

УДК 681.32

**О.І. Білой, В.М. Струнілін**Донецький національний технічний університет  
кафедра комп'ютерних наук і технологій**РОЗРОБОТКА АЛГОРИТМУ МІНІМІЗАЦІЇ  
ВНУТРИШНЬОСХЕМНИХ ПЕРЕСІКАНЬ ПЕЧАТНОГО МОНТАЖУ****Анотація**

*Білой О.І., Струнілін В.М. Розробка алгоритму мінімізації внутришньосхемних пересікань печатного монтажу. Виконано аналіз алгоритмів мінімізації внутришньосхемних пересікань печатних з'єднань. Пропонується метод розбиття графа пересікань схеми на яруси, що дозволяє виконати оптимальне розшарування печатних плат.*

**Ключеві слова:** розшарування, печатна плата, внутришньосхемні пересікання.

**Постановка проблеми.**

Сучасні САПР конструкторського проектування радіоелектроніки виконують компоновку і розташування елементів монтажних схем, трасування і розшарування електричних з'єднань по шарах. Задача розшарування печатних з'єднань є однією з найбільш трудомістких, що має місце при реалізації з'єднань у монтажному просторі. Метою цього етапу є розміщення провідників по шарах без пересічень. Критерієм якості розшарування є мінімум кількості шарів, рівномірність розподілу провідників по шарах. При цьому від того, наскільки вдало вирішена ця задача, залежать такі характеристики спроектованого пристрою як надійність і вартість.

**Аналіз літератури.** При вирішенні задачі конструкторського проектування комутаційну електричну схему можна представити з використанням теорії графів у вигляді математичних моделей: графу комутаційної схеми (ГКС), мультиграфу комутаційної схеми (МКС), гіперграфу комутаційної схеми (ГКС) [1, 3].

В літературі [1, 2] описано наступні методи розшарування печатних з'єднань.

**Послідовний метод виділення плоских суграфів.**

Ідея цього методу полягає в тому, що на кожному кроці алгоритму виділяється ребро з максимальною кількістю пересікань та переноситься на наступний шар. Процес триває до тих пір, поки на першому шарі не буде пересікань. Вся процедура виконується до тих пір, поки на останньому шарі не буде пересікань.

По матриці  $Q$  необхідно створити вектор-стовпець  $Ві = \|bill$ , елементи якого відповідають числу пересікань ребра  $U_i$  з іншими

ребрами. Метою є знайти ребро, яке необхідно перемістити на інший шар:  $b_k = \max(b_i)$ .

Усі ненульові елементи  $b_1, b_2, \dots$ , що відповідають ребрам  $k_1, k_2, \dots$ , і що пересікаються з ребром  $k$ , зменшуються на одиницю. Внаслідок цього одержимо вектор-стовпець  $B_1$ . З вектором  $B_1$  поступаємо аналогічним чином до тих пір, поки він не стане нульовим.

Через  $U_1$  позначимо множину ребер, що ми усунули на другий шар, на першому шарі залишаться  $U \setminus U_1$ . Перший плоский суграф буде містити  $U \setminus U_1$  ребер. Аналогічним чином виділяємо другий плоский суграф, що буде містити  $U_1 \setminus U_2$ , де  $U_2$  - множина ребер, які усунути на третій шар і т. д. Слід мати на увазі, що кожний елемент вектору  $B_2, B_3, \dots, B_k$  відповідає кількості пересікань ребра, видаленого на  $2, 3, \dots, k$  шар з ребрами  $2, 3, \dots, k$  шару.

### **Метод мінімізації внутрішньосхемних пересікань на основі розфарбування графа пересікань.**

Початковими даними є:  $G = (E, U)$   $G' = (E', U')$ :  $E' \leftrightarrow U$ .

Подальше завдання укладається у розфарбуванні графа з найменшою кількістю кольорів, що відповідає розбивці початкового графа  $G$  на найменше число суграфів.

При розфарбуванні графа  $G'$   $k$  - хроматичному числу надаються значення  $k=1, 2, 3, \dots, n$ . Крім того, треба обрати таке розфарбування, при якому число вершин кожного із цих підмножин було приблизно однаковим.

Завдання вирішується наступним чином. Із графа  $G'$  послідовно виймаємо вершини разом з інцидентними їм ребрами, локальна міра  $\rho$  кожного з яких менш  $k$ . Причому, починаємо з вершини з найменшим значенням  $\rho$ . Якщо на деякому кроці при видаленні буде декілька вершин з найменшим і однаковим значенням  $\rho$ , отже вибираємо меншу по номеру. Процес повторюємо до тих пір, поки вершини, що залишилися, не будуть мати локальну міру  $\rho$ , яка більш або дорівнює  $k$ , або вершин взагалі не залишиться.

Множину вершин, що ми видалили, позначимо через  $S$ .

Далі виділяємо множину  $S_1$  наступним чином.

Позначимо через  $G_1'$  граф, що ми одержуємо в наслідок видалення з  $G'$  вершин та інцидентних їм ребер множини  $S$ . В графі  $G_1'$  послідовно видаляються вершини і інцидентні їм ребра з максимальною локальною мірою  $\rho$ . Видаляємо до тих пір, поки не одержимо не суміжні між собою вершини. Ця множина вершин  $S_1'$  є частиною  $S_1$ . Вершини, котрі ми видалили з інцидентними їм ребрам з  $G_1'$ , утворять новий граф пересікань  $G_1''$ . Якщо в  $G_1''$  є вершини, у яких  $\rho < k-1$ , то ми їх також видаляємо та заносимо в список  $S$ , а в графі пересікань, що залишився, аналогічно визначається підмножина не суміжних між собою вершин:  $S_1'' \subset S_1'$ ,  $S_2''$

$\subset S2' \dots$ , причому в множину  $S$  додаються вершини, у яких  $\rho < k-1$ ,  $\rho < k-2$ , і т. д.

Після визначення останньої множини  $S_k$  в графі може опинитися не пуста множина вершин. Позначимо її через  $S'$ . Далі, вершини із кортежу  $S$  в порядку, зворотному включенню їх в кортеж, розподіляємо по підмножинам  $S1'$ ,  $S2' \dots S_k'$  так, щоб не порушити розфарбування, тобто, щоб в кожній підмножині не з'явилося жодного ребра і щоб кількість вершин була приблизно однаковою.

Аналогічно поступаємо з множиною  $S'$ . Якщо це зробити з множиною  $S'$  не вдається, не порушивши розфарбування, то розфарбувати граф  $G$  в  $k$  кольорів неможливо. Треба збільшувати значення  $k$ .

**Мета статті** — розробка оптимального алгоритму мінімізації внутрішньосхемних пересікань печатного монтажу багат шарових печатних плат на основі розбиття графу пересічень на яруси.

### Рішення задачі і результати досліджень.

Хай маємо початкове розміщення графа схеми в позиціях комутаційного поля (КП). Будемо вважати, що ребра графа жорстко розташовані в вузлах КП. Необхідно вирішити завдання мінімізації внутрішньосхемних пересікань, яке пов'язане з пересіканням ребер графа, відображених на КП.

Граф схеми необхідно розтрити на  $m$  плоских суграфів, так, щоб  $m$  прагнуло до найменшого. Початкову інформацію про пересічення задаємо у вигляді:

$$a) \text{ матриці пересікань } Q = Q \left\| Q_{ij} \right\|_{|U_i \times U_j|}; \quad Q_{ij} = \begin{cases} 1 \\ 0 \end{cases},$$

де  $Q_{ij} = 1$ , якщо ребро  $U_i$  та ребро  $U_j$  пересікаються між собою;

$Q_{ij} = 0$ , якщо ребро  $U_i$  та ребро  $U_j$  не пересікаються між собою;  $U_i, U_j$  - ребра графа.

б) списку пересікань: для будь-якого  $U_i$  записуємо список ребер ( $U_{i1}, U_{i2} \dots U_{ik}$ ), що пересікаються з  $U_i$ .

Кожний плоский суграф взаємно відповідає шару друкованої плати. Таким чином, мінімальне число плоских суграфів дає мінімальне число шарів.

Розглянемо приклади мінімізації кількості внутрішньосхемних пересікань трьома методами. Метою є виявлення умов та обмежень, при дотриманні котрих той чи інший метод стає доцільним.

Хай граф пересікань буде завданий у вигляді матриці пересікань (табл.1).

Таблиця 1 – Матриця пересікань

|                 | U <sub>0</sub> | U <sub>1</sub> | U <sub>2</sub> | U <sub>3</sub> | U <sub>4</sub> | U | U <sub>6</sub> | U <sub>7</sub> | U <sub>8</sub> | U <sub>9</sub> | U <sub>1</sub> | U <sub>1</sub> | U <sub>1</sub> |
|-----------------|----------------|----------------|----------------|----------------|----------------|---|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| U <sub>0</sub>  | 0              | 0              | 1              | 1              | 1              | 1 | 0              | 0              | 0              | 0              | 0              | 0              | 0              |
| U <sub>1</sub>  | 0              | 0              | 0              | 0              | 1              | 0 | 0              | 0              | 0              | 0              | 0              | 0              | 0              |
| U <sub>2</sub>  | 1              | 0              | 0              | 0              | 0              | 0 | 0              | 0              | 1              | 1              | 0              | 0              | 0              |
| U <sub>3</sub>  | 1              | 0              | 0              | 0              | 1              | 1 | 0              | 0              | 0              | 0              | 0              | 0              | 0              |
| U <sub>4</sub>  | 1              | 1              | 0              | 1              | 0              | 0 | 0              | 0              | 1              | 0              | 0              | 0              | 0              |
| U <sub>5</sub>  | 1              | 0              | 0              | 1              | 0              | 0 | 0              | 0              | 0              | 0              | 0              | 0              | 0              |
| U <sub>6</sub>  | 0              | 0              | 0              | 0              | 0              | 0 | 0              | 0              | 0              | 0              | 1              | 0              | 1              |
| U <sub>7</sub>  | 0              | 0              | 0              | 0              | 0              | 0 | 0              | 0              | 0              | 0              | 1              | 1              | 1              |
| U <sub>8</sub>  | 0              | 0              | 1              | 0              | 1              | 0 | 0              | 0              | 0              | 1              | 1              | 0              | 1              |
| U <sub>9</sub>  | 0              | 0              | 1              | 0              | 0              | 0 | 0              | 0              | 1              | 0              | 1              | 1              | 0              |
| U <sub>10</sub> | 0              | 0              | 0              | 0              | 0              | 0 | 1              | 1              | 1              | 1              | 0              | 1              | 1              |
| U <sub>11</sub> | 0              | 0              | 0              | 0              | 0              | 0 | 0              | 1              | 0              | 1              | 1              | 0              | 0              |
| U <sub>12</sub> | 0              | 0              | 0              | 0              | 0              | 0 | 1              | 1              | 1              | 0              | 1              | 0              | 0              |

Розглянемо послідовний метод виділення плоских суграфів.

Завдаємо кількістю шарів  $K=2$ , тобто граф треба розбити по можливості на два плоских суграфи.

Відповідно до алгоритму сформуємо вектори-стовпці  $V$ :

|       |    |    |    |    |    |    |    |   |    |    |    |    |    |   |    |    |    |   |
|-------|----|----|----|----|----|----|----|---|----|----|----|----|----|---|----|----|----|---|
| $V =$ | 0  | 4  | 0  | 4  | 0  | 4* | 0  | 0 | 0  | 0  | 0  | 0  | 0  | 0 | 0  | 0  | 0  |   |
|       | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1 | 1  | 1  | 1  | 1* | 1  | 0 | 1  | 0  | 0  |   |
|       | 2  | 3  | 2  | 2  | 2  | 2  | 1  | 2 | 1  | 2  | 1  | 2  | 0  | 2 | 0  | 2  | 0  |   |
|       | 3  | 3  | 3  | 3  | 3  | 3  | 2* | 3 | 0  | 3  | 0  | 3  | 0  | 3 | 0  | 3  | 0  |   |
|       | 4  | 4  | 4  | 3  | 4  | 3  | 4  | 2 | 4  | 1  | 4  | 1  | 4  | 1 | 4  | 0  | 4  | 0 |
|       | 5  | 2  | 5  | 2  | 5  | 2  | 5  | 1 | 5  | 0  | 5  | 0  | 5  | 0 | 5  | 0  | 5  | 0 |
|       | 6  | 2  | 6  | 2  | 6  | 1  | 6  | 1 | 6  | 1  | 6  | 1  | 6  | 1 | 6  | 1* | 6  | 0 |
|       | 7  | 3  | 7  | 3  | 7  | 2  | 7  | 2 | 7  | 2* | 7  | 0  | 7  | 0 | 7  | 0  | 7  | 0 |
|       | 8  | 6* | 8  | 0  | 8  | 0  | 8  | 0 | 8  | 0  | 8  | 0  | 8  | 0 | 8  | 0  | 8  | 0 |
|       | 9  | 4  | 9  | 3  | 9  | 2  | 9  | 2 | 9  | 2  | 9  | 2* | 9  | 0 | 9  | 0  | 9  | 0 |
|       | 10 | 6  | 10 | 5* | 10 | 0  | 10 | 0 | 10 | 0  | 10 | 0  | 10 | 0 | 10 | 0  | 10 | 0 |
|       | 11 | 4  | 11 | 3  | 11 | 2  | 11 | 2 | 11 | 2  | 11 | 1  | 11 | 0 | 11 | 0  | 11 | 0 |
|       | 12 | 4  | 12 | 3  | 12 | 2  | 12 | 2 | 12 | 2  | 12 | 1  | 12 | 1 | 12 | 1  | 12 | 0 |

\* відмічені елементи з максимальною кількістю зв'язків на кожному кроці. Вони заносяться до списку  $U_1$ .

До другого шару надійдуть  $U_1 = \{8, 10, 0, 3, 7, 9, 1, 6\}$ .

До першого шару надійдуть  $U_1 = \{2, 4, 5, 11, 12\}$ .

При використанні метода мінімізації пересікань на основі розфарбування графа пересікань граф пересікань, який, завданий у вигляді матриці пересікань (табл.1), розіб'ємо на  $K$  плоских суграфи.  $K=4$ .

Згідно алгоритму сформуємо список-кортеж  $S$ :

$S = (1, 5, 3, 0, 4, 2, 6, 7, 12, 8, 9, 10, 11)$ .

У порядку, зворотному включенню елементів до списку, розподіляємо їх по  $K$  підмножин, так щоб у довільному підмножині не з'явилось жодного ребра:

$S_1 = (11,12,4)$  - перший шар;

$S_2 = (10,2,1,3)$  - другий шар;

$S_3 = (9,7,0)$  - третій шар;

$S_4 = (8,6,5)$  - четвертий шар.

**Пропонується алгоритм зменшення внутрішньосхемних пересікань на основі розбиття графа пересічень на яруси.**

Суть алгоритму, що пропонується, укладається у наступному.

В графі пересічень шукається вершина  $E_0$  з максимальною локальною мірою  $\rho$ . Якщо опиниться декілька вершин з максимальним і однаковим значенням  $\rho$ , то вибираємо меншу по номеру. Вершині  $E_0$  надаємо вагу, яка дорівнює 0. Далі розповсюджується числова хвиля, поки її фронт не досягне всіх вершин, суміжних із  $E_0$ .

Формування чергового фронту числової хвилі  $F_k$  починається із надання всім суміжним з вершинами фронту  $F_{k-1}$  ваги  $P_k = P_{k-1} + 1$ . Процес триває до тих пір, поки всім вершинам не буде надана вага. Далі множину вершин з парною вагою вносимо в список  $L_0$ , множину вершин з непарною вагою вносимо в список  $L_1$ . Очевидно, що множина вершин списку  $L_0$  або  $L_1$  не будуть мати між собою ребер. Це справедливо лише в разі біхроматичності графа пересічень, тобто в ньому не повинно бути циклів непарної довжини. В цьому випадку одержуємо розбиття вихідного графа пересічень на два плоских суграфа.

Якщо це не так, то вибираємо вершину, що має максимальну кількість ребер всередині кожного списку-ярусу, і ця вершина переноситься на наступний ярус. Процес триває до тих пір, поки всередині кожного з ярусів не залишиться множина не суміжних між собою вершин. Треба прагнути, щоб кількість вершин всередині кожного ярусу було приблизно однакова. В підсумку одержуємо розбиття графа пересічень на плоскі суграфи, кількість яких відповідає кількості сформованих алгоритмом ярусів.

Граф пересікань, який задається у вигляді матриці пересікань (табл.1) треба розбити на  $K$  плоских суграфи.  $K=4$ .

Згідно алгоритму надамо кожному ребру графа вагу (табл.2).

Таблиця 2–Вага ребер графа

| Ребро $U_i$ | $U_0$ | $U_1$ | $U_2$ | $U_3$ | $U_4$ | $U_5$ | $U_6$ | $U_7$ | $U_8$ | $U_9$ | $U_{10}$ | $U_{11}$ | $U_{12}$ |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|----------|----------|
| Вага        | 3     | 3     | 2     | 3     | 2     | 4     | 1     | 1     | 1     | 1     | 0        | 1        | 1        |

Згідно алгоритму розподіляємо ребра по спискам

$L_0 = (U_2, U_4, U_5, U_{10});$   $L_1$

$= (U_0, U_1, U_3, U_6, U_7, U_8, U_9, U_{11}, U_{12}).$

Аналізуємо список  $L_0$ . Множина елементів цього списку – не суміжні між собою вершини, тому це є перший плоский суграф.

Аналізуємо список  $L_1$ . Ребро  $U_8$  має пересічення з ребром  $U_9$ , тому відносимо його до списку  $L_2$ .  $L_2=(U_8)$ .

Список  $L_1$  прийме вигляд  $L_1=(U_0, U_1, U_3, U_6, U_7, U_9, U_{11}, U_{12})$ . Ребро  $U_0$  має пересічення з ребром  $U_3$ , тому відносимо його до списку  $L_2$ .  $L_2=(U_8, U_0)$ .

Список  $L_1$  прийме вигляд  $L_1=(U_1, U_3, U_6, U_7, U_9, U_{11}, U_{12})$ . Ребро  $U_9$  має пересічення з ребром  $U_{11}$ . Це ребро не можливо розподілити до існуючих списків, тому відносимо його до списку  $L_3$ .  $L_3=(U_9)$ .

Список  $L_1$  прийме вигляд  $L_1=(U_1, U_3, U_6, U_7, U_{11}, U_{12})$ . Отже, у кожному списку  $L_0 - L_3$  знаходиться множина не суміжних між собою ребер, що відповідає поставленій задачі.

$L_0 = (U_2, U_4, U_5, U_{10})$  - перший плоский суграф;

$L_1 = (U_1, U_3, U_6, U_7, U_{11}, U_{12})$  - другий плоский суграф;

$L_2 = (U_8, U_0)$  - третій плоский суграф;

$L_3 = (U_9)$  - четвертий плоский суграф.

Треба відзначити, що алгоритм розбиття графа пересікань на яруси працює тільки зі зв'язними графами, тому при реалізації його на ЕОМ його треба модифікувати.

Для дослідження методів розшарування була розроблена програма, інтерфейс якої наведено на рис. 1.

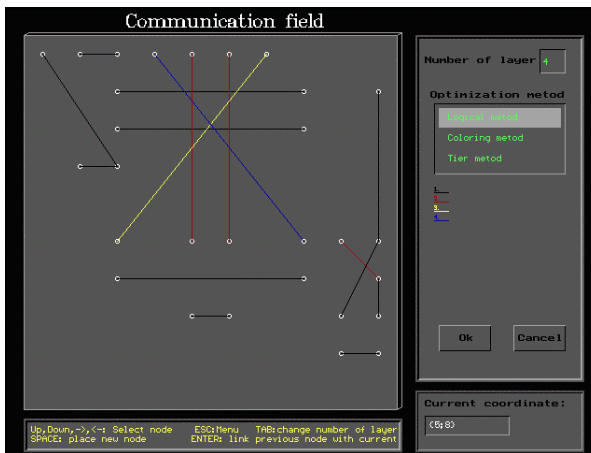


Рисунок 1–Інтерфейс програми розшарування

Дати порівняльну характеристику методів розшарування дуже важко, тому що всі розглянуті методи є евристичними алгоритмами. Як звісно, евристичні алгоритми не піддаються аналітичній оцінці, тому було прийнято рішення провести порівняльну характеристику методів шляхом аналізу результатів роботи кожного алгоритму з графами пересікань різної складності.

У таблиці 3 наведено порівняльну характеристику методів розшарування.

Таблиця 3– Порівняльна характеристика методів розшарування

| Параметр                                      | Метод розшарування   |                      |   |
|---|----------------------|----------------------|---|
|   | Послідовний          | Розфарбування        | Ярусний                                     |
| Тип графу                                     | Зв'язний, незв'язний | Зв'язний, незв'язний | Зв'язний, незв'язний (з доробкою алгоритму) |
| Швидкість                                     | Велика               | Низька               | Велика                                      |
| Виділення плоских суграфів                    | Не гарантується      | Гарантується         | Гарантується                                |
| Якість розшарування (рівномірність розподілу) | Низька               | Висока               | Висока                                      |
| Реалізація на ПЕОМ                            | Проста               | Складна              | Проста                                      |

**Висновки.** Аналіз результатів роботи алгоритмів розшарування печатних з'єднань з майже 50 графами пересікань дозволяє зробити висновок, що запропонований метод розбиття графа пересікань на яруси найбільш ефективний з точки зору швидкості реалізації, якості розшарування і простоти реалізації серед представлених методів.

### Перелік літератури

1. Автоматизированное проектирование узлов и блоков РЭС средствами современных САПР: Учеб. пособие вузов/Под ред. И.Г. Мироненко. – М., 2002. – 391 с.

2. Селютин В. А. Автоматизированное проектирование топологии БИС. - М.; Радио и связь, 1983. – 112с.

3. Модифікація ітераційного алгоритму компоновки на базі гіперграфової моделі електричної схеми/ Білої О.І., Струнілін В.М. Інформаційні управляючі системи та комп'ютерний моніторинг (ІУСКМ-2012): III Всеукраїнська науково-технічна конференція студентів, аспірантів та молодих вчених, 16-18 квітня 2012 р., м. Донецьк / Донец. націонал. техн. ун-т; редкол.: Є.О. Башков (голова) та ін.- Донецьк: ДонНТУ, 2012. - С. 567-571.

Как правильно ссылаться на этот доклад:

Розробка алгоритму мінімізації внутрішньо схемних пересікань печатного монтажу/ О.І. Білої, В.М. Струнілін. Інформаційні управляючі системи та комп'ютерний моніторинг (ІУС та КМ-2013): IV міжнародна науково-технічна конференція студентів, аспірантів та молодих науковців, 23-25 квітня 2013 р., м.Донецьк / Донец. націонал. техн. ун-т.- Донецьк: ДонНТУ, 2013. - С. 125-131.