

М. А. МІРОШНИК, д-р техн. наук, проф., Харківський національний університет імені В.Н. Каразіна, Харків;

Я. Ю. КОРОЛЬОВА, канд. техн. наук, Національний технічний університет "Харківський політехнічний інститут", Харків;

А. М. МІРОШНИК, PhD, ст. викладач Харківський національний інститут радіоелектроніки, Харків;

А. В. ШАФРАНСЬКИЙ, асп. Харківський національний університет імені В.Н. Каразіна, Харків

ГРАФОВА МОДЕЛЬ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ПОСЛІДОВНОГО АЛГОРИТМУ КОМПОНУВАННЯ ДЛЯ МУЛЬТИМЕДІЙНИХ ПРИСТРОЇВ

У роботі розглянуто задачу компоювання елементів мультимедійних пристроїв на основі графової математичної моделі. Запропоновано та проаналізовано послідовний алгоритм компоювання, що дозволяє здійснювати покрокове формування підграфів із урахуванням конструктивних обмежень та мінімізації кількості зовнішніх зв'язків між підграфами. Кожному елементу пристрою відповідає вершина графа, а зв'язки між ними відображаються у вигляді ребер. Для демонстрації роботи алгоритму розроблено програмну реалізацію на мові Python, яка дозволяє в автоматизованому режимі виконувати етапи вибору вершин та формування підграфів відповідно до заданих критеріїв. Наведено приклад роботи програми, що підтверджує ефективність запропонованого підходу для оптимізації структури мультимедійних пристроїв. Іл.: 1. Бібліогр.: 10 назв.

Ключові слова: модель, граф, вершина, ребро, зв'язки, компоювання, алгоритм, оптимізація, мультимедійний пристрій, мінімізація, програмна реалізація Python, автоматизований режим, ефективність.

Вступ. Сучасні тенденції розвитку радіоелектроніки супроводжуються постійним зростанням функціональної складності та конструктивної різноманітності елементної бази. У зв'язку з цим виникає необхідність формалізації процесів проектування та оптимізації складу конструктивних елементів для кожного конкретного класу схем. Одним із пріоритетних завдань на початкових етапах проектування є визначення оптимального набору елементів, що дозволяє зменшити кількість типів компонентів у

складі розроблюваного пристрою та, відповідно, підвищити економічність і технологічність його виробництва.

Ефективне вирішення задач компоновки елементів неможливе без переходу від конкретних функціональних і електричних схем до їх абстрактного математичного представлення. Використання графових моделей, зокрема мультиграфів, дозволяє формалізувати структуру пристрою, що суттєво спрощує процес аналізу й оптимізації. У запропонованому підході кожному конструктивному елементу відповідає вершина мультиграфа, а електричні зв'язки між елементами моделюються ребрами графа.

Завдання компоновки передбачає раціональне розміщення елементів схеми на комутаційних платах або інших конструктивних носіях із урахуванням мінімізації кількості міжмодульних зв'язків. Це, у свою чергу, дозволяє підвищити загальну надійність електронного пристрою, зменшити затримки сигналів у електричних ланцюгах, оптимізувати конструкцію та забезпечити її технологічність.

У межах даної роботи особлива увага зосереджена на детальному аналізі та практичному застосуванні **послідовного алгоритму компоновки**, що є одним із базових методів формування конструктивної структури мультимедійних і радіоелектронних пристроїв. Зазначений підхід ґрунтується на поетапному, структурованому розподілі елементів схеми між окремими підграфами відповідно до заданих конструктивних і технологічних обмежень.

До основних переваг послідовного алгоритму слід віднести його **високу швидкість**, а також здатність ефективно працювати у випадках, коли проектоване технічне рішення містить **жорсткі обмеження** щодо взаємного розташування функціональних елементів або мінімізації кількості міжмодульних зв'язків. Особливо це актуально на етапах розробки схем з підвищеними вимогами до компактності, технологічності та електричних характеристик, де навіть незначне збільшення кількості зовнішніх з'єднань може призвести до зростання паразитних параметрів та зниження надійності пристрою.

У рамках дослідження показано, що представлення електричної схеми у вигляді **мультиграфа**, де вершини відповідають окремим елементам конструкції, а ребра – електричним з'єднанням між ними, дозволяє формалізувати процес компоновання та застосувати

математично обґрунтовані методи оптимізації. Послідовний алгоритм, зокрема, передбачає вибір вершин із мінімальною локальною мірою та послідовне включення їх до складу підграфів таким чином, щоб мінімізувати кількість зовнішніх зв'язків між сформованими підграфами.

На відміну від ітераційних методів, які потребують значних обчислювальних ресурсів та часу на багаторазову оптимізацію структури, послідовний підхід демонструє високу ефективність саме у задачах з великою кількістю жорстких обмежень та при обробці схем середньої складності, що робить його практичним і доцільним для застосування в умовах інженерного проектування.

Для ілюстрації практичного використання послідовного алгоритму в роботі також представлено програмну реалізацію на мові Python, що дозволяє в автоматизованому режимі здійснювати всі ключові етапи процесу компоновання – від побудови таблиці суміжності до остаточного формування підграфів з урахуванням усіх обмежень.

Комплексне застосування графової моделі та послідовного алгоритму компоновання є дієвим засобом підвищення ефективності, оптимізації конструктивної структури та якості проектних рішень у сфері розробки мультимедійних та радіоелектронних пристроїв [1 – 5].

Послідовний алгоритм компоновки

Послідовний алгоритм компоновки ґрунтується на ітеративному розміщенні підграфів із урахуванням обмежень на кількість зовнішніх зв'язків між ними. Основною метою є таке розбиття заданого мультиграфа, що моделює електричну схему, на підграфи, за якого мінімізується кількість міжпідграфових з'єднань, що, у свою чергу, позитивно впливає на надійність та технологічність конструкції [6 – 10].

Формулювання задачі:

Нехай ненаправлений мультиграф задано виразом (1)

$$G(V,E) \tag{1}$$

де: V – множина вершин, що відповідають конструктивним елементам пристрою; E – множина ребер, що моделюють електричні зв'язки між елементами.

Завдання полягає у розбитті графа G на підграфи G_1, G_2, \dots, G_k , так, щоб кількість зовнішніх зв'язків між підграфами була мінімальною, тобто виконувалась умова (2):

$$\min \sum_{i=1, k} \sum_{j=i+1, k} |E_{ij}| \quad (2)$$

де: E_{ij} – множина ребер, що з'єднують підграфи G_i та G_j

До основних конструктивних обмежень належать:

- фіксована кількість підграфів k ;
- задана кількість вершин у кожному підграфі;
- максимальна допустима кількість зовнішніх зв'язків для кожного підграфа;
- вимоги на роздільне розташування окремих вершин у різних підграфах.

Етапи реалізації алгоритму:

1. Представлення електричної принципової схеми у вигляді ненаправленого мультиграфа $G(V, E)$.

2. Побудова матриці суміжності для графа та обчислення локального ступеня вершин.

3. Визначення вершини з мінімальним локальним ступенем. Якщо таких вершин кілька, обирається вершина з найбільшою кількістю кратних ребер. За подальшої рівності вибирається вершина з меншим порядковим номером.

4. Формування початкового підграфа G_1 , включення до нього обраної вершини.

5. З множини вершин, суміжних із включеними у підграф, вибирається вершина, що має мінімальний відносний приріст зовнішніх зв'язків при включенні її до підграфа (3), тобто:

$$P(v) = \Delta E_{\text{ext}}(v) = \sum_{u \in s} a_{uv} \quad (3)$$

де: s – поточне множина вершин підграфа; a_{uv} – елемент матриці суміжності, що відображає наявність зв'язку між вершинами u та v

Якщо таких вершин кілька, перевага надається вершині з найбільшим локальним ступенем, за подальшої рівності – вершині з меншим порядковим номером.

6. Включення обраної вершини до підграфа за умови, що кількість зовнішніх зв'язків не перевищує заданого обмеження (4):

$$|E_{\text{ext}}(G_i)| \leq E_{\text{max}} \quad (4)$$

7. Процес повторюється до досягнення максимально допустимої кількості вершин у підграфі або до моменту порушення обмеження на зовнішні зв'язки. У випадку неможливості подальшого включення вершин формується підграф із максимально можливим числом елементів за умови дотримання обмежень.

8. Після завершення формування підграфа процес повторюється для наступного підграфа, при цьому аналізуються лише вершини, що залишилися поза вже сформованими підграфами.

9. Процедура продовжується до повного розбиття графа на необхідну кількість підграфів.

10. Завершення алгоритму.

В якості переваг послідовного алгоритму є:

- висока швидкість виконання;
- придатність до використання у випадках із жорсткими конструктивними обмеженнями;
- простота реалізації.

До недоліків використання послідовного алгоритму відноситься:

- не забезпечую повного оптимального рішення для всього графа;
- чутливість до вибору початкових вершин.

Виходячи з цього застосування послідовного алгоритму є доцільним на ранніх етапах компоновки або в задачах, де критичними є часові та обчислювальні обмеження.

Програмна реалізація послідовного алгоритму компоновання.

У межах дослідження було здійснено програмну реалізацію послідовного алгоритму компоновання, що базується на представленні електричної схеми у вигляді ненаправленого мультиграфа. Застосування даного підходу забезпечує формалізацію структури пристрою, дозволяє здійснювати математичний аналіз та оптимізувати конструктивні рішення шляхом покрокового розподілу елементів схеми між підграфами з урахуванням заданих обмежень.

Для демонстрації роботи алгоритму використано матрицю суміжності, яка описує структуру зв'язків між елементами пристрою. Основними вхідними параметрами є:

- матриця суміжності графа;
- максимально допустима кількість зовнішніх зв'язків для підграфа;
- максимальна кількість вершин у підграфі.

Алгоритм працює за наступною схемою:

Визначення локальних степенів вершин, що відповідають кількості безпосередніх з'єднань кожного елемента схеми.

Вибір початкової вершини для формування підграфа здійснюється серед тих вершин, які ще не були включені до жодного з підграфів. Перевага надається вершині з мінімальним локальним ступенем, що дозволяє знизити кількість зовнішніх зв'язків.

Покрокове розширення підграфа шляхом додавання суміжних вершин, для яких приріст зовнішніх зв'язків є мінімальним, і які не порушують задані конструктивні обмеження.

Процес повторюється до моменту включення усіх вершин графа до складу підграфів.

Програмна реалізація алгоритму мовою Python.

Блок-схема послідовного алгоритму компоновки наведено на рис. 1.

Опис блок-схеми та програмної реалізації послідовного алгоритму компонування наведено нижче.

На першому етапі реалізації алгоритму ми представляємо електричну принципову схему у вигляді ненаправленого графа.

Далі будуємо матрицю суміжності для графа та проводимо обчислення локального ступеня вершин.

На наступному кроці алгоритму визначаємо вершини з мінімальним локальним ступенем та аналізуємо. Якщо таких вершин кілька, обирається вершина з найбільшою кількістю кратних ребер.

Далі, за подальшої рівності вибирається вершина з меншим порядковим номером.

На наступному кроці формуємо початковий підграф включенням до нього обраної вершини.

Після цього, з множини вершин, суміжних із включеними у підграф, вибирається вершина, що має мінімальний відносний приріст зовнішніх зв'язків при включенні її до підграфа.

Якщо таких вершин кілька, перевага надається вершині з найбільшим локальним ступенем, за подальшої рівності – вершині з меншим порядковим номером.

Включення обраної вершини до підграфа за умови, що кількість зовнішніх зв'язків не перевищує максимуму.

Даний процес повторюється до досягнення максимально допустимої кількості вершин у підграфі або до моменту порушення обмеження на зовнішні зв'язки.

У випадку неможливості подальшого включення вершин формується підграф із максимально можливим числом елементів за умови дотримання обмежень.

Після завершення формування підграфа процес повторюється для наступного підграфа, при цьому аналізуються лише вершини, що залишилися поза вже сформованими підграфами.

Процедура продовжується до повного розбиття графа на необхідну кількість підграфів. На цьому алгоритм завершується.

Програмна реалізація алгоритму здійснена мовою Python із використанням бібліотеки NumPy для зручної роботи з матрицями та векторними операціями.

Програмна реалізація алгоритму здійснена мовою Python із використанням бібліотеки NumPy для зручної роботи з матрицями та векторними операціями.

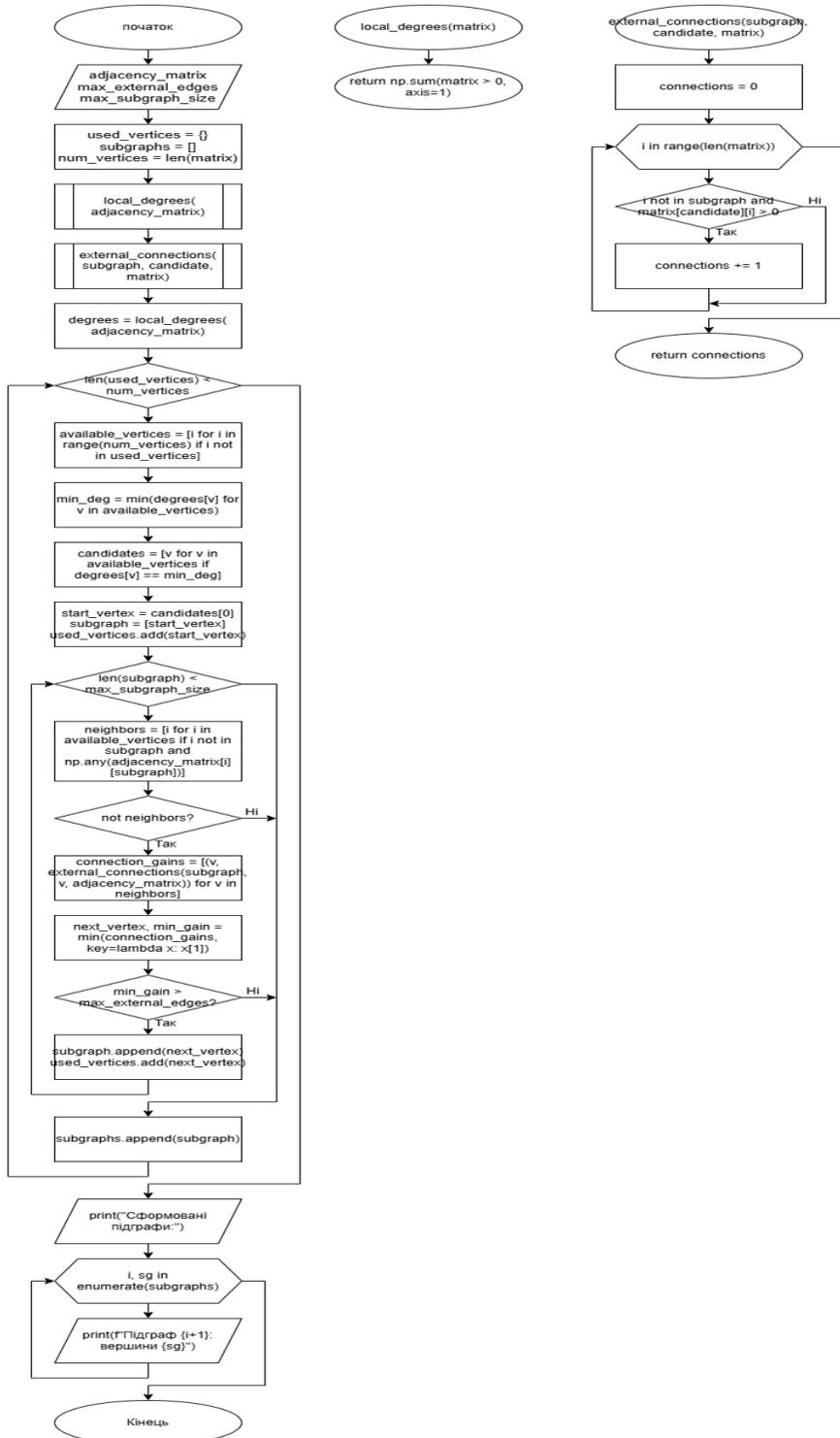


Рис. 1. Блок-схема послідовного алгоритму компоновки *import numpy as np*.

Наведений нижче фрагмент коду ілюструє основні етапи реалізації:

```
# Початкова матриця суміжності
adjacency_matrix = np.array([
    [0, 2, 0, 1, 0, 0],
    [2, 0, 1, 0, 0, 0],
    [0, 1, 0, 2, 0, 0],
    [1, 0, 2, 0, 1, 1],
    [0, 0, 0, 1, 0, 2],
    [0, 0, 0, 1, 2, 0]
])

num_vertices = adjacency_matrix.shape[0]
used_vertices = set()
subgraphs = []
max_external_edges = 2
max_subgraph_size = 3

def local_degrees(matrix):
    return np.sum(matrix > 0, axis=1)

degrees = local_degrees(adjacency_matrix)

def external_connections(subgraph, candidate, matrix):
    connections = 0
    for i in range(len(matrix)):
        if i not in subgraph and matrix[candidate][i] > 0:
            connections += 1
    return connections

while len(used_vertices) < num_vertices:
    available_vertices = [i for i in range(num_vertices) if i not in
used_vertices]
    min_deg = min(degrees[v] for v in available_vertices)
    candidates = [v for v in available_vertices if degrees[v] == min_deg]
    start_vertex = candidates[0]

    subgraph = [start_vertex]
    used_vertices.add(start_vertex)

    while len(subgraph) < max_subgraph_size:
        neighbors = [i for i in available_vertices if i not in subgraph and
np.any(adjacency_matrix[i][subgraph])]

        if not neighbors:
```

```
break

connection_gains = [(v, external_connections(subgraph, v,
adjacency_matrix)) for v in neighbors]
next_vertex, min_gain = min(connection_gains, key=lambda x:
x[1])

if min_gain > max_external_edges:
    break

subgraph.append(next_vertex)
used_vertices.add(next_vertex)

subgraphs.append(subgraph)

print("Сформовані підграфи:")
for i, sg in enumerate(subgraphs):
    print(f"Підграф {i+1}: вершини {sg}")
```

Результатом роботи програми є послідовне формування підграфів із урахуванням конструктивних обмежень, що відповідає практичним задачам розміщення елементів на комутаційних платах.

Висновки. Проведене дослідження показує, що послідовний алгоритм компоновання на основі графової моделі забезпечує ефективний підхід до оптимізації структур мультимедійних та радіоелектронних пристроїв на початкових етапах їх проектування. Застосування математичного формалізму у вигляді мультиграфа дозволяє знизити кількість міжмодульних з'єднань, підвищити компактність конструкції та технологічність виробництва. Програмна реалізація алгоритму на мові Python підтверджує практичну доцільність запропонованого методу, дозволяє автоматизувати процес компоновання та може бути адаптована для широкого спектра інженерних задач, що потребують оптимального розміщення функціональних елементів із врахуванням конструктивних обмежень.

Список літератури:

1. Саломатін В.А. Послідовний алгоритм компоновання конструктивних елементів на основі завдання схеми у вигляді гіперграфа. / В.А. Саломатін, В.М. Струнілін // Наукові праці ДонНТУ серія «Інформатика, кібернетика та обчислювальна техніка». – 2009. №10 (153). – С. 198 – 201.

2. J.G. Fernandes, R.E. Giudici Isomorphism between Cayley (di) graphs. *Discrete Mathematics*. – 2005. №305. – P. 361-364.
3. M.-Y. Xu, Some work on vertex-transitive graphs by Chinese mathematicians, in: *Group Theory in China*, Science Press/Kluwer Academic Publishers, Beijing/New York, 1996, pp. 224-254.
4. R.-J. Wang, M.-Y. Xu, A classification of symmetric graphs of order $3p$, *J. Combin. Theory Ser. B* 58 (1993) 197-216.
5. N. Biggs *Algebraic Graph Theory*. Cambridge University Press, Cambridge, 1974.
6. Мірошник М. и др. Метод аналізу помилок умов переходів станів у графових моделях кінцевих автоматів для систем автоматизованого синтезу // Інформаційно-керуючі системи на залізничному транспорті. – 2025. – Т. 30. – №. 2. – С. 107-117.
7. Miroshnyk M. et al. Graphical Models for Defectoscopy of Dielectrics //2023 XXXIII International Scientific Symposium Metrology and Metrology Assurance (MMA). – IEEE, 2023. – С. 1-5.
8. Miroshnyk M. et al. Graphic representation of component algorithms in multimedia devices // 2019 XXIX International Scientific Symposium" Metrology and Metrology Assurance" (MMA). – IEEE, 2019. – С. 1-4.
9. Мірошник М.А. Ітераційні алгоритми компонування в конструкціях мультимедіа / М.А. Мірошник, Корольова Я.Ю. // Інформаційно-керуючі системи на залізничному транспорті. – 2019. – №6. – С 3-7.
10. Корольова Я.Ю. Використання послідовного алгоритму компонування в конструкціях мультимедіа. / Я.Ю. Корольова, И.С. Беликов // Інформаційно-керуючі системи на залізничному транспорті. – 2018. – №6. – С 22-25.

References:

1. Salomatin, V.A. and Strunilin, V.M. (2009) ‘Sequential algorithm for arranging structural elements based on a hypergraph representation’, *Scientific Works of DonNTU. Series: Informatics, Cybernetics and Computer Engineering*, 10(153), pp. 198–201.
2. Fernandes, J.G. and Giudici, R.E. (2005) ‘Isomorphism between Cayley (di)graphs’, *Discrete Mathematics*, 305, pp. 361–364.
3. Xu, M.-Y. (1996) ‘Some work on vertex-transitive graphs by Chinese mathematicians’, in *Group Theory in China*. Beijing/New York: Science Press/Kluwer Academic Publishers, pp. 224–254.
4. Wang, R.-J. and Xu, M.-Y. (1993) ‘A classification of symmetric graphs of order $3p$ ’, *Journal of Combinatorial Theory, Series B*, 58, pp. 197–216.
5. Biggs, N. (1974) *Algebraic Graph Theory*. Cambridge: Cambridge University Press.
6. Miroshnyk, M. et al. (2025) ‘Method for analysing errors in state transition conditions in graph models of finite automata for automated synthesis systems’, *Information and Control Systems in Railway Transport*, 30(2), pp. 107–117.
7. Miroshnyk, M. et al. (2023) ‘Graphical models for defectoscopy of dielectrics’, in *Proceedings of the XXXIII International Scientific Symposium Metrology and Metrology Assurance (MMA)*. IEEE, pp. 1–5.

8. Miroshnyk, M. et al. (2019a) 'Graphic representation of component algorithms in multimedia devices', in *Proceedings of the XXIX International Scientific Symposium Metrology and Metrology Assurance (ММА)*. IEEE, pp. 1–4.
9. Miroshnik, M.A. and Koroleva, Y.Y. (2019b) 'Iterative algorithms for layout in multimedia designs', *Information and Control Systems in Railway Transport*, (6), pp. 3–7.
10. Koroleva, Y.Y. and Belikov, I.S. (2018) 'Use of a sequential layout algorithm in multimedia designs', *Information and Control Systems in Railway Transport*, (6), pp. 22–25.

Представив д-р техн. наук, проф., професор кафедри автоматизації проектування обчислювальної техніки Харківський Національний Університет Радіоелектроніки Кривуля Генадій Федорович.

Надійшла 11.01.2026

Стаття прийнята до друку 02.02.2026

Опублікована 27.02.2026

Miroshnyk Maryna, Doctor of Technical Sciences, Professor, Professor
Department of computer systems and robotics, V.N. Karazin Kharkiv National
University, Svobody Sq., 4, Kharkiv, Ukraine, 61012

e-mail: m.miroshnyk@karazin.ua

<https://orcid.org/0000-0002-2231-2529>

Koroleva Yana, Ph.D., associate professor of the Department of Multimedia
and internet technologies and systems, National Technical University «Kharkov
Polytechnic Institute». 2, Курпичова str., 61002, Kharkiv, Ukraine

e-mail: Yanakoroleva815@gmail.com

<https://orcid.org/0000-0002-7203-5603>.

Miroshnyk Anatolii, Associate Professor of the Department of Computer
Engineering Design Automation Kharkiv National University of Radio Electronics, 14
Nauky Ave, Kharkiv, Ukraine, 61000.

e-mail: yayaska@btu.kharkiv.ua

<https://orcid.org/0000-0002-3168-5351>

Shafranskyi Andrei, PhD student of the Department of of computer systems
and robotics, Kharkiv National University named after V.N. Karazin Kharkiv
National University, Svobody Sq., 4, Kharkiv, Ukraine, 61022

e-mail: shafranskyi.andrei@student.karazin.ua

<https://orcid.org/0009-0004-7725-3556>

УДК 004.03

Графова модель та програмна реалізація послідовного алгоритму компоновання для мультимедійних пристроїв / Мірошник М.А., Корольова Я.Ю., Мірошник А.М., Шафранський А.В. // Вісник НТУ "ХПІ". Серія: Інформатика та моделювання. – Харків: НТУ "ХПІ". – 2026. – № 1 (15). – С. 121 – 133.

У роботі розглянуто задачу компоновання елементів мультимедійних пристроїв на основі графової математичної моделі. Запропоновано та проаналізовано послідовний алгоритм компоновання, що дозволяє здійснювати покрокове формування підграфів із урахуванням конструктивних обмежень та мінімізації кількості зовнішніх зв'язків між підграфами. Кожному елементу пристрою відповідає вершина графа, а зв'язки між ними відображаються у вигляді ребер. Для демонстрації роботи алгоритму розроблено програмну реалізацію на мові Python, яка дозволяє в автоматизованому режимі виконувати етапи вибору вершин та формування підграфів відповідно до заданих критеріїв. Наведено приклад роботи програми, що підтверджує ефективність запропонованого підходу для оптимізації структури мультимедійних пристроїв. Л.: 1. Бібліогр.: 10 назв.

Ключові слова: модель, граф, вершина, ребро, зв'язки, компоновання, алгоритм, оптимізація, мультимедійний пристрій, мінімізація, програмна реалізація Python, автоматизований режим, ефективність.

UDC 004.03

Graph model and software implementation of a sequential layout algorithm for multimedia devices / Miroshnyk M.A., Korolyova Ya.Yu., Miroshnyk A.M., Shafransky A.V. // Herald of the National Technical University "KhPI". Series of "Informatics and Modeling". – Kharkiv: NTU "KhPI". – 2026. – № 1 (15). – P. 121 – 133.

The paper considers the problem of arranging elements of multimedia devices based on a graph mathematical model. A sequential arrangement algorithm is proposed and analyzed, which allows step-by-step formation of subgraphs, taking into account design constraints and minimizing the number of external connections between subgraphs. Each device element corresponds to a graph vertex, and the connections between them are represented as edges. To demonstrate the algorithm, a software implementation in Python has been developed, which allows the stages of vertex selection and subgraph formation to be performed automatically according to specified criteria. An example of the program's operation is provided, confirming the effectiveness of the proposed approach for optimizing the structure of multimedia devices. Figs.: 1. Bibl.: 10 items.

Keywords: model, graph, vertex, edge, connections, layout, algorithm, optimization, multimedia device, minimization, Python software implementation, automated mode, efficiency.