довгі без змін, обробляємо список так, щоб не було загальних ребер.

4. Вбудовування зворотних маршрутів. Користуючись підходом гама-алгоритму послідовно шукаємо підходящі маршрути, одночасно визначаємо цикли, на яких лежать кінці маршрутів і вбудовуємо їх у ці цикли.

5. Якщо вбудовування зворотних маршрутів відбулося успішно – робиться висновок, що граф планарен і тоді будується список обертань по отриманій системі ізометричних циклів.

Розглянемо ці кроки алгоритму більш докладно.

Формування матриці фундаментальних циклів. При побудова остовного дерева графа методом пошуку в глибину можна одержати множину циклів і множина відповідних хорд, які сформують матрицю фундаментальних циклів. Матриця фундаментальних циклів складається із двох подматриц: одиничної подматрицы й подматрицы π, що полягає тільки з гілок дерева. Таким чином, рядок подматрицы π разом з відповідною хордою буде утворювати цикл графа. Представимо рядки подматрицы π у вигляді списку ребер у тому порядку, у якому вони входять у відповідний цикл остовного дерева. Вибираємо з фундаментальної матриці циклів самий довгий цикл, утворений гілками дерева й однієї хордою – назвемо його опорним циклом. Розбиваємо подматрицю на дві частини: частина π', яка складається з гілок дерева приналежних опорному циклу, і частина π'', яка складається з гілок дерева не приналежних опорному циклу.

Формування матриці фундаментальних циклів. При побудова остовного дерева графа методом пошуку в глибину можна одержати множину циклів і множина відповідних хорд, які сформують матрицю фундаментальних циклів. Матриця фундаментальних циклів складається із двох подматриц: одиничної подматрицы й подматрицы π, що полягає тільки з гілок дерева. Таким чином, рядок подматрицы π разом з відповідною хордою буде утворювати цикл графа. Представимо рядки подматрицы π у вигляді списку ребер у тому порядку, у якому вони входять у відповідний цикл остовного дерева. Вибираємо з фундаментальної матриці циклів самий довгий цикл, утворений гілками дерева й однієї хордою – назвемо його опорним циклом. Розбиваємо подматрицу на дві частини: частина, яка складається з гілок дерева приналежних опорному циклу, і частина π'', яка складається з гілок дерева не приналежних опорному циклу.

Подовження опорного циклу. Зробимо дії, які спрямовані на те, щоб опорний цикл був максимальної довжини. Для цього використовуємо жадібний алгоритм, який на кожному кроці буде шукати такий рядок і у матриці π, щоб різниця Δ_i між кількістю ребер, які не входять в опорний цикл була строго більше, ніж кількість ребер, які належать опорному циклу. Далі змінимо матрицю фундаментальних циклів у такий спосіб: замінимо відповідну до рядка і хорду на останній елемент із і-й рядка подматрицы π'. Потім знайдемо такі цикли, що мають загальні ребра й замінимо їх циклічною сумою с і-м циклом. Таким чином, відбудеться подовження, у тому числі й опорного циклу, за рахунок ребер i-го. Наприкінці зробимо проектування подматрицы π на частині π' і π'' у зв'язку зміною опорного циклу.

Коли не можна буде знайти $\Delta_i > 0$ – процедура подовження вважається закінченою.

Побудова зворотних маршрутів. Назвемо *зворотним шляхом* орієнтований маршрут, що полягає з послідовно розташованих гілок дерева й однієї, і тільки однієї, хорди [8]. Причому, кінцеві вершини такого дороги назад не завжди належать опорному циклу. Таким чином, побудуємо список **S**, який складається з елементів рядків подматрицы π'' і соответвующих рядкам хорд. Ранжируем список **S**, по убуванню довжини маршруту. Тепер треба добитися, щоб у списку не було маршрутів, що проходять по тим самим ребрах. Для цього будемо по черзі забирати з **S** маршрути, поміщати в список доріг назад **W** і переглядати **S** у пошуках маршрутів із загальними ребрами, а саме:

1. Нехай $W = \varnothing$.
2. Для кожного i: поберемо $\boldsymbol{S_i}$.
3. Помістимо S_i у кінець списку **W**.
4. Переглянемо список **S**, знайдемо в ньому маршрути, що мають загальні ребра S_i , вилучимо з них загальні ребра й помістимо в **W.**
5. Знайдені на кроці 4 маршрути й S_i вилучимо зі списку **S.**

Таким чином, буде сформований список зворотних шляхів **W**, які ми зможемо використовувати на наступному етапі. При цьому слід звернути увагу на те, що порядок виписування маршрутів впливає на кількість переглядів списку **W** надалі. Тому бажане його попередньо отранжировать по зменшенню довжин шляхів.

Вбудовування зворотних маршрутів. У роботі [14] було введена операція включення ребра (маршруту) у цикл у такий спосіб. Наприклад, якщо в довільний цикл

$$c_i = \langle x_{18}, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{17}, x_1, x_{18} \rangle.$$

потрібно включити ребро $\{x_{10,}\ x_{17}\}$, те для виконання цієї операції необхідно щоб кінцеві вершини знову введеного ребра (або сукупності ребер) приналежали циклу. Тоді вміст циклу розбивається на частині, де кінцеві вершини ребра беруть участь у формуванні маршрутів. Потім з дотриманням послідовності до виділених частин приєднуються дуги. Таким чином, утворюються два нові обручі:

$$<x_{18}, x_9, x_{10}, x_{17}, x_1, x_{18}>;$$
$$<x_{10}, x_{11}, x_{12}, x_{13}, x_{17}, x_{10}>.$$

Використовуючи поняття операції включення маршруту в цикл, побудуємо алгоритм вбудовування зворотних шляхів зі списку **W** у цикли графа, модифікуючи аналогічний [3; 8]:

1. Нехай список **C** – список изметрических циклів.

Тому що опорний цикл розбиває поверхня R^2 на внутрішню й зовнішню області, то для ініціалізації додамо в список **C** опорний цикл як у прямому так і зворотному напрямку.

2. Переглянемо список **W** і для кожного *i* обчислимо $r(w_i)$ на множині **C** і знайдемо i_{min} таке, що відповідає $\min\Gamma(W_i)$.

3. Якщо $\min\Gamma(W_i) = 1$, то знайдений шлях Wi_{min} і існує елемент списку циклів, такий що включає початкову й кінцеву вершини маршруту $W i_{\min}$. Тоді c_j робимо включення шляха $W i_{\min}$ у цикл c_j, що породжує цикли c_j' і c_j'' . Заміняємо цикл c_j двома цими новими циклами c_j' і c_j''.

4. Якщо $\min\Gamma(W_i) \geq 2$, те це означає, що маршрут претендує більш ніж на одну грань і існує множина циклів $\Theta \in C$ таке, що може бути вбудований у будь-який цикл із цієї множини. Тоді введемо додатковий критерій

$$\theta_{ij} = \theta$$

для маршруту w_i – кількість елементів списку **W**, кінці яких лежать і на w_i і на даному циклі

$$\theta_j \in \boldsymbol{\theta}.$$

Якщо $\theta_{ij} = 0$, те зрозуміло, що більше маршрутів у j- і цикл вписати неможливо. Тому обчислимо θ_{ij} для всіх елементів $\theta_j \in \boldsymbol{\theta}$ і знайдемо j таке, щоб θ_{ij} було максимальне. Тоді скориставшись цим критерієм для $W i_{\min}$ знайдемо той цикл у який можна вмонтувати маршрут, не побоюючись за те, що на наступних ітераціях алгоритму виникне ситуація, коли не буде можливості вмонтувати маршрути, які лежать на даному $W i_{\min}$.

5. Послідовно вбудовуємо шляхи зі списку **W** до повного вичерпання. Якщо виявиться, що ніяк не можна добитися вичерпання списку **W**, тобто якщо на якімсь кроці зустрівся сегмент S, для якого немає грані, що вміщає, то граф непланарный

Якщо граф G – планарный, то алгоритм будує його плоске укладання у вигляді системи ізометричних циклів **C**.

6. По системі ізометричних циклів **C** побудуємо списки обертань для даного графа.

Особливі випадки. При подовженні опорного циклу можливі випадки, коли не можливо уникнути появу замкнених маршрутів, у тому числі, що й не мають загальних вершин з опорним циклом. Їхня присутність можна виявити виходячи з аналізу рядків подматрицы π' – а саме присутвие там порожніх рядків. Це означає, що необхідно зробити застосування даного методу вроздріб – а саме – вилученням таких рядків з матриці фундаментальних циклів ми утворюємо граф, який можна досліджувати методом ниток. Вилучені рядки дозволять побудувати трохи компонент связности, на яких можна рекурсивно використовувати даний алгоритм. Якщо даний алгоритм вдало укладає на площині ці компоненти связности окремо, то отже, вихідний граф планарен.

Висновки. У роботі представлений алгоритм, отриманий модифікацією методу ниток [8], який використовує вдосконалений спосіб вбудовування маршрутів із застосуванням додаткових критеріїв, що дозволяють уникати невірного вибору циклів на кінцевому етапі алгоритму. Отримана також підхід до модифікації гамма-алгоритму для вбудовування ланцюгів. Так пропонується оцінювати використовуючи не тільки гама критерій для шляхів, але тета-критерій для дозволу ситуацій, коли маршрут претендує на кілька циклів. Отримана система ізометричних циклів графа индуцирует обертання вершин для опису топологічного малюнка плоского графа. Топологічний малюнок плоскої частини графа дозволяє описувати процес планаризации алгебраїчними методами, не роблячи ніяких геометричних побудов на площині. Одержання обертання вершин графа відразу вирішує дві найважливіші задачі теорії графів: задачу перевірки графа на планарность і задачу побудови топологічного малюнка плоского графа.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ:

1. Goyal P., Ferrara E. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems.* 2018. Vol. 151. P. 78–94.
2. Tamassia R. Handbook of Graph Drawing and Visualization. *C&H/CRC.* 2013. P. 844.
3. Kurapov S. V., Davidovsky M. V., Tolok A. V. Generating a topological drawing of the flat part of a non-planar graph. *Scientific Visualization.* 2020. Vol. 12, No. 1, P. 90–102. DOI: 10.26583/sv.12.1.08
4. Kavitha T., Liebchen C., Mehlhorn K., Michail D., et al. Cycle bases in graphs – characterization, algorithms, complexity, and applications. *Comput. Sci. Rev..* 2009. No. 3. P. 199–243.
5. Ringel G. Map Color Theorem. Repr. of the orig. 1st ed. (1974). *Springer.* 2011. P. 212.
6. Patrignani M. Planarity testing and embedding. Chapter 1. Handbook of Graph Drawing and Visualization. *Roberto Tamassia, Editor. CRC Press.* 2013. June 24, P. 1–42.
7. Hopcroft J., Tarjan R. E. Efficient planarity testing. *Journal of the Association for Computing Machinery.* 1974. T. 21, No. 4, P. 549–568.
8. Kurapov S. V., Davidovsky M. V., Tolok A. V. A modified algorithm for planarity testing and constructing the topological drawing of a graph. The thread method. *Scientific Visualization.* 2018. T. 10, No. 4, P. 53–74.
9. Hrebeniuk B. Modification of the analytical gamma-algorithm for the flat layout of the graph. *Computer Science & Software Engineering.* 2018. Vol. 2292, P. 46–54.

REFERENSES:

1. Goyal P., & Ferrara E. (2018) Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems,* (Vols. 151), (pp. 78–94).
2. Tamassia R. (2013) Handbook of Graph Drawing and Visualization. *C&H/CRC*, (P. 844).
3. Kurapov S. V., & Davidovsky M. V., & Tolok A. V. (2020) Generating a topological drawing of the flat part of a non-planar graph. *Scientific Visualization*, (Vols. 12), (no. 1), (pp. 90–102). DOI: 10.26583/sv.12.1.08
4. Kavitha T., & Liebchen C., & Mehlhorn K., & Michail D., et al. (2009) Cycle bases in graphs – characterization, algorithms, complexity, and applications. *Comput. Sci. Rev.*, 3, 199–243.

5. Ringel G. (2011) Map Color Theorem. *Repr. of the orig*. 1st ed. (1974), *Springer*, P. 212.

6. Patrignani M. (2013) Planarity testing and embedding. Chapter 1. Handbook of Graph Drawing and Visualization. *Roberto Tamassia, Editor. CRC Press,* June 24, (pp. 1–42).

7. Hopcroft J. Tarjan R. E. (1974) Efficient planarity testing. *Journal of the Association for Computing Machinery*. (Vols. 21), (no. 4), (pp. 549–568).

8. Kurapov S. V., Davidovsky M. V., & Tolok A. V. (2018). A modified algorithm for planarity testing and constructing the topological drawing of a graph. The thread method. *Scientific Visualization*, (no. 10(4)), (pp. 53-74).

9. Hrebeniuk B. (2018) Modification of the analytical gamma-algorithm for the flat layout of the graph. *Computer Science & Software Engineering,* (Vols. 2292), (pp. 46–54).