

## ЗАСТОСУВАННЯ СТРАТЕГІЙ РОЙОВОГО ІНТЕЛЕКТУ ПРИ ПРОЕКТУВАННІ ПАРАЛЕЛЬНИХ НЕЙРОЕВОЛЮЦІЙНИХ МЕТОДІВ СИНТЕЗУ НЕЙРОМОДЕЛЕЙ

Леощенко С. Д., Олійник А. О., Субботін С. О., Зайко Т. А.

*У роботі запропоновано способи застосування стратегій ройового інтелекту для паралелізації нейроеволюційних методів синтезу штучних нейронних мереж. Запропоновані підходи дозволять вирішити ряд проблем, які зазвичай виникають при проектуванні високопродуктивних обчислень, пов'язаних з синтезом нейронних мереж. Об'єктом дослідження є процес розроблення паралельного підходу для нейроеволюційного синтезу штучних нейронних мереж, а саме використання стратегій ройового інтелекту для вирішення низки проблем при проектуванні такого методу, що використовував би ресурси паралельної комп'ютерної системи.*

*Одними з найбільш проблемних місць є високоітеративна природа та значний час роботи нейроеволюційних методів. Одним із способів вирішення цих проблем є використання паралельних комп'ютерних систем та розподілених обрахунків. Проте ряд питань виникає при проектуванні паралельного нейроеволюційного методу.*

*В ході дослідження було вирішено ряд завдань, які включали аналіз та дослідження нейроеволюційних методів синтезу штучних нейронних мереж та проблем їх паралелізації. Також увагу приділено методам ройового інтелекту, що набули популярності останнім часом та демонструють гарні результати роботи.*

*Новий метод, що розроблявся під час роботи, базувався на стратегіях організації роботи із ройовими частинками. Так, розподілені між ядрами підпопуляції та окремі індивіди аналізувалися в якості окремих частинок, які взаємодіють між один одним та залежать від локального середовища. Класичні генетичні оператори були модифіковані критеріальними механізмами для підвищення адаптивності.*

*Під час експериментів розроблюваний метод порівнювався із класичними методами. Під час роботи особливу увагу було приділено не тільки характеристикам отриманих результуючих нейромоделей, але й навантаженню на процесор під час роботи. Розроблюваний метод продемонстрував прийнятні результати за всіма порівняннями. Новий підхід допоміг значно підвищити рівень якості роботи паралельного нейроеволюційного методу синтезу, дозволяючи рівномірно використовувати можливості обчислювальних вузлів паралельної системи.*

**Ключові слова:** *нейроеволюційний метод, генетичний алгоритм, ройовий інтелект, паралельна система, високопродуктивні обчислення.*

## 1. Вступ

Використання методів машинного навчання та апарату штучних нейронних мереж (ШНМ) полегшує, автоматизує та підвищує точність під час обробки багатьох типів даних. Моделі, що базуються на ШНМ, використовуються під час діагностування та прогнозування в техніці, виробництві, медицині, економіці, тощо. Особливо помітно зміни при обробці великих масивів даних, коли, наприклад, маємо складну технічну систему та множину сенсорів, встановлених на різних етапах. Під час тестування системи кожен сенсор щосекунди замірює певну кількість значень. Іноді такі виміри можуть вимірюватися гігабайтами інформації в секунду. Звісно, що для точної обробки такого об'єму необхідно використовувати моделі з високою точністю, які дозволять відстежити зв'язок незалежних параметрів, їх вплив на категоріальні (залежні) змінні та, наприклад, побудувати прогноз.

Проте важливим питанням є саме синтез оптимальної моделі. Синтез моделі на базі ШНМ складається з двох основних етапів: початкове визначення топології (архітектури або структури ШНМ) та навчання (визначення вагів зв'язків, нейронів і т. п.). Для етапу навчання широкого розповсюдження та популярності набув метод зворотного поширення помилки. Проте цей метод відрізняється низкою недоліків [1]:

- невизначено довгий процес навчання – складно спрогнозувати, який час займе саме навчання, адже багато залежить від вхідних даних, визначеної архітектури мережі, тощо;

- проблема перенавчання мережі – є певні ризики, пов'язані із локальними екстремумами;

- надмірна залежність від експерта – досить багато вільних параметрів налаштовується експертом перед початком навчання, тож великі ризики пов'язані саме із роллю експерта.

В решті решт, метод зворотного поширення помилки застосовується тільки для етапу навчання вже розробленої топології ШНМ. Альтернативним підходом завжди були еволюційні методи [1], які в контексті синтезу ШНМ позначаються, як нейроеволюційні методи.

## 2. Об'єкт дослідження та його технологічний аудит

*Об'єкт дослідження* – процес проектування паралельного методу нейроеволюційного синтезу ШНМ, а саме використання стратегій ройового інтелекту для вирішення низки проблем при проектуванні методу, що використовував би ресурси паралельної комп'ютерної системи.

Більшість нейроеволюційних методів має низку переваг в порівнянні із, наприклад, градієнтними методами [1–3]. Проте одними із головних недоліків таких методів є їх високоітеративна природа та значний час роботи [4, 5]. Ці проблеми зазвичай пояснюються стохастичною сутністю методів. Одним із способів вирішення цих проблем є використання паралельних комп'ютерних систем та розподілених обрахунків.

Ряд робіт [6–8] визначає типові проблеми, що виникають при проектуванні паралельного синтезу ШНМ, заснованому на еволюційних стратегіях. Зазвичай

такі проблеми пов'язані або із архітектурними особливостями системи, або із певними складнощами визначення вільних параметрів методів.

В першому випадку проблеми можуть виникати через досить ресурсоемні операції обміну інформацією між потоками, що значно підвищує накладні витрати та може загальмувати загальне виконання роботи. Іншою розповсюдженою проблемою є недосить прораховане розподілення важких обчислень між легковаговими потоками систем, що базуються на графічних процесорах (GPU) [6, 8].

В другому ж випадку проблеми зазвичай пов'язані із визначенням об'єму інформації, що розподіляється між потоками системи, а потім надходить до головного потоку. Так, досить великі об'єми інформації можуть значно знизити швидкість роботи. Тому, наприклад, можливо розподілити певні етапи, пов'язані з ініціалізацією безпосередньо між потоками [6, 7].

Оскільки на сьогоднішній день паралельні комп'ютерні системи стають все більш розповсюдженими задача осмисленого використання потенціалу таких систем набуває особливої важливості.

### **3. Мета та задачі дослідження**

*Мета дослідження* – аналіз особливостей стратегій та процедур ройового інтелекту для подальшого їх використання при проектуванні паралельних нейроеволюційних методів. Це повинно забезпечити раціональне використання потенціалу паралельних комп'ютерних систем на етапі синтезу ШНМ.

Для досягнення поставленої мети необхідно виконати такі задачі:

1. Дослідити стратегії та процедури методів ройового інтелекту.
2. Використовуючи результати дослідження спроектувати паралельний нейроеволюційний метод синтезу ШНМ.
3. На основі аналізу експериментальних результатів роботи розробленого методу розробити подальші рекомендації та інструкції.

### **4. Дослідження існуючих рішень проблеми**

#### **4.1. Нейроеволюційний синтез штучних нейронних мереж**

Нейроеволюція – форма машинного навчання, яка використовує еволюційні алгоритми для навчання нейромережі [2]. Цей підхід використовується в багатьох галузях: медицина, розробка інтелектуальних систем цивільного та військового призначення, ігри, навчання та управління автоматизованими агентами та роботами. У цих випадках досить просто виміряти продуктивність нейромережі, в той час як реалізувати навчання з вчителем дуже важко або практично неможливо. Цей метод навчання відноситься до категорії методів навчання з підкріпленням.

Існує велика кількість нейроеволюційних алгоритмів, які діляться на дві групи. До першої відносяться алгоритми, які виробляють еволюцію ваг при заданій топології мережі, до іншої – алгоритми, які крім еволюції ваг також виробляють еволюцію топології мережі. Хоча і не існує загальноприйнятих умов для проведення відмінностей, прийнято, що додавання або видалення зв'язків в мережі в ході еволюції називається ускладненням або спрощенням відповідно [2, 3].

Генетичний метод, як один з найбільш поширених і популярних еволюційних методів, набув широкого поширення для синтезу ШНМ [4, 5]. Його застосування дозволяє синтезувати одночасно популяцію нейронних мереж і надалі виробляти точкові зміни (оператор мутації). Після цього популяція оцінюється (відбувається селекція кращих особин популяції) і запускається схрещування. Як правило, використовуються більш прості види схрещування, що гарантують порівняно непогані результати при економному споживанні обчислювальних ресурсів: одноточковий або двоточковий кросовер. Однак для більш тонкого налаштування нового покоління або ж реалізації багатобатьківського схрещування використовується оператор рівномірного схрещування. Нове покоління проходить всі попередні кроки – метод виконується, поки не буде отримана найбільш прийнятна ШНМ.

Як зрозуміло, використання подібних методів в перспективі надає цілий ряд переваг [1, 3]:

- велика різноманітність отриманих топологій дозволяє підібрати найбільш прийнятне рішення. Так, наприклад, для простих завдань, можна встановити жорсткі обмеження для отримання простих ШНМ, а більш складні структури (рекурентні, глибокі, згорткові ШНМ та ін.) будуть отримані для більш складних завдань;

- адаптивність отриманих ШНМ. Тут мається на увазі, що під час синтезу метод може допомогти синтезувати більш точну та структурно прийнятну ШНМ під конкретну задачу та набір вхідних даних;

- універсальність. Подібні методи можуть працювати з різноманітними наборами вхідних даних або різноманітними топологіями ШНМ, не вимагаючи точних даних про структурні особливості;

- можливість використання паралельного підходу. Це особливо важливо в умовах стрімкого розвитку сучасних паралельних обчислювальних систем з використанням безлічі ядер процесорів (CPU) або легковажних потоків графічних процесорів (GPU).

#### **4.2. Проблеми паралелізації генетичних алгоритмів**

Однак використання паралельного підходу для генетичних алгоритмів (ГА) пов'язане з низкою проблемних питань. Розглянемо найбільш типові проблеми [6, 7]:

- вибір або розробка стратегії взаємодії складових частин методу;
- вибір частоти міграцій між під-популяціями;
- визначення мігруючих особин та їх кількості;
- визначення структури еволюції окремих під-популяцій.

Розглянемо проблеми більш детально. Структура паралельної системи є важливим фактором у продуктивності паралельного алгоритму, оскільки вона визначає, наскільки швидко (або наскільки повільно) краще рішення поширюється між популяціями [7]. Якщо система є сильно зв'язаною, то кращі рішення будуть швидко поширюватися в усі потоки та можуть швидко наситити популяцію. З іншого боку, якщо мережа слабо зв'язана, рішення будуть поширюватися повільніше і потоки будуть більш ізольованими один від

одного. Може відбуватися подальший паралельний розвиток і схрещування різних рішень для отримання потенційно кращих рішень.

Загальна тенденція в паралельних генетичних алгоритмах (ПГА) – це використання статичних структур системи, які визначаються до запуску методу і залишаються незмінними [6, 8].

Частота міграцій також має великий вплив на кінцеве рішення. Як відомо, занадто часті міграції призводять до виродження популяцій, а рідкісні, навпаки, до зниження збіжності. Для регулювання частоти міграції застосовуються різні методи, які можна розбити на два типи: адаптивні та подієві. У першому випадку методи адаптації використовуються для налаштування частоти міграції в процесі роботи алгоритму. У другому випадку застосовуються методи, що визначають необхідність міграції, тобто міграція здійснюється тільки при настанні якоїсь події [8].

Для відбору особин для міграції застосовуються механізми селекції. Відомо, що окремі хромосоми можуть містити важливі фрагменти генетичного коду, але дані частини можуть перебувати в хромосомах, які відрізняються поганою пристосованістю. Але в той же час виключення подібних рішень може привести до того, що відбудеться передчасна збіжність, або пропуск глобального оптимуму.

Використання різних стратегій накладає головне обмеження – необхідність формування однотипної структури хромосоми. Але ефект, який можливий при вдалому формуванні, може бути набагато більшим, ніж при використанні однієї структури ПГА у всіх під-популяціях [6, 7].

Також варто відзначити, що велика кількість міграцій і навіть динамічного обміну проміжною інформацією між потоками вимагає додаткових накладних витрат, що іноді в значній мірі нівелює досягнення паралельного виконання обчислень.

### **4.3. Ройовий інтелект. Стратегії та процедури**

Помітно, що ряд проблем пов'язаний з вертикальними зв'язками всередині паралельної системи та операціями пересилання проміжних даних між потоками. Найбільш популярними еволюційними біоінспірованими алгоритмами на сьогоднішній день є, так звані, ройові процедури (PSO) [9]. Алгоритми ройової поведінки запозичені з природи, де групи тварин (зграї, рої) демонструють несподівано гарні результати при тому, що кожна з особин має досить примітивний розум [9, 10]. В останні кілька десятиліть в різних областях науки робляться небезуспішні спроби відтворити ройові механізми поведінки в штучних моделях. Багато моделей при цьому прагнуть відтворити властивість, притаманне природним системам, що самоорганізуються. Система в процесі взаємодії її елементів як ціле набуває ознак, які не були притаманні елементам окремо.

Основну ідею ройових методів можна представити наступним чином. Штучні частинки, що переміщаються в  $n$ -мірному пошуковому просторі, можуть вести себе подібно зграї або рою [10]. Представники рою шукають щось, на кшталт корму – екстремум заданої цільової функції. За аналогією з генетичними обчисленнями можна стверджувати, що рій подібний до популяції, а частинки (індивідууми) відповідають хромосомам. Частинки

переміщуються в пошуковому просторі, змінюючи напрямок відповідно до власного досвіду та досвіду сусідів. Кожна частинка володіє вектором швидкості та вектором позиції. При оптимізації  $N$ -мірної функції такі вектори мають розмірність  $N$ . Як і в біологічному житті, передбачається, що кожна частка регулює свої вектори позиції та швидкості відповідно до власного досвіду (когнітивна складова), а також за інформацією, отриманою від інших членів рою. Тут когнітивний досвід частинки розуміється, як її знання про кращу позицію, в якій вона сама перебувала, а соціальне знання частинки – як знання про кращу позицію, через яку пройшла одна з частинок в одній групі з нею. У задачі оптимізації найкраща позиція частинки розуміється як позиція, в якій мінімізується значення функції.

Порівнюючи методи генетичних алгоритмів (ГА) та PSO можна відзначити ряд подібностей:

- обидва алгоритми є стохастичними;
- рішення шукається на базі популяції індивідумів;
- початкова популяція, найчастіше, генерується випадковим чином;
- для роботи методу потрібен обрахунок фітнес-функції кожного індивідуума;
- області можливого використання методів збігаються.

Але існують і відмінності:

- число вільних параметрів, що на початку вказуються в методах PSO менше;
- класичні еволюційні оператори відсутні, а самі частинки не «вмирають» – не видаляються після роботи.

## **5. Методи досліджень**

### **5.1. Використання принципів ройового інтелекту для паралелізації генетичного алгоритму**

Розглянемо ж, які підходи методів PSO можуть допомогти вдосконалити ПГА. У класичній теорії штучного інтелекту для певної задачі створюється одна інтелектуальна система, що має всі необхідні ресурси для її вирішення. У теорії багатоагентних систем використовується протилежний принцип. Вважається, що один агент має неповне уявлення про глобальну проблему, тому створюють деяку безліч агентів і забезпечують ефективну взаємодію між ними [9, 10]. В рамках «колективного» інтелекту глобальна поведінка всієї системи розглядається, як результат взаємодій ряду простих агентів.

Таким чином, з ройових методів можна почерпнути наступні принципи:

- багатоагентна система – це популяція простих і залежних один від одного агентів;
- кожен агент самостійно визначає свої реакції на події в локальному середовищі та взаємодії з іншими агентами;
- зв'язки між агентами є горизонтальними, тобто не існує агента-супервізора, керуючого взаємодією інших агентів;
- немає точних правил, щоб визначити глобальну поведінку агентів;
- поведінка, властивості та структура на колективному рівні породжуються тільки локальними взаємодіями агентів.

## 5.2. Проектування та конструювання динамічної паралельної структури

Таким чином, можна прийти до проектування та конструювання динамічної паралельної системи. У цьому випадку потік не обмежується зв'язками з певною фіксованою кількістю потоків; замість цього мігранти направляються до потоків, які задовольняють певні критерії. В якості подібного критерію береться ступінь різноманітності популяції або міра генотиповості, тобто відстані між двома популяціями (або відстані від характерної особини популяції, наприклад, улюбленої). При такій структурі необхідні механізми відстеження подій в сусідніх популяціях, причому якщо в одній з сусідніх популяцій деяка подія настала, то слід очікувати події і в другій популяції.

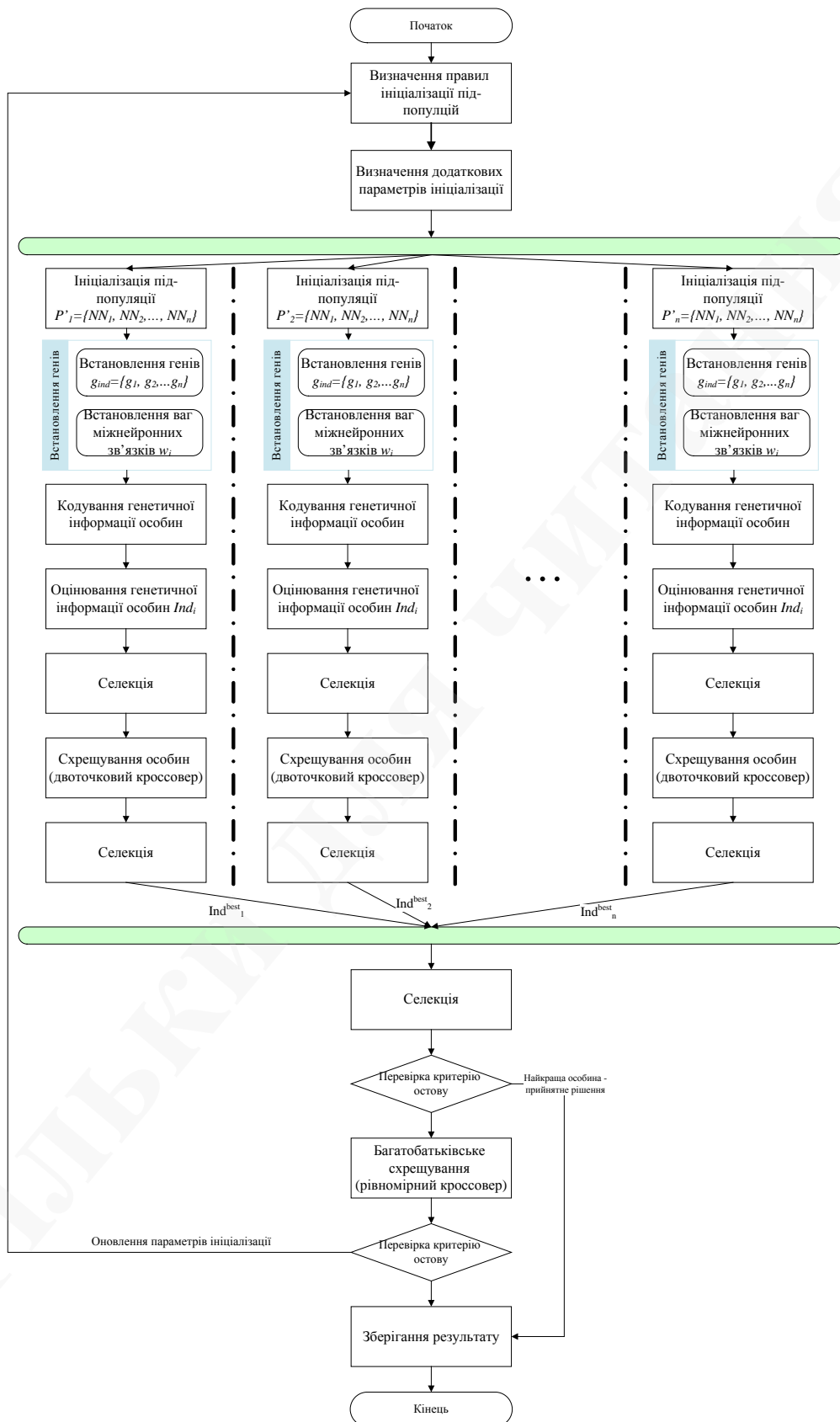
## 6. Результати досліджень

На рис. 1 представлено загальну схему розробленого паралельного нейроеволюційного методу. Зі схеми видно, що як такого головного потоку в системі не передбачається. На початку визначаються лише основні та додаткові параметри майбутніх під-популяцій. Серед таких параметрів можна відзначити: розмір кожної популяції, топологічні особливості, тощо. Ця інформація розподіляється на потоки паралельної системи. Вже на потоках самостійно формуються та кодуються під-популяції для роботи. Основні генетичні оператори залишаються, проте поки процес виконується на окремих потоках слід використовувати найпростіші оператори, наприклад, доточковий кросовер або рангову селекцію. Коли окремо на потоках було отримано найкращі особини вони синхронізуються. Виконується селекція та перевірка найкращої особини з отриманих. Якщо така особина поки не є прийнятною запускається механізм багатобатьківського кросоверу, з визначенням долі генетичної інформації, що буде переходити від кожного з індивідуумів за ранговим принципом. Введення на цьому етапі рангової селекції, з додаванням додаткових критеріїв для оцінки не лише точності, а й топологічних особливостей, та рівномірного схрещування, допустимо. Адже використовується один потік, тому навіть, якщо система буде використовувати для паралелізації GPU, цей етап буде виконуватися на CPU, тому не буде гальмування через складність операцій.

У випадку, якщо із нового покоління краща особина не задовольнить критерії якості, нове покоління не буде видалено, а буде використано для оновлення даних для формування нових під-популяцій.

Таким чином спроектована паралельна структура методу забезпечує:

- прості під-популяції ШНМ, що розподілені між потоками та не залежать один від одного;
- кожний потік агент самостійно визначає свої реакції (типи мутацій, схрещування, тощо) на події в локальному середовищі;
- головний потік використовується лише, щоб спочатку запуснути роботу, а потім синхронізувати результати, а не для управління взаємодією інших потоків;
- визначення кращих рішень на загальному рівні залежить лише від локальних сценаріїв потоків.



**Рис. 1.** Загальна схема розробленого методу

Для експериментального дослідження було обрано набір даних з роботи [11], присвяченій дослідженню індексу пневмонії для визначення хвороби серед пацієнтів. У табл. 1 наведено загальні характеристики вибірки даних.



Таблиця 1

## Загальні характеристики набору даних

Загальна кількість значень	77490	Кількість ознак	54
Характер даних	Числові (після опрацювання)	Кількість екземплярів	1435

У табл. 2 наведено порівняння виконання синтезу ШНМ розробленого методу, зі звичайним ПГА [12] та паралельною реалізацією методу NeuroEvolution of Augmenting Topologies (NEAT) [13].

Таблиця 2

## Порівняння процесу синтезу з використанням різних методів

Метод	Час синтезу, с	Середня помилка під час синтезу	Середня помилка під час тестування
PSO+ПГА	3198.86	0.0000	0.0019
ПГА	4017.92	0.1402	0.0847
Parallel NEAT	3269.30	0.0136	0.0214

Графік на рис. 2 демонструє завантаженість CPU під час роботи.

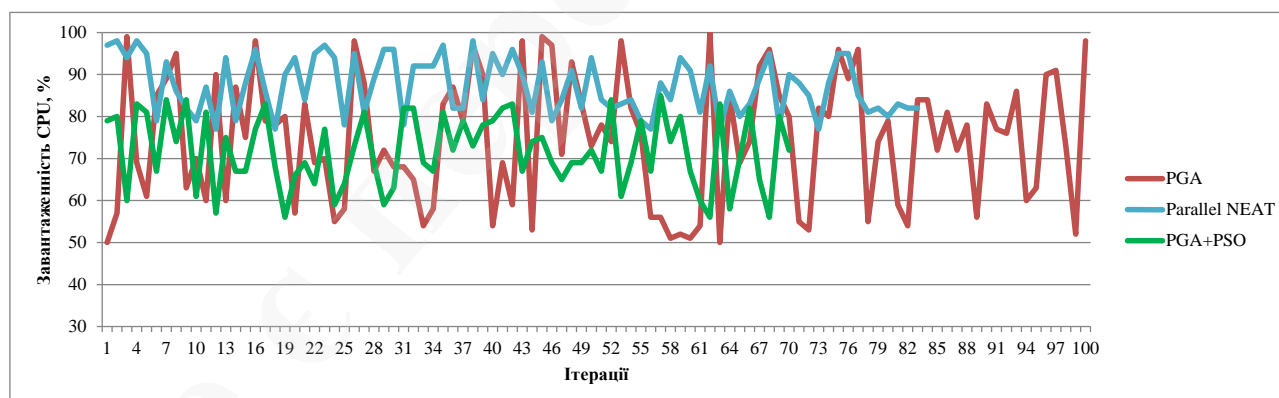


Рис. 2. Навантаження на CPU під час роботи

Як видно з графіку (рис. 2) під час виконання простого ПГА розрив між навантаженням значний – це пояснюється частим обміном даними між ядрами, що значно гальмує процес роботи та перевантажує систему. Parallel NEAT більш оптимізований, тому розриви не такі значні, проте завантаженість все ж сягає 98 %. Пікова завантаженість під час виконання запропонованого методу не перевищувала 85 %, більш того через незначну долю пересилок даних не спостерігається таких значних розривів пікових значень.

## 7. SWOT-аналіз результатів досліджень

*Strengths.* Запропонований підхід базується на особливому класі еволюційних методів. Після дослідження були запозичені стратегії, що змогли

допомогти в розробці та проектуванні нового методу, значно підвищили рівень якості роботи паралельного нейроеволюційного методу синтезу ШНМ.

*Weaknesses.* До певних слабких сторін слід віднести все ж таки певну неспроможність повністю відмовитися від генетичних операторів, що певною мірою обмежує роботу запропонованого методу.

*Opportunities.* До нових можливостей слід віднести значно розширені рамки застосування паралельних підходів до синтезу ШНМ. Новий метод може бути реалізовано для різних типів паралельних систем без суттєвих втрат в якості роботи.

*Threats.* Певні ризики присутні під час модифікування саме генетичних операторів методу. Важко спрогнозувати як на якість роботи може вплинути, наприклад, ускладнення мутацій введенням критеріального вибору типу мутації.

## **8. Висновки**

1. Досліджено стратегії та процедури методів ройового інтелекту. Під час дослідження були виокремлені певні особливості, які забезпечують високу продуктивність ройових методів. Увагу було сфокусовано саме на тих засобах, які можуть бути використані при проектуванні паралельних підходів до нейроеволюції.

2. Використовуючи результати дослідження було спроектовано паралельний нейроеволюційний метод синтезу ШНМ. Новий метод максимально використовував стратегії запозичені від ройових методів, а саме структура методу, що забезпечує:

– прості під-популяції ШНМ, що розподілені між потоками та не залежать один від одного;

– кожний потік агент самостійно визначає свої реакції (типи мутацій, схрещування, тощо) на події в локальному середовищі;

– головний потік використовується лише, щоб спочатку запустити роботу, а потім синхронізувати результати, а не для управління взаємодією інших потоків;

– визначення кращих рішень на загальному рівні залежить лише від локальних сценаріїв потоків.

3. Під час дослідження експериментальних результатів роботи було проведено порівняння розробленого методу із найближчими альтернативними методами та проаналізовано результати. Порівняння продемонструвало покращені характеристики розроблюваного методу за точністю та використанням ресурсів. Подальші модифікації та інструкції базуються на введенні додаткових критеріальних механізмів для адаптивного налаштування роботи методу.

## **Подяка**

Робота була написана за підтримки держбюджетного науково-дослідного проекту державного бюджету Національного університету «Запорізька політехніка» «Інтелектуальні методи та програмне забезпечення для діагностики та неруйнівного контролю якості військової та цивільної техніки» (номер державної реєстрації 0119U100360).

## **Література**

1. Albrightsen, S. I., Imenes, A., Goodwin, M., Jiao, L., Nunavath, V.; Pimenidis, E., Jayne, C. (Eds.) (2018). Neuroevolution of Actively Controlled Virtual Characters – An

Experiment for an Eight-Legged Character. *Engineering Applications of Neural Networks. EANN 2018. Communications in Computer and Information Science. Vol. 893*. Cham: Springer, 94–105. doi: [http://doi.org/10.1007/978-3-319-98204-5\\_8](http://doi.org/10.1007/978-3-319-98204-5_8)

2. Bohrer, J., Grisci, B., Dorn, M. (2020). *Neuroevolution of Neural Network Architectures Using CoDeepNEAT and Keras Computer Science*. Arxiv. Available at: <https://arxiv.org/abs/2002.04634>

3. Bergel, A. (2020). Neuroevolution. *Agile Artificial Intelligence in Pharo*. Berkeley: Apress, 283–294. doi: [http://doi.org/10.1007/978-1-4842-5384-7\\_14](http://doi.org/10.1007/978-1-4842-5384-7_14)

4. Mason, K., Duggan, J., Howley, E. (2018). Watershed management using neuroevolution. *Modeling Earth Systems and Environment, 4 (4)*, 1445–1448. doi: <https://doi.org/10.1007/s40808-018-0508-z>

5. Arellano, W. R., Silva, P. A., Molina, M. F., Ronquillo, S., Ortega-Zamorano, F.; Rojas, I., Joya, G., Catala, A. (Eds.). (2019). Red-Black Tree Based NeuroEvolution of Augmenting Topologies. *Advances in Computational Intelligence. IWANN 2019. Lecture Notes in Computer Science. Vol. 11507*. Cham: Springer, 678–686. doi: [http://doi.org/10.1007/978-3-030-20518-8\\_56](http://doi.org/10.1007/978-3-030-20518-8_56)

6. Gonçalves, I., Seca, M., Castelli, M.; Banzhaf, W., Goodman, E., Sheneman, L., Trujillo, L., Worzel, B. (Eds.) (2020). Explorations of the Semantic Learning Machine Neuroevolution Algorithm: Dynamic Training Data Use, Ensemble Construction Methods, and Deep Learning Perspectives. *Genetic Programming Theory and Practice XVII. Genetic and Evolutionary Computation*. Cham: Springer, 39–62. doi: [http://doi.org/10.1007/978-3-030-39958-0\\_3](http://doi.org/10.1007/978-3-030-39958-0_3)

7. Hoseini Alinodehi, S. P., Moshfe, S., Saber Zaeimian, M., Khoei, A., Hadidi, K. (2016). High-Speed General Purpose Genetic Algorithm Processor. *IEEE Transactions on Cybernetics, 46 (7)*, 1551–1565. doi: <http://doi.org/10.1109/tcyb.2015.2451595>

8. Hou, N., He, F., Zhou, Y., Chen, Y., Yan, X. (2018). A Parallel Genetic Algorithm With Dispersion Correction for HW/SW Partitioning on Multi-Core CPU and Many-Core GPU. *IEEE Access, 6*, 883–898. doi: <http://doi.org/10.1109/access.2017.2776295>

9. Cleghorn, C. W., Engelbrecht, A. P. (2017). Particle swarm stability: a theoretical extension using the non-stagnate distribution assumption. *Swarm Intelligence, 12 (1)*, 1–22. doi: <http://doi.org/10.1007/s11721-017-0141-x>

10. Nobile, M. S., Cazzaniga, P., Besozzi, D., Colombo, R., Mauri, G., Pasi, G. (2018). Fuzzy Self-Tuning PSO: A settings-free algorithm for global optimization. *Swarm and Evolutionary Computation, 39*, 70–85. doi: <http://doi.org/10.1016/j.swevo.2017.09.001>

11. Kim, M.-A., Park, J. S., Lee, C. W., Choi, W.-I. (2019). Pneumonia severity index in viral community acquired pneumonia in adults. *PLOS ONE, 14 (3)*, e0210102. doi: <http://doi.org/10.1371/journal.pone.0210102>

12. Nugroho, E. D., Wibowo, M. E., Pulungan, R. (2017). Parallel implementation of genetic algorithm for searching optimal parameters of artificial neural networks. *3rd International Conference on Science and Technology – Computer (ICST)*. Yogyakarta, 136–141. doi: <http://doi.org/10.1109/icstc.2017.8011867>

13. *Parallel NEAT*. Python Encyclopedia. Available at: [https://neat-python.readthedocs.io/en/latest/\\_modules/parallel.html](https://neat-python.readthedocs.io/en/latest/_modules/parallel.html)