

**О.С. Магас***Дніпровський національний університет імені Олеся Гончара***КОНСТРУЮВАННЯ НЕЙРОННОЇ МЕРЕЖІ ДЛЯ АПРОКСИМАЦІЇ  
РОЗВ'ЯЗКУ ОБЕРНЕНОЇ ЗАДАЧІ З НЕЛІНІЙНИМ  
ЕЛІПТИЧНИМ ОПЕРАТОРОМ**

Розглянута задача ідентифікації властивостей деформованої системи, що описується нелінійним еліптичним оператором. Для цього випадку використовується варіаційна постановка задачі у поєднанні із чисельними методами оптимізації. Основним напрямком роботи є не тільки пошук наближеного розв'язку із використанням нейромережевого підходу, а ще й отримання простих архітектур нейронних мереж із мінімальною кількістю компонентів. Для досягнення цієї мети запропоновані модифікації до генетичного алгоритму конструювання нейронних мереж NEAT. Модифікований варіант цього алгоритму було застосовано для ідентифікації декількох типових моделей навантаження на тонкостінну циліндричну оболонку. Модифікації ставлять за мету пристосувати алгоритм до особливостей розглянутих моделей та оптимізувати перебір топологій за допомогою збереження кількості нейронів у вузлах мережі та відокремлення навчання нейронної мережі від генетичного ітераційного процесу. У результаті були отримані мережі для ідентифікації властивостей розглянутих моделей. Наведені результуючі архітектури згаданих нейронних мереж.

**Ключові слова:** нейронна мережа, генетичний алгоритм, видоутворення, NEAT, обернена задача, ідентифікація властивостей, еліптичний оператор, тонкостінна оболонка.

**O.S. Mahas***Oles Honchar Dnipro National University***NEURAL NETWORK CONSTRUCTION FOR APPROXIMATING THE  
SOLUTION OF THE INVERSE PROBLEM WITH NONLINEAR  
ELLIPTIC OPERATOR**

The problem of identifying the characteristics of a deformed system described by a nonlinear elliptic operator is considered. For this case, the variational problem statement is used in combination with numerical optimization methods. To solve this problem, a neural network approach with genetic algorithm tuning is used. The possible ways of encoding networks in such genetic algorithms can be divided into direct and indirect encoding methods. The direct encoding methods require a fixed neural network topology and weight coefficient space exploration using the recombination of the weight vectors. However, the neural network behavior is not only determined by its weights but also by its topology. So, the main goal of the current work is to find not only an approximate solution but also a simple neural network topology with a minimal component number. In this regard, the algorithm of the permutating through various network topologies (NEAT) is considered, and its adaptation to the described problem is proposed. Finding the minimal possible network topology is an important problem due to the ease of use such networks in real systems and

simpler further analysis of the processes occurring in the network nodes. The additional modifications of the genetic algorithm for constructing a neural network are proposed. Those modifications aim to adapt the algorithm to the features of the considered model and to optimize the topology search. The first approach is to encode the genome with the additional list representing the neuron count in respective network nodes. The second approach is to separate network learning process from the genetic iterative process. This modified algorithm is applied to identify typical loading patterns on a thin-walled cylindrical shell. The results of iterative process of the genetic algorithm are obtained, and the neural networks are constructed to identify the characteristics of the considered models. The current paper presents the resulting topologies of the mentioned neural networks.

**Keywords:** neural network, genetic algorithm, speciation, NEAT, inverse problem, characteristics identification, elliptic operator, thin shell.

**А.С. Магас**

*Днепропетровский национальный университет имени Олеся Гончара*

## **КОНСТРУИРОВАНИЕ НЕЙРОННОЙ СЕТИ ДЛЯ АППРОКСИМАЦИИ РЕШЕНИЯ ОБРАТНОЙ ЗАДАЧИ С НЕЛИНЕЙНЫМ ЭЛЛИПТИЧЕСКИМ ОПЕРАТОРОМ**

Рассмотрена задача идентификации свойств деформированной системы, описываемой нелинейным эллиптическим оператором. Для этого случая используется вариационная постановка задачи в сочетании с численными методами оптимизации. Основным направлением работы является не только поиск приближенного решения с использованием нейросетевого подхода, но и получение простых архитектур нейронных сетей с минимальным количеством компонентов. Для достижения этой цели предложены модификации к генетическому алгоритму конструирования нейронных сетей NEAT. Модифицированный алгоритм был применен для идентификации нескольких типовых моделей нагрузки на тонкостенную цилиндрическую оболочку. Целью модификаций стоит приспособить алгоритм к особенностям рассматриваемых моделей и оптимизировать перебор топологий с помощью хранения количества нейронов в узлах сети и отделения обучения нейронной сети от генетического итерационного процесса. В результате были получены сети для идентификации свойств рассматриваемых моделей. Приведены результирующие архитектуры упомянутых нейронных сетей.

**Ключевые слова:** нейронная сеть, генетический алгоритм, видообразование, NEAT, обратная задача, идентификация свойств, эллиптический оператор, тонкостенная оболочка.

**Вступ.** Розглядається задача ідентифікації властивостей деформованої системи, що описується нелінійним еліптичним оператором. У цьому випадку використовується варіаційна постановка задачі у поєднанні із чисельними методами оптимізації [1]. Для розв'язання цієї задачі пропонується використовувати нейромережвий підхід [2-3] у комбінації із генетичним алгоритмом. Безліч можливих способів кодування мережі можна поділити на методи прямого та опосередкованого кодування. Методи прямого кодування передбачають наявність фіксованої топології мережі в поєднанні з дослідженням простору вагових коефіцієнтів за допомогою схрещування вагових векторів мережі. Але поведінка мережі визначається не лише ваговими коефіцієнтами, а ще її архітектурою.

У зв'язку з цим пропонується використовувати алгоритм перебору різних мережевих топологій [4] та адаптувати його під потрібну задачу. Більш того, отримання мінімально можливої архітектури нейронних мереж є важливою задачею через зручність їх використання в реальних системах та подальшого аналізу процесів, що відбуваються у вузлах таких мереж.

**Постановка прямої задачі.** Таким чином, розглянемо спочатку пряму задачу знаходження розв'язків нелінійного еліптичного рівняння типу фон Кармана [1]. Нехай в деякій заданій обмеженій області  $\Omega = \{X | X = (x_1, x_2) \in R^2\}$  із границею  $\Gamma$  вектор-функція  $u(X, H)$  описується рівняннями:

$$Q_1 \equiv \nabla_{ij} (A_1^{ijkl}(H_1) \nabla_{kl} u_1) - 1^{ik} 1^{jl} \nabla_{kl} u_2 (B_{ij}(H_1) + \nabla_{ij} u_1) = \lambda H_2, \quad (1)$$

$$Q_2 \equiv \nabla_{ij} (A_2^{ijkl}(H_1) \nabla_{kl} u_2) - 1^{ik} 1^{jl} \nabla_{kl} u_1 (B_{ij}(H_1) + \nabla_{ij} u_1) = 0, \quad (2)$$

де  $Q = \{Q_1, Q_2\}$  – вектор-функції лівих частин рівняння прямої задачі,  $X \in \Omega$  – вектор просторових координат;  $H = \{H_1, H_2\}$  – вектор-функції власностей системи, причому  $H_1 = H_1(X)$  – вектор-функція коефіцієнтів рівняння,  $H_2 = H_2(X)$  – вектор-функція правої частини рівняння;  $u = (u_1, u_2)$  – вектор-функція розв'язків прямої задачі;  $A_2^{ijkl}(H_1) > 0$ ,  $A_2^{ijkl}(H_2) > 0$ ,  $B_{ij}(H_1)$  – відомі функціонали відомої вектор-функції коефіцієнтів  $H_1(X)$ ;  $\Gamma$  – кусково-гладка межа області  $\Omega$ ;  $\Gamma_i$  – кусково-гладкі контури класу  $C^1_\Gamma$ , причому  $\sum_i \Gamma_i = \Gamma$ ;  $\nabla_i = \frac{\partial}{\partial x_i}$ ,  $\nabla_{ij} = \frac{\partial^2}{\partial x_i \partial x_j}$  для  $i, j = 1, 2$ ,  $\lambda$  – вектор параметрів.

Розглядаються нульові граничні умови:

$$u_\Gamma = 0 ; \frac{\partial u_\Gamma}{\partial n} = u_n = 0, \text{ де } n \text{ – нормаль до поверхні } \Omega. \quad (3)$$

Встановимо множини допустимих розв'язків прямої та оберненої задач.

$$\tilde{u} : \begin{cases} \tilde{u}(X) \in V_\Omega^{1,2}; Q(\tilde{u}) = 0 \\ \tilde{u}|_\Gamma = 0, \tilde{u}_n|_\Gamma = 0 \end{cases}, \quad \tilde{H} : \begin{cases} \underline{H} \leq H \leq \overline{H}; H \in W_{2,\Omega}^1 \\ a \leq \frac{\partial H}{\partial u} \leq b; \frac{\partial^2 H(X)}{\partial u^2} \geq 0 \end{cases}. \quad (4)$$

Тут  $\tilde{u}$  – множина розв'язків задачі (1)-(3),  $\tilde{H}$  – множина невідомих функцій оберненої задачі,  $\underline{H}, \overline{H}, a, b$  – задані границі.

**Постановка оберненої задачі.** До прямої задачі (1-3) сформулюємо обернену задачу відповідно до [1]. Нехай вектор-функція  $u(X, H)$  задана в обмеженій просторовій області  $\Omega = \{X | X = (x_1, x_2) \in R^2\}$  наступним рівнянням із використанням диференційних операторів:

$$G(H_1(X), u(X)) = H_2(X), \quad (5)$$

$$G^\Gamma(u(X))|_{X=X^\Gamma \in \Gamma} = 0, \quad (6)$$

де  $X \in \Omega$  – вектор просторових координат;  $\Gamma$  – кусково-гладка межа області  $\Omega$ ;  $\Gamma_i$  – кусково-гладкі контури класу  $C_\Gamma^1$ , причому  $\sum_i \Gamma_i = \Gamma$ ;  $H = \{H_1, H_2\}$  – невідомі функції оберненої задачі, причому  $H_1(X)$  – функція властивостей моделі, що задані диференціальним оператором  $G$ , а  $H_2(X)$  – функція, яка описує праву частину рівняння;  $G(\cdot)$ ,  $G^\Gamma(\cdot)$  – задані диференціальні оператори;  $u = (u_1, u_2)$  – функція розв’язків прямої задачі.

Розв’язок оберненої задачі, отриманий шляхом вимірювання функцій  $u(X)$  в точках  $\gamma_r$  із використанням відомих результатів спостереження за поведінкою розглянутої системи, передбачає знаходження функцій коефіцієнтів  $H_1(X)$  та правої частини  $H_2(X)$ . При цьому виміряні результати спостережень можна записати наступним чином:

$$u(\gamma_r, H) = u_r^*, \quad r = \overline{1, N}, \quad (7)$$

Область вимірювань  $\tilde{H}$  невідомих функцій  $H$  оберненої задачі визначається їх фізичним змістом. Відтак, обернена задача зводиться до задачі мінімізації цільового функціонала:

$$J(H) = \rho_W(u(X, H), u^*) \rightarrow \min, \quad (8)$$

де  $\rho_W$  – функціонал-нев’язка, що визначається метрикою в деякому функціональному просторі  $W$  відповідно до (4).

У даній роботі пропонується розглянути ідентифікацію властивостей моделі, які задані функцією  $H_2(X)$ , та використати для цього нейромеревий підхід у поєднанні із генетичним алгоритмом. Доведення можливості застосування такого підходу та проблеми, пов’язані із некоректністю постановки задачі, а саме із порушенням умови існування розв’язку, порушенням умови неперервності розв’язку за даними та можливим існуванням декількох розв’язків, розглянуті окремо в роботі [2]. При цьому, для використання нейронної мережі пропонується перейти від функціонала (8) до:

$$J^*(H) = \rho_W(H(X), H^*) \rightarrow \min, \quad (9)$$

де  $H_r^*$  – властивості моделі, що відповідають спостереженням  $u_r^*$ .

**Метод розв’язання.** Для оберненої задачі (5-6) моделюються спостереження  $(u_n^*, H_n^*)$  для  $n = \overline{1, K}$ , які є розв’язками прямих задачах на множині допустимих значень. Кількість елементів  $K$  визначається розміром множини допустимих розв’язків оберненої задачі.

У рамках даної роботи для конструювання та структурної оптимізації нейронної мережі був застосований генетичний підхід на основі алгоритму NEAT [4], а також запропоновані модифікації до нього, причому вхідними сигналами нейронної мережі виступають значення  $\{u_n^*\}$ , а вихідними сигналами – відомі з прямої задачі властивості моделі  $\{H_n^*\}$ . Основні принципи функціонування нейронної мережі описані в [3].

Такий підхід є евристичним алгоритмом пошуку шляхом випадкового підбору, комбінування та перебору шуканих параметрів із використанням принципів, які імітують біологічну еволюцію. Алгоритм передбачає паралельну обробку множини альтернативних розв'язків, зосереджуючи пошук на найбільш перспективних серед них. Спочатку розглянемо наступні етапи узагальненого генетичного алгоритму:

1) Підготовчий етап. На цьому етапі необхідно створити початкову популяцію, тобто сформувати початковий набір розв'язків.

2) Етап відбору. На цьому етапі визначається напрямок розвитку популяції, в основному, шляхом відкидання розв'язків із найменшими значеннями функції пристосованості, що сприяє покращенню її середнього значення за всією популяцією.

3) Етап схрещування. На цьому етапі за допомогою оператора схрещування відбувається утворення нових розв'язків для відновлення чисельності популяції. Особливістю оператора схрещування є його застосування до декількох геномів представників-батьків з метою утворення шляхом рекомбінації інформації із батьківських генів нових представників-нащадків, які залишаються у популяції.

4) Етап мутацій. Оскільки основною ідеєю оператора схрещування є комбінація відомої інформації про можливий розв'язок, то його використання не обов'язково призводить до розширення простору пошуку, що в свою чергу може привести до втрати можливо кращого розв'язку. На цьому етапі для розширення простору пошуку розв'язку до геномів представників, які залишились у популяції, застосовується оператор мутації. Завдяки використанню випадкового процесу він вносить до популяції нову інформацію, тим самим збільшує різноманітність її представників.

5) Етап оцінки розв'язку. На цьому етапі проводиться порівняння значень функцій пристосованості усіх представників популяції, та із використанням заданого критерію зупинки алгоритму приймається одне із двох рішень: повторити етапи 2-5; позначити найбільш пристосованого представника популяції як остаточний розв'язок. У разі продовження алгоритму поточна популяція вважається створеною для етапу 2. Критеріями зупинку можуть бути як кількість створених популяцій, так і значення функції пристосованості.

Усі сенсори та нейрони послідовно нумеруються. Додатково вводиться поняття інновації – пари індексів, що відповідають з'єднаним нейронам (та сенсорів), причому кожна унікальна інновація має свій індекс. Геном кодується списком інновацій, ваговими коефіцієнтами відповідних до інновацій з'єднань нейронної мережі, а також бітами активності цих з'єднань. Таким чином можна закодувати нейронну мережу прямого поширення довільної топології. Варто також зауважити, що утворювати геноми із циклічними з'єднаннями забороняється, адже це призведе до утворення рекурентних нейронних мереж, які в даній роботі не розглядаються.

Функція пристосованості  $F_i$   $i$ -ого представника обчислюється наступним чином:

$$F_i = 1 - E(i), \quad (10)$$

де  $E(i)$  – середньоквадратичне відхилення результатів ідентифікації нейронної мережі  $i$ -ого представника від еталонних спостережень.

Наведений алгоритм на етапі схрещування передбачає використання спеціального оператора, робота якого полягає у наступному. Утворений оператором геном включає всі спільні батьківські гени з однаковими індексами інновацій, при цьому значення біта активності вибирається випадково в одного із батьків. Гени, які не присутні одночасно в обох батьківських геномах, додаються в новий геном із заданою імовірністю. Відповідні значення вагових коефіцієнтів із однаковими інноваціями або випадковим чином вибираються із батьківських генів, або обчислюються як їх середнє значення.

Для етапу мутацій розглядаються наступні оператори.

1) Оператор структурної мутації для додавання нового з'єднання. Із використанням випадкового процесу обираються два вузли нейронної мережі (сенсор-нейрон або нейрон-нейрон) та з'єднуються між собою. Тобто до геному додається активний ген із номером інновації, який відповідає з'єднанню між цими вузлами мережі (рис. 1).

2) Оператор структурної мутації для додавання нового нейрона. Із використанням випадкового процесу обирається існуюче з'єднання вузлів мережі, на місці якого створюється новий нейрон, що з'єднується із відповідними вузлами. Тобто у геномі випадково вибраний активний ген стає неактивним, натомість до геному додаються два активні гени із номерами інновацій, яким відповідають з'єднання між попередньо пов'язаними вузлами та новим нейроном (рис. 2).

3) Мутація збурення вагових коефіцієнтів. Вагові коефіцієнти кожного гену збурюються на випадкову величину  $\Delta w_g$  із наперед заданого розподілу імовірностей  $D_M$ , де  $g$  – індекс гену.

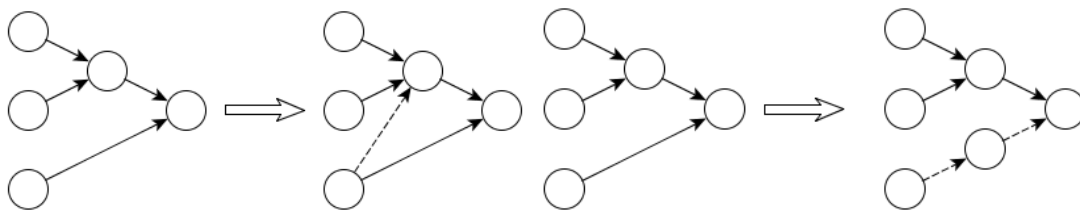


Рис. 1. Додавання зв'язку

Рис. 2. Додавання нейрону

Із міркувань бажаної мінімальності структури проміжних та результуючої нейронних мереж, для підготовчого етапу початкова популяція геномів кодує нейронні мережі без прихованих нейронів. Такий підхід дозволяє уникнути необхідності використовувати додаткові штрафні компоненти до функції пристосованості, а також зменшує простір пошуку розв'язку та використання обчислювальних ресурсів.

Для етапу відбору застосовується механізм видоутворення, основна ідея якого полягає у додаванні конкуренції між топологічно подібними геномами. Цей механізм має забезпечити багатовекторність пошуку та захист перспек-

тивних геномів від знищення. У роботі використовується відоме поняття відстані між геномами [4]:

$$\delta = c_1 \frac{N_E}{N_G} + c_2 \frac{N_D}{N_G} + c_3 \overline{\Delta W}, \quad (11)$$

де  $c_1, c_2, c_3$  – коефіцієнти видоутворення;  $N_E$  – кількість спільних генів із відмінними бітами активності;  $N_D$  – кількість відмінних генів;  $N_G$  – розмір більшого геному,  $\overline{\Delta W}$  – середнє значення абсолютних різниць вагових коефіцієнтів відповідних спільних інновацій незалежно від біту активності.

Уведене поняття відстані необхідне для обчислення модифікованої функції пристосованості, значення якої буде використане для відбору замість оригінальної. Модифікована функція пристосованості  $i$ -ого геному обчислюється за формулою [4]:

$$F_i^* = \frac{F_i}{\sum_{j=1}^n sh(\delta(i, j))}, \text{ де } sh(\delta) = \begin{cases} 0 & \delta > \delta_t \\ 1 & \delta \leq \delta_t \end{cases}, \quad (12)$$

а  $\delta_t$  – порогове значення приналежності геному до одного виду.

Оскільки класичне використання механізму видоутворення потребує ретельного евристичного підбору коефіцієнту  $\delta_t$ , пропонується використовувати модифікований алгоритм видоутворення, що полягає у виборі кількості видів, яка обґрунтована розміром популяції та доступними обчислювальними ресурсами. А також у відмові використання модифікованої функції пристосованості (12) на користь наступного алгоритму. Популяція представників рівномірно поділяється на задану кількість видів, причому на етапі відбору виконуються наступні кроки:

- 1) Інформація щодо видів популяції із попереднього етапу знищується.
- 2) Найкращий геном серед тих, які ще не належать жодному виду, вважається основним представником нового виду.
- 3) Список представників цього виду заповнюється до своєї максимальної ємності такими, що мають геноми із найменшими значеннями відстані (11) до найкращого геному виду із кроку 2.
- 4) Якщо залишились представники, які ще не належать жодному із видів, перейти до кроку 2.
- 5) Модифікувати етап відбору таким чином, щоб відбір найкращих представників за оригінальною функцією пристосованості відбувався не у популяції в цілому, а окремо всередині кожного виду.

Для зменшення простору пошуку розв'язку пропонується модифікація, яка використовує відому інформацію про характер вхідних даних, а саме симетричність модельованої оболонки. Її сутність полягає в тому, що на кроці кодування геному усі сенсори записуються не окремо, а як єдина зв'язка. Така модифікація зменшує імовірність утворення випадкової залежності асиметричної до певної ділянку входу та поліпшує кінцевий розв'язок.

Тестування роботи алгоритму показало, що дуже часто топологія нейронних мереж розвивається швидше за налаштування вагових коефіцієнтів, а

довготривалий підбір параметрів генетичного алгоритму не завжди призводить до вирішення цієї проблеми. Це призводить до громіздких, але недостатньо оптимізованих у своїх нейронних з'єднань мереж. Відтак, застосування додаткових методів навчання до побудованих нейронних мереж показує суттєве зменшення середньоквадратичної похибки. Тому для вирішення цієї проблеми пропонується застосувати одну із наступних модифікацій.

1) Розділити загальну вибірку на вибірку навчання та тестування (80% та 20%). Застосувати алгоритм навчання нейронної мережі за вибіркою навчання до моменту, поки середньоквадратична похибка на вибірці тестування не почне зростати, або поки кількість епох навчання не досягне певного заданого значення. Використати значення середньоквадратичної похибки на вибірці тестування для обчислення функції пристосованості (10)

2) Відмовитись від зберігання вагових коефіцієнтів у геномі та встановити  $c_3 = 0$  у формулі відстані між геномами (11). Навчання нейронної мережі та обчислення функції пристосованості відбувається аналогічно модифікації 1, за винятком того, що одночасно розглядається множина нейронних мереж однакової топології потужністю  $n_r$ , але із різними початковими значеннями вагових коефіцієнтів. Нейронна мережа із найменшим значенням середньоквадратичної похибки потрапляє до банку нейронних мереж та надалі повторно використовується для ідентичних геномів з метою уникнення зайвих обчислень. Вибір початкових вагових коефіцієнтів ґрунтується на методі Ксав'є [5] із використанням випадкового процесу за формулою:

$$W_{i,j} \sim N\left(\mu = 0, \sigma^2 = \frac{2}{in(j) + out(j)}\right), \quad (13)$$

де  $W_{i,j}$  – ваговий коефіцієнт з'єднання від вузла з індексом  $i$  до вузла з індексом  $j$ ,  $in(j)$  – кількість вхідних з'єднань вузла з індексом  $j$ ,  $out(j)$  – кількість вихідних з'єднань вузла з індексом  $j$ .

Під час спостережень за ітераціями генетичного алгоритму було виявлено, що продуктивні геноми представників часто утворюють структури, які мають частини, що є типовими для класичних багатошарових нейронних мереж. Але для їх конструювання необхідне вдаль застосування операторів схрещування та мутацій. У зв'язку з цим, у кодуванні геномів та інноваціях пропонується розглядати не окремі вузли, а їх зв'язки. Відтак, пропонується модифікація, згідно якої до геному додається список пар  $(i, n_i)$ , який встановлює кількість нейронів  $n_i$  у зв'язці вузла з індексом  $i$ . Кількість нейронів у зв'язці не може бути меншою за 1, а зв'язка вихідного вузла завжди має мінімальний розмір.

Застосування цієї модифікації вимагає додавання до оператора схрещування додаткового кроку для об'єднання вузлових списків: відповідні значення розміру зв'язки із однаковими вузловим індексам випадковим чином або вибираються із батьківських генів, або обчислюються як їх середнє значення з округленням до цілого.



Крім того, необхідно додати новий оператор мутації для збурення розмірів зв'язок, алгоритм якого полягає у наступному. Із використанням випадкового процесу вибирається індекс вузла, та із заданою імовірністю розмір зв'язки цього вузла або збільшується на 1, або зменшується на 1, або ділиться навпіл з округленням до цілого.

В якості функції активації нейронів використаний гіперболічний тангенс, а для навчання нейронної мережі для розглянутих модифікацій застосовано алгоритм RPROP [6], оскільки він потребує меншої кількості ітерацій навчання та не вимагає налаштування фіксованого коефіцієнту навчання.

**Результати.** Розв'язання задачі (9) здійснювалось поетапно із послідовним введенням описаних алгоритмів. Надалі демонструються найкращі розв'язки із усіма розглянутими модифікаціями основного алгоритму, який був застосований для ідентифікації функції правої частини  $H_2(X)$  рівняння (5) з наступними характеристиками. Область  $\Omega$  – замкнута тонкостінна циліндрична оболонка з параметрами  $L/R = 4$ ,  $R/h = 100$ ,  $E = 2 \cdot 10^4$  МПа,  $\mu = 0,3$ , де  $L, R, h$  – довжина, радіус, товщина оболонки;  $E, \mu$  – модуль Юнга, коефіцієнт Пуассона. Причому,  $0 \leq x_1 \leq L/R$ ,  $-\pi \leq x_2 \leq \pi$ .

Функція правої частини  $H_2(X)$ , яка характеризує закон зміни навантаження, описується моделями  $A, B, C$ , та задається відповідними формулами:

$$H_A(X) = \begin{cases} \lambda_0 & x_2 \in [-q\pi; q\pi], \\ 0 & x_2 \notin [-q\pi; q\pi], \end{cases} \quad (14)$$

$$H_B(X) = \lambda_0(1 + \cos x_2)^q, \quad (15)$$

$$H_C(X) = \lambda_0(1 + \cos qx_2), \quad (16)$$

де  $\lambda_0, q$  – параметри навантаження. Типові елементи вибірки з цих моделей показано відповідно на рис. 3-5.

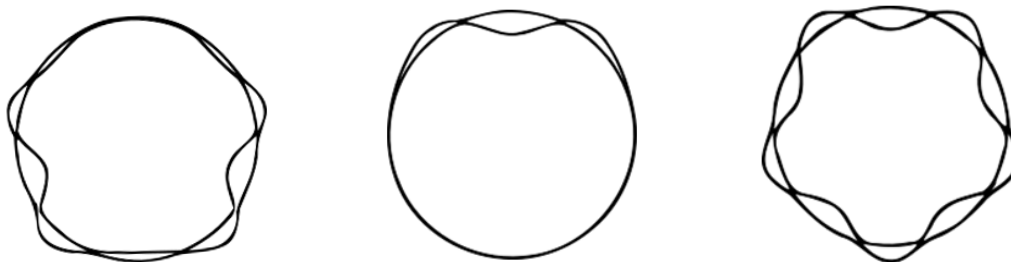


Рис. 3. Елемент із моделі  $A$ . Рис. 4. Елемент із моделі  $B$ . Рис. 5. Елемент із моделі  $C$ .

Інформація про розв'язки прямих задач  $(u_n^*, H_n^*)$ ,  $n = \overline{1, K}$  отримана на основі нелінійного скінченного елемента, який реалізований у пакеті інженерних розрахунків «COSMOS/M».

За результатами роботи на рис. 6-9 показні архітектури нейронних мереж відповідно до моделей правої частини  $A, B, C$ , а також комбінованої моделі, вибірка якої включає усі зазначені моделі одночасно.

Архітектура результуючої нейронної мережі для моделі *A* (рис. 6) передбачає поєднання одного прихованого вузла з двома незалежними підмережами другого порядку (вузли 2-4 та 5-7).

Для моделі *B* архітектура її результуючої нейронної мережі (рис. 7) передбачає поєднання одного прихованого вузла із двома незалежними підмережами другого (вузли 6, 8) та третього (вузли 2-5, 7) порядків.

У моделі *C* відповідна архітектура (рис. 8) передбачає поєднання одного прихованого вузла із двома підмережами третього порядку (вузли 2-4 та 5-8) зі спільним вузлом 2.

У свою чергу, нейронна мережа комбінованої моделі (рис. 9) передбачає поєднання мережі з одним прихованим прошарком (вузли 1-4) із двома незалежними підмережами другого (вузли 5-6) та третього (вузли 7-10) порядків.

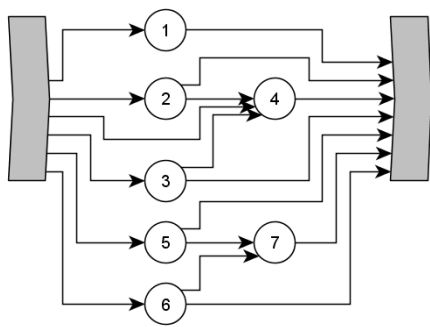


Рис. 6. Нейронна мережа моделі *A*.

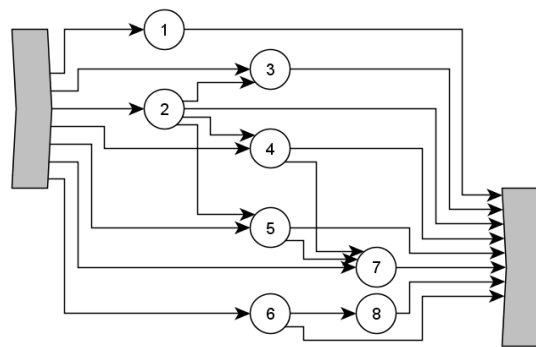


Рис. 7. Нейронна мережа моделі *B*.

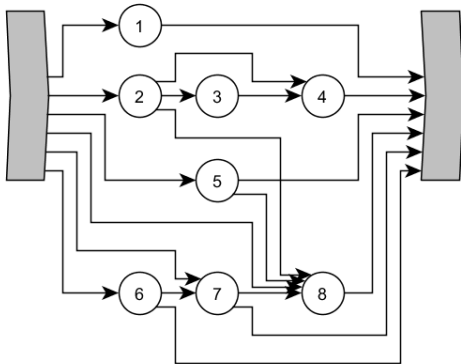
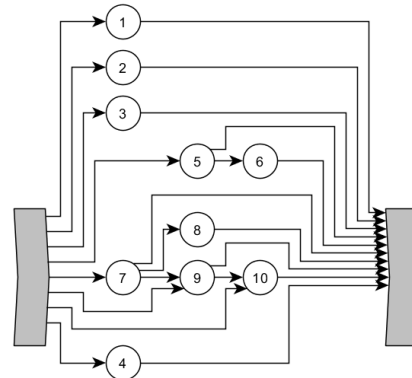


Рис. 8. Нейронна мережа моделі *C*. Рис. 9. Нейронна мережа комбінованої моделі.



**Висновки.** Середньоквадратична похибка на вибірці для перевірки узагальнюючої здатності нейронних мереж склала менше 1%. Сконструйовані нейронні мережі для розглянутої задачі є простими, що робить доцільним їх подальше використання та аналіз.

#### Бібліографічні посилання

1. Ободан Н.І., Гук Н.А. Ідентифікація навантажень за допомогою динамічної нейронної мережі. *Машинознавство*. 2013. №4. С. 38-45.
2. Ободан Н.І., Гук Н.А., Магас А.С. Коректність нейросетевой аппроксимации решения обратной задачи для нелинейного эллиптического оператора. *Питання прикладної математики і математичного моделювання*. Д., 2015. С. 147-156.

3. **Haykin S.S.** Neural networks and learning machines. Third. *Pearson Education*. 2009. P. 934.
4. **Stanley K.O, Miikkulainen R.** Evolving Neural Networks Through Augmenting Topologies. *Evolutionary Computation*. 2002. V. 10. №2. P. 99-127.
5. **Glorot X., Bengio Y.** Understanding the Difficulty of Training Deep Feedforward Neural Networks. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research*. 2010. №9. P. 249-256.
6. **Riedmiller M.A., Braun H.** A direct adaptive method for faster backpropagation learning: the RPROP algorithm. *IEEE International Conference on Neural Networks*. 1993. V.1. P. 586-591.

Надійшла до редколегії 26.09.2022.