

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Чорноморський національний університет імені Петра Могили**

д.т.н., доцент **О.В. Козлов**

д.т.н., професор **Ю.П. Кондратенко**

**Методи та моделі інтелектуальних обчислень**

Навчальний посібник

Миколаїв - 2024

УДК 004.896

Рекомендовано до друку Вченою радою ЧНУ ім. Петра Могили  
(протокол №\_ від \_\_\_\_\_)

**Рецензенти:** д.т.н., професор Атаманюк І.П.  
д.т.н., професор Максимов М.В.

д.т.н., доцент О.В. Козлов, д.т.н., професор Ю.П. Кондратенко.  
**Методи та моделі інтелектуальних обчислень.** Навчальний посібник. –  
Миколаїв: ЧНУ ім. П. Могили, 2024. – 184 с.

Навчальний посібник містить систематизоване викладення основних методів, моделей і систем теорії інтелектуальних обчислень, які застосовуються при вирішенні прикладних завдань прийняття рішень, керування, оптимізації, прогнозування та ін. Розглянуто нечіткі системи, нейронні та нейро-нечіткі мережі, біоінспіровані еволюційні та мультиагентні алгоритми. Поряд із теоретичними відомостями для кращого засвоєння матеріалу та проведення самостійного тестування наприкінці кожного розділу наведено контрольні питання.

Навчальний посібник призначений для студентів спеціальностей 122 «Комп'ютерні науки» та 124 «Системний аналіз» вищих навчальних закладів, зокрема, при вивченні дисциплін «Математичні методи інтелектуальних обчислень», «Інтелектуальні методи оптимізації складних систем» та «Нейромережеві методи обчислювального інтелекту», а також може використовуватися аспірантами, науково-педагогічними працівниками та практичними фахівцями.

## З М І С Т

ВСТУП .....	4
РОЗДІЛ 1 НЕЧІТКІ СИСТЕМИ, НЕЙРОННІ ТА НЕЙРО-НЕЧІТКІ МЕРЕЖІ .....	7
1.1 Інтелектуальні системи на основі нечіткої логіки .....	7
1.2 Штучні нейронні мережі .....	25
1.3 Гібридні нейро-нечіткі мережі .....	47
1.4 Контрольні питання .....	55
РОЗДІЛ 2 БІОІНСПІРОВАНІ ЕВОЛЮЦІЙНІ ТА МУЛЬТИАГЕНТНІ МЕТОДИ ОПТИМІЗАЦІЇ .....	59
2.1 Генетичні методи .....	62
2.2 Методи штучних імунних систем .....	74
2.3 Методи оптимізації на основі біогеографії .....	81
2.4 Методи мурашиних колоній .....	88
2.5 Методи рою частинок .....	99
2.6 Методи сірих вовків .....	106
2.7 Контрольні питання .....	113
РОЗДІЛ 3 ЗАСТОСУВАННЯ БІОІНСПІРОВАНИХ МЕТОДІВ ДЛЯ СИНТЕЗУ ТА ОПТИМІЗАЦІЇ НЕЙРОМЕРЕЖЕВИХ ТА НЕЧІТКИХ СИСТЕМ .....	120
3.1 Нейроеволюційний метод керування багатозв'язними складними технічними об'єктами .....	121
3.2 Мультиагентний метод генерації баз правил нечітких систем на основі мурашиних алгоритмів .....	141
3.3 Контрольні питання .....	165
ЛІТЕРАТУРА .....	168

## ВСТУП

Сучасний стан розвитку штучного інтелекту (ШІ) відображає вражаючий прогрес у галузі обчислювальних наук та інформаційних технологій. Штучний інтелект, як інтердисциплінарна наука, спрямований на створення комп'ютерних систем, здатних аналізувати інформацію, робити розумові висновки і вирішувати складні творчі завдання, які раніше потребували виключно людського інтелекту.

Важливість розвитку ШІ полягає в його потенціалі змінити більшість сфер життя людини, забезпечити автоматизацію процесів прийняття рішень та керування, що приводить до підвищення ефективності та продуктивності. Штучний інтелект допомагає розв'язувати нетривіальні задачі в області медицини, промисловості, наукових досліджень, транспорту, енергетики та багатьох інших галузях. Крім того, зі збільшенням ролі ШІ у суспільстві постають виклики, пов'язані з етикою, приватністю, безпекою та соціальними наслідками. Ці питання вирішуються шляхом розробки стандартів та регуляторних політик для забезпечення відповідального використання ШІ, збереження приватності даних і врахування впливу на робочі місця та суспільство.

Подальший розвиток ШІ залежить від поєднання зусиль академічних дослідників, індустріальних компаній та урядових організацій. Інноваційні дослідження, нові методи, моделі, алгоритми та технології є ключем до вдосконалення систем інтелектуальних обчислень та досягнення нових висот у вирішенні складних завдань. У майбутньому можна очікувати на динамічний розвиток ШІ, що дозволить зробити його більш широко застосовуваним і ефективним. Водночас, вирішення етичних, правових і соціальних питань стане важливим кроком у забезпеченні сталого розвитку і позитивного впливу штучного інтелекту на наше суспільство.

Згідно з прийнятою Стратегією розвитку штучного інтелекту в Україні [1] основними напрямками сучасної теорії інтелектуальних

обчислень, за якими проводяться дослідження, є системи, що базуються на нечітких множинах та нечіткій логіці, штучні нейронні мережі, гібридні нейро-нечіткі мережі, біоінспіровані метаевристичні алгоритми оптимізації, до яких відносяться еволюційні та мультиагентні алгоритми, та ін. Тому, **метою** даного навчального посібника є систематизоване викладення основних теоретичних аспектів, методів, моделей, інформаційних технологій та систем за наведеними напрямками ШІ та теорії інтелектуальних обчислень.

Навчальний посібник складається з трьох основних розділів та списку використаних джерел.

У **першому розділі** розглянуто основні питання та теоретичні аспекти інтелектуальних нечітко-логічних систем, штучних нейронних мереж та гібридних нейро-нечітких мереж. Наведено базові аспекти нечіткої логіки, принципи побудови нечітких систем керування та підтримки прийняття рішень, а також основні механізми нечіткого логічного виведення. Розглянуто загальні відомості теорії штучних нейронних мереж. Наведено основні властивості та класифікацію існуючих нейромереж, коротко розглянуто питання їх синтезу та навчання. Представлено загальне уявлення про гібридні нейро-нечіткі мережі із зазначенням їх особливостей та переваг. Детально розглянуто нейро-нечітку мережу типу ANFIS.

**Другий розділ** присвячено біоінспірованим еволюційним та мультиагентним методам оптимізації. Наведено класифікацію, переваги та недоліки, а також особливості застосування біоінспірованих методів пошуку для вирішення прикладних задач. Докладно представлено базові принципи та основні кроки наступних методів: генетичний, штучних імунних систем, біогеографічний, мурашиних колоній, рою частинок та сірих вовків.

У **третьому розділі** розглянуто основні аспекти застосування біоінспірованих еволюційних та мультиагентних методів для вирішення

задач синтезу та оптимізації нечітких та нейромережових інтелектуальних систем із наведенням конкретних прикладів.

Для кращого засвоєння матеріалу та проведення самостійного тестування після теоретичних відомостей наприкінці кожного розділу наведено контрольні питання.

Навчальний посібник призначений для студентів спеціальностей 122 «Комп'ютерні науки» та 124 «Системний аналіз» вищих навчальних закладів, а також може використовуватися аспірантами, науково-педагогічними працівниками та практичними фахівцями.

Матеріал, наведений у посібнику, призначений для вивчення та апробований авторами при викладанні курсів «Математичні методи інтелектуальних обчислень», «Інтелектуальні методи оптимізації складних систем» та «Нейромережові методи обчислювального інтелекту» у Чорноморському національному університеті імені Петра Могили. Навчальний посібник також може використовуватися при вивченні окремих розділів дисциплін «Нечіткі моделі та методи обчислювального інтелекту» та «Нейромережові та нечіткі АСК».

## РОЗДІЛ 1

### НЕЧІТКІ СИСТЕМИ, НЕЙРОННІ ТА НЕЙРО-НЕЧІТКІ МЕРЕЖІ

У даному розділі розглянуто основні питання та теоретичні аспекти інтелектуальних нечітко-логічних систем, штучних нейронних мереж та гібридних нейро-нечітких мереж. У підрозділі 1.1 наведено базові аспекти нечіткої логіки, принципи побудови нечітких систем керування та підтримки прийняття рішень, а також основні механізми нечіткого логічного виведення. У підрозділі 1.2 розглянуто загальні відомості теорії штучних нейронних мереж. Наведено основні властивості та класифікацію існуючих нейромереж, коротко розглянуто питання їх синтезу та навчання. У підрозділі 1.3 представлено загальне уявлення про гібридні нейро-нечіткі мережі із зазначенням їх особливостей та переваг. Детально розглянуто нейро-нечітку мережу типу ANFIS.

#### 1.1 Інтелектуальні системи на основі нечіткої логіки

*Базові аспекти нечіткої логіки.* На сьогоднішній день інтелектуальні системи широко застосовуються в різних галузях людської діяльності для вирішення завдань управління, прийняття рішень, моделювання складних фізичних та економічних явищ, класифікації, розпізнавання образів, ідентифікації, прогнозування та ін. Серед представлених систем доцільно виділити окремий клас нечітких систем (НС), в основі яких лежить теорія нечітких множин (fuzzy sets) та нечіткої логіки (fuzzy logic), що запропонована Л.А. Заде [2, 3]. Ця теорія дозволяє будувати нечіткі аналоги різних математичних понять і створювати необхідний формальний апарат для моделювання людських міркувань та способів вирішення завдань. У свою чергу, нечітка множина являє собою сукупність елементів довільної природи, щодо яких не можна зі стовідсотковою впевненістю

стверджувати – чи належить той або інший елемент до даної сукупності чи ні. Також, теорія нечітких множин має справу з людськими знаннями, які прийнято називати експертною інформацією.

Згідно з теоремою про нечітку апроксимацію [4], будь-яка, скільки завгодно складна математична залежність може бути апроксимована системою, заснованою на нечіткій логіці. Це дозволяє в нечітких системах з досить високою точністю відтворювати довільні взаємозв'язки «входи-виходи» за допомогою природничо-мовних тверджень-правил типу «ЯКЩО ..., – ТО ...», з подальшою їх формалізацією на базі алгоритмів теорії нечітких множин, без використання складних розрахунків на основі диференціальних та інтегральних рівнянь [5, 6]. Тому, дані системи дозволяють ефективно використовувати експертну інформацію, формалізувати механізми мислення та прийняття рішень людини, а також формувати лінгвістичні моделі складних об'єктів та процесів [7].

Для формалізації експертної інформації нечіткі системи використовують різні (а) лінгвістичні змінні (наприклад, «швидкість руху автомобіля», «температура повітря», «рівень втоми оператора»), (б) лінгвістичні терми (наприклад, «низький», «малий», «вище середнього»), (в) функції належності (наприклад, «трикутна», «трапецеїдальна», «π-подібна»), (г) продукційну базу лінгвістичних правил, що складаються з антецедентів (умов) та консеквентів (заключень), а також (д) механізм нечіткого логічного виведення, який складається з послідовних етапів фазифікації, агрегації, активації, акумуляції та дефазифікації [8].

Традиційні математичні методи, зазвичай, дозволяють здійснювати обробку точних даних. Так, наприклад, якщо температура у приміщенні дорівнює 22 °C ( $T_1 = 22$  °C), то графічно це можна відобразити за допомогою одноточкової множини, представленої на рис. 1.1. При цьому, для точного вимірювання значення температури необхідно застосовувати різнотипні сенсорні системи.



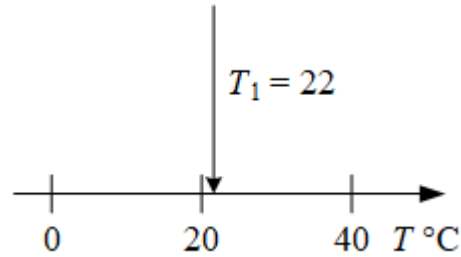


Рис. 1.1. Графічне зображення точного вимірювання температури

У свою чергу, людина може оцінювати температуру у приміщенні тільки приблизно, оперуючи такими термінами, як «низька», «середня» та «висока». Дані висловлювання у межах теорії нечіткої логіки називаються «лінгвістичними термами» (ЛТ) і графічно зображені на рис. 1.2, де прийнято наступні позначення: L – низька (Low); A – середня (Average); H – висока (High);  $\mu(T)$  – ступінь належності значень температури до певних лінгвістичних термів (може приймати значення від 0 до 1);  $\mu_A(T_1)$  – ступінь належності значення температури  $T_1$  до лінгвістичного терма «A».

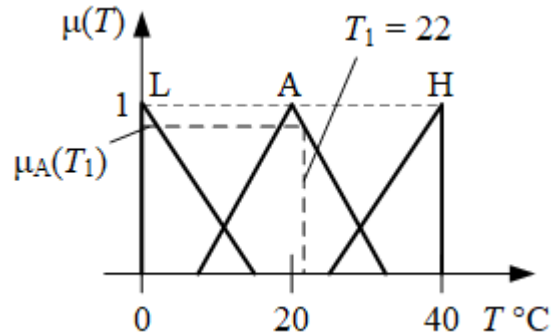


Рис. 1.2. Графічне зображення наближених оцінок температури

Якщо людина перебуває у приміщенні, температура в якому становить  $22\text{ }^{\circ}\text{C}$ , то вона не в змозі точно оцінити це значення, а може тільки зазначити, що температура «середня» (рис. 1.2). У даному випадку, значення температури  $T_1 = 22\text{ }^{\circ}\text{C}$  належить до лінгвістичного терма «A» (середня) зі ступенем належності  $0,87$ , що математично записується як

$$\mu_A(T_1) = 0,87. \quad (1.1)$$

При цьому, ступінь належності  $\mu(T)$  змінної  $T$  до певних лінгвістичних термів (L, A, H) визначається на основі їх функцій належності (ФН). У даному випадку на рис. 1.2 для всіх ЛТ використовуються ФН трикутного типу.

Нечітка логіка, також, надає широкі можливості для моделювання поведінки різнотипних об'єктів із заданою точністю. Наприклад, певний об'єкт з однією вхідною змінною  $x$  та однією вихідною змінною  $y$  має графічну характеристику  $y = f(x)$ , представлену на рис. 1.3.

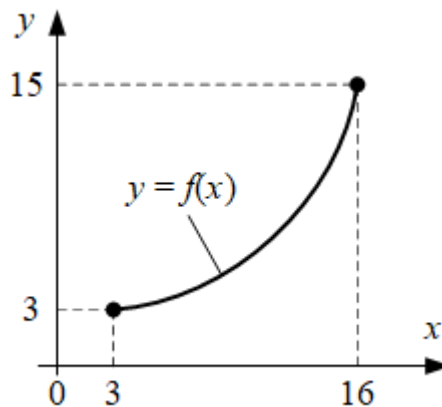


Рис. 1.3. Графічна характеристика об'єкта  $y = f(x)$

Для моделювання даного об'єкта на нечіткій логіці можна застосувати по 2 лінгвістичних терма для вхідної та вихідної змінних  $x$  та  $y$ : S – Small (малий); B – Big (великий). Для реалізації взаємозв'язку наведених термів вхідної та вихідної змінних доцільно використовувати 2 правила (R1 та R2):

$$\begin{aligned} \text{R1: IF "x = S" THEN "y = S";} \\ \text{R2: IF "x = B" THEN "y = B".} \end{aligned} \quad (1.2)$$

Графічне зображення представленого взаємозв'язку наведено на рис. 1.4, де  $y = f(x)$  – характеристика сформованої моделі.

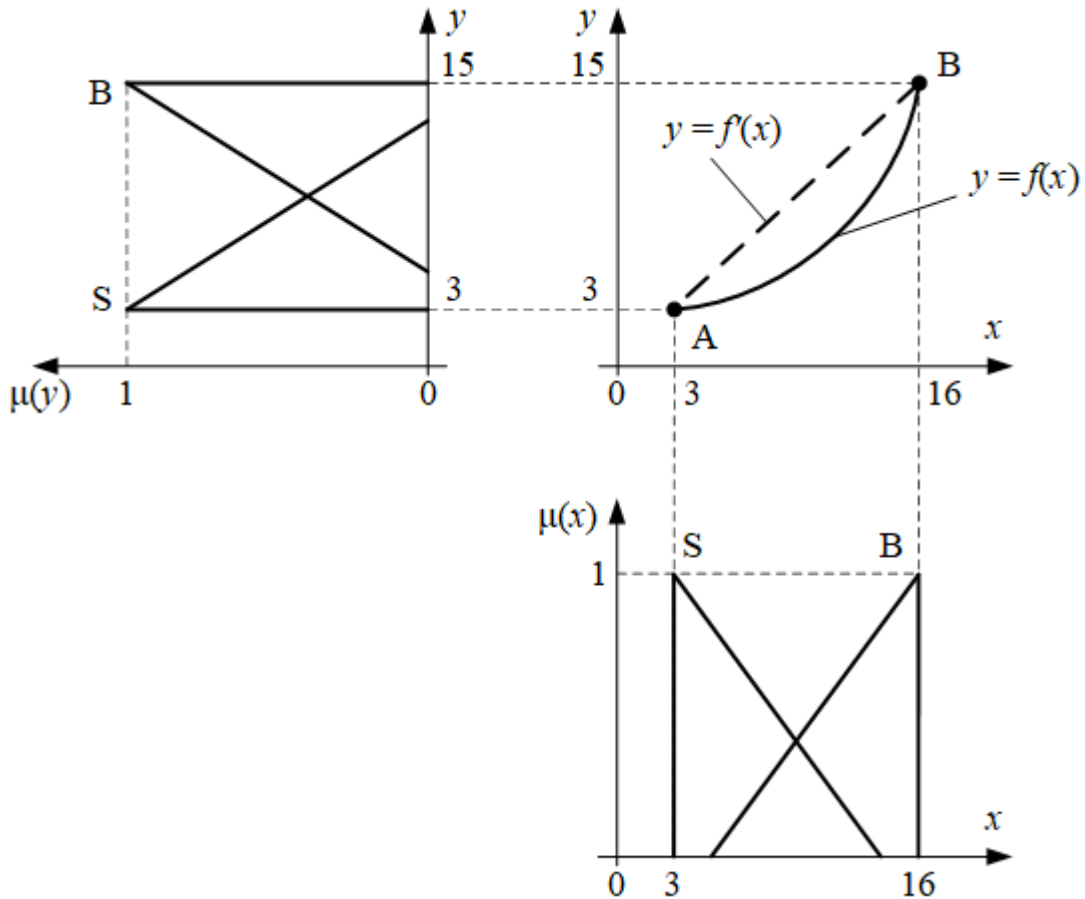


Рис. 1.4. Графічне зображення моделі об'єкта при застосуванні двох ЛТ для вхідної та вихідної змінних

Як видно з рис. 1.4, сформована модель має невисоку точність, оскільки при реалізації зв'язку між вихідною та вхідною змінними приймаються до уваги лише 2 кінцеві точки характеристики об'єкта А та В.

Для підвищення точності наведеної моделі можна прийняти до уваги додаткову третю точку С посередині характеристики об'єкта та додати ще один лінгвістичний терм для опису значень вхідної та вихідної змінних: М – Middle (середній). При цьому, взаємозв'язок лінгвістичних термів змінних моделі в даному випадку буде побудовано із застосуванням вже 3-х правил (R1, R2 та R3):

$$\begin{aligned}
 \text{R1: IF "x = S" THEN "y = S";} \\
 \text{R2: IF "x = M" THEN "y = M";} \\
 \text{R3: IF "x = B" THEN "y = B".}
 \end{aligned}
 \tag{1.3}$$

Графічне зображення представленого взаємозв'язку наведено на рис. 1.5.

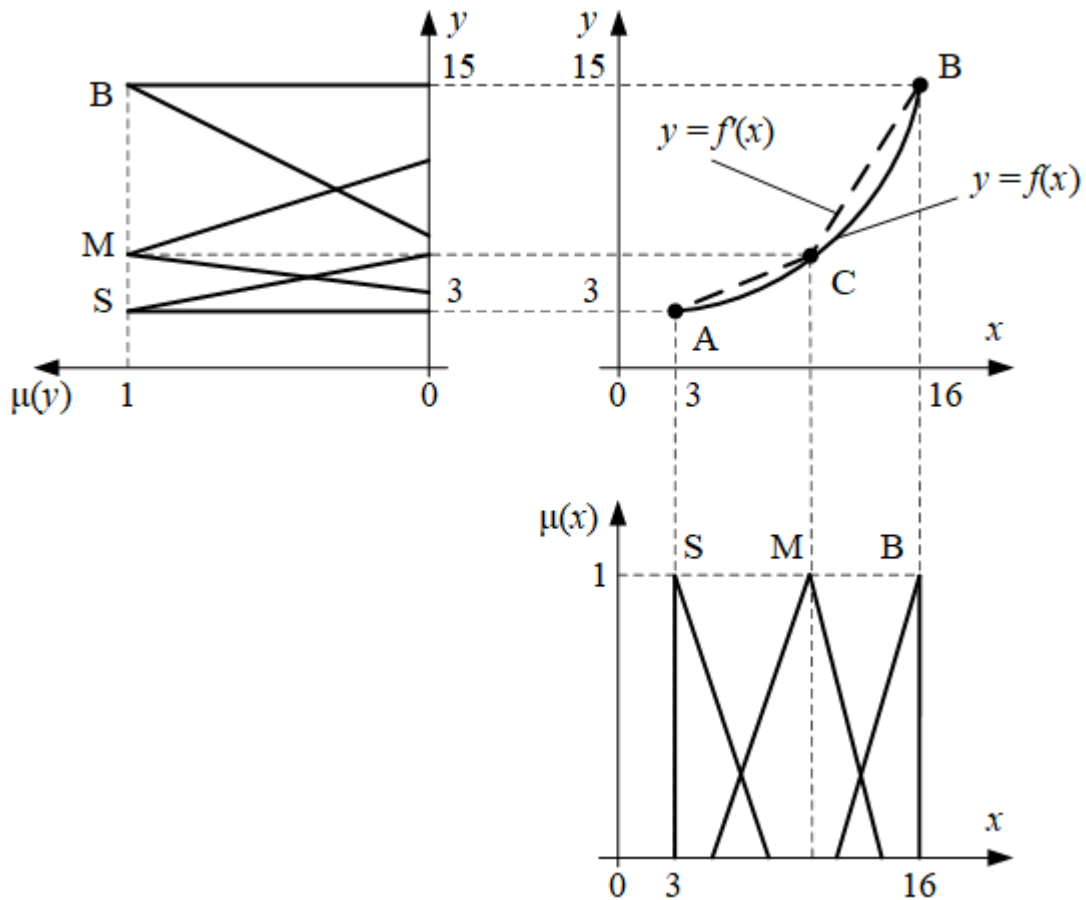


Рис. 1.5. Графічне зображення моделі об'єкта при застосуванні трьох ЛТ для вхідної та вихідної змінних

У свою чергу, при необхідності подальшого підвищення точності отриманої моделі можна розглядати ще інші точки на характеристиці об'єкта та додавати ще лінгвістичні терми та правила для опису значень змінних та їх взаємозв'язку. Однак, проведені дослідження [8] показують, що для опису будь якого параметру у більшості існуючих об'єктів та систем достатньо застосовувати не більше 5-9 ЛТ.

*Нечіткі системи керування та підтримки прийняття рішень.* Одним з найбільш перспективних напрямків застосування математичного апарату нечітких множин та нечіткої логіки є його використання при

створенні інтелектуальних систем автоматичного керування (САК) та систем підтримки прийняття рішень (СППР). Розглянемо коротко особливості побудови нечітких САК та СППР.

На рис. 1.6 представлена узагальнена структура одномірної нечіткої системи автоматичного керування, де прийнято наступні позначення: ЗП – задавальний пристрій; НК – нечіткий контролер; БРВС – блок розрахунку вхідних сигналів; БФ – блок фазифікації; БНЛВ – блок нечіткого логічного виведення; АГР – агрегація; АКТ – активація; АК – акумуляція; БДФ – блок дефазифікації; ОК – об'єкт керування; Д – датчик керованої змінної;  $y_s$ ,  $y_R$  – задане і реальне значення керованої змінної;  $\varepsilon$  – помилка керування змінною  $y$ ;  $\mathbf{E}$  – вектор вхідних сигналів НК, який розраховується на основі помилки  $\varepsilon$ ;  $u_{FC}$  – вихідний сигнал НК;  $\mathbf{F}_D$  – вектор збурювальних впливів, що діють на об'єкт керування;  $u_{SD}$ ,  $u_S$  – вихідні сигнали задавального пристрою і датчика системи керування, відповідно.

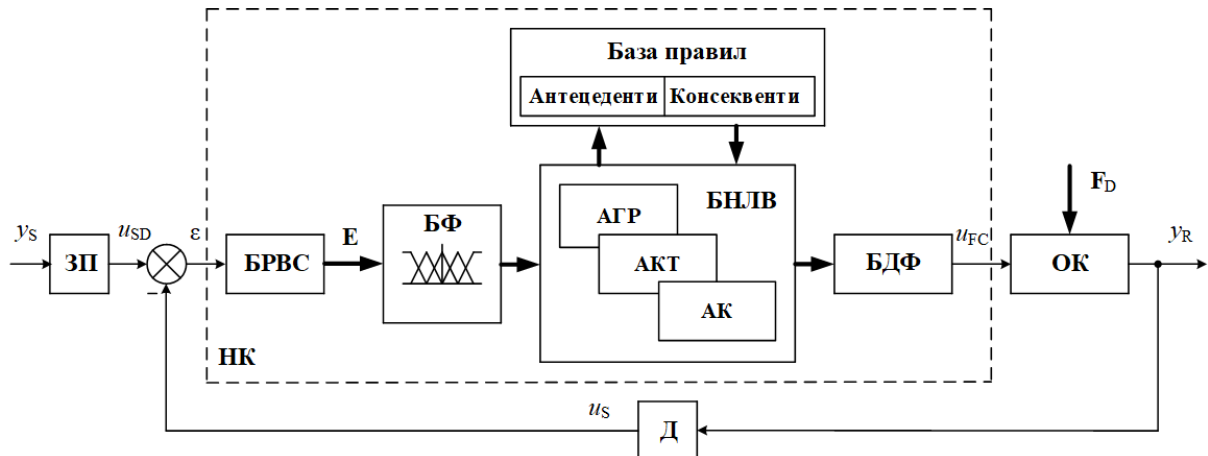


Рис. 1.6. Узагальнена структура нечіткої системи автоматичного керування

У представленій нечіткій системі керування із узагальненою структурою блок розрахунку вхідних сигналів БРВС формує вектор  $\mathbf{E}$ , який може складатися з різних комбінацій сигналів помилки  $\varepsilon$ , її похідних різних порядків, інтеграла і т.д. Наприклад, вектор  $\mathbf{E}$  може бути представлений виразами (1.4), (1.5), (1.6), (1.7) або (1.8)

$$\mathbf{E} = \{k_p \varepsilon, k_D \dot{\varepsilon}\}; \quad (1.4)$$

$$\mathbf{E} = \{k_p \varepsilon, k_I \int \varepsilon dt\}; \quad (1.5)$$

$$\mathbf{E} = \{k_p \varepsilon, k_D \dot{\varepsilon}, k_I \int \varepsilon dt\}; \quad (1.6)$$

$$\mathbf{E} = \{k_p \varepsilon, k_D \dot{\varepsilon}, k_{2D} \ddot{\varepsilon}\}; \quad (1.7)$$

$$\mathbf{E} = \{k_p \varepsilon, k_D \dot{\varepsilon}, k_{2D} \ddot{\varepsilon}, k_I \int \varepsilon dt\}, \quad (1.8)$$

де  $k_p, k_D, k_{2D}, k_I$  – нормуючі коефіцієнти, які застосовуються для приведення вхідних сигналів до відносних одиниць.

Блок фазифікації БФ визначає ступені належності чисельних значень вхідних сигналів вектору  $\mathbf{E}$  до відповідних нечітких вхідних лінгвістичних термів НК [9, 10]. При цьому, для фазифікації чітких значень змінних можуть застосовуватися наступні функції належності: трикутна, трапецеїдальна, дзвоноподібна, гаусова 1-го та 2-го роду,  $\pi$ -подібна, S-подібна, Z-подібна, сигмоїдна, двостороння сигмоїдна функція різниці, а також двостороння сигмоїдна функція добутку [11, 12]. Блок нечіткого логічного виведення БНЛВ, в свою чергу, на основі фазифікованих сигналів і отриманих даних з бази правил (БП) послідовно виконує операції агрегації, активації і акумуляції [8]. База правил являє собою набір правил, що складаються з певних антецедентів та консеквентів. У загальному випадку вектор антецедентів являє собою набір всіх можливих комбінацій ЛТ вхідних сигналів НК [10]. В свою чергу, вектор консеквентів БП може визначатись різним чином в залежності від конкретного механізму НЛВ.

Блок дефазифікації БДФ реалізує перетворення консолідованого нечіткого логічного виведення в чіткий чисельний вихідний сигнал НК [8, 10]. В свою чергу, для різних типів механізму НЛВ дефазифікація виконується різним чином, в залежності від методу дефазифікації,

кількості ЛТ вихідного сигналу, а також типів і параметрів їх функцій належності.

Нечіткі САК показують досить високу ефективність при автоматизації складних технічних об'єктів у різних галузях промисловості, транспорті, енергетиці та ін. Їхні переваги включають здатність адаптуватися до змінних умов та невизначеності в процесі керування нелінійними та нестационарними об'єктами.

У свою чергу, для вирішення проблеми автоматизації процесів прийняття рішень, оцінювання, вибору та обробки інформації в умовах неповноти даних та невизначеності досить ефективно можуть бути застосовані нечіткі СППР.

Узагальнена нечітка СППР, структура якої представлена на рис. 1.7 складається з блоку початкової підготовки даних БПД, блоку фазифікації, бази правил, блоку нечіткого логічного виведення, який здійснює процедури агрегації, активації та акумуляції, блоку дефазифікації та блоку видачі рішення БВР [10, 13, 14].

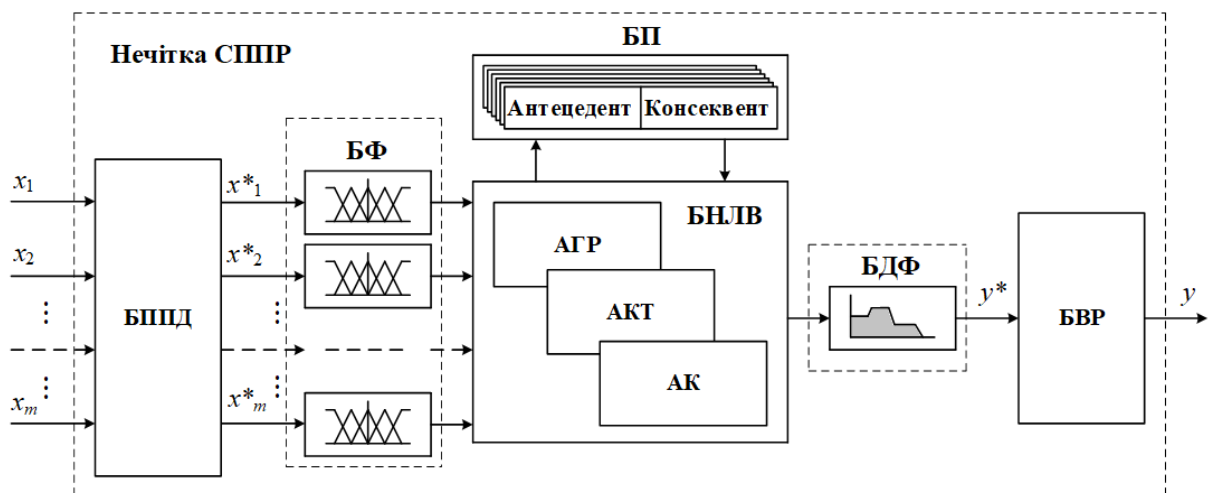


Рис. 1.7. Узагальнена структура нечіткої системи підтримки прийняття рішень

Наведена узагальнена нечітка СППР (рис. 1.7) формалізує взаємозв'язки вхідних та вихідної змінних на основі нелінійної залежності  $f_{\text{НС}}$  наступним чином

$$y = f_{\text{НС}}(x_1, x_2, \dots, x_m). \quad (1.9)$$

Блок початкової підготовки даних провадить попередню підготовку набору інформації для подальшого аналізу та визначає конкретні атрибути, за якими буде проводитися оцінка. Наприклад, у цьому блоку дані конвертуються у числовий формат та піддаються нормалізації шляхом вагомого чи порядкового розташування. Вагомість може бути призначена категоріальним атрибутам шляхом надання конкретних числових значень експертами. У випадку відсутності експертної оцінки початкові дані можуть бути впорядковані шляхом присвоєння кожному категоріальному атрибуту певного порядкового номеру. Крім того, деякі атрибути можуть бути виключені з розгляду у випадку, якщо немає можливості впорядкування або відсутні оцінки від експертів. Таким чином, БППД отримує на вхід змінні  $x_1, x_2, \dots, x_m$  та надає їх нормовані значення  $x_1^*, x_2^*, \dots, x_m^*$  на блок фаззифікації. У свою чергу, блок видачі рішення формує результуюче значення вихідної змінної  $y$  на основі розрахованого за допомогою БНЛВ та дефаззифікованого сигналу  $y^*$ .

У наведених нечітких системах керування та підтримки прийняття рішень можуть бути застосовані різні механізми нечіткого логічного виведення: Мамдані, Такагі-Сугено, Цукамото, Заде, Ларсена, Лукасевича, Гьоделя та ін. [5, 7, 8]. Далі розглянемо більш детально найбільш поширені механізми НЛВ на прикладі узагальненої нечіткої САК (рис. 1.6).

*Механізм НЛВ Мамдані.* Механізм нечіткого логічного виведення типу Мамдані складається з наступних послідовних етапів: фаззифікація, агрегація, активація, акумуляція та дефаззифікація [15, 16]. Фаззифікація представляє собою процес перетворення конкретних числових значень вхідних змінних на відповідні нечіткі терм-множини з обчисленням



ступенів належності. Наприклад, при застосуванні вектору (1.4) в якості вектору вхідних змінних НК, двох лінгвістичних термів для кожної вхідної змінної ( $A_1, A_2$  та  $B_1, B_2$ ) та функцій належності гаусового типу 1-го роду, поточні ступені належності для передумов правил будуть розраховуватися на основі залежностей (1.10)-(1.13)

$$\mu_{A_1}(k_P \varepsilon) = \exp \left[ \frac{-(k_P \varepsilon - x_{A_1})^2}{2\sigma_{A_1}^2} \right]; \quad (1.10)$$

$$\mu_{A_2}(k_P \varepsilon) = \exp \left[ \frac{-(k_P \varepsilon - x_{A_2})^2}{2\sigma_{A_2}^2} \right]; \quad (1.11)$$

$$\mu_{B_1}(k_D \dot{\varepsilon}) = \exp \left[ \frac{-(k_D \dot{\varepsilon} - x_{B_1})^2}{2\sigma_{B_1}^2} \right]; \quad (1.12)$$

$$\mu_{B_2}(k_D \dot{\varepsilon}) = \exp \left[ \frac{-(k_D \dot{\varepsilon} - x_{B_2})^2}{2\sigma_{B_2}^2} \right], \quad (1.13)$$

де  $x_{A_1}, \sigma_{A_1}, x_{A_2}, \sigma_{A_2}, x_{B_1}, \sigma_{B_1}, x_{B_2}, \sigma_{B_2}$  – регульовані параметри функцій належності  $A_1, A_2$  та  $B_1, B_2$ .

При цьому, продукційні правила бази правил для даної системи типу Мамдані задаються у вигляді [15]:

$$\text{IF } "k_P \varepsilon = A_1" \text{ AND } "k_D \dot{\varepsilon} = B_1" \text{ THEN } "u_{FC} = C_1", \quad (1.14)$$

де  $C_1$  – відповідний ЛТ вихідної змінної  $u_{FC}$ . У свою чергу, вираз (1.14) є першим правилом БП.

На етапі агрегації визначаються ступені істинності умов для кожного з правил у нечіткій системі [8]. Це досягається шляхом використання отриманих на етапі фаззифікації значень ФНЛТ змінних, які утворюють антецеденти нечітких продукційних правил. В свою чергу, знаходження ступенів належності антецедентів виконується за допомогою t-норми на основі операції «min» відповідно до виразу (1.15)

$$\mu_A(k_P \varepsilon) \cap \mu_B(k_D \dot{\varepsilon}) = \min[\mu_A(k_P \varepsilon), \mu_B(k_D \dot{\varepsilon})]. \quad (1.15)$$

У свою чергу, рівні зрізу (відсікання)  $\alpha_1$  та  $\alpha_2$  для умов правил знаходяться на основі виразів (1.16) та (1.17) відповідно

$$\alpha_1 = \min[A_1(k_p \varepsilon), B_1(k_D \dot{\varepsilon})] = A_1(k_p \varepsilon) \cap B_1(k_D \dot{\varepsilon}); \quad (1.16)$$

$$\alpha_2 = \min[A_2(k_p \varepsilon), B_2(k_D \dot{\varepsilon})] = A_2(k_p \varepsilon) \cap B_2(k_D \dot{\varepsilon}). \quad (1.17)$$

Етап активації представляє собою процес визначення ступеня істинності кожного з елементарних логічних умов (висловлювань), які утворюють консеквенти всіх нечітких продукційних правил [8, 10]. Наприклад, при застосуванні двох лінгвістичних термів  $C_1$  та  $C_2$  для вихідної змінної  $u_{FC}$  на даному етапі знаходяться усічені функції належності  $C_1^*(u_{FC})$  та  $C_2^*(u_{FC})$  для підзаключень правил за допомогою операції «min»:

$$C_1^*(u_{FC}) = \min[\alpha_1, C_1] = \alpha_1 \cap C_1; \quad (1.18)$$

$$C_2^*(u_{FC}) = \min[\alpha_2, C_2] = \alpha_2 \cap C_2. \quad (1.19)$$

На етапі акумуляції виконується об'єднання знайдених на попередньому етапі усічених функцій належності  $C_1^*(u_{FC})$  та  $C_2^*(u_{FC})$  для отримання підсумкової нечіткої підмножини  $\mu_\Sigma(u_{FC})$  вихідної змінної  $u_{FC}$  за допомогою операції «max» на основі виразу (1.20) [8]

$$\mu_\Sigma(u_{FC}) = \max[C_1^*, C_2^*] = C_1^* \cup C_2^*. \quad (1.20)$$

Дефазифікація представляє собою процес переходу від функції належності вихідної лінгвістичної змінної  $\mu_\Sigma(u_{FC}^*)$  до її чіткого числового значення  $u_{FC}$  [15, 16]. Для механізму Мамдані числове значення вихідного сигналу розраховується зазвичай за допомогою методу центру тяжіння на основі виразу (1.21)

$$u_{FC} = \frac{\int u_{FC}^* \mu_\Sigma(u_{FC}^*) du_{FC}^*}{\int \mu_\Sigma(u_{FC}^*) du_{FC}^*}. \quad (1.21)$$

*Механізм НЛВ Такагі-Сугено.* Механізм нечіткого логічного виведення Такагі-Сугено включає етапи фаззифікації, агрегації, активації та дефаззифікації. У цьому випадку етап акумуляції відсутній, оскільки на етапі активації вже отримані дискретні множини чітких значень для вихідної лінгвістичної змінної [17]. Етапи фаззифікації та агрегації виконуються таким же чином як і для НЛВ типу Мамдані. В свою чергу, рівні зрізу  $\alpha_1$  та  $\alpha_2$  для антецедентів правил знаходяться також за допомогою операції «min» на основі виразів (1.16) та (1.17). У механізмі Такагі-Сугено в якості консеквентів правил використовуються функції належності імпульсного типу, що представляють собою суми зважених миттєвих значень всіх вхідних змінних [17]. При цьому, правила БП для системи типу Такагі-Сугено задаються у вигляді:

$$\text{IF } "k_p \varepsilon = A_1" \text{ AND } "k_D \dot{\varepsilon} = B_1" \text{ THEN } "u_{FCr} = k_{1r}(k_p \varepsilon) + k_{2r}(k_D \dot{\varepsilon})", \quad (1.22)$$

де  $k_{1r}$ ,  $k_{2r}$  – вагові коефіцієнти правил БП;  $r$  – номер правила БП ( $r = 1, 2, \dots, s$ ).

На етапі активації для кожного з продукційних правил замість побудови функцій належності підзаключень в явному вигляді знаходиться чітке значення вихідної змінної  $u_{FCr}$  [17]. Таким чином, кожному  $r$ -му правилу ставиться у відповідність точка  $\alpha_r u_{FCr}$ .

Дефаззифікація являє собою перехід від дискретної множини чітких значень  $\{u_{FC1}, \dots, u_{FCr}, \dots, u_{FCs}\}$  для кожної лінгвістичної змінної до єдиного чіткого значення на основі дискретного методу центру тяжіння [17]. Дане чітке значення розраховується за допомогою виразу (1.23)

$$u_{FC} = \frac{\sum_{r=1}^s \alpha_r u_{FCr}}{\sum_{r=1}^s \alpha_r}, \quad (1.23)$$

де  $s$  – загальна кількість правил БП.

*Механізм НЛВ Цукамото.* В механізмі нечіткого логічного виведення Цукамото також відсутній етап акумуляції, подібно до

механізму Такагі-Сугено [8, 10]. Характерною особливістю цього механізму є використання монотонно спадаючих або зростаючих функцій у якості функцій належності вихідної змінної. Етапи фазифікації та агрегації, а також база правил є аналогічними до механізму Мамдані. На етапі активації для кожного з продукційних правил замість побудови функцій належності підзаключень вирішується рівняння  $\mu_c(u_{FC}) = \alpha$  і визначається чітке значення вихідної змінної системи  $u_{FC}$ . Наприклад, при застосуванні двох лінгвістичних термів  $C_1$  та  $C_2$  для вихідної змінної  $u_{FC}$  її чіткі значення  $u_{FC1}$  та  $u_{FC2}$  на даному етапі визначаються на основі виразів (1.24) та (1.25)

$$u_{FC1} = \frac{1}{\mu_{C1}(\alpha_1)}; \quad (1.24)$$

$$u_{FC2} = \frac{1}{\mu_{C2}(\alpha_2)}, \quad (1.25)$$

де  $\mu_{C1}$  та  $\mu_{C2}$  – функції належності лінгвістичних термів  $C_1$  та  $C_2$  відповідно.

Етап дефазифікації, в свою чергу, виконується таким же чином, як і в механізмі Такагі-Сугено, використовуючи дискретний метод центру тяжіння на основі виразу (1.23).

*Механізм НЛВ Ларсена.* У механізмі нечіткого логічного виведення Ларсена етапи фазифікації, агрегації, акумуляції та дефазифікації виконуються аналогічно до механізму Мамдані. Проте, етап активації реалізується за допомогою операції «prod» [8]. При застосуванні двох лінгвістичних термів  $C_1$  та  $C_2$  для вихідної змінної  $u_{FC}$  на етапі активації усічені функції належності  $C_1^*(u_{FC})$  та  $C_2^*(u_{FC})$  розраховуються на основі наступних виразів

$$C_1^*(u_{FC}) = \alpha_1 \cdot C_1; \quad (1.26)$$

$$C_2^*(u_{FC}) = \alpha_2 \cdot C_2. \quad (1.27)$$

*Механізм НЛВ типу-2.* Для більш ефективного та якісного врахування різноманітних невизначеностей в системах керування та прийняття рішень застосовують механізми нечіткого логічного виведення на основі нечітких множин типу-2, які були запропоновані N.N. Karnik, J.M. Mendel і Q. Liang [18, 19]. Зокрема, серед таких систем найбільш розповсюдженими є системи, які використовують інтервальні нечіткі множини типу-2. Ці системи відзначаються високою ефективністю у обробці інформації з невизначеністю та, в той же час, вимагають порівняно невеликих обчислювальних витрат [20, 21]. Спрощена структура механізму нечіткого логічного виведення на основі інтервальних нечітких множин типу-2 для вищенаведеної узагальненої нечіткої системи керування (рис. 1.6) наведена на рис. 1.8. Цей механізм містить наступні складові: блок фаззифікації, базу продукційних правил, блок нечіткого виведення, що виконує основні операції з нечіткими множинами типу-2, блок пониження типу результуючої вихідної нечіткої множини та блок дефаззифікації [20].

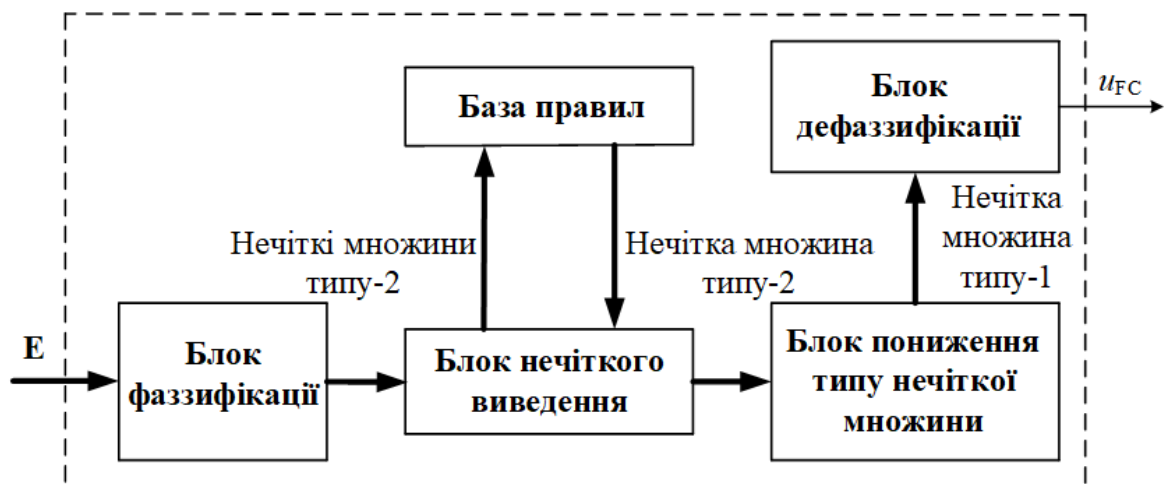


Рис. 1.8. Структура механізму НЛВ на базі інтервальних нечітких множин типу-2

На етапі фаззифікації поточна ступінь належності для першої змінної вхідного вектору  $E$  в загальному вигляді розраховуються на основі виразу (1.28) [20, 21]

$$\mu_{\tilde{A}}(k_p \varepsilon) = \int_{v \in F_{k_p \varepsilon}} f_{k_p \varepsilon}(v) / v = \int_{v \in F_{k_p \varepsilon}} 1/v, F_{k_p \varepsilon} \subseteq [0,1], \quad (1.28)$$

де  $F_{k_p \varepsilon}$  – первинна належність змінної  $k_p \varepsilon$  до інтервальної нечіткої множини типу-2  $\tilde{A}$ , яка, в свою чергу, є областю визначення вторинної функції належності;  $f_{k_p \varepsilon}(v)$  – вторинна ступінь належності,  $f_{k_p \varepsilon}(v) = 1$ , так як вона є інтервальною нечіткою множиною типу-2.

Наприклад, при використанні гаусових функцій належності для фазифікації вхідних змінних у системах, що базуються на інтервальних нечітких множинах типу-2, наведені функції належності можуть бути визначені двома способами: 1) функціями з невизначеним центром та визначеним відхиленням; 2) функціями з визначеним центром та невизначеним відхиленням [21, 22]. В свою чергу, для першої змінної вхідного вектору  $\mathbf{E}$  поточні ступені належності для першого та другого способу задання розраховуються на основі виразів (1.29) та (1.30)

$$\mu_{\tilde{A}}(k_p \varepsilon) = \exp \left[ \frac{-\left(k_p \varepsilon - \left[x_{\tilde{A}1}, x_{\tilde{A}2}\right]\right)^2}{2\sigma_{\tilde{A}}^2} \right]; \quad (1.29)$$

$$\mu_{\tilde{A}}(k_p \varepsilon) = \exp \left[ \frac{-(k_p \varepsilon - x_{\tilde{A}})^2}{2\left(\left[\sigma_{\tilde{A}1}, \sigma_{\tilde{A}2}\right]\right)^2} \right], \quad (1.30)$$

де  $x_{\tilde{A}1}, x_{\tilde{A}2}$  – межі інтервалу невизначеності центру ФН;  $\sigma_{\tilde{A}1}, \sigma_{\tilde{A}2}$  – межі інтервалу невизначеності відхилення ФН.

Таким же чином визначаються поточні ступені належності для другої змінної вектору входів  $\mathbf{E}$ .

На рис. 1.9 представлені графічні зображення наведених типів ФН.

Продукційні правила БП для розглянутої системи на основі інтервальних нечітких множин типу-2 задаються у вигляді [20, 21]:

$$\text{IF } "k_p \varepsilon = \tilde{A}_1" \text{ AND } "k_D \dot{\varepsilon} = \tilde{B}_1" \text{ THEN } "u_{FC} = \tilde{C}_1", \quad (1.31)$$

де  $\tilde{A}_1, \tilde{B}_1, \tilde{C}_1$  – відповідні ЛТ вхідних та вихідної змінних інтервальної нечіткої системи типу-2.

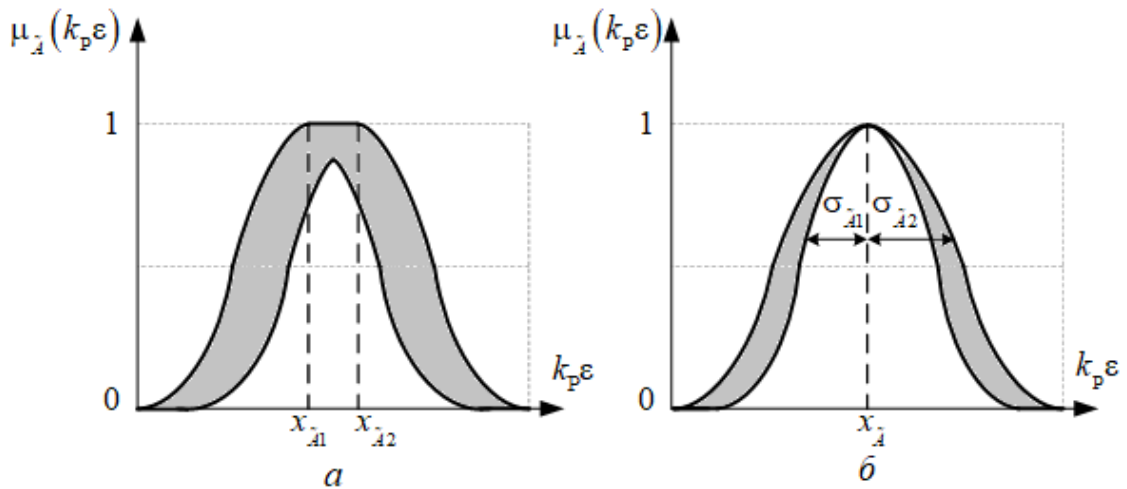


Рис. 1.9. Графічні зображення ФН гаусового типу: а) з невизначеним центром та визначеним відхиленням; б) з визначеним центром та невизначеним відхиленням

На етапі нечіткого виведення проводяться основні операції з вхідними нечіткими множинами типу-2 та отримується результуюча активізована вихідна множина типу-2 [20-22]. Надалі, на етапах пониження типу результуючої вихідної нечіткої множини та дефазифікації отримується вихідна нечітка множина типу-1 та відповідне чітке значення сигналу  $u_{FC}$ , використовуючи метод центру множин Карніка-Менделя [23].

*Механізм фрактального НЛВ.* Фрактальні (дробові) нечіткі системи є нечіткими системами нового покоління, які були запропоновані М. Mazandarani та L. Xiu [24]. Дані системи базуються на фрактальному НЛВ та при розв'язанні певних задач дозволяють досягати кращих результатів у порівнянні з традиційними нечіткими системами. Механізм фрактального НЛВ ґрунтується на використанні дробових горизонтальних функцій належності для визначення консеквентів правил БП та дробових композиційних правил виведення [25-28]. В свою чергу, дробові

горизонтальні функції належності характеризуються певним дробовим індексом, який може бути сталим або змінюватися з часом [24, 28]. Також, цей механізм може застосовуватися в нечітких САК та СППР різних типів замість інших відомих НЛВ (Мамдані, Такагі-Сугено та ін.) без будь-яких змін у вхідних та вихідних змінних, структурі баз правил та у загальній структурі системи. Результатом є те, що механізм фрактального нечіткого логічного виведення ураховує не лише ступені істинності включеної інформації у ФН, але й обсяг цієї інформації у процесі нечіткого виведення. Також, слід зазначити, що обсяг інформації у функціях належності залежить від її ступеня істинності (чим вищий ступінь істинності, тим більший обсяг інформації використовується у процесі нечіткого виведення) [24].

У підсумку цього підрозділу можна зазначити, що нечіткі системи мають наступні переваги:

- висока ефективність при вирішенні завдань керування, прийняття рішень, ідентифікації, прогнозування на ін. в умовах неповноти інформації або невизначеності;
- можливість створення моделей складних об'єктів та процесів без використання розрахунків на основі диференціальних та інтегральних рівнянь;
- здатність опрацьовувати вхідні дані, які визначені нечітко;
- можливість використання нечітких критеріїв оцінки та порівняння.

До недоліків систем на основі нечіткої логіки можна віднести певну складність їх проектування та налаштування. В багатьох випадках для вирішення задач складання продукційної бази правил виникає необхідність залучення висококваліфікованих експертів-фахівців. Також, певні труднощі виникають при виборі оптимальної кількості лінгвістичних термів для кожної вхідної та вихідної змінної, визначенні типів та параметрів функцій належності, а також процедур агрегації, активації,



акумуляції і методу дефазифікації. Дана проблема може вирішуватись за допомогою прогресивних методів автоматизованого проєктування, які містять процедури структурно-параметричної оптимізації на основі цільових функцій та можуть використовувати знання експертів для формування попередніх гіпотез про структуру та параметри нечіткої системи.

## 1.2 Штучні нейронні мережі

*Загальні відомості теорії штучних нейронних мереж.* Теорія штучних нейронних мереж є ключовою складовою сучасних областей штучного інтелекту та машинного навчання. Ця теорія базується на математичних моделях, які дозволяють відтворювати роботу біологічних нейронних систем. В основі неї лежить ідея використання великої кількості простих обчислювальних елементів (нейронів), які взаємодіють між собою, щоб вирішувати складні завдання. При цьому, кожен нейрон в мережі приймає вхідні сигнали, зважує їх, застосовує активаційну функцію та генерує вихідний сигнал. Ці сигнали подаються на вхід наступним шарам нейронів, що дозволяє інформації проходити через мережу та вирішити поставлену задачу.

Нейронні мережі виявилися надзвичайно ефективними в таких областях, як розпізнавання образів, природно-мовній обробці, автоматичне керування, прогнозування та багатьох інших [29]. Вони використовуються у різноманітних сферах, включаючи високотехнологічне виробництво, фінансове прогнозування, медичні дослідження та ін. [29-31]. З плином часу та з розвитком обчислювальних технологій, нейронні мережі стали більш потужними та ефективними, що дозволило вирішувати більш складні завдання. Вони є основою для багатьох інноваційних застосунків, які полегшують та поліпшують життя людей.

Штучні нейронні мережі – це обчислювальні системи, що за структурою, функціонуванням та принципами перетворення інформації являють собою спрощені математичні та комп'ютерні моделі біологічних нейронних мереж (мереж нервових клітин живих організмів).

У свою чергу, базовими елементами нейромереж живих організмів є специфічні нервові клітини – нейрони, які окрім низки інших функцій (прийом, обробка, передача електрохімічних сигналів тощо) мають здатність запам'ятовувати інформацію і застосовувати попередній досвід до подальших дій [29]. Так, кора головного мозку людини містить близько  $10^{11}$  нейронів, які між собою пов'язані за допомогою  $10^{14} \dots 10^{15}$  взаємозв'язків (кожен окремий нейрон зв'язаний з  $10^3 \dots 10^4$  іншими нейронами) [29, 32].

При цьому, кожен окремий нейрон є досить складним елементом, має різні підсистеми та механізми управління, що передають інформацію через велику кількість електрохімічних зв'язків. Загалом, налічують біля сотні різних класів нейронів, що дозволяє біологічним нейромережам здійснювати недвійковий, нестійкий та несинхронний обчислювальний процес, що суттєво відрізняється від процесів обчислень традиційних комп'ютерів. Потужність людського розуму залежить від кількості нейронів, різноманіття з'єднань між ними, а також від ефективності їх навчання. У свою чергу, штучні нейромережі моделюють лише головні елементи природного мозку, що надихає науковців продовжувати дослідження у напрямку створення нових моделей природного інтелекту.

Спрощена структура біологічного нейрону наведена на рис. 1.10 [29]. Біологічний нейрон складається з тіла клітини – соми (soma), і двох типів зовнішніх деревоподібних відгалужень: дендритів (dendrites) і аксона (axon). Тіло клітини вміщує ядро (nucleus), яке містить інформацію про властивості нейрона, і плазму, що продукує необхідні для нейрона матеріали. Нейрон отримує сигнали (імпульси) від інших нейронів через дендрити (приймачі) і передає сигнали, що згенеровані тілом клітини,

через аксон (передавач). Аксон наприкінці розгалужується на волокна, на закінченнях яких знаходяться синапси (synapses). У свою чергу, синапс є функціональним вузлом між двома нейронами (волокно аксона одного нейрона і дендрит іншого). Коли імпульс досягає синаптичного закінчення, то в ньому продукуються хімічні речовини – нейротрансмітери. Нейротрансмітери проходять через синаптичну щілину і в залежності від типу синапсу, підсилюють або послаблюють здатність нейрона-приймача генерувати електричні імпульси. Результативність синапсу налаштовується сигналами, що проходять скрізь нього, тому синапси навчаються у відповідності до активності процесів, у яких вони приймають участь.

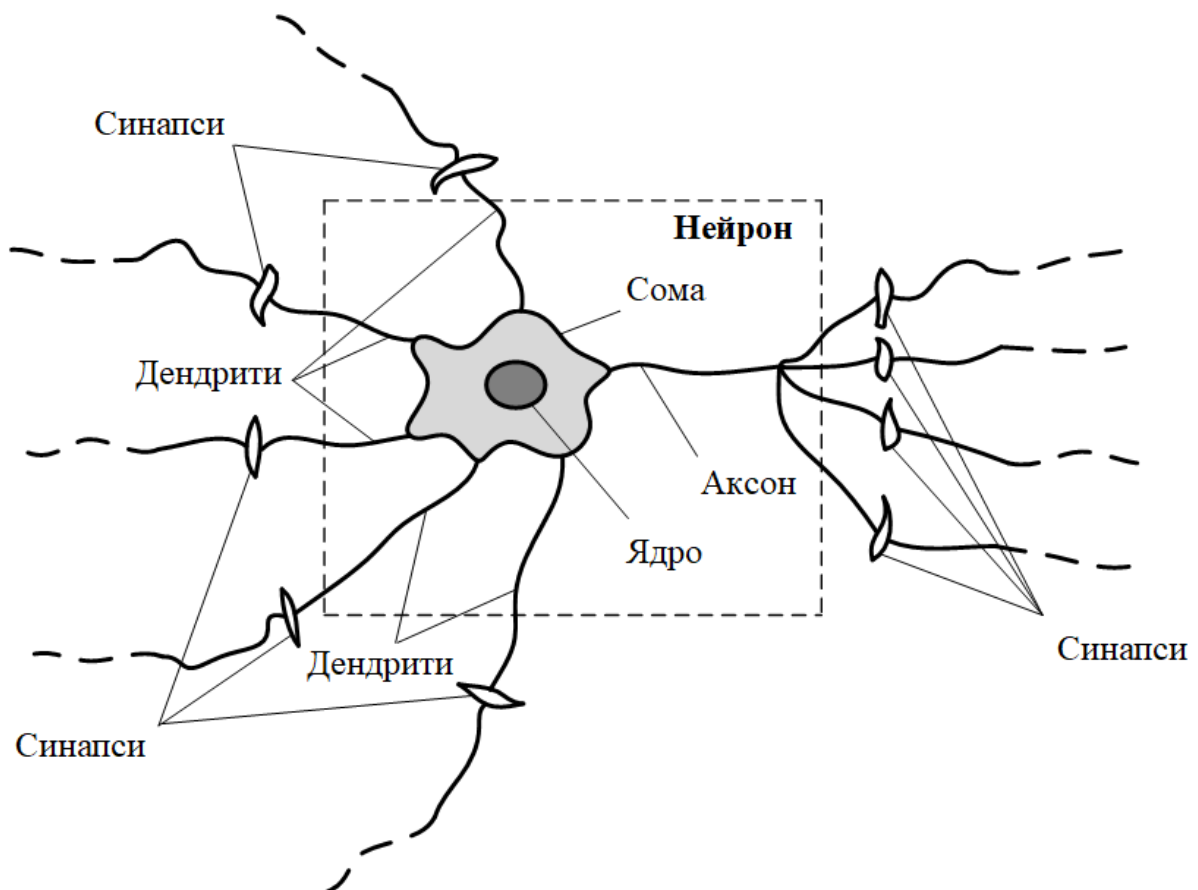


Рис. 1.10. Спрощена структура біологічного нейрона

Реальні біологічні нейрони є значно складнішими за своєю структурою, ніж штучні нейрони, проте більшість штучних нейромереж моделюють лише наведені прості властивості.

Штучний нейрон є базовим обчислювальним модулем штучних нейронних мереж. Він має структуру, наведену на рис. 1.11, і моделює основні обчислювальні функції природного нейрона.

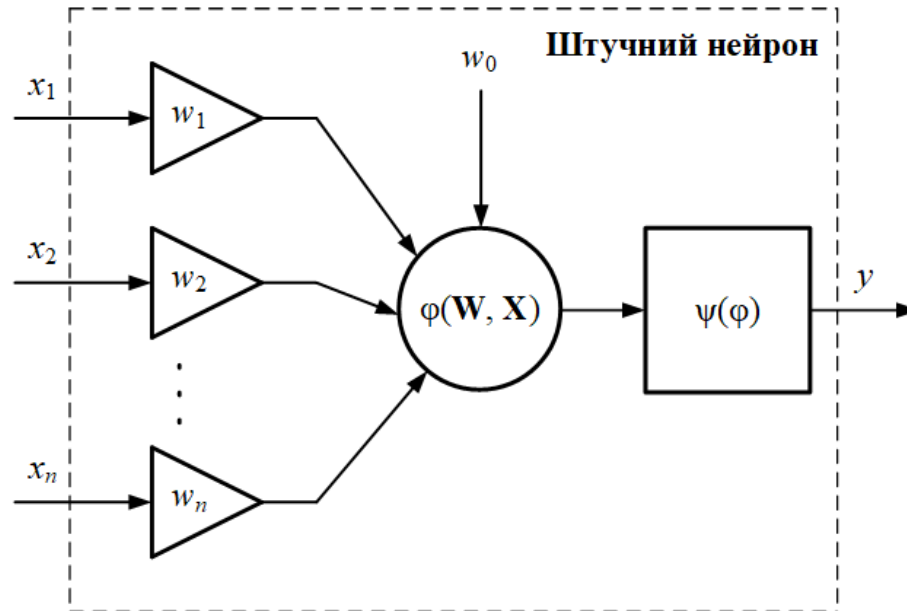


Рис. 1.11. Загальна структура штучного нейрона

Даний елемент має декілька входів і один вихід. При цьому, на входи подаються сигнали  $x_1, x_2, \dots, x_n$ , які утворюють вхідний вектор  $\mathbf{X}$  з розмірністю  $n$ . Дані вхідні сигнали зважуються за допомогою синаптичних вагових коефіцієнтів  $w_1, w_2, \dots, w_n$  (разом із зсувом  $w_0$  утворюють вектор  $\mathbf{W}$  з розмірністю  $n + 1$ ), що моделює процес передачі сигналів у синапсах біологічного нейрону. Після цього, зважені змінні вектора  $\mathbf{X}$  (вхідні сигнали нейрона) послідовно проходять через дискримінантну функцію  $\varphi(\mathbf{W}, \mathbf{X})$  та функцію активації  $\psi(\varphi)$ , в результаті чого обчислюється вихідний сигнал нейрона  $y$ . Це імітує перетворення сигналів всередині тіла (соми) біологічного нейрона. Отже, штучний нейрон моделює перетворення інформації у біологічному нейроні за допомогою скалярної функції векторного аргумента (1.32)

$$y = \psi(\varphi(\mathbf{W}, \mathbf{X})). \quad (1.32)$$

Таким чином, можна сказати, що штучний нейрон є пристроєм, що зберігає інформацію у вигляді набору вагових коефіцієнтів  $\mathbf{W}$ , які представляють його «пам'ять». Також, його можна розглядати як базовий обчислювальний елемент нейронної мережі, який виконує операції на основі вхідних сигналів, їхніх відповідних ваг та функції активації.

Зазвичай, в якості дискримінантної функції  $\varphi(\mathbf{W}, \mathbf{X})$  нейронів застосовується зважена сума:

$$\varphi(\mathbf{W}, \mathbf{X}) = \sum_{j=1}^n w_j x_j + w_0, \quad (1.33)$$

де  $j$  – порядковий номер вхідної змінної нейрона.

У деяких нейромережах, також, можуть бути застосовані й інші види дискримінантної функції, такі як зважений добуток, евклідова відстань тощо [29].

У свою чергу, в якості функції активації можуть бути застосовані різні математичні функції, найбільш поширені з яких:

– порогова функція Хевісайда (приймає значення в діапазоні від 0 до 1)

$$\psi(\varphi) = \begin{cases} 1, & \text{при } \varphi \geq 0; \\ 0, & \text{при } \varphi < 0; \end{cases} \quad (1.34)$$

– порогова знакова (приймає значення в діапазоні від  $-1$  до  $1$ )

$$\psi(\varphi) = \begin{cases} 1, & \text{при } \varphi \geq 0; \\ -1, & \text{при } \varphi < 0; \end{cases} \quad (1.35)$$

– сигмоїдна логістична (приймає значення в діапазоні від  $0$  до  $1$ )

$$\psi(\varphi) = \frac{1}{1 + e^{-c\varphi}}, \quad (1.36)$$

де  $c$  – константа;

– гіперболічний тангенс (приймає значення в діапазоні від  $-1$  до  $1$ )

$$\psi(\varphi) = \tanh(c\varphi) = \frac{e^{c\varphi} - e^{-c\varphi}}{e^{c\varphi} + e^{-c\varphi}}; \quad (1.37)$$

– лінійна (приймає значення в діапазоні від  $-\infty$  до  $+\infty$ )

$$\psi(\varphi) = c\varphi; \quad (1.38)$$

– лінійна біполярна з насиченням (приймає значення в діапазоні від – 1 до 1)

$$\psi(\varphi) = \begin{cases} 1, & \text{при } \varphi > a_2; \\ c\varphi, & \text{при } a_1 \leq \varphi \leq a_2; \\ -1, & \text{при } \varphi < a_1, \end{cases} \quad (1.39)$$

де  $a_1$  та  $a_2$  – константи;

– радіально-базисна (приймає значення в діапазоні від 0 до  $+\infty$ )

$$\psi(\varphi) = e^{-c\varphi^2}. \quad (1.40)$$

Також, окрім представлених функцій можуть використовуватися лінійна уніполярна з насиченням, синусоїдальна, косинусоїдальна та ін. [29, 32, 33].

Представлений на рис. 1.11 штучний нейрон дозволяє на своїй основі будувати різні типи нейронних мереж. Однією з найбільш простих нейромереж є одношаровий перцептрон, який містить всього один нейрон зі структурою, наведеною на рис. 1.11, та певний механізм його навчання. В якості дискримінантної функції в одношаровому перцептроні використовується зважена сума (1.33), а в якості функції активації може бути застосовано декілька різних функцій, в залежності від його типу. Так, порогова функція активації (1.34) застосовується у дискретних одношарових перцептронах, сигмоїдна функція (1.36) – у дійсних перцептронах [32]. У свою чергу, при застосуванні лінійної функції активації (1.38) одношаровий перцептрон можна назвати ADALINE (ADaptive LInear NEuron) [29]. Дискретні та дійсні перцептрони можуть використовуватися переважно для розпізнавання образів та класифікації, а мережа ADALINE дозволяє вирішувати задачі адаптивної фільтрації.

Одношарові перцептрони є одними з перших запропонованих нейромереж і на сьогоднішній день мають досить обмежене застосування.

Більші обчислювальні можливості та, відповідно, більш складну структуру мають багатошарові перцептрони.

Багатошаровий перцептрон є багатошаровою нейронною мережею прямого поширення сигналу, тобто мережею, в якій нейрони організовані у шари, а сигнали передаються в одному напрямку – від вхідного шару, який сприймає зовнішній сигнал, до вихідного, що формує загальний вихідний сигнал. Кожен нейрон у внутрішньому шарі отримує сигнали тільки від нейронів попереднього шару, обробляє їх і передає тільки нейронам наступного шару. Крім того, зв'язки між нейронами всередині кожного шару відсутні. Всі внутрішні шари нейронів, що знаходяться між вхідним та вихідним шарами прийнято називати «прихованими». Таким чином, багатошаровий перцептрон повинен містити як мінімум один прихований шар.

Узагальнена структура багатошарового перцептрона наведена на рис. 1.12.

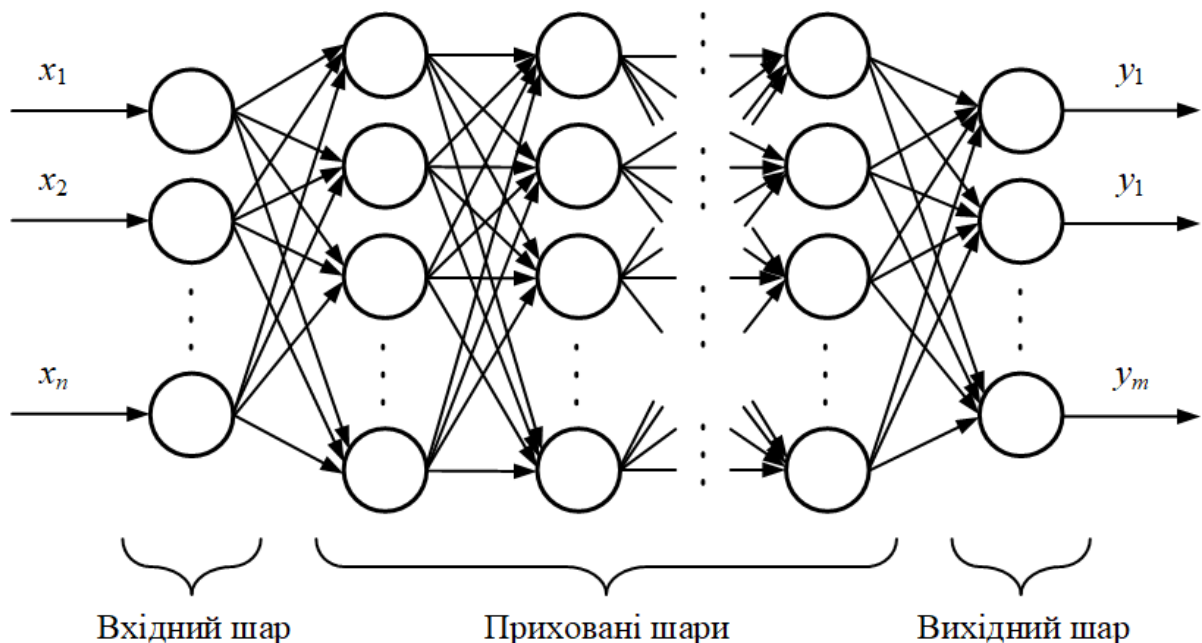


Рис. 1.12. Узагальнена структура багатошарового перцептрона

Представлена нейронна мережа (багатошаровий перцептрон) має  $n$  вхідних сигналів,  $t$  вихідних сигналів та  $M$  шарів з нейронами. У свою

чергу, кожен  $\mu$ -й шар ( $\mu = 1, \dots, M$ ) містить  $N_\mu$  нейронів. Вхідні сигнали  $x_1, x_2, \dots, x_n$  надходять тільки на входи нейронів першого шару ( $\mu = 1$ ). Кожен нейрон  $\mu$ -го шару з'єднаний зі всіма нейронами попереднього шару ( $\mu - 1$ ) та всіма нейронами наступного шару ( $\mu + 1$ ). Вихідний сигнал кожного нейрону мережі обчислюється на основі його вхідних сигналів за допомогою дискримінантної функції та функції активації. В якості дискримінантних функцій нейронів, зазвичай, використовуються функції зваженої суми (1.33), а в якості функцій активації, в залежності від задачі нейромережі, можуть бути застосовані сигмоїдні функції (1.36) або функції гіперболічного тангенсу (1.37). Також, в деяких випадках, для нейронів вихідного шару доцільним є використання лінійних функцій активації (1.38) або (1.39).

Загальна кількість вагових коефіцієнтів разом із зсувами  $N_w$  багатошарового персептрона визначається на основі залежності (1.41)

$$N_w = \sum_{\mu=2}^M N_\mu (N_{\mu-1} + 1) + n. \quad (1.41)$$

Ефективність багатошарового персептрона та якість отриманих рішень суттєво залежить від загальної кількості його шарів  $M$ , кількості нейронів  $N_\mu$  у кожному  $\mu$ -му шарі, типів функцій активації, розміру навчальної вибірки та обраного методу навчання.

Далі розглянемо загальні властивості штучних нейронних мереж.

*Основні властивості штучних нейронних мереж.* Головною перевагою штучних нейронних мереж є те, що вони надають універсальний метод розв'язання складних нетрадиційних завдань. Нейромережі можуть вирішувати широкий спектр завдань, і це особливо цінно, оскільки немає потреби розробляти окремі ЕОМ для кожного типу завдання. Крім того, використання нейромереж передбачає навчання, а не програмування. Програмісту необхідно лише створити навчальні сценарії та вказати нейромережі, яким чином до них пристосуватися. Тобто виконати функції вчителя, що формує освітнє середовище для нейронної



мережі. Крім того, нейромережі демонструють високу ефективність у тих ситуаціях, де потрібно враховувати людську інтуїцію, що складно формалізувати.

Також, штучні нейронні мережі мають наступні властивості:

– можливість узагальнення даних, що досягається за допомогою структури та принципів функціонування нейронних мереж. Їхні можливості у цьому відношенні базуються на здатності розпізнавати та аналізувати складні залежності в наборах даних. Це дозволяє підтримувати високий рівень точності у випадках, коли інші традиційні методи справляються із завданням менш ефективно. Однією з ключових особливостей нейронних мереж є їхня здатність виявляти внутрішні зв'язки та закономірності в наборах даних, навіть якщо ці залежності неочевидні або не виражені математичними функціями. Це особливо корисно у ситуаціях, коли важко апроксимувати функціональні зв'язки за допомогою аналітичних методів. Крім того, нейронні мережі можуть працювати з дуже великими наборами даних та враховувати навіть тонкі деталі в інформації. Це дає їм перевагу у ситуаціях, коли потрібно обробити великі обсяги інформації та виявити складні взаємозв'язки серед даних;

– можливість відтворення складних нелінійних взаємозалежностей, що реалізується за допомогою нелінійних функцій активації нейронів, та є особливо важливим при вирішенні задач автоматичного керування складними об'єктами, які мають суттєві нелінійності;

– адаптивність, що в контексті нейронних мереж означає їхню здатність до зміни свого внутрішнього стану та параметрів відповідно до змін у навколишньому середовищі. Це включає в себе здатність мережі реагувати на різноманітні сигнали з навколишнього середовища та змінювати свої внутрішні параметри для ефективного вирішення завдань. Зокрема, нейронні мережі можуть відстежувати зміни у вхідних сигналах та адаптуватися до них, щоб надавати точні та надійні відповіді. Ця

адаптивність забезпечує нейромережам можливість ефективно вирішувати різноманітні завдання у різних умовах і сценаріях. Наприклад, у випадку змінюючогося вхідного середовища або вимог до рішення завдань, нейронна мережа може автоматично адаптувати свою структуру та параметри, щоб забезпечити оптимальні результати. Це важливий аспект їхньої ефективності та придатності до вирішення різноманітних завдань у реальних умовах;

– універсальна апроксимація, що виражається у здатності нейромереж наближати будь-яку функцію з деякого набору входів до будь-якого рівня точності. Ця властивість нейронних мереж є теоретично обґрунтованою та доведеною, що робить їх невід'ємним інструментом в численних областях, де потрібно моделювати складні, нелінійні залежності. Універсальність у наближенні функцій дозволяє нейронним мережам вирішувати широкий спектр завдань, включаючи розпізнавання образів, прогнозування, класифікацію та багато інших. Це дозволяє застосовувати їх у великому ряді напрямків, таких як обробка сигналів, комп'ютерне бачення, фінанси, медична діагностика та багато інших;

– точність, що визначається здатністю нейронних мереж адекватно моделювати складні та нелінійні залежності в даних. Наприклад, в задачах розпізнавання образів, де об'єкти можуть мати різні форми, кольори та розміри, нейронні мережі можуть досягти дуже високої точності завдяки їхній здатності виявляти та аналізувати складні взаємозв'язки. Більш того, з використанням методів навчання, нейронні мережі можуть адаптуватися до нових даних та умов, що робить їх вельми гнучкими та потужними інструментами для рішення різноманітних завдань. Навіть у задачах де деяка ступінь неточності припустима, нейронні мережі можуть бути налаштовані таким чином, щоб забезпечити оптимальний баланс між точністю та ресурсами, що витрачаються на обробку даних та навчання;

– здатність до навчання, що дозволяє нейромережам удосконалювати своє функціонування та підвищувати точність за рахунок налаштування

синаптичних вагових коефіцієнтів, використовуючи методи навчання та навчальні приклади для вирішення певної поставленої задачі;

– однорідність складових елементів (нейронів), їх обчислювальних операцій (додавання, множення, обчислення нелінійної функції) та об'єднуючих зав'язків, що є ключовим аспектом побудови нейромереж, а також дозволяє їм досягти технологічну простоту програмно-апаратної реалізації;

– розподіленість пам'яті, що реалізується у нейромережах через розподілення інформації між багатьма адресами. Кожен елемент даних виражається у вигляді активності, яка розподіляється між численними обчислювальними елементами. Кожен такий елемент бере участь у відтворенні багатьох різних даних. У традиційних комп'ютерних системах використовується локальне подання, коли кожен елемент даних прив'язаний до окремого обчислювального елемента. У той час як у нейромережах з розподіленою архітектурою інформацію можна поділити на частини для подальшої обробки;

– здатність отримувати знання з даних, що дозволяє нейронним мережам в процесі навчання вміщувати або апроксимувати загальні закономірності, які містяться в навчальних даних. Це дозволяє видобувати навіть ті знання, які для людини можуть бути неявними;

– стійкість до відмов окремих складових елементів, що дозволяє навіть в разі відмови одного або кількох елементів нейромережі продовжувати надійно працювати без суттєвого впливу на її загальну продуктивність.

Далі розглянемо класифікацію штучних нейронних мереж за основними ознаками.

*Класифікація штучних нейронних мереж.* Зазвичай класифікацію нейромереж проводять за наступними основними ознаками.

За типом вхідних сигналів розрізняють мережі з аналоговими сигналами, які використовують інформацію у вигляді дійсних чисел, та

мережі з бінарними сигналами, що оперують інформацією у двійковому вигляді.

За топологією зв'язків розрізняють повнозв'язні (інтерактивні), неповнозв'язні (шаруваті, багат шарові, ієрархічні) та слабозв'язні (з локальними зв'язками) нейронні мережі (рис. 1.13).

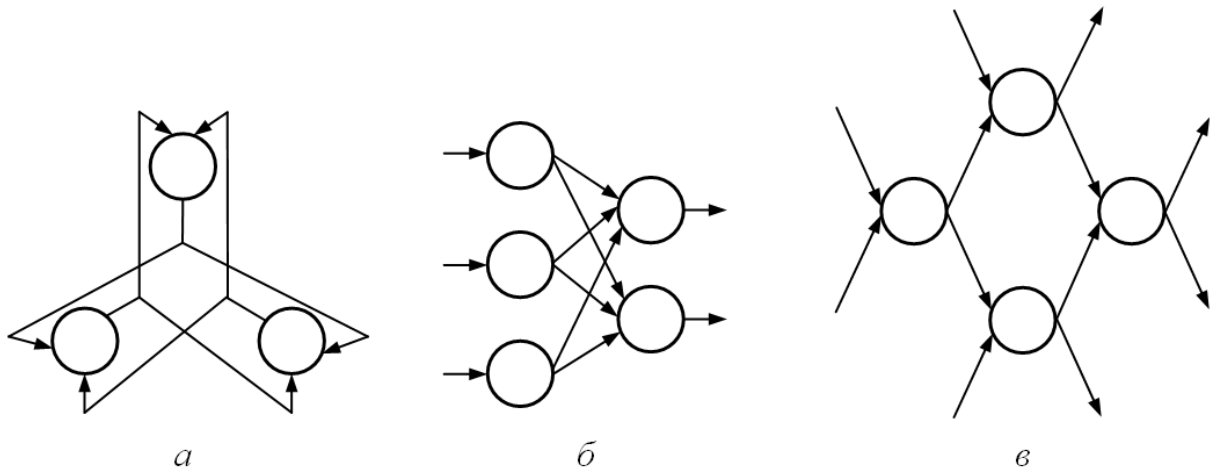


Рис. 1.13. Топологія зв'язків нейромереж: а) повнозв'язна; б) неповнозв'язна; в) слабозв'язна

У повнозв'язних кожен нейрон має зв'язки з кожним іншим нейроном у мережі, і передає свій вихідний сигнал всім іншим нейронам, включаючи самого себе. У неповнозв'язних мережах, крім основних зв'язків, що з'єднують різні шари нейронів, присутні також і бічні зв'язки, які охоплюють нейрони в межах одного шару. Деякі нейрони мають додаткові зовнішні входи, які утворюють рецепторні поля. Нейрони першого шару приймають вхідні сигнали, обробляють їх і, через точки галуження, передають нейронам наступного шару, і так далі до останнього шару, який формує вихідні сигнали. В цьому випадку інформація передається від вхідного до вихідного шару, і кожен наступний шар забезпечує більш високий рівень обробки, ніж попередній. Слабозв'язні мережі (або мережі з локальними зв'язками) організовані у вузлах

прямокутної або гексагональної решітки, де кожен нейрон має зв'язки лише з чотирма, шістьма або вісьмома своїми найближчими сусідами.

За типом міжнейронних зв'язків поділяють на мережі без циклів (ациклічні мережі) та мережі з циклами, які можуть бути рівноважними мережами з циклами і мережами з обмеженими циклами.

За характером зв'язків виділяють два типи мереж: мережі прямого поширення (feedforward), до яких відсутні зворотні зв'язки, та рекурентні мережі (feedforward / feedback), які мають зворотні зв'язки, та в яких інформація поширюється як вперед, так і назад між різними шарами мережі.

За типом структур нейронів можна виділити дві категорії: гомогенні (однорідні) нейронні мережі, які складаються з нейронів одного типу та які мають однакові функції активації, та гетерогенні (неоднорідні) нейронні мережі, що включають нейрони різних типів з різними функціями активації.

За кількістю шарів нейронів можна виділити одношарові нейронні мережі, які складаються з одного шару нейронів, та багатшарові нейронні мережі, що включають більше одного шару взаємопов'язаних нейронів.

За типом дискримінантної функції нейронів можна виділити дві категорії мереж: зважені мережі, де присутні ваги зв'язків, та мережі без ваг зв'язків.

За типом функцій активації нейронів розрізняють три типи мереж: неперервні (аналогові, дійсні, диференційовані) мережі, в яких кожен компонент має неперервну та диференційовану функцію; дискретні (бінарні, порогові, недиференційовані) мережі, нейрони яких оперують недиференційованими функціями, що можуть бути бінарними або пороговими; дискретно-неперервні мережі, що включають в себе елементи з як неперервними, так і дискретними функціями активації.

За характером навчання розрізняють контрольоване, неконтрольоване та змішане навчання. Контрольоване (піднаглядне, з

вчителем, з супервізором) навчання передбачає навчання, коли вихідний простір рішень нейромережі відомий. Під час навчання даного типу порівнюються відомі правильні відповіді з виходами мережі. Неконтрольоване (без нагляду, без вчителя, без супервізора) навчання представляє собою випадок, коли нейромережа формує вихідний простір рішень тільки на основі вхідних впливів, та навчається, не знаючи заздалегідь правильних вихідних значень. Проте вона групує «близькі» вхідні вектори так, щоб вони формували один і той же вихід мережі. Цей метод використовується, наприклад, при вирішенні задачі кластеризації. При застосуванні змішаного навчання випадку частина ваг визначається при спостереженні, а частина – при неспостережному навчанні. Навчання відбувається шляхом пред'явлення прикладів, що складаються з наборів вхідних даних разом з відповідними результатами.

За методом навчання розрізняють нейромережі: з методом зворотного поширення помилки, з застосуванням правила Хебба, з конкурентним навчанням, з неітеративним навчанням, з гібридним навчанням, що комбінує декілька різних методів навчання, з еволюційним навчанням та ін.

За сферами використання виділяються мережі для: аналізу сигналів, обробки мовлення, обробки та класифікації зображень і тексту, технічної та медичної діагностики, створення моделей в природничих науках і техніці, прогнозування соціально-економічних процесів, керування в технічних та економічних системах, створення інформаційних систем для пошуку та аналізу даних, а також в застосуваннях криптографії.

Для успішного застосування нейронних мереж з метою вирішення вищенаведених задач, спочатку необхідно здійснити їх ефективний синтез та навчання.

*Синтез та навчання штучних нейронних мереж.* Задача синтезу та навчання нейронної мережі зводиться до визначення таких її структури та параметрів, при яких буде досягнуто оптимального значення певного

попередньо обраного критерія якості. Наприклад, при використанні навчання з учителем в процесі створення нейромережі, критерієм якості  $E$  може виступати середньоквадратична помилка (Mean Squared Error – MSE)

$$E = \frac{1}{k} \sum_{i=1}^k (y_i^D - y_i^R)^2 \rightarrow \min, \quad (1.42)$$

де  $y_i^D$  – бажане значення вихідної змінної нейромережі при певних значеннях її входів, що визначається навчальною вибіркою;  $y_i^R$  – реальне значення вихідної змінної, що видає нейромережа при певних значеннях її входів;  $i$  – номер набору даних (строки) навчальної вибірки, що подається на нейромережу;  $k$  – загальна кількість наборів даних навчальної вибірки.

Першим етапом створення нейронної мережі, як правило, є її структурний синтез, який включає в себе вибір кількості вхідних та вихідних змінних, топології нейронної мережі, кількості шарів та нейронів у кожному шарі, типів зв'язків, дискримінантних функцій та функцій активації. Кількість вхідних та вихідних змінних, зазвичай, обирається в залежності від особливостей задачі, яка буде розв'язуватися за допомогою синтезованої мережі. В деяких випадках кількість змінних може бути зменшена при якісному відборі тільки тих змінних, які несуть у собі достатню для побудови нейромережі інформацію. Це дозволяє певним чином спростити процес синтезу та подальшої програмно-апаратної реалізації.

Щодо інших задач структурного синтезу (визначення топології нейромережі, кількості шарів та нейронів, а також типів зв'язків та функцій), на сьогоднішній день немає загальноприйнятих стандартних методів, які мали б тверду теоретичну основу для їх вирішення. Цей етап є ключовим у процесі синтезу нейромереж, і його втілення не супроводжується чіткими математичними директивами. Тому для його успішної реалізації необхідна спільна праця експерта у відповідній галузі та фахівця з розробки нейромереж, спрямована на максимально глибоке

розуміння процесів та зв'язків досліджуваних об'єктів та можливостей їх коректного відтворення. У багатьох випадках для вирішення наведених задач доцільним є застосування певних практичних рекомендацій та експериментальних досліджень.

Так, наприклад, мережі з невеликою кількістю нейронів і лінійними функціями активації, як правило, обмежуються в апроксимаційних можливостях і не дозволяють вирішувати реальні практичні задачі. З іншого боку, використання надмірної кількості шарів та нейронів може призвести до проблеми перенавчання та втрати узагальнюючих властивостей нейромережі. Також, для вирішення більшості практичних задач (класифікації, розпізнавання, прогнозування та ін.) можуть бути використані багатошарові мережі прямого поширення сигналу з нейронами, що використовують зважену суму в якості дискримінантної функції.

Після вирішення всіх вищенаведених задач структурного синтезу необхідно здійснити параметричний синтез нейронної мережі, який являє собою її навчання. У свою чергу, задача параметричного синтезу мережі з визначеною структурою полягає у виявленні такого набору значень синаптичних вагових коефіцієнтів та зсувів, за яких досягається найменше значення критерію помилки для вирішуваної задачі. Процес навчання нейромережі передбачає зміну значень даних коефіцієнтів шляхом послідовного подання прикладів навчальної вибірки. При проведенні навчання з учителем кожен приклад з вибірки подається на входи мережі, потім проходить обробку всередині її структури, та обчислюється вихідний сигнал мережі, який порівнюється з бажаним значенням цільового вектора. Після цього за певним правилом обчислюється помилка, і відбувається зміна вагових коефіцієнтів мережі залежно від обраного алгоритму. Вектори навчальної множини пред'являються послідовно, обчислюються помилки і ваги підлаштовуються для кожного вектора до



тих пір, поки помилка по всьому навчальному масиву не досягне прийнятно низького рівня.

З математичної точки зору процес навчання нейромережі являє собою задачу багатовимірної оптимізації, яка може бути вирішена за допомогою наступних методів: метод Коші, метод спряжених градієнтів, метод Ньютона, квазиньютонівські методи, метод Левенберга-Марквардта, стохастичні методи, біоінспіровані методи та ін.

Одним з найпростіших методів навчання нейронних мереж є метод Уідроу-Хоффа, який застосовується для навчання одношарового персептрона [32, 34]. Даний метод дозволяє здійснювати мінімізацію середньоквадратичної помилки (1.42) нейронної мережі шляхом підлаштування вагових коефіцієнтів та зсуву за допомогою градієнтного спуску. За цим методом для мінімізації помилки  $E$ , ваги і зсув змінюються протягом ітерацій згідно з виразами (1.43) та (1.44):

$$w_j(t+1) = w_j(t) - \alpha \cdot \frac{\partial E(t)}{\partial w_j(t)}; \quad (1.43)$$

$$w_0(t+1) = w_0(t) - \alpha \cdot \frac{\partial E(t)}{\partial w_0(t)}, \quad (1.44)$$

де  $t$  – номер ітерації,  $\alpha$  – швидкість навчання.

Для навчання багатшарових нейромереж прямого поширення сигналу найчастіше застосовується метод зворотного поширення помилки [32]. Навчання на основі даного метода передбачає два проходи через всі шари мережі: прямий та зворотний. При прямому проході вхідний вектор подається на вхідний шар нейронної мережі, після чого поширюється мережею від шару до шару. В результаті генерується набір вихідних сигналів, який є фактичною реакцією мережі на даний вхідний образ. Під час прямого проході всі синаптичні ваги мережі є фіксованими. При зворотному проході всі синаптичні ваги налаштовуються відповідно до правила корекції помилок, а саме: фактичний вихід мережі віднімається від бажаного, у результаті формується сигнал помилки. Цей сигнал згодом

поширюється по мережі у напрямку, зворотному напрямку синаптичних зв'язків. У свою чергу синаптичні ваги налаштовуються для максимального наближення вихідного сигналу мережі до бажаного. Метод зворотного поширення помилки може бути реалізований у пакетному та стохастичному режимах [29].

Окрім традиційних багат шарових нейромереж прямого поширення сигналу, на сьогодні, для вирішення різнотипних прикладних задач досить широко застосовуються й інші типи мереж, серед яких: нейронні мережі з латеральними зв'язками (мережі Кохонена), рекурентні нейронні мережі (мережі Хопфілда, Елмана та ін.), глибинні нейронні мережі (згорткові, довгої короткочасної пам'яті та ін.). У підсумку цього підрозділу коротко розглянемо найбільш поширені з наведених нейромереж.

*Нейронна мережа Кохонена.* Нейромережа Кохонена, також відома як карта самоорганізації Кохонена (self-organizing map – SOM), представляє собою вид нейромережі, який використовується для кластеризації та візуалізації даних. Одна з основних особливостей мережі такого типу – самонавчання, що дозволяє організовувати вхідні дані у структуровані групи без введення заздалегідь визначених категорій [29].

Структура нейронної мережі Кохонена складається з двох основних шарів: вхідного та вихідного. Вхідний шар складається з  $n$  нейронів, що відповідає кількості вхідних змінних. Кожен нейрон у вхідному шарі отримує вхідні дані, що представляють собою  $n$  чисел (вхідних параметрів). Кожен параметр взаємодіє з вагою, яка автоматично генерується для кожного нейрона. Вихідний шар являє собою карту Кохонена і складається з  $m$  нейронів. Кожен нейрон вихідного шару представляє собою вектор ваг для кожного вхідного параметра. Ці ваги визначаються випадковим чином на початковому етапі навчання. Також, кожен нейрон вхідного шару пов'язаний з кожним нейроном вихідного шару, а відповідний зв'язок має свою вагу, яка визначає, наскільки сильно вхідний параметр впливає на конкретний нейрон вихідного шару. Для

кожного нейрона вихідного шару визначається функція відстані, яка вказує, наскільки схожі його ваги на вхідні дані. Зазвичай використовується евклідова відстань. Нейрон вихідного шару, для якого функція відстані є мінімальною, вважається найкращим (Best Matching Unit – BMU). Цей нейрон вважається переможцем на конкретному кроці навчання. Після визначення BMU, ваги нейронів вихідного шару оновлюються таким чином, щоб найближчі до BMU нейрони були більше схильні до навчання. У свою чергу, процес навчання триває певну кількість ітерацій, під час яких нейрони вихідного шару адаптують свої ваги, щоб краще апроксимувати структуру даних. Завдяки цьому процесу нейронна мережа Кохонена має здатність до самоорганізації, тобто адаптації до складних структур даних з самостійним виявленням внутрішніх зв'язків. Вихід результату: В результаті вихідні нейрони утворюють карту Кохонена, яка відображає структуру даних в просторі нейронів. Це застосовується для класифікації та візуалізації даних.

Застосування нейронних мереж Кохонена розповсюджене в багатьох сферах. Вони використовуються для кластеризації даних, аналізу зображень та обробки сигналів. Крім того, вони знаходять застосування в сферах, де потрібно виявити складні залежності в даних, таких як аналіз геоінформаційних систем, біологічних досліджень, аналізу тексту та ін.

*Нейронна мережа Хопфілда.* Нейромережа Хопфілда – це одна з перших та найвідоміших рекурентних нейронних мереж. Її особливість полягає в тому, що вона може працювати як асоціативна пам'ять, здатна відновлювати асоціації за частково втраченими або спотвореними вхідними даними [29].

Структура нейронної мережі Хопфілда досить проста. Вона складається з одного шару нейронів, кожен з яких зв'язаний з усіма іншими нейронами в мережі. Це означає, що кожен нейрон взаємодіє з кожним іншим. Зв'язки між нейронами мають ваги, які визначають силу

зв'язку між ними. Крім того, нейрони не мають зворотних зв'язків самих з собою.

Однією з основних властивостей нейромережі Хопфілда є її здатність асоціювати певні стани нейронів з певними вхідними зразками. Це означає, що нейронна мережа може відновлювати запам'ятовані зразки навіть при появі частково спотворених або неповних вхідних даних. Застосування даної нейронної мережі Хопфілда досить поширене у таких областях, як розпізнавання образів, зокрема в різних технічних пристроях, де може бути шум або спотворення вхідних даних. Також ця мережа використовується в задачах оптимізації, де потрібно знайти глобальний мінімум функції в багатовимірному просторі. Наприклад, в задачах комівояжера.

*Нейронна мережа Елмана.* Нейромережа Елмана – це тип рекурентної нейронної мережі, яка вперше була запропонована вченим Рональдом Елманом у 1990 році [35]. Основна особливість цієї мережі полягає у наявності зворотних зв'язків у нейронах прихованого шару, які дозволяють інформації «циркулювати» через елементи мережі, що надає їй здатність до тимчасового аналізу та передбачення послідовностей даних.

Структура нейронної мережі Елмана включає в себе вхідний шар, прихований шар, контекстний шар зі зворотними зв'язками й елементами затримки та вихідний шар. Найбільш важливою особливістю є те, що прихований шар має зворотні зв'язки, які через контекстний шар передають сигнали назад до себе та до інших нейронів цього ж шару. Це дозволяє мережі аналізувати часові залежності вхідних даних та вирішувати задачі, пов'язані з послідовностями.

Нейронні мережі Елмана знайшли широке застосування в задачах прогнозування часових рядів, розпізнавання мови, обробки природних мов, керування рухомими системами та багатьох інших сферах. Їхні здатності до аналізу та прогнозування послідовностей роблять їх

корисними у таких областях, де важливий аспект часу та залежності від попередніх подій.

*Згорткові нейронні мережі.* Згорткові нейронні мережі (Convolutional Neural Networks – CNN) є глибинними нейромережами прямого поширення і являють собою потужні інструменти у сфері комп'ютерного зору та обробки зображень [36, 37]. Однією з основних особливостей цих мереж є їх здатність ефективно розпізнавати шаблони в зображеннях, незалежно від їхнього розташування. Це досягається завдяки використанню згорток – фільтрів, які ковзають по вхідному зображенню, виділяючи його певні особливості.

Структура згорткових нейронних мереж включає в себе кілька основних типів шарів. Згорткові шари (Convolutional Layers) використовуються для виявлення локальних особливостей на зображеннях. Кожен шар включає в себе набір фільтрів, які ковзають по вхідному зображенню. Фільтри використовуються для виділення певних патернів, таких як краї, кутові точки або інші важливі деталі. Після кожного згорткового шару застосовують функцію активації, яка допомагає в нелінійному моделюванні. Найбільш поширеною функцією активації є ReLU (Rectified Linear Unit). Пулінг-шари (Pooling Layers) призначені для скорочення просторового розміру подання, зменшуючи кількість параметрів і обчислень. Наприклад, застосовується метод Max Pooling, який вибирає максимальне значення з певної області. Повнозв'язні шари (Fully Connected Layers) використовуються для обробки векторів ознак, отриманих після згортки та пулінгу. Ці шари відповідають за вирішення кінцевої задачі, такої як класифікація.

Застосування згорткових нейронних мереж вкрай різноманітне. Вони з успіхом використовуються у великій кількості завдань, включаючи класифікацію зображень, визначення об'єктів, у робототехніці, аналізі медичних зображень, автономному водінні та багатьох інших областях.

Їхні високі результати в обробці великих обсягів зображень роблять їх незамінними в індустріях, де важлива точність та швидкість аналізу.

*Нейронні мережі довгої короткочасної пам'яті.* Нейронні мережі довгочасної короткочасної пам'яті (Long Short-Term Memory networks – LSTM) є особливим типом глибинних рекурентних нейромереж і призначені для ефективної обробки та моделювання послідовних даних [38, 39]. LSTM мережі мають наступні основні риси та характеристики.

Контроль потоку інформації – є основним принципом роботи LSTM мереж і передбачає наявність механізму керування потоком інформації в кожному нейроні. Це дозволяє мережі визначати, яку частину інформації слід зберегти, яку частину відкинути, а яку – використати для подальшого передавання.

Наявність «гейтів» (gates) – принцип, що передбачає наявність у LSTM мереж трьох типів «гейтів»: входу (input gate), виходу (output gate) і «гейту» для «забування» інформації (forget gate). Вони визначають, яка частина інформації буде використана, а яка – проігнорована чи відкинута. «Гейти», також, допомагають контролювати потік інформації в нейронах.

Наявність комірок пам'яті – принцип, що реалізований внутрішньою структурою LSTM для збереження та оновлення інформації протягом тривалого періоду часу. Комірки пам'яті дозволяють зберігати корисну інформацію протягом тривалого часу, що робить їх ефективними для роботи з послідовностями.

LSTM мережі мають наступну архітектуру та зв'язки. Нейромережа складається з певної кількості шарів, де кожен шар має велику кількість нейронів зі своїми вагами та функціями активації. Зв'язки між шарами реалізуються з використанням «гейтів» та інших наведених принципів.

Застосування LSTM мереж є досить різноманітним. Вони використовуються для обробки та аналізу послідовних даних у таких сферах, як мовний аналіз, машинний переклад, генерація тексту,

розпізнавання мови, прогнозування часових рядів, аналіз біомедичних сигналів та ін.

### 1.3 Гібридні нейро-нечіткі мережі

*Загальна характеристика та основні властивості гібридних нейро-нечітких мереж.* Кожен з розглянутих у попередніх підрозділах типів інтелектуальних систем має свої унікальні риси, особливо щодо їх здатностей у навчанні, узагальненні та отриманні результатів. Це робить їх більш адаптованими до вирішення певних класів завдань і менш ефективними для інших. Наприклад, системи, на основі нечіткої логіки, відмінно підходять для відтворення залежностей на основі зрозумілих логічних правил, проте вони не можуть самостійно набувати знання для подальшого використання у процесі виведення. Також, вони мають обмеження при значному збільшенні кількості вхідних змінних та необхідності додаткового розбиття загальних наборів даних на окремі категорії.

Нейронні мережі навпаки відмінно справляються з завданнями узагальнення великої кількості інформації, проте дуже складно пояснити утворені для цього залежності. Хоча, вони можуть автоматично здобувати знання, процес навчання часто відбувається досить повільно, і аналіз навченої мережі представляє собою складну задачу (навчена мережа залишається, як правило, "чорною скринькою" для користувача). Запровадження апріорних знань чи експертної інформації для прискорення процесу навчання в нейронній мережі також є нетривіальним завданням.

У свою чергу, у роботі [40] доведено, що нейронна мережа прямого поширення сигналу може ефективно апроксимувати системи, які базуються на нечітких правилах, і, відповідно, кожна нейромережа прямого поширення може бути апроксимована системою, що використовує нечіткі правила. В теоретичному плані, системи з нечіткою логікою та

штучні нейронні мережі схожі, але на практиці вони мають свої переваги та недоліки. Це розуміння лягло в основу розробки гібридних нечітких нейронних мереж, де виведення базується на принципах нечіткої логіки, але відповідні функції належності побудовані за допомогою методів навчання нейронних мереж, таких як метод зворотного поширення помилки. Ці системи не лише використовують апіорні знання, але й можуть набувати нові знання, залишаючись логічно зрозумілими.

У загальному вигляді гібридна нейро-нечітка мережа представляє собою модель нечіткого виведення, організовану у формі нейронної мережі [34]. Це робить її зручною для навчання, розширення, аналізу та використання. Архітектура нейро-нечіткої мережі відповідає основним компонентам систем нечіткого виведення.

Нейро-нечітка мережа – це багат шарова нейронна мережа специфічної структури, що не використовує зворотні зв'язки. В ній застосовуються звичайні чіткі сигнали, ваги та функції активації, а операція підсумовування базується на використанні фіксованих Т-норм, Т-конорм або інших безперервних операцій. Основною метою нейро-нечітких мереж є формалізація знань, застосовуючи нечіткі правила на основі нейронних мереж. Цей підхід дозволяє компенсувати один з головних недоліків нейронних мереж – їх непрозорість. Таким чином, нейронні мережі стають більш пояснюваними.

Гібридні нейронні мережі або нечіткі гібридні системи об'єднують переваги як нейронних мереж, так і систем нечітко-логічного виведення. З одного боку, вони дозволяють представляти моделі систем у вигляді набору нечітких правил, які легко і зрозуміло інтерпретувати. З іншого боку, вони використовують методи нейронних мереж для побудови цих нечітких правил, що є зручним та менш трудомістким для експертів-розробників. Головною ідеєю, що стоїть в основі побудови гібридних мереж, є використання наявної вибірки даних для визначення параметрів нечітких функцій належності, які найкраще підходять для певної системи



нечіткого виведення. Для цього використовуються відомі процедури навчання нейронних мереж для налаштування основних параметрів. Крім того, найбільш поширеними типами нечітких систем є системи типу Мамдані та Тагакі-Сугено, які мають модульну структуру, що дозволяє представити їх у вигляді однорідної багатошарової структури, яка має схожу з класичними нейронними мережами організацію.

Гібридні нейро-нечіткі мережі мають наступні властивості:

- ґрунтуються на нечітких системах, які навчаються за допомогою методів, аналогічних тим, що використовуються у нейромережах;
- мають декілька шарів (зазвичай три або більше). Перший шар представляє вхідні дані, другий містить нечіткі правила, а третій – вихідні дані. Ваги з'єднань відповідають нечітким множинам вхідних і вихідних даних. Інколи використовується архітектура з п'яти шарів. У загальному випадку нечітка система не обов'язково має бути подана у цьому вигляді, але це зручна модель для використання існуючих методів навчання;
- можуть бути завжди (перед, під час і після навчання) інтерпретовані у вигляді системи нечітких правил;
- процес навчання враховує семантичні аспекти нечіткої системи, що виражається у важливості обмеження можливих модифікацій, що застосовуються до параметрів, які підлягають налаштуванню. Важливо зауважити, що не всі методи навчання нейро-нечітких мереж враховують семантичну складову системи;
- можуть здійснювати апроксимацію  $n \times m$ -розмірної невідомої функції, яка частково описується навчальними даними.

На сьогодні, існують наступні типи інтелектуальних систем, що поєднують в собі нечітку логіку та штучні нейронні мережі [34].

*Нечіткі нейронні системи (fuzzy neural systems).* Системи, які використовують принципи нечіткої логіки для прискорення процесу навчання та налаштування певних параметрів у нейронних мережах. У

цьому підході, нечітка логіка виступає як інструмент для оптимізації функціонування нейронних мереж, не перетворюючись при цьому на набір чітких правил. Така система не може бути інтерпретована як набір нечітких правил, оскільки представляє собою «чорну скриньку».

*Паралельні нейро-нечіткі системи (cooperative neuro-fuzzy systems).* Інтелектуальні системи, які використовують нейронні мережі для знаходження та оптимізації значень параметрів. Після налаштування, нечітка система працює автономно. В цьому контексті виділяється декілька типів паралельних нейро-нечітких мереж: 1) нечітка асоціативна пам'ять (fuzzy associative memory); 2) системи, що створюють нечіткі правила за допомогою карт самоорганізації (fuzzy rule extraction using self-organizing maps); 3) системи, які дозволяють визначати параметри нечітких множин шляхом навчання (systems capable of learning fuzzy set parameters).

*Конкуруючі нейро-нечіткі системи (concurrent neuro-fuzzy systems).* Системи, в яких модель нечіткого логічного виведення і нейронна мережа виконують одне спільне завдання і не впливають на параметри одна однієї. Також, в даних системах обробка даних може виконуватися послідовно – спочатку за допомогою однієї системи, а потім із застосуванням іншої.

*Інтегровані нейро-нечіткі системи (integrated neuro-fuzzy systems).* Системи, що тісно поєднують основні принципи нечіткої логіки та нейронних мереж. У більшості випадків, коли говорять про «гібридні нейро-нечіткі мережі», мають на увазі саме цей тип систем. Наведені інтегровані системи, як правило, належать до нечітких систем типу Мамдані або Такагі-Сугено, параметри в яких визначаються за допомогою алгоритмів навчання штучних нейронних мереж.

Також, існуючі нейро-нечіткі мережі можна розділити за способом відображення нечітких множин у структурі мережі на наступні основні типи.

*Системи, що базуються на вибіркових нечітких множинах.* У таких системах функції належності представлені у вигляді векторів, а ступені

належності визначаються лише для обмеженої кількості значень з усього діапазону. Крім того, кожному ступеню належності відповідає тільки один вхідний або вихідний нейрон. Для реалізації систем такого типу може бути застосовано два підходи. Перший підхід полягає в тому, що система є так званою «чорною скринькою» і здійснює апроксимацію залежностей виходів від входів. Другий підхід передбачає створення системи зі спеціальною архітектурою, в якій реалізовані нечіткі правила.

*Системи, в яких параметризовані функції належності зберігаються в нейронах.* Найбільш поширеним прикладом інтелектуальних систем такого типу є адаптивна система нейро-нечіткого виведення ANFIS (Adaptive Neuro-Fuzzy Inference System) [41, 42].

*Системи, в яких параметризовані функції належності виступають в якості вагів для зв'язків між нейронами.* Системи такого типу можна називати «персептронами з нечіткими зв'язками» або «нечіткими персептронами». До таких систем можна віднести мережі NEFCON, NEFCLASS, NEFPROX [43, 44].

В залежності від типу навчання, нейро-нечіткі мережі можна розділити на дві основні категорії: мережі з самоорганізацією та адаптивні нейро-нечіткі мережі. У свою чергу, мережі з самоорганізацією можуть адаптувати свою структуру та параметри в процесі навчання.

Адаптивні нейро-нечіткі мережі являють собою системи з жорсткою структурою, в яких в процесі навчання змінюються лише параметри. Зокрема, в якості наведених параметрів можуть виступати параметри функцій належності, вагові коефіцієнти правил, параметри оператора агрегації та ін.

Далі розглянемо більш детально нейро-нечітку мережу типу ANFIS, яка є однією з найбільш поширених.

*Нейро-нечітка мережа ANFIS.* ANFIS являє собою інтегровану нейро-нечітку систему, яка сформована на основі гібридного підходу, поєднуючи в собі переваги нейронних мереж і нечіткої логіки. З одного

боку, ANFIS представляє собою нейронну мережу прямого поширення сигналу спеціального типу. З іншого боку, ця система реалізує Сугено-подібне нечітке логічне виведення. Таким чином, архітектура даної нейро-нечіткої мережі аналогічна нечіткій базі знань і представлена у вигляді п'ятишарової нейронної мережі прямого поширення [41]. Вона використовує трикутні норми, які диференціюються (операції множення та імовірнісне АБО), а також гладкі функції належності. Це надає можливість використовувати ефективні алгоритми навчання традиційних нейронних мереж для налаштування ANFIS, які ґрунтуються на методі зворотного поширення помилки [42].

Основною особливістю мережі ANFIS є адаптація функцій належності та ваг правил на основі навчального набору даних. Серед головних переваг слід зазначити те, що вона надає легку інтерпретацію моделей, оскільки кожен нейрон відповідає конкретному входу та правилу.

Кожне  $r$ -те правило бази правил ANFIS має вигляд:

$$\text{IF } x_1 = a_{1,r} \text{ AND...AND } x_n = a_{n,r} \text{ THEN } y = b_{0,r} + b_{1,r}x_1 + \dots + b_{n,r}x_n, \quad (1.45)$$

де  $r = 1, \dots, s$  – номер відповідного правила;  $a_{i,r}$  – лінгвістичний терм із функцією належності  $\mu_r(x_i)$ , який застосовується для фаззифікації змінної  $x_i$  у правилі  $r$  ( $r = 1, \dots, s$ ;  $i = 1, \dots, n$ );  $b_{q,r}$  – дійсні числа у виведенні правила  $r$  ( $r = 1, \dots, s$ ;  $q = 1, \dots, n$ ).

Узагальнена структура нейро-нечіткої мережі ANFIS, яка має дві вхідних змінних  $x_1, x_2$  та одну вихідну змінну  $y$ , представлена на рис. 1.14.

Принцип функціонування мережі ANFIS полягає у наступному.

Перший шар мережі має вхідні нейрони, кожен з яких відповідає за певний вхідний параметр та має певний терм із заданою функцією належності. У свою чергу, дана функція визначає належність параметру до кожного набору правил. Вхідні змінні нейро-нечіткої мережі  $x_1$  та  $x_2$  з'єднані тільки з їх відповідними термами. Кількість нейронів першого шару дорівнює сумі лінгвістичних термів усіх вхідних змінних.

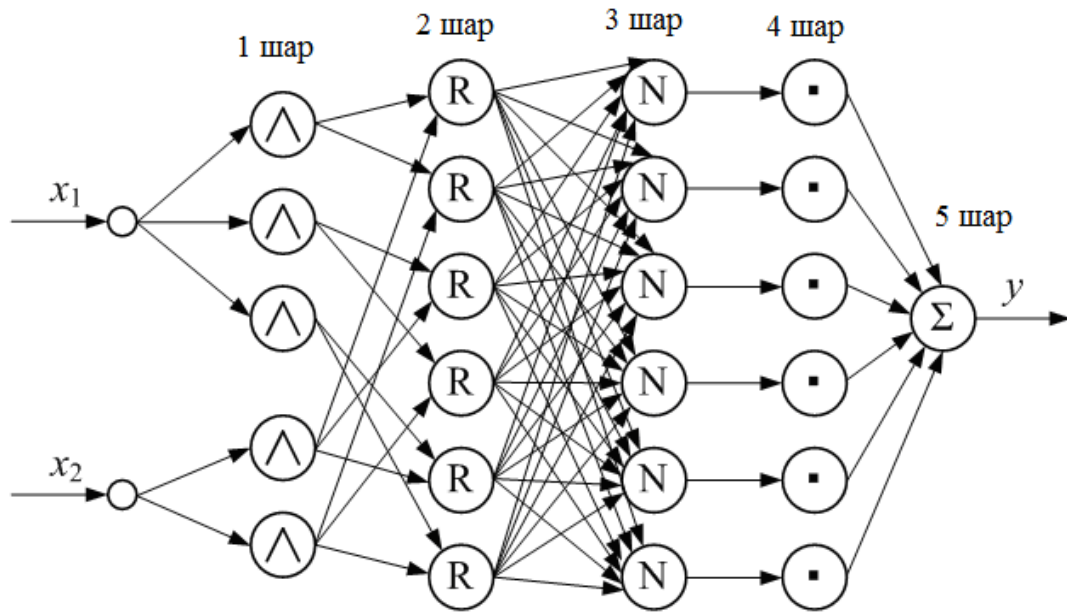


Рис. 1.14. Узагальнена структура нейро-нечіткої мережі ANFIS

На виході кожного нейрона формується ступінь належності певної вхідної змінної до відповідного лінгвістичного терма згідно з виразом (1.46)

$$\mu_r(x_i) = \frac{1}{1 + \left| \frac{x_i - g}{c} \right|^{2d}}, \quad (1.46)$$

де  $c, d, g$  – регульовані параметри функції належності  $\mu_r(x_i)$ .

Другий шар містить нейрони, кожен з яких представляє собою вагу кожного  $r$ -го правила. Кількість нейронів даного шару дорівнює кількості нечітких правил  $s$ . У свою чергу, кожен нейрон другого шару з'єднаний з тими нейронами першого шару, які формують антецеденти відповідного правила. Таким чином, кожен нейрон другого шару може приймати від 1 до  $n$  вхідних змінних. На виході нейронів даного шару формуються ступені виконання правил, які розраховуються як добутки відповідних вхідних змінних. Зазвичай, вихідні сигнали нейронів даного шару позначаються як  $\tau_r, r = \overline{1, s}$ .

Третій шар мережі містить також  $s$  нейронів. Кожен нейрон цього шару розраховує відносну ступінь виконання відповідного  $r$ -го правила на основі виразу (1.47)

$$\tau_r^* = \frac{\tau_r}{\sum_{j=1,s} \tau_j}. \quad (1.47)$$

Кількість нейронів четвертого шару мережі також дорівнює кількості правил  $s$ . Кожен нейрон даного шару з'єднаний тільки з одним нейроном попереднього (третього) шару, а також з усіма вхідними змінними мережі (на рис. 1.14 зв'язки із вхідними змінними не показано). У свою чергу, кожен нейрон четвертого шару визначає внесок одного нечіткого правила у вихідну змінну мережі згідно з виразом (1.48)

$$y_r = \tau_r^* \cdot (b_{0,r} + b_{1,r}x_1 + \dots + b_{n,r}x_n). \quad (1.48)$$

П'ятий шар мережі ANFIS має тільки один нейрон, який розраховує вихідний сигнал  $y$  як суму внесків усіх правил

$$y = y_1 + \dots + y_r + \dots + y_s. \quad (1.49)$$

При цьому, лінгвістичні терми вхідних змінних описуються стандартними для нечітких систем функціями належності, а вихідна змінна може бути представлена лінійною функцією або константою. Навчання представленої нейро-нечіткої мережі може здійснюватися за допомогою відомих методів навчання традиційних нейронних мереж.

Розглянуті у вищенаведених підрозділах системи, що базуються на різних методах інтелектуальних обчислень (нечітко-логічні, нейронні, нейро-нечіткі), мають свої переваги на недоліки. Ефективність їх використання в значній мірі залежить від особливостей конкретної розв'язуваної задачі та низки інших факторів, які повинен враховувати проєктувальник. Кожна окрема задача потребує ретельного підбору інтелектуальних методів та систем для її вирішення, які, також, повинні бути додатково адаптовані до її певних особливостей. Крім того, досить суттєвий вплив на отриманий результат мають застосовані методи синтезу,

налаштування та навчання. Правильний вибір цих методів може значно підвищити ефективність функціонування системи, допомагаючи досягти найкращих результатів у конкретному випадку. Таким чином, комплексний підхід до вибору типу системи та методів її синтезу є визначальним у досягненні успіху в застосуванні інтелектуальних технологій.

#### **1.4 Контрольні питання**

1. Сформулюйте основні принципи нечіткої логіки.
2. Наведіть особливості опису нечітких або неоднозначних явищ.
3. В чому полягає основна відмінність нечіткої логіки від класичної логіки?
4. Наведіть приклади реальних завдань, які можуть бути вирішені за допомогою нечітко-логічних систем.
5. Яким чином визначається функція належності в нечіткій логіці та як вона використовується?
6. Застосування нечітких множин та лінгвістичних змінних для формалізації експертної інформації в нечітких системах.
7. Наведіть узагальнену структуру одномірної нечіткої системи автоматичного керування та поясніть призначення її основних елементів.
8. Поясніть призначення операцій фазифікації, агрегації, активації, акумуляції та дефазифікації в нечітких системах.
9. Наведіть структуру узагальненої нечіткої СППР та поясніть призначення її основних елементів.
10. Перерахуйте основні типи функцій належності, які застосовуються в нечітких системах.
11. Які ви знаєте механізми нечіткого логічного виведення, що застосовуються в нечітких системах?

12. Наведіть основні етапи нечіткого логічного виведення типу Мамдані.
13. Охарактеризуйте механізм нечіткого логічного виведення Ларсена.
14. У чому полягають особливості нечіткого логічного виведення Цукамото?
15. Наведіть основні відмінності механізмів нечіткого логічного виведення Мамдані та Такагі-Сугено.
16. Охарактеризуйте механізм НЛВ типу-2.
17. У чому полягають переваги механізму фрактального НЛВ?
18. Назвіть основні переваги та недоліки інтелектуальних систем на основі нечіткої логіки.
19. На чому базується теорія штучних нейронних мереж?
20. Що являє собою штучна нейронна мережа?
21. Назвіть основні сфери застосування нейронних мереж та основні задачі, які вони можуть вирішувати.
22. Наведіть спрощену структуру біологічного нейрону та охарактеризуйте його основні компоненти.
23. Які основні властивості біологічних нейронів моделюють штучні нейрони?
24. Для чого у штучних нейронах застосовуються синаптичні ваги, дискримінантні функції та функції активації?
25. Перерахуйте основні типи дискримінантних функцій.
26. Наведіть найбільш поширені функції активації нейронів.
27. Наведіть особливості одношарового персептрону.
28. Наведіть структуру та основні характеристики багатошарового персептрону.
29. Наведіть загальні властивості штучних нейронних мереж.
30. За якими ознаками проводять класифікацію нейромереж?



31. Чим відрізняються повнозв'язна, неповнозв'язна, та слабозв'язна топології нейромереж?
32. Наведіть класифікацію нейронних мереж за характером зв'язків.
33. Наведіть основні етапи синтезу штучних нейронних мереж.
34. Охарактеризуйте процес навчання нейронних мереж.
35. Які ви знаєте методи навчання штучних нейронних мереж?
36. В чому полягають особливості методу Уїдроу-Хоффа?
37. Назвіть основні етапи методу зворотного поширення помилки.
38. Наведіть структуру нейронної мережі Кохонена та поясніть алгоритми її функціонування.
39. Назвіть особливості нейронної мережі Хопфілда.
40. Наведіть основні переваги та недоліки нейронної мережі Елмана.
41. У чому полягає концепція глибинних нейромереж?
42. Наведіть базові аспекти, особливості навчання та основні сфери застосування згорткових нейронних мереж.
43. Перерахуйте основні характеристики LSTM нейромереж.
44. Наведіть основні властивості гібридних нейро-нечітких мереж.
45. Що являє собою гібридна нейро-нечітка мережа?
46. Які типи інтелектуальних систем, що поєднують в собі нечітку логіку та штучні нейронні мережі, ви знаєте?
47. Назвіть основні властивості нечітких нейронних систем.
48. Чим відрізняються між собою паралельні та конкуруючі нейро-нечіткі системи?
49. Які ви знаєте типи інтегрованих нейро-нечітких систем?
50. Що являють собою системи, які базуються на вибіркових нечітких множинах?
51. До якого типу нейро-нечітких систем можна віднести мережі NEFCON, NEFCLASS та NEFPROX?

52. Наведіть структуру та опишіть основні елементи нейро-нечіткої мережі типу ANFIS.
53. Назвіть основні особливості мережі ANFIS.
54. У чому полягає адаптація функцій належності та ваг правил мережі ANFIS?
55. Скільки шарів має нейро-нечітка мережа ANFIS?
56. Наведіть вираз, яким описується  $r$ -те правило бази правил ANFIS.
57. Охарактеризуйте процес навчання нейро-нечіткої мережі ANFIS.
58. Які функції виконує п'ятий шар нейро-нечіткої мережі ANFIS?

## РОЗДІЛ 2

### БІОІНСПРОВАНІ ЕВОЛЮЦІЙНІ ТА МУЛЬТИАГЕНТНІ МЕТОДИ ОПТИМІЗАЦІЇ

Протягом останніх двох десятирічь значна увага науковців і практичних фахівців у сфері ШІ та інформаційних технологій була спрямована на дослідження наближених методів та засобів глобальної оптимізації. Ці методи дозволяють знаходити високоякісні рішення для різноманітних науково-технічних, інженерних та економічних задач за прийнятний (з практичної точки зору) час [45-47]. Серед них особливу популярність здобули біоінспіровані метаевристичні методи та алгоритми. На відміну від класичних методів оптимізації, вони можуть успішно застосовуватися у ситуаціях, коли відсутня практично будь-яка інформація щодо характеру та властивостей досліджуваних функцій [48]. Крім того, біоінспіровані інтелектуальні методи базуються на інтуїтивних підходах і дозволяють ефективно вирішувати завдання, не вимагаючи формального доказу коректності застосовуваних пошукових процедур. Ці методи та алгоритми також легко адаптуються для вирішення задач оптимізації та структурно-параметричного синтезу різних інтелектуальних систем та пристроїв. Використовуючи прості базові обчислювальні процедури, дані методи моделюють фізичні явища, поведінку соціальних тварин, еволюційні концепції та ін. в процесі оптимізації [46, 49]. Водночас, біоінспіровані інтелектуальні методи ефективно комбінуються з методами локальної оптимізації різних типів, що дозволяє раціонально поєднувати глобальні та локальні стратегії пошуку [50, 51].

Наведені біоінспіровані методи інтелектуальної оптимізації можна умовно розділити на чотири основні категорії (рис. 2.1): 1) еволюційні; 2) мультиагентні; 3) методи, які моделюють фізичні процеси; 4) методи, що відтворюють поведінку людей.

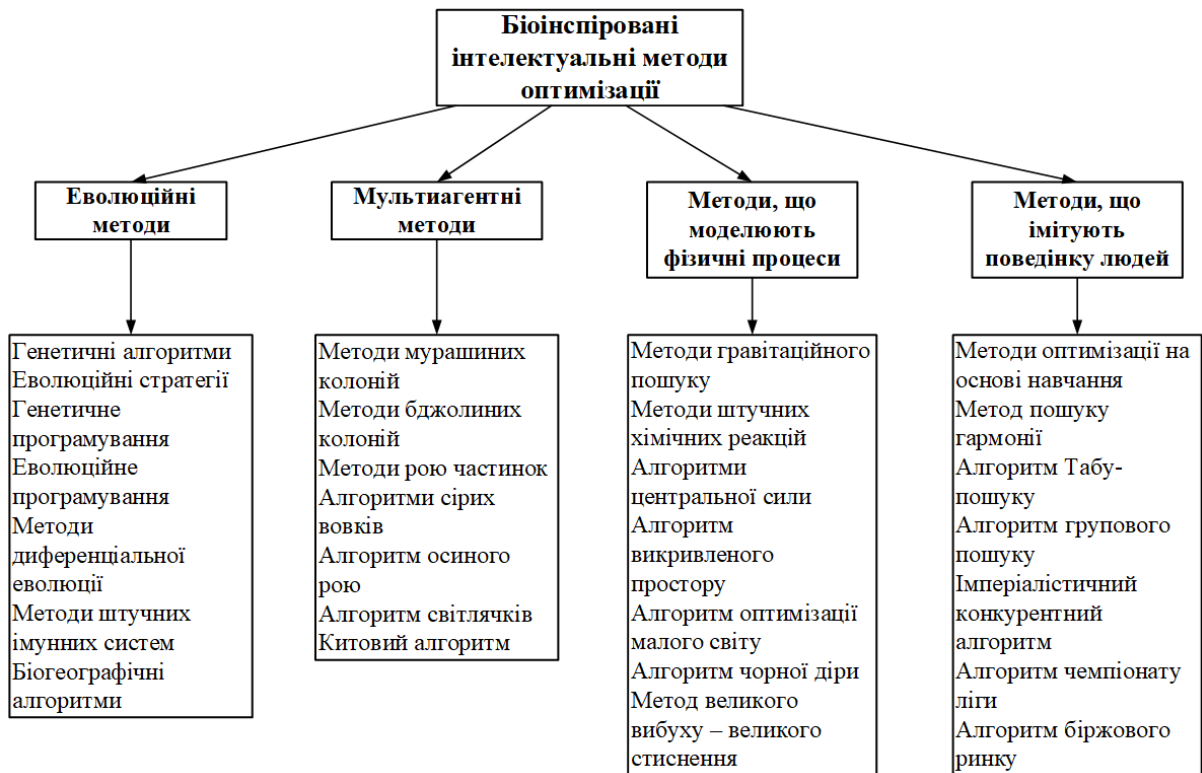


Рис. 2.1. Класифікація біоінспірованих методів інтелектуальної оптимізації

Найбільш поширеними еволюційними методами та алгоритмами ШІ на сьогодні є: генетичні алгоритми (Genetic Algorithms) [52], генетичне програмування (Genetic Programming) [53], еволюційні стратегії (Evolutionary Strategies) [54], еволюційне програмування (Evolutionary Programming) [55], методи диференціальної еволюції (Differential Evolution) [56, 57], метод оптимізації на основі динамічних сіток (Variable Mesh Optimization) [58], методи штучних імунних систем (Artificial Immune Systems) [59, 60], метод оптимізації, що імітує поширення бур'янів (Weed Colonization) [61], методи оптимізації на основі біогеографії (Biogeography-Based Optimization) [62], метод, що імітує поведінку зозуль (Cuckoo Search) [63, 64] та ін.

Серед мультиагентних методів колективного (ройового) інтелекту, які успішно застосовуються для вирішення різноманітних оптимізаційних задач, можна виокремити такі: методи мурашиних (Ant Colony Optimization) [65, 66] та бджолиних (Bee Colony Optimization) [67, 68]

колоній, метод імітації поведінки бактерій (Bacterial Foraging Optimization) [69], методи рою частинок (Particle Swarm Optimization) [70, 71] та зграї риб (Fish School Search) [72-74], оптимізаційні алгоритми сірих вовків (Grey Wolf Optimization) [75, 76] та світлячків (Firefly Algorithm) [77], метод, що імітує поведінку кажанів (Bat-Inspired Algorithm) [78], алгоритм осинового рою (Wasp Swarm Algorithm) [79], метод, що імітує поведінку популяції криля (Krill Herd) [80], китовий алгоритм (Whale Optimization Algorithm) [81] та багато інших.

До групи оптимізаційних методів, що моделюють різнотипні фізичні процеси, відносяться: метод імітації відпалу (Simulated Annealing) [82], методи гравітаційного пошуку (Gravitational Search Algorithm) [83, 84], алгоритми центральної сили (Central Force Optimization) [85] та викривленого простору (Curved Space Optimization) [86], методи штучних хімічних реакцій (Chemical Reaction Optimization Algorithm) [87], алгоритми оптимізації малого світу (Curved Space Optimization) [88] та чорної діри (Black Hole Algorithm) [89], метод великого вибуху – великого стиснення (Big-Bang Big-Crunch) [90].

У свою чергу, до інтелектуальних алгоритмів та методів, що імітують поведінку людей в процесі оптимізації, відносяться: методи пошуку гармонії (Harmony Search) [91, 92], методи оптимізації на основі навчання (Teaching Learning Based Optimization) [93, 94], алгоритм Табу-пошуку (Tabu Search) [95], імперіалістичний конкурентний алгоритм (Imperialist Competitive Algorithm) [96], алгоритм групового пошуку (Group Search Optimizer) [97], алгоритм чемпіонату ліги (League Championship Algorithm) [98], алгоритми оптимізації на основі групового консультування (Group Counseling Optimization) [99, 100], алгоритм біржового ринку (Exchange Market Algorithm) [101], алгоритм внутрішнього пошуку (Interior Search Algorithm) [102] та ін.

Напрямок біоінспірованої оптимізації у галузі ІІІ постійно розширюється, зокрема, щороку з'являються нові високоефективні методи

та алгоритми, що робить неможливим охоплення всіх їхніх аспектів в рамках одного навчального посібника. Тому, у даному розділі розглянуто лише декілька методів, які, на сьогодні, є добре апробованими та досить широко застосовуються для вирішення різного плану оптимізаційних задач. У свою чергу, у підрозділах 2.1-2.6 коротко висвітлено базові принципи та основні кроки таких широко розповсюджених біоінспірованих методів глобальної оптимізації, як: генетичні, штучних імунних систем, біогеографічні, мурашиних колоній, рою частинок та сірих вовків.

## 2.1 Генетичні методи

*Базові аспекти еволюційних обчислень.* Еволюційні методи оптимізації являють собою клас біоінспірованих методів, які моделюють природний процес еволюції для пошуку оптимальних рішень у просторі параметрів. Ці методи та алгоритми виникли у контексті еволюційного обчислення та стали популярними у різних галузях завдяки їх здатності ефективно розв'язувати складні оптимізаційні завдання.

Основні компоненти еволюційних алгоритмів включають популяцію індивідів, функцію пристосованості, оператори варіації (кросовер та мутація) та стратегії відбору [34]. Початкова популяція, зазвичай, генерується випадковим чином, індивідуальні елементи якої представляють потенційні рішення задачі оптимізації. Функція пристосованості (фітнес-функція) визначає якість кожного індивіда в контексті пошукової задачі. На кожній ітерації еволюційного алгоритму використовуються оператори варіації для створення нових індивідів, які зазвичай отримуються шляхом кросовера (комбінування частин хромосом батьківських індивідів) та мутації (випадкових змін частин хромосоми). Після генерації нових індивідів відбувається відбір, де обираються ті, які мають найвищу пристосованість для подальшої репродукції [48].

Еволюційні алгоритми виявляються ефективними для оптимізації функцій у просторі параметрів, що має багато локальних оптимумів, а також в ситуаціях, коли функція має велику кількість параметрів або не має аналітичного виразу. Вони знаходять застосування у багатьох галузях, таких як інженерія, економіка та біоінформатика.

Унаслідок того, що еволюційні методи в процесі пошуку використовують кодування множини параметрів, а не самі параметри, вони досить ефективно застосовуються для розв'язання завдань дискретної оптимізації, які можуть бути визначені як на числових, так і на скінчених множинах будь-якої природи. Еволюційні методи не потребують інформації щодо похідних чи інших характеристик функції, яку оптимізують, а використовують лише її значення у конкретних точках простору пошуку. Ці методи можуть успішно застосовуватися до широкого спектру функцій, включаючи ті, для яких відсутні аналітичні описи. Еволюційні методи є гнучкими та придатними для рішення різноманітних задач, зокрема тих, для яких не існує загальновідомих методів [34, 48].

Порівняно з класичними методами оптимізації, еволюційні методи є більш ефективним інструментом в умовах (а) вирішення завдань, які не вимагають досягнення точного глобального оптимуму, але потребують швидкого знаходження задовільного рішення, (б) великого, негладкого та мультимодального простору пошуку, (в) цільової функції з шумами [46, 48].

Отже, еволюційний пошук є ефективним інструментом для пошуку оптимальних значень мультимодальних функцій та успішно використовується для розв'язання комбінаторних задач.

*Узагальнений генетичний алгоритм.* Генетичні алгоритми (Genetic Algorithms – GA) є одними з найбільш поширених методів еволюційної оптимізації. В GA створюється популяція особин, кожна з яких представляється у вигляді хромосоми. У задачі оптимізації множина

допустимих рішень кодується таким чином, щоб кожна хромосома відповідала одному з можливих рішень. У свою чергу, хромосома складається з кінцевого числа генів, представляючи генотип об'єкта. Пошук екстремуму проводиться на рівні генотипів [103].

Блок-схема узагальненого генетичного алгоритму наведена на рис. 2.2 [48].

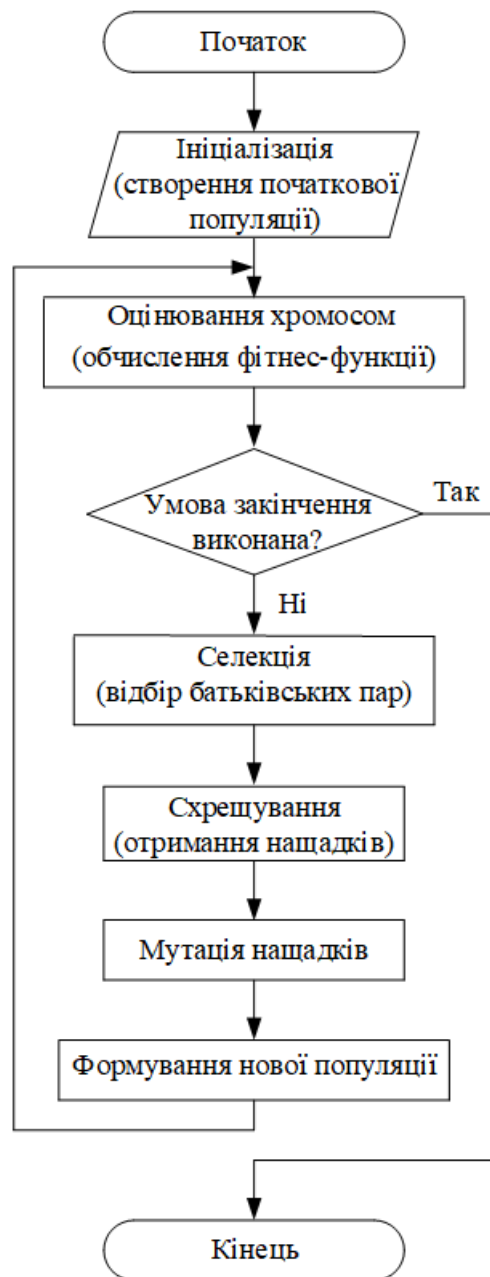


Рис. 2.2. Блок-схема узагальненого генетичного алгоритму



Основними етапами узагальненого GA є: ініціалізація, оцінювання хромосом, перевірка виконання умови закінчення роботи алгоритму, селекція, схрещування, мутація та формування нової популяції. Після створення початкової популяції на етапі ініціалізації виконується циклічний ітераційний процес із застосуванням основних біологічних операторів (селекція, схрещування та мутація) до тих пір, поки не буде знайдено прийнятне рішення задачі оптимізації, або не буде виконана інша умова закінчення алгоритму (виконання максимальної кількості ітерацій тощо) [46].

Основною процедурою етапу ініціалізації GA є формування початкової популяції. На даному етапі хромосоми в початковій популяції можуть бути випадково обрані або введені користувачем базуючись на певній апріорній інформації. Вибір початкової популяції не впливає на збіжність оптимізаційного процесу у великих масштабах, однак створення добре підбраної початкової популяції, наприклад, із використанням набору локальних оптимумів, може помітно зменшити час досягнення глобального оптимуму. Отже, при наявності необхідної інформації, визначення початкової популяції користувачем має певні переваги. У деяких випадках доцільно встановлювати початкові значення генів хромосом випадковим чином у рамках попередньо визначених діапазонів. Також, може бути використано рівномірне формування початкової популяції, або формування, засноване на знаннях [34].

Механізм селекції, також відомий як оператор відбору, визначає процес вибору хромосом для подальшого застосування до них різних еволюційних операторів та формування нової популяції. У різних моделях еволюційного пошуку оператор відбору може використовуватися для виконання різних процедур, серед яких: вибір хромосом-батьків для схрещування (кросоверу), формування пар для кросоверу чи вибір хромосом для створення нового покоління із хромосом-батьків та хромосом-нащадків, отриманих через кросовер і мутацію [48].

Зазвичай, оператор селекції використовується для вибору хромосом-батьків для схрещування, де вони обираються базуючись на значеннях цільової функції так, щоб будь-який елемент популяції мав ймовірність бути обраним як один із батьків для схрещування. У свою чергу, найпоширеніші типи операторів відбору включають пропорційний відбір (пропорційно-імовірнісний), відбір на основі ранжування, турнірний відбір, відбір з використанням порогу та ін. [34].

Оператор схрещування в генетичних алгоритмах відіграє важливу роль у формуванні нових індивідів у популяції. Його основна функція полягає в об'єднанні генетичної інформації двох батьківських особин для створення нащадків з унікальним генетичним складом. Цей механізм імітує природний процес генетичного обміну, що відбувається у природній еволюції.

Наприклад, при обранні точкового схрещування випадковим чином обираються точки або місця на генетичних ланцюгах батьків, і генетична інформація обмінюється між цими точками. Цей обмін генетичною інформацією призводить до створення нових комбінацій генів у нащадків. Схрещування сприяє різноманітності популяції та допомагає забезпечити шлях до знаходження оптимальних рішень під час еволюційного процесу GA [46, 103].

Перед тим, як виконувати, безпосередньо, схрещування індивідів необхідно певним чином обрати конкретні батьківські пари. Для цього існує декілька способів, найбільш поширеними з яких є: випадковий та селективний способи, інбридинг та аутбридинг [34]. Після визначення конкретних батьківських пар здійснюється операція схрещування за допомогою певного оператора. На сьогодні, існує досить багато різних типів схрещування, серед яких: точкове (одноточкове, двоточкове), рівномірне, однорідне, порівняльне, лінійне, арифметичне, нечітке та ін. Докладний опис основних обчислювальних процедур згідно з наведеними типами операторів схрещування наведений в [48].

Після проведення схрещування виконується мутація хромосом. Оператор мутації в GA відіграє роль введення випадкових змін у генетичні коди окремих особин у популяції. Цей елемент важливий для збереження різноманітності генетичного матеріалу і дозволяє випробувати нові варіанти, які можуть призвести до вдосконалення популяції. Оператор мутації використовується для інтродукції обмежених, але важливих змін в генетичний код, що допомагає уникнути передчасної збіжності генетичного алгоритму. На сьогодні існує досить багато різних типів операторів мутації, серед яких проста мутація, мутація обміну, мутація на основі інвертування, мутація з випадковим додаванням або видаленням генів та ін. [48]. Вибір конкретного виду мутації впливає на ефективність алгоритму та його здатність досліджувати генетичний простір для знаходження оптимальних рішень.

Процедура формування нового покоління є заключним етапом на кожній ітерації GA, що визначає, які індивіди входитимуть у нову популяцію. Після завершення операцій схрещування та мутації, формування нового покоління, яке буде використане для подальших ітерацій еволюційного процесу, здійснюється у кілька етапів. На першому етапі визначається, які індивіди потраплять у нове покоління. Воно може включати всіх батьківських особин, а також нащадків, отриманих в результаті кросоверу та мутації. Застосування критеріїв відбору, таких як значення цільової функції або ранжирування, дозволяє відбирати індивідів, які найбільше відповідають меті оптимізації. Наступний етап – безпосередньо формування нового покоління на основі обраних індивідів. Даний етап може включати просте копіювання обраних особин або застосування додаткових операцій, таких як рекомбінація або фільтрація, для забезпечення більшої різноманітності та вдосконалення популяції. При цьому, усі нові хромосоми-нащадки можуть замінити всіх хромосом-батьків, або може бути застосована стратегія елітизму, коли певна кількість обраних хромосом з найбільшою пристосованістю гарантовано

переходять у нову популяцію [34]. В результаті, сформоване нове покоління замінює попереднє, і процес еволюції продовжується. Ця процедура сприяє поступовій конвергенції популяції до оптимальних рішень та ефективній експлорації генетичного простору в пошуках оптимальних рішень [46, 103].

Далі розглянемо більш докладно основні кроки узагальненого генетичного алгоритму.

*Крок 1.* Ініціалізація генетичного алгоритму. На цьому етапі здійснюється постановка задачі оптимізації, а також обираються параметри та типи генетичних операторів GA. Крім того, обирається структура, параметри та оптимальне значення цільової функції (фітнес-функції)  $f$  і критерій завершення оптимізації. Кожен шуканий параметр (змінна) задачі подається у певному закодованому вигляді. У свою чергу, вектор даних параметрів  $X_p$  кодується у вигляді хромосоми, розмір якої визначається загальним числом невідомих у задачі. На даному етапі обираються генетичні оператори відбору (селекції), схрещування (кросоверу) та мутації. Також, встановлюються ймовірності схрещування  $P_C$  і мутації  $P_M$  та розмір початкової популяції  $Z_{\max}$ .

В якості критерія завершення оптимізації можна використовувати виконання наступних умов [46, 104].

*Умова 1.* Досягнення оптимального значення фітнес-функції  $f_{\text{opt}}$ . Якщо значення фітнес-функції  $f$  деяких хромосом досягає  $f_{\text{opt}}$  в процесі функціонування GA, то критерій вважається досягнутим.

*Умова 2.* Виконання максимальної кількості ітерацій  $N_{\max}$ . У цьому випадку максимальна кількість ітерацій встановлюється попередньо, перед початком генетичного пошуку.

*Умова 3.* Виродження популяції. У даному випадку перед початком роботи GA необхідно встановити кількість ітерацій  $N_p$  та поріг коефіцієнта покращення значень фітнес-функції найкращої хромосоми  $\rho_p$ . Починаючи з ітерації  $N_p$ , на кожній наступній ітерації ( $N_p + 1$ ) обчислюється коефіцієнт

покращення  $\rho$  і порівнюється з пороговим значенням  $\rho_p$ . У свою чергу, коефіцієнт покращення  $\rho$  розраховується як

$$\rho = \frac{f_{best(N)} - f_{best(N-1)}}{f_{best(N-1)}}, \quad (2.1)$$

де  $f_{best(N)}$  – найкраще значення фітнес-функції  $f$  на ітерації  $N$ ;  $f_{best(N-1)}$  – найкраще значення фітнес-функції  $f$  на ітерації  $N - 1$ . Якщо значення  $\rho$  менше  $\rho_p$ , то критерій завершення вважається досягнутим.

*Крок 2.* Кодування вектору параметрів задачі в хромосому. На даному етапі вектор шуканих параметрів кодується в хромосому, що складається з генів, кожен з яких відповідає певному параметру задачі. Зазвичай, використовується бінарне або дійсне кодування (іноді застосовується змішане кодування) [46, 48]. При виборі бінарного кодування використовується двійковий алфавіт для кодування точок на множині допустимих рішень (гени можуть приймати значення 0 або 1). У свою чергу, при обранні другого варіанту кодування, рішення в хромосомі представляється у вигляді набору дійсних чисел.

*Крок 3.* Створення початкової популяції хромосом. На даному кроці створюється початкова популяція хромосом  $P_0$ , що містить інформацію про  $Z_{\max}$  різних варіантів вектора  $\mathbf{X}_P$  невідомих у задачі (кінцевий набір допустимих рішень задачі). У свою чергу, початкова популяція  $P_0$  описується виразом

$$P_0 = \left\{ \mathbf{X}_{P_1}, \mathbf{X}_{P_2}, \dots, \mathbf{X}_{P_Z}, \dots, \mathbf{X}_{P_{Z_{\max}}} \right\}, \quad (2.2)$$

де  $\mathbf{X}_{P_Z}$  –  $Z$ -та хромосома популяції. У багатьох випадках доцільно встановлювати початкові значення генів хромосом випадковим чином в рамках попередньо визначених діапазонів [103].

*Крок 4.* Оцінка хромосом поточної популяції. Оцінка хромосом здійснюється за такими кроками.

*Крок 4.1.* Декодування хромосом. На цьому кроці реалізується декодування кожної хромосоми популяції у вектор  $\mathbf{X}_{P_x}$  невідомих задачі.

*Крок 4.2.* Розрахунок фітнес-функції хромосом. На цьому етапі проводиться підстановка значень вектору  $\mathbf{X}_{P_x}$  у розв'язувану задачу, і розраховується значення фітнес-функції  $f_Z(N)$  для кожної  $Z$ -ї хромосоми популяції ( $Z = 1, 2, \dots, Z_{\max}$ ).

*Крок 5.* Перевірка критеріїв завершення оптимізації. Розв'язання задачі оптимізації може бути закінченим, коли виконується одна з умов завершення, встановлених на кроці 1. Якщо виконується умова 1, то здійснюється перехід до кроку 10. Якщо виконуються умови 2 або 3 – здійснюється перехід до кроку 1 для перезапуску алгоритму.

*Крок 6.* Відбір хромосом для схрещування. На цьому етапі хромосоми популяції відбираються для кросоверу з метою створення нових рішень на основі значень фітнес-функції  $f_Z(N)$ . Наприклад, при використанні пропорційного відбору [104] реалізуються наступні підкроки.

*Крок 6.1.* Визначення середнього значення фітнес-функції популяції. Визначення середнього значення фітнес-функції популяції  $f_M(N)$  як середнього арифметичного від значень фітнес-функції всіх індивідів здійснюється відповідно до рівняння [105]

$$f_M(N) = \frac{1}{Z_{\max}} \sum_{Z=1}^{Z_{\max}} f_Z(N). \quad (2.3)$$

*Крок 6.2.* Розрахунок коефіцієнта відбору. На даному кроці значення коефіцієнта відбору  $P_S^Z$  для кожної  $Z$ -ї особи (хромосоми) розраховується наступним чином [103]

$$P_S^Z(N) = \frac{f_Z(N)}{f_M(N)}. \quad (2.4)$$

*Крок 6.3.* Формування масиву хромосом для схрещування. На цьому етапі в залежності від значення  $P_S^Z$  формується масив хромосом, допущених до кросоверу (якщо  $P_S^Z > 1$ , тоді індивідуум вважається добре пристосований і допускається до схрещування).

*Крок 7.* Схрещування хромосом популяції. На даному етапі застосовується оператор кросоверу для хромосом, вибраних на попередньому кроці. Наприклад, при застосуванні випадкового відбору батьківських пар [106] виконуються підкроки 7.1-7.3.

*Крок 7.1.* Нумерація хромосом. На цьому етапі нумерація всіх представників поточної популяції здійснюється довільним чином.

*Крок 7.2.* Вибір першого з батьків. На даному етапі випадковим чином обирається число на інтервалі  $[0, 1]$  для кожної  $Z$ -ї хромосоми, починаючи з першої. Першим з батьків у парі буде перша хромосома, для якої дане випадкове число не менше ймовірності кросоверу  $P_C$  (встановленої на кроці 1).

*Крок 7.3.* Вибір другого з батьків. Перегляд популяції продовжується, починаючи з рішення, наступного за першим батьком (вибраного на попередньому кроці), доки знову випадково вибране число не стане не меншим за  $P_C$ . Хромосома, для якої виконується така умова, буде другою з батьків. Крок 7.3 продовжується до тих пір, поки не буде обрано необхідну кількість пар батьків.

*Крок 7.4.* Кросовер вибраних батьківських пар хромосом. На цьому етапі для кожної батьківської пари застосовується оператор кросоверу. Наприклад, при використанні одноточкового кросоверу [48] точка розриву хромосоми вибирається випадковим чином.

*Крок 8.* Мутація хромосом. Оператор мутації застосовується для хромосом, вибраних на кроці 6 відповідно до ймовірності мутації  $P_M$  (встановленої на кроці 1). Наприклад, при застосуванні простої мутації [34, 48] виконуються наступні підкроки.

*Крок 8.1.* Копіювання батьківських хромосом у хромосоми-нащадки. На цьому етапі батьківські хромосоми, допущені до мутацій, копіюються в хромосоми-нащадки.

*Крок 8.2.* Вибір гена, що мутує. Ген, що мутує, для кожної хромосоми, допущеної до мутації, вибирається випадковим чином [103, 106].

*Крок 8.3.* Вибір нового значення гена. Вибір нового значення гена, яке не дорівнює поточному, реалізується з набору всіх можливих значень параметра, відповідного гену, що мутує.

*Крок 9.* Формування нового покоління. На даному кроці формується нове покоління з елітних хромосом і хромосом-нащадків, отриманих шляхом схрещування та мутації. Потім здійснюється перехід до кроку 4.

*Крок 10.* Завершення процесу оптимізації.

У цілому, генетичні методи виявляються більш ефективними порівняно з традиційними методами оптимізації в умовах, коли досліджуваний простір пошуку має значний розмір, є негладким (з точками розриву) та мультимодальним (з кількома оптимумами). Враховуючи можливі шуми у цільовій функції пошуку та той факт, що завдання не вимагає точного визначення глобального оптимуму, генетичний пошук виявляється особливо ефективним для вирішення комбінаторних задач та знаходження оптимальних значень мультимодальних функцій. У реальних задачах часто важливо швидко знаходити прийнятне рішення, і генетичний пошук ефективно справляється з цією вимогою [48, 107].

*Переваги, недоліки та рекомендації до застосування GA.* Таким чином, до основних переваг генетичних алгоритмів можна віднести наступні [34]:

- концептуальна простота та логічна прозорість. Простота реалізації GA дозволяє легко розуміти та використовувати ці методи;
- менша ймовірність зациклення у локальних оптимумах. Використання популяційного підходу у GA допомагає уникнути зациклення у локальних мінімумах при вирішенні оптимізаційних задач;



- відсутність необхідності в специфічних знаннях. Дані алгоритми можуть застосовуватися без глибоких знань про вирішувану задачу, хоча додаткова інформація може підвищити їхню ефективність;
- простота кодування вхідних та вихідних даних. ГА легко використовуються з різними типами даних та об'єктів, надаючи гнучкість у виборі представлення інформації;
- можливість розпаралелювання. Роботу даних алгоритмів можна ефективно розпаралелювати, що поліпшує їхню швидкодію в розподілених обчисленнях;
- можливість адаптивності параметрів. ГА можуть адаптувати свої параметри до конкретних властивостей задачі, що дозволяє оптимізаційному процесу адекватно реагувати на різні зміни умов;
- можливість застосування в інших пошукових процедурах. ГА можуть бути легко комбіновані з іншими методами пошуку для загального підвищення ефективності процесу оптимізації.

Основними недоліками ГА є [46, 48]:

- суттєва залежність від регульованих параметрів. Ефективність ГА сильно залежить від налаштування параметрів, таких як розмір популяції, початкова точка пошуку, імовірнісні характеристики генетичних операторів тощо;
- висока ітеративність. Дані алгоритми вимагають виконання досить великої кількості ітерацій для збіжності, особливо при їх застосуванні у вирішенні складних задач оптимізації;
- передчасна збіжність. Недостатня різноманітність хромосом у популяції може призводити до передчасної збіжності, особливо при невеликому розмірі популяції;
- епістазис (наявність внутрішніх залежностей між генами у хромосомі). Взаємозалежність між генами може призводити до проблем, зокрема, до появи нащадків з низькою пристосованістю.

Крім того, для більш ефективного застосування GA при вирішенні різнотипних задач оптимізації доцільно дотримуватись наступних рекомендацій [46, 107]:

- перед використанням GA необхідно провести аналіз особливостей конкретної задачі оптимізації, типу простору пошуку, характеристик цільової функції та можливих обмежень;
- в залежності від властивостей кожної конкретної задачі та умов її вирішення, параметри GA (розмір популяції, типи та параметри відбору, кросоверу та мутації) повинні бути ретельно налаштовані на етапі ініціалізації;
- при створенні початкової популяції необхідно забезпечити достатню різноманітність хромосом для реалізації адекватного покриття простору пошуку та уникнення передчасної збіжності;
- для уникнення епістазису та зменшення його негативного впливу на процес оптимізації доцільно виконати групування залежних генів шляхом їх розташування близько один до одного в хромосомі;
- при реалізації GA доцільно застосовувати принцип елітизму для збереження кращих рішень та прискорення збіжності.

## **2.2 Методи штучних імунних систем**

*Базові аспекти оптимізаційних обчислень на основі штучних імунних систем (Artificial Immune Systems – AIS).* Алгоритми штучних імунних систем використовують ідеї та принципи, запозичені з імунології, імітуючи роботу біологічних імунних систем живих організмів для вирішення оптимізаційних задач [108]. Концепція AIS базується на емуляції природних механізмів імунної системи організму, таких як розпізнавання та боротьба з інфекціями, для розв'язання проблем оптимізації та пошуку рішень в складних задачах [60]. Дані алгоритми базуються на метаевристичному підході до пошуку, який використовує механізми

самоорганізації, адаптації та навчання для ефективного вирішення різноманітних завдань.

Основні принципи AIS включають в себе використання понять, таких як імунні клітини, антитіла, агенти, та їх взаємодія у системі [60]. Моделювання процесів клітинної взаємодії, розпізнавання антигенів та формування антитіл відображається через структуру алгоритмів штучних імунних систем. Ці алгоритми працюють на основі навчання імунної системи на прикладах, а також здатності до саморегулювання та самовідтворення. AIS можуть бути застосовані до широкого спектру задач оптимізації, включаючи пошук оптимальних рішень у складних просторах параметрів, вирішення задач класифікації та розпізнавання образів, а також в інших галузях, де необхідне ефективне вирішення оптимізаційних завдань. Застосування методів AIS відкриває широкі можливості для розв'язання проблем глобальної оптимізації, які важко або неможливо вирішити за допомогою класичних алгоритмів пошуку.

Для кращого розуміння роботи AIS-алгоритмів в процесі оптимізації розглянемо деякі терміни, що використовуються в імунології. Імунною системою живого організму називається система, яка об'єднує органи та тканини, що захищають організм від захворювань [60]. Призначення імунної системи живого організму полягає в тому, що вона ідентифікує та знищує чужорідні тіла, що потрапили в організм, та вдосконалюється, накопичуючи досвід боротьби з ними. Антигеном називається речовина, яка сприймається живим організмом як чужорідна, і від якої організм намагається захиститись. Щоб організм зміг захиститися від антигенів, у ньому за допомогою спеціальних імунних клітин виробляються антитіла. Антитілом називається речовина, яка розпізнає антиген та сприяє його знищенню. Якщо імунні клітини виробили антитіла, які змогли розпізнати антиген, інформація про ці антитіла зберігається в клітинах пам'яті. Клітиною пам'яті називається імунна клітина, яка зберігає в собі інформацію про нові антитіла, здатні розпізнати антиген, щоб наступного

разу, коли в організм потрапить такий же або схожий антиген, імунна система змогла працювати ефективніше [108].

В AIS-методах вектори-рішення представлені імунними клітинами (антитілами), афінність (ефективність боротьби з антигенами) яких безпосередньо залежить від значень цільової функції у розв'язуваній задачі. У свою чергу, чим менше отримане значення цільової функції для знайденого рішення, тим краще афінність відповідної імунної клітини. У процесі оптимізації AIS-алгоритми, як і інші еволюційні методи, оперують з множиною можливих розв'язків задачі, яка у даному випадку являє собою популяцією імунних клітин. Для знаходження оптимальних рішень у процесі глобального пошуку дані методи від ітерації до ітерації застосовують процедури клонування, мутації та селекції [108, 109].

Процедура клонування в методах штучних імунних систем є ключовим етапом, спрямованим на розмноження виділених клітин, які проявили високу афінність до антигенів [60]. Під час клонування імунних клітин їх кількість збільшується пропорційно до їхньої афінності. Цей процес імітує природний механізм клонування лімфоцитів, що активується відповіддю на зараження. Клоновані клітини, які успішно розпізнали антигени, стають домінуючими в популяції. Такий механізм дозволяє AIS концентрувати увагу на тих клітинах, які ефективно реагують на конкретні загрози, сприяючи підвищенню природної схильності системи до визначених стимулів та оптимізації її відповіді на антигени.

Після процедури клонування в AIS-алгоритмах використовується процедура мутації клонів. Мутація включає в себе введення випадкових змін в генетичний матеріал клонів, що моделює природний процес розвитку різноманітності [108]. Цей етап дозволяє системі експериментувати з різними варіантами імунних клітин та вдосконалювати їх афінність до антигенів. Мутація також підтримує генетичну різноманітність популяції, запобігаючи зацикленню на одному типі клітин і сприяючи пошуку оптимальних рішень. Поєднання клонування і мутації

в AIS-алгоритмах дозволяє системі ефективно адаптуватися до змінних умов і забезпечує динамічний розвиток відповіді на нові антигени.

Після етапів клонування та мутації застосовується процедура селекції для визначення кращих клонів-мутантів, які далі потраплять до нового покоління [109]. Селекція базується на показниках успішності кожного клону, який визначається афінністю до антигенів та ефективністю їхнього розпізнавання та усунення. Клони-мутанти, які проявляють високий рівень афінності та адекватну відповідь на антигени, мають більше шансів бути відібраними в процесі селекції. Ця процедура сприяє концентрації ресурсів на розвиток тих імунних клітин, які найефективніше справляються з поточними загрозами. Селекція гарантує еволюцію популяції в напрямку підвищення її адаптивних властивостей і покращення реакції на зміну зовнішніх умов.

*Узагальнений алгоритм штучних імунних систем.* Узагальнений AIS-алгоритм складається з наступних послідовних кроків.

*Крок 1.* Ініціалізація AIS-алгоритму. На етапі ініціалізації створюється початкова популяція з  $Z_{\max}$  імунних клітин (варіантів вектора шуканих параметрів  $X_p$ ), вибираються функція афінності  $f$  та критерій закінчення пошуку. Також, задаються параметри біологічних операторів клонування, мутації та селекції: число клітин пам'яті  $Z_m$ , число клітин у популяції з найгіршою афінністю  $Z_w$ , параметр оператора клонування  $\rho$  або  $Z_c$ , залежно від типу оператора клонування, а також параметр оператора мутації  $r$  [108, 110]. Створена початкова популяція  $P_0$  складається з  $Z_{\max}$  клітин, відповідних певним варіантам рішень задачі, змінні якого (параметри клітини) задаються випадковим чином у рамках множини можливих значень. У свою чергу, початкова популяція імунних клітин  $P_0$  також може бути описана виразом (2.2).

Функція афінності  $f$  подібно до фітнес-функції в генетичних алгоритмах також розраховується на базі цільової функції задачі оптимізації. В якості умови закінчення пошуку може бути обрано

досягнення оптимального значення афінності  $f \geq f_{\text{opt}}$  або виконання максимальної кількості ітерацій  $N_{\text{max}}$ .

*Крок 2.* Оцінювання клітин популяції та перевірка критерію закінчення пошуку. На даному етапі для кожного  $Z$ -го варіанта вектора  $\mathbf{X}_{PZ}$  ( $Z = \{1, 2, \dots, Z_{\text{max}}\}$ ) виконується розрахунок значення функції афінності  $f_Z(N)$ , а також здійснюється перевірка умови закінчення процесу оптимізації. При досягненні оптимального значення функції афінності  $f_Z(N) \geq f_{\text{opt}}$  для будь-якого вектора  $\mathbf{X}_{PZ}$  виконується перехід на крок 9, інакше – на крок 3.

*Крок 3.* Клонування клітин пам'яті. На даному кроці виконується впорядкування варіантів вектора  $\mathbf{X}_P$  (клітин популяції) за спаданням їх афінності, наприклад,  $\mathbf{X}_{P1}, \mathbf{X}_{P2}, \dots, \mathbf{X}_{PZ_{\text{max}}}$ , де  $f(\mathbf{X}_{P1}) = f_{\text{max}}(N)$ ,  $f(\mathbf{X}_{PZ_{\text{max}}}) = f_{\text{min}}(N)$  [110]. Далі з упорядкованої популяції (з новою нумерацією індексів варіантів векторів  $\mathbf{X}_{PZ}$  відповідно до рейтингу їхньої афінності) обирається  $Z_m$  перших векторів з найкращою афінністю. При використанні пропорційного оператора клонування для кожного  $Z$ -го вектора  $\mathbf{X}_{PZ}$  ( $Z = \{1, 2, \dots, Z_m\}$ ) генерується  $Z_l$  клонів на основі виразу (2.5) [108]

$$Z_l = \frac{\rho Z_{\text{max}}}{Z}, Z = (1, 2, \dots, Z_m). \quad (2.5)$$

У свою чергу, при застосуванні рівномірного оператора клонування для кожного  $Z$ -го вектора (клітини пам'яті) генерується заздалегідь обрана кількість  $Z_l = Z_c$  клонів [109].

*Крок 4.* Мутація клонів клітин пам'яті. Для всіх  $Z_l$  клонів кожної  $Z$ -ї клітини пам'яті ( $Z = \{1, 2, \dots, Z_m\}$ ) виконується процедура мутації – зміна  $r$  випадково вибраних параметрів клітини випадковим чином у рамках множини можливих значень [108, 111].

*Крок 5.* Оцінювання клонів-мутантів клітин пам'яті та перевірка критерію закінчення пошуку. На даному кроці для всіх  $Z_l$  клонів-мутантів кожної  $Z$ -ї клітини пам'яті ( $Z = \{1, 2, \dots, Z_m\}$ ) виконується розрахунок значення функції афінності  $f(N)$  та перевірка умови закінчення процесу

оптимізації. При досягненні оптимального значення функції афінності  $f(N) \geq f_{\text{opt}}$  для будь-якого клона-мутанта виконується перехід на крок 9, інакше – на крок 6.

*Крок 6.* Селекція та оновлення клітин. Серед усіх  $Z_l$  клонів-мутантів кожної  $Z$ -ї клітини пам'яті ( $Z = \{1, 2, \dots, Z_m\}$ ) обирається клон-мутант з найкращою афінністю  $f_{\text{max}}$  і порівнюється з цією клітиною пам'яті. Якщо афінність обраного клона-мутанта вища за афінність його клітини пам'яті, то даний клон-мутант стає новою клітиною пам'яті [109].

*Крок 7.* Формування нової популяції клітин. На цьому етапі виконується впорядкування оновлених векторів-рішень задачі за спаданням їх афінності [60]. Після цього із впорядкованої популяції видаляється  $Z_w$  останніх векторів з найгіршою афінністю з їх подальшою заміною новими варіантами, згенерованими випадковим чином.

*Крок 8.* Оцінювання нових клітин популяції та перевірка умови закінчення пошуку. На даному етапі для всіх згенерованих нових варіантів вектора  $\mathbf{X}_p$  виконується розрахунок значення функції афінності  $f(N)$ , а також здійснюється перевірка критерію закінчення процесу оптимізації. При досягненні оптимального значення функції афінності  $f(N) \geq f_{\text{opt}}$  для будь-якого нового варіанта вектора параметрів  $\mathbf{X}_p$  або виконанні максимальної кількості ітерацій  $N_{\text{max}}$  здійснюється перехід на крок 9, інакше – на крок 3.

*Крок 9.* Завершення процесу оптимізації.

На рис. 2.3 наведено блок-схему узагальненого алгоритму штучних імунних систем.

На сьогодні розроблено досить багато різних методів штучних імунних систем, які застосовують різні механізми клонування та мутації, різні типи клітин пам'яті, стратегії адаптації, гібридизації, динамічної зміни розміру популяції та інші процедури для прискорення збіжності та підвищення ефективності дослідження простору пошуку. Серед них

найбільш поширеними є методи CLONALG [112], НІА [113], I-opt-AiNet [114] та T-Cell Model [115].

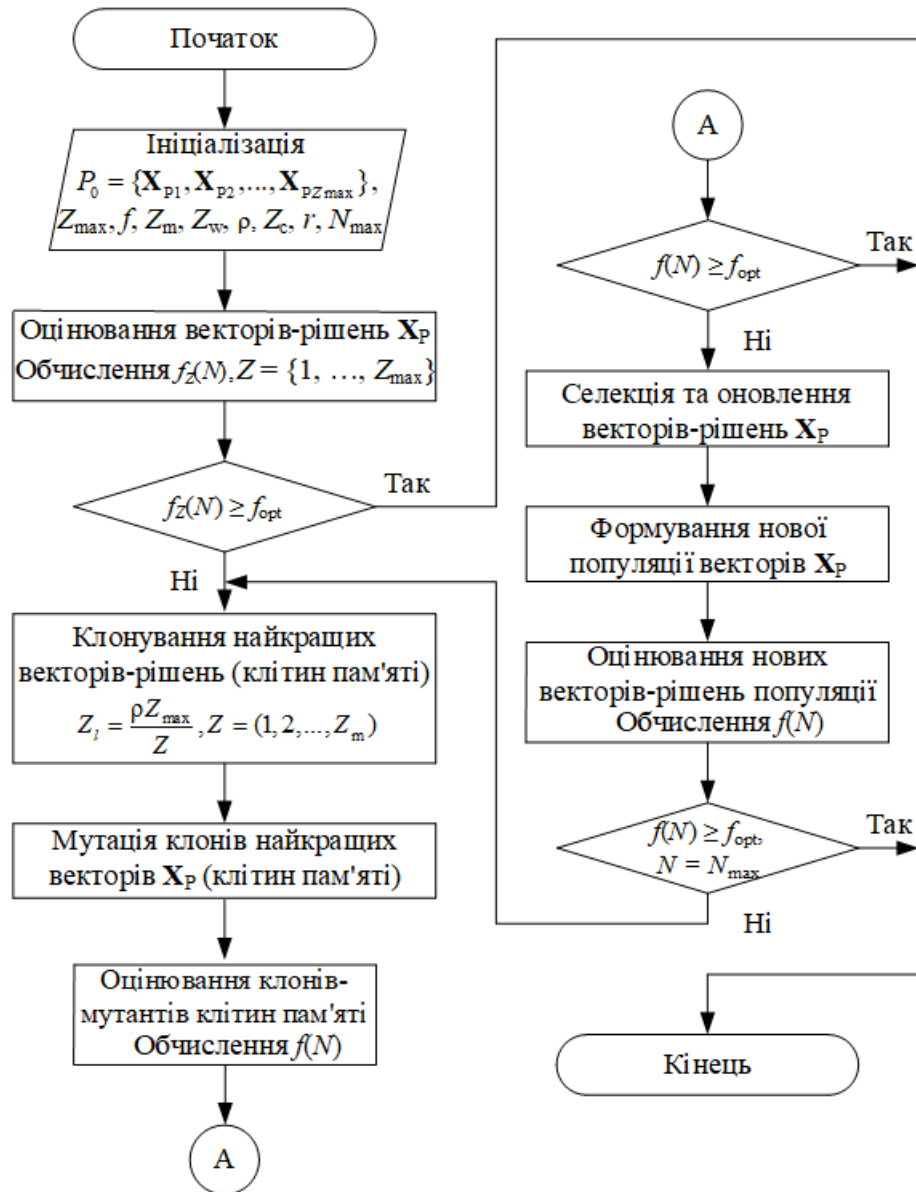


Рис. 2.3. Блок-схема узагальненого алгоритму штучних імунних систем

*Переваги, недоліки та рекомендації до застосування AIS-алгоритмів.* Загалом алгоритми штучних імунних систем мають такі ж самі переваги та недоліки, як і інші еволюційні методи, зокрема генетичні, якщо їх порівнювати з класичними методами пошуку. Однак, у даних алгоритмах відсутня операція схрещування, що дозволяє їх гнучкіше застосовувати при зміні розмірності вектора шуканих параметрів задачі в процесі



оптимізації. З іншого боку, в AIS-методах основною процедурою пошуку нових рішень є мутація, що вносить більш високу невизначеність у процес їх реалізації. У ряді випадків це може призводити до повільнішої збіжності і, як наслідок, до більшої кількості ітерацій порівняно з генетичними алгоритмами.

Для ефективного застосування даних алгоритмів також доцільно ретельно аналізувати особливості кожної конкретної задачі оптимізації, правильно налаштовувати регульовані параметри (розмір популяції, типи та параметри операторів клонування, мутації та селекції) та використовувати принцип елітизму [108, 111].

### **2.3 Методи оптимізації на основі біогеографії**

*Базові аспекти оптимізаційних обчислень на основі біогеографії (Biogeography-Based Optimization – BBO).* Біогеографічні оптимізаційні методи базуються на принципах біогеографії, яка вивчає розподіл організмів на Землі та взаємовпливи на їхнє довкілля [62]. Концепція BBO базується на ідеї обміну інформацією між різними «островами» або «середовищами» у просторі параметрів, призначеними для представлення потенційних рішень задачі оптимізації. Основною ідеєю алгоритмів BBO є моделювання розподілу видів (рішень) на різних «островах» (середовищах). Кожне середовище представляє собою підмножину параметрів, а види визначають рішення у цьому просторі. Процес обміну інформацією між середовищами відбувається через міграцію видів, що дозволяє алгоритму знаходити глобально оптимальні рішення.

Основою концепції BBO є ідея обміну інформацією між рішеннями, що допомагає підвищити різноманітність та ефективність пошуку [62, 116]. Цей обмін може бути здійснений через механізми, такі як міграція, мутація та інші еволюційні процеси. Іншим важливим аспектом є ідея збереження

рішень, які виявилися найкращими протягом деякого часу, що дозволяє уникнути передчасної збіжності до субоптимальних рішень.

Загалом, методи оптимізації на основі біогеографії відкривають нові можливості для ефективного пошуку оптимальних рішень у різних областях застосування, використовуючи випробувані в природі механізми міграції та еволюції.

Для глибшого усвідомлення особливостей реалізації оптимізаційних процесів за допомогою алгоритмів ВВО розглянемо основні поняття, що використовуються в теорії біогеографії.

Екосистема є складною функціональною одиницею, що включає всі живі організми, які взаємодіють між собою та з неживою природою у певному обмеженому просторі [62]. Вона представляє собою динамічну спільноту різних видів, які пристосовуються до місцевих умов та взаємодіють у складних взаємозв'язках. Екосистеми визначаються своєрідністю та взаємодією компонентів, що забезпечує баланс існування в середовищі.

Середовище в контексті біогеографії означає всі зовнішні умови, які впливають на живі організми. Це включає абіотичні фактори, такі як клімат, ґрунт, топографія, а також біотичні взаємодії з іншими видами. Середовище визначає придатність для існування різних видів та формує основу для їхнього розподілу та взаємодій [62].

Індекс придатності довкілля (Habitat Suitability Index – HSI) використовується для кількісної оцінки того, наскільки середовище сприятливе для існування конкретного виду чи популяції. Цей індекс враховує різноманітні аспекти, такі як кліматичні умови, наявність ресурсів та інші фактори, які впливають на придатність середовища для життя.

Імміграція в біогеографії вказує на процес введення чи вторгнення виду в нове середовище чи регіон. Цей процес може відбуватися як

природним шляхом, так і під впливом зовнішніх факторів, таких як зміни клімату чи людська діяльність.

Еміграція, навпаки, вказує на відхід виду з певного середовища чи регіону. Цей процес може бути спричинений різними факторами, включаючи конкуренцію за ресурси, зміни у середовищі чи природні катастрофи [62].

Мутація представляє собою генетичні зміни в організмах, що можуть виникати внаслідок помилок у реплікації ДНК чи впливу зовнішніх чинників. У біогеографії мутації можуть впливати на придатність виду до існування у нових умовах та сприяти його еволюції та адаптації до середовища, що динамічно змінюється.

У методах ВВО вектори-рішення представлені середовищами (островами) з характеризуючими їх змінними, а ефективність розв'язання задачі оптимізації оцінюється за допомогою індексу придатності довкілля [116, 117]. Основними механізмами ВВО-алгоритмів, що використовуються у процесі пошуку оптимальних рішень, є імміграція, еміграція, а також мутація видів [62]. У свою чергу, імміграція та еміграція застосовуються для обміну інформацією між видами у середовищах, що покращує диверсифікацію популяції та дозволяє враховувати інформацію з різних ділянок простору. Це сприяє ефективному виявленню оптимальних рішень. У той час як мутація в контексті ВВО-алгоритмів визначає процес випадкових змін параметрів рішень у популяції. Цей механізм сприяє введенню випадкових варіацій у рішення, що може бути корисним для експлорації нових областей простору рішень та виявлення потенційно оптимальних точок. Мутація допомагає забезпечити баланс між дослідженням та використанням знань у популяції, підвищуючи шанси знаходження глобальних оптимумів.

*Узагальнений алгоритм оптимізації на основі біогеографії.*  
Узагальнений ВВО-алгоритм складається з наступних послідовних кроків.

*Крок 1.* Ініціалізація ВВО-алгоритму. На етапі ініціалізації створюється екосистема з  $Z_{\max}$  середовищ існування (островів), обираються функція індексу придатності середовища  $f$  та умова закінчення пошуку. Крім того, задаються основні параметри алгоритму: максимальний рівень імміграції  $\lambda_{\max}$ , максимальний рівень еміграції  $v_{\max}$ , коефіцієнт оператора мутації  $r$ , максимально можлива кількість видів на острові  $N_{S\max}$ , що відповідає оптимальному значенню індексу придатності острова  $f_{\text{opt}}$ , а також коефіцієнт  $k_{NS}$  взаємозв'язку кількості видів  $N_S$  на острові від HSI [116]. Також, задаються залежності міграції видів від кількості видів на островах  $N_S$ :  $\lambda(N_S)$  та  $v(N_S)$ . У свою чергу, створена екосистема  $P_0$  складається з  $Z_{\max}$  островів, що відповідають певним варіантам вектора-рішення, змінні якого (що характеризують острів) задаються випадковим чином у рамках сформованої множини допустимих значень [62]. Індекс придатності  $f$  розраховується на основі цільової функції подібно фітнес-функції в GA або функції афінності в AIS-алгоритмах. В якості умови закінчення пошуку можна вибрати (а) досягнення оптимального значення індексу придатності  $f \geq f_{\text{opt}}$  або (б) виконання максимальної кількості ітерацій  $N_{\max}$ . Залежності міграції видів від їх кількості на островах  $N_S$  в певних модифікаціях методу можуть мати лінійний характер [117] з максимальними значеннями імміграції та еміграції рівними одиниці ( $\lambda_{\max} = v_{\max} = 1$ ), як показано на рис. 2.4.

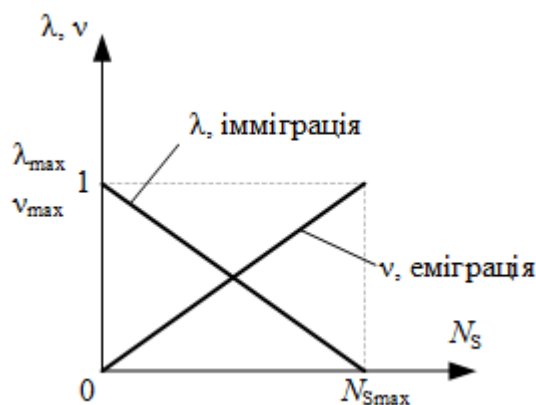


Рис. 2.4. Залежності міграції видів від кількості видів на островах

*Крок 2.* Оцінювання островів екосистеми та перевірка умови закінчення пошуку. На даному етапі для кожного  $Z$ -го варіанта вектора (острова)  $\mathbf{X}_{PZ}$  ( $Z = \{1, 2, \dots, Z_{\max}\}$ ) виконується розрахунок значення HSI  $f_Z(N)$ , а також здійснюється перевірка умови закінчення процесу оптимізації. При досягненні оптимального значення індексу придатності  $f_Z(N) \geq f_{\text{opt}}$  для будь-якого вектора шуканих параметрів або виконання максимальної кількості ітерацій  $N_{\max}$  здійснюється перехід на крок 7, інакше – на крок 3.

*Крок 3.* Визначення рівнів міграції видів. Для кожного  $Z$ -го варіанта вектора  $\mathbf{X}_P$  на основі значення індексу придатності  $f_Z$  визначається кількість видів  $N_{SZ}$  та відповідні рівні імміграції  $\lambda_Z$  та еміграції  $v_Z$ . При використанні лінійних залежностей кількість видів на острові та рівні міграції розраховуються на основі виразів (2.6)-(2.8) [62]

$$N_{SZ} = k_{NS} f_Z, Z = (1, 2, \dots, Z_{\max}); \quad (2.6)$$

$$\lambda_Z = \lambda_{\max} \left( 1 - \frac{N_{SZ}}{N_{S\max}} \right), Z = (1, 2, \dots, Z_{\max}); \quad (2.7)$$

$$v_Z = \frac{v_{\max} N_{SZ}}{N_{S\max}}, Z = (1, 2, \dots, Z_{\max}). \quad (2.8)$$

*Крок 4.* Міграція видів на островах. Міграція видів з острова на острів здійснюється наступним чином: випадково обрана змінна  $x_p$  вектора-рішення, обраного для імміграції, замінюється відповідною змінною іншого вектора-рішення, обраного для еміграції [116]. При виборі векторів для імміграції видів по черзі перебираються всі  $Z_{\max}$  варіантів (островів екосистеми) і кожен  $Z$ -й варіант вектора змінних може бути обраний з ймовірністю, пропорційною  $\lambda_Z$ . Таким же чином, для кожного обраного для імміграції вектора здійснюється вибір відповідного вектора еміграції видів серед інших  $Z_{\max} - 1$  векторів популяції. При цьому, кожен варіант вектора  $\mathbf{X}_P$  може бути обраний для еміграції видів з ймовірністю, пропорційною  $v_Z$ .

*Крок 5.* Оцінювання островів екосистеми та перевірка умови закінчення пошуку. На цьому етапі для кожного  $Z$ -го вектора  $\mathbf{X}_{PZ}$  ( $Z = \{1, 2, \dots, Z_{\max}\}$ ) виконується розрахунок значення HSI  $f_Z(N)$ , а також здійснюється перевірка умови закінчення процесу оптимізації. При досягненні оптимального значення індексу придатності  $f_Z(N) \geq f_{\text{opt}}$  для будь-якого варіанта вектора  $\mathbf{X}_{PZ}$  виконується перехід на крок 7, інакше – на крок 6.

*Крок 6.* Мутація видів на островах. Для кожного  $Z$ -го острова екосистеми з ймовірністю  $P_M$  здійснюється процедура мутації видів – зміна випадково обраної змінної вектора  $\mathbf{X}_P$  випадковим чином у рамках сформованої множини допустимих значень. При цьому ймовірність мутації  $P_M$  розраховується на основі виразів (2.9)-(2.12) [62]

$$P_M = r \left( \frac{1 - P_Z}{P_{Z_{\max}}} \right); \quad (2.9)$$

$$P_Z = \frac{v_{N_S}}{\sum_{N_S=1}^{N_{S_{\max}}+1} v_{N_S}}; \quad (2.10)$$

$$v_{N_S} = \begin{cases} \frac{N_{S_{\max}}!}{(N_{S_{\max}} + 1 - N_S)!(N_S - 1)!}, & \text{при } N_S \in (1, \dots, N'_S); \\ v_{N_{S_{\max}} + 2 - N_S}, & \text{при } N_S \in (N'_S + 1, \dots, N_{S_{\max}} + 1); \end{cases} \quad (2.11)$$

$$N'_S = \text{ceil} \left( \frac{N_{S_{\max}} + 1}{2} \right), \quad (2.12)$$

де  $P_Z$  – ймовірність, що  $Z$ -й острів має  $N_S$  видів;  $v_{N_S}$  – параметр, який характеризує взаємозв'язок кількості видів  $N_S$  і ймовірності  $P_Z$ ;  $N'_S$  – найближче ціле число, яке більше або дорівнює  $(N_{S_{\max}} + 1)/2$  (розраховується за допомогою операції  $\text{ceil}$ ).

Далі виконується перехід на крок 2.

*Крок 7.* Завершення процесу оптимізації.

На рис. 2.5 наведено блок-схему узагальненого біогеографічного алгоритму.

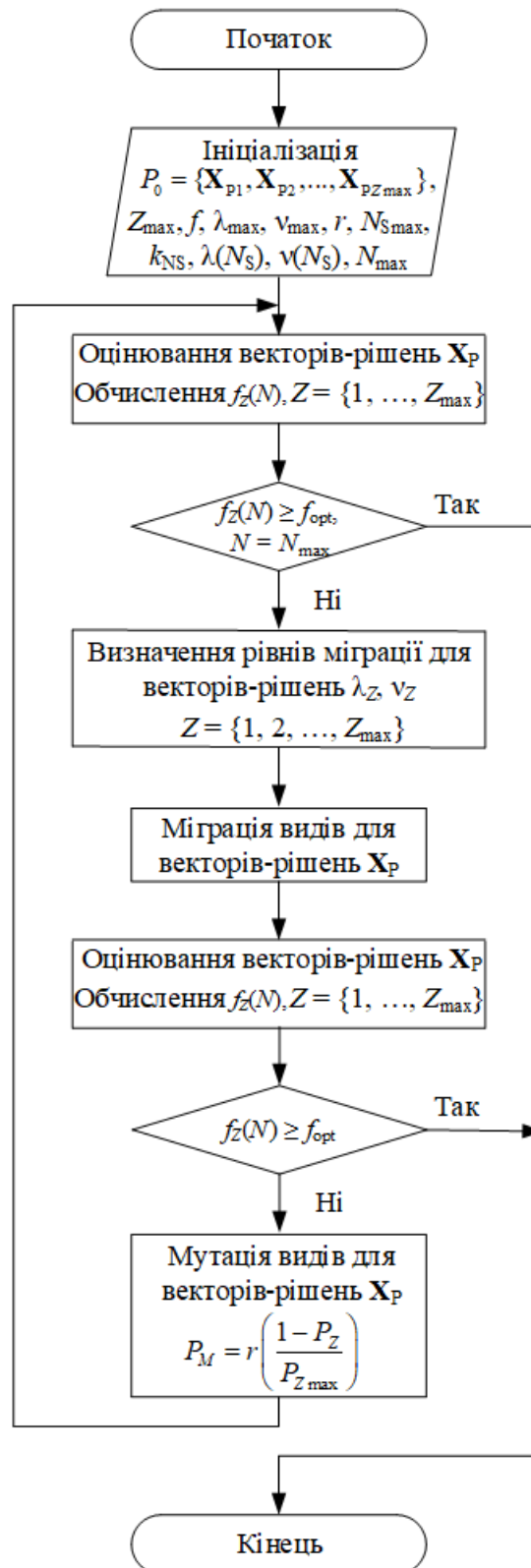


Рис. 2.5. Блок-схема узагальненого біогеографічного алгоритму

*Переваги, недоліки та рекомендації до застосування ВВО-алгоритмів.* У цілому алгоритми оптимізації на основі біогеографії мають такі ж самі переваги та недоліки, як і генетичні та AIS-алгоритми, якщо їх порівнювати з традиційними пошуковими методами. Однак, у даних алгоритмах замість операції схрещування застосовуються процедури міграції (імміграції та еміграції), що надає деяку додаткову гнучкість при оптимізації, оскільки дозволяє векторам-рішенням виходити за межі своїх поточних ділянок простору пошуку, де вони можуть значно покращувати значення індексу придатності [62]. Такий підхід особливо корисний у випадках, коли глобальний оптимум забезпечується розташуванням параметрів у різних частинах пошукового простору, і дозволяє більш ефективно наближатися до оптимального рішення. Крім того, ряд проведених досліджень показує, що ВВО-алгоритми мають більший потенціал для гнучкого налаштування та показують кращі результати за збіжністю при суттєвому збільшенні розмірності задачі оптимізації у порівнянні з GA [118].

Для ефективного застосування ВВО-алгоритмів також доцільно здійснювати попередній аналіз особливостей обраних задач оптимізації, використовувати принцип елітизму та ретельно налаштовувати регульовані параметри (розмір популяції, типи та параметри операторів імміграції, еміграції та мутації) [110, 111]. Також, в якості додаткових налаштувань можна застосовувати різні нелінійні функції для залежностей міграції видів від їх кількості на островах [118].

## **2.4 Методи мурашиних колоній**

*Базові аспекти мультиагентної оптимізації.* Мультиагентна оптимізація базується на концепції мультиагентних систем, в яких низка агентів співпрацюють чи конкурують між собою для вирішення складних



задач оптимізації [48]. В основі мультиагентної оптимізації лежить ідея імітації природних взаємодій та колективного інтелекту в природі.

У мультиагентних системах агенти представляють собою незалежні сутності, які можуть взаємодіяти для досягнення групових або індивідуальних цілей [34]. Вони можуть обмінюватися інформацією, виконувати операції або взаємодіяти іншими способами. Важливим аспектом мультиагентних систем є застосування групової поведінки агентів. Синергетичний ефект, що виникає від взаємодії, може призводити до успішного розв'язання проблем, які важко або неможливо вирішити індивідуально. Крім того, мультиагентні системи використовують розподілені обчислення, де кожен агент має обмежені можливості та обсяг інформації. Це дозволяє збільшити масштабність та ефективність застосованих алгоритмів. При виконанні задач оптимізації агенти системи використовуються для знаходження оптимальних рішень у просторі пошуку. У свою чергу, агенти можуть пристосовуватися та оптимізувати свою поведінку, базуючись на локальній інформації та сприйнятті головної цілі.

Методи мультиагентної інтелектуальної оптимізації відзначаються біонічною природою, а саме, вони ґрунтуються на моделюванні поведінки різних видів живих організмів, таких як комахи, птахи та інші тварини [48]. Ця поведінка є колективною, що дозволяє досягти наявності так званого «колективного інтелекту». Таким чином, для розуміння основних принципів колективного інтелекту достатньо вивчити принципи поведінки різних групових комах, таких як мурахи, бджоли, терміти та інші. Ці комахи формують колонії, де основою їхньої взаємодії є самоорганізація. У свою чергу, самоорганізація представляє собою систему динамічних механізмів, які забезпечують глобальне регулювання системи за рахунок взаємодії компонентів на нижньому рівні без прямої взаємодії між ними. Цей процес базується на таких основних складових: додатний зворотний

зв'язок, від'ємний зворотний зв'язок, нестійкість додатного зворотного зв'язку та множинна взаємодія між агентами [34].

Додатний зворотний зв'язок забезпечується використанням простих емпіричних прийомів, спрямованих на створення рішень. Наприклад, фуражування мурах може включати збільшення феромонів на шляху до джерела їжі, утворюючи мережу можливих шляхів. Аналогічно, у бджіл, позитивний зворотний зв'язок виявляється в їхній здатності летіти до зазначеного джерела ресурсів на підставі інформації від інших особин.

Від'ємний зворотний зв'язок компенсує додатний та призводить до стабілізації колективної поведінки. Наприклад, при фуражуванні мурах, від'ємний зворотний зв'язок може бути викликаний обмеженою кількістю фуражирів, виснаженням джерела їжі або випаровуванням феромону.

Нестійкість додатного зворотного зв'язку базується на стохастичних правилах у поведінці соціальних комах, де випадкові відхилення в рішеннях служать основою для виникнення нових рішень. Хаотичність може бути корисною, дозволяючи колонії виявляти нові рішення, наприклад, коли фуражир випадково збивається з курсу і виявляє нові джерела харчування.

Множинна взаємодія між агентами (особинами) призводить до появи нових ефективних та стійких рішень. Взаємодія у бджіл виявляється у тому, що інформація про знайдені ресурси, однією особиною, стає доступною всім іншим за допомогою так званого виляючого танцю.

Поміж основних аспектів, характеризуючи самоорганізацію, можна виділити ще такі характеристики, як динамічність, емерджентність, нелінійна взаємодія та мультистійкість [48].

Динамічність полягає у постійній взаємодії між окремими агентами системи та їхнім навколишнім середовищем у процесі вирішення оптимізаційних задач. Ця взаємодія формує додатний зворотний зв'язок, який утворює колективні групи та сприяє їхньому існуванню. Водночас від'ємний зворотний зв'язок зменшує вплив додатного зв'язку.

Емерджентність полягає у тому, що властивості систем з самоорганізацією виявляються більш складними, ніж властивості її окремих агентів. Ці властивості утворюються у системі через наявність нелінійних взаємодій між агентами.

У свою чергу, взаємодії між агентами в системах, які самоорганізуються, є нелінійними, що веде до розгалуження в їхній поведінці. Таке розгалуження породжує нові стійкі рішення при зміні конкретних параметрів системи, що призводить до якісного поліпшення колективної поведінки агентів.

Мультистійкість забезпечує наявність різних стійких станів системи при її фіксованих параметрах в залежності від початкового стану та випадкових відхилень.

Для кращого розуміння функціонування колективного інтелекту розглянемо основні функції, які соціальні комахи та тварини виконують під час розв'язання різних завдань. Ці функції включають координацію, кооперацію, колективне прийняття рішень і спеціалізацію [48]. Реалізація даних функцій разом дозволяє колонії успішно досягати поставленої мети.

Координація – це організація завдань окремих агентів у часі і просторі для вирішення поставленої перед колонією задачі. Ця функція призводить до просторово-часового розподілу дій та результатів для досягнення мети.

Кооперація полягає у спільному виконанні завдання агентами, яке кожен з них не зміг би виконати окремо. Об'єднання зусиль дозволяє успішно вирішувати загальні завдання, виходячи за межі можливостей окремих агентів.

Колективне прийняття рішень є механізмом, що реалізується при зіткненні колонії з проблемою вибору. Цей механізм здійснює колективний вибір одного з можливих рішень.

Спеціалізація полягає у наявності різних типів агентів у системі, які можуть виконувати окремі спеціалізовані операції в процесі вирішення задачі.

Отже, організацію колективної поведінки у соціальних комах та тварин можна розглядати як сукупність чотирьох розглянутих складових: координації, кооперації, колективного прийняття рішень і спеціалізації [34]. Кожна з цих складових системи проявляється на колективному рівні завдяки непрямій взаємодії між агентами, що забезпечує можливість оцінювання зовнішньої інформації відповідно до двох основних напрямків.

1. Координація та спеціалізація визначають формування просторових, часових і соціальних зв'язків, що виникають у процесі функціонування колонії. Координація регулює просторово-часове розташування окремих особин, в той час як спеціалізація визначає розподіл їхніх дій.

2. Кооперація та колективне прийняття рішень слугують засобами, що дозволяють колонії адаптуватися до змін у навколишньому середовищі. Механізми колективного прийняття рішень визначають процеси прийняття рішень в колонії, тоді як кооперація забезпечує засоби для виходу за рамки можливостей окремих агентів.

Спільно вказані аспекти у поведінці соціальних комах утворюють враження, що колонія організовано планує свою діяльність з метою досягнення поставлених цілей.

*Узагальнений метод мурашиних колоній.* Серед групи мультиагентних методів колективного інтелекту одними з найбільш поширених є методи мурашиних колоній (Ant Colony Optimization – ACO) [65, 66]. Базовий метод мурашиних колоній розроблений М. Dorigo для дискретних множин допустимих рішень і застосовується для вирішення різнотипних комбінаторних задач [119]. У свою чергу, в роботі [120] запропонована модифікація наведеного методу для оптимізації неперервних функцій. Методи мурашиних колоній базуються на ідеї моделювання колективного інтелекту колонії мурах при пошуку

оптимального маршруту у змінному середовищі. При проходженні по певній ділянці шляху кожна мураха відкладає певну кількість феромону, змінюючи оточуючий простір. Чим сильніше концентрація феромону на стежці, тим більш привабливою вона є для інших мурах. Таким чином, мурахи дізнаються, які маршрути найчастіше використовуються колонією. Крім того, в процесі пошуку мурахи керуються також об'єктивною інформацією про навколишнє середовище.

Блок-схема узагальненого методу мурашиних колоній наведена на рис. 2.6 [65, 119].

Основними етапами наведеного методу є: ініціалізація, рух мурах (агентів) по ребрах та вузлах графу, оцінювання знайдених шляхів, перевірка виконання умови закінчення роботи методу, збільшення феромону на ребрах графу, випаровування та оновлення феромону на ребрах графу.

Для вирішення задачі дискретної оптимізації за допомогою узагальненого методу АСО, простір пошуку даної задачі необхідно подати у вигляді графа рішень, що складається з вузлів та ребер між вузлами. При цьому, фіксовані дискретні значення змінних вектора-рішення задачі оптимізації відповідатимуть певним вузлам даного графа. Також, дані вузли виступатимуть пунктами призначення, по яких будуть рухатися мурахи (агенти) колонії через певні ребра в процесі реалізації АСО-методу.

Шлях кожного  $Z$ -го агента на кожній  $N$ -й ітерації проходить через певні вузли та ребра графа рішень. Після проходження  $Z$ -м агентом визначених вузлів даного графа на кожній окремій ітерації з номером  $N$  формується відповідний вектор-рішення  $\mathbf{X}_Z(N)$  ( $Z = 1, \dots, Z_{\max}$ ;  $N = 1, \dots, N_{\max}$ ), змінні якого відповідають відвіданим вузлам. Довжина шляху, який пройшов  $Z$ -й агент на  $N$ -й ітерації, визначається розрахованим значенням цільової функції  $J_Z(N)$  задачі оптимізації, що відповідає знайденому рішенню.

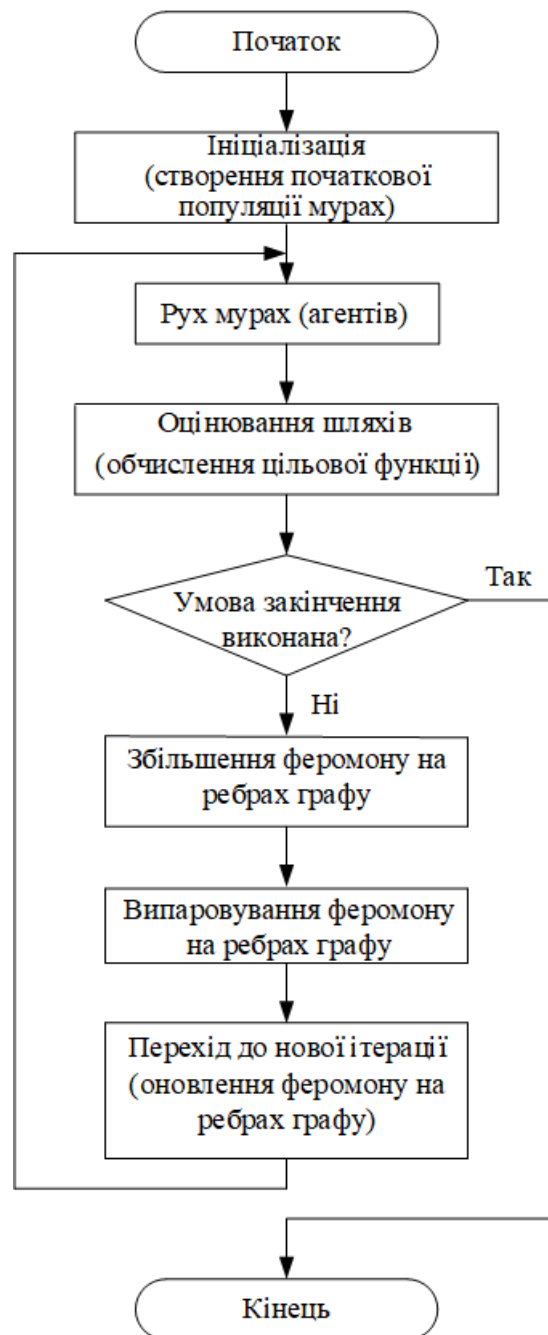


Рис. 2.6. Блок-схема узагальненого методу мурашиних колоній

Наприклад, при розв'язанні задачі комівояжера кожен  $Z$ -й агент на кожній  $N$ -й ітерації повинен відвідати всі наявні вузли графа по одному разу. У даному випадку вузли відповідають певним пунктам призначення, обраним у задачі, а значення цільової функції  $J_Z(N)$  розраховується як сумарна довжина пройденого шляху  $Z$ -го агента на  $N$ -й ітерації після відвідання всіх вузлів [48].

Далі розглянемо більш докладно основні кроки узагальненого методу мурашиних колоній.

*Крок 1.* Ініціалізація АСО-алгоритму. На цьому етапі здійснюється побудова графа рішень для задачі оптимізації та вибирається цільова функція  $J$  і її оптимальне значення  $J_{\text{opt}}$ . Також, створюється популяція агентів, і встановлюються основні параметри алгоритму: кількість агентів у популяції  $Z_{\text{max}}$ , регульовані параметри  $\alpha$ ,  $\beta$ ,  $Q$  і  $\rho$ , максимальна кількість ітерацій  $N_{\text{max}}$ . Крім того, на цьому кроці здійснюється попереднє розподілення агентів по вузлах графа, та встановлюється невелике початкове позитивне значення феромону на ребрах до початку руху агентів. Наприклад, при вирішенні задачі комівояжера агенти колонії рівномірно розподіляються між вузлами, щоб всі вузли мали однакові шанси стати відправною точкою.

*Крок 2.* Рух агентів по вузлах графа рішень. На даному етапі послідовно здійснюється рух кожного  $Z$ -того агента популяції по всіх необхідних вузлах графа, кількість яких визначається умовою задачі. Якщо  $Z$ -й агент ще не закінчив рух і знаходиться у вузлі  $j$ , то для визначення наступного ребра шляху  $jq$  розраховується ймовірність  $P_{Zjq}(N)$  переходу даного агента у вузол  $q$  згідно з виразом (2.13)

$$P_{Zjq}(N) = \frac{[\tau_{jq}(N)]^\alpha \cdot [\eta_{jq}]^\beta}{\sum_{k \in G} [\tau_{jq}(N)]^\alpha \cdot [\eta_{jq}]^\beta}, \quad (2.13)$$

де  $\tau_{jq}(N)$  – інтенсивність феромону на ребрі між вузлами  $j$  і  $q$  на ітерації  $N$ ;  $\eta_{jq}$  – величина, зворотна відстані  $D_{jq}$  між вузлами  $j$  і  $q$ ;  $\alpha$  – регульований параметр, який задає відносну значущість сліду феромону на ребрі при виборі наступного вузла;  $\beta$  – регульований параметр, який задає відносну значущість відстані  $D_{jq}$  між вузлами  $j$  і  $q$ ;  $G$  – множина вузлів, які ще не були відвідані агентом;  $k$  – номер обраного вузла з множини  $G$ .

У свою чергу,

$$\eta_{jq} = \frac{1}{D_{jq}}. \quad (2.14)$$

Крок 2 продовжується, поки кожен  $Z$ -й агент популяції не завершить свій шлях.

*Крок 3.* Обчислення значень цільової функції. На даному етапі для кожного вектора-рішення  $\mathbf{X}_Z(N)$ , сформованого кожним  $Z$ -м агентом популяції ( $Z = 1, \dots, Z_{\max}$ ) під час його руху на кроці 3, виконується розрахунок значення цільової функції задачі  $J_Z(N)$ .

*Крок 4.* Перевірка на закінчення роботи методу. На даному етапі здійснюється вибір найкращого вектора-рішення  $\mathbf{X}_{best}(N)$ , сформованого під час руху агентів на кроці 2, за найменшим значенням цільової функції  $J_{\min}(N)$  із значень, розрахованих на кроці 3. Для обраного найкращого вектора-рішення  $\mathbf{X}_{best}(N)$  здійснюється перевірка на закінчення роботи методу, що може визначатися досягненням оптимального значення цільової функції ( $J_{\min}(N) \leq J_{opt}$ ) або обмеженням на максимальну кількість ітерацій  $N_{\max}$ . Закінчення роботи методу може вважатися досягнутим, якщо протягом певної кількості ітерацій значення цільової функції  $J$  не зменшувалося. Якщо ця перевірка дала позитивний результат, здійснюється перехід на крок 8. У протилежному випадку виконується перехід на крок 5.

*Крок 5.* Збільшення феромону на ребрах графа рішень. На даному етапі на основі значень цільової функції  $J_Z(N)$ , розрахованих на кроці 3, обчислюється кількість феромону  $\Delta\tau_{Zjq}(N)$ , залишене на кожному ребрі  $jq$  графа кожним  $Z$ -м агентом ( $Z = 1, 2, \dots, Z_{\max}$ ) під час руху на кроці 2 на основі виразу (2.15)

$$\begin{cases} \Delta\tau_{Zjq}(N) = \frac{Q}{J_Z(N)}, \text{ при } jq \in H_Z(N); \\ \Delta\tau_{Zjq}(N) = 0, \text{ при } jq \notin H_Z(N), \end{cases} \quad (2.15)$$



де  $Q$  – регульований параметр, який відноситься до кількості феромону, залишеного на шляху агентом;  $H_Z(N)$  – набір ребер (шлях), який пройшов  $Z$ -й агент на ітерації  $N$ .

У свою чергу, параметр  $Q$  доцільно вибирати одного порядку з оптимальним значенням цільової функції  $J_{\text{opt}}$  [48]. Таким чином, значення феромону  $\Delta\tau_{Zjq}(N)$ , залишеного  $Z$ -м агентом на ребрі  $jq$  на ітерації  $N$ , зворотно-пропорційно залежить від значення цільової функції  $J_Z(N)$ : мале значення цільової функції характеризується високою концентрацією феромону, велике – низькою.

Далі, розраховані значення кількості феромону  $\Delta\tau_{Zjq}(N)$ , відкладеного кожним  $Z$ -м агентом на кожному ребрі  $jq$  при русі, використовуються для збільшення феромону на ребрах графа згідно з виразом (2.16)

$$\tau_{Zjq}(N) = \tau_{Zjq}(N-1) + \rho \cdot \sum_{Z=1}^{Z_{\max}} \Delta\tau_{Zjq}(N), \quad (2.16)$$

де  $\rho$  – коефіцієнт кількості феромону, який залишають агенти на шляху (доцільно вибирати у діапазоні від 0 до 1).

*Крок 6.* Випаровування феромону на ребрах графа. На даному етапі для кожного ребра графа  $jq$  застосовується операція випаровування феромону згідно з виразом (2.17), та обчислюється його фінальне значення  $\tau_{jqF}$

$$\tau_{jqF}(N) = \tau_{jq}(N) \cdot (1 - \rho). \quad (2.17)$$

*Крок 7.* Перехід на наступну ітерацію методу. На даному етапі здійснюється повернення всіх агентів популяції на початкові вузли графа, а також оновлення феромону на кожному ребрі  $jq$  згідно з виразом (2.18)

$$\tau_{jq}(N+1) = \tau_{jqF}(N). \quad (2.18)$$

Після цього здійснюється перехід на крок 2.

*Крок 8.* Завершення процесу оптимізації.

На даний момент розроблено досить багато різних модифікацій методу мурашиних колоній, серед яких: метод на основі ранжування

(AS<sub>rank</sub>) [121], метод мурашиних систем, який використовує елітну стратегію [122], максі-мінний метод мурашиних систем (MAX-MIN AS – MMAS) [123], метод системи мурашиної колонії [124], покращений метод 3-opt [125] та інші.

*Переваги, недоліки та рекомендації до застосування АСО-методів.*

Проаналізувавши викладені основні принципи та особливості методу мурашиних колоній, можна відзначити його наступні переваги [34]:

- гарантія збіжності до оптимального рішення;
- здатність застосовувати метод у динамічних середовищах, які адаптуються до змін зовнішніх умов;
- мультиагентність, що дозволяє одночасно працювати з великою множиною рішень;
- використання пам'яті всієї колонії, що досягається за рахунок моделювання виділення феромону;
- стохастична природа, що включає випадковий пошук та дозволяє уникати зациклення у локальних мінімумах;
- вища швидкість знаходження оптимальних рішень, порівняно з традиційними методами, при застосуванні у дискретних задачах великої розмірності;
- відносна простота застосування до широкого спектру задач оптимізації.

До недоліків методів АСО відносяться наступні [48]:

- складність теоретичного аналізу, оскільки остаточне рішення формується за результатами послідовності випадкових процесів, та розподіл ймовірностей змінюється протягом ітерацій. Дослідження властивостей та ефективності методів АСО значною мірою ґрунтуються на експериментах;
- складність визначення часу збіжності;
- висока ітеративність методу, що може впливати на час виконання оптимізаційних завдань;

- значна залежність отриманих результатів від регульованих параметрів методу, які визначаються експериментальним шляхом.

Враховуючи розглянуті переваги та недоліки АСО-методу, можна зробити висновок, що його використання є більш доцільним для розв'язання завдань дискретної оптимізації та в динамічних сценаріях, оскільки даний метод здатен досить легко адаптуватися до змінних умов [34]. Також, слід відзначити, що ефективність АСО-методу суттєво підвищується у випадках, коли розмірність простору пошуку є досить великою. Що стосується базового методу мурашиних колоній, то він має обмежене застосування для вирішення задач безперервної оптимізації. Однак, за допомогою впровадження певних додаткових пошукових процедур, що ґрунтуються на ймовірнісному підході, він може бути досить легко розширений та адаптований до розв'язання різного класу задач як дискретної, так і неперервної оптимізації.

## **2.5 Методи рою частинок**

*Базові аспекти оптимізаційних обчислень із застосуванням рою частинок.* Методи рою частинок (Particle Swarm Optimization – PSO) базуються на аналогії з колективною поведінкою зграй тварин у природі, зокрема, роїв птахів, косяків риб та інших колективних організмів. Такі природні системи проявляють певний ройовий інтелект у процесі їх руху в просторі при пошуку найкращих джерел їжі [48, 70]. Підходячи до оптимізаційних задач біоінспіровані PSO-методи використовують принципи кооперативної поведінки та обміну інформацією.

Біологічні основи PSO-методу визначаються взаємодією агентів, що рухаються у просторі параметрів, подібно до частинок у рої. Цей підхід віддзеркалює механізм самоорганізації та взаємодії, який дозволяє природним системам адаптуватися до змін у зовнішньому середовищі та ефективно вирішувати складні задачі оптимізації. При розгляді

біологічних аспектів PSO-алгоритмів слід враховувати принципи координації, соціальної конкуренції та адаптивності частинок. Заснований на аналізі колективної поведінки, PSO використовує модель взаємодії, що дозволяє агентам пристосовуватися до змін у динаміці задачі оптимізації. Такий підхід стимулює ефективний обмін інформацією між агентами, сприяючи знаходженню оптимальних рішень у просторі параметрів [71].

Агенти у рої частинок проявляють досить простий спосіб взаємодії: вони націлюються на вдосконалення власних позицій та на перевершення досягнень сусідніх частинок. Таким чином, емергентна властивість цієї системи полягає в дослідженні оптимальних областей багатовимірного пошукового простору. Моделі взаємодіючих особин рою в PSO-методах представлені агентами-частинками єдиної мультиагентної системи, які для знаходження оптимального рішення задачі рухаються у багатовимірному просторі пошуку, обмінюючись між собою інформацією [70]. Поточне положення  $Z$ -ї частинки рою в  $m$ -мірному просторі пошуку на ітерації  $N$  є потенційним розв'язком задачі оптимізації та, відповідно, певним значенням вектору шуканих параметрів  $\mathbf{X}$ .

Зміна поточної позиції  $Z$ -ї частинки в процесі ітеративного пошуку здійснюється шляхом додавання до неї вектора швидкості  $\mathbf{V}$  [48]

$$\mathbf{X}_Z^j(N+1) = \mathbf{X}_Z^j(N) + \mathbf{V}_Z^j(N+1), Z = \{1, \dots, Z_{\max}\}, j = \{1, \dots, m\}, \quad (2.19)$$

де  $Z_{\max}$  – загальна кількість частинок рою;  $j$  – номер параметра, що оптимізується;  $m$  – загальна кількість параметрів, які одночасно оптимізуються у розв'язуваній задачі.

Вектор швидкості  $\mathbf{V}$  в різних PSO-методах визначається різним чином [71]. Найбільш поширеним є PSO-метод *Gbest* [48], в якому кожна частинка пов'язана з усіма іншими частинками рою і тяжіє до найкращої позиції всього рою. Відповідно до цього методу вектор швидкості  $Z$ -ї частинки  $\mathbf{V}$  розраховується на основі залежності (2.20)

$$\begin{aligned} \mathbf{V}_Z^j(N+1) = & \mathbf{V}_Z^j(N) + \mathbf{C}_1^j \mathbf{R}_1^j(N) [\mathbf{L}_Z^j(N) - \mathbf{X}_Z^j(N)] + \\ & + \mathbf{C}_2^j \mathbf{R}_2^j(N) [\mathbf{G}^j(N) - \mathbf{X}_Z^j(N)], \end{aligned} \quad (2.20)$$

де  $\mathbf{C}_1^j$ ,  $\mathbf{C}_2^j$  – вектори прискорень частинок для параметрів, що оптимізуються,  $j = \{1, 2, \dots, m\}$ ;  $\mathbf{R}_1^j$ ,  $\mathbf{R}_2^j$  – вектори випадкових чисел з діапазону  $[0, 1]$ , що вносять стохастичність у реалізацію методу,  $\mathbf{R}_1^j, \mathbf{R}_2^j = \text{rand}[0, 1]$ ;  $\mathbf{L}_Z^j$  – найкраща позиція, яка знайдена  $Z$ -ю частинкою в процесі пошуку, починаючи з першої ітерації;  $\mathbf{G}^j$  – глобальна найкраща позиція всього рою частинок, яка знайдена з початку процесу пошуку.

У свою чергу, найкраща позиція  $Z$ -ї частинки  $\mathbf{L}_Z^j$  на ітерації  $N + 1$  визначається на основі цільової функції  $J$  задачі оптимізації відповідно до залежності (2.21) [48]

$$\mathbf{L}_Z^j(N+1) = \begin{cases} \mathbf{L}_Z^j(N), & \text{ЯКЩО } J(\mathbf{X}_Z^j(N+1)) \geq J(\mathbf{L}_Z^j(N)); \\ \mathbf{X}_Z^j(N+1), & \text{ЯКЩО } J(\mathbf{X}_Z^j(N+1)) < J(\mathbf{L}_Z^j(N)). \end{cases} \quad (2.21)$$

Глобальна найкраща позиція всього рою  $\mathbf{G}^j$  на ітерації  $N$  обирається з найкращих позицій усіх частинок рою  $\mathbf{L}_Z^j, Z = \{1, 2, \dots, Z_{\max}\}$  за найменшим значенням їх цільової функції  $J$  [71]

$$\begin{aligned} \mathbf{G}^j(N) \in & \{\mathbf{L}_1^j(N), \dots, \mathbf{L}_Z^j(N), \dots, \mathbf{L}_{Z_{\max}}^j(N)\} \mid J(\mathbf{G}^j(N)) = \\ & = \min\{J(\mathbf{L}_1^j(N)), \dots, J(\mathbf{L}_Z^j(N)), \dots, J(\mathbf{L}_{Z_{\max}}^j(N))\}. \end{aligned} \quad (2.22)$$

Для підвищення стабільності пошукових процедур PSO-методів та перешкодження надмірно швидкому зростанню швидкостей руху частинок доцільно застосовувати механізм обмеження [48], згідно з яким вектор реальних швидкостей  $Z$ -ї частинки  $\mathbf{V}_Z^j$  на ітерації  $N + 1$  визначається на основі виразу (2.23)

$$\mathbf{V}_Z^j(N+1) = \begin{cases} \mathbf{V}_Z^j(N+1), & \text{ЯКЩО } \mathbf{V}_Z^j(N+1) < \mathbf{V}_{\max}^j; \\ \mathbf{V}_{\max}^j, & \text{ЯКЩО } \mathbf{V}_Z^j(N+1) \geq \mathbf{V}_{\max}^j, \end{cases} \quad (2.23)$$

де  $\mathbf{V}_Z^j(N+1)$  – вектор швидкостей  $Z$ -ї частинки, який розраховується на основі виразу (2.20);  $\mathbf{V}_{\max}^j$  – вектор максимально допустимих значень швидкостей частинок для параметрів, що оптимізуються,  $j = \{1, 2, \dots, m\}$ .

На етапі ініціалізації PSO-методів встановлюються наступні параметри: загальна кількість частинок рою  $Z_{\max}$ ; загальна кількість параметрів  $m$  задачі, що одночасно оптимізуються; вектори обмежень для параметрів, що оптимізуються  $\mathbf{X}_{\min}^j$  та  $\mathbf{X}_{\max}^j$ ,  $j = \{1, 2, \dots, m\}$ ; вектори прискорень частинок  $\mathbf{C}_1^j$ ,  $\mathbf{C}_2^j$ ; початкові позиції частинок  $\mathbf{X}_Z^j(0)$ ,  $j = \{1, 2, \dots, m\}$ ; початкові швидкості частинок  $\mathbf{V}_Z^j(0)$ ; вектор максимально допустимих значень швидкостей частинок  $\mathbf{V}_{\max}^j$ ; тип, параметри та оптимальне значення цільової функції  $J$ ; максимальна кількість ітерацій  $N_{\max}$ ; критерій закінчення оптимізації та ін. [48]. У свою чергу, початкові позиції частинок  $\mathbf{X}_Z^j(0)$  можуть задаватися рівномірно на інтервалах пошуку  $[\mathbf{X}_{\min}^j, \mathbf{X}_{\max}^j]$ ,  $j = \{1, 2, \dots, m\}$ , або випадковим чином на основі залежності (2.24) [48]

$$\begin{aligned} \mathbf{X}_Z^j(0) &= \mathbf{X}_{\min}^j + \mathbf{R}_{0Z}^j (\mathbf{X}_{\max}^j - \mathbf{X}_{\min}^j), \\ \mathbf{R}_{0Z}^j &= \text{rand}[0,1], Z = \{1, \dots, Z_{\max}\}, j = \{1, \dots, m\} \end{aligned} \quad (2.24)$$

де  $\mathbf{R}_{0Z}^j$  – вектор випадкових чисел, що використовуються для визначення початкових позицій частинок рою.

Початкові значення швидкості частинок  $\mathbf{V}_Z^j(0)$  у більшості випадків доцільно встановити рівними 0 [71].

*Узагальнений метод рою частинок.* Узагальнений PSO-метод складається з наступних послідовних кроків.

*Крок 1.* Ініціалізація методу. На даному етапі здійснюється ініціалізація параметрів методу PSO. На етапі ініціалізації обираються: вектор параметрів розв’язуваної задачі, що одночасно оптимізуються  $\mathbf{X}$ , а також його обмеження  $\mathbf{X}_{\min}^j$  та  $\mathbf{X}_{\max}^j$ ,  $j = \{1, 2, \dots, m\}$ ; тип, параметри та

оптимальне значення цільової функції  $J$ ; максимальна кількість ітерацій при реалізації методу  $N_{\max}$ ; критерій закінчення оптимізації. В якості критерія закінчення оптимізації доцільно обрати досягнення оптимального значення цільової функції  $J \leq J_{\text{opt}}$  або виконання максимальної кількості ітерацій  $N_{\max}$ .

Крім того, на даному етапі створюється рій з  $Z_{\max}$  частинок, визначаються початкові позиції частинок  $\mathbf{X}_Z^j(0)$ , їх початкові швидкості  $\mathbf{V}_Z^j(0)$ , вектор максимально допустимих значень швидкостей  $\mathbf{V}_{\max}^j$ , а також вектори прискорень  $\mathbf{C}_1^j$  та  $\mathbf{C}_2^j$ ,  $j = \{1, 2, \dots, m\}$ . У свою чергу, початкові позиції частинок  $\mathbf{X}_Z^j(0)$  у більшості випадків доцільно визначати випадковим чином на основі виразу (2.24), а значення швидкостей  $\mathbf{V}_Z^j(0)$  – встановити рівними 0. За наявності однієї або декількох попередніх гіпотез про значення вектору параметрів  $\mathbf{X}$ , у створений рій можуть додаватися частинки з початковими позиціями, що відповідають даним значенням.

*Крок 2.* Обчислення значень цільової функції. На даному етапі для кожної  $Z$ -ї частинки рою з позицією  $\mathbf{X}_Z^j(N)$ ,  $Z = \{1, 2, \dots, Z_{\max}\}$  виконується розрахунок значення цільової функції  $J_Z(N)$  розв'язуваної задачі, обраної на кроці 1.

*Крок 3.* Перевірка завершення оптимізаційного процесу. На даному етапі за розрахованими на попередньому кроці значеннями цільової функції  $J_Z(N)$  для всіх частинок рою здійснюється перевірка на закінчення оптимізаційного процесу, що визначається досягненням оптимального значення цільової функції задачі ( $J_Z(N) \leq J_{\text{opt}}$ ) або виконанням максимальної кількості ітерацій  $N_{\max}$ . Якщо ця перевірка дала позитивний результат, тоді здійснюється перехід на крок 8. У протилежному випадку виконується перехід на крок 4.

*Крок 4.* Визначення найкращих позицій усіх частинок рою. На даному етапі на основі значень цільової функції  $J_Z(N)$  здійснюється

визначення найкращої позиції  $\mathbf{L}_Z^j$  для кожної  $Z$ -ї частинки рою,  $Z = \{1, 2, \dots, Z_{\max}\}$  відповідно до залежності (2.21).

*Крок 5.* Визначення глобальної найкращої позиції всього рою. На даному етапі з найкращих позицій всіх частинок рою  $\mathbf{L}_Z^j, Z = \{1, 2, \dots, Z_{\max}\}$ , визначених на попередньому кроці, за найменшим значенням цільової функції  $J_Z(N)$  обирається глобальна найкраща позиція всього рою  $\mathbf{G}^j$  на основі виразу (2.22).

*Крок 6.* Оновлення вектору швидкості частинок рою. На цьому етапі здійснюється розрахунок значень вектору швидкості  $\mathbf{V}_Z^j$  на основі виразу (2.20) для всіх частинок рою. Також, для даних частинок за допомогою залежності (2.23) розраховується вектор реальних швидкостей  $\mathbf{V}_Z'^j$  для обмеження надмірно швидкого зростання швидкостей руху агентів.

*Крок 7.* Оновлення позицій частинок рою. На даному етапі здійснюється розрахунок значень положень частинок  $\mathbf{X}_Z^j$  в  $m$ -мірному просторі пошуку на основі залежності (2.19) для всіх частинок рою. Далі здійснюється перехід на крок 2.

*Крок 8.* Завершення процесу оптимізації.

Блок-схема узагальненого мультиагентного методу рою частинок наведена на рис. 2.7 [48].

*Переваги, недоліки та рекомендації до застосування PSO-методів.* У цілому алгоритми оптимізації на основі рою частинок мають такі ж самі переваги та недоліки, як і інші біоінспіровані метаевристичні алгоритми, якщо їх порівнювати з традиційними пошуковими методами. Однак, при порівнянні з еволюційними алгоритмами, зокрема, генетичними, штучних імунних систем та біогеографічними, PSO-методи мають кращу збіжність при вирішенні задач неперервної оптимізації та, у цілому, більш просту реалізацію, кращу масштабованість та меншу кількість регульованих



параметрів [48, 71]. З іншого боку, PSO-методи мають значні обмеження при вирішенні задач дискретної оптимізації.

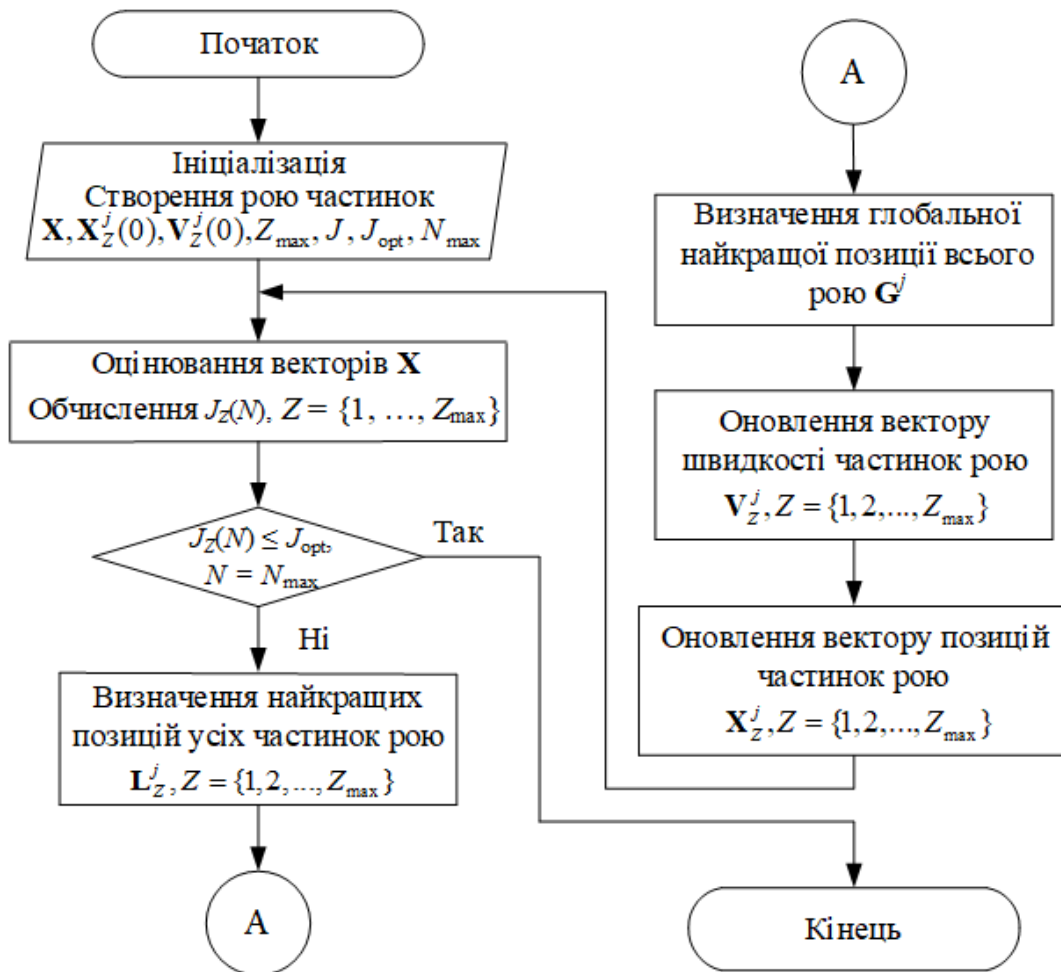


Рис. 2.7. Блок-схема узагальненого мультиагентного методу рою частинок

Для ефективного застосування PSO-алгоритмів доцільно здійснювати попередній аналіз особливостей обраних задач оптимізації, ретельно налаштовувати регульовані параметри (розмір рою, значення векторів прискорень, початкові положення частинок) та використовувати принцип обмеження швидкостей руху частинок у просторі пошуку [48, 70, 71]. Також, для пришвидшення збіжності методу PSO доцільним є здійснення його гібридизації з іншими методами, зокрема локального пошуку [126].

## 2.6 Методи сірих вовків

*Базові аспекти методу сірих вовків.* Методи оптимізації сірих вовків (Grey Wolf Optimization – GWO) відносяться до групи сучасних біоінспірованих мультиагентних методів, які моделюють групову поведінку вовків у зграї під час їх руху у просторі в процесі полювання [75, 76, 127]. Концепція оптимізації GWO базується на імітації соціальної структури вовчої зграї у природі та використанні її для розв'язання оптимізаційних завдань. Ці методи запозичують ключові характеристики вовчої ієрархії, індивідуальну роль кожної особини у зграї та співпрацю між ними.

Моделі переміщення та взаємодії вовків у зграї при пошуку здобичі в GWO-методі представлені агентами єдиної мультиагентної системи, які переміщуються у багатовимірному просторі пошуку, обмінюючись між собою інформацією, для знаходження оптимуму розв'язуваної задачі. Поточне положення кожного вовка зграї у багатовимірному просторі пошуку на кожній ітерації є потенційним розв'язком оптимізаційної задачі та, у свою чергу, є певним значенням вектору-рішення  $\mathbf{X}$ .

Крім того, найбільш важливою особливістю GWO-алгоритму є те, що він моделює сувору соціальну 4-рівневу ієрархію сірих вовків у зграї [75]. Зграя складається з  $Z_{\max}$  вовків, і кожен  $Z$ -й вовк знаходиться на одному з чотирьох рівнів ієрархії. У свою чергу, перші три рівні ієрархії містять тільки по одному вовку, а всі інші вовки зграї знаходяться на 4-му рівні. Вожаком зграї є альфа-вовк ( $\alpha$ ), який знаходиться на 1-му рівні ієрархії. Бета- і дельта-вовки ( $\beta$ ,  $\delta$ ) знаходяться, відповідно, на 2-му та 3-му рівнях в ієрархії зграї. Всі інші вовки зграї, які знаходяться на найнижчому 4-му рівні ієрархії, є вовками типу омега ( $\omega$ ) [75].

Для математичного моделювання вищенаведеної соціальної ієрархії вовків у зграї в GWO-методах, три агенти з найкращими значеннями цільової функції  $J_{1\text{best}}(N)$ ,  $J_{2\text{best}}(N)$  та  $J_{3\text{best}}(N)$  на поточній  $N$ -й ітерації є

альфа-, бета- і дельта-вовками з позиціями  $\mathbf{X}_\alpha(N)$ ,  $\mathbf{X}_\beta(N)$  та  $\mathbf{X}_\delta(N)$ . Усі інші агенти в популяції – омега-вовки з позиціями  $\mathbf{X}_Z(N)$ ,  $Z = \{4, 5, \dots, Z_{\max}\}$ . У свою чергу, для визначення альфа-, бета- та дельта-агентів на кожній ітерації процесу оптимізації усі  $Z_{\max}$  агентів у популяції попередньо ранжуються відповідно до поточного значення їх цільової функції  $J_Z(N)$  [75]. Отже, агент з першим найкращим значенням цільової функції вважається альфа-агентом. Тоді, як агенти з другим і третім найкращими значеннями цільової функції є бета- і дельта-вовками, відповідно.

У процесі оптимізації базовий GWO-метод моделює стратегію групового полювання, яка ґрунтується на трьох основних механізмах поведінки вовків у зграї: оточення здобичі, полювання та атакування здобичі [75]. Механізм оточення здобичі можна представити наступними рівняннями

$$\mathbf{X}_Z(N+1) = \mathbf{X}_{Pr}(N) - \mathbf{A}_Z(N)\mathbf{D}_Z(N); \quad (2.25)$$

$$\mathbf{D}_Z(N) = |\mathbf{C}_Z(N)\mathbf{X}_{Pr}(N) - \mathbf{X}_Z(N)|, Z = \{1, 2, \dots, Z_{\max}\}, \quad (2.26)$$

де  $\mathbf{X}_{Pr}$  – вектор положення здобичі, що відповідає оптимальному значенню параметрів вектора-рішення розв’язуваної задачі;  $\mathbf{D}_Z$  – вектор відстаней від здобичі до вовків;  $\mathbf{A}_Z$ ,  $\mathbf{C}_Z$  – вектори коефіцієнтів.

У свою чергу, вектори  $\mathbf{A}_Z$  та  $\mathbf{C}_Z$  розраховуються наступним чином:

$$\mathbf{A}_Z(N) = 2a_N(N)\mathbf{R}_{1Z}(N) - a_N(N); \quad (2.27)$$

$$\mathbf{C}_Z(N) = 2\mathbf{R}_{2Z}(N), Z = \{1, 2, \dots, Z_{\max}\}, \quad (2.28)$$

де  $a_N$  – коефіцієнт, який лінійно зменшується у ході ітерацій від 2 до 0;  $\mathbf{R}_{1Z}$  та  $\mathbf{R}_{2Z}$  – вектори випадкових чисел у діапазоні від 0 до 1,  $\mathbf{R}_{1Z}$ ,  $\mathbf{R}_{2Z} = \text{rand}[0, 1]$ .

Наведені вище рівняння описують лише принцип поведінки вовків у процесі оточення здобичі. У реальному процесі оптимізації позиція здобичі, яка відповідає оптимальному значенню параметрів вектора-рішення задачі, невідома. Тому, в GWO-алгоритмі всі омега-вовки з

позиціями  $\mathbf{X}_Z(N)$ ,  $Z = \{4, 5, \dots, Z_{\max}\}$  у процесі пошуку та оточення здобичі керуються не положенням, безпосередньо, самої здобичі  $\mathbf{X}_{Pr}$ , а позиціями трьох найкращих вовків у зграї: альфа  $\mathbf{X}_\alpha$ , бета  $\mathbf{X}_\beta$  та дельта  $\mathbf{X}_\delta$ . Це є другим механізмом поведінки вовків у зграї – механізмом полювання. Даний механізм моделюється в GWO-методі за допомогою таких виразів [75]

$$\mathbf{X}_Z(N+1) = \frac{\mathbf{X}_{1Z}(N) + \mathbf{X}_{2Z}(N) + \mathbf{X}_{3Z}(N)}{3}; \quad (2.29)$$

$$\mathbf{X}_{1Z}(N) = \mathbf{X}_\alpha(N) - \mathbf{A}_{1Z}(N)\mathbf{D}_{\alpha Z}(N); \quad (2.30)$$

$$\mathbf{X}_{2Z}(N) = \mathbf{X}_\beta(N) - \mathbf{A}_{2Z}(N)\mathbf{D}_{\beta Z}(N); \quad (2.31)$$

$$\mathbf{X}_{3Z}(N) = \mathbf{X}_\delta(N) - \mathbf{A}_{3Z}(N)\mathbf{D}_{\delta Z}(N); \quad (2.32)$$

$$\mathbf{D}_{\alpha Z}(N) = |\mathbf{C}_{1Z}(N)\mathbf{X}_\alpha(N) - \mathbf{X}_Z(N)|; \quad (2.33)$$

$$\mathbf{D}_{\beta Z}(N) = |\mathbf{C}_{2Z}(N)\mathbf{X}_\beta(N) - \mathbf{X}_Z(N)|; \quad (2.34)$$

$$\mathbf{D}_{\delta Z}(N) = |\mathbf{C}_{3Z}(N)\mathbf{X}_\delta(N) - \mathbf{X}_Z(N)|, Z = \{1, 2, \dots, Z_{\max}\}, \quad (2.35)$$

де  $\mathbf{X}_{1Z}$ ,  $\mathbf{X}_{2Z}$  та  $\mathbf{X}_{3Z}$  – вектори напрямку для розрахунку наступної позиції  $Z$ -го омега-вовка;  $\mathbf{D}_{\alpha Z}$ ,  $\mathbf{D}_{\beta Z}$  та  $\mathbf{D}_{\delta Z}$  – вектори відстаней від альфа-, бета- та дельта-вовків до  $Z$ -го омега-вовка;  $\mathbf{A}_{1Z}$ ,  $\mathbf{A}_{2Z}$  та  $\mathbf{A}_{3Z}$  – вектори коефіцієнтів, які розраховуються аналогічним чином відповідно до рівняння (2.27);  $\mathbf{C}_{1Z}$ ,  $\mathbf{C}_{2Z}$  та  $\mathbf{C}_{3Z}$  – вектори коефіцієнтів, які розраховуються аналогічним чином відповідно до рівняння (2.28).

Процес полювання закінчується атакуванням здобичі, коли вона перестає рухатися [76]. Математично це відображається тим, що коефіцієнт  $a_N$  лінійно зменшується протягом ітерацій від значень 2 до 0. Його поточне значення може бути розраховано за рівнянням (2.36)

$$a_N(N) = 2 \left( 1 - \frac{N}{N_{\max}} \right), \quad (2.36)$$

де  $N_{\max}$  – максимальна кількість ітерацій методу.

*Узагальнений метод сірих вовків.* Узагальнений GWO-метод складається з наступних послідовних кроків.

*Крок 1.* Ініціалізація GWO-методу. На даному етапі обирається вектор  $\mathbf{X}$  параметрів задачі, що оптимізуються, та задаються його основні обмеження  $\mathbf{X}_{\min}$  та  $\mathbf{X}_{\max}$ . Крім того, на цьому етапі вибирається тип, параметри та оптимальне значення цільової функції  $J$ . Також, на цьому кроці створюється зграя вовків із встановленням в ній загальної кількості особин  $Z_{\max}$  та їх початкових позицій  $\mathbf{X}_Z(0)$ ,  $Z = 1, 2, \dots, Z_{\max}$ . У свою чергу, у багатьох випадках доцільним є випадкове розподілення початкових позицій особин у просторі пошуку у діапазоні  $[\mathbf{X}_{\min}, \mathbf{X}_{\max}]$  відповідно до залежності (2.24) [48].

При наявності попередньої гіпотези про значення змінних вектора-рішення  $\mathbf{X}$  один із вовків зграї матиме початкове положення, що відповідає цим значенням. Якщо таких гіпотез декілька, то у зграї буде декілька вовків з початковими позиціями, що відповідають даним гіпотезам. Також, обирається критерій закінчення оптимізації та максимальна кількість ітерацій  $N_{\max}$ .

*Крок 2.* Оцінка сформованих векторів-рішень задачі оптимізації. На даному етапі для кожного вектора-рішення  $\mathbf{X}_Z(N)$ , яке відповідає кожному  $Z$ -му вовку зграї ( $Z = 1, 2, \dots, Z_{\max}$ ), розраховується значення цільової функції  $J_Z(N)$ .

*Крок 3.* Перевірка на завершення процесу оптимізації. На даному кроці у відповідності до значень цільової функції  $J_Z(N)$ , розрахованих для всіх  $Z_{\max}$  вовків зграї, проводиться перевірка на завершення процесу оптимізації, що визначається досягненням оптимального значення цільової функції задачі ( $J_Z(N) \leq J_{\text{opt}}$ ) або виконанням максимальної кількості ітерацій  $N_{\max}$ . Процес оптимізації також можна вважати завершеним, якщо значення цільової функції  $J$  не зменшувалося протягом певної (попередньо заданої) кількості ітерацій. Якщо ця перевірка дала позитивний результат,

тоді здійснюється перехід до кроку 6. В протилежному випадку виконується перехід до кроку 4.

*Крок 4.* Відбір альфа-, бета- та дельта- вовків (агентів) зграї (популяції). На даному етапі всі  $Z_{\max}$  агентів у популяції ранжуються відповідно до поточного значення їх цільової функції  $J_Z(N)$ , та вибираються 3 перші агенти з найкращими значеннями цільової функції  $J_{1\text{best}}(N)$ ,  $J_{2\text{best}}(N)$  і  $J_{3\text{best}}(N)$ , які вважаються альфа-, бета- та дельта-вовками з позиціями  $\mathbf{X}_\alpha(N)$ ,  $\mathbf{X}_\beta(N)$  та  $\mathbf{X}_\delta(N)$ . Усі інші агенти в популяції є омега-вовками з позиціями  $\mathbf{X}_Z(N)$ ,  $Z = \{4, 5, \dots, Z_{\max}\}$ .

*Крок 5.* Визначення нових значень вектора-рішення  $\mathbf{X}$  для омега-агентів на основі стратегії групового полювання. На цьому кроці обчислюються нові значення векторів  $\mathbf{X}(N+1)$  (нові позиції) для всіх омега-агентів популяції на основі рівнянь (2.29)-(2.36) згідно стратегії групового полювання. Далі здійснюється перехід до кроку 2.

*Крок 6.* Завершення процесу оптимізації.

Блок-схема узагальненого мультиагентного методу сірих вовків наведена на рис. 2.8 [75].

На сьогоднішній день розроблено досить багато різних модифікацій методу сірих вовків, серед яких: метод GWO-EPD, що застосовує додатковий оператор еволюційної динаміки популяції [128], метод wdGWO, в якому обчислюється зважене середнє значення із кращих рішень замість простого арифметичного середнього [129], гібридний метод HGWO, в якому додатково застосовуються оператори мутації та кросовера для особин зграї [130], покращений метод IGWO, що додатково використовує стратегію розмірного полювання на основі навчання (Dimension Learning-based Hunting) [131], гібридний покращений метод HIGWO, в якому здійснюється паралельний локальний пошук трьома кращими агентами на основі алгоритмів розширеного фільтра Калмана або градієнтного спуску [132] та ін.

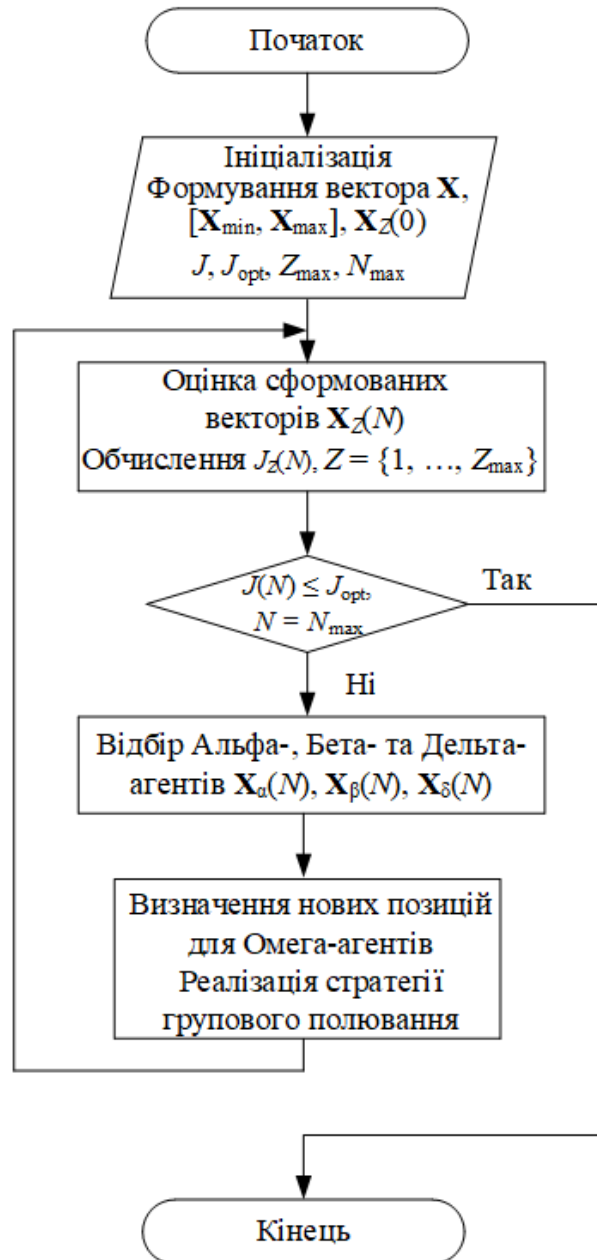


Рис. 2.8. Блок-схема узагальненого мультиагентного методу сірих вовків

*Переваги, недоліки та рекомендації до застосування GWO-методів.*

У цілому алгоритми сірих вовків мають такі ж самі переваги та недоліки, як і інші мультиагентні ройові алгоритми. При їх порівнянні з традиційними пошуковими методами, вони мають переваги у випадках відсутності інформації щодо характеру та властивостей цільових функцій, дозволяють уникати зациклення у локальних оптимумах та більш ретельно досліджувати великі негладкі простори пошуку. Також, методи GWO

мають суттєві переваги при розв'язанні задач оптимізації великої розмірності [75].

Якщо їх порівнювати з еволюційними алгоритмами, то дані методи, у цілому, мають кращу збіжність при вирішенні задач неперервної оптимізації та більш просту реалізацію. Суттєвою особливістю саме GWO-методу є мінімальна кількість його регульованих параметрів [75]. У свою чергу, даний метод, також, може бути легко адаптований до вирішення різнотипних задач та гібридизований з низкою інших методів як локального, так і глобального пошуку.

Для ефективного застосування GWO-методів доцільно попередньо здійснювати ретельний аналіз особливостей обраних задач оптимізації та раціонально налаштовувати регульовані параметри (розмір зграї, початкові положення агентів) [75, 76]. Також, введення додаткових пошукових процедур може значно підвищити збіжність та зменшити ризик зациклення методу у локальних мінімумах [131, 132].

Розглянуті у вищенаведених підрозділах біоінспіровані еволюційні та мультиагентні методи є потужним інструментом для вирішення складних оптимізаційних завдань. Однак, ефективність використання їх зазначених переваг та можливість реалізації усього закладеного потенціалу суттєво залежить від значної кількості різноманітних факторів. Тому, на сьогодні, не існує єдиного універсального методу оптимізації, який би однаково добре підходив для вирішення будь-яких задач. Зокрема, кожна окрема задача оптимізації вимагає індивідуального підбору конкретного біоінспірованого методу та його подальшої адаптації. Таким чином, для досягнення бажаних результатів користувач повинен ретельно враховувати всі особливості конкретної розв'язуваної задачі оптимізації (розмірність, характер обмежень, структура цільової функції, наявна обчислювальна потужність та ін.). Також, визначальним є правильне налаштування всіх регульованих параметрів обраного методу (розмір популяції, тип та параметри пошукових процедур).



## 2.7 Контрольні питання

1. Сформулюйте основні принципи біоінспірованих методів оптимізації.
2. Що являє собою глобальна оптимізація?
3. Які основні категорії біоінспірованих методів інтелектуальної оптимізації ви знаєте?
4. Які методи відносяться до еволюційних?
5. Перерахуйте основні мультиагентні методи глобальної оптимізації.
6. Які ви знаєте оптимізаційні методи, що моделюють фізичні процеси?
7. До якої категорії біоінспірованих методів відноситься імперіалістичний конкурентний алгоритм?
8. Сформулюйте основні принципи еволюційних обчислень.
9. Перерахуйте основні переваги еволюційних методів у порівнянні з класичними методами оптимізації.
10. Які принципи визначають базову структуру генетичних алгоритмів?
11. Що включає в себе поняття "популяція" в контексті генетичних алгоритмів?
12. Яким чином використовується оператор кросоверу в GA?
13. Як обираються батьківські особини для формування наступного покоління?
14. Які критерії відбору визначають ефективність генетичного алгоритму?
15. Як визначається та використовується функція придатності у GA?
16. Як впливає процес мутації на еволюцію популяції в генетичних алгоритмах?

17. Які підходи використовуються для ініціалізації початкової популяції у генетичних алгоритмах?
18. Наведіть блок-схему узагальненого генетичного алгоритму.
19. Які альтернативні варіації GA можуть використовуватися в залежності від характеру задачі?
20. Як контролюється розмір популяції для оптимізації у GA?
21. Як генетичні алгоритми адаптуються до змінних умов задачі в процесі оптимізації?
22. Перерахуйте основні оператори селекції в GA.
23. Чим відрізняється одноточкове схрещення від двоточкового?
24. Які способи кодування параметрів у хромосоми застосовуються в GA?
25. Яким чином фітнес-функція відображає придатність індивіда в популяції?
26. Перерахуйте основні принципи, що лежать в основі алгоритмів штучних імунних систем.
27. Яким чином AIS-алгоритми використовують концепцію імунної системи для вирішення оптимізаційних завдань?
28. Опишіть основні функції клітин пам'яті в алгоритмах штучних імунних систем при вирішенні задач оптимізації.
29. Що принципово відрізняє алгоритми штучних імунних систем від інших еволюційних алгоритмів?
30. Як в алгоритмах штучних імунних систем моделюються антигени та антитіла при вирішенні задач оптимізації?
31. Яким чином здійснюється взаємодія між пам'яттю імунної системи та пошуковою стратегією алгоритмів AIS?
32. Які оператори клонування застосовуються в алгоритмах штучних імунних систем?
33. Яким чином розв'язується проблема збереження різноманітності популяції під час оптимізації в алгоритмах AIS?

34. Як впливає розмір та складність простору пошуку на ефективність алгоритмів штучних імунних систем?
35. Перерахуйте основні переваги та обмеження алгоритмів AIS у порівнянні з іншими оптимізаційними методами.
36. Яким чином виконується процедура селекції клітин в популяції при реалізації алгоритмів AIS?
37. Які регульовані параметри алгоритмів штучних імунних систем необхідно налаштовувати для досягнення найбільшої продуктивності?
38. Наведіть блок-схему узагальненого AIS-алгоритму.
39. Яким чином відбувається мутація клонів клітин пам'яті в AIS-алгоритмах?
40. Які переваги та недоліки мають AIS-алгоритми у порівнянні з GA?
41. В чому полягає основна ідея алгоритмів оптимізації на основі біогеографії?
42. Як впливає кількість островів екосистеми в ВВО-алгоритмах на пошук оптимальних рішень?
43. Механізм міграції в ВВО-алгоритмах та його роль у пошуку оптимальних рішень.
44. Складові процедури мутації в методах ВВО та їх вплив на різноманітність популяції.
45. Яким чином зберігається баланс між глобальним та локальним пошуком в алгоритмах ВВО?
46. Стратегії вибору індивідів для міграції в ВВО-методах.
47. Яким чином визначається інтенсивність міграції в ВВО-алгоритмах?
48. Поясніть призначення процедур імміграції та еміграції видів у ВВО-алгоритмах.

49. Оператор мутацій для зміни характеристик індивідів в ВВО-методах.
50. У чому полягає принципова відмінність алгоритмів ВВО від GA та інших еволюційних методів?
51. Особливості застосування алгоритмів ВВО для вирішення оптимізаційних задач в різних галузях.
52. Яким чином впливають залежності ймовірності імміграції та еміграції від кількості видів на збіжність алгоритмів ВВО?
53. Як реалізований взаємозв'язок кількості видів  $N_S$  на острові від HSI у ВВО-методах?
54. За рахунок чого ВВО-алгоритми мають більший потенціал для гнучкого налаштування у порівнянні з GA?
55. Наведіть блок-схему узагальненого алгоритму ВВО.
56. Назвіть основні принципи функціонування мультиагентних систем.
57. Основні аспекти мультиагентної інтелектуальної оптимізації.
58. Опишіть основні механізми самоорганізації в мультиагентних системах.
59. Яким чином реалізуються додатний та від'ємний зворотні зв'язки в методах мультиагентної інтелектуальної оптимізації?
60. За рахунок чого досягається наявність колективного інтелекту в мультиагентних системах?
61. У чому полягає основна ідея методу мурашиних колоній?
62. Як визначаються феромони в методах мурашиних колоній і яку роль вони виконують?
63. Яким чином мурахи взаємодіють із середовищем під час пошуку оптимального рішення?
64. Як відбувається вибір шляху мурахами в АСО-методах, і як це впливає на процес глобального пошуку?

65. Яким чином феромони оновлюються на різних етапах алгоритму мурашиних колоній?
66. Назвіть основні етапи алгоритму АСО.
67. Як визначається ймовірність вибору наступного вузла на шляху агентом в АСО-алгоритмі?
68. Як впливає параметр інтенсивності відкладання та випаровування феромону на швидкість збіжності алгоритмів АСО?
69. Яким чином враховуються обмеження задачі оптимізації в методі мурашиних колоній?
70. Як впливають регульовані параметри  $\alpha$  та  $\beta$  на швидкість збіжності алгоритмів АСО?
71. Наведіть блок-схему базового методу АСО.
72. Яким чином метод АСО може бути застосований для розв'язання комбінаторних оптимізаційних задач?
73. Особливості вибору оптимальних параметрів при вирішенні конкретної задачі за допомогою методу мурашиних колоній.
74. Яким чином впливає зворотна відстань між вузлами на ймовірність обрання агентом певного наступного вузла шляху?
75. У чому полягають основні принципи методу рою частинок?
76. Яким чином визначається рух окремих частинок у методі PSO?
77. Як впливає швидкість частинок на їх рух і пошук оптимального рішення?
78. Наведіть основні рівняння обчислення локальної та глобальної найкращих позицій частинок рою.
79. Яким чином здійснюється обмеження швидкостей руху частинок у методі PSO?
80. За рахунок чого метод PSO здатен виходити з зациклення у локальних оптимумах?
81. Які різновиди PSO-методів ви знаєте?

82. Яким чином враховуються обмеження та умови задачі у методі рою частинок?
83. Як впливає розмір рою на швидкість збіжності алгоритму PSO?
84. Наведіть блок-схему базового методу рою частинок.
85. Особливості налаштування регульованих параметрів методу PSO при застосуванні для вирішення конкретної задачі оптимізації.
86. Наведіть основні переваги та недоліки PSO-методів у порівнянні з еволюційними методами штучного інтелекту.
87. Яким чином задаються вектори прискорень частинок у методах PSO?
88. Які критерії закінчення процесу оптимізації застосовуються у методах PSO?
89. Яким чином задаються початкові положення та швидкості частинок на етапі ініціалізації у методах PSO?
90. Базові принципи методу сірих вовків.
91. Що виступає потенційним розв'язком оптимізаційної задачі при застосуванні методу GWO?
92. Яким чином використовується ідея ієрархії вовків для розв'язання оптимізаційних задач за допомогою GWO-методів?
93. Яким чином обчислюються та оновлюються нові позиції вовків у методі GWO?
94. Назвіть три основні механізми поведінки вовків у зграї, що застосовуються у базовому GWO-методі.
95. Яким чином реалізований механізм оточення здобичі у GWO-методі?
96. Як впливає кількість вовків у зграї на ефективність алгоритму GWO?
97. Яким чином реалізований механізм атакування здобичі у GWO-методі?

98. Яким чином здійснюється відбір альфа-, бета- та дельта-агентів зграї у GWO-методі?
99. Які регульовані параметри має метод GWO?
100. Наведіть блок-схему базового методу GWO.
101. Які переваги та недоліки мають GWO-методи?
102. Які модифікації GWO-методу ви знаєте?
103. Яким чином враховуються обмеження та умови задачі оптимізації у методі GWO?

## РОЗДІЛ 3

### ЗАСТОСУВАННЯ БІОІНСПІРОВАНИХ МЕТОДІВ ДЛЯ СИНТЕЗУ ТА ОПТИМІЗАЦІЇ НЕЙРОМЕРЕЖЕВИХ ТА НЕЧІТКИХ СИСТЕМ

Значною особливістю нейромережєвих та нечітко-логічних інтелектуальних систем, розглянутих у розділі 1, є досить велика кількість регульованих параметрів та структурних елементів, які суттєво впливають на їх ефективність, і повинні бути визначені та налаштовані у процесі розробки [9, 32-34]. З одного боку, це дозволяє реалізовувати складні, гнучкі та найбільш ефективні алгоритми обробки інформації, а також багатоетапні стратегії інтелектуального керування та прийняття рішень. Однак, з іншого боку, це суттєво ускладнює процедури проектування та налаштування наведених систем [29, 133].

Водночас, передові дослідження в галузі штучного інтелекту показують доцільність та перспективність застосування розглянутих у 2-му розділі біоінспірованих інтелектуальних методів глобальної оптимізації для створення нейронних, нечітких та нейро-нечітких систем різного типу [45, 134]. Зокрема, при вирішенні задач синтезу наведених інтелектуальних систем, дані еволюційні та мультиагентні методи мають низку переваг у порівнянні з класичними методами локального пошуку, а саме: (а) не накладають додаткові обмеження на цільові функції; (б) дозволяють детально досліджувати великі, негладкі і мультимодальні простори пошуку, виключаючи можливість зациклення у локальних екстремумах; (в) легко адаптуються для вирішення різнотипних задач синтезу інтелектуальних систем; (г) дозволяють ефективно оптимізувати складні інтелектуальні системи великої розмірності [46, 48].

Так, в опублікованих за останні роки наукових роботах є ряд прикладів успішного застосування біоінспірованих методів та алгоритмів для вирішення задач синтезу та структурно-параметричної оптимізації



інтелектуальних систем різних типів і конфігурацій [105, 135]. Наприклад, підхід до створення нейронних мереж на основі різнотипних еволюційних методів отримав назву «Нейроеволюція» (Neuroevolution) [136, 137]. Також, нечіткі системи, що розробляються за допомогою генетичних алгоритмів, утворюють окремий клас інтелектуальних систем: генетичні нечіткі системи (Genetic Fuzzy Systems) [138, 139].

У даному розділі розглянуто особливості застосування біоінспірованих еволюційних та мультиагентних методів для вирішення задач синтезу та оптимізації нейромережевих та нечітко-логічних інтелектуальних систем із наведенням конкретних прикладів. Зокрема, у підрозділі 3.1 представлено нейроеволюційний метод керування багатозв'язними складними технічними об'єктами. Даний метод дозволяє створювати багатозв'язні нейромережеві системи автоматичного керування з виконанням процедур структурно-параметричного синтезу на базі генетичного алгоритму. У свою чергу, у підрозділі 3.2 наведено мультиагентний метод генерації баз правил нечітких систем на основі мурашиних алгоритмів. Також, у даних підрозділах представлені приклади застосування розглянутих методів для створення реальних інтелектуальних систем.

### **3.1 Нейроеволюційний метод керування багатозв'язними складними технічними об'єктами**

*Особливості керування багатозв'язними складними технічними об'єктами.* При проектуванні різнотипних САК досить часто постає задача автоматизації складних технічних об'єктів (СТО), що характеризуються великою кількістю керованих змінних [140]. Для вирішення цієї задачі повинна бути створена багатоканальна система керування. Складність вирішення цієї задачі значною мірою залежить від певних особливостей багатокординатного об'єкта [140, 141]. Зокрема, при дослідженні

багатокоординатного об'єкта можливі два основні випадки: (а) вихідні змінні об'єкта не пов'язані між собою, і зміна однієї змінної не впливає на інші; (б) вихідні змінні взаємопов'язані, і зміна однієї з них призводить до зміни поточних значень інших [141].

У першому випадку кожна керована змінна об'єкта залежить лише від одного керуючого впливу і не залежить (або несуттєво залежить) від інших керуючих впливів. Для таких систем вторинними зв'язками можна знехтувати і розглядати САК як таку, що складається з окремих підсистем. Для управління кожною вихідною змінною об'єкта використовується власний керуючий пристрій (КП) певного типу. При цьому, система керування кожною вихідною величиною є повністю автономною і при дослідженні може розглядатися як самостійна САК. У даному випадку така система керування являє собою багатоканальну САК з незалежними каналами управління.

У другому випадку змінні керування СТО суттєво залежать від інших керуючих впливів. Тому, взаємозв'язками між керованими змінними нехтувати не можна, і об'єкт вважається багатозв'язним.

На рис. 3.1 наведена функціональна структура САК багатозв'язним СТО, де прийнято наступні позначення:  $Y_{Si}$ ,  $i = \{1, 2, \dots, m\}$  – задані значення керованих змінних багатозв'язного об'єкта;  $Y_{Ri}$ ,  $i = \{1, 2, \dots, m\}$  – реальні значення керованих змінних багатозв'язного об'єкта;  $\varepsilon_i$ ,  $i = \{1, 2, \dots, m\}$  – помилки керування змінними багатозв'язного СТО;  $КП_i$ ,  $i = \{1, 2, \dots, m\}$  – окремі керуючі пристрої змінних СТО;  $u_{Ci}$ ,  $i = \{1, 2, \dots, m\}$  – вихідні сигнали керуючих пристроїв;  $m$  – загальна кількість керованих змінних багатозв'язного об'єкта;  $F_D$  – вектор збурювальних впливів, що діють на об'єкт. Датчики та інші допоміжні засоби керування на рис. 3.1 не показано.

Аналіз багатозв'язних систем керування є досить складним через те, що об'єкт керування у цьому випадку описується великою кількістю передаточних функцій, які враховують усі взаємозв'язки між  $m$

керуваними змінними, керуючими впливами  $u_{Ci}$ ,  $i = \{1, 2, \dots, m\}$  та збуреннями  $F_D$  [141].

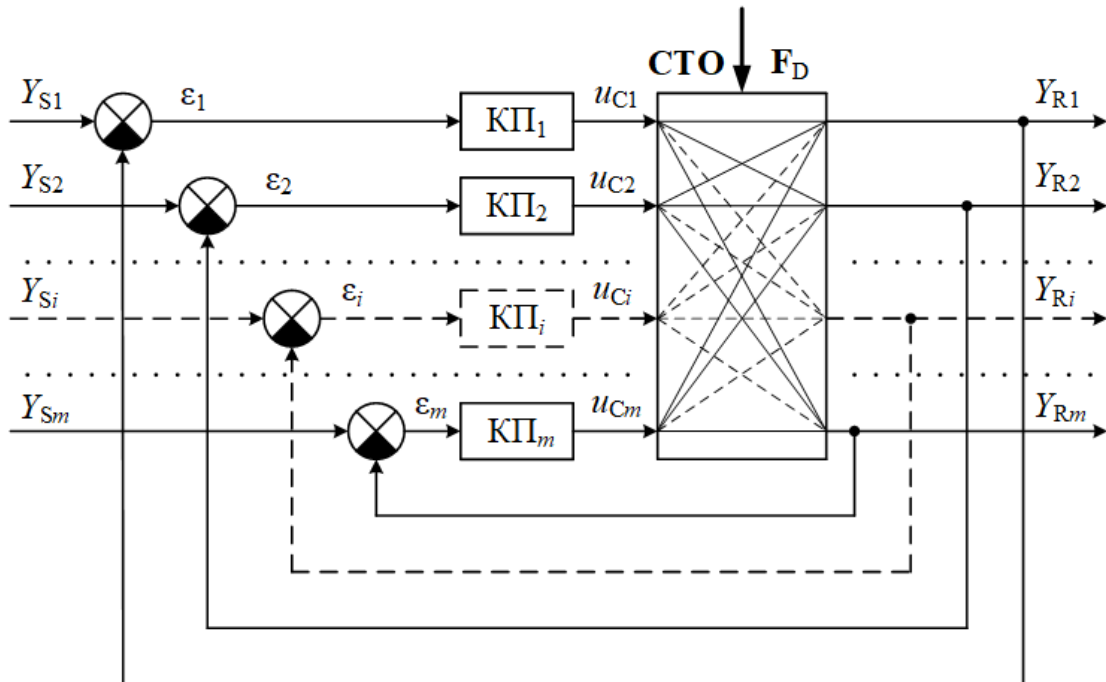


Рис. 3.1. Функціональна структура САК багатозв'язним об'єктом

На практиці багатозв'язним системам керування у більшості випадків прагнуть надати властивість автономності [141]. Автономне керування – це таке керування, при якому зміна будь-якої  $i$ -ї регульованої змінної не призводить до зміни іншої. Системи керування можуть бути автономними щодо вхідних чи збурювальних впливів. Автономність САК може бути досягнута за рахунок реалізації спеціалізованих коригувальних зворотних зв'язків [140]. У даному випадку, керуючі пристрої системи  $КП_i$ ,  $i = \{1, 2, \dots, m\}$  повинні бути розроблені з додатковими зв'язками між каналами керування для компенсації їх взаємного впливу. При проектуванні даних регуляторів необхідно враховувати характер взаємовпливу керуваних змінних, а це можливо лише за наявності детермінованого опису цього впливу [141].

Для достатньо великої кількості складних технічних об'єктів математичні моделі взаємовпливу їх керуваних змінних занадто складні або взагалі відсутні. Тому, керуючі пристрої їх САК розробляються окремо

для кожної змінної (рис. 3.1), переважно, на основі традиційних методів, а взаємний вплив контрольованих координат розглядається як збурення. У багатьох випадках це не дозволяє отримувати високі показники якості [140].

Для підвищення показників якості та загальної ефективності систем керування багатозв'язними СТО доцільно використовувати метод керування на основі нейроеволюції, розроблений у роботі [142]. Наведений метод передбачає побудову багатозв'язної системи керування СТО на основі єдиного комплексного нейроконтролера (НК) з багатьма входами та виходами. Основна ідея полягає у тому, що складна багатошарова нейронна мережа контролера при належному навчанні на основі генетичного алгоритму оптимально враховує взаємовплив усіх змінних багатокординатного об'єкта і дозволяє реалізувати взаємозв'язане автоматичне керування з високими показниками якості.

*Основні кроки нейроеволюційного методу керування багатозв'язними складними технічними об'єктами.* Запропонований у [142] метод складається з наступних етапів.

*Крок 1.* Побудова структури багатозв'язної САК на основі єдиного комплексного нейроконтролера. На цьому етапі будується структура системи керування, узагальнений варіант якої наведено на рис. 3.2. Дана САК має комплексний нейроконтролер, кількість виходів якого дорівнює числу керованих змінних  $m$ . При цьому кількість входів НК  $l$  обирається таким чином, щоб  $l \geq m$ , оскільки входами контролера можуть бути не тільки сигнали помилок  $\varepsilon_i$ ,  $i = \{1, 2, \dots, m\}$ , але і їх похідні різних порядків та інтеграли  $(\dot{\varepsilon}_i, \ddot{\varepsilon}_i, \dots, \int \varepsilon_i dt)$ . У свою чергу, комплексний НК крім багатошарової нейронної мережі містить у собі ще блоки попереднього обчислення похідних різних порядків та інтегралів від вхідних сигналів.

Також, до складу наведеної САК входять  $m$  еталонних моделей (ЕМ) процесів керування змінними СТО та  $m$  пристроїв порівняння, які

обчислюють розбіжності  $E_i$  між вихідними сигналами ЕМ  $Y_{RMi}$  та реальними виходами об'єкта  $Y_{Ri}$ ,  $i = \{1, 2, \dots, m\}$ . У свою чергу, розбіжності  $E_i$ ,  $i = \{1, 2, \dots, m\}$  використовуються для розрахунку значення комплексної цільової функції  $J_\Sigma$  системи керування за допомогою блоку розрахунку цільової функції (БРЦФ).

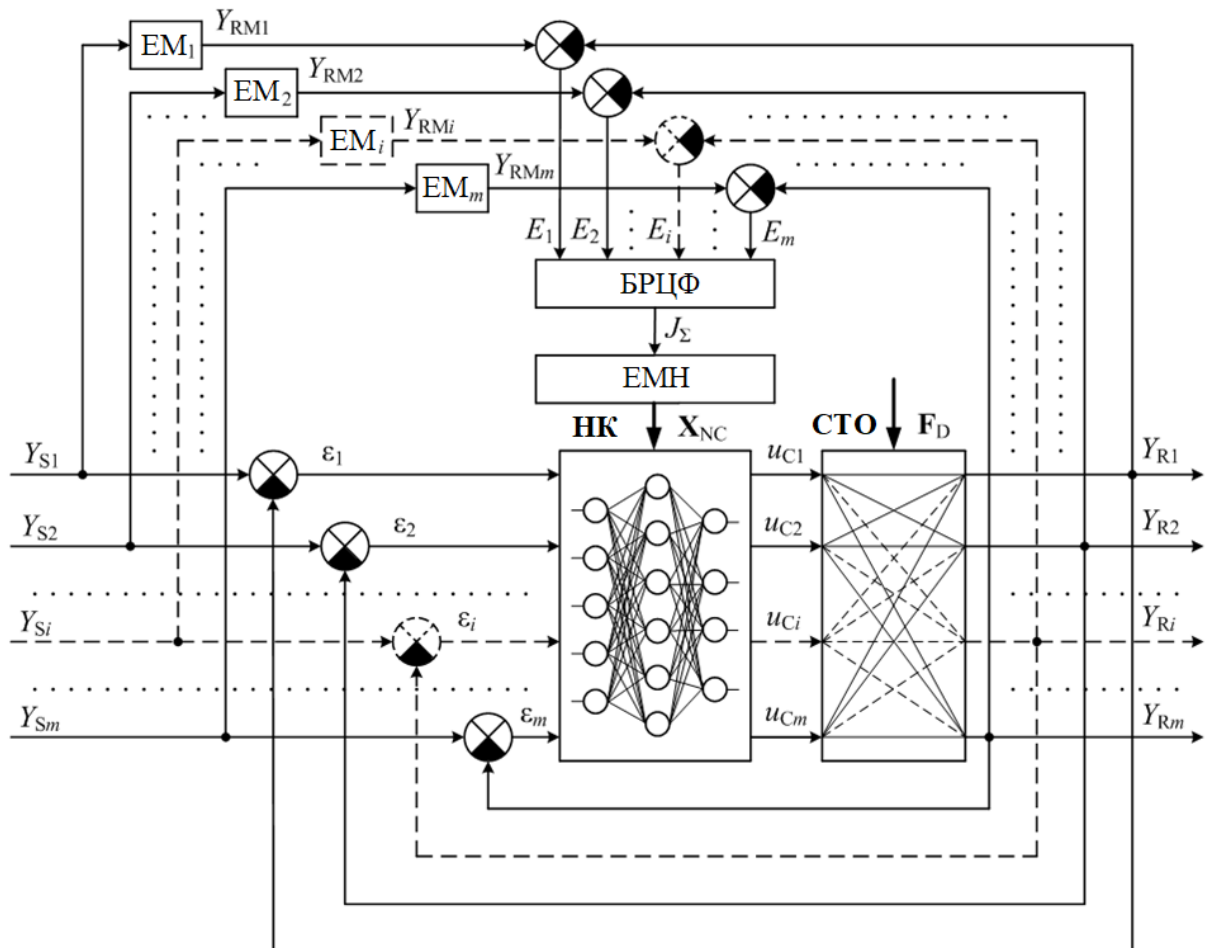


Рис. 3.2. Узагальнена структура багатозв'язної САК на основі єдиного комплексного нейроконтролера

Навчання комплексного НК системи керування з метою знаходження оптимального значення вектора  $\mathbf{X}_{NC}$  структурних елементів і параметрів багатошарової нейронної мережі реалізується за допомогою еволюційного механізму навчання (ЕНМ) з використанням поточного значення комплексної цільової функції  $J_\Sigma$ . У свою чергу, ЕНМ застосовує послідовне збільшення шарів і нейронів для визначення оптимальної

структури НК, а також генетичний алгоритм для знаходження його оптимальних параметрів (синаптичних вагових коефіцієнтів).

*Крок 2.* Вибір структур і параметрів еталонних моделей процесів керування змінними СТО. На даному етапі обираються структури та параметри еталонних моделей залежно від бажаних перехідних процесів об'єкта. При цьому, для достатньо великої кількості об'єктів керування бажаний перехідний процес повинен мати аперіодичний характер з мінімальною тривалістю [142]. Таким чином, еталонні моделі для змінних СТО можуть бути описані передаточними функціями  $W_{RMi}(p)$  наступного вигляду

$$W_{RMi}(p) = \frac{Y_{RMi}(p)}{Y_{Si}(p)} = \frac{1}{(T_{RMi}p + 1)^{v_i}}, i = \{1, \dots, m\}, \quad (3.1)$$

де  $T_{RMi}$  та  $v_i$  – стала часу та порядок передаточної функції  $i$ -ї еталонної моделі.

*Крок 3.* Вибір структури, параметрів та оптимального значення комплексної цільової функції системи керування СТО. На даному етапі обирається структура, параметри та оптимальне значення комплексної цільової функції  $J_\Sigma$  САК. При цьому, задача синтезу структури та параметрів нейромережевого контролера є задачею багатокритеріальної оптимізації, в якій здійснюється пошук оптимуму множини цільових функцій [46, 48]. У свою чергу, у різних задачах багатокритеріальної оптимізації комплексна цільова функція може бути обчислена різними способами. У даному випадку цільову функцію  $J_\Sigma$  доцільно розраховувати на основі методу агрегувальних функцій [48] згідно з виразом (3.2)

$$J_\Sigma = \sum_{i=1}^m k_i J_i, i = \{1, \dots, m\}, \quad (3.2)$$

де  $J_i$  – цільова функція  $i$ -ї керованої змінної САК;  $k_i$  – коефіцієнт, який нормалізує значення  $i$ -ї цільової функції  $J_i$  та визначає важливість якості керування  $i$ -ю змінною об'єкта. Наприклад, якщо в якості цільової функції

$i$ -ї керованої змінної обрати середню інтегральну квадратичну помилку, тоді комплексна цільова функція САК матиме вигляд

$$J_{\Sigma} = \sum_{i=1}^m \frac{k_i}{t_{\max}} \int_0^{t_{\max}} (E_i)^2 dt, i = \{1, \dots, m\}, \quad (3.3)$$

де  $t_{\max}$  – загальний час перехідного процесу СТО.

У свою чергу, оптимальне значення комплексної цільової функції  $J_{\Sigma}$  вибирається з урахуванням властивостей конкретного багатозв'язного об'єкта.

*Крок 4.* Вибір початкової структури та обмежень багатошарової нейронної мережі системи керування СТО. На даному етапі вибирається початкова структура та обмеження багатошарової нейронної мережі НК. Оскільки, при застосуванні даного нейроеволюційного методу, розмір хромосом на пряму залежить від кількості синаптичних вагових коефіцієнтів  $i$ , відповідно, від кількості шарів мережі та нейронів у кожному шарі, то початкову архітектуру нейронної мережі доцільно обирати максимально простою, щоб зменшити обчислювальні витрати під час реалізації GA на кроці 5. Далі, при необхідності, кількість нейронів і шарів доцільно поступово збільшувати на кроках 7 і 8, відповідно.

Таким чином, спочатку в мережі встановлюються три шари: вхідний, прихований і вихідний. Кількість нейронів у вхідному та вихідному шарах встановлюється рівною кількості входів та виходів НК, відповідно до його структури та загальної структури системи керування. У свою чергу, кількість нейронів прихованого шару попередньо може бути встановлена рівною кількості виходів НК. Крім того, на цьому кроці вибирається максимально можлива кількість прихованих шарів і нейронів в одному прихованому шарі, а також типи міжнейронних зв'язків і функцій активації. Обмеження на максимально можливу кількість прихованих шарів і нейронів в одному прихованому шарі вибираються таким чином, щоб процес синтезу синаптичних ваг за допомогою GA з відповідним

максимальним розміром хромосом мав прийнятні обчислювальні та часові витрати.

*Крок 5.* Синтез синаптичних ваг нейронної мережі для обраної архітектури за допомогою ГА. На даному кроці пошук оптимальних значень ваг НК здійснюється за допомогою генетичного алгоритму згідно з наступними кроками.

*Крок 5.1.* Кодування ваг мережі НК у хромосому. Хромосома складається з генів, кожен з яких є ваговим коефіцієнтом мережі НК. Для представлення генів можна використовувати двійкові чи дійсні числа. Наприклад, ваги синапсів між першим вхідним нейроном і першим нейроном прихованого шару кодуються як  $w_{11}$ , між другим вхідним нейроном і третім нейроном прихованого шару –  $w_{23}$ . У цьому випадку всі синаптичні ваги між вхідним і прихованим шарами утворюють вектор  $\mathbf{W}_1$ , між прихованим і вихідним шарами – вектор  $\mathbf{W}_2$ .

*Крок 5.2.* Вибір генетичних операторів. Основними операторами ГА є селекція, кросовер і мутація. Існують різні модифікації цих трьох операторів. Вибір того чи іншого варіанту впливає на швидкість збіжності і якість отриманих рішень. На цьому етапі, також, встановлюються значення ймовірності кросоверу  $P_C$  і мутації  $P_M$ .

*Крок 5.3.* Створення початкової популяції хромосом. На даному етапі створюється початкова популяція з  $Z_{\max}$  хромосом, що містить інформацію про  $Z_{\max}$  різних варіантів значень вагових коефіцієнтів для даної структури нейромережі НК. У свою чергу, початкова популяція  $P_0$  описується виразом (2.2). Значення генів хромосом у даному випадку доцільно встановити випадковим чином.

*Крок 5.4.* Оцінювання хромосом поточної популяції. Оцінювання хромосом проводиться згідно з наступними кроками.

*Крок 5.4.1.* Декодування хромосом. На цьому етапі здійснюється декодування кожної хромосоми популяції у набір вагових коефіцієнтів нейромережі НК.



*Крок 5.4.2.* Побудова нейронних мереж НК. На даному етапі здійснюється побудова мереж НК, які відповідають усім хромосомам популяції.

*Крок 5.4.3.* Розрахунок фітнес-функції для всіх хромосом популяції. На цьому етапі проводиться моделювання системи керування СТО із розробленим НК та розраховується значення фітнес-функції  $f_Z(N)$  для кожної  $Z$ -ї хромосоми популяції ( $Z = 1, \dots, Z_{\max}$ ). При цьому, значення фітнес-функції розраховуються на основі відповідних значень комплексної цільової функції  $J_{\Sigma}(N)$  САК. Оскільки ГА є алгоритмами максимізації функцій, то для мінімізації комплексної цільової функції  $J_{\Sigma}(N)$  фітнес-функцію  $f_Z(N)$  для кожної  $Z$ -ї хромосоми на кожній  $N$ -й ітерації доцільно розраховувати наступним чином:

$$f_Z(N) = \frac{1}{J_{\Sigma Z}(N)}. \quad (3.4)$$

Також, моделювання багатозв'язної САК виконується у всіх можливих режимах роботи об'єкта (при різних вхідних і збурювальних впливах), і на основі рівняння (3.2) розраховується комплексна цільова функція  $J_{\Sigma}(N)$ .

*Крок 5.5.* Перевірка критеріїв завершення пошуку. Пошук може вважатися завершеним, якщо виконується одна з наступних умов.

*Умова 1.* Досягнення оптимального значення фітнес-функції  $f_{\text{opt}}$ , що відповідає оптимальному значенню комплексної цільової функції  $J_{\Sigma\text{opt}}$  для САК з синтезованим НК. Якщо значення фітнес-функції  $f$  деяких хромосом досягає  $f_{\text{opt}}$  в процесі функціонування ГА, то критерій вважається досягнутим.

*Умова 2.* Виконання максимальної кількості ітерацій  $N_{\max}$ . У цьому випадку максимальна кількість ітерацій встановлюється попередньо, перед початком генетичного пошуку.

*Умова 3.* Виродження популяції. У даному випадку перед початком реалізації ГА необхідно встановити кількість ітерацій  $N_p$  та поріг

коефіцієнта покращення значень фітнес-функції найкращої хромосоми  $\rho_p$ . Починаючи з ітерації  $N_p$ , на кожній наступній ітерації ( $N_p + 1$ ) обчислюється коефіцієнт покращення  $\rho$  і порівнюється з пороговим значенням  $\rho_p$ . У свою чергу, коефіцієнт покращення  $\rho$  розраховується на основі виразу (2.1).

У разі виникнення однієї з наведених вище умов здійснюється перехід до кроку 5.10.

*Крок 5.6.* Відбір хромосом для кросоверу. На цьому етапі хромосоми популяції відбираються для кросоверу з метою створення нових рішень на основі значень фітнес-функції. У даному випадку доцільно використовувати пропорційний відбір, який реалізується за допомогою наступних кроків.

*Крок 5.6.1.* Визначення середнього значення фітнес-функції популяції. Визначення середнього значення фітнес-функції популяції  $f_M(N)$  як середнього арифметичного від значень фітнес-функції всіх індивідів здійснюється відповідно до рівняння (2.3).

*Крок 5.6.2.* Розрахунок коефіцієнта відбору. На даному кроці значення коефіцієнта відбору  $P_S^Z(N)$  для кожної  $Z$ -ї хромосоми розраховується згідно з виразом (2.4).

*Крок 5.6.3.* Формування масиву хромосом для кросоверу. На цьому етапі в залежності від значення  $P_S^Z(N)$  формується масив хромосом, допущених до кросоверу (якщо  $P_S^Z(N) > 1$ , тоді хромосома вважається добре пристосованою і допускається до кросоверу).

*Крок 5.7.* Кросовер хромосом. На даному етапі застосовується оператор кросоверу для хромосом, вибраних на попередньому кроці. У цьому випадку застосовується випадковий відбір батьківських пар.

*Крок 5.7.1.* Нумерація хромосом. На цьому етапі нумерація всіх хромосом поточної популяції здійснюється довільним чином.

*Крок 5.7.2.* Вибір першого з батьків. На даному етапі випадковим чином обирається число на інтервалі  $[0, 1]$  для кожної  $Z$ -ї хромосоми, починаючи з першої. Першим з батьків у парі буде перша хромосома, для якої дане випадкове число не менше ймовірності кросоверу  $P_C$ .

*Крок 5.7.3.* Вибір другого з батьків. Перегляд популяції продовжується, починаючи з рішення, наступного за першим батьком (вибраного на попередньому кроці), доки знову випадково вибране число не стане не меншим за  $P_C$ . Індивід, для якого виконується така умова, буде другим з батьків. Крок 5.7.3 продовжується до тих пір, поки не буде обрано необхідну кількість пар батьків.

*Крок 5.7.4.* Кросовер обраних батьківських пар хромосом. На цьому етапі для кожної батьківської пари застосовується одноточковий кросовер. У свою чергу, точка розриву хромосоми вибирається випадковим чином.

*Крок 5.8.* Мутація хромосом. Оператор мутації застосовується для хромосом, вибраних на кроці 5.6 відповідно до ймовірності мутації  $P_M$  (встановленої на кроці 5.2). У даному випадку застосовується проста мутація.

*Крок 5.8.1.* Копіювання батьківських хромосом у хромосоми-нащадки. На цьому етапі батьківські хромосоми, допущені до мутацій, копіюються у хромосоми-нащадки.

*Крок 5.8.2.* Вибір гена, що мутує. Ген, що мутує, для кожної хромосоми, допущеної до мутації, вибирається випадковим чином.

*Крок 5.8.3.* Вибір нового значення гена. Вибір нового значення гена, яке не дорівнює поточному, реалізується з набору всіх можливих значень параметра, відповідного гену, що мутує.

*Крок 5.9.* Формування нового покоління. На даному кроці формується нове покоління з елітних хромосом і хромосом-нащадків, отриманих шляхом кросоверу та мутації. Потім здійснюється перехід до кроку 5.4.

*Крок 5.10.* Завершення процесу реалізації GA.

*Крок 6.* Перевірка критеріїв завершення генетичного пошуку. На цьому етапі перевіряються критерії завершення генетичного пошуку, і при виконанні умови 1 здійснюється перехід до кроку 9. Якщо виконуються умови 2 або 3 – здійснюється перехід до наступного кроку.

*Крок 7.* Додавання одного нейрона до останнього прихованого шару. Якщо останній прихований шар вже має максимально допустиму кількість нейронів, то здійснюється перехід до наступного кроку. У протилежному випадку на цьому кроці до останнього прихованого шару додається один нейрон і виконується перехід до кроку 5.

*Крок 8.* Додавання нового прихованого шару. На даному кроці додається новий прихований шар, який складається з одного нейрона перед шаром виходів. Якщо досягнуто максимальної кількості прихованих шарів, здійснюється перехід до кроку 4. Також, може бути реалізовано повернення до кроків 3, 2 або 1 для зміни типів або параметрів фітнес-функції та еталонних моделей, а також вхідних змінних системи керування з НК. Якщо максимальна кількість прихованих шарів ще не досягнута, здійснюється перехід до кроку 5 і генетичний пошук стартує знову.

*Крок 9.* Реалізація розробленої багатозв'язної САК на основі НК.

Далі розглянемо приклад застосування наведеного нейроеволюційного методу для керування реальним складним багатозв'язним технічним об'єктом. Зокрема, у роботі [142] представлено розробку та дослідження ефективності багатозв'язної САК просторовим рухом багатоцільового мобільного робота (МР) вертикального переміщення на основі розглянутого нейроеволюційного методу.

*Розробка та дослідження ефективності багатозв'язної САК просторовим рухом багатоцільового МР на основі нейроеволюційного методу.* Багатоцільові гусеничні мобільні роботи, що здатні рухатися по похилих і вертикальних феромагнітних поверхнях, досить ефективно застосовуються для переміщення різних робочих інструментів за заданими траєкторіями з метою автоматичного виконання таких технологічних

операцій, як інспекція, очищення, видалення іржі, фарбування, зварювання тощо, у сільському господарстві, судноремонті, суднобудуванні, газо- та нафтопереробці, а також в інших галузях промисловості [143]. Однак, для реалізації вищенаведених операцій багатоцільові гусеничні МР повинні мати можливість рухатися по заданих траєкторіях в умовах невизначеності робочої поверхні, зумовленої їх технологічними особливостями, наявністю перешкод, конструктивних пошкоджень тощо [144]. Таким чином, автоматичне керування просторовим рухом під дією різнотипних збурень є однією з найважливіших задач автоматизації МР такого типу [143]. У свою чергу, ця задача передбачає одночасне керування двома взаємопов'язаними змінними МР: лінійною швидкістю переміщення та кутом повороту (курсом). Наведений взаємозв'язок змінних, а також ряд особливостей робота як нелінійного багатозв'язного об'єкта не дозволяють досягати високих показників якості керування при використанні окремих традиційних контролерів у його двоканальній САК [145]. Тому, для автоматизації просторового руху наведеного МР доцільно використати розглянутий нейроеволюційний метод керування.

Розробка багатозв'язної САК просторовим рухом у роботі [142] проведена для гусеничного МР з наступними параметрами: повна маса МР з обладнанням – 150 кг, довжина МР – 1 м, ширина МР – 0,8 м, радіус приводного колеса гусениці – 0,15 м, задана лінійна швидкість переміщення – 0,3 м/с. Математична модель МР складається з рівнянь (3.5)-(3.16):

$$D_M \frac{d\omega_{M_p}}{dt} = M_{EM_p} \eta_M - M_{L_p}; \quad (3.5)$$

$$M_{EM_p} = C_{mM} I_{M_p}; \quad (3.6)$$

$$L_M \frac{dI_{M_p}}{dt} + R_M I_{M_p} = u_{M_p} - C_{m\omega} \omega_{M_p}; \quad (3.7)$$

$$T_{III} \frac{du_{M_p}}{dt} + u_{M_p} = k_{III} u_{FC}; \quad (3.8)$$

$$\omega_{w\rho} = k_R \omega_{M\rho}; \quad (3.9)$$

$$v_R = (v_{C2} + v_{C1})/2; \quad (3.10)$$

$$v_{C1} = \omega_{MR} (R_T - 0,5B) = \omega_{w1} R_W; \quad (3.11)$$

$$\begin{aligned} M_{L\rho} = & \frac{1}{k_R \eta_{MR}} \left[ D_{\Sigma W} \frac{d\omega_{w\rho}}{dt} + R_W \left( \zeta \frac{G \cos \gamma + F}{2} + \right. \right. \\ & \left. \left. + G \sin \gamma \left( \frac{\cos \varphi_{MR}}{2} + \sigma \frac{x_O}{B} \sin \varphi_{MR} + \sigma \zeta \frac{h_C}{B} \sin \varphi_{MR} \right) - \right. \\ & \left. - \sigma \frac{\mu_T L (G \cos \gamma + F)}{4B} \left( 1 + \frac{4x_O^2}{L^2} \right) + F_{TO} \left( \frac{1}{2} \cos \beta - \frac{(x_1 - x_O)}{B} \sin \beta \right) + \right. \\ & \left. + \left( \frac{G \cos \gamma + F}{2g} + \sigma G \sin \gamma \sin \varphi_{MR} \frac{h_C}{gB} \right) \frac{dv_{C\rho}}{dt} + \right. \\ & \left. + \left( m_{MR} + \frac{F}{g} \right) \frac{L^2 + B^2}{12B} \frac{d\omega_{MR}}{dt} \right]; \end{aligned} \quad (3.12)$$

$$v_{C2} = \omega_{MR} (R_T + 0,5B) = \omega_{w2} R_W; \quad (3.13)$$

$$\omega_{MR} = v_R / R_T = (v_{C2} - v_{C1}) / B; \quad (3.14)$$

$$F = \frac{F_{TO} + G(\zeta \cos \gamma + \sin \gamma - \xi \cos \gamma) + m_{MR} a_{MR} \lambda}{\xi - \zeta}; \quad (3.15)$$

$$R_T = \frac{0,5B(v_{C2} + v_{C1})}{(v_{C2} - v_{C1})}, \quad (3.16)$$

де  $\omega_{M\rho}$  – кутова швидкість  $\rho$ -го двигуна електропривода гусеничного рушія;  $\rho$  та  $\sigma$  – коефіцієнти, що враховують напрямок повороту робота ( $\rho = 1$ ,  $\sigma = 1$  – для відстаючої гусениці;  $\rho = 2$ ,  $\sigma = -1$  – для забігаючої гусениці);  $D_M$  – сумарний момент інерції якоря двигуна;  $M_{EM\rho}$  – електромагнітний момент  $\rho$ -го двигуна електропривода;  $\eta_M$ ,  $\eta_{MR}$  – коефіцієнти корисної дії двигуна та MR, відповідно;  $M_{L\rho}$  – момент навантаження  $\rho$ -го двигуна електропривода;  $I_{M\rho}$  – струм якоря  $\rho$ -го двигуна;  $C_{mM}$  – коефіцієнт електромагнітного моменту, який визначається параметрами якоря двигуна і величиною його магнітного потоку;  $k_R$  – передавальне число редуктора електропривода;  $F_{TO}$

і  $F$  – значення питомих сил навантаження від технологічної операції та притискних магнітів, відповідно;  $G$  – загальна вага робота та технологічного обладнання;  $\gamma$  – кут нахилу робочої поверхні;  $\zeta$  – коефіцієнт тертя кочення;  $\xi$  – коефіцієнт адгезії;  $m_{MR}$  – маса робота;  $\lambda$  – приведений коефіцієнт маси,  $\lambda = 1,15 + 0,001 k_R^2$ ;  $a_{MR}$  – прискорення руху МР;  $v_{C1}, v_{C2}$  – поточні лінійні швидкості гусениць, що відстає та забігає;  $R_T$  – радіус повороту робота;  $\varphi_{MR}$  – поточний курс МР;  $\omega_{MR}$  – швидкість повороту робота,  $\omega_{MR} = d\varphi_{MR}/dt$ ;  $B$  – відстань між центрами гусениць;  $\omega_{W1}, \omega_{W2}$  – кутові швидкості відстаючого і забігаючого приводних коліс гусениць;  $R_W$  – радіус ведучого колеса;  $D_{\Sigma W}$  – сумарний момент інерції колеса та гусениці;  $L, h_C$  – довжина та висота центру тяжіння робота;  $x_O$  – відстань від поперечної осі робота до центрів повороту гусениць;  $x_1$  – відстань від поперечної осі робота до точки закріплення технологічного обладнання;  $\beta$  – кут відхилення лінії дії сили  $F_{TO}$  від поздовжньої осі робота;  $\mu_T$  – коефіцієнт опору повороту, що залежить від радіусу повороту;  $L_M$  та  $R_M$  – індуктивність та опір обмотки електродвигуна;  $u_{MP}$  – напруга живлення  $p$ -го двигуна електропривода;  $C_{m\omega}$  – коефіцієнт електрорушійної сили, який визначається параметрами якоря двигуна і величиною його магнітного потоку;  $T_{III}$  та  $k_{III}$  – стала часу та коефіцієнт підсилення перетворювача потужності.

На кроці 1 представленого методу побудована функціональна структура багатозв'язної САК просторовим рухом МР на основі єдиного комплексного нейроконтролера, зовнішній вигляд якої наведено на рис. 3.3. У свою чергу, на рис. 3.3 прийнято наступні позначення:  $v_{RS}, \varphi_{MRS}$  – задані значення швидкості та курсу руху МР;  $v_{RD}, \varphi_{MRD}$  – бажані значення швидкості та курсу руху МР, отримані на виходах відповідних еталонних моделей  $EM_v$  та  $EM_\varphi$ ;  $\varepsilon_v, \varepsilon_\varphi$  – помилки керування змінними МР;  $E_v, E_\varphi$  – розбіжності між вихідними сигналами ЕМ та реальними виходами мобільного робота.

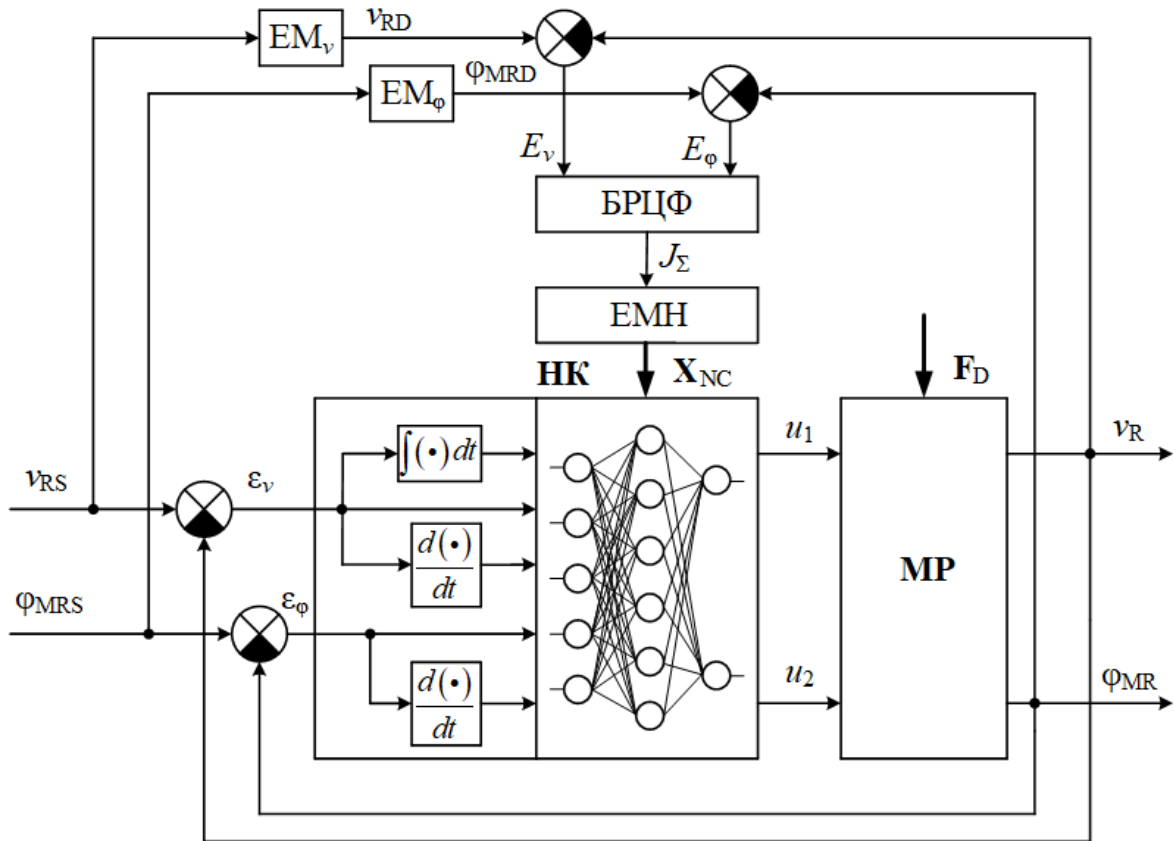


Рис. 3.3. Структура багатозв'язної САК просторовим рухом МР

Дана САК має комплексний нейроконтролер з п'ятьма входами:  $\varepsilon_v, \dot{\varepsilon}_v, \int \varepsilon_v dt, \varepsilon_\varphi, \dot{\varepsilon}_\varphi$  і двома виходами:  $u_1$  та  $u_2$ , які, у свою чергу, є сигналами керування приводами відповідних гусениць робота.

На кроці 2 обрано дві еталонні моделі для змінних МР  $v_R$  і  $\varphi_{MR}$ , які визначаються наступними передаточними функціями:

$$W_{RM_v}(p) = \frac{v_{RD}(p)}{v_{RS}(p)} = \frac{1}{(T_{RM_v}p + 1)^2}; \quad (3.17)$$

$$W_{RM_\varphi}(p) = \frac{\varphi_{MRD}(p)}{\varphi_{MRS}(p)} = \frac{1}{(T_{RM_\varphi}p + 1)^2}, \quad (3.18)$$

де  $T_{RM_v}, T_{RM_\varphi}$  – сталі часу передаточних функцій ЕМ ( $T_{RM_v} = 0,35; T_{RM_\varphi} = 0,21$ ).

На кроці 3 задана комплексна цільова функція  $J_\Sigma$  системи керування МР у вигляді



$$J_{\Sigma} = J_v + J_{\phi} = \frac{k_v}{t_{\max}} \int_0^{t_{\max}} (E_v)^2 dt + \frac{k_{\phi}}{t_{\max}} \int_0^{t_{\max}} (E_{\phi})^2 dt, \quad (3.19)$$

де  $k_v$  та  $k_{\phi}$  – коефіцієнти, що нормалізують значення окремих цільових функцій  $J_v$  та  $J_{\phi}$  контурів керування швидкістю та курсом відповідно ( $k_v = 22$ ;  $k_{\phi} = 1$ ).

У свою чергу, оптимальне значення комплексної цільової функції  $J_{\Sigma \text{opt}}$  обрано рівним 160.

На кроці 4 обрано початкову структуру та обмеження багатошарової нейронної мережі НК для мобільного робота. У свою чергу, початкова структура вибрана наступним чином: п'ять вхідних нейронів, два нейрони в одному прихованому шарі та два вихідні нейрони (5-2-2). У даному випадку хромосома генетичного алгоритму мінімального розміру складається з 14 генів. Що стосується обмежень, то максимально можлива кількість прихованих шарів становить 5, а максимально можлива кількість нейронів в одному прихованому шарі – 4. Таким чином, максимальний розмір хромосоми становить 92 гени, що дозволяє реалізувати процес синтезу синаптичних вагових коефіцієнтів мережі з прийнятними обчислювальними та часовими витратами. Крім того, на цьому кроці для двох вихідних нейронів обрано лінійні функції активації, а для інших нейронів – функції гіперболічного тангенсу.

Подальше навчання комплексного НК САК просторовим рухом МР з метою пошуку оптимальної структури та параметрів багатошарової нейронної мережі реалізовано за допомогою еволюційного механізму навчання на наступних кроках методу. У свою чергу послідовне збільшення шарів і нейронів здійснювалося згідно з кроками 7 і 8, а генетичний пошук оптимальних вагових коефіцієнтів зв'язків – відповідно до кроку 5. Кількість хромосом  $Z_{\max}$  у популяції дорівнювала 150, максимальна кількість ітерацій  $N_{\max} = 100$ .

У результаті проведеного навчання знайдено оптимальну структуру мережі НК: п'ять нейронів у вхідному шарі, перший і другий приховані

шари складаються з чотирьох і трьох нейронів, відповідно, а також два нейрони у вихідному шарі (5-4-3-2). Для даної структури кожна хромосома ГА складалася з 38 генів, а найкраще значення комплексної цільової функції  $J_{\Sigma_{\text{best}}} = 159,33$  було досягнуте на 87-й ітерації. Графік зміни значення комплексної цільової функції (3.19) у процесі навчання нейроконтролера для отриманої структури представлено на рис. 3.4.

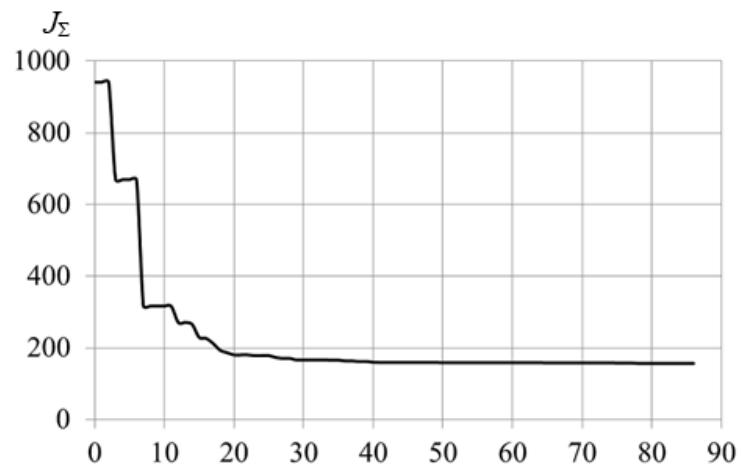


Рис. 3.4. Графік зміни значення комплексної цільової функції у процесі навчання нейроконтролера

Для дослідження ефективності розробленого НК на основі нейроеволюційного методу проведено моделювання перехідних процесів САК просторовим рухом гусеничного робота. Також, здійснено порівняння отриманих результатів моделювання для двох координат МР  $v_R$  та  $\varphi_{MR}$  із аналогічною САК з окремими традиційними регуляторами швидкості та курсу. У свою чергу, традиційні ПД-регулятор швидкості та ПД-регулятор курсу налаштовані шляхом параметричної оптимізації за допомогою градієнтного методу. Знайдені значення їх коефіцієнтів складають:  $k_{Pv} = 6,5$ ;  $k_{Iv} = 422,4$ ;  $k_{Dv} = 0,23$  – для ПД-регулятора швидкості і  $k_{P\varphi} = 48,5$ ;  $k_{D\varphi} = 5,2$  – для ПД-регулятора курсу. Також, у даному випадку збурення моделювалося як постійно діюча на мобільний робот

максимально допустима сила навантаження від технологічної операції  $F_{TO} = 1000$  Н.

Графіки перехідних процесів САК для ступінчатих змін заданих значень швидкості  $v_{RS} = 0,3$  м/с та курсу  $\varphi_{MRS} = \pi/3$  рад, а також при куті нахилу робочої поверхні  $\gamma = 60^\circ$  наведено на рис. 3.5 та 3.6, відповідно.

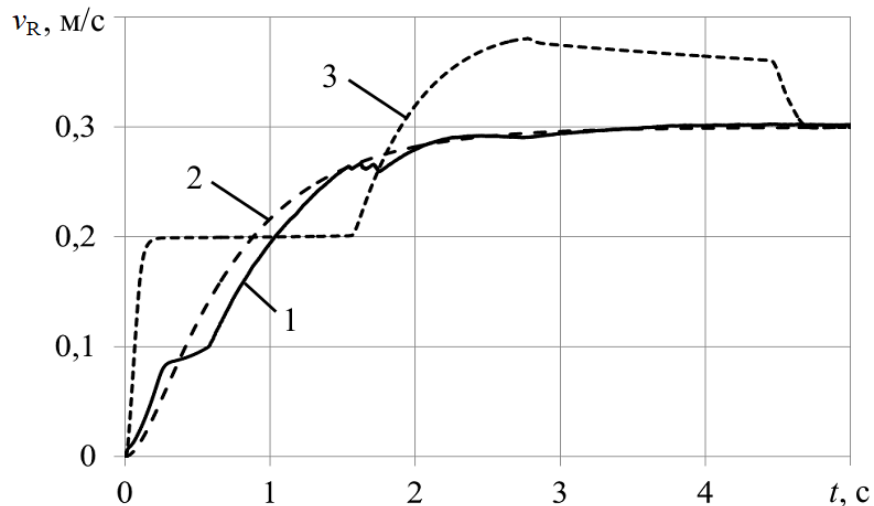


Рис. 3.5. Графіки перехідних процесів САК за швидкістю: 1 – з НК просторового руху; 2 – еталонна модель; 3 – з традиційним ПІД-регулятором швидкості

Порівняльний аналіз показників якості САК просторовим рухом МР із розробленим НК та оптимально налаштованими традиційними контролерами наведено в табл. 3.1, де  $t_t$  – час перехідного процесу;  $\Delta$  – статична похибка.

Як видно з табл. 3.1, рис. 3.5 та рис. 3.6, розроблена САК просторовим рухом МР на основі розглянутого нейроеволюційного методу має значно вищі показники якості при керуванні швидкістю та курсом порівняно з використанням аналогічної САК з окремими традиційними регуляторами даних змінних.

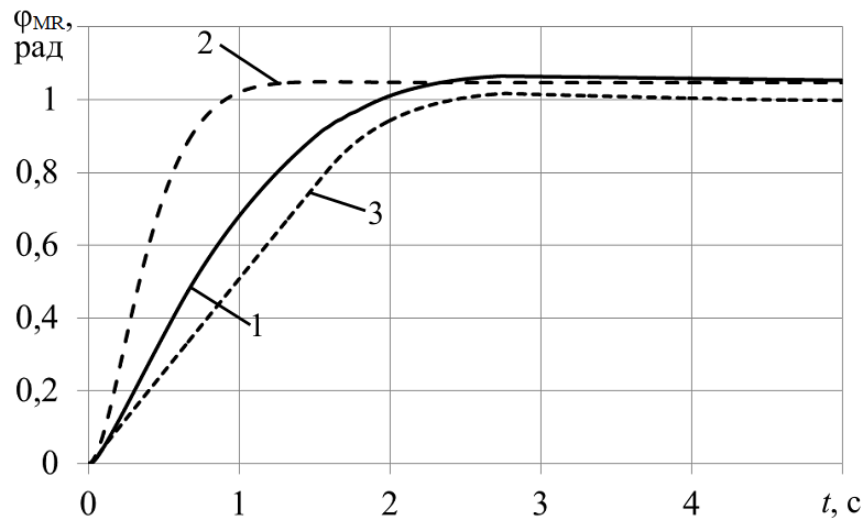


Рис. 3.6. Графіки переходних процесів САК за курсом: 1 – з НК просторового руху; 2 – еталонна модель; 3 – з традиційним ПД-регулятором курсу

Таблиця 3.1. Аналіз показників якості САК просторовим рухом МР

Моделювання САК	Показники якості САК			
	$t_t, c$	$\Delta, \%$	$J_v$	$J_\phi$
$EM_v$	2,11	0	0	-
Швидкість з традиційним ПД-регулятором	4,6	1	22,7	-
Швидкість з НК	2,09	0,7	0,8	-
$EM_\phi$	0,89	0	-	0
Курс з традиційним ПД-регулятором	2,36	4,8	-	292,4
Курс з НК	1,91	0,6	-	138,1

Зокрема, використання багатозв'язної САК з єдиним комплексним НК дозволяє скоротити час переходних процесів регулювання швидкості та курсу на 55 % та 19 %, відповідно, а також, усунути небажане перерегулювання у 27 % при керуванні швидкістю та значно зменшити (у 8 разів) статичну помилку при керуванні курсом. При цьому, статична

помилка регулювання швидкості МР становить менше 1%, що не є критичним при виконанні різних технологічних операцій.

Синтезована САК просторовим рухом МР на базі нейроконтролера показує вищу якість керування, краще враховує взаємний вплив каналів керування швидкістю та курсом робота, що, відповідно, дозволяє значно підвищити продуктивність та якість різних технологічних операцій. Таким чином, наведені дослідження підтверджують високу ефективність розглянутого нейроеволюційного методу керування складними багатозв'язними об'єктами. Застосування даного підходу при проектуванні САК для різнотипних СТО дає можливість враховувати взаємовплив їх керуючих змінних і, як наслідок, значно підвищувати показники якості керування та загальну ефективність, забезпечуючи оптимальні режими роботи.

### **3.2 Мультиагентний метод генерації баз правил нечітких систем на основі мурашиних алгоритмів**

*Постановка задачі синтезу бази правил нечіткої системи.* Успішне застосування нечітких систем різних типів (Мамдані, Такагі-Сугено, Цукамото, Ларсена та ін.) вимагає вирішення двох основних завдань: 1) визначення оптимальної структури НС; 2) знаходження оптимальних параметрів системи [2, 6]. Перше завдання включає вибір: кількості вхідних і вихідних змінних НС; кількості лінгвістичних термів для кожної вхідної та вихідної змінної; кількості правил бази правил; типів функцій належності кожного ЛТ; типів процедур агрегації, активації та акумуляції, а також методу дефазифікації [2, 8]. Друге завдання, у свою чергу, полягає (а) у визначенні параметрів функцій належності ЛТ кожної вхідної та вихідної змінної та нормуючих коефіцієнтів для їх приведення до відносних одиниць, а також (б) у генерації консеквентів правил для БП нечіткої системи [6, 8].

У багатьох випадках оптимальні структура та параметри НС для вирішення тієї чи іншої задачі визначаються на основі експертних оцінок та рекомендацій [3]. При цьому суб'єктивний фактор має значний вплив на процес розробки НС. В умовах недостатньо повного обсягу вихідної інформації та знань експертів, а також при прийнятті помилкових проєктних рішень ефективність нечітких систем може суттєво знижуватися або їх функціонування здійснюватиметься за занижених, з погляду потенційних можливостей, показників. Для підвищення ефективності функціонування НС, а також для зниження негативного впливу суб'єктивних факторів на процес проєктування вченими різних країн світу останнім часом розробляються та впроваджуються методи синтезу, які базуються на певних оптимізаційних процедурах [133, 146, 147]. Зокрема, методи та алгоритми структурної оптимізації НС на основі оптимального вибору типів функцій приналежності ЛТ, методів дефазифікації, редукції та інтерполяції БП наведені у [46, 148]. У свою чергу, в роботах [149, 150] представлені методи синтезу, що включають процедури параметричної оптимізації функцій належності лінгвістичних термів НС типу Мамдані, а також генерації вагових коефіцієнтів для консеквентів правил систем типу Такагі-Сугено.

Серед наведених вище задач синтезу та оптимізації інтелектуальних систем на основі нечіткої логіки особливої уваги заслуговує задача синтезу високоефективних баз правил з оптимальним набором консеквентів для НС типу Мамдані при недостатньому обсязі вихідної інформації (в умовах високого ступеня невизначеності інформації) та знань експертів.

На початковому етапі синтезу нечіткої системи типу Мамдані формується вектор вхідних змінних  $\mathbf{X}$  ( $x_1, x_2, \dots, x_i, \dots, x_m$ ) та вихідна змінна  $y$  [8]. Після цього здійснюється вибір кількості лінгвістичних термів  $w_i$  для кожної  $i$ -ї вхідної змінної вектора  $\mathbf{X}$  ( $i = 1, \dots, m$ ) і лінгвістичних термів  $v$  для вихідної змінної  $y$  НС [3, 8]. Крім того, попередньо

вибираються типи та параметри функцій належності ЛТ для кожної вхідної та вихідної змінної НС.

У свою чергу, база правил включає в себе набір правил, що складаються з відповідних антецедентів (умов) і консеквентів (висновків). Загальна кількість правил  $s$  БП системи визначається кількістю всіх можливих комбінацій ЛТ вхідних змінних НС  $\mathbf{X}$  на основі залежності [8]

$$s = \prod_{i=1}^m w_i. \quad (3.20)$$

У свою чергу, кожне  $r$ -те правило БП ( $r = 1, \dots, s$ ) є лінгвістичним твердженням виду [10]

$$\begin{aligned} \text{IF } "x_1 = a" \text{ AND } "x_2 = b" \text{ AND } \dots \text{ AND } "x_i = c" \dots \\ \dots \text{ AND} \dots \text{ AND } "x_m = d" \text{ THEN } "y = h", \end{aligned} \quad (3.21)$$

де  $a, b, c, d, h$  – відповідні лінгвістичні терми вхідних та вихідної змінної НС.

Антецеденти правил являють собою різні комбінації лінгвістичних термів вхідних змінних НС [8], а консеквент  $LT_r$  кожного  $r$ -го правила БП ( $r = 1, 2, \dots, s$ ) обирається з множини всіх можливих консеквентів правил  $\{LT^1, LT^2, \dots, LT^v\}$ , яка включає у себе  $v$  лінгвістичних термів вихідної змінної у НС

$$LT_r \in \{LT^1, LT^2, \dots, LT^v\}. \quad (3.22)$$

Так, при виборі кількості лінгвістичних термів вихідної змінної  $v = 3$  має місце множина, наприклад,

$$LT_r \in \{LT^1, LT^2, LT^3\} = \{S, M, B\}, \quad (3.23)$$

а при  $v = 5$ , відповідно,

$$LT_r \in \{LT^1, LT^2, LT^3, LT^4, LT^5\} = \{S, LM, M, HM, B\}, \quad (3.24)$$

де S – малий; LM – нижче середнього; M – середній; HM – вище середнього; B – великий.

Вектор консеквентів  $\mathbf{R}$  БП може бути сформований по-різному, при цьому задача оптимізації консеквентів зводиться до задачі знаходження оптимального вектора консеквентів  $\mathbf{R}_{\text{opt}}$  з багатьох можливих альтернативних варіантів, який забезпечує оптимальні показники якості НС.

Вектор консеквентів  $\mathbf{R}_\gamma$  БП для  $\gamma$ -го альтернативного варіанта БП у загальному вигляді можна подати наступним чином

$$\begin{aligned} \mathbf{R}_\gamma &= \{LT_{\gamma 1}, LT_{\gamma 2}, \dots, LT_{\gamma r}, \dots, LT_{\gamma s}\}, \\ LT_{\gamma r} &\in \{LT^1, LT^2, \dots, LT^v\}, \gamma \in \{1, 2, \dots, v^s\}, \end{aligned} \quad (3.25)$$

де  $v^s$  – кількість усіх можливих варіантів вектора  $\mathbf{R}$ , яке визначається як кількість ЛТ вихідної змінної  $v$ , що піднесене до ступеня загальної кількості правил БП  $s$ .

Таким чином, задача генерації консеквентів БП НС зводиться до знаходження такого вектора консеквентів БП  $\mathbf{R}_{\text{opt}} = \mathbf{R}_\gamma$ , при якому цільова функція НС  $J$  досягне оптимального значення ( $J \leq J_{\text{opt}}$ ) [151].

Ця задача є складною задачею дискретної оптимізації великої розмірності, для вирішення якої необхідна наявність ефективного стохастичного методу глобального пошуку, який враховуватиме особливості формування консеквентів правил БП за умов невизначеності інформації. При вирішенні цієї задачі за допомогою повного перебору всіх можливих векторів  $\mathbf{R}_\gamma$  ( $\gamma = 1, 2, \dots, v^s$ ) значення цільової функції  $J$  необхідно буде обчислити  $v^s$  разів, що навіть при невеликій розмірності БП вимагатиме значних обчислювальних та часових витрат [151]. Наприклад, для НС з досить простою структурою з дев'ятьма правилами ( $s = 9$ ) у БП і трьома лінгвістичними термами вихідної змінної ( $v = 3$ ) значення цільової функції  $J$  при повному переборі  $\mathbf{R}_\gamma$  необхідно обчислити  $3^9 = 19683$  рази, що займе досить багато часу при використанні суттєвих обчислювальних ресурсів.



Для ефективного вирішення представленої задачі у роботі [152] запропоновано мультиагентний метод генерації баз правил нечітких систем на основі мурашиних алгоритмів.

*Базові аспекти та основні кроки мультиагентного методу генерації баз правил нечітких систем.* Основною ідеєю запропонованого у [152] мультиагентного методу є представлення структури бази правил НС у вигляді спеціального графа (рис. 3.7), а також подання пунктів призначення  $DP_{rj}$  у вигляді консеквентів  $LT_r^j$  ( $r \in \{1, \dots, s\}; j \in \{1, \dots, v\}$ ) правил БП, по яких переміщуються агенти-мурахи.

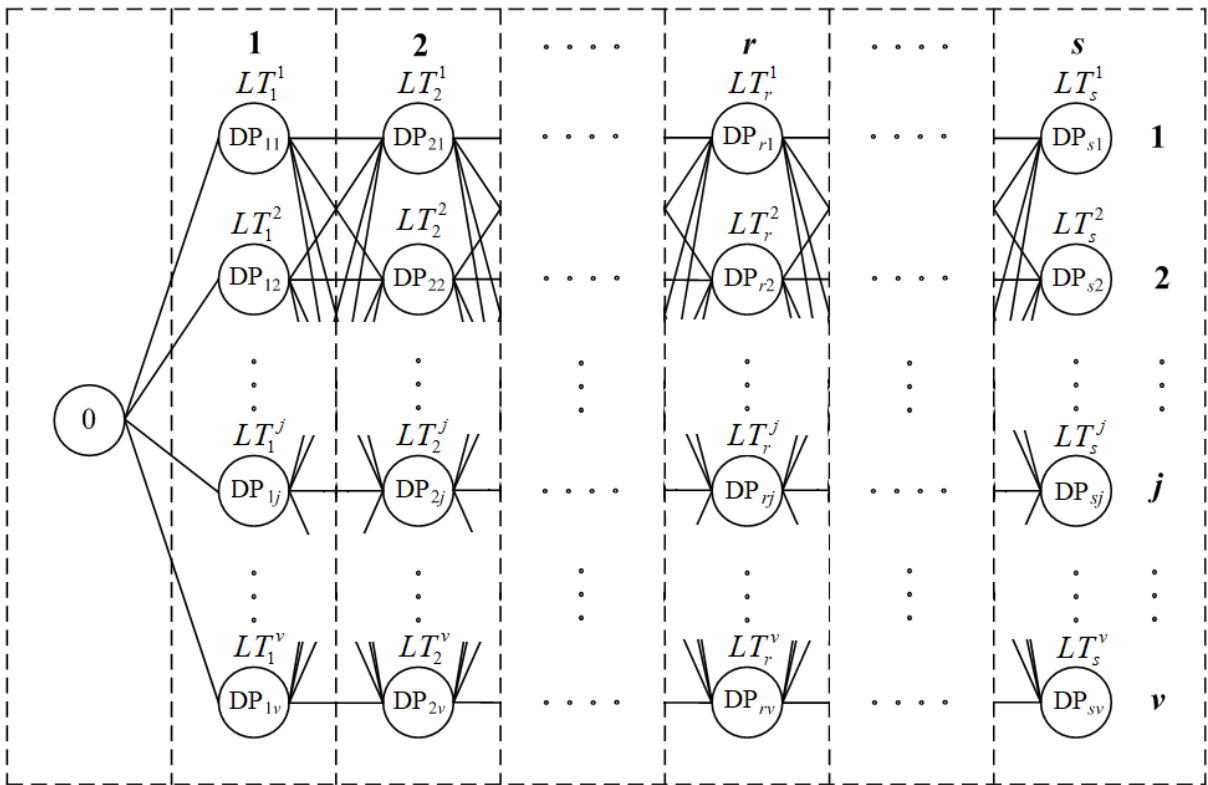


Рис. 3.7. Граф бази правил нечіткої системи

Даний граф складається з шарів, вузлів та ребер. Шлях кожного  $k$ -го агента проходить через певні вузли та ребра графа на кожній окремій  $n$ -й ітерації. Довжина шляху, який пройшов певний агент, є абстрактною величиною та визначається значенням цільової функції  $J$  нечіткої системи. Наприклад, для нечіткої системи автоматичного керування в якості

цільової функції  $J$  може бути обрана середня інтегральна квадратична помилка керування

$$J(t, \mathbf{R}) = \frac{1}{t_{\max}} \int_0^{t_{\max}} \varepsilon^2 dt \rightarrow \min, \quad (3.26)$$

де  $t_{\max}$  – загальний час перехідного процесу САК;  $\varepsilon$  – помилка керування.

Структура даного графа відповідає структурі бази правил НС та визначається загальною кількістю нечітких правил  $s$  та кількістю їх можливих консеквентів  $v$ . Граф БП складається з одного нульового шару, який має один вузол і є початковою відправною точкою для всіх  $z$  агентів популяції, і  $s$  основних шарів, кожен з яких має порядковий номер  $r$  і відповідає певному  $r$ -му правилу БП ( $r = 1, 2, \dots, s$ ). Крім того, кожен  $r$ -й шар має  $v$  вузлів, що відповідає кількості можливих консеквентів правил БП і, безпосередньо, кількості лінгвістичних термів вихідної змінної у НС. У свою чергу, кожен  $rj$ -й вузол шару з порядковим номером  $r$  даного графа відповідає  $j$ -му можливому консеквенту  $r$ -го нечіткого правила сформованої БП НС ( $r = 1, 2, \dots, s; j = 1, 2, \dots, v$ ).

Після проходження кожним  $k$ -м агентом певних вузлів усіх  $s$  шарів даного графа на кожній окремій ітерації з номером  $n$ , формується відповідна база правил НС з певним вектором консеквентів  $\mathbf{R}^k(n)$  ( $k = 1, \dots, z; n = 1, \dots, n_{\max}$ ).

Особливістю даного графа БП є те, що його ребра встановлені лише між вузлами різних шарів (ребра між вузлами одного шару відсутні). Перехід  $k$ -го агента з будь-якого  $rj$ -го вузла шару з порядковим номером  $r$  можливий тільки в будь-який  $(r+1)q$ -й вузол наступного шару з порядковим номером  $r+1$  ( $r \in \{1, \dots, s\}; j, q \in \{1, \dots, v\}$ ). Таким чином, рух кожного  $k$ -го агента на кожній окремій ітерації з номером  $n$  починається з нульового шару і здійснюється послідовно через всі  $s$  шарів графа від першого до останнього тільки в напрямку збільшення порядкового номера шарів  $r$ . Кількість переходів кожного  $k$ -го агента по вузлах графа на одній

ітерації дорівнює загальній кількості шарів графа і, відповідно, нечітких правил  $s$ .

Наприклад, на рис. 3.8 наведено граф БП для НС з двома вхідними змінними  $x_1$  і  $x_2$  ( $m = 2$ ), кожна з яких оцінюється відповідними трьома ЛТ ( $w_1 = 3; w_2 = 3$ ), і з вихідною змінною  $y$ , яка має також 3 ЛТ ( $v = 3$ ). Таким чином, кількість правил БП і, відповідно, основних шарів графа згідно з виразом (3.20) становить  $s = 9$ . Кількість вузлів у кожному  $r$ -му шарі, у свою чергу, становить  $v = 3$ .

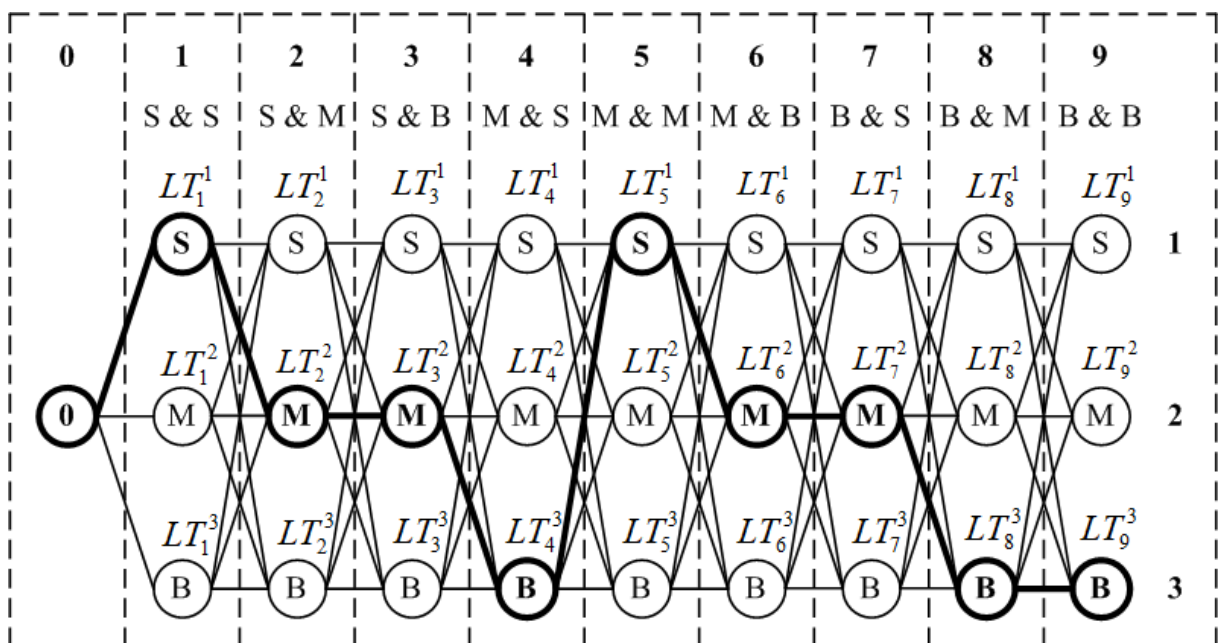


Рис. 3.8. Граф бази правил НС з параметрами:  $m = 2; w_1 = 3; w_2 = 3; v = 3$

Для вхідних та вихідних змінних обрані такі лінгвістичні терми: S – малий; M – середній; B – великий. Антецеденти правил БП сформовані наступними комбінаціями лінгвістичних термів вхідних змінних  $x_1$  і  $x_2$ : 1) S & (AND) S; 2) S & M; 3) S & B; 4) M & S; 5) M & M; 6) M & B; 7) B & S; 8) B & M; 9) B & B. У свою чергу, вектор консеквентів нечітких правил  $\mathbf{R}^k(n)$ , сформований в результаті проходження графа  $k$ -м агентом на ітерації з номером  $n$  (виділено жирним на рис. 3.8) має вигляд:

$$\mathbf{R}^k(n) = \{\mathbf{S}, \mathbf{M}, \mathbf{M}, \mathbf{B}, \mathbf{S}, \mathbf{M}, \mathbf{M}, \mathbf{B}, \mathbf{B}\}. \quad (3.27)$$

Імовірність  $P_{rj,(r+1)q}^k(n)$  переходу  $k$ -го агента з  $rj$ -го вузла шару з порядковим номером  $r$  у  $(r+1)q$ -й вузол наступного шару з порядковим номером  $r+1$  ( $r \in \{1, \dots, s\}; j, q \in \{1, \dots, v\}$ ) на ітерації  $n$  ( $n = 1, \dots, n_{\max}$ ) визначається виразом [48]

$$P_{rj,(r+1)q}^k(n) = \frac{\left[ \tau_{rj,(r+1)q}(n) \right]^\alpha \cdot \left[ \eta_{rj,(r+1)q} \right]^\beta}{\sum_{rj,(r+1)q=rj,(r+1)1}^{rj,(r+1)v} \left[ \tau_{rj,(r+1)q}(n) \right]^\alpha \cdot \left[ \eta_{rj,(r+1)q} \right]^\beta}, \quad (3.28)$$

де  $\tau_{rj,(r+1)q}(n)$  – інтенсивність феромону на ребрі між вузлами  $rj$  і  $(r+1)q$  на ітерації  $n$ ;  $\eta_{rj,(r+1)q}$  – величина, зворотна відносної відстані  $D_{rj,(r+1)q}$  між вузлами  $rj$  і  $(r+1)q$ ;  $\alpha$  – регульований параметр, який задає відносну значущість сліду феромону на ребрі при виборі наступного вузла;  $\beta$  – регульований параметр, який задає відносну значущість відстані  $D_{rj,(r+1)q}$  між вузлами  $rj$  і  $(r+1)q$ .

У свою чергу,

$$\eta_{rj,(r+1)q} = \frac{1}{D_{rj,(r+1)q}}; \quad (3.29)$$

$$D_{rj,(r+1)q} = \left| j_r - q_{(r+1)} \right| + 1, \quad (3.30)$$

де  $j_r$  – порядковий номер  $j$ -го вузла у шарі  $r$ ;  $q_{(r+1)}$  – порядковий номер  $q$ -го вузла у шарі  $r+1$ .

Відносна відстань  $D_{01,1q}$  між єдиним вузлом нульового шару і будь-яким  $q$ -м вузлом першого шару ( $q = 1, \dots, v$ ) дорівнює одиниці.

Після проходження кожним  $k$ -м агентом певних  $rj$ -х вузлів усіх  $s$  шарів графа БП на кожній ітерації з номером  $n$  для сформованого вектора консеквентів нечітких правил  $\mathbf{R}^k(n)$  розраховується довжина шляху  $k$ -го агента, яка визначається значенням цільової функції НС  $J^k(n)$ .

Кількість феромону, залишеного на кожному ребрі  $rj,(r+1)q$  графа  $k$ -м агентом на ітерації  $n$  визначається на основі значення цільової функції  $J^k(n)$  [65] наступним чином

$$\begin{cases} \Delta\tau_{rj,(r+1)q}^k(n) = \frac{Q}{J^k(n)}, \text{ при } rj,(r+1)q \in H^k(n); \\ \Delta\tau_{rj,(r+1)q}^k(n) = 0, \text{ при } rj,(r+1)q \notin H^k(n), \end{cases} \quad (3.31)$$

де  $Q$  – регульований параметр, який відноситься до кількості феромону, залишеного на шляху агентом;  $H^k(n)$  – набір ребер (шлях), який пройшов  $k$ -й агент на ітерації  $n$ . Параметр  $Q$  доцільно вибирати одного порядку з оптимальним значенням цільової функції НС  $J_{\text{opt}}$ . Значення феромону  $\Delta\tau_{rj,(r+1)q}^k(n)$ , залишеного  $k$ -м агентом на ребрі  $rj,(r+1)q$  на ітерації  $n$ , зворотно пропорційно залежить від значення цільової функції  $J^k(n)$ : мале значення цільової функції характеризується високою концентрацією феромону, велике – низькою.

Після розрахунку на основі залежності (3.31) значень феромону  $\Delta\tau_{rj,(r+1)q}^k(n)$ , відкладеного кожним  $k$ -м агентом на кожному ребрі  $rj,(r+1)q$  графа, отримані результати використовуються для збільшення феромону на ребрах графа [48] згідно з виразом

$$\tau_{rj,(r+1)q}(n) = \tau_{rj,(r+1)q}(n-1) + \rho \cdot \sum_{k=1}^z \Delta\tau_{rj,(r+1)q}^k(n), \quad (3.32)$$

де  $\rho$  – коефіцієнт кількості феромону, який залишають агенти на шляху (доцільно вибирати в діапазоні від 0 до 1);  $z$  – загальна кількість агентів у популяції.

Для використання елітної стратегії мурашиних алгоритмів у цьому методі вводяться так звані елітні агенти, що дозволяє суттєво збільшити швидкість збіжності методу [48, 65]. При цьому на кожній ітерації застосовується додаткове збільшення феромону для ребер, які входять у кращий на цій ітерації шлях із найменшим значенням цільової функції  $J_{\text{min}}(n)$ . Додаткова кількість феромону, залишеного на кожному ребрі, що входить у кращий шлях на ітерації  $n$ , визначається на основі значення його цільової функції  $J_{\text{min}}(n)$  та кількості елітних агентів у популяції  $e$  наступним чином

$$\Delta\tau_e(n) = \frac{e \cdot Q}{J_{\min}(n)}. \quad (3.33)$$

У кінці кожної ітерації  $n$  для кожного ребра  $rj,(r+1)q$  графа застосовується операція випаровування феромона [65] відповідно до виразу

$$\tau_{rj,(r+1)qF}(n) = \tau_{rj,(r+1)q}(n) \cdot (1 - \rho), \quad (3.34)$$

де  $\tau_{rj,(r+1)qF}(n)$  – фінальне значення кількості феромону на ребрі  $rj,(r+1)q$  графа в кінці ітерації  $n$  після застосування операції випаровування. При переході з ітерації  $n$  на ітерацію  $n+1$  ( $n = 1, \dots, n_{\max}$ ) на кожному ребрі  $rj,(r+1)q$  ( $r \in \{1, \dots, s\}; j, q \in \{1, \dots, v\}$ ) графа здійснюється оновлення феромону згідно з виразом

$$\tau_{rj,(r+1)q}(n+1) = \tau_{rj,(r+1)qF}(n). \quad (3.35)$$

Запропонований у [152] метод генерації баз правил нечітких систем типу Мамдані із формуванням оптимальних консеквентів на основі мурашиних алгоритмів складається з наступних послідовних кроків.

*Крок 1.* Ініціалізація методу. На даному етапі здійснюється побудова графа БП нечіткої системи на основі попереднього вибору кількості лінгвістичних термів  $w_i$  для кожної  $i$ -ї вхідної змінної вектора  $\mathbf{X}$  ( $i = 1, \dots, m$ ) і лінгвістичних термів  $v$  для вихідної змінної у НС. Кількість основних шарів графа  $s$  розраховується відповідно до залежності (3.20) на основі параметрів  $m$  і  $w_i$ , кількість вузлів  $v$  у кожному  $r$ -му шарі вибирається рівною кількості всіх можливих консеквентів правил  $LT^j$  ( $j = 1, \dots, v$ ) і, відповідно, кількості всіх ЛТ вихідної змінної системи. Також, на цьому кроці вибирається цільова функція  $J$  НС та її оптимальне значення  $J_{\text{opt}}$ .

*Крок 2.* Ініціалізація параметрів мурашиного алгоритму. На даному етапі створюється популяція агентів і встановлюються основні параметри алгоритму: кількість агентів популяції  $z$ , регульовані параметри  $\alpha$ ,  $\beta$ ,  $Q$  і  $\rho$ , максимальна кількість ітерацій  $n_{\max}$ , а також кількість елітних агентів у

популяції  $e$ . Крім того, на цьому кроці на ребрах графа встановлюється невелике початкове додатне значення феромону до початку руху агентів.

*Крок 3.* Рух агентів по вузлах графа бази правил НС. Рух кожного  $k$ -го агента створеної на кроці 2 популяції починається з вузла нульового шару і здійснюється послідовно через всі  $s$  шарів графу від першого до останнього тільки у напрямку зростання порядкового номера шарів  $r$ . Перехід  $k$ -го агента з будь-якого  $rj$ -го вузла шару з порядковим номером  $r$  можливий тільки у будь-який  $(r+1)q$ -й вузол наступного шару з порядковим номером  $r+1$ . Кількість переходів кожного  $k$ -го агента по вузлах графа на одній ітерації дорівнює загальній кількості шарів графа  $s$ .

Якщо  $k$ -й агент знаходиться у вузлі  $rj$  і  $r < s$ , то для визначення наступного ребра шляху  $rj, (r+1)q$  розраховується ймовірність  $P_{rj, (r+1)q}^k(n)$  переходу даного агента у вузол  $(r+1)q$  згідно з виразом (3.28). У свою чергу, відносна відстань  $D_{rj, (r+1)q}$  між вузлами  $rj$  і  $(r+1)q$ , а також зворотна їй величина  $\eta_{rj, (r+1)q}$  розраховуються на основі залежностей (3.30) і (3.29), відповідно. Крок 3 продовжується, поки кожен  $k$ -й агент популяції не пройде всі  $s$  шарів графа.

*Крок 4.* Обчислення значень цільової функції НС. На даному етапі для кожного вектора консеквентів нечітких правил  $\mathbf{R}^k(n)$ , сформованого кожним  $k$ -м агентом популяції ( $k = 1, \dots, z$ ) під час його руху на кроці 3, виконується розрахунок значення цільової функції  $J^k(n)$  НС, обраної на кроці 1.

*Крок 5.* Перевірка на закінчення роботи методу. На даному етапі здійснюється вибір найкращого вектора консеквентів нечітких правил  $\mathbf{R}_{best}(n)$  БП, сформованого під час руху агентів на кроці 3, за найменшим значенням цільової функції НС  $J_{min}(n)$  із значень, розрахованих на кроці 4. Для обраного найкращого вектора консеквентів нечітких правил  $\mathbf{R}_{best}(n)$  БП здійснюється перевірка на закінчення роботи методу, що може визначатися досягненням оптимального значення цільової функції НС

( $J_{\min}(n) \leq J_{\text{opt}}$ ), або виконанням максимальної кількості ітерацій  $n_{\max}$ . Закінчення роботи методу також може вважатися досягнутим, якщо протягом певної кількості ітерацій значення цільової функції  $J$  не зменшувалося. Якщо ця перевірка дала позитивний результат, здійснюється перехід на крок 9. У протилежному випадку виконується перехід на крок 6.

*Крок 6.* Збільшення феромону на ребрах графа бази правил НС. На даному етапі на основі значень цільової функції  $J^k(n)$ , розрахованих на кроці 4, обчислюється кількість феромону, залишеного на кожному ребрі  $rj,(r+1)q$  графа кожним  $k$ -м агентом ( $k = 1, \dots, z$ ) під час руху на кроці 3 на основі (3.31). Розраховані значення кількості феромону  $\Delta\tau_{rj,(r+1)q}^k$ , відкладеного кожним  $k$ -м агентом на кожному ребрі  $rj,(r+1)q$  графа під час руху, використовуються для збільшення феромону на ребрах графа згідно з виразом (3.32). Також, згідно з елітною стратегією на цьому кроці застосовується додаткове збільшення феромону для ребер, які входять до кращого на даній ітерації шляху з найменшим значенням цільової функції  $J_{\min}(n)$ . Це значення феромону розраховується на основі (3.33).

*Крок 7.* Випаровування феромону на ребрах графа БП НС. На даному етапі для кожного ребра  $rj,(r+1)q$  графа застосовується операція випаровування феромону згідно з виразом (3.34), та обчислюється його фінальне значення  $\tau_{rj,(r+1)qF}$ .

*Крок 8.* Перехід на наступну ітерацію методу. На даному етапі здійснюється повернення всіх агентів популяції у вузол нульового шару графа, а також оновлення феромону на кожному ребрі  $rj,(r+1)q$  графа згідно з виразом (3.35). Після цього здійснюється перехід на крок 3.

*Крок 9.* Завершення реалізації методу. Після цього може здійснюватися параметрична оптимізація НС та її програмно-апаратна реалізація для подальшого застосування.



Далі розглянемо приклад застосування наведеного мультиагентного методу для створення бази правил нечіткої системи керування реальним складним технічним об'єктом. Зокрема, у роботі [152] представлено розробку БП нечіткої САК температурою реактора спеціалізованого піролізного комплексу (СПК) на основі розглянутого мультиагентного методу з дослідженням його ефективності.

*Синтез бази правил нечіткого регулятора типу Мамдані для САК температурою реактора спеціалізованого піролізного комплексу на основі мультиагентного методу.* СПК ефективно застосовуються для утилізації полімерних відходів з метою отримання альтернативного рідкого та газоподібного палива [153]. У свою чергу, дані СПК є складними технічними об'єктами, для автоматизації яких успішно застосовуються інтелектуальні керуючі пристрої на основі нечіткої логіки [152, 153]. Однією з найбільш важливих задач автоматизації СПК є стабілізація та автоматичне керування температурними режимами їх піролізних реакторів [153, 154]. Функціональна структура САК температурними режимами реактора СПК наведена на рис. 3.9 [153], де прийняті такі позначення: ВРК – верхній рівень керування; ЗП – задавальний пристрій; НРТ – нечіткий регулятор температури; СНР – система нагріву реактора СПК; ЗГ – зріджений газ; П – повітря; ЛРВГ, ЛРВП – лінійні регулятори витрати ЗГ та П, відповідно; ГП – газовий пальник СНР реактора; ДТ – датчик температури;  $T_3$ ,  $T_p$  – задане та реальне значення температури нагріву реактора СПК;  $u_{ЗП}$ ,  $u_{НРТ}$ ,  $u_{ДТ}$  – вихідні сигнали ЗП, НРТ і ДТ, відповідно;  $\varepsilon_T$  – помилка керування температурою реактора;  $Q_{ЗГ}$ ,  $Q_{П}$  – значення витрат ЗГ та П, відповідно;  $P_{ГП}$  – значення потужності нагріву ГП СНР;  $F_3$  – збурення, що діють на реактор СПК.

У роботі [152] для дослідження ефективності метода генерації БП проводилась розробка НРТ типу Мамдані для САК температурою реактора, робочий об'єм якого становить 100 літрів, а максимальна потужність його системи нагріву  $P_{ГП\max} = 25$  кВт.

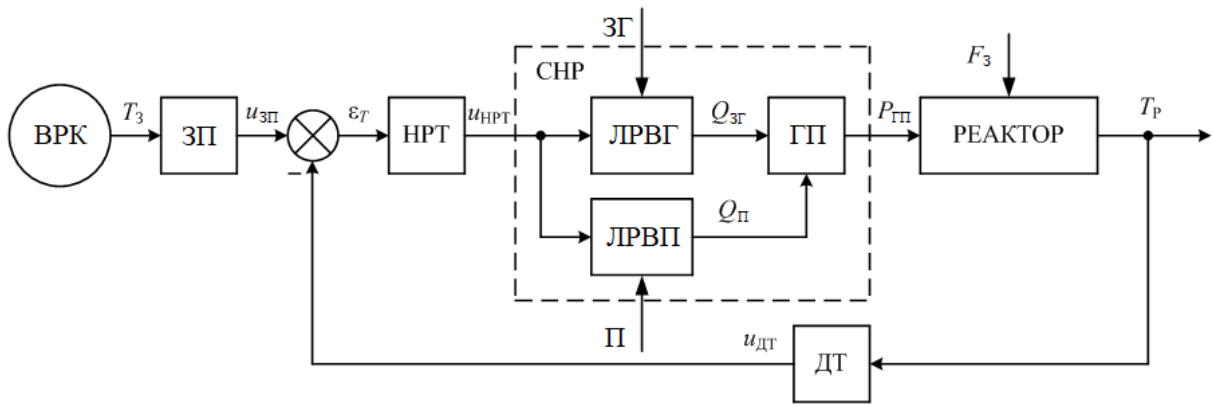


Рис. 3.9. Структура нечіткої САК температурою нагріву реактора СПК

Математичні моделі основних елементів даної САК температурою розроблені та наведені у роботі [155]. Зокрема, математична модель піролізного реактора як об'єкта керування температурою може бути представлена у вигляді передаточної функції  $W_P(p)$  за каналом керування

$$W_P(p) = \frac{T_P(p)}{P_{ГП}(p)} = \frac{\tilde{K}_P e^{-\tau_p p}}{(T_{\sim p1} p + 1)(T_{\sim p2} p + 1)^2}, \quad (3.36)$$

де  $T_P(p)$  – зображення температури нагріву реактора СПК  $T_P(t)$ ;  $P_{ГП}(p)$  – зображення потужності нагріву газового пальника СНР  $P_{ГП}(t)$ ;  $\tilde{K}_P$ ,  $\tau_p$ ,  $T_{\sim p1}$ ,  $T_{\sim p2}$  – коефіцієнт підсилення та сталі часу передаточної функції реактора СПК, відповідно.

Крім того, параметри даної передаточної функції  $\tilde{K}_P$ ,  $T_{\sim p1}$ ,  $T_{\sim p2}$  нелінійно залежать від потужності нагріву  $P_{ГП}$  та рівня завантаження реактора  $L_P$ :

$$\begin{cases} \tilde{K}_P = f(P_{ГП}, L_P); \\ T_{\sim p1} = f(P_{ГП}, L_P); \\ T_{\sim p2} = f(P_{ГП}, L_P). \end{cases} \quad (3.37)$$

Нелінійні залежності (3.37) отримані на основі експериментальних даних та реалізовані за допомогою спеціалізованої нечітко-логічної моделі у роботі [155].

У свою чергу, НРТ даної САК температурними режимами реалізує закон керування на основі нелінійної залежності [152]

$$u_{\text{НРТ}} = k_U f_{\text{НРТ}} \left( k_P \varepsilon_T, k_D \frac{d\varepsilon_T}{dt}, k_I \int \varepsilon_T dt \right), \quad (3.38)$$

де  $k_P, k_D, k_I, k_U$  – нормуючі коефіцієнти НРТ.

Таким чином, вектор вхідних змінних  $\mathbf{X}$  НРТ має вигляд

$$\mathbf{X} = \{x_1, x_2, x_3\}, m = 3, \quad (3.39)$$

де  $x_1 = k_P \varepsilon_T$ ;  $x_2 = k_D \frac{d\varepsilon_T}{dt}$ ;  $x_3 = k_I \int \varepsilon_T dt$ .

В якості вихідної змінної  $u$  у даному випадку виступає вихідний сигнал НРТ  $u_{\text{НРТ}}$ .

Для кожної вхідної змінної НРТ обрано по 5 лінгвістичних термів ( $w_1 = 5$ ;  $w_2 = 5$ ;  $w_3 = 5$ ) трикутного типу: BN – великий від’ємний; SN – малий від’ємний; Z – нульовий; SP – малий додатний; BP – великий додатний. У свою чергу, для вихідної змінної  $u_{\text{НРТ}}$  обрано 7 ЛТ ( $v = 7$ ) трикутного типу: Z – нульовий; S – малий; LM – нижче середнього; M – середній; NM – вище середнього; B – великий; VB – дуже великий. Зовнішній вигляд обраних термів із встановленими параметрами наведено на рис. 3.10.

Таким чином, загальна кількість правил БП  $s$  відповідно до виразу (3.20) дорівнює 125,  $s = 5 \cdot 5 \cdot 5 = 125$ . Кожне  $r$ -те правило даної БП ( $r = 1, 2, \dots, 125$ ) є лінгвістичним твердженням виду

$$\begin{aligned} &\text{IF } "k_P \varepsilon_T = a" \text{ AND } "k_D \frac{d\varepsilon_T}{dt} = b" \\ &\text{AND } "k_I \int \varepsilon_T dt = c" \text{ THEN } "u_{\text{НРТ}} = h". \end{aligned} \quad (3.40)$$

Консеквенти кожного  $r$ -го правила БП у даному випадку вибираються з набору можливих консеквентів правил, що складається з 7 ЛТ (Z; S; LM; M; NM; B; VB) вихідної змінної НРТ  $u_{\text{НРТ}}$ . Вектор консеквентів (3.25)  $\mathbf{R}_\gamma$  БП даного НРТ має вигляд

$$\mathbf{R}_\gamma = \{LT_{\gamma 1}, LT_{\gamma 2}, \dots, LT_{\gamma r}, \dots, LT_{\gamma 125}\},$$

$$LT_{\gamma r} \in \{LT^1, LT^2, \dots, LT^7\}, \gamma \in \{1, 2, \dots, 7^{125}\}.$$
(3.41)

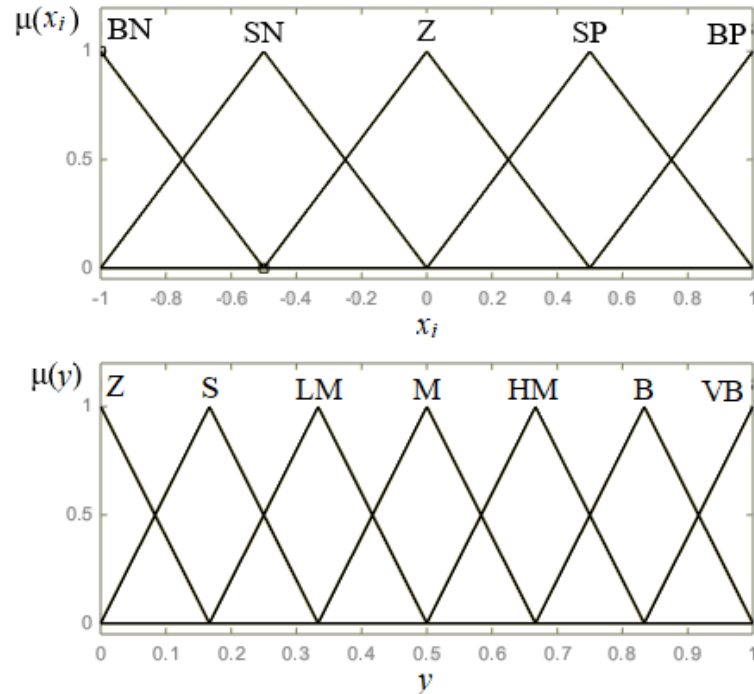


Рис. 3.10. Лінгвістичні терми (з встановленими параметрами) для вхідних  $x_i, (i \in \{1, 2, 3\})$  та вихідного у сигналів НРТ

В якості методу дефазифікації у даному НРТ обрано метод центру тяжіння [8].

У роботі [152] проведено розробку БП НРТ реактора СПК на основі знань експертів та за допомогою розглянутого методу генерації баз правил з формуванням оптимальних консеквентів на основі мурашиних алгоритмів. Фрагмент розробленої на базі знань експертів БП НРТ представлений у табл. 3.2.

При генерації БП НРТ за допомогою розглянутого вище мультиагентного методу на етапі його ініціалізації (крок 1) побудовано граф БП з кількістю основних шарів  $s = 125$ . У свою чергу, кількість вузлів у кожному  $r$ -му шарі  $v$  дорівнює 7.

Таблиця 3.2. Фрагмент бази правил НРТ реактора, синтезованої на основі знань експертів

№ правила	Лінгвістичні терми вхідних та вихідної змінних			
	$x_1$	$x_2$	$x_3$	$y$
1	BN	BN	BN	Z
4	BN	BN	SP	S
34	SN	SN	SP	S
44	SN	SP	SP	M
59	Z	SN	SP	M
64	Z	Z	SP	M
90	SP	Z	BP	HM
94	SP	SP	SP	HM
113	BP	Z	Z	B
125	BP	BP	BP	VB

В якості цільової функції  $J$  у даному випадку обрано узагальнене інтегральне відхилення реальної перехідної характеристики САК температурою  $T_P(t, \mathbf{R})$  від бажаної перехідної характеристики її еталонної моделі ЕМ  $T_B(t)$  [151], яка представлена передаточною функцією  $W_{EM}(p)$

$$W_{EM}(p) = \frac{T_B(p)}{T_3(p)} = \frac{1}{(T_{EM}p + 1)^3}, \quad (3.42)$$

де  $T_{EM}$  – стала часу ЕМ.

Дана цільова функція розраховується відповідно до залежності

$$J(t, \mathbf{R}) = \frac{1}{t_{\max}} \int_0^{t_{\max}} \left[ (E_T)^2 + k_1 \left( \frac{dE_T}{dt} \right)^2 + k_2 \left( \frac{d^2E_T}{dt^2} \right)^2 \right] dt, \quad (3.43)$$

де  $t_{\max}$  – загальний час перехідного процесу САК температурними режимами реактора;  $k_1, k_2$  – вагові коефіцієнти;  $E_T$  – відхилення  $T_P(t, \mathbf{R})$  від  $T_B(t)$ ,

$$E_T = T_B(t) - T_P(t, \mathbf{R}). \quad (3.44)$$

У свою чергу, оптимальним значенням цільової функції прийнято  $J_{\text{opt}} = 6,5 \cdot 10^4$ , при якому відхилення реальної перехідної характеристики  $T_P(t, \mathbf{R})$  від бажаної  $T_B(t)$  лежить у допустимих межах.

Для дослідження ефективності розглянутого мультиагентного методу на етапі ініціалізації параметрів мурашиного алгоритму (крок 2) було створено 16 окремих популяцій агентів із різними параметрами  $z$  і  $e$ : 1)  $z = 10, e = 0$ ; 2)  $z = 10, e = 5$ ; 3)  $z = 10, e = 10$ ; 4)  $z = 10, e = 15$ ; 5)  $z = 20, e = 0$ ; 6)  $z = 20, e = 5$ ; 7)  $z = 20, e = 10$ ; 8)  $z = 20, e = 15$ ; 9)  $z = 30, e = 0$ ; 10)  $z = 30, e = 5$ ; 11)  $z = 30, e = 10$ ; 12)  $z = 30, e = 15$ ; 13)  $z = 40, e = 0$ ; 14)  $z = 40, e = 5$ ; 15)  $z = 40, e = 10$ ; 16)  $z = 40, e = 15$ . У свою чергу, генерація консеквентів бази правил НРТ проводилася по черзі за допомогою кожної окремої популяції. Також, дані експерименти для кожної окремої популяції проводилися по 5 разів з подальшим вибором найкращих результатів. Крім того, для кожної зі створених популяцій використовувалися однакові значення регульованих параметрів методу  $\alpha, \beta, Q, \rho$ , які склали:  $\alpha = 2$ ;  $\beta = 1$ ;  $Q = 7 \cdot 10^4$ ;  $\rho = 0,5$ .

При розрахунку значень цільової функції (3.43) для кожного  $k$ -го агента на кроці 4 кожної  $n$ -ї ітерації методу моделювання перехідних процесів САК температурою реактора проводилося у всіх можливих режимах роботи (при дії різних вхідних та збурювальних впливів) для ефективної генерації консеквентів усіх правил БП. У свою чергу, перевірка на закінчення роботи методу на кроці 5 здійснювалася за умовою виконання максимальної кількості ітерацій  $n_{\text{max}} = 200$ . Збільшення феромону на ребрах графа бази правил НРТ на кроці 6 кожної ітерації методу проводилося на основі залежностей (3.31)-(3.33). Крім того, процедури випаровування та оновлення феромону на кроках 7 та 8 здійснювалися відповідно до виразів (3.34) та (3.35), відповідно.

Найкращі результати експериментів, отримані при генерації БП НРТ за допомогою кожної окремої популяції, зведені у табл. 3.3, де прийняті

такі позначення:  $g_{Jopt}$  – загальна кількість обчислень цільової функції  $J$ , необхідних для досягнення її оптимального значення;  $n_{Jopt}$  – загальна кількість ітерацій методу, необхідних для досягнення оптимального значення цільової функції  $J$ .

Таблиця 3.3. Найкращі результати експериментів, отримані під час генерації БП НРТ

№ популяції	Параметри експериментів				
	$z$	$e$	$J_{min}$	$n_{Jopt}$	$g_{Jopt}$
1	10	0	$6,305 \cdot 10^4$	292	2920
2	10	5	$6,402 \cdot 10^4$	223	2230
3	10	10	$6,378 \cdot 10^4$	211	2110
4	10	15	$6,486 \cdot 10^4$	219	2190
5	20	0	$6,432 \cdot 10^4$	142	2840
6	20	5	$6,322 \cdot 10^4$	121	2420
7	20	10	$6,374 \cdot 10^4$	103	2060
8	20	15	$6,490 \cdot 10^4$	117	2340
9	30	0	$6,482 \cdot 10^4$	81	2430
10	30	5	$6,425 \cdot 10^4$	74	2220
11	30	10	$6,344 \cdot 10^4$	62	1860
12	30	15	$6,412 \cdot 10^4$	79	2370
13	40	0	$6,309 \cdot 10^4$	68	2720
14	40	5	$6,382 \cdot 10^4$	59	2360
15	40	10	$6,459 \cdot 10^4$	53	2120
16	40	15	$6,498 \cdot 10^4$	58	2320

У свою чергу,  $g_{Jopt}$  є основним параметром, що характеризує обчислювальні та часові витрати методу, і визначається наступним чином

$$g_{Jopt} = zn_{Jopt}. \quad (3.45)$$

На рис. 3.11 наведено сімейства кривих зміни найкращих значень цільової функції (3.43) у процесі генерації консеквентів бази правил НРТ на основі розглянутого методу за допомогою різних популяцій.

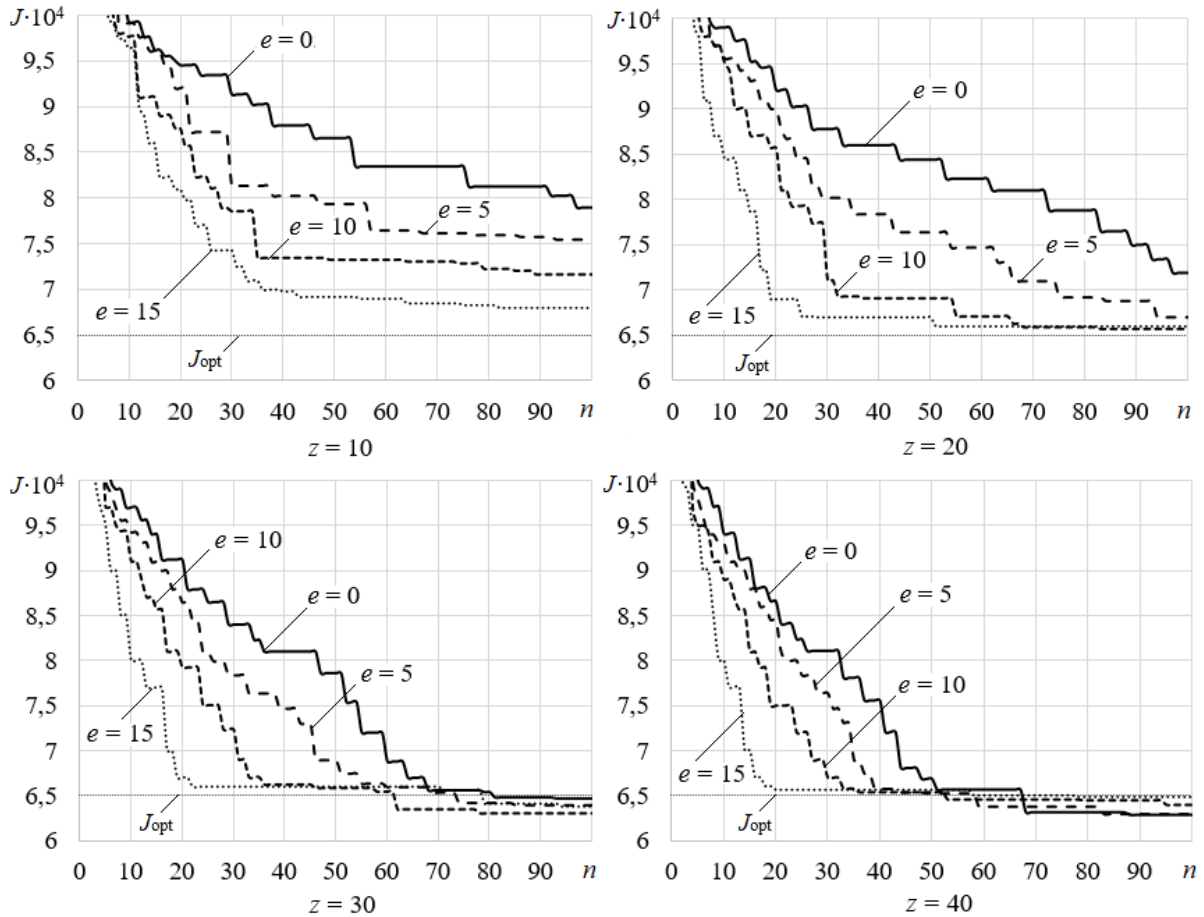


Рис. 3.11. Сімейства кривих зміни найкращих значень цільової функції (3.43) у процесі генерації консеквентів БП НРТ (у діапазоні  $n = 0 \dots 100$ )

Як видно з табл. 3.3 та рис. 3.11, чим більше кількість агентів  $z$  популяції, тим менше ітерацій необхідно для знаходження оптимального вектора консеквентів БП  $\mathbf{R}_{\text{opt}}$ . Так, наприклад, для 14-ї ( $z = 40$ ;  $e = 5$ ) популяції цільова функція  $J$  досягла оптимального значення  $J_{\text{opt}}$  на 164 ітерації раніше, ніж для 2-ї ( $z = 10$ ;  $e = 5$ ).

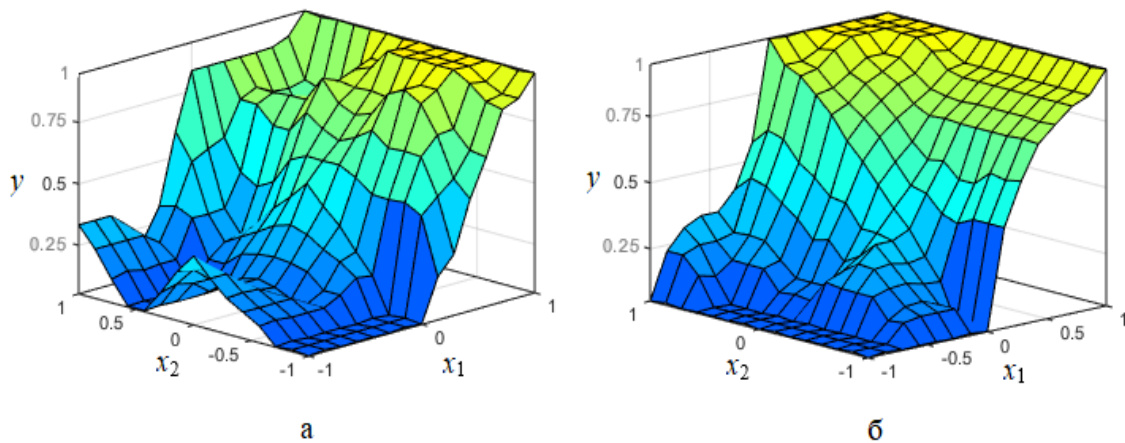
Однак, при цьому загальна кількість обчислень  $g_{J_{\text{opt}}}$  значення цільової функції  $J$  для 14-ї популяції трохи більше, ніж для 2-ї. Також, збільшення кількості елітних агентів  $e$  істотно збільшує швидкість





Таблиця 3.4. Фрагмент згенерованої БП НРТ за допомогою 11-ї популяції

№ правила	Лінгвістичні терми вхідних та вихідної змінних			
	$x_1$	$x_2$	$x_3$	$y$
1	BN	BN	BN	Z
4	BN	BN	SP	Z
34	SN	SN	SP	Z
44	SN	SP	SP	HM
59	Z	SN	SP	LM
64	Z	Z	SP	M
90	SP	Z	BP	B
94	SP	SP	SP	B
113	BP	Z	Z	VB
125	BP	BP	BP	VB

Рис. 3.12. Характеристичні поверхні НРТ  $y = f(x_1, x_2)$  при  $k_1 \int \varepsilon_T dt = 0,5$ 

На рис. 3.13 наведено графіки перехідних процесів зміни: (а) заданого значення температури реактора  $T_3(t)$ , (б) збурення  $F_3(t)$ , що діє на виході системи, (в) реального значення температури реактора  $T_P(t)$  для САК з розробленими БП НРТ: 1 – на основі знань експертів; 2 – на основі

розглянутого мультиагентного методу генерації з використанням 11-ї популяції.

Таблиця 3.5. Аналіз показників якості САК температурою з розробленими БП НРТ

Показники якості	Показники якості САК температурою	
	БП на основі знань експертів	Згенерована БП на основі мультиагентного методу
Перерегулювання $\sigma_{\max}, \%$	36,2	19,8
Швидкодія $t_{\text{pmax}}, \text{c}$	932	578
Коливальність $\mu$	3	1
Значення цільової функції $J$	$6,98 \cdot 10^4$	$6,344 \cdot 10^4$

З табл. 3.5 та рис. 3.13 видно, що САК температурою з розробленою БП НРТ на основі розглянутого мультиагентного методу генерації має суттєво менше значення цільової функції  $J$  і вищі показники якості керування, ніж САК з БП на основі знань експертів. Крім того, для знаходження оптимального вектора консеквентів БП  $\mathbf{R}_{\text{opt}}$  за допомогою даного методу не знадобилося суттєвих обчислювальних та часових затрат ( $g_{\text{Jopt}} = 1860$ ), що підтверджує його високу ефективність.

Розглянутий мультиагентний метод доцільно застосовувати для генерації баз правил з оптимальними консеквентами для нечітких систем типу Мамдані у таких випадках: (а) при недостатньому обсязі вихідної інформації (в умовах високого ступеня невизначеності інформації); (б) при досить великій кількості правил, для якої складання БП НС на основі знань експертів не завжди є ефективним; (в) при різному рівні кваліфікації експертів.

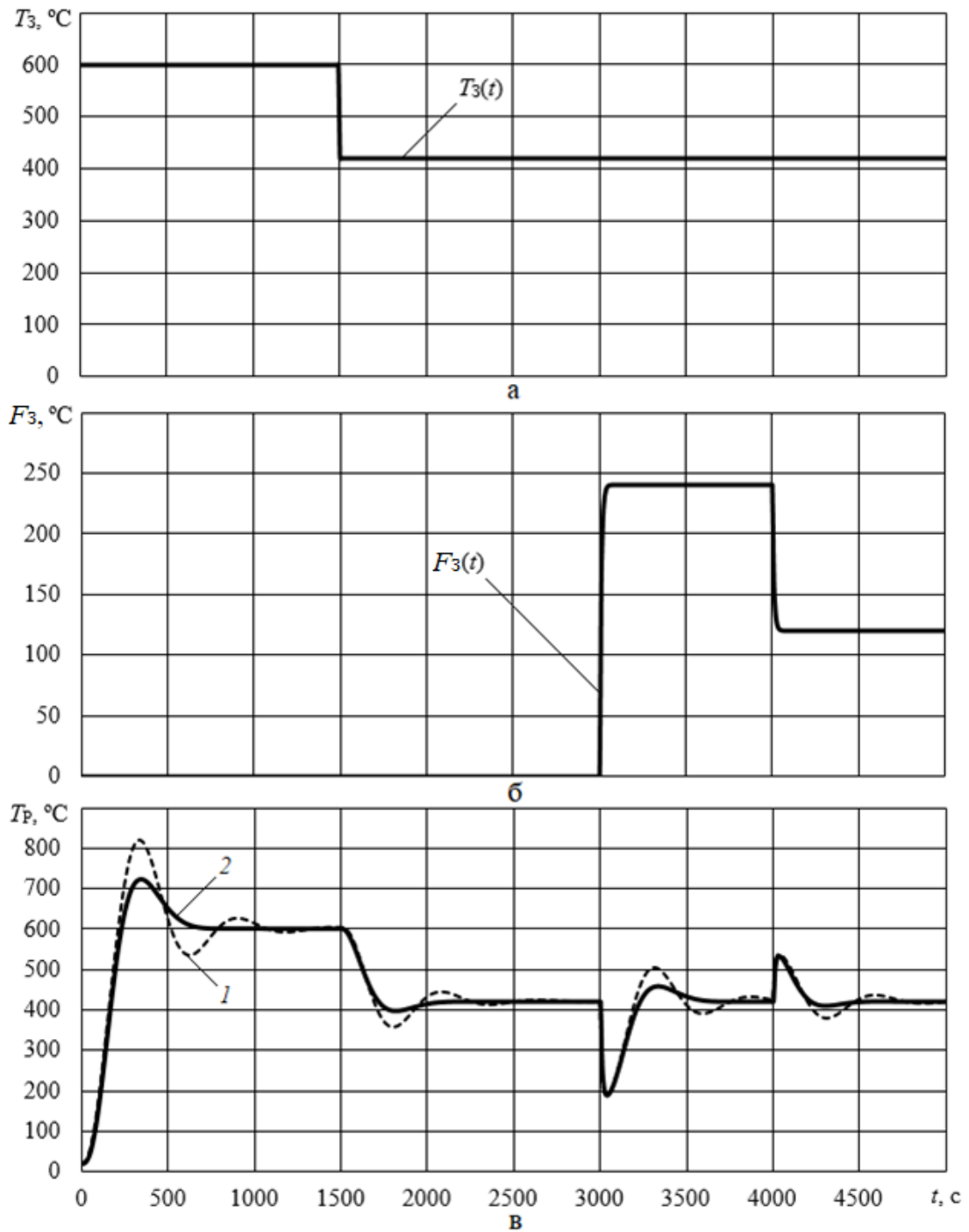


Рис. 3.13. Графіки перехідних процесів САК температурою реактора СПК

У свою чергу, при необхідності подальшого підвищення показників якості даної САК та спрощення її реалізації після генерації консеквентів бази правил за допомогою розглянутого методу можна здійснювати додаткову структурно-параметричну оптимізацію НРТ, зокрема, редукцію кількості правил БП [149], оптимізацію параметрів функцій належності ЛТ [148], нормуючих коефіцієнтів [153] тощо.

### 3.3 Контрольні питання

1. Назвіть основні переваги синтезу нейронних мереж за допомогою еволюційних методів у порівнянні з іншими підходами.
2. Сформулюйте основні принципи нейроеволюції.
3. Назвіть основні етапи синтезу нейронних мереж при застосуванні нейроеволюційного підходу.
4. Яким чином будується початкова архітектура нейронної мережі при застосуванні нейроеволюційного підходу?
5. Які еволюційні методи дозволяють оптимізувати гіперпараметри нейронних мереж?
6. Як визначається фітнес-функція для нейронних мереж при застосуванні нейроеволюційного підходу?
7. Які параметри нейронних мереж кодуються у хромосоми при застосуванні нейроеволюційного підходу?
8. Які перспективи застосування нейроеволюції при вирішенні прикладних задач?
9. Як вибрати критерії оцінювання ефективності нейроеволюційних методів при вирішенні прикладних задач?
10. У чому полягають особливості керування багатозв'язними складними технічними об'єктами?
11. Наведіть узагальнену функціональну структуру багатозв'язної системи автоматичного керування.
12. Наведіть основні кроки нейроеволюційного методу керування багатозв'язними складними технічними об'єктами.
13. Наведіть узагальнену структуру багатозв'язної САК на основі єдиного комплексного нейроконтролера.
14. Яким чином будуються еталонні моделі при застосуванні нейроеволюційного методу керування багатозв'язними складними технічними об'єктами?

15. Яким чином обирається комплексна цільова функція при застосуванні нейроеволюційного методу керування багатозв'язними складними об'єктами?

16. Наведіть залежність фітнес-функції від комплексної цільової функції, що застосовується при реалізації нейроеволюційного методу керування багатозв'язними складними об'єктами.

17. Яким чином відбувається збільшення кількості прихованих шарів та кількості нейронів у прихованих шарах при синтезі нейроконтролера для багатозв'язного складного об'єкта?

18. Охарактеризуйте особливості керування просторовим рухом багатоцільового мобільного робота, що здатен переміщуватися по похилих та вертикальних феромагнітних поверхнях.

19. Наведіть структуру багатозв'язної САК просторовим рухом багатоцільового мобільного робота.

20. Наведіть математичну модель багатоцільового мобільного робота як об'єкта керування просторовим рухом.

21. Наведіть еталонні моделі, що застосовуються для синтезу комплексного нейроконтролера просторового руху багатоцільового мобільного робота.

22. Назвіть основні переваги синтезу нечітких систем за допомогою еволюційних та мультиагентних методів у порівнянні з іншими підходами.

23. Назвіть основні задачі структурної та параметричної оптимізації нечітко-логічних систем.

24. Які параметри нечітких систем доцільно оптимізувати за допомогою еволюційних та мультиагентних методів?

25. Наведіть постановку задачі синтезу бази правил нечіткої системи типу Мамдані.

26. У чому полягає основна ідея застосування мурашиного алгоритму для генерації баз правил нечітких систем?

27. Наведіть основні кроки мультиагентного методу генерації баз правил нечітких систем.
28. Яким чином визначається цільова функція нечіткої системи при застосуванні мультиагентного методу генерації баз правил?
29. Яким чином структура бази правил нечіткої системи може бути представлена у вигляді спеціального графа?
30. Скільки шарів повинен мати граф бази правил нечіткої системи з дванадцятьма правилами у базі?
31. Яким чином відбувається рух агентів по вузлах графа бази правил нечіткої системи?
32. Охарактеризуйте особливості керування температурними режимами реактора спеціалізованого піролізного комплексу, що застосовується для утилізації полімерних відходів.
33. Наведіть структуру нечіткої САК температурними режимами реактора спеціалізованого піролізного комплексу.
34. Наведіть математичну модель реактора спеціалізованого піролізного комплексу як об'єкта керування температурою.
35. Яким чином впливає кількість елітних агентів у популяції на швидкість знаходження оптимального значення цільової функції нечіткої системи?

## ЛІТЕРАТУРА

1. Стратегія розвитку штучного інтелекту в Україні: монографія / А.І. Шевченко, С.В. Барановський, О.В. Білокобильський, Є.В. Бодянський, А.Я. Бомба, А.С. Довбиш, Т.В. Єрошенко, А.С. Жохін, В.В. Казимир, М.С. Клименко, С.В. Ковалевський, О.В. Козлов, Ю.П. Кондратенко, та ін. [За заг. ред. А.І. Шевченка]. Київ: ІПШІ, 2023. 305 с.
2. Zadeh, L.A. The role of fuzzy logic in modeling, identification and control / L.A. Zadeh // Modeling Identification and Control, 15(3). – 1994. – P. 191–203.
3. Recent developments and new directions in soft computing. L.A. Zadeh, A.M. Abbasov, R.R. Yager, S.N. Shahbazova, M.Z. Reformat, Eds. STUDFUZ 317, Cham: Springer, 2014. 466 p.
4. Kosko, B. Fuzzy Systems as Universal Approximators / B. Kosko // IEEE Trans. on Computers, Vol. 43, №11, 1994. – P. 1329-1333.
5. Mendel, J. M. Uncertain Rule-Based Fuzzy Systems, Introduction and New Directions. Second Edition, Springer International Publishing, 2017, 684 p.
6. Advance trends in soft computing. M. Jamshidi, V. Kreinovich, J. Kacprzyk, Eds. Cham: Springer-Verlag, 2013, 468 p.
7. Hampel, R. Fuzzy Control: Theory and Practice / R. Hampel, M. Wagenknecht, N. Chaker. – New York: Physika-Verlag, Heidelberg, 2000. – 410 p.
8. Piegat, A. Fuzzy Modeling and Control. Physica, 2013, Vol. 69.
9. Ямпольський Л.С. Системи штучного інтелекту в плануванні, моделюванні та управлінні / Л.С. Ямпольський, Б.П. Ткач, О.І. Лісовиченко. – Київ: ДП «Видавничий дім «Персонал», 2011. – 544 с.
10. Zimmermann, H. J. Fuzzy Set Theory and Its Applications // Springer Netherlands, Springer Science+Business Media New York, 514 p., 2001.
11. Mahmoud, M.S. Fuzzy Control, Estimation and Diagnosis. Single and Interconnected Systems. Springer International Publishing, 2018, 689 p.



12. Driankov, D., Hellendoorn, H., Reinfrank, M. An introduction to fuzzy control. Springer Science & Business Media, 2013.
13. Kondratenko Y.P. Encheva, S.B., Sidenko, I.V. Synthesis of Modular-Structured Systems for Automatization of Decision-Making Processes in Transport Logistics. *International Journal of Computing*. 2012. Vol. 11. Issue 4. P. 336–344.
14. Kondratenko Y.P., Sidenko, I.V. Method of Actual Correction of the Knowledge Database of Fuzzy Decision Support System with Flexible Hierarchical Structure. In book: V.Krasnoprosin, A.M. Gil Lafuente, C. Zopounidis (Eds.), *Computational Techniques in Modeling and Simulation, Series: Studies in Financial Optimization and Risk Management*, Nova Science Publishers, New York. 2013. P. 55-74.
15. Mamdani, E.H. Application of fuzzy algorithms for control of simple dynamic plant. *Proceedings of IEEE*, vol. 121. 1974. P. 1585-1588.
16. Mamdani, E.H., Assilian, S. An experiment in linguistic synthesis with a fuzzy logic controller. *International journal of man-machine studies*, No. 7(1). 1975. P. 1-13.
17. Takagi T., Sugeno M. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*. 1985. SMC–15, N 1. P. 116-132.
18. Karnik, N.N., Mendel, J.M., Liang, Q. Type-2 fuzzy logic systems. *IEEE Transactions on Fuzzy Systems*, Vol. 7 (5), 1999.
19. Karnik, N.N., Mendel, J.M., Liang, Q. Applications of type-2 fuzzy logic systems to forecasting of time series. *Inform. Sci.*, Vol. 120, 1999, pp. 89-111.
20. Liang, Q., Mendel, J.M. Interval type-2 fuzzy logic system theory and design. *IEEE Transactions on Fuzzy Systems*, Vol. 8, 2000, pp. 535-550.
21. Mendel, J.M., John, R.I. Interval Type-2 Fuzzy Logic Systems Made Simple. *IEEE Transactions on Fuzzy Systems*, Vol. 14, N0. 6, 2006.

22. Castillo, O., Melin, P. Type-2 Fuzzy Logic Theory and Applications. Springer-Verlag, Berlin, 2008.
23. Karnik, N.N., Mendel, J.M. Centroid of a type-2 fuzzy set. Inform. Sci., Vol. 132, 2001, pp. 195-220.
24. Mazandarani, M., Xiu, L. Fractional Fuzzy Inference System: The New Generation of Fuzzy Inference Systems. IEEE Access, Vol. 8, 2020, pp. 126066-126082.
25. Piegat, A., Landowski, M. Horizontal Membership Function and Examples of its Applications. Int. J. Fuzzy Syst., Vol. 17, 2015, pp. 22-30.
26. Landowski, M. Horizontal Fuzzy Numbers for Solving Quadratic Fuzzy Equation. Advances in Intelligent Systems and Computing, Vol. 889, 2019, pp. 45-55.
27. Landowski, M. Method with horizontal fuzzy numbers for solving real fuzzy linear systems. Soft Computing, Vol. 23, 12, 2019, pp. 3921-3933.
28. Piegat, A., Landowski, M. Fuzzy Arithmetic Type 1 with Horizontal Membership Functions. Studies in Computational Intelligence, Vol. 683, 2017, pp. 233-250.
29. Haykin, S.O. Neural Networks and Learning Machines. Pearson Education, 2011, 936 p.
30. Garcia-Hernandez, R., Lopez-Franco, M., Sánchez, E.N., Alanis, A.Y., Ruz-Hernandez, J.A. Decentralized Neural Control: Application to Robotics. Studies in Systems, Decision and Control, Vol. 96, Springer International Publishing, 2017, 111 p.
31. Sánchez, E.N., Djilali, L. Neural Control of Renewable Electrical Power Systems. Studies in Systems, Decision and Control, Vol. 278, Springer International Publishing, 2020, 206 p.
32. Субботін С. О. Нейронні мережі : теорія та практика: навч. посіб. Житомир : Вид. О. О. Євенок, 2020. – 184 с.
33. Субботін, С. О. Олійник, А. О. Нейронні мережі: навчальний посібник / під заг. ред. проф. С. О. Субботіна. – Запоріжжя : ЗНТУ, 2014. – 132 с.

34. Олійник, А. О., Субботін, С. О., Олійник, О. О. Інтелектуальний аналіз даних : навч. посіб. Запоріжжя : ЗНТУ, 2011. – 271 с.
35. Elman, J.L. Finding structure in time. *Cognitive Science*. 1990. P. 179-211.
36. Schmidhuber, J. Deep Learning in Neural Networks: An Overview // *Neural Networks*. 2015. Vol. 61. P. 85-117.
37. Bengio, Y. LeCun, Y., Hinton, G. Deep Learning. *Nature*. 2015. Vol. 521. P. 436-444.
38. Hochreiter, S., Schmidhuber, J. Long short-term memory. *Neural Computation*. 1997. Vol. 9, No. 8. P. 1735-1780.
39. Gers, F., Schraudolph, N., Schmidhuber, J. Learning precise timing with LSTM recurrent networks. *Journal of Machine Learning Research*. 2002. Vol. 3. P. 115-143.
40. Hayashi, Y., Imura, A. Fuzzy neural expert system with automated extraction of fuzzy If-Then rules from a trained neural network. [1990] *Proceedings. First International Symposium on Uncertainty Modeling and Analysis, College Park, MD, USA, 1990*, pp. 489-494.
41. Jang, J.-S. R. ANFIS: Adaptive-Network-based Fuzzy Inference Systems. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 23, No. 3, 1993. pp. 665-685.
42. Jang, J.-S. R., Sun, C.-T., Mizutani, E. *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Prentice Hall, 1996.
43. Nauck, D., Klawonn, F. Kruse, R. *Foundations of Neuro-Fuzzy Systems*. John Wiley & Sons, 1997.
44. Rutkowska, D. Type 2 fuzzy neural networks: an interpretation based on fuzzy inference neural networks with fuzzy parameters. 2002 IEEE World Congress on Computational Intelligence. 2002 IEEE International Conference on Fuzzy Systems. FUZZ-IEEE'02. *Proceedings (Cat. No.02CH37291)*, Honolulu, HI, USA, 2002, vol. 2, pp. 1180-1185.

45. Castillo, O., Melin, P. An Approach for Optimization of Intuitionistic and Type-2 Fuzzy Systems in Pattern Recognition Applications. 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), New Orleans, LA, USA, 2019, pp. 1-5, doi: 10.1109/FUZZ IEEE.2019.8858951.
46. Simon, D. Evolutionary Optimization Algorithms: Biologically Inspired and Population-Based Approaches to Computer Intelligence / D. Simon // John Wiley & Sons, 2013.
47. Gogna, A, Tayal, A. Metaheuristics: review and application. Journal of Experimental & Theoretical Artificial Intelligence, Vol. 25, P. 2013, P. 503-526.
48. Субботін, С.О. Ітеративні, еволюційні та мультиагентні методи синтезу нечіткологічних і нейромережних моделей / С.О. Субботін, А.О. Олійник, О.О. Олійник // Монографія / Під заг. ред. С.О. Субботіна. – Запоріжжя: ЗНТУ, 2009. – 375 с.
49. Boussaïd, I., Lepagnot, J., Siarry, P. A survey on optimization metaheuristics. Information Sciences, Vol. 237, 2013, P. 82-117.
50. Faris, H., Mafarja, M. M., Heidari, A.A., Aljarah, I., Ala'M, Al-Z., Mirjalili, S., Fujita, H. An efficient binary salp swarm algorithm with crossover scheme for feature selection problems. Knowledge-Based Systems, 154, 2018, P. 43-67.
51. Radosavljević, J., Klimenta, D., Jevtić, M., Arsić, N. Optimal power flow using a hybrid optimization algorithm of particle swarm optimization and gravitational search algorithm. Electric Power Components and Systems, 43(17), 2015, P. 1958-1970.
52. Brownlee, J. Clever algorithms: Nature-inspired programming recipes. LuLu, 2011.
53. Langdon, W. B., Poli, R. Foundations of Genetic Programming, Springer-Verlag, 2002.

54. Hansen, N., Kern, St. Evaluating the CMA evolution strategy on multimodal test functions // *Parallel Problem Solving from Nature, PPSN 2004*. Vol. 3242, 2004, P. 282-291.
55. Eiben, A.E., Smith, J.E. *Introduction to Evolutionary Computing*, Springer, 2003.
56. Price K., Storn R., Lampinen J. *Differential evolution – a practical approach to global optimization*. Springer, 2005.
57. Chakraborty, U. K. (ed.). *Advances in Differential Evolution*. Springer, 2008.
58. Puris, A., Bello, R., Molina, D., Herrera, F. Variable mesh optimization for continuous optimization problems // *Soft Computing*. 2011. Vol. 16, No. 3, P. 511-525.
59. Dasgupta, D. (Ed.) *Artificial Immune Systems and Their Applications*, Springer-Verlag, Inc. Berlin, 1999.
60. Alves, R.T., Delgado, M.R., Lopes, H.S., Freitas, A.A. An Artificial Immune System for Fuzzy-Rule Induction in Data Mining // Yao X. et al. (Eds.) *Parallel Problem Solving from Nature - PPSN VIII*. PPSN 2004. *Lecture Notes in Computer Science*, Vol 3242. Springer, Berlin, Heidelberg, 2004, P. 1011-1020.
61. Mehrabian, A. R., Lucas, C. A novel numerical optimization algorithm inspired from weed colonization // *Ecological Informatics*. 2006. Vol. 1, P. 355-366.
62. Simon, D. Biogeography-Based Optimization. *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, 2008, pp. 702-713. doi: 10.1109/TEVC.2008.919004.
63. Yang, X. S., Deb, S. Engineering optimization by cuckoo search // *Intern. J. Mathematical Modelling and Numerical Optimization*. 2010. Vol. 1. No 4, P. 330-343.

64. Valian, E., Mohanna, H., Tavakoli, S. Improved cuckoo search algorithm for global optimization // Intern. J. Communications and Information Technology. 2011. Vol. 1. No 1, P. 31-44.
65. Dorigo, M. Ant Colony Optimization / M. Dorigo, M. Birattari // Encyclopedia of Machine Learning, Sammut C., Webb G.I. (eds.), 2011. – Springer, Boston, MA.
66. Dorigo, M., Birattari, M., Stutzle, T. Ant colony optimization. IEEE Comput. Intell., 1, 2006, P.28-39.
67. Karaboga, D. An idea based on honey bee swarm for numerical optimization. Technical Report TR06. Erciyes, Turkiye: Erciyes University Press, 2005.
68. Pham, D. T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., Zaidi, M. The bees algorithm – a novel tool for complex optimization problems // Proceedings of IPROMS 2006 Conference. 2006. P. 454-461.
69. Passino, K. M. Biomimicry of bacterial foraging for distributed optimization and control // IEEE Control Systems Magazine, Vol. 22, No 3, 2002, P. 52-67.
70. Kennedy, J. Particle swarm optimization / J. Kennedy, R. C. Eberhart // Proceedings of IEEE Int. Conf. Neural Networks. – Perth, Australia. – 1995. P. 1942-1948. DOI: 10.1109/ICNN.1995.488968.
71. Clerc, M. Particle swarm optimization. ISTE Ltd, 2006.
72. Bastos-Filho, C., de Lima-Neto, F., Lins, A., Nascimento, A., Lima, M. A novel search algorithm based on fish behavior // Paper SMC, 2008.
73. Bastos-Filho, C., de Lima-Neto, F., Lins, A., Nascimento, A., Lima, M. Fish school search // Nature-inspired Algorithms for Optimization SCI, Vol. 193, 2009, P. 261-277.
74. Madeiro, S., de Lima-Neto, F., Bastos-Filho, C., Figueiredo, E. Density as the segregation mechanism in fish school search for multimodal optimization problems // 2nd Intern. Conference on Swarm Intelligence, Part 2, LNCS 6729, 2011, P. 563-572.

75. Mirjalili, S., Mirjalili, S. M. Lewis, A. Grey wolf optimizer. *Adv. Eng. Software*, 69, 2014, P. 46-61.
76. Precup, R.-E., David, R.-C., Petriu, E. M., Szedlak-Stinean, A.-I., Bojan-Dragos, C.- A. Grey wolf optimizer-based approach to the tuning of PI-fuzzy controllers with a reduced process parametric sensitivity. *Proc. 4th IFAC Intl. Conf. Intell. Control Autom. Sci.*, Reims, France, 2016, P. 55-60.
77. Krishnanand, K. N., Ghose, D. Glowworm swarm optimization: a new method for optimizing multi-modal function // *Intern. J. Computational Intelligence Studies*, Vol. 1, No 1, 2009, P. 93-119.
78. Yang, X. S. A new metaheuristic bat-inspired algorithm // *Nature Inspired Cooperative Strategies for Optimization // Studies in Computational Intelligence*, Vol. 284, 2010, P. 65-74.
79. Pinto, P.C., Runkler, T.A., Sousa, J.M. Wasp swarm algorithm for dynamic MAX- SAT problems. *Adaptive and natural computing algorithms*. Springer, 2007, P. 350-357.
80. Gandomi, A.H., Alavi, A.H. Krill Herd: a new bio-inspired optimization algorithm. *Communications in Nonlinear Science and Numerical Simulation*, Volume 17, Issue 12, 2012, P. 4831-4845.
81. Mirjalili, S. The Whale Optimization Algorithm. *Advances in Engineering Software*, Vol. 95, 2016, P. 51-67.
82. Aarts, E. H. L., van Laarhoven, P. J. M. *Simulated annealing: theory and applications*. L.: Kluwer, 1987.
83. Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S. GSA: a gravitational search algorithm. *Information Sciences*, Volume 179, Issue 13, 2009, P. 2232-2248.
84. Webster, B., Bernhard, P.J. A local search optimization algorithm based on natural principles of gravitation. In: *Proceedings of the 2003 international conference on information and knowledge engineering (IKE'03)*, 2003, p. 255-261.

85. Formato, R.A. Central force optimization: A new metaheuristic with applications in applied electromagnetics. *Progress in Electromagnetics Research*, Vol. 77, 2007, P. 425-491.
86. Moghaddam, F.F., Moghaddam, R.F., Cheriet, M. Curved space optimization: A random search based on general relativity theory. *arXiv: Neural and Evolutionary Computing*, 2012, 1208.2214.
87. Alatas, B. ACROA: Artificial Chemical Reaction Optimization Algorithm for global optimization. *Expert Systems with Applications*, Volume 38, Issue 10, 2011, P. 13170-13180.
88. Du, H., Wu, X., Zhuang, J. Small-world optimization algorithm for function optimization. *Advances in natural computation*. Springer, 2006, P. 264-273.
89. Hatamlou. A. Black hole: a new heuristic optimization approach for data clustering. *Information Sciences*, Volume 222, 2013, P. 175-184.
90. Erol, O.K., Eksin, I. A new optimization method: big bang–big crunch. *Advances in Engineering Software*, Volume 37, Issue 2, 2006, P. 106-111.
91. Mahdavi, M. An improved harmony search algorithm for solving optimization problems // *Applied Mathematics and Computation*, Vol. 188, No 2, 2007, P. 1567-1579.
92. Omran, M. G. H., Mahdavi, M. Global-best harmony search // *Applied Mathematics and Computation*, Vol. 198, 2008, P. 643-656.
93. Rao, R.V., Savsani, V.J., Vakharia, D.P. Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems. *Information Sciences*, Volume 183, Issue 1, 2012, P. 1-15.
94. Rao, R.V., Savsani, V.J., Vakharia, D.P. Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. *Computer-Aided Design*, Volume 43, Issue 3, 2011, P. 303-315.
95. Fogel, D. *Artificial intelligence through simulated evolution*. Wiley-IEEE Press, 2009
96. Atashpaz-Gargari, E., Lucas, C. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In:



- Proceedings of the 2007 IEEE congress on evolutionary computation, CEC, 2007, P. 4661-4667.
97. He, S., Wu, Q. H., Saunders, J. R. Group Search Optimizer: An Optimization Algorithm Inspired by Animal Searching Behavior. *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, 2009, pp. 973-990. doi: 10.1109/TEVC.2009.2011992.
  98. Husseinzadeh Kashan, A. An efficient algorithm for constrained global optimization and application to mechanical engineering design: league championship algorithm (LCA). *Computer-Aided Design*, Volume 43, Issue 12, 2011, P. 1769-1792.
  99. Eita, M.A., Fahmy, M.M. Group counseling optimization: a novel approach. In: Bramer M, Ellis R, Petridis M, editors. *Research and development in intelligent systems XXVI*. London: Springer, 2010, P. 195-208.
  100. Eita, M.A., Fahmy, M.M. Group counseling optimization. *Applied Soft Computing*, Volume 22, 2014, P. 585-604.
  101. Ghorbani, N., Babaei, E. Exchange market algorithm. *Applied Soft Computing*, Volume 19, 2014, P. 177-187.
  102. Gandomi, A.H. Interior search algorithm (ISA): a novel approach for global optimization. *ISA Transactions*, Volume 53, Issue 4, 2014, P. 1168-1183.
  103. Haupt, R., Haupt, S. *Practical genetic algorithms*. New Jersey: John Wiley & Sons, 2004, 261 p.
  104. Nagata, Y., Soler, D. A new genetic algorithm for the asymmetric traveling salesman problem. *Expert Systems with Applications*, Vol. 39, No. 10, 2012, P. 8947-8953.
  105. Zhao, J., Han, L., Wang, L., Yu, Z. The fuzzy PID control optimized by genetic algorithm for trajectory tracking of robot arm, 2016 12th World Congress on Intelligent Control and Automation (WCICA), Guilin, China, 2016, P. 556-559.

106. Piwonska, A. Genetic algorithm finds routes in travelling salesman problem with profits. *Zeszyty Naukowe Politechniki Białostockiej, Informatyka*, 2010, P. 51-65.
107. Kozlov, O., Kondratenko, Y. Automatic Synthesis of Rule Bases of Fuzzy Control Systems Based on Genetic Algorithms. In: Shahbazova, S.N., Abbasov, A.M., Kreinovich, V., Kacprzyk, J., Batyrshin, I.Z. (Eds.) *Recent Developments and the New Directions of Research, Foundations, and Applications. Studies in Fuzziness and Soft Computing*, vol. 422. Springer, Cham. 2023, P. 101-123. [https://doi.org/10.1007/978-3-031-20153-0\\_9](https://doi.org/10.1007/978-3-031-20153-0_9).
108. Prakash, A., Deshmukh, S.G. A multi-criteria customer allocation problem in supply chain environment: An artificial immune system with fuzzy logic controller based approach // *Expert Systems with Applications*, Volume 38, Issue 4, 2011, P. 3199-3208. <https://doi.org/10.1016/j.eswa.2010.09.008>.
109. Visconti, A., Tahayori, H. Artificial immune system based on interval type-2 fuzzy set paradigm // *Applied Soft Computing*, Vol. 11, No. 6, 2011, P. 4055-4063. <https://doi.org/10.1016/j.asoc.2010.12.011>.
110. Kozlov, O.V. Optimal Selection of Membership Functions Types for Fuzzy Control and Decision Making Systems. *Proceedings of the 2nd International Workshop on Intelligent Information Technologies & Systems of Information Security with CEUR-WS, Khmelnytskyi, Ukraine, IntelITSIS 2021, CEUR-WS, Vol-2853*, 2021, P. 238-247.
111. Kozlov, O.V., Kondratenko, Y.P., Skakodub, O.S. Intelligent Information Technology for Structural Optimization of Fuzzy Control and Decision-Making Systems. In: Kondratenko, Y.P., Kreinovich, V., Pedrycz, W., Chikrii, A., Gil-Lafuente, A.M. (Eds.) *Artificial Intelligence in Control and Decision-making Systems. Studies in Computational Intelligence*, vol 1087. Springer, Cham. 2023, P. 127-165. [https://doi.org/10.1007/978-3-031-25759-9\\_7](https://doi.org/10.1007/978-3-031-25759-9_7).

112. De Castro L.N., Von Zuben F.J. Learning and Optimization Using the Clonal Selection Principle // IEEE Transactions on Evolutionary Computation, 2002, vol. 6, No. 3, pp. 231-259.
113. Lucinska M., Wierzchon S.T. Hybrid Immune Algorithm for Multimodal Function Optimization // Recent Advances in Intelligent Information Systems, 2009, pp. 301-313.
114. Agiza H.N., Hassan A.E., Salah A.M. An Improved Version of opt-AiNet Algorithm (I-opt-AiNet) for Function Optimization // IJCSNS International Journal of Computer Science and Network Security, March 2011, vol. 11, No. 3, pp. 80-85.
115. Aragon V.S., Escuivel S.C., Coello Coello C.A. Solving Constrained Optimization using a T-Cell Artificial Immune System, 2008.
116. Thomas, G., Lozovyy, P., Simon, D. Fuzzy Robot Controller Tuning with Biogeography-Based Optimization // Modern Approaches in Applied Intelligence: 24th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2011, Syracuse, NY, USA, June 28 – July 1, Proceedings, Part II, 2011. P. 319-327.
117. Zhang, M., Jiang, W., Zhou, X. et al. A hybrid biogeography-based optimization and fuzzy C-means algorithm for image segmentation // Soft Computing 23, 2019. P. 2033-2046.
118. D. Simon, R. Rarick, M. Ergezer, D. Du. Analytical and numerical comparisons of biogeography-based optimization and genetic algorithms. Information Sciences, Vol. 181, 2011, pp. 1224-1248.
119. Dorigo, M., Stutzle, T. Ant colony optimization. MIT Press, 2004.
120. Dorigo, M., Socha, K. Ant colony optimization for continuous domains // Eur. J. of Operational Research, Vol. 185, 2008, P. 1155-1173.
121. Bullnheimer B., Hartl R. F., Strauss C. A new rank-based version of the ant system: A computational study // Central European Journal for Operations Research and Economics. – 1999. – №7 (1). – P. 25–38.

122. Dorigo M., Maniezzo V., Colomi A. The ant system: optimization by a colony of cooperating agents // IEEE Transactions on Systems, Man, and Cybernetics. – 1996. – Part B, №26 (1). – P. 29-41.
123. Stützle T., Hoos H.H. MAX-MIN ant system // Future Generation Computer Systems. – 2000. – №16(8). – P. 889–914.
124. Dorigo M., Gambardella L. M. Ant colony system: a cooperative learning approach to the traveling salesman problem // IEEE Transactions on Evolutionary Computation. – 1997. – №1(1). – P. 53–66.
125. Tuani, A. F., Keedwell, E., Collett, M. Heterogenous Adaptive Ant Colony Optimization with 3-opt local search for the Travelling Salesman Problem. Applied Soft Computing, Vol. 97, Part B, 2020, 106720.
126. Kondratenko, Y. P., Kozlov, A. V. Parametric optimization of fuzzy control systems based on hybrid particle swarm algorithms with elite strategy // Journal of Automation and Information Sciences, Vol. 51, Issue 12, New York: Begel House Inc., 2019. P. 25-45, DOI: 10.1615/JAutomatInfScien.v51.i12.40.
127. Hernandez, E., Castillo O., Soria, J. Optimization of fuzzy controllers for autonomous mobile robots using the grey wolf optimizer. 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), New Orleans, LA, USA, 2019. – P. 1-6.
128. Saremi, S., Mirjalili, S. Z., Mirjalili, S. M. Evolutionary population dynamics and grey wolf optimizer. Neural Computing and Applications, 26(5), 2015, 1257-1263.
129. Malik, M. R. S., Mohideen, E. R., Ali, L. Weighted distance grey wolf optimizer for global optimization problems. 2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Madurai, India, 2015, pp. 1-6. DOI: 10.1109/ICCIC.2015.7435714.
130. Jayabarathi, T., Raghunathan, T., Adarsh, B.R., Suganthan, P. N. Economic dispatch using hybrid grey wolf optimizer. Energy, 11, 2016, 630-641.

131. Nadimi-Shahraki, M.H., Taghian, S., Mirjalili, S. An improved grey wolf optimizer for solving engineering problems. *J. Expert Systems with Applications*, Vol. 166, 2021, 113917.
132. Kozlov, O.V., Kondratenko, Y.P., Skakodub, O.S. Information Technology for Parametric Optimization of Fuzzy Systems Based on Hybrid Grey Wolf Algorithms // *SN Computer Science*, Volume 3, issue 6, 2022. 463.
133. Kondratenko, Y., Simon, D. Structural and parametric optimization of fuzzy control and decision making systems. Recent developments and the new direction in soft-computing foundations and applications. *Selected Papers from the 6th World Conference on Soft Computing*. Berkeley, USA, 2016. Series: Studies in Fuzziness and Soft Computing. 2018. 361. Springer International Publishing. P. 273-289.
134. Fernández, J. P., Vargas, M. A., García, J. M. V., Carrillo, J. A. C., Aguilar, J. J. C. Coevolutionary Optimization of a Fuzzy Logic Controller for Antilock Braking Systems Under Changing Road Conditions, *IEEE Transactions on Vehicular Technology*, 70, 2, 2021, 1255-1268.
135. Chen, Ch., Li, M., Sui, J., Wei, K., Pei, Q. A genetic algorithm-optimized fuzzy logic controller to avoid rear-end collisions. *Journal of advanced transportation*, Vol. 50, 2016, P. 1735-1753.
136. Stanley, K.O., Clune, J., Lehman, J., Miikkulainen, R. Designing neural networks through neuroevolution. *Nature Machine Intelligence* 1, 2019, pp. 24-35.
137. Bower, J., Shahverdi, M., Blekhman, D. Neuroevolution Based Optimization of Hybrid Transmission Shift Points. *2018 IEEE Conference on Technologies for Sustainability (SusTech)*, 2018, pp. 1-5.
138. Cordon, O., Gomide, F., Herrera, F., Hoffmann, F., Magdalena, L. Ten Years of Genetic Fuzzy Systems: Current Framework and New trends. *Fuzzy Sets and Systems* Vol. 141, Iss. 1, 2004, P. 5-31.

139. Khan, S. et al. Design and Implementation of an Optimal Fuzzy Logic Controller Using Genetic Algorithm. *Journal of Computer Science*, Vol. 4, No. 10, 2008, P. 799-806.
140. N.K. Bose, B. Buchberger, J.P. Guiver, *Multidimensional systems theory and applications*, Berlin: Springer, 268 p., 2003.
141. N. K. Bose, *Applied multidimensional systems theory*, Springer International Publishing AG, Springer, Cham, 192 p., 2017.
142. Kondratenko, Y.P., Kozlov, O.V., Gerasin, O.S. Neuroevolutionary approach to control of complex multicoordinate interrelated plants // *International Journal of Computing*, 18(4), 2019. – P. 502-514.
143. Kozlov, O. V. Gerasin, O. S., Kondratenko, G. V. Complex of tasks of monitoring and automatic control of mobile robots for vertical movement // *Int. J. Shipbuilding and marine infrastructure*. – Mykolaiv, NUS, 2017. – № 2 (8). – P. 77-87. DOI 10.15589/SMI20170215.
144. Y. Kondratenko, Y. Zaporozhets, J. Rudolph, O. Gerasin, A. Topalov, O. Kozlov, Modeling of clamping magnets interaction with ferromagnetic surface for wheel mobile robots, *International Journal of Computing*, Vol. 17, Issue 1, pp. 33-46, 2018.
145. Y.P. Kondratenko, Y.M. Zaporozhets, J. Rudolph, O.S. Gerasin, A.M. Topalov, O.V. Kozlov, Features of clamping electromagnets using in wheel mobile robots and modeling of their interaction with ferromagnetic plate, in *Proc. of the 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, Bucharest, Romania, 2017, Vol. 1, pp. 453-458.
146. Lodwick, W.A., Kacprzych, J. (Eds.) *Fuzzy Optimization* // STUDEFUZ 254, 2010. – Berlin, Heidelberg. – Springer-Verlag.
147. Kondratenko, Y.P., Al Zubi, E.Y.M. The Optimisation Approach for Increasing Efficiency of Digital Fuzzy Controllers // *Annals of DAAAM for 2009 & Proceeding of the 20th Int. DAAAM Symp. "Intelligent*

- Manufacturing and Automation", Published by DAAAM International, Vienna, Austria, 2009. – P. 1589-1591.
148. Simon, D.  $H_{\infty}$  estimation for fuzzy membership function optimization // International Journal of Approximate Reasoning, 40, 2005. – P. 224-242.
  149. Kondratenko, Y. P., Klymenko, L. P. Al Zu'bi, E. Y. M. Structural Optimization of Fuzzy Systems' Rules Base and Aggregation Models // Kybernetes. – Vol. 42, Iss: 5, 2013. – P. 831-843.
  150. Ishibuchi, H., Yamamoto, T. Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining // Fuzzy Sets and Systems. – Vol. 141, no. 1, 2004. – P. 59-88.
  151. Kondratenko, Y.P., Kozlov, O.V., Korobko O.V. Two Modifications of the Automatic Rule Base Synthesis for Fuzzy Control and Decision Making Systems // Chapter in a book: "Information Processing and Management of Uncertainty in Knowledge-Based Systems. Theory and Foundations". – Medina, J., Ojeda-Aciego, M., Verdegay, J.L., Pelta, D.A., Cabrera, I.P., Bouchon-Meunier, B., Yager, R.R. (Eds.). Book Series: Communications in Computer and Information Science, Vol. 854. – Berlin. Heidelberg: Springer International Publishing, 2018. – pp. 570-582.
  152. Kondratenko, Y.P., Kozlov, A.V. Generation of Rule Bases of Fuzzy Systems Based on Modified Ant Colony Algorithms. Journal of Automation and Information Sciences, Vol. 51, Issue 3, New York: Begel House Inc., pp. 4-25, 2019.
  153. Kozlov O., Kondratenko G., Gomolka Z., Kondratenko Y. Synthesis and Optimization of Green Fuzzy Controllers for the Reactors of the Specialized Pyrolysis Plants. In: Kharchenko V., Kondratenko Y., Kacprzyk J. (eds.) Green IT Engineering: Social, Business and Industrial Applications. Studies in Systems, Decision and Control, vol. 171. – Springer, Cham, 2019. – P. 373-396.
  154. Kondratenko, Y.P., Kozlov, O.V., Gerasin, O.S., Topalov, A.M., Korobko, O.V. Automation of Control Processes in Specialized Pyrolysis Complexes

Based on Web SCADA Systems // Proceedings of the 9<sup>th</sup> IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Vol. 1. – Bucharest, Romania. – 2017. – P. 107-112.

155. Kondratenko, Y.P., Kozlov, O.V., Kondratenko, G.V., Atamanyuk, I.P. Mathematical Model and Parametrical Identification of Ecopyrogenesis Plant Based on Soft Computing Techniques // Chapter in a book: “Complex Systems: Solutions and Challenges in Economics, Management and Engineering”. – Christian Berger-Vachon, Anna María Gil Lafuente, Janusz Kacprzyk, Yuriy Kondratenko, José M. Merigó, Carlo Francesco Morabito (Eds.). Book Series: Studies in Systems, Decision and Control, Vol. 125. – Berlin. Heidelberg: Springer International Publishing, 2018. – P. 201-233.