

РЕАЛІЗАЦІЯ ПРОЦЕСУ ВІДНОВЛЕННЯ КОНСТРУКТИВНО-ПРОДУКЦІЙНИХ МОДЕЛЕЙ ФРАКТАЛЬНИХ ЧАСОВИХ РЯДІВ

Представлений метод відновлення конструктивних моделей для наперед заданих фрактальних часових рядів. У цій роботі узагальнено підхід до реалізації програмної системи, що забезпечує ефективну автоматизацію з урахуванням особливостей роботи з модельними рядами різної природи – як детермінованими, так і стохастичними. Було проаналізовано два основні підходи до реалізації системи на основі генетичного алгоритму: монолітний та мультиагентний. Враховуючи складнощі обчислення показників життєздатності хромосом у роботі зі стохастичними рядами, було ухвалено рішення розділити окремі елементи генетичного алгоритму – зокрема, кросовер і мутацію – від етапу селекції шляхом впровадження підпопуляцій. Це дозволило здійснювати розподілене обчислення показників фітнесу хромосом. Запроваджене рішення уможливило незалежне масштабування різних елементів процесу відновлення, що підвищило загальну ефективність системи. Для забезпечення узгодженої взаємодії між елементами було розглянуто різні типи комунікацій, серед яких найбільш ефективним виявився асинхронний підхід. Були впроваджені механізми для оптимізації взаємодії між обчислювальними сутностями, а також вузловий патерн для реалізації операцій кросоверу та мутації. Такий підхід дозволив усунути проблеми, пов'язані з обробкою стохастичних часових рядів, і забезпечив можливість контрольованого та ефективного горизонтального масштабування процесу відновлення конструктивних моделей.

Ключові слова: програмна інженерія, конструктивно-продукційне моделювання, інформаційні технології, фрактали, фрактальні часові ряди, генетичний алгоритм, L-система, хмарні обчислення.

V. I. Shynkarenko, A. A. Zhadan

IMPLEMENTATION OF THE RECONSTRUCTION PROCESS FOR CONSTRUCTIVE-SYNTHESIZING MODELS OF FRACTAL TIME SERIES

The presented method focuses on reconstructing constructive models for predefined fractal time series. This work generalizes the approach to implementing the software system that ensures effective automation while considering the specifics of working with model series of various natures – both deterministic and stochastic. Two main approaches to system implementation based on a genetic algorithm were analyzed: the monolithic and the multi-agent. Considering the complexity of calculating chromosome viability indicators when working with stochastic series, it was decided to separate certain elements of the genetic algorithm – specifically, crossover and mutation – from the selection stage by introducing subpopulations. This made it possible to perform distributed computation of chromosome fitness indicators. The introduced solution enabled independent scaling of various elements of the reconstruction process, which increased the overall efficiency of the system. To ensure consistent interaction between elements, different types of communication were considered, among which the asynchronous approach proved to be the most effective. Mechanisms were implemented to optimize interaction between computational entities, as well as a node pattern for implementing crossover and mutation operations. This approach made it possible to eliminate problems associated with processing stochastic time series and ensure controlled and efficient horizontal scaling of the process of reconstructing constructive models.

Key words: software, constructive-synthesizing modeling, information technologies, fractals, fractal time series, genetic algorithm, L-system, cloud computing.

Вступ

Часовий ряд є одним із найпоширеніших способів представлення зміни станів складної системи, що забезпечує можливість формального опису динаміки процесів та їхнього розвитку упродовж

визначеного проміжку часу для подальшого аналізу закономірностей їх наступних змін. За своєю суттю часовий ряд являє собою впорядковану послідовність числових значень, кожне з яких відповідає певному

моменту або інтервалу часу. Аналіз послідовностей дозволяє виявляти внутрішні закономірності та тенденції, що визначають поведінку системи [1].

Формалізація часових рядів уможливорює застосування до них широкого спектру математичних і статистичних методів, спрямованих на встановлення зв'язків між попередніми та поточними значеннями ряду. Виявлення таких залежностей дозволяє визначити математичну модель, яка описує структуру процесу та може бути використана для подальшого аналізу, прогнозування або керування системою [2]. Застосування подібних моделей є одним із ключових етапів дослідження складних процесів у різних галузях – від економіки та фінансів до технічних систем, екологічного моніторингу тощо [3, 4].

Розв'язання задач, пов'язаних із моделюванням часових рядів, може здійснюватися за допомогою широкого спектра підходів – як класичних математичних, так і сучасних методів штучного інтелекту. Класичні підходи включають, зокрема, лінійні моделі авторегресії [5], моделі ковзного середнього [6], комбіновані моделі ARIMA [7], сезонні моделі SARIMA [8], тощо. Ці методи ґрунтуються на припущенні про стаціонарність процесу та лінійну залежність між послідовними спостереженнями.

Методи штучного інтелекту, зокрема, нейронні мережі, дозволяють розглядати складні нелінійні залежності, адаптуватися до змін середовища та ефективно працювати з рядами різного походження [9].

В основу будь-якої моделі покладено гіпотезу про характер зв'язку між послідовними значеннями часового ряду. Цей зв'язок може бути лінійним або нелінійним.

Нелінійні залежності є більш поширеними та враховують широкий спектр взаємодій між змінними, зокрема, ефекти затримки, хаотичну поведінку та фрактальні властивості [10], що сприяє їх використанню для дослідження природних або техногенних процесів.

Конструктивно-продукційне моделювання [12] є новим підходом формування модельних фрактальних часових

рядів [13]. Такі моделі передбачають використання породжувальної L-системи [14], яка визначає правила розгортки процесу p , та початкового набору математичних параметрів, таких як дисперсія D та математичне очікування наступного значення ряду V_f та його зміни dV_f . У процесі інтерпретації фінальної розгортки системи значення відповідних параметрів змінюються згідно із заданими правилами, що забезпечує гнучке формування динамічної структури ряду. Самоподібність досягається на основі ітеративної розгортки L-системи шляхом поетапного заміщення символів лівої частини правил правою на кожній ітерації.

Залежно від набору терміналів і математичних параметрів можливо здійснювати моделювання як детермінованих (рис. 1), так і стохастичних (рис. 2) часових рядів. Зокрема, на основі однієї конструктивної моделі стохастичного часового ряду може бути згенерована множина рядів різної форми через варіювання значень, отриманих за допомогою генерації точок відповідно до закону нормального розподілу з урахуванням дисперсії (рис. 3). Такий підхід дозволяє відтворювати широкий спектр поведінкових сценаріїв системи, що підвищує точність та адаптивність моделі.

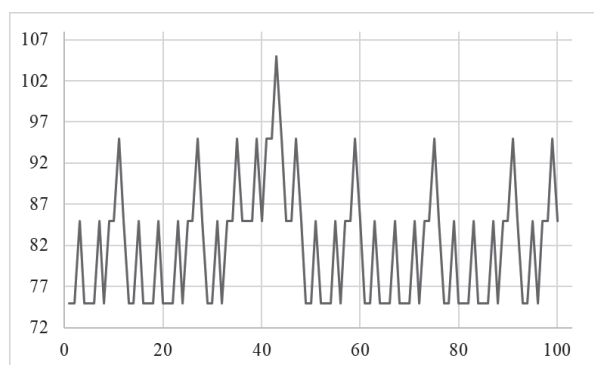


Рис. 1. Детермінований модельний ряд. Визначна частина моделі:

$$\{p: f \rightarrow +ff + f - f-, V_f = 35, dV_f = 10\}$$

Окрім зазначеного напрямку була також розглянута обернена задача – визначення конструктивної моделі на основі наданого часового ряду. Даний процес передбачає ітераційний підбір параметрів і структурних елементів моделі з метою отримання ряду, який би максимально від-

творював наданий часовий ряд. Були розглянуті підходи до роботи як з детермінованими, так й зі стохастичними часовими рядами з використанням генетичного алгоритму як базису процесу із впровадженням необхідних модифікацій.

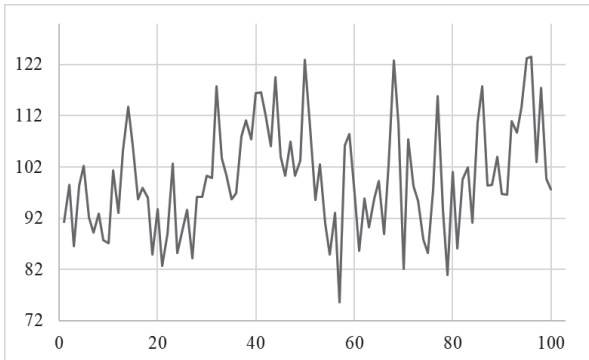


Рис. 2. Стохастичний ряд. Визначна частина моделі: $\{p: f \rightarrow +f + f - f-, V_f = 35, dV_f = 10, D = 25\}$

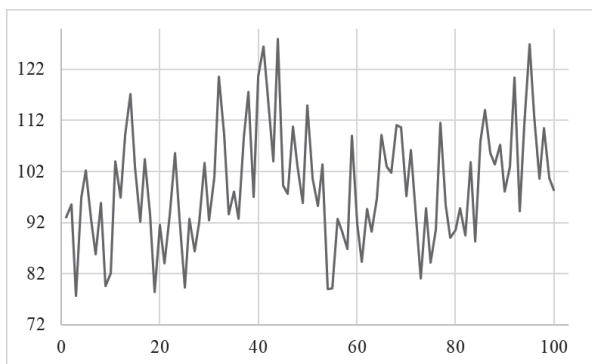


Рис. 3. Інша варіація стохастичного ряду. Визначна частина моделі: $\{p: f \rightarrow +f + f - f-, V_f = 35, dV_f = 10, D = 25\}$

Зважаючи на складність визначеного підходу відновлення, окремою задачею було визначення та розробка ефективної реалізації програмного додатку для автоматизації процесу випробувань та подальшого впровадження зазначеного методу.

Не зважаючи на поверхневу визначеність та вичерпність функціональних вимог, окремі елементи потребували ухвалення нетривіальних рішень на рівні імплементації програмного додатку.

Методологія процесу відновлення моделі ряду

Відповідно до визначеної методології та на основі базових сутностей констру-

ктивно-продукційного моделювання були визначені складові конструктори в межах загального композитного конструктора C_M [15]:

- C_{MS} – формує фрактальну мультисимвольну послідовність як у L-системах;
- C_{TS} – перетворює вищезазначену символьну послідовність у фрактальний часовий ряд відповідно до параметрів V_f, dV_f, D ;
- C_{RS} – відновлює похідну модель наданого часового ряду.

Відповідно до поставлених цілей, були визначені два можливі режими виконання композитного конструктора C_M : контроль якості перетворення часових рядів у конструктивну модель (MQ) та екстраполяція часових рядів для прогнозування (MF) [15].

Основною метою режиму MQ є передбачення процесу послідовного виконання для проведення масових випробувань на модельних часових рядах, згенерованих за заздалегідь визначеною конструктивною моделлю, що описується наступним поетапним циклічним процесом:

- випадковим чином генерує визначна частина моделі: аксіома, правила підстановки та параметри V_f, dV_f, D
- послідовно ініціює виконання конструкторів C_{MS} , C_{TS} та C_{RS} ;
- передає параметри та отримує згенеровані структури (правило підстановки знайдене конструктором C_{RS});
- порівнює дані, передані конструктору C_{MS} та отримані від C_{RS} правила підстановки, визначає різницю між ними.

У свою чергу, виконання в режимі MF передбачає послідовність для обробки наданого ряду з генерацією його конструктивної моделі та його прогнозовані значення. Порядок виконання включає етапи:

- отримує на вході модельний часовий ряд;
- послідовно ініціює реалізацію конструкторів C_{RS} , C_{MS} та C_{TS} ;
- отримує сформовані структури та часовий ряд із прогнозними значеннями.

Основним конструктором, який відповідає саме за процес відновлення конструктивної моделі є C_{RS} . Він у свою чергу також базується на наборі визначених алгоритмів обробки ітерації [15]:

- $B_1 \left| \begin{matrix} r, V_f, dV_f, D \\ Y_1 \end{matrix} \right|$ – формування випадкової правої частини r правила p з заданого алфавіту $\{f, +, -\}$, значення параметрів у заданих межах $Y_1 = \{f, +, -\}, V_{fmin}, V_{fmax}, dV_{fmin}, dV_{fmax}, D_{min}, D_{max}$;
- $B_2 \left| \begin{matrix} X, P \\ P, r, V_f, dV_f, D \end{matrix} \right|$ – випадковим чином заповнює параметрами хромосому X та додає її до поточної популяції P ;
- $B_3 \left| \begin{matrix} \Omega(C_{MS}) \\ \parallel \Rightarrow (C_{MS}), r(X_n) \end{matrix} \right|$ – виконує операцію виведення $\parallel \Rightarrow$ конструкції Ω конструктором C_{MS} – мультисимвольного ланцюжка за вихідним правилом r хромосоми X_n ;
- $B_4 \left| \begin{matrix} \Omega(C_{RS}) \\ \parallel \Rightarrow (C_{RS}), \Omega(C_{MS}) \end{matrix} \right|$ – визначає часовий ряд на основі $\Omega(C_{RS})$ конструктором C_{RS} та за допомогою мультисимвольного ланцюжка $\Omega(C_{MS})$;
- $B_5 \left| \begin{matrix} q(X_n) \\ \Omega(C_{RS}), TS_{In} \end{matrix} \right|$ – визначає показник життєздатності q хромосоми X_n за розбіжністю між вихідним рядом TS_{In} та результатом роботи конструктора C_{MS} ;
- $B_6 \left| \begin{matrix} P \\ P \end{matrix} \right|, B_7 \left| \begin{matrix} P \\ P \end{matrix} \right|, B_8 \left| \begin{matrix} P \\ P \end{matrix} \right|$ – сортує хромосоми в популяції та видаляє найгі-

рші з них. Також вони додають нові хромосоми, що утворилися під час фаз кросинговеру та мутації.

Формування детермінованих модельних часових рядів виконується при заданій дисперсії $D = 0$. Коли $D > 0$, формуються стохастичні часові ряди.

Для детермінованих часових рядів показник життєздатності визначається за середньоквадратичним відхиленням між модельним та відновленим часовими рядами:

$$q(X_n) = \sum_{k=1}^K (TS(C_{MS}, C_{TS})_k - TS_{In_k})^2. \quad (1)$$

Для стохастичних часових рядів розрахунок показника життєздатності здійснюється з урахуванням варіативності можливих значень часових рядів, породжених однією конструктивною моделлю. Щоб врахувати цю особливість, у розрахунки було впроваджено генерацію кількох часових рядів на основі поточної хромосоми, а вхідні параметри було розширено відповідно до декількох рядів:

$$q(X_n) = \sum_{i=0}^N \min_j \left\{ \sum_{k=1}^K (TS_{In_{j,k}} - TS(C_{MS}, C_{RS})_{i,k})^2 \right\}_{j=1}^M \quad (2)$$

де N – визначена кількість рядів до генерації на основі конструктивної моделі поточної хромосоми, M – кількість вхідних модельних часових рядів.

У випадку модельних часових рядів, для відповідності вимогам до вхідних моделей та очікуваної конструктивної моделі генерується відповідний набір часових рядів [16].

Автоматизація процесу відновлення моделі для наданого часового ряду

Відповідно до вимоги автоматизації процесу роботи відновлення конструктивних моделей було спроектовано та розроблено декілька варіацій програмних додатків. Початковий прототип, який використовувався для роботи з детермінованими часовими рядами різної складності,

спроектований за допомогою монолітної архітектури й поєднував у собі реалізацію композитного конструктора C_M та всіх його зазначених складових (рис. 4).

Кожен із модулів відповідає за реалізацію відповідного конструктора або конструкторів, загальне керування процесом:

- модуль обробки обраного режиму роботи – визначає обраний режим роботи конструктора C_M та визначення порядку його виконання;
- модуль інтерпретації конструктивної моделі – відповідає за реалізацію обох конструкторів C_{MS} та C_{TS} у фінальний ланцюжок символів та його інтерпретацію у фрактальний ряд;
- модуль генетичного алгоритму – відповідає за реалізацію конструктора C_{RS} і відповідно за імплементацію генетичного алгоритму та процесу відновлення конструктивної моделі.

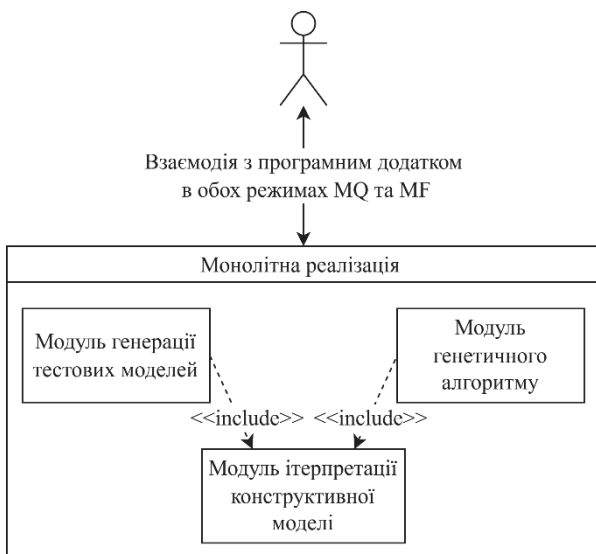


Рис. 4. Монолітна реалізація автоматизації відновлення конструктивної моделі

Дана реалізація використовувалася для початкового тестування підходу на основі модельних детермінованих часових рядів [15]. Слід зазначити, що вона була оптимізована саме для роботи із цим типом рядів. Утім, з урахуванням відмінностей при роботі зі стохастичними рядами та потреби обробки їхніх даних було ухва-

лено рішення про розширення програмного додатку та можливості масштабування ресурсів.

За відправну точку було обрано реалізацію на основі мультиагентного середовища [16], де кожний агент відповідає за окремий процес відновлення конструктивної моделі на основі окремої популяції (рис. 5). Для взаємного обміну генами між довільно обраними сутностями було впроваджено механізм міграції хромосом між вузлами обчислень, за керування якої відповідає окремий міграційний агент. Основною особливістю зазначеного підходу є можливість масштабування кількості робочих агентів, що дозволяло пришвидшити процес відновлення. Комунікація у розробленому середовищі реалізовувалася за допомогою шини повідомлень.

Відповідно до порівняння з монолітним підходом, кожен робочий агент є окремою реалізацією композитного конструктора C_M та усіх його складових.

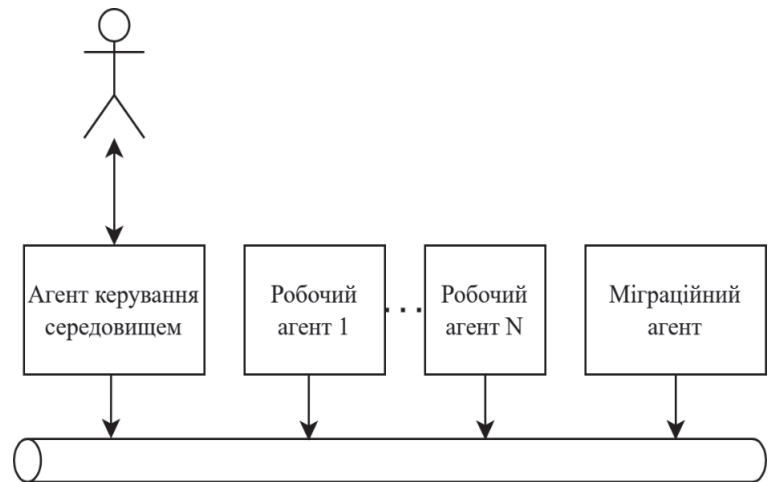


Рис. 5. Мільтиагентний підхід до реалізації

Розподілення обчислень показників для стохастичних рядів

На основі проведених досліджень подібний підхід показав ефективність роботи з детермінованими часовими рядами, а також забезпечив можливість обробки стохастичних часових рядів для тестування методу у відповідних умовах [16].

Попри отримані позитивні результати, для роботи зі стохастичними рядами

мультиагентна реалізація не є оптимальною. Основною проблемою є нерівноцінність операцій конструктора C_{RS} , а саме складність обчислення показника $q(X_n)$ для стохастичних рядів.

Відповідно до вищезгаданих особливостей та значень параметрів N та M , зазначена операція для кожної хромосоми популяції є вузьким місцем процесу відновлення моделей.

Для вирішення визначеної проблеми було ухвалено рішення щодо незалежного масштабування окремих елементів конструктора C_{RS} , а саме алгоритмів $\{B_3, B_4, B_5\}$:

$$\left\langle \begin{array}{l} \beta_\theta \rightarrow \langle B_3 \cdot B_4 \cdot B_5 \rangle_{sub_i} \cdot \beta_\theta \\ \beta_\theta \rightarrow \varepsilon \end{array} \right\rangle \quad (3)$$

де масштабування реалізується за рахунок виділення підпопуляцій sub_i , для кожної з яких виділяється окремий вузол обчислень, інтегрований у правило заміщення β [15].

В свою чергу необхідність впровадження розбиття та агрегації розширює функціонал алгоритмів $B_7 \left|_P^P, B_8 \left|_P^P\right.\right.$, а саме:

- $B_7 \left|_P^P, \{P_{sub_i}\}\right.$ – забезпечує розширення популяції новими хромосомами та виділяє підпопуляції;
- $B_8 \left|_P^P, \{P_{sub_n}\}\right.$ – разом із операцією селекції виконує агрегацію підпопуляцій.

Було розглянуто два основні підходи до реалізації [17], а саме із синхронним та асинхронним методами комунікації між сутностями. В першому випадку транспорт інформації між сутностями реалізовувався за допомогою технології синхронної передачі, або за допомогою протоколу HTTP, або з використанням технології RPC [18]. Даний підхід забезпечує надійний та простий механізм пересилки та виключає необхідність впровадження складної мануальної синхронізації. Загалом цей підхід є розширенням вже зазначеної мультиагентної реалізації.

Втім, основним його недоліком є складність самої топології, яка б забезпечувала легкість масштабування та гнучкі залежності між обчислювальними сутностями (рис. 6).

Другий підхід забезпечує простоту масштабування та низький рівень зв'язності між сервісами (рис. 7). Водночас основним викликом залишається реалізація операції агрегації підпопуляцій.

Ключова проблема полягає у поєднанні збереження проміжних підпопуляцій у виділеному сховищі з одноразовим виконанням операції злиття без створення вузьких місць та дублюючих популяцій (рис. 8).

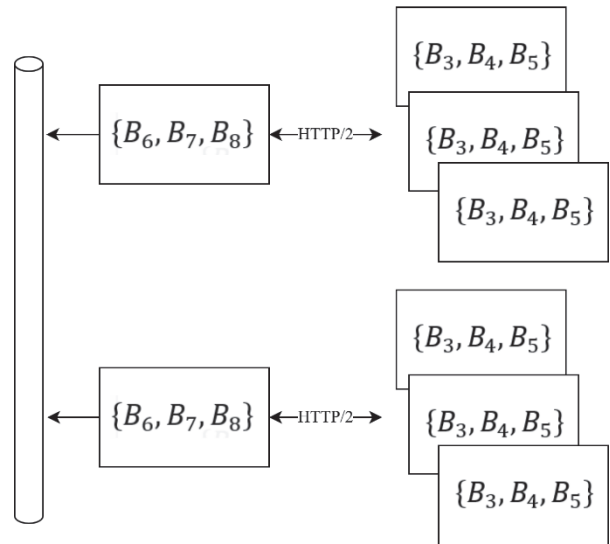


Рис. 6. Розподілення етапу обчислень показників хромосом із синхронною комунікацією

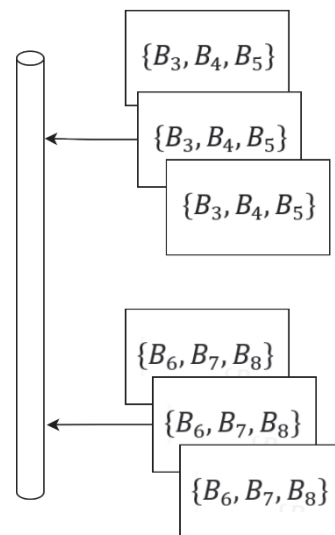


Рис. 7. Розподілення етапу обчислень показників хромосом із асинхронною комунікацією

Вирішенням зазначеної проблеми стало впровадження механізму загального синхронізуючого запису, який оновлюється лише за умови дотримання визначеного критерію. Суть підходу полягає у створенні синхронізуючого запису перед початком процесу відновлення. Під час виконання агрегації, після досягнення необхідної кількості підпопуляцій, система оновлює цей запис, інкрементуючи параметр поточної ітерації лише за умови, якщо його значення відповідає номеру поточної ітерації. Якщо зазначена умова не виконується, операція агрегації не проводиться.

Такий підхід забезпечує одноразове виконання операції завдяки поєднанню конт-

рольної умови та відповідного параметра. Крім того, він дозволяє масштабувати кількість сервісів-агрегаторів відповідно до кількості сервісів, що здійснюють розрахунок фітнесу хромосом, забезпечуючи тим самим підвищення загальної ефективності системи.

Загалом роздільне масштабування зазначених компонентів дозволило ефективніше розпоряджатися обчислювальними ресурсами відповідно до складності ітерацій. Але окрім необхідності у збільшенні контролю за синхронізацією слід також зазначити й ефект самого розподілення на швидкодію роботи системи у негативному плані.

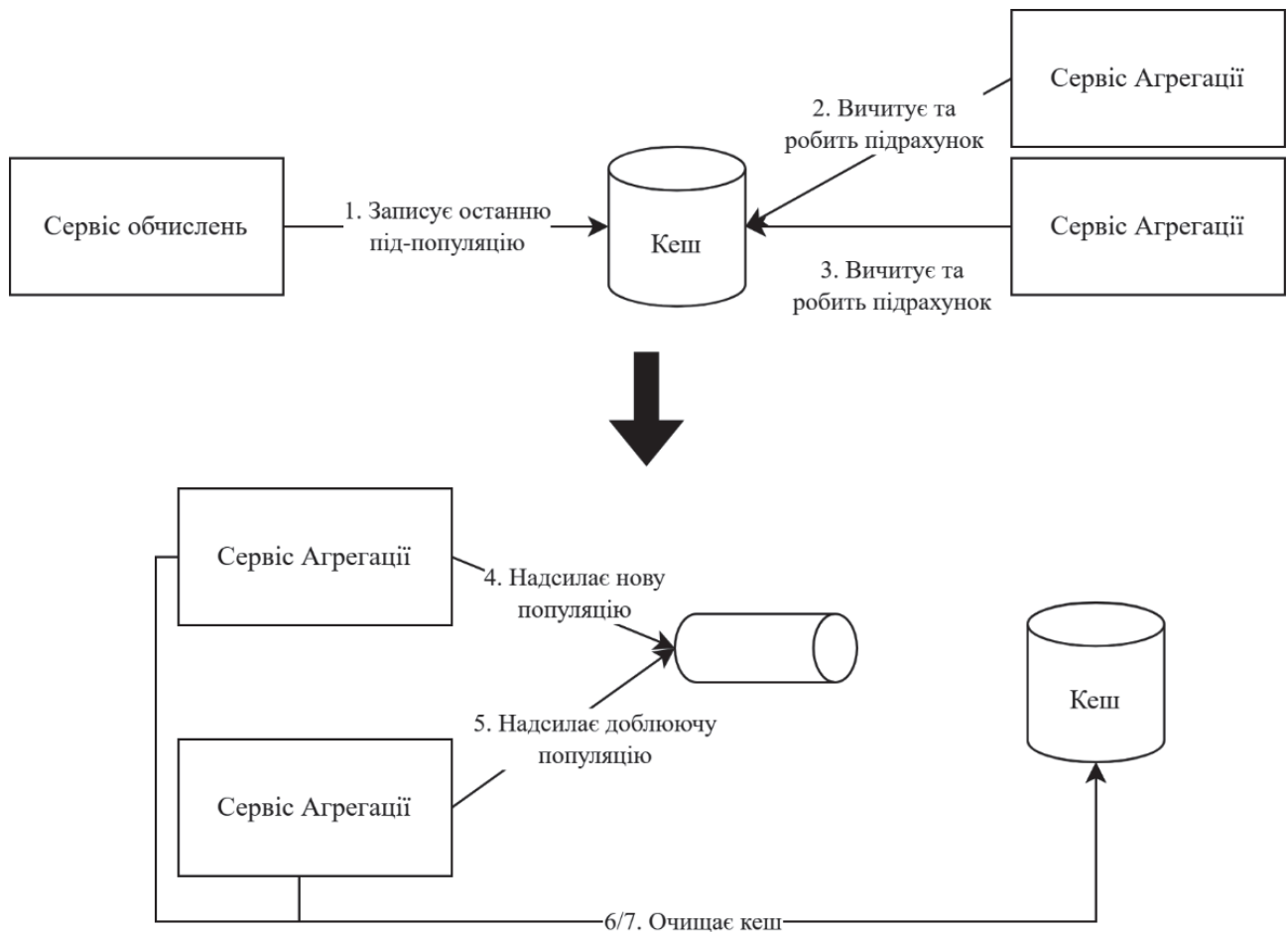


Рис. 8. Візуалізація проблеми синхронізації та дублювання популяції

Працюючи з детермінованими часовими рядами та за малих кількостей хромосом у популяції й, відповідно, меншої кількості підпопуляції, система витрачає більше часу на пересилання даних між розподіленими сутностями, ніж у разі звичайного монолітного обчислення, що є передбачувано. Втім даний недолік нівелю-

ється перевагами роботи з великими об'ємами даних, під час яких загальний середній час ітерацій виходить на плато за умови рівного масштабування сервісів обчислення та агрегації підпопуляцій.

Окрім розподілення загального процесу обчислень, було реалізовано механізм нелокуючого потоку керування для основ-

них операцій генетичного алгоритму – кросоверу та мутації. Основна ідея реалізації полягала у виділенні окремих вузлів обробки, кожен з яких відповідає за певний етап обчислень алгоритму B_7 . Взаємодія між вузлами здійснюється за допомогою неблокуючих черг повідомлень (рис. 9).

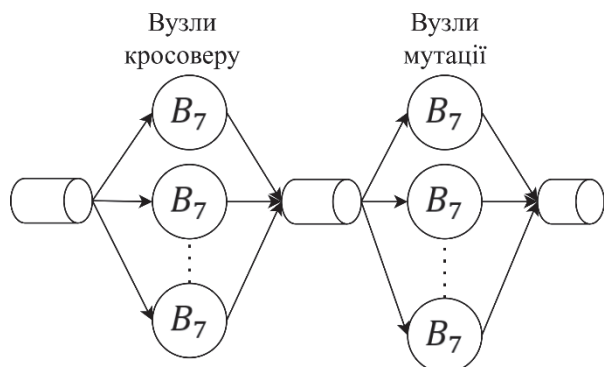


Рис. 9. Вузлова реалізація основних операцій генетичного алгоритму

Додатково визначено глобальні черги, призначені для пересилання сигналів про завершення процесу та повідомлень про помилки, що можуть виникати під час обчислень.

Основними перевагами впровадженого підходу є можливість окремого масштабування конкретних операцій генетичного алгоритму, а також зручність подальшого впровадження нових етапів і операцій у процес.

Обговорення

У ході аналізу ідентифіковані основні вузькі місця процесу обчислень, зокрема, розрахунок показника фітнесу для стохастичних часових рядів. Це зумовлено необхідністю врахування їхньої варіативності під час генерації на основі однієї моделі. Для вирішення цієї проблеми було ухвалено рішення про розподіл процесу генетичного алгоритму на кілька обчислювальних сутностей та введення підпопуляцій.

Таке розподілення дозволяє розпаралелити процес обчислення показників життєздатності особин популяції. Залежно від розміру підпопуляцій досягається зменшення загальної кількості обчислювальних операцій і підвищення швидкодії системи.

Для досягнення максимальної ефективності кількість сутностей, що виконують розрахунок показників життєздатності,

має відповідати кількості підпопуляцій. Це дає змогу виконувати обчислення за один прохід. В іншому випадку набори хромосом очікуватимуть своєї черги.

Ефективність запропонованого підходу обмежується можливістю масштабування фізичних обчислювальних вузлів, необхідних для досягнення максимальної продуктивності системи.

Висновки

У результаті проведених досліджень та аналізу попередніх реалізацій було спроектовано та реалізовано систему для відновлення конструктивних моделей методами генетичного алгоритму. Під час роботи були визначені основні недоліки попередніх імплементацій, що проявлялися у роботі з різними типами часових рядів і враховували їхні специфічні особливості в межах уже відомих монолітного та мульти-агентного підходів.

У подальшому вдосконалення архітектурного підходу може передбачати використання підпопуляцій не лише на етапі обчислення показників фітнесу, а й у межах інших фаз генетичного алгоритму.

References

1. He X. Time Series Analysis. In: Geographic Data Analysis Using R. – Singapore: Springer, 2024. – ISBN 978-981-97-4021-5. – DOI: 10.1007/978-981-97-4022-2_6.
2. Verma, P., Reddy, S. V., Ragha, L., Datta, D. Comparison of Time-Series Forecasting Models. International Conference on Intelligent Technologies (CONIT). – 2021. – P. 1–7. – DOI: 10.1109/CONIT51480.2021.9498451.
3. Arashi M., Rounaghi M.M. Analysis of market efficiency and fractal feature of NASDAQ stock exchange: Time series modeling and forecasting of stock index using ARMA-GARCH model. Future Business Journal. – 2022. – Vol. 8. – P. 14. – DOI: 10.1186/s43093-022-00125-9.
4. Schaffer A.L., Dobbins T.A., Pearson S.A. Interrupted time series analysis using autoregressive integrated moving average (ARIMA) models: a guide for evaluating large-scale health interventions. BMC Medical Research Methodology. – 2021. – Vol. 21. – P. 58. – DOI: 10.1186/s12874-021-01235-8.

5. Kaur J., Parmar K.S., Singh S. Autoregressive models in environmental forecasting time series: a theoretical and application review. *Environmental Science and Pollution Research*. – 2023. – Vol. 30. – P. 19617–19641. – DOI: 10.1007/s11356-023-25148-9.
6. Su Y., Cui C., Qu H. Self-Attentive Moving Average for Time Series Prediction. *Applied Sciences*. – 2022. – Vol. 12, No. 7. – P. 3602. – DOI: 10.3390/app12073602.
7. Wang G., Su H., Mo L., Yi X., Wu P. Forecasting of soil respiration time series via clustered ARIMA. *Computers and Electronics in Agriculture*. – 2024. – Vol. 225. – 109315. – DOI: 10.1016/j.compag.2024.109315.
8. Daryl A., Winata S., Kumara S., Suhartono D. Predicting Stock Market Prices using Time Series SARIMA. 2021 1st International Conference on Computer Science and Artificial Intelligence (ICCSAI), Jakarta, Indonesia, 2021. – P. 92–99. – DOI: 10.1109/ICCSAI53272.2021.9609720.
9. Wen X., Li W. Time Series Prediction Based on LSTM-Attention-LSTM Model. *IEEE Access*. 2023. – Vol. 11. – P. 48322–48331. – DOI: 10.1109/ACCESS.2023.3276628.
10. Porcaro C., Moaveninejad S., D'Onofrio V., Di Ieva A. Fractal Time Series: Background, Estimation Methods, and Performances. In: Di Ieva A. (ed.) *The Fractal Geometry of the Brain*. *Advances in Neurobiology*, vol. 36. Cham: Springer, 2024. – DOI: 10.1007/978-3-031-47606-8_5.
11. Gospodinova E. Fractal Time Series Analysis by Using Entropy and Hurst Exponent. In: *Proceedings of the 23rd International Conference on Computer Systems and Technologies (CompSysTech '22)*. 2022. Pp. 69–75. DOI: 10.1145/3546118.3546133.
12. Skalozub V., Ilman V., Bilyy B. Constructive multiplayer models for ordering a set of sequences, taking into account the complexity operations of formations. *Science and Transport Progress. Bulletin of Dnipropetrovsk National University of Railway Transport*, 2020, pp. 61–76. DOI: 10.15802/stp2020/213232.
13. Shynkarenko K., Lytvynenko R., Chyhir I., Nikitina I. Modeling of lightning flashes in thunderstorm front by constructive production of fractal time series. In: *Advances in Intelligent Systems and Computing*, vol. 1080. Springer, 2020, pp. 173–185. DOI: 10.1007/978-3-030-33695-0_13.
14. Lupiani I. L-Systems for Plant Generation. In: *Procedural Content Generation for Games*. Berkeley, CA: Apress, 2025. DOI: 10.1007/979-8-8688-1787-8_7.
15. Shynkarenko V., Zhadan A. Modeling of the deterministic fractal time series by one rule constructors. 2020 IEEE 15th International Conference on Computer Sciences and Information Technologies (CSIT), Zbarazh, Ukraine, 2020, pp. 336–339. DOI: 10.1109/CSIT49958.2020.9321923.
16. Shynkarenko V., Zhadan A., Halushka O. Multi-Agent System for Reconstruction Constructive Models of Stochastic Fractal Time Series. *ITIAS@IT&I 2024*, 2024, pp. 66–77. Available: https://ceur-ws.org/Vol-3955/Paper_6.pdf
17. Shynkarenko V., Zhadan A. Multiservice Architecture of Software for Stochastic Fractal Time Series Forecasting. 2024 IEEE 19th International Conference on Computer Sciences and Information Technologies (CSIT), 2024, pp. 1–4. DOI: 10.1109/CSIT65290.2024.10982626.
18. Muniyandi V. Utilizing gRPC for High-Performance Inter-Service Communication in .NET. *SSRN Electronic Journal*, 2025. URL: <https://doi.org/10.2139/ssrn.5381441>.

Одержано: 10.11.2025

Внутрішня рецензія оримана: 17.11.2025

Зовнішня рецензія оримана: 18.11.2025

Про авторів:

Шинкаренко Віктор Іванович,
доктор технічних наук, професор,
<https://orcid.org/0000-0001-8738-7225>
E-mail: shinkarenko_vi@ua.fm

Жадан Артем Анатолійович,
аспірант,
<https://orcid.org/0009-0003-1133-1630>
E-mail: gadan998@gmail.com

Місце роботи авторів:

Український державний університет
науки і технологій,
49010, Україна, Дніпро,
вул. академіка Лазаряна, 2.
E-mail: office@ust.edu.ua