

# УДОСКОНАЛЕННЯ МЕТОДУ ПОШУКУ РІШЕНЬ ДЛЯ РОЗВ'ЯЗКУ ЗАДАЧІ ОПТИМІЗАЦІЇ З ВИКОРИСТАННЯМ ГЕНЕТИЧНОГО АЛГОРИТМУ ШЛЯХОМ ПОПЕРЕДНЬОЇ КЛАСТЕРИЗАЦІЇ

К.т.н. Н.І. Фурманова<sup>1</sup>, к.т.н. О.Ю. Фарафонов<sup>1</sup>, к.т.н. О.Ю. Малий<sup>1</sup>, Ю.О. Сіциліцин<sup>2</sup>

1. Запорізький національний технічний університет

2. Таврійський державний агротехнологічний університет, м.Мелітополь

*В статті запропоновано метод попередньої кластеризації з використанням самоорганізаційних карт Кохонена для генетичного алгоритму пошуку рішень. Проведено дослідження запропонованих алгоритмів на прикладі визначення розмірів для елементів топології подільника потужності.*

*В статье предложен метод предварительной кластеризации с использованием самоорганизационных карт Кохонена для генетического алгоритма поиска решений. Проведено исследование предложенных алгоритмов на примере определения размеров элементов топологии делителя мощности.*

*The article suggests a method for preliminary clustering using self-organizing maps for a genetic algorithm of the solution finding. The proposed algorithms are investigated using the example of determining the dimensions of the topology elements of the power divider.*

**Ключові слова:** генетичний алгоритм, розпаралелювання, оптимізація, самоорганізаційні карти Кохонена

## Вступ

Значним недоліком використання генетичних алгоритмів для пошуку рішень є значні вимоги до апаратного забезпечення та витрати часу. Крім того, особливістю генетичних алгоритмів, що інколи приводить до незадовільних результатів, є скочування до локальних мінімумів. Використання попередньої кластеризації дозволить розв'язати ці проблеми.

Кластеризація може бути реалізована різними способами [1-3]. Проте, на нашу думку, одним із найбільш перспективних є використання самоорганізаційних карт Кохонена для виконання кластеризації.

## Особливості кластеризації з використанням самоорганізаційних карт Кохонена

Самоорганізаційна карта Кохонена (СОКК) складається із компонентів, що називаються вузлами або нейронами, кількість яких визначається користувачем. Кожен із вузлів описується двома векторами. Перший – вектор ваги  $m$ , що має таку ж розмірність, що і вхідні дані. Другий – вектор  $r$ , що представляє собою координати вузла на карті. Заздалегідь відома розмірність вхідних даних, за нею деяким чином будуватиметься початковий варіант карти. В процесі навчання вектори ваги вузлів наближуються до вихідних даних. Для кожного спостереження обирається найбільш подібний по вектору ваги вузол, і значення його вектора ваги

наближається до спостереження. Також до спостереження наближаються вектори ваги декількох вузлів, розташованих поряд. Таким чином, якщо в множині вихідних даних два спостереження були схожими, на карті їм будуть відповідати близькі вузли. Циклічний процес навчання, що перебирає вхідні дані, завершується при досягненні картою допустимої раніше заданої користувачем похибки, або при здійсненні заданої кількості ітерацій.

Після цього може бути запущений механізм реалізації пошуку мінімуму з використанням генетичного алгоритму.

Параметри карт Кохонена включають в себе розмірність масиву нейронів, кількість нейронів в кожному вимірі, закон стиснення околу та швидкість навчання.

Кількість нейронів в реальних нейронних мережах, що реалізовані на базі відомих програмних пакетів, зазвичай не перевищує декілька сотень, а частіше складає десятки. В розробленій системі ця величина рівна  $5n$ , де  $n$  – кількість геометричних розмірів, що будуть задіяні в оптимізації.

Швидкість навчання – управляючий параметр деяких алгоритмів навчання, який контролює величину кроку при ітераційній корекції ваг.

Параметрами, що визначають швидкість навчання карти Кохонена, є:

- форма комірки;
- епоха, по досягненню якої навчання буде завершено;
- спосіб початкової ініціалізації;
- тип функції сусідства;
- кількість кластерів.

Загалом комірки можуть бути прямокутними або шестикутними. Шестикутні комірки більш коректно відображають відстань між об'єктами на карті, оскільки для таких комірок відстань між центрами суміжних комірок однакова (рис. 1).

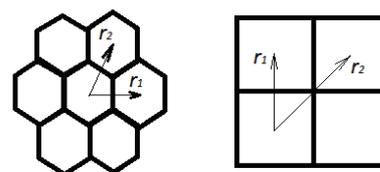


Рис. 1. Шестикутні та прямокутні комірки

Тому в модулі для проведення кластеризації розробленого програмного комплексу ми застосовуємо комірки шестикутної форми [4].

Епоха, по досягненню якої навчання буде завершено, залежить від складності задачі, що вирішується. Тому для подальшої оптимізації топологій, для яких існує точні математичні моделі та методики розрахунку (відрізок одиночної МСЛ, відрізок зв'язаних МСЛ, щілина в МСЛ, поворот МСЛ тощо), задається значення епохи, рівне 300. У випадках, коли аналізуються більш складні топології, що містять у своєму складі фрактальну геометрію, відрізок нерегулярної лінії і т.ін., значення епохи задається рівним 500. Користувачем може бути визначено інші значення цього параметра, проте його збільшення призводить до зростання терміну обчислень без значного уточнення отриманих результатів.

Початкова ініціалізація карти може здійснюватись наступними способами:

- ініціалізація початковими значеннями, коли всім вагам надаються малі випадкові величини;
- ініціалізація прикладами, коли в якості початкових значень задаються значення випадково обраних прикладів з навчальної вибірки;
- лінійна ініціалізація власними векторами.

Для прискорення навчання нами обраний останній спосіб ініціалізації, коли ваги ініціюються значеннями векторів, лінійно впорядкованих вздовж лінійного підпростору, що проходить між двома головними власними векторами матриці коваріації вхідних значень навчальної вибірки.

Тип функції сусідства у загальному випадку може бути прямокутним або Гаусового типу. В першому випадку кожний нейрон, що знаходиться в околі нейрона-переможця, адаптується в рівній мірі, проте це може призводити до появи топологічних дефектів в конфігурації карти. Тому в модулі кластеризації нами застосовується сусідство Гаусового типу (рис.2), при цьому функція сусідства має вид:

$$G(j, \vec{x}) = \exp\left(-\frac{d^2(j, i)}{2\sigma^2}\right),$$

де  $\sigma$  - ефективна ширина топологічного околу, визначає рівень сусідства, тобто рівень, до якого нейрони із топологічного околу нейрона-переможця приймають участь в процесі навчання;

$d(j, i)$  - відстань між нейронами. Для двовимірної решітки (відповідно до обраної шестикутної форми комірок):

$$d^2(j, i) = \|\vec{r}_j - \vec{r}_i\|^2,$$

де  $\vec{r}_j$  визначає позицію нейрону  $j$ , що збуджується,  
 $\vec{r}_i$  - нейрона-переможця  $i$ .

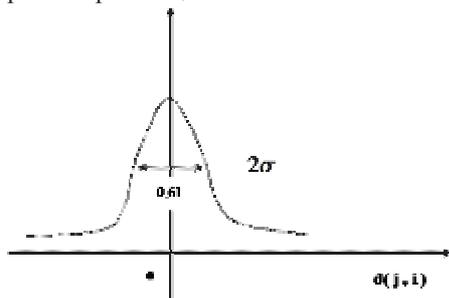


Рис. 2. Сусідство Гаусового типу

На відміну від прямокутного сусідства, рівень адаптації при сусідстві Гаусового типу залежить від значення функції Гауса. Ця функція є опуклою, що призводить до більш швидкого топологічного впорядкування, оскільки вона не містить метастабільних станів. Тому Гаусове сусідство дає кращі результати навчання.

Кластери – замкнені області на СОКК, які включають нейрони з однаковими значеннями виходів. Кількість кластерів визначає максимальна кількість груп, на які система може розділити вхідні дані. Значення цього параметру залежить від раніше визначеної кількості нейронів та повторюваності їх значень, тому визначається системою автоматично.

Властивість топологічної упорядкованості досягається в алгоритмі Кохонена з допомогою додаткового використання поняття околу. Окіл – це декілька нейронів, що оточують нейрон-переможця. Подібно до швидкості навчання, розмір околу спадає з часом так, що спочатку до нього належить досить велика кількість нейронів (можливо, майже вся топологічна карта); на останніх етапах окіл стає нульовим, тобто складається лише з самого нейрона-переможця. Розрахунки показали, що оптимальним є початковий розмір околу в межах від 1/2 до 2/3 розміру карти. Зменшення розміру околу здійснюється за обраним законом – за експоненційно спадною залежністю. Під час навчання модифікуються всі ваги, пов'язані з нейроном-переможцем і його сусідніми елементами.

Застосування самоорганізаційних карт Кохонена для попередньої кластеризації з використанням наведених рекомендацій перед запуском генетичного алгоритму пошуку рішень дозволить уникнути скочування результатів розрахунків до локальних мінімумів, прискорити розрахунки, отримати більш точні результати та застосувати розпаралелювання при проведенні обчислень.

#### Застосування запропонованого методу для мікросмужкових пристроїв

Мікросмужкова топологія використовується для створення пристроїв, призначених для функціонування в надвисокочастотному діапазоні [5, 6]. Широке застосування у мікросмужковій техніці знайшли спрямовані кільцеві подільники потужності з різними коефіцієнтами поділу. Їх застосовують для розгалуження і підсумовування сигналів в колах живлення фазованих антенних решіток, потужних підсилювачах на транзисторах, в змішувачах, перемикачах тощо. Часто необхідно розподілити вхідну потужність на кількість виходів більше двох. Розглянемо синтез і аналіз чотирьох каналного подільника потужності (рис. 3).

В якості вихідних даних для розрахунку геометричних параметрів топології подільника потужності виступають середня частота діапазону, ширина смуги пропускання, коефіцієнт поділу потужності, хвильові опори вхідного та вихідного плечей подільника, поверхневий опір тонкоплівкового резистора. В загальному випадку подільники потужності мають задовольняти вимогам:

- поділ сигналу в заданому співвідношенні;
- узгодження всіх плечей;
- розв'язка вихідних плечей;
- широкосмуговість.

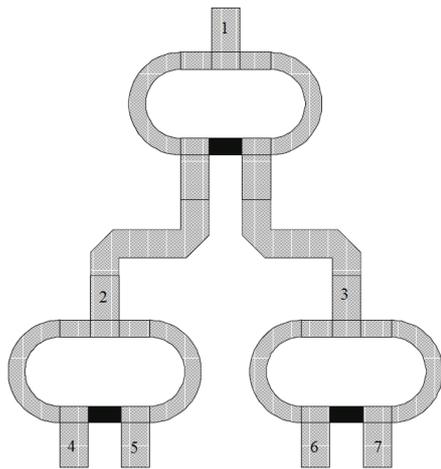


Рис. 3. Чотирьохканальний подільник потужності

Розглянемо можливість розпаралелювання розрахунків чотирьохканального подільника потужності з досягненням мінімального коефіцієнта затухання. Для цього виконаємо наступні кроки:

Крок 1. Визначимо вектор вхідних параметрів, за якими нейронна мережа буде групувати значення: вихідна потужність  $i$ -го каналу ( $P_i$ ), опір розв'язуючого резистора  $R$ , ширина друкованих провідників і сумарна довжина кожного напівкільця. Звідси кількість вимірювань  $n=4$ , а відповідно, і ширина вектору нейронної мережі Кохонена буде такою ж.

Крок 2. Розрахуємо масив вихідних векторів в смузі частот.

Крок 3. За допомогою алгоритму навчання СОКК згрупуємо в кластери значення розрахункових параметрів.

Існує можливість розпаралелювання і під час навчання нейронної мережі Кохонена:

#### 1. Паралелізація фази навчання.

При розв'язанні задачі з допомогою нейронної мережі звичайно проводиться велика кількість експериментів з метою встановлення необхідних параметрів, таких як параметр швидкості навчання і кількість нейронів в різних шарах. З використанням паралелізації фази навчання в певній мірі різних конфігурацій можуть бути досліджені одночасно. Єдина відмінність між двома фазами навчання буде полягати в значенні параметру швидкості навчання.

При використанні паралелізації фази навчання лінійний приріст швидкодії легко досягається, оскільки зникає необхідність міжпроцесорної взаємодії.

#### 2. Паралелізація навчальної вибірки

Часто розмір навчальної вибірки стосовно задачі, що розв'язується з допомогою нейронних мереж, може бути достатньо великим. В однопоточній системі навчальні вектори будуть направлені в мережу по одному. В паралельній системі ці навчальні вектори можна розділити між процесорами. Тоді кожному процесору необхідна повна копія нейронної мережі. Таким чином, процесори можуть навчатися одночасно на різних навчальних вибірках.

#### 3. Паралелізація на рівні шару.

В таких моделях інтелектуальних нейронних мереж як мережа зі зворотним розповсюдженням помилки та неоконітрон, навчальні вектори проходять

через мережу за принципом конвеєра. Таким чином, декілька навчальних векторів можуть знаходитися в мережі одночасно. На відміну від вищеперерахованих, СОКК не є багатошаровою моделлю, крім випадку, коли розповсюдження навчальних векторів представлено у виді шару. Оскільки вхідний шар не виконує ніяких обчислень, то при розподіленні розрахункових ресурсів вхідного шару не можна досягти жодного приросту продуктивності.

#### 4. Паралелізація на рівні нейрона.

Паралелізація на рівні нейрона є найбільш очевидною з присутніх в моделі нейронної мережі, оскільки нейрон як оброблювальний елемент нейронної мережі аналогічний окремому процесору. Паралелізм на рівні нейрона полягає у розподіленні нейронів (в середині шару, якщо використовується модель з декількома шарами) серед процесорів, і наступному паралельному обчисленні. Один із нейронів зіставляється з кожним процесором. Паралелізація на рівні нейрона присутня у всіх моделях нейронних мереж, і це найбільш популярний метод у більшості паралельних реалізацій, незалежно від моделі нейронної мережі, що використовується.

#### 5. Паралелізація на рівні ваг.

Обчислення в межах нейрона також можуть бути розподілені між декількома процесорами. Це дуже мілкий модульний паралелізм і він в першу чергу представлений в апаратних реалізаціях.

Для розрахунку параметрів з використанням нейронної мережі Кохонена будемо використовувати паралелізацію на рівні нейрона.

Алгоритм розрахунків буде мати наступний вид (рис. 4).



Рис. 4. Блок-схема алгоритму розрахунку

Крок 4. Для кожного із отриманих кластерів в паралельному режимі проведемо аналіз з використанням генетичного алгоритму (рис. 5).

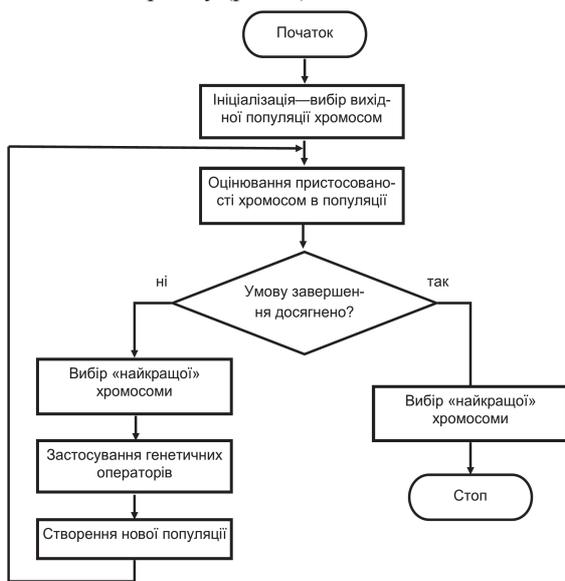


Рис. 5. Блок-схема аналізу з використанням генетичного алгоритму

Обчислення для кожного із кластерів проводимо в паралельному режимі.

Для розрахунку були використані наступні значення:

- середня частота діапазону: 1-10 ГГц;
- смуга частот відносно середньої 10-45%;
- довжина напівкільця  $l$ , мм;
- довжина резистора.

Для отримання оптимального значення розв'язки необхідно дослідити зміну довжини хвилі і довжини резистора в діапазоні частот і зміні смуги пропускання.

Крок для зміни довжини на півкільця  $\pm 0,01$  мм до  $\pm 0,05$  мм від розрахункового значення, крок зміни частот у вказаному діапазоні  $+0,1$  ГГц; крок зміни частотної смуги  $+1\%$ .

Звідси кількість можливих комбінацій складе  $2^{141}$ .

В ручному режимі перебір з використанням існуючих методик займе неприпустимо великий час, тому обираємо усереднені значення і отримуємо:

- середня частота діапазону: 5 ГГц;
- смуга частот відносно середньої: 35%;
- довжина напівкільця  $l = 14,99$  мм;
- довжина резистору  $lr=1,5$  мм.

Як видно, отримати потрібні значення по частоті буває досить складно і необхідне коригування. Використаємо запропоновану схему пошуку рішень за допомогою генетичного алгоритму та СОКК.

Створимо масив вихідних даних вхідних векторів значень, запропонованих для розрахунку. Застосуємо до нього СОКК з паралелізацією фази навчання. До отриманих кластерів застосуємо генетичний алгоритм пошуку рішень. Отримаємо наступні значення.

СОКК сформувала три кластера даних. Застосування до них генетичного алгоритму дозволяє отримати наступні характеристики чотирьох каналного подільника.

Кластер 1:

- середня частота діапазону: 3,4 ГГц;

– смуга частот відносно середньої: 33%;

– довжина на півкільця  $l = 22,50$  мм;

– довжина резистору  $lr=1,0$  мм.

Кластер 2:

– середня частота діапазону: 4,4 ГГц;

– смуга частот відносно середньої: 37%;

– довжина на півкільця  $l = 11,75$  мм;

– довжина резистору  $lr=1,5$  см.

Кластер 3 (оптимальні значення):

– Середня частота діапазону: 5,2 ГГц;

– Смуга частот відносно середньої: 41%;

– Довжина на півкільця  $l = 14,90$  мм;

– Довжина резистору  $lr=1,3$  мм.

Повний час розрахунку склав 1 год 45 хв, з них 1 год на додавання матриці вхідних значень в MATLAB R2011a, 20 хв на навчання СОКК, 10 хв на додавання отриманих даних для розрахунку за допомогою генетичного алгоритму пошуку рішень, 15 хв – безпосередньо розрахунок.

## Висновки

В статті запропоновано метод попередньої кластеризації для оптимізації пошуку рішень із застосуванням самоорганізаційних карт Кохонена. Наведено алгоритми аналізу, проведено перевірку запропонованого методу на розрахунок мікросмужкового подільника потужності.

Отримані результати свідчать про те, що такий розрахунок є переважним, оскільки дає необхідні значення без коригування, при цьому відмічаються значні скорочення витрат часу. Як висновок треба відмітити, що ефективність паралельних обчислень буде залежати від кількості каналів подільника або інших елементів топології мікросмужкового пристрою.

## СПИСОК ЛІТЕРАТУРИ:

1. Гитис, Л.Х. Статистическая классификация и кластерный анализ / Л.Х. Гитис. – М.: Издательство Московского государственного горного университета, 2003. – 157 с.
2. Macia, L. Using Clustering as a Tool: Mixed Methods in Qualitative Data Analysis. / L. Macia // *The Qualitative Report*, 20(7), 2015. – PP. 1083-1094.
3. Fogel, D. B Fuzzy Clustering and Classification, in *Fundamentals of Computational Intelligence* / D.B. Fogel, D. Liu, J.M. Keller // *Neural Networks, Fuzzy Systems, and Evolutionary Computation*. - John Wiley & Sons, Inc., Hoboken, NJ, USA, 2016. – PP. 140-182
4. Фурманова, Н.І. Удосконалення методу пошуку рішень для розв'язку задачі оптимізації з використанням генетичного алгоритму шляхом попередньої кластеризації / Н.І. Фурманова, Ю.О. Сицилицин // *Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій: Тези доповідей VII Міжнародної науково-практичної конференції (17–19 вересня 2014 р., м. Запоріжжя)*. – Запоріжжя: ЗНТУ, 2014. – с. 315-316.
5. Сицилицин, Ю.А. Параллельные вычисления при квазистатическом анализе устройств на микрополосковых линиях / Ю.А. Сицилицин, М.В. Мищенко, Н.И. Фурманова, А.Ю. Фарафонов // *Materials of the IV International Scientific Conference "Information-Management Systems and Technologies"*, Odessa. - 2015. - p. 124-126
6. Sitsilitsin, Yu. Using of parallel computing for the quasi-static analysis of microstrip filters topology / Yu. Sitsilitsin, M. Mishchenko, N. Furmanova, O. Farafonov // *Modern problems of radio engineering, telecommunications and computer science. Proceedings of the International Conference TCSET'2014*. – Lviv-Slavske, 2014. – P. 65.