УДК 004.8 http://doi.org/10.15407/pp2024.02-03.263

Я.В. Омельяненко, А.Ю. Дорошенко, Є.С. Родін

ЗАСТОСУВАННЯ СТРАТЕГІЇ КОЕВОЛЮЦІЇ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ АВТОНОМНОГО ПРОХОДЖЕННЯ ЛАБІРИНТУ

У статті розглянуто застосування методу коеволюції для вирішення задачі пошуку виходу зі складного лабіринту автономним агентом, що керується штучною нейронною мережею (ШНМ). Вирішення цієї задачі висвітлює фундаментальну проблему, яка перешкоджає алгоритмічній оптимізації – тісне поєднання цілі завдання із цільовою функцією, що використовується для пошуку оптимального рішення. Для вирішення цієї задачі ціль є очевидною – знаходження оптимального шляху крізь лабіринт. У той же час, з цільовою функцією все дещо складніше. В складних конфігураціях лабіринту зазвичай налічується багато оманливих зон, які наближені до точки виходу з лабіринту, але є глухими кутами, що не мають шляху до виходу. Таким чином, проста цільова функція, заснована тільки на вимірюванні відстані до виходу з лабіринту, має багато локальних оптимумів в оманливому просторі пошуку рішення. З підвищенням складності конфігурації лабіринту, проста цільова функція рано чи пізно застряє у одному з локальних оптимумів, що унеможливлює вирішення задачі навігації.
Для вирішення цієї проблеми було запропоновано використання стратегії коеволюції популяції агентів-вирішувачів та популяції кандидатів у цільові функції. На відміну від попередніх робіт у цьому напрямку, було запропоновано використання алгоритму NEAT для керування процесом коеволюції обох популяцій. Водночас для оптимізації пошуку у просторі рішень було запропоновано використання методу Novelty Search (NS), який полягає у наданні переваги найбільш інноваційним рішенням. Було наведено опис математичного апарату для визначення шаблону цільової функції, що включає у себе як значення інноваційності рішення, так і значення відстані до виходу з лабіринту. Цей шаблон був взятий за основу при визначенні генному організмів – кандидатів у цільові функції. Для порівняння з попередніми роботами було проведено експеримент для визначення ефективності запропонованого методу коеволюції у вирішенні задачі навігації у складному лабіринті.
Ключові слова: генетичні алгоритми, нейроеволюція наростаючих топологій, автономне проходження лабіринту, NEAT, коеволюція.

Ia.V. Omelianenko, A.Yu. Doroshenko, Ye.S. Rodin

APPLICATION OF COEVOLUTION STRATEGY TO SOLVE THE PROBLEM OF AUTONOMOUS NAVIGATION TROUGH THE MAZE

This study explores the use of coevolutionary methods to address the challenge of navigating through complex mazes using autonomous agents controlled by artificial neural networks (ANNs). It underscores a critical impediment to algorithmic optimization: the close interdependence between the task's goal and the objective function used for optimal solution discovery. The task's goal is clear—identify the most efficient route through the maze. However, the objective function's formulation is more complex. In complex maze layouts, numerous deceptive areas may appear proximate to the exit but culminate in dead ends. Consequently, an elementary objective function that merely gauges the proximity to the exit can encounter numerous local optima within this deceptive search space, hindering the search for optimal solution. As maze complexity increases, such an objective function inevitably becomes ensnared in a local optimum, rendering the navigation issue unsolvable.
To counteract this, the study proposes a coevolution strategy involving a population of decision-making agents and a population of objective function candidates. This approach diverges from prior research by incorporating the NEAT algorithm to steer the coevolution of both populations. Additionally, the Novelty Search (NS) method was suggested to optimize the search within the potential solution space, favoring the most novel solutions.
The paper details the mathematical framework for crafting the objective function template, which integrates the novelty value of the discovered solution and its distance from the maze's exit. This framework serves as the foundation for defining the genomes of the organisms — candidates for the objective functions.
For comparison with preceding works, an experiment was conducted to evaluate the efficacy of the proposed coevolution method in resolving the problem of navigation within a complex maze environment.
Keywords: genetic algorithms, neuroevolution of augmenting topologies, autonomous maze navigation, NEAT, coevolution

Вступ

Застосування будь-якого еволюційного алгоритму (EA) для навігації у складному середовищі зводиться до вирішення задачі пошуку оптимального шляху для досягнення заданої мети. Це, у свою чергу, базується на визначенні цільової функції (функції пристосованості), яку ми бажаємо мінімізувати або максимізувати. У попередній роботі [1, 2] було показано, що існує фундаментальна проблема, яка виникає на практиці. У той час, як ціль завдання може бути чітко визначена та добре відома, цільова функція може бути оманливою.

Це можна наочно проілюструвати на прикладі побудови моделі автономного агента для проходження двовимірного лабіринту, зображеного на Рис. 1. Завданням контролюючої моделі є керування роботом таким чином, щоб за визначену кількість кроків він міг пройти лабіринт з початкового положення до виходу. Контролююча модель водночас визначає поведінку робота (напрямок руху) на кожному кроці, залежно від поточного стану середовища: позиція у лабіринті, відстань до стін.

Інтуїтивно можливо визначити цілеорієнтовану функцію як функцію відстані між поточною позицією робота у лабіринті та виходом з нього, як було зроблено у [1]. Але, у складній конфігурації лабіринту агент-вирішувач може зіткнутися з наростаючою складністю вибору адекватного напрямку руху на основі відстані до виходу.

Як показано на Рис. 1, навіть, якщо відстань до виходу здається мінімальною, це не означає, що шлях до виходу знайдено. Функція пристосованості може мати локальні оптимуми у глухих кутах лабіринту, де реєструються круті градієнти значень показників пристосованості (fitness score) [3, 4]. Отож, ми маємо оманливий ландшафт показників пристосованості, який не може бути вирішений за допомогою цілеорієнтованої функції пристосованості (goal-oriented fitness function).

Таким чином, можна зазначити, що навіть, якщо оптимальне рішення може бути визначене контролюючою моделлю, воно не завжди буде знайдене в процесі еволюції, використовуючи просту цільову функцію. Ба більше, наше інтуїтивне визначення функції пристосованості здається хибним, адже воно ототожнює кінцеву мету (вихід) та дизайн цільової функції (відстань до виходу).

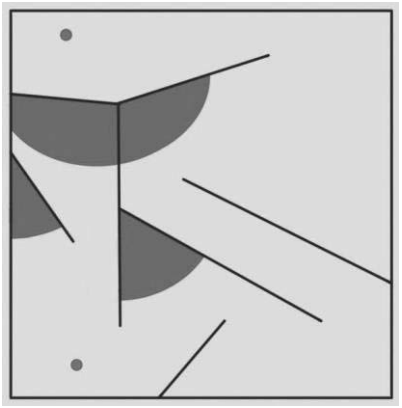

Рис. 1. Двовимірний лабіринт з локально оптимальними глухими кутами (затемнені).

Для розв'язання цієї проблеми у [5] було запропоновано розділити оптимізацію контролюючої моделі та оптимізацію цільової функції, використовуючи метод SAFE (Solution And Fitness Evolution). Іншими словами, навіть, якщо метою агента-вирішувача є досягнення точки виходу з лабіринту, це не обов'язково означає, що цільовою функцією є відстань до виходу. Таким чином, пропонується використовувати метод коеволюції двох популяцій організмів – популяції контролюючих моделей та популяції кандидатів у цільові функції.

Новизна підходу, розглянутого у цій роботі, полягає у використанні алгоритму нейроеволюції NEAT [2, 6, 7, 9] для еволюції популяції контролюючих моделей.

У даній роботі будуть розглянуті принципи та існуючі типи коєволюції популяцій організмів. Після цього буде запропоновано метод побудови коеволюційного процесу на основі алгоритму NEAT для навчання контролюючих моделей та алгоритму NEAT, що поєднаний з пошуковою оп-

тимізацією методом NS для визначення оптимальної цільової функції.

Розглянутий метод коєволюції дозволяє вирішувати задачу оптимальної навігації у складному середовищі та вирішувати завдання навчання Штучних Нейронних Мереж (ШНМ), здатних контролювати поведінку автономних роботизованих агентів.

1. Огляд аналогів

Оптимальна навігація у складному оточенні – важлива задача для побудови автономних агентів. Вирішенню цієї задачі присвячено низку наукових досліджень.

У [1] було запропоновано використання алгоритму NEAT із простою цільовою функцією, визначеною як відстань між поточною позицією агента та кінцевою метою (вихід з лабіринту). Цей підхід добре працює для простих конфігурацій лабіринту, але стикається з труднощами у процесі розв'язання складніших конфігурацій.

Для вирішення задачі побудови оптимальних контролюючих моделей, здатних розв'язувати складні конфігурації лабіринту у [3, 4, 8] було запропоновано метод оптимізації під назвою Novelty Search (NS). Згідно із цим методом, оптимізація еволюційного процесу відбувається так, щоб найбільшу винагороду отримували організми, які знайшли більш нове рішення у просторі вже знайдених рішень. Таким чином, цільова функція задається як фактор новизни, а не відстань до мети. Головним рушієм цього методу є натуральна необмежена еволюція, яка також у багатьох випадках оптимізує геном організмів, використовуючи нові комбінації для максимального поширення на невідомі території і пошук нових поведінкових моделей. Цей метод має цілий ряд переваг, таких як - необмежений простір пошуку і можливість знаходження нетривіального рішення. Крім того, пошук новизни заради самої новизни не дає стимулу для контролюючої моделі, що наблизилась до мети, залишатися на цій траєкторії та покращувати результати. Ба більше, з ускладненням конфігурації лабіринту, час потрібний на розв'язання задачі навігації методом NS, може бути досить значним, що робить використання цього методу непрактичним у більшості випадків.

На відміну від зазначених вище методів оптимізації пошуку оптимального рішення, запропонований у цій роботі метод коеволюції популяції агентів-вирішувачів та популяції кандидатів у цільові функції, дозволяє знаходити оптимальні рішення задачі навігації у складних конфігураціях лабіринту за прийнятний проміжок часу. Крім того, використання алгоритму NEAT для керування еволюцією агентів-вирішувачів дає змогу отримувати оптимальні топології ШНМ контролюючих моделей. Це робить їх енергоефективними та уможливлює їх використання у системах з обмеженими обчислювальними та енергетичними можливостями, такими, як промислові роботи, автономні дрони і таке інше.

Новизна методу, розглянутого у даній роботі, полягає у поєднані алгоритму NEAT для еволюції популяції агентів-вирішувачів (контролюючих моделей) та алгоритму NEAT з пошуковою оптимізацією на основі методу NS (пошук новизни) для еволюції популяції кандидатів у цільові функції. Крім того, ця робота представляє нову модифікацію методу NS, спрямовану на обмеження простору пошуку для підвищення продуктивності.

У статті наведені основні програмні модулі для реалізації методу коеволюції на мові програмування GO.

У наступному розділі ми розглянемо основні особливості коеволюції.

2. Особливості коеволюції

Природна еволюція біологічних систем не може розглядатися окремо від концепції коеволюції. Коеволюція є однією з основних рушійних сил еволюції, від якої залежить поточний стан біосфери та різноманітність організмів.

Ми можемо визначити коеволюцію як взаємовигідну стратегію одночасної еволюції безлічі генеалогій різних організмів. Водночас, еволюція одного виду неможлива без інших. У ході еволюції спільно еволюціонуючі види взаємодіють один з одним, і ці міжвидові відносини формують

їхню еволюційну стратегію. Існує три основних типи коеволюції:

– *мутуалізм*, коли два або більше види мирно співіснують та отримують взаємну вигоду один від одного;

– *конкурентна* коеволюція:

хижацтво, коли один організм убиває інший і споживає його ресурси;

паразитизм, коли один організм використовує ресурси іншого, але не вбиває його;

– *комменсалізм*, коли представники одного виду отримують вигоду, не завдаючи шкоди та не приносячи вигоди іншим видам.

Останній тип коеволюційної стратегії привернув увагу дослідників [5] як багатообіцяючий для створення ефективного методу навчання автономних агентів. Одночасно, було запропоновано реалізацію алгоритму SAFE, який ми розглянемо далі.

3. Особливості алгоритму SAFE

Як випливає з назви, метод коеволюції вирішувача та пристосованості заснований на спільній еволюції рішення та функції пристосованості, котра спрямовує оптимізацію пошуку рішення. Метод SAFE створений навколо стратегії коменсалістичної коеволюції двох популяцій:

– популяція потенційних рішень, які розвиваються, щоб вирішити безпосередньо поставлене завдання;

– популяція кандидатів на цільові функції, які еволюціонують, щоб спрямовувати еволюцію популяції рішень.

Основна ідея алгоритму SAFE полягає в тому, щоб отримати вигоду з використання різних методів оптимізації пошуку для кожної з згаданих популяцій окремо.

У даній роботі запропоновано використання алгоритму NEAT для керування еволюцією популяції потенційних рішень та його поєднання з NS (пошук новизни) для оптимізації еволюційного процесу у популяції кандидатів на цільові функції.

Далі ми розглянемо модифікований експеримент з лабіринтом, який буде використано для оцінки ефективності запропонованого рішення.

4. Модифікований експеримент із лабіринтом

У [1] було наведено реалізацію класичної задачі з проходження лабіринту, використовуючи алгоритм NEAT для керування еволюційним процесом. Для оптимізації пошуку використовувалась проста цільова функція, заснована на відстані від поточного положення агенту-вирішувача до виходу з лабіринту.

У цій роботі були використані засоби, які вже були реалізовані в [1] з певними змінами, необхідними для реалізації методу коеволюції.

Головною відмінністю від попередньої праці є визначення функцій пристосованості для обох коеволюційних популяцій та їх відповідна реалізація. Таким чином, нам потрібно визначити дві функції пристосованості: одну - для кандидатів на вирішувачі лабіринтів та іншу - для кандидатів на цільові функції. Тут ми розглянемо обидва варіанти.

5. Функція пристосованості для агенту-вирішувача

В кожному поколінні еволюції кожен індивідуум (вирішувач лабіринту) оцінюється всіма кандидатами на цільові функції з іншої коменсалістичної популяції. Ми використовуємо максимальну оцінку пристосованості, отриману під час оцінювання кожного окремого агенту-вирішувача кожним кандидатом на цільову функцію, як оцінку пристосованості знайденого рішення.

Функція пристосованості вирішувача лабіринту являє собою сукупність двох метрик - відстані від виходу з лабіринту (оцінка близькості до мети) і новизни кінцевої позиції вирішувача (оцінка новизни). Ці оцінки арифметично поєднуються з використанням пари коефіцієнтів, отриманих як *вихідні дані від конкретного індивідуума* в популяції кандидатів на цільову функцію.

Наступна формула дає комбінацію цих показників для оцінки пристосованості:

$$O_i(S_i) = a \times \frac{1}{D_i} + b \times NS_i\ , \quad (1)$$

Тут $O_i(S_i)$ – це значення пристосованості, отримані шляхом оцінки кандидата рішення S_i, щодо цільової функції O_i. Пара коефіцієнтів $[a, b]$ - це вихід конкретного кандидата на цільову функцію. Ця пара визначає, якою мірою відстань до виходу з лабіринту (D_i) і поведінкова новизна (NS_i) рішення впливають на кінцеву оцінку пристосованості вирішувача лабіринту в кінці траєкторії.

Відстань до виходу з лабіринту (D_i) визначається як евклідова відстань між останніми координатами вирішувача лабіринту в його траєкторії та координатами виходу з лабіринту. Відстань обчислюється за такою формулою:

$$D_i = \sqrt{\sum_{i=1}^{2}(a_i - b_i)^2}\ , \qquad (2)$$

Де a – координати кінцевої позиції агенту та b – координати виходу з лабіринту.

Оцінка новизни NS_i кожного агенту-вирішувача лабіринту визначається його остаточним положенням в лабіринті (точка x). Вона розраховується як середня відстань від цієї точки до k-найближчих сусідніх точок, які є остаточними позиціями інших вирішувачів лабіринтів.

Наступна формула дає значення оцінки новизни в точці x поведінкового простору:

$$NS_i = \frac{1}{k}\sum_{i=0}^{k} dist(x, \mu_i)\ , \qquad (3)$$

Де μ_i – це i-й найближчий сусід x, а $dist(x, \mu_i)$ – відстань між x та μ_i.

Відстань між двома точками є метрикою новизни, що вимірює, наскільки поточне рішення (x) відрізняється від іншого (μ_i), створеного різними вирішувачами лабіринтів. Показник новизни розраховується як евклідова відстань між двома точками:

$$dist(x, \mu) = \sqrt{\sum_{j=1}^{2}\left(x_j - \mu_j\right)^2}\ , (4)$$

Тут μ_j і x_j – значення позиції j координатних векторів, що містять ко-ординати точок μ і x відповідно.

Далі ми обговоримо, як визначити функцію адаптації для оптимізації кандидатів на роль цільової функції.

6. Функція пристосованості для кандидатів на цільову функцію

Метод SAFE заснований на коменсалістичному коеволюційному підході, який означає, що одна зі спільно еволюціонуючих популяцій не отримує ні користі, ні шкоди в ході еволюції. У експерименті комменсалістична популяція є сукупністю кандидатів на цільові функції. Для цієї сукупності нам необхідно визначити таку функцію пристосованості, яка залежить від якості роботи агентів-вирішувачів лабіринту (контролюючих моделей).

Придатним варіантом тут є цільова функція пристосованості, яка для оцінки пристосованості використовує показник новизни. Формула для розрахунку оцінки новизни кожного кандидата на роль цільової функції така ж сама, як і для агентів-вирішувачів лабіринтів (3). Єдина відмінність полягає в тому, що у випадку кандидатів на роль цільової функції ми розраховуємо оцінку новизни, використовуючи вектори з вихідними значеннями кожного індивідуума у популяції. Після цього застосовуємо значення показника новизни як показника пристосованості організму.

У [3, 4, 8] наведено реалізацію методу оптимізації пошуку NS, в який ми вимушено внесли модифікації для обмеження простору пошуку. Це зроблено, щоб підвищити продуктивність алгоритму та гарантувати його виконання у заданий проміжок часу.

7. Середовище моделювання задачі лабіринту

У рамках цієї роботи було розроблене програмне забезпечення для моделювання задачі проходження лабіринту схоже до наведеного у [1], але використовуючи мову програмування *GO* з відповідними змінами для організації коеволюції двох по-

пуляцій. Воно складається з таких основних компонентів, які реалізовані у вигляді окремих класів:

– *Agent* – клас, який зберігає інформацію, пов'язану з агентом навігатора лабіринту, задіяного в симуляції.

– *RecordStore* - клас, який управляє зберіганням записів, що належить до оцінок усіх вирішальних агентів в ході еволюційного процесу. Зібрані записи можна використовувати для аналізу еволюційного процесу після його завершення.

– *Environment* – клас, який містить інформацію про середовище моделювання лабіринту. Цей клас також надає методи, які керують середовищем моделювання, керують положенням вирішального агенту, виявляють зіткнення зі стінами лабіринту та генерують вхідні дані для датчиків агента.

– *NoveltyItem* – клас, для інкапсуляції інформації про певний елемент, який містить інформацію про оцінку новизни, пов’язану з певним геномом, разом із допоміжною інформацією. Він використовується в поєднанні з NoveltyArchive

– *NoveltyArchive* – клас, який управляє збереженням усіх знайдених *NoveltyItem* та надає можливість для оцінки їх новизни в міру знаходження нових точок.

Далі ми розглянемо деталі проведеного експерименту.

8. Експеримент із складною конфігурацією лабіринту

У цій роботі було проведено дослідження з метою побудови контролюючої моделі, яка здатна вирішити задачу навігації у складному лабіринті, схема якого наведена на Рис. 1. Це дозволяє порівняти метод коеволюції з методом простої еволюції, що наведений у попередній роботі [1]. За результатами уієї роботи було показано, що простий еволюційний процес, керуючись алгоритмом NEAT, може вирішити задачу навігації в простому лабіринті, але стикається з труднощами у застосуванні його до складніших конфігурацій лабіринту. Тому цей експеримент є показовим для доведення ефективності застосування методу коеволюції.

Лабіринт має дві фіксовані позиції, відзначені забарвленими кругами. Нижній лівий круг позначає початкову позицію агента-вирішувача лабіринту. Верхній лівий круг позначає точне місце виходу з лабіринту, яке повинно бути знайдене. Щоб виконати завдання, робот повинен досягти точки поблизу виходу з лабіринту.

9. Результати експерименту

Експеримент був проведений з використанням мови програмування GO версії 1.24.1 та бібліотек goNEAT версії 4.0.2 [10], goNEAT_NS версії 4.0.2 [11]. Робоча станція, що використовувалась для цього, має наступні параметри: CPU 2,3 GHz 8-Core Intel Core i9, 16 GB 2667 MHz DDR4, macos 14.4.1.

Успішний агент-вирішувач був знайдений після 211 поколінь еволюції та має наступну конфігурацію - 22 вузли поєднані 47 зв’язками.

У процесі коеволюції також були знайдені оптимальні коефіцієнти [a, b] цільової функції, яка була використана для навчання успішного агента-вирішувача: [-0.53283, 0.95889]. Таким чином, формулу (1) можна переписати, підставивши знайдені коефіцієнти:

$$O_i(S_i) = -0.53 \times \frac{1}{D_i} + 0.96 \times NS_i, \quad (5)$$

Згідно з формулою (5) можна зробити висновки, що знайдена цільова функція акцентує навчання на пошуку найбільш інноваційних рішень (NS_i), приділяючи менше уваги значенню відстані до виходу з лабіринту (D_i). Це підтверджує тезу, наведену на початку цієї роботи про те, що у складному оточенні успішна цільова функція не тотожна відстані до мети (виходу з лабіринту).

Рис. 2. Шлях успішного агента-вирішувача (знизу – початок, зверху – вихід).

Доречним буде розглянути візуалізацію проходження лабіринту успішним агентом-вирішувачем (Рис. 2), звертаючи увагу на плавність та наближеність до оптимального знайденого маршруту.

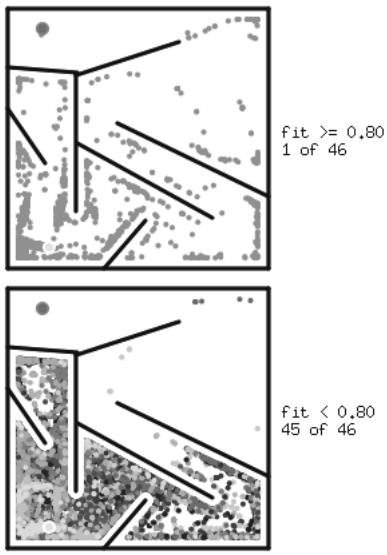

Рис. 3. Візуалізація оцінки популяції організмів для агентів-вирішувачів.

Крім того, на Рис. 3 можна побачити, що протягом усіх епох еволюції тільки один вид організмів з наявних 46 породив конфігурацію контролюючої ШНМ успішного агента-вирішувача. На цьому рисунку ми також бачимо, що більшість еволюційних невдах застрягло у локальних оптимумах. Водночас, успішний агент виявив найбільшу інноваційність, надаючи перевагу дослідженню нових ділянок, на противагу використанню вже відомих (exploration vs exploitation).

Висновки

У роботі показано, як метод коеволюції двох популяцій – популяції агентів-вирішувачів та популяції у кандидати на цільову функцію може бути використано для розв'язання задачі навігації у складному лабіринті. Експериментально доведено, що цей метод є ефективнішим за метод розглянутий у [1].

Порівняно з попередніми роботами [5], було запропоновано новий підхід до реалізації методу коеволюції SAFE (Solution and Fitness Coevolution), який полягає у застосуванні алгоритму NEAT для керування еволюційним процесом двох популяцій, об'єднанних коменсалістичною коеволюцією. Водночас для оптимізації пошуку рішень у оманливому середовищі потенційних рішень було застосовано метод Novelty Search (пошук новизни).

Було створено програмне забезпечення для проведення експерименту з коеволюції мовою програмування GO. Крім того, були використанні передові засоби візуалізації, які дозволяють наочно оцінювати результати коеволюції зі зміною вхідних параметрів.

References

1. Iaroslav Omelianenko. 2023. Simulation of the autonomous maze navigation using the NEAT algorithm. Problems in programming 4, (2023), 76-89. DOI: 10.15407/pp2023.04.076
2. Iaroslav Omelianenko. Hands-On Neuroevolution with Python: Build high-performing artificial neural network architectures using neuroevolution-based algorithms. Birmingham, UK: Packt Publishing Ltd, 2019. ISBN: 9781838824914, 368 pp.
3. Joel Lehman and Kenneth O Stanley. 2010. Revising the evolutionary computation abstraction: minimal criteria novelty search. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2010). ACM, 103–110.
4. Joel Lehman and Kenneth O Stanley. 2011. Abandoning objectives: Evolution through the search for novelty alone. Evolutionary computation 19, 2 (2011), 189–223.
5. M. Sipper, J. H. Moore and R. J. Urbanowicz, "Solution and Fitness Evolution (SAFE): A Study of Multiobjective Problems," 2019 IEEE Congress on Evolutionary Computation (CEC), Wellington, New Zealand, 2019, pp. 1868-1874, doi: 10.1109/CEC.2019.8790274.
6. Kenneth O Stanley and Risto Miikkulainen. 2002. Evolving neural networks through augmenting topologies. Evolutionary computation 10, 2 (2002), 99–127.
7. Сініцин І.П., Дорошенко А.Ю., Мамедов Т.А., Яценко О.А. Метод автоматизованого проєктування нейроеволюційних алгоритмів з використанням алгебри алгоритмів Глушкова. Проблеми керування та інформатики, 2023, №3. С.74-85. https://doi.org/10.34229/1028-0979-2023-3-8
8. Iaroslav Omelianenko. Creation of Autonomous Artificial Intelligent Agents Using Novelty Search Method of Fitness Function Opti-

mization. NewGround LLC, Sept. 2018, https://hal.science/hal-01868756.
9. Iaroslav Omelianenko. “Autonomous Artificial Intelligent Agents”. In: Machine Learning and the City. John Wiley Sons, Ltd, 2022. Chap. 12, pp. 263–285. ISBN: 9781119815075, DOI: 10.1002/9781119815075.ch21, URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119815075.ch21, SCOPUS: 2-s2.0-85147956837, https://www.scopus.com/record/display.uri?eid=2-s2.0-85147956837&origin=inward&txGid=b119d677e846feb8f319ba241f759c75
10. Omelianenko, Iaroslav (2023). The GoLang implementation of NeuroEvolution of Augmenting Topologies (NEAT) algorithm (v4.0.2). [Computer software]. Zenodo. https://doi.org/10.5281/zenodo.10119451
11. Omelianenko, Iaroslav (2024). The GOLang implementation of NeuroEvolution of Augmenting Topologies (NEAT) with Novelty Search optimization (v4.0.2). [Computer software]. Zenodo. https://doi.org/10.5281/zenodo.10951521

Одержано: 12.04.2024
Внутрішня рецензія отримана: 20.04.2024
Зовнішня рецензія отримана: 27.04.2024

Про авторів:

[1]*Омельяненко Ярослав Вікторович,*
аспірант,
молодший науковий співробітник
https://orcid.org/0000-0002-2190-5664

[1,2]*Дорошенко Анатолій Юхимович*,
доктор фізико-математичних наук,
професор, завідувач відділу теорії комп'ютерних обчислень
http://orcid.org/0000-0002-8435-1451

[1]*Родін Євген Сергійович,*
молодший науковий співробітник
https://orcid.org/0000-0003-2416-8572

Місце роботи авторів:

[1]ІПС НАН України,
03187, м. Київ-187,
проспект Академіка Глушкова, 40.
E-mail: yaric@newground.com.ua
Сайт: www.iss.nas.gov.ua

[2]НТУУ «КПІ імені Ігоря Сікорського».
м. Київ, проспект Перемоги 37.