УДК 621.382.33:681

І. Казимира, О. Крук
Національний університет "Львівська політехніка", кафедра ТРР

АНАЛІЗ ПРОБЛЕМИ ОПТИМАЛЬНОЇ ПЕРЕНУМЕРАЦІЇ ВУЗЛІВ СХЕМИ З ОГЛЯДУ ВПЛИВУ НА ЕФЕКТИВНІСТЬ СХЕМОТЕХНІЧНОГО ПРОЕКТУВАННЯ

Проаналізовано розв'язання проблеми оптимальної перенумерації вузлів схеми в процедурах автоматизованого схемотехнічного для забезпечення найбільшої ефективності процесу проектування. Детально проаналізовано методику розріджених матриць та визначено її основні задачі та специфіку їх розв'язання в схемотехнічному моделюванні.

Analysis of optimum renumeration of scheme nodes problem solving in procedures of automated schemotechnical design from most effective process of design point of view is undertaken in this article. Methodology of rarefied matrixes was analysed in details and main tasks and specisics of their solving in schemotechnical modeling were determined.

1. Вступ

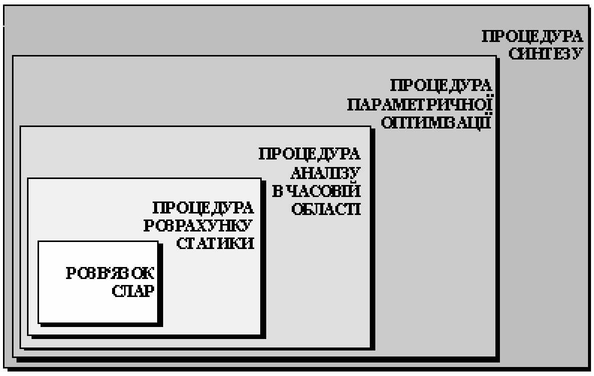
Забезпечення ефективності процесу автоматизованого схемотехнічного проектування (АСхП) передбачає пошук відповідної послідовності процедур та вибір, модифікацію чи розробку математичних моделей, що використовуються в цих процедурах, на кожному з горизонтальних ієрархічних рівнів – синтезу, оптимізації, аналізу та розрахунку [12].

Якщо розглянути найнижчий ієрархічний рівень – рівень розрахунку, то з одного боку ієрархічна структура процесу АСхП сильно обмежує пошук шляхів забезпечення ефективності на цьому рівні за допомогою вибору оптимальної послідовності проектних процедур і відповідних їм моделей. З іншого боку, враховуючи типову структуру автоматизованого процесу схемотехнічного проектування та ітераційність і вкладеність процедур, проектні процедури розрахунку та проектні операції, які вони містять, сильно впливають на ефективність процесу загалом [12]. Адже такі процедури і операції десятки і сотні разів використовуються в процедурах аналізу, сотні і тисячі – у процедурах оптимізації, і тисячі та мільйони разів у процедурах синтезу. Тому незначне підвищення ефективності проектних процедур і проектних операцій найнижчого рівня ієрархії – рівня розрахунку може значно підвищити ефективність всього процесу проектування.

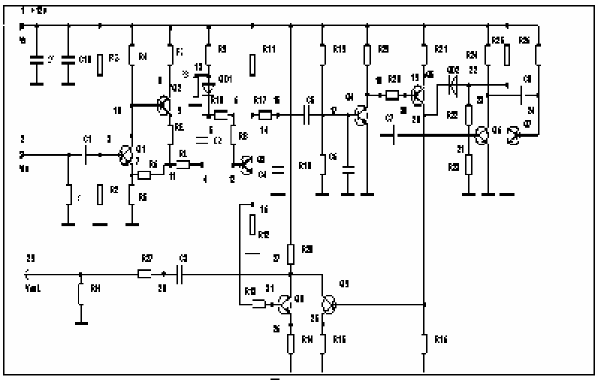
При пошуку тих методів і алгоритмів, що багатократно викликаються процедурами розрахунку, аналізу, оптимізації та синтезу, слід опиратися на парадигму щодо використання відповідних методів та алгоритмів у вищезгаданих процедурах [8, 12].

2. Процедура розв'язання СЛАР як показник ефективності процедур оптимального проектування загалом

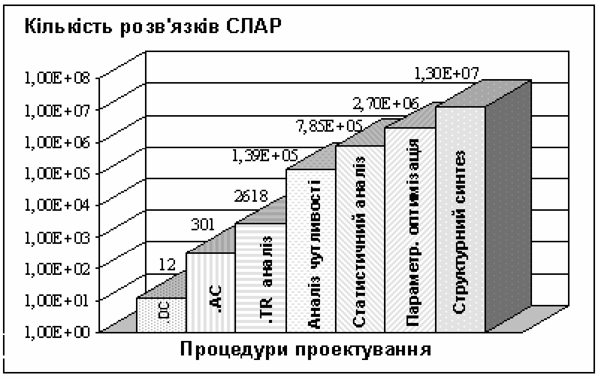
Числове розв'язання різних задач розрахунку, аналізу, оптимізації та синтезу електронних схем в остаточному випадку зводиться до розв'язання системи лінійних алгебраїчних рівнянь (СЛАР) [1,14,16,18]. Процедура розв'язання СЛАР – це процедура, яка сотні разів використовується в задачах розрахунку моделі схеми, тисячі – в задачах аналізу і мільйони і десятки мільйонів разів – в задачах оптимізації та синтезу. На рис. 1, а показана вкладеність процедури розв’язання СЛАР в задачах розрахунку, аналізу, оптимізації та синтезу. Як конкретний приклад багатократності розв’язання СЛАР в процедурах схемотехнічного проектування розглянемо схему інвертора відеосигналу, принципова схема якого показана на рис. 1, б. Діаграма зростання кількості розв’язків СЛАР показана на рис. 1, в. Результати оцінки кількості розв’язків СЛАР в процесі розрахунку статики з використанням модифікованої форми основного алгоритму Ньютона-Рафсона, аналізу в частотній області, аналізу перехідної характеристики в часовій області з використанням методу неявного інтегрування – методу трапецій, аналізу параметричної чутливості з використанням методу малих приростів, параметричній оптимізації схеми з використанням методу Розенброка та структурному синтезі схеми (оцінці п’яти варіантів структури) показані на рис. 1, г.



а)



б)



в)

№	Проектні операції та процедури проектування	Кількість розв´яз-ків СЛАР	Часові витрати T_M(сек)
1.	Розв´язання системи ЛАР	1	0.25667
2.	Розрахунок статики	12	3.08
3.	Аналіз в частотній області	301	69.92
4.	Аналіз перехідної характеристики	2618	655.54
5.	Оптимізація перехідної характеристики	$2.7*10^6$	$7.131*10^5$

г)

Рис.1. Статистика проектування:
а) вкладеність процедури розв’язання СЛАР; б) принципова схема відеоінвертора;
в) діаграма зростання кількості розв’язків СЛАР; г) оцінка кількості розв’язків СЛАР

Наведені результати яскраво ілюструють вплив ефективності процедури розв’язання СЛАР на ефективність всього процесу автоматизованого схемотехнічного проектування. Так, наприклад, підвищення швидкодії процедури розв’язання СЛАР всього на 10 мікросекунд скорочує часові витрати на процедуру параметричної оптимізації на 30 годин. Це власне і є тим визначальним фактором, що спричинює пошук шляхів підвищення ефективності процесу проектування на найнижчому ієрархічному рівні через модифікацію методів та алгоритмів, що реалізують методику розв’язання СЛАР із врахуванням особливостей об’єкта схемотехнічного проектування.

Задача розв'язання СЛАР в схемотехнічному моделюванні виникає при використанні методу вузлових потенціалів (який є обов'язковою частиною парадигми аналізу) при розрахунку статичного режиму. Цей розрахунок базується на лінеаризації на кожній ітерації модифікованого алгоритму Ньютона-Рафсона (алгоритму розв'язання системи нелінійних рівнянь) нелінійних рівнянь компонентів шляхом розкладу їх в ряд Тейлора з використанням перших членів ряду. На *k-й* ітерації алгоритму Ньютона-Рафсона розв'язується така СЛАР [26]:

$$\boldsymbol{Y}_n^{(k)} \times \boldsymbol{V}_n^{(k)} = \boldsymbol{I}_n^{(k)}, \tag{1}$$

де $\boldsymbol{Y}_n^{(k)}$– матриця вузлової провідності розмірністю $N \times N$, $\boldsymbol{V}_n^{(k)}$ – вектор невідомих вузлових напруг, $\boldsymbol{I}_n^{(k)}$ – вектор еквівалентних джерел струму, N – кількість вузлів у схемі.

Однією з особливостей об'єкта АСхП є те, що матриця вузлової провідності $\boldsymbol{Y}_n$ схеми є розрідженою матрицею, причому при зростанні кількості вузлів N розрідженість матриці зростає. Так, у схемі з $N=10$ розрідженість матриці становить не більше 50%, при $N=100$ – не більше *95%*. [3,25]. Очевидно, що неврахування цієї особливості при розв'язанні СЛАР є неприпустимим з точки зору забезпечення ефективності. Тому методика розріджених матриць і є обов'язковим елементом парадигми аналізу схем.

Якщо б необхідно було навести класичний приклад адаптивного алгоритму чи адаптивної методики, які підлаштовуються під особливості конкретного об'єкта проектування і використовують ці особливості з максимальною ефективністю, то кращого прикладу, ніж методика розріджених матриць у схемотехнічному проектуванні, напевно, не існує [21, 25]. Адже топологія електричних з'єднань у кожній принциповій схемі своєрідна, а отже, і структура матриці вузлової провідності, її розрідженість також особливі. Максимальне врахування розрідженості матриці для кожного окремого випадку для забезпечення максимальної ефективності розв'язання СЛАР і є завданням адаптивної методики розріджених матриць.

Основним завданням методики розріджених матриць є вирішення трьох проблем.

Проблема 1. Забезпечення зберігання в оперативній пам'яті тільки ненульових елементів матриці з необхідною індексною інформацією.

Проблема 2. Забезпечення невиконання операцій з нульовими елементами матриці.

Проблема 3. Максимальне збереження розрідженості матриці, яка може погіршуватись відносно початкової розрідженості в процесі розв'язання СЛАР.

Для вирішення проблем 1 і 2 в методиці розріджених матриць використовуються спеціальний метод розв'язання СЛАР та спеціальні методи програмування розріджених матриць. Для розв'язання СЛАР використовується один із загальних методів розв'язання – метод **LU**-розкладу [7,17-19,24,25]. Суть методу полягає в тому, що початкову матрицю $\boldsymbol{Y}_n$ рівняння (1) можна подати у вигляді добутку двох трикутних матриць:

$$\boldsymbol{Y}_n = \boldsymbol{L}_n \boldsymbol{U}_n, \tag{2}$$

де $\boldsymbol{L_n}$ – нижня трикутна матриця, $\boldsymbol{U_n}$ – верхня трикутна матриця з одиничною діагоналлю. Система (1) має тоді такий вигляд:

$$\boldsymbol{L}_n \boldsymbol{U}_n \boldsymbol{V}_n = \boldsymbol{I}_n \tag{3}$$

і розв'язується в два етапи [18]:

1-й етап називається прямим ходом:

$$\boldsymbol{L}_n \boldsymbol{Z} = \boldsymbol{I}_n \,, \tag{4}$$

2-й етап називається зворотним ходом:

$$\boldsymbol{U}_n \boldsymbol{V}_n = \boldsymbol{Z} \,. \tag{5}$$

Як метод виключення Гаусса, так і метод **LU**-розкладу для великих N вимагають приблизно $N^3/3$ операцій ділення і множення [25]. Тобто з точки зору витрат машинного часу метод **LU**-розкладу не має переваг перед методом виключення Гаусса. Реальна перевага методу **LU**-розкладу полягає в меншій помилці заокруглення і, в основному, в його застосовності в методиці розріджених матриць [11, 22]. Іншими словами, основною заслугою методу **LU**-розкладу можна вважати те, що поєднано з методикою програмування розріджених матриць [25], яка дозволяє зберігати тільки ненульові елементи матриці (розв'язує проблему 1), алгоритм методу дозволяє розв'язати проблему 2 – невиконання дій з нульовими елементами. Для розкладу матриці $\boldsymbol{Y}$ на **L** і **U**-множники використовуються алгоритм Краута [22] чи алгоритм Дулітла [32], який порівняно з першим є більш зручним з точки зору програмування розріджених матриць [25]. Розглянемо на прикладі роботи алгоритму Дулітла проблеми, що можуть виникати в процесі **LU**-розкладу та розв'язання системи.

У розгорненому вигляді співвідношення (2) можна записати як:

$$\begin{bmatrix} y_{11} & y_{12}, & \dots, & y_{1N} \\ y_{21} & y_{22}, & \dots, & y_{2N} \\ \dots & \dots & \dots, & \dots \\ y_{N1} & y_{N2}, & \dots, & y_{NN} \end{bmatrix} = \begin{bmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \dots & \dots & \dots & \\ l_{N1} & l_{N2}, & \dots, & l_{NN} \end{bmatrix} \times \begin{bmatrix} 1 & u_{12}, & \dots, & u_{1N} \\ & 1, & \dots, & u_{2N} \\ & & \dots & \dots \\ & & & 1 \end{bmatrix}. \tag{6}$$

Множники **LU** матриці $\boldsymbol{Y}$ можна подати як допоміжну матрицю θ:

$$\theta = \boldsymbol{L} + (\boldsymbol{U} - \boldsymbol{1}), \tag{7}$$

або

$$\begin{bmatrix} \theta_{11} & \theta_{12}, & \dots, & \theta_{1N} \\ \theta_{21} & \theta_{22}, & \dots, & \theta_{2N} \\ \dots & \dots & \dots, & \dots \\ \theta_{N1} & \theta_{N2}, & \dots, & \theta_{NN} \end{bmatrix} = \begin{bmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \dots & \dots & \dots & \\ l_{N1} & l_{N2}, & \dots, & l_{NN} \end{bmatrix} + \left(\begin{bmatrix} 1 & u_{12}, & \dots, & u_{1N} \\ & 1, & \dots, & u_{2N} \\ & & \dots & \dots \\ & & & 1 \end{bmatrix} - \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & .. & \\ & & & 1 \end{bmatrix} \right). \tag{8}$$

Розклад матриці $\boldsymbol{Y}$ на **LU**-множники за алгоритмом Дулітла наглядно можна проілюструвати схемою перетворення матриці $\boldsymbol{Y}$ в матрицю θ ($\boldsymbol{Y} \Rightarrow \theta$), що показана на рис.2.

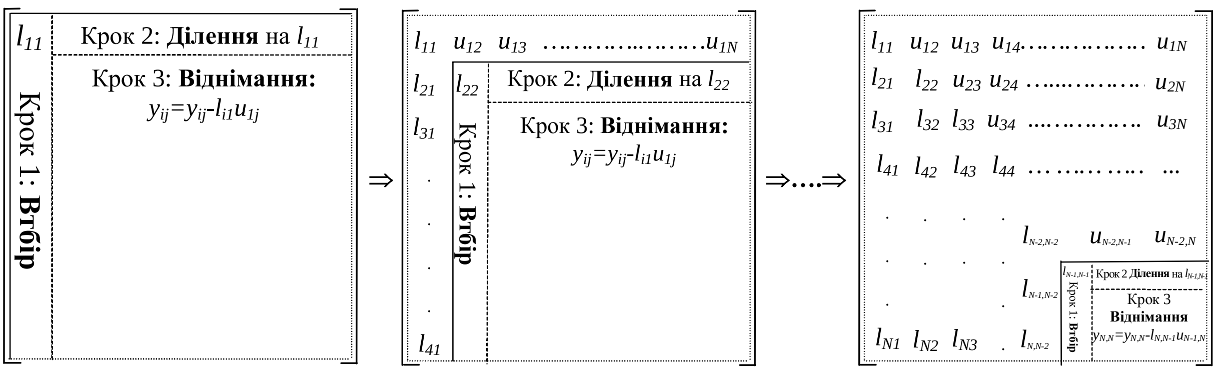


*Рис. 2. Схема виконання кроків алгоритму розкладу матриці **Y** на **LU**-множники*

Покроково алгоритм Дулітла можна записати так.

Вхідні дані алгоритму: матриця ***Y*** розмірністю $N \times N$, $k=1$.

Крок 1. *Вибір.* Вибирають *k*-й стовпець (без обчислень):

$$l_{ik} = y_{ik}^{(k-1)}, \quad i = \overline{k, N} . \tag{9}$$

Крок 2. *Ділення.* Ділять всі наддіагональні елементи *k*-го рядка на діагональний:

$$u_{kj} = \left(y_{kj}^{(k-1)} \big/ l_{kk} \right), \quad j = \overline{k+1, N} . \tag{10}$$

Крок 3. *Віднімання.* З кожного елементу $y_{ij}^{(k-1)}$, $i > k$, $j > k$ віднімають добуток елементів

$$l_{ik} \times u_{kj}, \quad i = \overline{k+1, N}; \quad j = \overline{k+1, N} :$$

$$y_{ij}^{(k)} = \left[y_{ij}^{(k-1)} - \left(l_{ik} \times u_{kj} \right) \right], \quad i = \overline{k+1, N}; \quad j = \overline{k+1, N} . \tag{11}$$

Крок 4. *Перевірка.* Якщо $(N - k) \geq 2$, то $k = k + 1$ і перехід на *Крок 1*; інакше отримана матриця є матрицею θ. *Кінець*.

Як видно з покрокового подання, алгоритм є доволі простим і прозорим. Для того, щоб зрозуміти, як за допомогою такого алгоритму розв’язується проблема невиконання дій з нульовими елементами, розглянемо методи програмування розріджених матриць, які використовуються для розв’язання проблеми зберігання в оперативній пам’яті тільки ненульових елементів. Наприклад, один із методів програмування для структурно симетричних матриць [3] передбачає запам’ятовування ненульових елементів матриці θ в таких трьох одновимірних масивах:

1. *DIAG* – де запам’ятовуються діагональні елементи матриці θ;

2. *UPACK* – де запам’ятовуються в упакованому виді ненульові недіагональні елементи матриці ***U*** по рядках;

3. *LPACK* – де запам’ятовуються в упакованому виді ненульові недіагональні елементи матриці ***L*** по стовпцях.

Індексна інформація міститься в двох одновимірних масивах:

1. *UCOL* – що містить інформацію про те, в яких стовпцях (чи рядках) матриці θ знаходяться елементи масиву *UPACK* (чи *LPACK*);

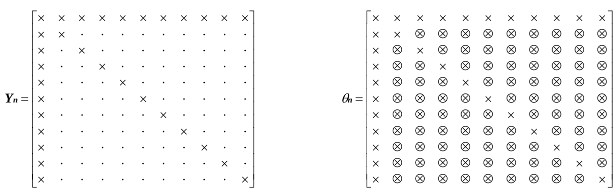
2. *UROWST* – в якому зберігається інформація про те, з якого елементу масиву *UPACK* (чи *LPACK*) починається кожний рядок (чи стовпець) матриці ***U*** (чи матриці ***L***).

Якщо сумісно проаналізувати роботу алгоритму Дулітла та структуру одновимірних масивів, в яких в упакованому виді зберігаються ненульові елементи матриці θ, то видно, що без ніякх зусиль проводиться вибір (крок 1), ділення (крок 2) та віднімання (крок 3) від діагональних елементів шляхом операцій з відповідними елементами масивів *DIAG*, *UPACK* і *LPACK*. Невеликі додаткові зусилля, а зокрема, обчислення прямої адресації, необхідні при відніманні від довільного недіагонального елементу матриці ***Y***. Також ніяких проблем не виникає при прямому ході – розв’язання системи (4) та зворотному ході – розв’язання системи (5). Отже схема запам’ятовування ненульових елементів спеціально “придумана” для полегшення **LU**-розкладу та реалізації прямого і зворотного ходів. Разом з алгоритмом **LU**-розкладу метод програмування розрідженої матриці забезпечує розв’язання проблем 1 і 2, що ставляться перед методикою розріджених матриць.

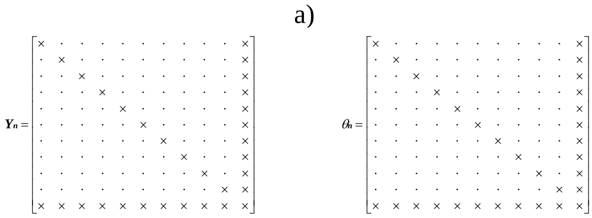
Залишається розв'язання проблеми 3 – максимальне збереження розрідженості матриці θ, яка може погіршуватись відносно початкової розрідженості матриці $\boldsymbol{Y}$ в процесі **LU**-розкладу. Виявляється, що всі зусилля щодо розв'язання проблем 1 і 2, які є основним завданням методики розріджених матриць, можуть бути зведені нанівець, оскільки отримана в результаті **LU**-розкладу матриця θ буде мати набагато меншу розрідженість, ніж початкова матриця $\boldsymbol{Y}$. Розрідженість погіршується з появою в процесі розкладу на множники в матриці θ нових ненульових елементів, так званих *заповнень*. Заповнення визначається так: якщо в початковій матриці $\boldsymbol{Y}$ елемент $y_{ij}=0$, а в результаті розкладу в матриці θ відповідний елемент $\theta_{ij} \neq 0$, то цей елемент називається заповненням (новим ненульовим елементом – ННЕ [18]). Як видно з описаного алгоритму розкладу, заповнення можуть виникати на третьому кроці при відніманні: $y_{ij}^{(k)} = \left[y_{ij}^{(k-1)} - \left(l_{ik} \times u_{kj} \right) \right]$, $i = \overline{k+1,N};\ j = \overline{k+1,N}$. Якщо $y_{ij}^{(k-1)} = 0$, а $l_{ik} \neq 0$ і $u_{ik} \neq 0$, то будемо мати заповнення $y_{ij}^{(k)} \neq 0$. Іншими словами, якщо на k-й ітерації елемент матриці **Y** дорівнює нулю ($y_{ij}^{(k-1)} = 0,\ i = \overline{k+1,N};\ \ j = \overline{k+1,N}$), а відповідні елементи k-го рядка і k-го стовпця не дорівнює нулю ($u_{kj} \neq 0 \wedge l_{ik} \neq 0$, $i = \overline{k+1,N}; j = \overline{k+1,N}$), то в матриці θ буде утворюватись заповнення: $\theta_{ij}^{(k)} = y_{ij}^{(k)} \neq 0$.

Процес утворення заповнень може лавиноподібно наростати від ітерації до ітерації (заповнення будуть породжувати нові заповнення) і в результаті матрицю θ отримуємо з низькою розрідженістю, що значно понижує ефективність методики вирахування розрідженості матриць.

З визначення заповнення видно, що породження заповнення можна уникнути, якщо хоча б один з елементів l_{ik} чи u_{kj} був би нульовим. Ця ситуація визначається як початковою структурою матриці **Y,** так і змінною структурою підматриць на кожній ітерації алгоритму розкладу (де порядок підматриці визначається як N-k). Іншими словами, кількість заповнень визначається структурою матриці $\boldsymbol{Y}$, яка визначає порядок виключення змінних. Справедливе і зворотне твердження – порядок виключення змінних визначає структуру матриці $\boldsymbol{Y}$, яка в свою чергу визначає кількість заповнень.



а)



б)

Рис. 3. Вплив початкової структури матриці ***Y*** *на заповненість матриці θ*
в процесі ***LU****-розкладу. (×– ненульовий елемент; •– нульовий елемент; ⊗– заповнення)*

Існують три причини, за якими заповнення треба вважати небажаним явищем [19]:

1. Необхідно виділяти пам'ять для зберігання нових ненульових елементів. При цьому міра заповнень може виявитись дуже значною. Відомий класичний приклад [31] найбільш невдалої початкової структури матриці, де вже на першій ітерації розкладу на множники проходить повне заповнення матриці (схематично структура матриці $\boldsymbol{Y}$ і θ для даного прикладу показана на рис.3, а). Цього можна уникнути і повністю зберегти початкову розрідженість матриці, якщо поміняти місцями 1-й і N-й рядок та 1-й і N-й стовпець (рис.3, б).

2. Машинний час, що витрачається на виконання **LU**-розкладу, швидко збільшується із ростом заповнень, оскільки доводиться виконувати набагато більшу кількість арифметичних операцій.

3. Границі похибок збільшуються разом з ростом заповнень. Надмірне заповнення матриці може привести до надто сильного накопичення обчислювальних похибок.

Отже, розв'язання проблеми 3, що вирішується методикою розріджених матриць, а саме мінімізація кількості заповнень при **LU**-розкладі, актуальне щодо забезпечення ефективності як процедури розв'язання СЛАР, так і всього процесу проектування.

Постановка задачі мінімізації кількості заповнень при **LU**-розкладі зводиться до оптимального упорядкування рівнянь у системі (визначення оптимального порядку виключення змінних чи оптимальної структури матриці) [6,22] і є еквівалентною задачі оптимальної перенумерації вузлів схеми в схемотехнічному проектуванні [25].

3. Проблеми розв'язання задачі оптимальної перенумерації схеми в схемотехнічному проектуванні

З погляду забезпечення ефективності розв'язання проблеми 3 методикою розріджених матриць є надзвичайно важливим. Успішність розв'язання задачі збереження максимальної розрідженості матриці вузлової провідності (структура якої визначається топологією електричних з'єднань конкретної схеми) буде визначати ступінь адаптивності методики розріджених матриць до конкретного об'єкта проектування та її ефективність загалом.

Отже, початкова розрідженість матриці вузлової провідності схеми визначається топологією електричних з'єднань. Структура матриці, від якої залежить заповненість **LU** множників, буде визначатися нумерацією вузлів схеми. Щодо чиселового розв'язання СЛАР для вирішення проблеми 3 ставиться така задача: задана розріджена матриця $\mathbf{Y_n}$, провести оптимальне упорядкування рядків і стовпців матриці так, щоб **L** і **U** матриці були максимально розрідженими.

З точки зору проектувальника схем така задача трансформується в таку дуже важливу задачу етапу схемотехнічного проектування: задана схема, знайти найкращу (оптимальну) нумерацію вузлів схеми, щоби матриця вузлової провідності $\mathbf{Y_n}$ мала множники **L** і **U** максимально розріджені.

Оптимальному упорядкуванню рядків і стовпців розрідженої матриці при розв'язанні СЛАР присвячена велика кількість робіт [6, 7, 17 – 19, 22, 24, 27, 31 – 32], адже проблема ця досить стара. Сьогодні існує ряд алгоритмів квазіоптимального упорядкування [6,13,25,31], адже оптимальний розв'язок задачі практично неможливо отримати для систем великої розмірності. Обчислювальна складність алгоритму отримання глобального оптимального розв'язання становить *N!* (задача повного перебору) і практичне застосування його можливе тільки для систем розмірністю $N<10$ [25]. Результат розв'язання задачі оптимального упорядкування залежить (як і розв'язання будь-якої задачі прийняття оптимальних рішень) від критерію, за яким упорядковується матриця. Якщо говорити про оптимальне упорядкування з огляду мінімізації заповнень, то це є упорядкування, що приводить до найменш можливої кількості заповнень при **LU**-розкладі. Упорядкування вважається оптимальним щодо затрат на арифметичні операції, якщо кількість операцій, що виконуються при обробці переупорядкованих матриць, є мінімальною. З *N!* можливих упорядкувань одне або більше є оптимальним з огляду мінімізації заповнень і декілька упорядкувань (не менше одного) виявляються оптимальними щодо мінімізації кількості арифметичних операцій. Звичайно, можна тільки хотіти знайти такі оптимальні упорядкування, оскільки задача їх пошуку надзвичайно важка і про існування ефективних алгоритмів, придатних для цієї цілі, не відомо нічого [21]. Існуючі процедури є евристичними і, як правило, базуються на спробах пошуку упорядкувань, що забезпечують зменшення заповнень і кількості арифметичних операцій, але не гарантують точного мінімуму цих величин. З іншого боку при оптимальному упорядкуванні не треба забувати про критерій забезпечення точності розв'язку, що передбачає вибір порядку виключення змінних за критерієм мінімізації похибок заокруглень. Сама по собі мінімізація заповнень забезпечує зменшення похибок за рахунок зменшення кількості арифметичних операцій, хоча в ряді випадків порядок вибору провідного елемента за критерієм мінімізації похибок заокруглення конфліктує з критерієм мінімізації заповнень.

Задача оптимального упорядкування рядків і стовпців матриці вузлової провідності в схемотехнічному проектуванні (задача оптимальної перенумерації вузлів схеми) має свою специфіку, обумовлену особливостями об'єкта проектування – інтегральної схеми.

Отже, задача оптимального упорядкування матриці вузлової провідності $\boldsymbol{Y_n}$ (чи задача оптимальної перенумерації вузлів схеми) – це задача вибору оптимального порядку виключення невідомих вузлових напруг (задача вибору провідних елементів матриці, не обов'язково з діагональних). Вибраний оптимальний порядок виключення змінних, що еквівалентне оптимальній нумерації вузлів схеми, повинен забезпечувати максимальну ефективність процедури розв'язання СЛАР. У схемотехнічному проектуванні при забезпеченні ефективності процедури розв'язання СЛАР необхідно йти на розумний компроміс між часом розв'язання та точністю. Власне при моделюванні схем може виявитись, що критерій упорядкування за мінімумом кількості заповнень (що зводить до мінімуму кількість операцій) суперечить результату упорядкування за мінімумом помилок заокруглення. Порядок виключення може суттєво вплинути на точність розв'язання системи рівнянь з погано обумовленими матрицями [14]. Невдалий порядок виключення може привести до появи малих головних елементів в редукованих матрицях, що також впливає на точність розв'язку. Місце розташування таких елементів в матрицях електричних схем, як правило зберігається на багатьох послідовних кроках за часом чи за частотою в процесі чиселового інтегрування нелінійних диференційних рівнянь або обчислення частотних характеристик, оскільки поява цих елементів обумовлена деякими особливостями електричних схем (наявність додатного зворотного зв'язку в схемі, резонансного тощо). Тому в загальному випадку рекомендується вибирати такий критерій упорядкування, при якому об'єм обчислень буде мінімальним як при вибраному порядку виключення, так і при зміні порядку виключення через мале значення якогось з головних елементів.

Існує велика кількість стратегій упорядкування, які відрізняються критеріями вибору головних елементів і як наслідок, різним часом упорядкування. Критерії упорядкування достатньо детально висвітлені у джерелах [3,5,6,14,18,22,27,31,32]. В загальному критерії упорядкування можна розділити на три види:

а) упорядкування за критерієм мінімуму заповнень;

б) упорядкування за критерієм мінімізації похибки заокруглення;

в) упорядкування за векторним критерієм: мінімуму заповнень та похибки заокруглення.

Розглянемо основні характеристики критеріїв мінімізації кількості заповнень [4]:

- довжина пошуку l, тобто кількість рівнянь, що аналізуються для однократного застосування критерію;
- глибина пошуку h, тобто кількість кроків **LU**-розкладу, необхідних для однократного застосування критерію;
- ширина пошуку g, тобто кількість різних порядків розкладу на h кроках, що аналізуються і необхідні для однократного застосування критерію;
- потужність критерію w, тобто кількість рівнянь послідовності розкладу, що визначається при однократному застосуванні критерію.

У більшості випадків $w=1$, $l=N-i$ (i – номер кроку розкладу), $h=1$ і $g=1$.

Наведемо приклади найбільш розповсюджених критеріїв упорядкування, що мінімізують кількість заповнень [8]:

1. Упорядкування в порядку зростання кількості ненульових елементів в рядку початкової матриці **Y**.

2. Упорядкування в порядку зростання кількості заповнень, створених на даному кроці розкладу.

3. Упорядкування в порядку зростання суми ненульових елементів в рядку початкової матриці **Y** та кількості заповнень, що створюються на даному кроці розкладу.

4. Упорядкування в порядку зростання максимально можливої кількості заповнень (в найнесприятливішому випадку), які можуть утворитися на даному кроці розкладу.

Критерій упорядкування застосовується на кожному кроці **LU**-розкладу, в результаті чого визначається номер рівняння (рядок матриці $\boldsymbol{Y}$ чи порядковий номер вузла схеми), обробка якого дає мінімальну кількість заповнень. Застосування критерію на кожному з N кроків розкладу дає послідовність номерів рівнянь (вузлів), при якій при розкладі матриці $\boldsymbol{Y}$ на множники отримуємо кількість заповнень, близьку до мінімально можливої.

На практиці широке розповсюдження отримала реалізація критерію 1, що застосовується в відомому методі Марковіца [27], та критерію 2, що застосовується в найбільш часто використовуваному алгоритмі Беррі [31].

Проведені дослідження ефективності застосування різних критеріїв упорядкування [3, 5, 8] показують, що:

- суттєве зменшення кількості заповнень отримується вже при застосуванні найпростіших критеріїв;

- з врахуванням витрат пам’яті і часу на зберігання програм упорядкування і їх роботу процедури упорядкування слід застосовувати при порядку системи $N>15\div20$;
- при збільшенні N ($N>100$) різниця в ефективності алгоритмів згладжується.

Останній висновок пояснюється [8] необхідністю багатократного застосування критерія в так званих ситуаціях невизначеності, коли критерій задовольняється одночасно для різних номерів рядків (пар “стовпець-рядок”) і, як наслідок, порядок розкладу може вибиратися неоднозначно. Для усунення неоднозначності необхідно збільшувати параметри критерію h і g, що приводить до додаткових витрат часу і ускладнює алгоритм реалізації. На жаль, більшість авторів обмежуються цим поясненням і не аналізують доцільність чи недоцільність збільшення глибини і ширини пошуку в алгоритмах упорядкування для певних ситуацій. Наприклад, специфічній початковій структурі матриці, при якій збільшення параметрів h і g може дати суттєвий виграш в ефективності розв’язання СЛАР, який повністю компенсує витрати на одноразове упорядкування, а при багатократному розв’язанні системи дасть значний виграш.

Щодо критеріїв упорядкування за точністю розв’язання СЛАР [18, 20, 23, 24], то це критерії, що передбачають вибір на кожному кроці розкладу провідного елемента (не обов’язково з діагональних) з найбільшим абсолютним значенням з метою мінімізації похибки заокруглення при діленні на цей елемент на черговому кроці прямого ходу розв’язання системи. Застосування цього критерію в процесі розкладу на ***LU***-множники часто суперечить вимозі мінімуму заповнень на чергових кроках розкладу. Критерії конфліктують між собою, і проектувальник змушений задовольнятися деяким компромісним рішенням.

Парадигма аналізу схем містить алгоритми упорядкування за критерієм мінімуму заповнень, а саме алгоритм Марковіца або алгоритм Беррі. В схемотехнічному проектуванні невикористання алгоритмів упорядкування за точністю має свій резон, що можна пояснити наступними причинами:

1. Введення поряд з критерієм упорядкування за мінімумом заповнень критерію упорядкування за мінімумом похибок заокруглення передбачає упорядкування при кожній зміні значень елементів матриці **Y**. Такий підхід в ітераційних процедурах проектування, коли СЛАР розв’язується тисячі і мільйони разів з суттєвою зміною значень елементів матриці, є завеликою розкішшю. В таких ситуаціях упорядкування необхідно проводити одноразово, тобто за мінімумом кількості заповнень до зміни структури матриці, яка може відбуватися тільки при структурному синтезі.

2. Критерій упорядкування за мінімумом заповнень за своєю суттю вже приводить до зменшення обчислювальної похибки, адже границі похибок зменшуються разом із зменшенням кількості заповнень. Надмірне же заповнення матриці може привести до надто сильного накопичення обчислювальних похибок.

3. Однократне використання процедури оптимального упорядкування дозволяє ускладнювати алгоритм з метою глибшого врахування розрідженості матриці, збільшення витрат на роботу буде компенсовано підвищенням ефективності процесу проектування за рахунок багатократності викликів процедури розв’язання СЛАР.

Отже, шляхи підвищення ефективності процедур етапу схемотехнічного проектування за рахунок поглибленого врахування розрідженості матриць вузлової провідності необхідно шукати в алгоритмах оптимальної перенумерації вузлів схеми за критерієм мінімуму заповнень, що використовуються в сучасних системах АСхП.

Найчастіше використовуваним алгоритмом оптимальної перенумерації вузлів схеми в сучасних програмних системах АСхП є алгоритм Беррі [31]. Особливо добре він себе зарекомендував при упорядкуванні структурно-симетричних матриць. За ствердженням авторів [25] алгоритм Беррі майже у всіх випадках дає оптимальний результат. Цим і пояснюється включення його в парадигму аналізу електронних схем та використання в таких програмних системах схемотехнічного моделювання як “P-Spice” [28], “Micro-PC”[29], “MicroCAP” і інших. Однак дослідження показують, що існує низка схем, для яких даний алгоритм дає далекий від оптимального результат [9,10]. Оптимальність чи неоптимальність розв’язання залежить від поєднання двох факторів: структури електричних з’єднань схеми та її початкової нумерації вузлів. Така ситуація відкриває шлях до вдосконалення алгоритму.

Таке вдосконалення RO-алгоритму було виконано. Детальний опис модифікації цього алгоритму наведено у []. Цей алгоритм використано у системі схемотехнічного моделювання Micro-PC та, показано, як він покращує ефективність проектування [].

4. Висновки

Проведено обгрунтування пошуків шляхів підвищення ефективності процедур схемотехнічного проектування на найнижчому ієрархічному рівні – рівні розрахунку через удосконалення проектних процедур і операцій, що багатократно використовуються в процедурах розрахунку, а отже, ще більше в процедурах аналізу, оптимізації та синтезу. Проаналізовано вплив ефективності процедури розв’язання СЛАР на ефективність процедур розрахунку, аналізу, оптимізації, та синтезу. Визначено, що ефективність методик, які реалізують процедуру розв’язання СЛАР, є визначальною при вирішенні завдань забезпечення ефективності процедур оптимального проектування на найнижчому ієрархічному рівні. Детально проаналізовано методику розріджених матриць, яка є обов’язковим компонентом парадигми аналізу. Визначено основні три проблеми, що вирішує методика розріджених матриць та специфіку розв’язання цих проблем в схемотехнічному моделюванні.. Визначено проблему максимального збереження розрідженості матриці як таку, що визначає ступінь адаптивності методики розріджених матриць до особливостей об’єкта проектування. Проаналізовано методи вирішення цієї проблеми шляхом оптимального упорядкування рядків і стовпців матриці вузлової провідності, яка еквівалентна задачі оптимальної перенумерації вузлів схеми.

1. Анисимов В.И., Дмитревич Г.Д. Автоматизация схемотехнического проектирования на мини-ЭВМ. – Л., 1983. 2. Бармаков Ю.Н., Бахов В.А. и др. Результаты сравнения ряда программ анализа электронных схем. // Известия вузов. Радиоэлектроника. – 1981. – Т.24, №6. – С. 27-37. 3. Баталов Б.В., Егоров Ю.Б., Русаков С.Г. Основы математического моделирования больших интегральных схем на ЭВМ. – М., 1982. 4. Батищев Д.И. Методы оптимального проектирования. – М., 1984. 5. Глориозов Е.Л., Ссорин В.Г., Сыпчук П.П. Введение в автоматизацию схемотехнического проектирования. – М., 1976. 6. Дафф И. Обзор исследований по разреженным матрицам // ТИИЭР. – 1985. – Т.65, №4. 7. Джорж А., Лю Дж. Численное решение больших разреженных систем уравнений. Пер. с англ. –М., 1984. 8. Ильин В.Н., Коган В.Л. Разработка и применение программ автоматизации *схемотехнического проектирования. – М., 1984. 9. Казимира І.Я. Квазіоптимальний алгоритм перенумерації вузлів схеми. “Досвід розробки та застосування приладо-технологічних САПР мікроелектроніки”. Тези доповідей 3-ої Міжнар. наук.-техн. конф. Ч.2. – Львів, 1995. – С.184. 10. Казимира І.Я. Підвищення ефективності алгоритмів схемотехнічного проектування радіоелектронних засобів. Вісник ДУ”ЛП”. № 307. – Львів, 1996. – С.34-39. 11. Калахан Д. Методы машинного расчета электронных схем: Пер. с англ. – М., 1970.12. Казимира І.Я. Методи та засоби забезпечення ефективності процедур оптимального проектування ІС на схемотехнічному етапі.. – Дисертація ... канд. техн. наук за спец. 05.13.12 “Системи автоматизації проектування”. ДУ “Львівська політехніка”, Львів, 1998. 13. C.Britton Rorabaugh. Circuit Design & Analysis. – McGraw-Hill, 1992. 14. Бененсон З.М., Елистратов М.Р. Моделирование и оптимизация на ЭВМ радиоэлектронных устройств. – М., 1981.[15. Пападимитриу Х., Стайглиц К. Комбинаторная оптимизация. Алгоритмы и сложность. – М., 1985. 16. Петренко А.И. Основы автоматизации проектирования. – К., 1982. 17. Петренко А.И., Власов А.И. Алгоритм упорядочивания больших разреженных систем линейных алгебраических уравнений. – Сб. “Автоматизация проектирования в электронике”, вып.12. – К., 1975. 18. Петренко А.И., Власов А.И., Тимченко А.П. Табличные методы моделирования электронных схем на ЭЦВМ. – К., 1977. 19. Писсанецки С. Технология разреженных матриц. – М., 1988. 20. Сигорский В.П., Петренко А.И. Алгоритмы анализа электронных схем. – М., 1976. 21. Shannon R.E. Systems Simulation: the Art and Science. – Prentice-Hall Inc., Englewood Cliffs, N.J., 1975. 22. Тиннэй У.Ф., Уолкер Дж. Прямые решения квазиоптимальных уравнений цепей оптимально упорядоченным разложением на треугольные сомножители // ТИИЭР. – 1967. – Т.55, №11. – С.31-41. 23. Форсайт Дж., Малькольм М., Моулер К. Машинные методы математических вычислений. – М., 1980. 24. Форсайт Дж., Моулер К. Численное решение систем линейных алгебраических уравнений. Пер. с англ. –М., 1969. 25. Чуа Л.О., Пен-Мин Лин. Машинный анализ электронных схем: Алгоритмы и вычислительные методы. Пер. с англ. – М., 1980. 26. Director S.W. Circuit Theory. A Computational Approach. – New York, 1975. 27. H.M.Markowitz. The Elimination Form of the Inverse and Its Application to Linear Programming. // Management Sci. –Vol.3,No.3. –1957. – P. 255-269. 28. Hayt W.H., Jr., Kemmerly J.E. Engineering Circuit Analysis. – New York, 1993. 29. Kazymyra I.Y., Blyzniuk M.B., Lobur M.V. Circuit Simulation Program for Training Specialists in the Field of Circuit Design Software Development. // Proc. of the 4th Int. Workshop “Mixed Design of Integrated Circuits and Systems”. – Poznan, Poland, 1997. – P.673-678. 30. Koval V.A., Blyzniuk M.B., Kazymyra I.Y. Acceleration of Circuit Design: One of the Approaches. // Proc. of the 3rd Advanced Training Course “Mixed Design of Integrated Circuits and Systems” – Lodz, Poland, 1996. – P.113-118. 31. R.D.Berry. An Optimal Ordering of Electronic Circuit Equations for Sparse Matrix Solution.// IEEE, Trans.1971, VCT-18, № 1, P. 40-50. 32. Sato D., Tinney W.F. Technique for exploiting the sparsity of the network admittance matrix // IEEE Trans.– 1963, PAS-82, № 69, P. 944-949.*