

**ПОКРАЩЕННЯ ПРОДУКТИВНОСТІ ПОБУДОВИ ОПТИМАЛЬНОЇ ТОПОЛОГІЇ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ З ВИКОРИСТАННЯМ НЕЙРОЕВОЛЮЦІЙНИХ АЛГОРИТМІВ ЗА РАХУНОК ЗАСТОСУВАННЯ МЕТОДУ ПАРАЛЕЛЬНОЇ ОЦІНКИ ПРИДАТНОСТІ / ENHANCING THE EFFICIENCY OF CREATION OF OPTIMAL ARTIFICIAL NEURAL NETWORK TOPOLOGIES WITH NEUROEVOLUTIONARY ALGORITHMS THROUGH PARALLEL FITNESS EVALUATION**

**Омельяненко Я.В. / Omelianenko Ia.V.**

Інститут програмних систем НАН України / Institute of software systems NAS of Ukraine  
03187, м. Київ, просп. Академіка Глушкова, 40,  
e-mail: [yaric.mail@gmail.com](mailto:yaric.mail@gmail.com), тел. (073) 540-22-68

Neuroevolution algorithms, a subset of genetic algorithms family, utilize evolutionary processes inspired by natural evolution to train artificial neural networks. A crucial aspect of neuroevolution is evaluating the fitness scores of each candidate solution at the end of each evolutionary epoch. This often involves running complex simulations of physical processes, which can significantly slow down the training if done sequentially for each candidate. This work proposes a solution that leverages the inherent parallel processing capabilities of the GO programming language to evaluate fitness scores concurrently. The efficiency of this method is tested using two classic reinforcement learning experiments: single and double pole balancing. Experimental results show a significant improvement in training speed when using the proposed parallel evaluation method compared to sequential evaluation.

Нейроеволюційні алгоритми зазвичай використовуються для створення оптимальних контролерів на основі сигналів зворотного зв'язку від середовища, що є основою навчання з підкріпленням. При цьому, важливо зазначити, що деякі варіанти нейроеволюційних алгоритмів, такі як нейроеволюція наростаючих топологій (NEAT) [1, 2], здатні створювати енергоефективні топології ШНМ, які можуть використовуватися в умовах з обмеженими обчислювальними та енергетичними ресурсами. Це робить їх ідеальними для створення систем керування автономних роботизованих платформ або безпілотних літальних апаратів.

Незважаючи на це, використання еволюційних принципів для навчання ШНМ має значний недолік, який суттєво впливає на продуктивність навчального процесу. Кожен організм у популяції потенційних кандидатів повинен бути оцінений за допомогою функції пристосованості, яка визначає його придатність для створення наступного покоління. Варто зазначити, що обчислення цієї функції зазвичай вимагає запуску складної симуляції реального фізичного процесу. Таким чином, якщо оцінка придатності для кожного організму в популяції виконується послідовно наприкінці кожної епохи еволюції, це значно уповільнює загальну швидкість процесу навчання ШНМ. Для вирішення цієї проблеми в даній роботі запропоновано метод паралельної оцінки придатності на прикладі використання вбудованих можливостей мови програмування GO.

**Аналіз існуючих досліджень.** В одному з досліджень [3] автори запропонували використовувати велику мережу комп'ютерів, наданих добровольцями, для масштабної еволюції згорткових нейронних мереж (CNN). Для цього вони реалізували модифіковану версію алгоритму NEAT, названу еволюційне дослідження розширених згорткових топологій (EXACT). Протягом двох місяців близько 4500 комп'ютерів, наданих добровольцями через Citizen Science Grid, змогли навчити приблизно 120000 CNN. Цей підхід виявився креативним і показав хороші результати, але вимагає залучення великої кількості добровільно наданих комп'ютерів.

Крім того, нещодавно було опубліковано кілька робіт, які досліджують використання комерційних обчислювальних хмар для розпаралелювання навчання алгоритму NEAT. Одне з перспективних рішень запропонували автори бібліотеки EvoJAX [4]. Ця бібліотека базується на фреймворку JAX, який забезпечує легку інтеграцію з Google Cloud Platform. JAX дозволяє компілювати та запускати програми NumPy на графічних процесорах (GPU) і тензорних процесорах (TPU). Однак не всі середовища оцінки придатності можуть бути описані як програми NumPy, особливо якщо це симуляція складного природного процесу.

**Зміст дослідження.** Для оцінки впливу паралельної оцінки придатності на продуктивність алгоритму NEAT було проведено серію експериментів з навчання контролюючих ШНМ, здатних вирішувати класичні задачі навчання з підкріпленням: балансування простого оберненого маятника та балансування оберненого маятника з двома щоглами. Експерименти виконувалися з використанням бібліотеки goNEAT [5] для реалізації послідовної та паралельної оцінки функції придатності організмів. При цьому вимірювалися такі метрики продуктивності: середній час виконання одного експерименту, середній час виконання однієї епохи еволюції та середня кількість епох еволюції на експеримент до створення успішної контролюючої ШНМ.

Для експериментів з **балансування простого оберненого маятника**, популяція складалася зі 150 організмів-вирішувачів, а максимальна тривалість еволюції становила 100 епох. Результати експериментів наведені у таблиці 1.

	Середня тривалість одного експерименту, мс	Середня тривалість однієї епохи, мс	Середня кількість епох еволюції	Кількість проведених експериментів
Послідовне	435.83	67.21	11.6	100
Паралельне	426.92	70.04	12.4	100

Таблиця 1. Результати експериментів за балансування просто маятника.

При вирішенні задачі балансування простого оберненого маятника успішне рішення було знайдено у всіх експериментах приблизно за 12 епох. Результати незначно відрізнялися, що можна пояснити відносною простотою та легкістю обчислення цієї задачі. Таким чином, накладні витрати на організацію паралельних обчислень нейтралізують досягнуті покращення у швидкості виконання.

Експерименти з **балансування оберненого маятника з двома щоглами** проводилися на популяції з 1000 організмів-вирішувачів, з максимальною тривалістю до 100 епох еволюції для пошуку рішення. Результати експериментів наведені у таблиці 2.

	Середня тривалість одного експерименту, сек	Середня тривалість епохи, сек	Середня кількість епох еволюції	Кількість проведених експериментів
Послідовне	82.53	0.807	66.7	10
Паралельне	<b>17.80</b>	<b>0.097</b>	66.7	10

Таблиця 2. Результати експериментів за балансування маятника з двома щоглами.

В обох режимах успішне рішення було знайдено у 8 з 10 експериментів протягом 67 епох еволюції. Застосування паралельної оцінки функції пристосованості значно покращило швидкість навчання контролюючої ШНМ, **збільшивши її більш ніж у 4 рази**. Це пояснюється тим, що симуляція балансування оберненого маятника з двома щоглами є дуже ресурсомісткою задачею зі складними обчисленнями. Отже, паралельне виконання таких симуляцій під час оцінки пристосованості організмів у популяції наприкінці кожної епохи еволюції значно скорочує час нейроеволюційного процесу навчання ШНМ.

#### Література

1. Stanley, O. Kenneth and Miikkulainen, Risto. Evolving Neural Networks through Augmenting Topologies. *Evolutionary Computation*, 2002. Vol. 10, N. 2. pp. 99-127. DOI: <https://doi.org/10.1162/106365602320169811>
2. Omelianenko, Iaroslav. *Hands-On Neuroevolution with Python: Build high-performing artificial neural network architectures using neuroevolution-based algorithms*. Birmingham, UK : Packt Publishing, 2019. ISBN: 9781838824914, 368 p.
3. Desell, Travis. Large scale evolution of convolutional neural networks using volunteer computing. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO '17)*. Association for Computing Machinery, 2017. New York, NY, USA. pp. 127–128. DOI: <https://doi.org/10.1145/3067695.3076002>
4. Tang, Yujin, Tian, Yingtao, and Ha, David. EvoJAX: hardware-accelerated neuroevolution. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO '22)*. Association for Computing Machinery, New York, NY, USA, 2022. pp. 308–311. DOI: <https://doi.org/10.1145/3520304.3528770>
5. Omelianenko, Iaroslav. The GoLang implementation of NeuroEvolution of Augmented Topologies (NEAT) algorithm. Version v4.2.0, 2024. DOI: <https://doi.org/10.5281/zenodo.13628842>