

УДК 004.421:519.863

DOI: 10.31673/2412-9070.2025.042240

М. В. КОРЕПАНОВ, аспірант,

ORCID: 0009-0005-8735-6388

Державний університет інформаційно-комунікаційних технологій, Київ

АЛГОРИТМИ ОПТИМІЗАЦІЇ: ПОРІВНЯННЯ І ВИКОРИСТАННЯ В ІНФОРМАЦІЙНИХ СИСТЕМАХ

У статті розглянуто можливості підвищення ефективності інформаційних систем (ІС) підприємств шляхом використання алгоритмів оптимізації. Представлена загальна постановка проблеми з оглядом важливості впровадження сучасних та максимально адаптованих інформаційних систем підприємств, важливість їх покращення та оптимізації. Проаналізовано п'ять методів: алгоритм рою частинок, генетичний алгоритм, диференціальну еволюцію, алгоритм імітації відпалу і бджолиний алгоритм. Досліджено їх основні переваги та недоліки при використанні. Визначено сфери їх застосування в інформаційних системах, зокрема управління ресурсами, логістика, планування, тощо. У сучасних інформаційних системах підприємств, де критично важливими є обробка великих обсягів даних та прийняття рішень у реальному часі, вибір ефективного алгоритму оптимізації має важливе значення. Результати порівняльного аналізу показали, що жоден з алгоритмів не є універсальним для всіх типів задач, проте всі вони мають потенціал бути частиною процесу оптимізації в інформаційних системах підприємств. Проведене дослідження дозволяє узагальнити, що вибір оптимального алгоритму залежить від конкретного поставленого завдання, вимог до швидкості та точності прийняття рішення, наявності ресурсів, тощо. У багатьох випадках використання гібридних підходів, що поєднують переваги кількох алгоритмів, можуть забезпечити найкращі результати. Розглянуто попередні плани щодо подальших досліджень використання алгоритмів оптимізації у процесах вдосконалення інформаційних систем підприємств, в тому числі розробка та огляд гібридних моделей, які поєднують переваги кількох алгоритмів, для досягнення кращого балансу між точністю, швидкістю та стабільністю.

Ключові слова: інформаційні системи підприємства; генетичний алгоритм; алгоритм рою частинок; диференціальна еволюція; алгоритм імітації відпалу; бджолиний алгоритм; алгоритми оптимізації.

Вступ

Постановка проблеми. Інформаційні системи (ІС) все частіше стають вирішальною складовою у сучасній цифровій трансформації підприємств. Вони сприяють автоматизації управління, логістики, виробництва та інших робочих процесів, що дозволяє зменшити витрати, підвищити продуктивність, гарантувати прозорість діяльності та оперативність ухвалення рішень. Компанії повинні вміти швидко адаптуватися до змін зовнішніх чинників в існуючому конкурентному середовищі, що вимагає від їхніх інформаційних систем високої гнучкості і адаптивності. Тим не менш при фактичному застосуванні, незважаючи на широке розповсюдження інформаційних технологій, багато компаній продовжують стикатися з проблемами низької ефективності власних ІС.

Причини різні, наприклад, використання застарілих підходів до обробки інформації, слабка інтеграція між складовими системи, недостатній рівень адаптації до змін у бізнес процесах, невисокий рівень інтелектуалізації системи, тощо. Однією з найгостріших проблем є неефективне використання ресурсів ІС, що проявляється у тривалому часі обробки запитів, надмірному навантаженні на обчислювальні потужності, нераціональному розподілі обмежених ресурсів. Саме в цьому контексті постає актуальність використання сучасних алгоритмів оптимізації.

Методи оптимізації – потужний інструмент, який дозволяє знаходити найбільш вигідні рішення, враховуючи наявні ресурси, обмеження і умови. Їх застосування особливо підходить до завдань, де стандартні евристичні або жорстко детерміновані підходи виявляються недостатньо ефективними. Зокрема, алгоритми, розроблені по аналогії з природними явищами (алгоритми рою частинок, генетичні алгоритми, диференціальна еволюція, тощо) вже зарекомендували себе у вирішенні складних задач пошуку у багатовимірних просторах [1-5]. Ці алгоритми демонструють помітну ефективність у ситуаціях, коли необхідно забезпечити баланс між якістю рішення і часом обчислення. Практичне впровадження таких алгоритмів в інформаційні системи дозволяє оптимізувати розподіл ресурсів, зменшити кількість помилок, підвищити швидкість обробки даних, покращити загальну продуктивність системи. Численні приклади з наукової літератури та практики ІТ-проектів свідчать про значне зростання ефективності роботи ІС після впровадження компонентів оптимізації, в тому числі при розв’язанні завдань управління логістикою, планування виробництва, розподілу навантаження на сервери та ін. [6-9].

Отже, проблема підвищення ефективності інформаційних систем підприємств шляхом застосування алгоритмів оптимізації є надзвичайно актуальною як у науковому, так і в практичному аспектах. Вона пов’язана з різними дисциплінами, поєднуючи елементи інформатики, прикладної математики, інженерії та економіки. Важливість цього питання виходить за межі суто технологічного інтересу, адже ефективне функціонування ІС безпосередньо впливає на продуктивність та прибутковість підприємства, конкурентоспроможність галузі й в результаті загальний економічний розвиток країни. Крім того, ця тема є важливою з огляду на розвиток самої сфери інформаційних технологій, в тому числі удосконалення алгоритмів оптимізації сприяє появі нових методів інтелектуального аналізу даних, автоматизації та зростанню ефективності рішень, створенню адаптивних, самонавчальних ІС. Таким чином, дослідження в цьому напрямку відкривають перспективи для подальших розробок у сфері інформаційних технологій, штучного інтелекту, цифрової трансформації підприємств, тощо.

Аналіз останніх досліджень і публікацій

У сучасних дослідженнях значна увага зосереджується на підвищенні продуктивності та ефективності інформаційних систем підприємств за допомогою застосування алгоритмів оптимізації. В тому числі, проводяться активні дослідження генетичних алгоритмів (GA) та алгоритмів рою частинок (PSO) для оптимізації виробничих планів і процесів управління ресурсами [6, 7, 9]. Наприклад, дослідження, опубліковане на ResearchGate [9], демонструє ефективність GA для вирішення питань планування виробництва, поєднуючи ідеї ІТ-виробництва (політика постачання, що вимагає доставки матеріалів, товарів та послуг саме у той час, коли вони стають потрібні для роботи або процесу) з реальними виробничими даними. PSO алгоритми також застосовуються для покращення загального управління підприємствами. Опубліковане в ScienceDirect дослідження [7] свідчить, що запропонований алгоритм PSO гарантує високу надійність і точність у процесі підвищення ефективності управління. Інше дослідження на ResearchGate [6] розглядає використання покращеного PSO для впровадження внутрішніх ресурсів компанії, що дозволяє ефективно поєднувати матеріальні та людські ресурси для досягнення різних поставлених цілей. Варто відмітити опубліковане в PeerJ Computer Science дослідження [8], яке описує особливості фінансового управління підприємствами шляхом оптимізації фінансових моделей, при цьому робиться акцент на підвищенні точності під час прогнозування ризиків.

Ці дослідження підкреслюють необхідність та важливість використання різних сучасних алгоритмів оптимізації для підвищення ефективності ІС підприємств, особливо беручи до уваги, що складність бізнес процесів постійно зростає та є необхідність швидкої адаптації до змін ринку.

Формулювання мети статті

З огляду на актуальність проблеми підвищення ефективності інформаційних систем підприємств та велику кількість існуючих підходів до її вирішення, виникає потреба у система-

тичному порівняльному аналізі сучасних алгоритмів оптимізації для вибору тих, що мають потенціал до впровадження в ІС різного типу.

Метою даної статті є проведення теоретичного аналізу п'яти алгоритмів оптимізації, які широко використовуються для вирішення складних задач у багатовимірних просторах, характерних для інформаційних систем підприємств.

Для аналізу були відібрані такі алгоритми: генетичний алгоритм [5], алгоритм рою частинок [2], диференціальна еволюція [1], алгоритм імітації відпалу [3] та бджолиний алгоритм [4]:

- **Генетичний алгоритм (Genetic Algorithm, GA)** — один з найпопулярніших еволюційних алгоритмів, що імітує механізми природного відбору та спадковості.

- **Алгоритм рою частинок (Particle Swarm Optimization, PSO)** — метод чисельної оптимізації, для використання якого не потрібно знати точного градієнта оптимізованої функції.

- **Диференціальна еволюція (Differential Evolution, DE)** — потужний метод для пошуку глобального оптимуму у безперервних просторах.

- **Алгоритм імітації відпалу (Simulated Annealing, SA)** — стохастичний метод, що імітує процес охолодження металів і дозволяє уникнути локальних мінімумів.

- **Бджолиний алгоритм (Bee Algorithm, BA)** — біоінспірований підхід, що моделює поведінку бджіл при пошуку джерел нектару, ефективний у комбінаторних задачах.

Критеріями відбору стали: поширеність алгоритмів у наукових дослідженнях і практичних розробках, наявність доведеної ефективності в реальних застосуваннях, адаптивність до задач з великим числом змінних і обмежень, а також наявність відповідних джерел для проведення обґрунтованого аналізу.

Подальші розділи статті присвячено короткому опису кожного з вказаних алгоритмів, аналізу їхніх ключових переваг і недоліків, а також порівнянню їх потенціалу у контексті підвищення ефективності інформаційних систем підприємств.

Результати дослідження

Для кожного з алгоритмів, що досліджувались, представлений короткий опис, їх основні переваги, окремо виділені недоліки, а також застосування в інформаційних системах підприємств. На підставі проведеної роботи складено порівняльну таблицю.

Генетичний алгоритм (GA) — це еволюційний метод пошуку та оптимізації, що нагадує процеси природного відбору в генетиці, біологічну еволюцію. Він оперує популяцією можливих рішень (хромосом), які еволюціонують через ітеративні процеси відбору, схрещування (кросоверу) та випадкові зміни (мутації). В процесі кожна хромосома оцінюється за допомогою функції пристосованості, що визначає якість запропонованого рішення. Поступово відбираються кращі представники, на основі їхніх властивостей формуються нові популяції, що забезпечує покрокове наближення до оптимального результату.

Серед головних переваг генетичного алгоритму можна зазначити здатність знаходити глобальні найкращі рішення у складних і багатовимірних просторах, де традиційні методи часто зупиняються на локальних екстремумах. GA також не потребує інформації про похідні функцій або їхню безперервність, це дозволяє використовувати його у широкому спектрі задач, зокрема — комбінаторних і стохастичних. Також алгоритм може легко адаптуватися до паралельних обчислень, що надає змогу пришвидшити процес оптимізації при наявності відповідних обчислювальних ресурсів.

До недоліків відносимо повільну збіжність — GA часом може потребувати великої кількості ітерацій для досягнення оптимального рішення, що збільшує витрати на обчислення. Також є ризик передчасної збіжності — є ймовірність, що алгоритм зійдеться до неоптимального рішення, особливо при недостатній різноманітності популяції. Ефективність GA доволі сильно залежить від правильного вибору параметрів, таких як розмір популяції, ймовірності схрещення та мутації.

У сфері інформаційних систем різних підприємств генетичний алгоритм зарекомендував себе як дієвий інструмент, що використовується для оптимізації широкого спектра задач. Наприклад, його застосовують для формування ефективних виробничих розкладів, що дозволяє

мінімізувати витрати й скоротити час виконання замовлень. Окрім того, ГА використовується для оптимального розподілу обмежених ресурсів, таких як трудові або обчислювальні потужності, для побудови логістичних маршрутів з урахуванням множини умов і обмежень. Також у рамках ІС його можуть застосовувати для аналізу даних, підвищення надійності архітектури системи, а також при прийнятті рішень в умовах невизначеності [6, 9]. Загалом завдяки своїй універсальності та адаптивності, генетичний алгоритм – це перспективний інструмент для реалізації складних функціональних блоків у сучасних інформаційних системах.

Алгоритм рою частинок (PSO) – це метод чисельної оптимізації, для використання якого не потрібно знати точного градієнта оптимізованої функції. Цей підхід імітує процес взаємодії між частинками, які шукають оптимальні рішення задачі у багатовимірному просторі. Кожна частинка зберігає інформацію про свою поточну позицію, найкраще знайдену позицію в минулому, а також враховує досвід інших частинок у популяції, що дозволяє поєднувати індивідуальний та колективний інтелект у процесі пошуку.

До ключових переваг цього алгоритму відносять його простоту реалізації та невелику кількість параметрів, які необхідно налаштувати. PSO не вимагає обчислення похідних від цільової функції, що робить його універсальним для широкого спектру задач, включаючи ті, в яких функція пристосованості є дискретною або складною [6, 7]. Завдяки своїй здатності до глобального пошуку рішень у просторі, цей алгоритм демонструє стійкість до локальних екстремумів і може працювати у випадках, коли інформація неповна або нечітка. Також важливою перевагою є можливість ефективного масштабування та адаптації до різних типів задач оптимізації.

Серед недоліків знову виділяємо повільну збіжність – PSO часом може потребувати великої кількості ітерацій для досягнення оптимального рішення, що збільшує витрати на обчислення. Також «популярний» недолік серед алгоритмів оптимізації – існує ризик передчасної стабілізації, тобто коли всі частинки зосереджуються в обмеженій ділянці простору, не досягнувши глобального оптимуму. І складне налаштування параметрів – ефективність PSO доволі сильно залежить від правильного вибору параметрів, таких як розмір популяції, ймовірності схрещення та мутації.

В інформаційних системах підприємств алгоритм рою частинок може застосовуватися для розв’язання низки практичних задач. Перед усім – це оптимізація розподілу ресурсів у системах з великою кількістю користувачів, планування виробничих графіків, керування логістичними мережами, адаптивне налаштування компонентів інформаційних систем. У сфері автоматизації прийняття рішень PSO добре зарекомендував себе при вирішенні задач, що пов’язані із багатьма змінними та обмеженнями, які притаманні сучасним підприємствам. Його гнучкість та здатність до самоорганізації роблять цей метод ефективним інструментом для підвищення адаптивності й продуктивності інформаційних систем у сучасному конкурентному середовищі.

Алгоритм диференціальної еволюції (DE) – це потужний інструмент для глобальної оптимізації, який є особливо ефективним у багатовимірних просторах пошуку. Він був запропонований як простий та ефективний евристичний метод для мінімізації нелінійних та недиференційованих функцій у безперервних просторах. Принцип роботи DE базується на еволюційних механізмах, які подібні до генетичних алгоритмів, але з унікальним підходом до мутації та кросоверу. Алгоритм оперує популяцією рішень, які представляються у вигляді векторів з реальними значеннями. На кожній ітерації для кожного вектора (індивіда) створюється мутантний вектор шляхом додавання зваженої різниці між двома випадково обраними векторами до третього вектора. Цей мутантний вектор надалі комбінується з поточним вектором через операцію схрещення, утворюючи пробний вектор. Якщо пробний вектор має кращі показники пристосованості ніж поточний, то він замінює останній у наступному поколінні.

Серед основних переваг DE відзначаємо його простоту реалізації та мінімальну кількість параметрів, які необхідно налаштувати. Алгоритм не вимагає інформації про похідні функції, це дозволяє застосовувати його в широкому спектрі задач, включаючи ті, де функція пристосованості є недиференційованою або має багато локальних екстремумів. Крім того, DE демон-

струє високу стійкість до застрягання в локальних мінімумах та здатність ефективно досліджувати простір пошуку.

До недоліків відносимо повільну збіжність – при складних або високовимірних функціях пристосованості. При обранні недостатньої різноманітності популяції з метою скорочення часу прийняття рішення існує також ймовірність передчасної збіжності до субоптимального рішення. Не можемо знов не згадати про високу чутливість до налаштувань параметрів (коефіцієнт масштабування, ймовірність схрещення), що може впливати на стабільність та якість рішення.

В інформаційних системах підприємств алгоритм диференціальної еволюції знаходить широке застосування. Його використовують як для оптимізації розподілу ресурсів, так і для планування виробничих процесів, управління на різних етапах постачання та логістики, а також для налаштування параметрів складних інформаційних систем [6]. Завдяки здатності ефективно обробляти багатовимірні та складні задачі, DE є потужним інструментом для підвищення ефективності та адаптивності інформаційних систем у динамічному бізнес-середовищі.

Алгоритм імітації відпалу (Simulated Annealing, SA) – загальний алгоритмічний метод розв'язання задачі глобальної оптимізації, особливо дискретної та комбінаторної оптимізації, в якому процедура пошуку глобального розв'язку імітує фізичний процес відпалу. Він схожий на процес відпалу в металургії, де матеріал нагрівають і повільно охолоджують для досягнення стабільної кристалічної структури. У контексті оптимізації SA імітує цей процес, поступово зменшуючи параметр "температури" для керування ймовірністю прийняття нових рішень, включаючи ті, які можуть тимчасово погіршити поточне рішення, щоб уникнути локальних мінімумів і наблизитись до глобального оптимуму. SA розпочинається з початкового рішення та високої температури. На кожній ітерації генерується нове сусіднє рішення і залежно від різниці у функції пристосованості та поточної температури, нове рішення може бути прийняте. У подальшому цей процес повторюється з поступовим зменшенням температури згідно із визначеним графіком охолодження. Це дозволяє алгоритму досліджувати простір рішень на початкових етапах і зосереджуватися на локальному пошуку на наступних етапах.

Переваги SA включають його здатність уникати застрягання у локальних екстремумах через можливе прийняття гірших рішень на попередніх етапах пошуку. Це робить його особливо корисним для задач з великою кількістю локальних мінімумів або складною топологією простору рішень. Крім того, SA є гнучким і може бути використаний у різних задачах оптимізації, включаючи дискретні та неперервні проблеми.

Недоліки включають чутливість до параметрів – ефективність SA значною мірою залежить від точного налаштування параметрів, таких як початкова температура, графік охолодження та критерії зупинки. Це потребує наявності досить високого досвіду у оператора, оскільки неправильний вибір цих параметрів може призвести до неефективного пошуку або знову ж таки до передчасної збіжності до субоптимального рішення. Також через поступове зниження температури та ймовірний характер прийняття рішень, SA може потребувати великої кількості ітерацій для досягнення оптимального або близького до оптимального рішення, а це значно збільшує обчислювальні витрати. Існує відсутність гарантій оптимальності – хоч SA і має високий потенціал знаходити глобальний оптимум, немає гарантії, що він завжди досягне найкращого можливого рішення, особливо, якщо час прийняття рішення або наявні ресурси обмежені.

У сфері інформаційних систем підприємств SA знаходить застосування у певних задачах, де є необхідність оптимізувати складні процеси або системи із значною кількістю змінних та обмежень. Наприклад, його використовують для оптимізації управління ресурсами, планування виробництва та логістики, налаштування параметрів складних інформаційних систем, тощо [7]. Завдяки своїй гнучкості та вмінню ефективно досліджувати простір рішень, SA є потужним інструментом для підвищення ефективності та адаптивності інформаційних систем.

Бджолиний алгоритм (Bee Algorithm, BA) – алгоритм для знаходження глобальних екстремумів (максимумів чи мінімумів) складних багатовимірних функцій. Цей алгоритм моделює процеси дослідження та експлуатації джерел їжі бджолами, поєднуючи глобальний та локальний пошук для знаходження оптимальних рішень у складних просторах задач. У бджолиному

алгоритмі кожне потенційне рішення поставленого завдання розглядається як "квітка" з певною кількістю "нектару", що відповідає значенню функції пристосованості. Алгоритм розпочинається з генерації випадкової популяції рішень (бджіл), вони оцінюються за допомогою функції пристосованості. Найкращі рішення обираються для подальшого локального пошуку, а навколо них генеруються нові рішення з метою покращення результату. Інші бджоли виконують глобальний пошук та досліджують нові області простору рішень. Цей підхід дозволяє ефективно балансувати між дослідженням нових рішень та покращенням вже існуючих з метою вибору оптимальних.

До переваг бджолиного алгоритму відносять його здатність ефективно досліджувати простір рішень завдяки поєднанню глобального та локального пошуків. Алгоритм є доволі гнучким і може бути застосований до широкого спектра задач оптимізації, вирішуючи в тому числі дискретні, безперервні та комбінаторні проблеми. Окрім того, бджолиний алгоритм також має високу стійкість до застрягання у локальних екстремумах, його ефективно використовують у задачах з великою кількістю локальних мінімумів.

Один з основних недоліків – висока чутливість до параметрів, ефективність алгоритму значною мірою залежить від правильного налаштування параметрів: кількість бджіл, розмір області локального пошуку, критерії зупинки. Неправильний вибір цих параметрів може призвести до неефективного пошуку або передчасного субоптимального рішення. Повільна збіжність – через поступове зниження температури і ймовірний характер прийняття рішень, ВА може потребувати великої кількості ітерацій для досягнення оптимального або близького до оптимального рішення, а це значно збільшує обчислювальні витрати. Існує складність реалізації бджолиного алгоритму, що може бути складнішою порівняно з іншими метаевристичними методами через необхідність моделювання поведінки бджіл та управління різними типами пошуку. Відсутність гарантій оптимальності – хоч ВА і має потенціал знаходити глобальний оптимум, немає гарантії, що він завжди досягне найкращого можливого рішення, особливо, якщо час або ресурси обмежені.

У сфері інформаційних систем підприємств ВА знаходить застосування у певних задачах, де є необхідність оптимізувати складні процеси або системи із значною кількістю змінних та обмежень. Наприклад, його використовують для оптимізації управління ресурсами, планування виробництва та логістики, налаштування параметрів складних інформаційних систем [6, 9]. Завдяки своїй гнучкості та здатності ефективного дослідження простору рішень, бджолиний алгоритм є потужним інструментом для підвищення ефективності та адаптивності інформаційних систем у динамічному бізнес-середовищі.

У сучасних інформаційних системах підприємств, де обробка великих обсягів даних та прийняття рішень у реальному часі є критично важливими, вибір ефективного алгоритму оптимізації має важливе значення. Далі розглянемо порівняльний аналіз п'яти обраних алгоритмів за ключовими критеріями:

- **Ефективність пошуку.** Алгоритми DE та PSO демонструють високу ефективність у глобальному пошуку завдяки своїм механізмам еволюційного та колективного пошуку відповідно. GA також може бути достатньо ефективним, особливо при належному налаштуванні операторів мутації та схрещення. SA як правило менш ефективний, особливо у складних просторах через ймовірнісний характер пошуку, тоді як ВА забезпечує баланс між глобальним та локальним пошуком.

- **Стійкість до локальних екстремумів.** Алгоритми SA та ВА мають вбудовані механізми для уникнення застрягання у локальних мінімумів, тоді як PSO та DE, хоча й також ефективні, часом можуть потребувати прийняття додаткових стратегій для уникнення передчасної збіжності.

- **Швидкість збіжності.** PSO зазвичай має швидку збіжність завдяки простоті оновлення позицій частинок. DE в свою чергу також демонструє хорошу швидкість, особливо при точному налаштуванні параметрів. GA може бути повільнішим через складність операцій селекції та схрещення. SA має повільну збіжність через поступове охолодження, тоді як ВА, як правило, найбільш повільний, особливо при великій кількості бджіл та складних функціях.

- **Чутливість до параметрів.** Складно вибрати алгоритми за цим критерієм. Наприклад, SA та BA можна вважати найбільш чутливими до налаштувань, таких як температура охолодження або кількість бджіл. PSO та DE мають меншу кількість параметрів, але їх налаштування суттєво впливає на ефективність. GA потребує ретельного налаштування операторів мутації та схрещення.

- **Складність реалізації.** PSO та DE відзначаються простотою реалізації з мінімальною кількістю параметрів. GA та BA можуть бути складнішими через необхідність реалізації складних операторів та механізмів. SA є відносно простим, але вимагає точного налаштування температурного графіка.

- **Адаптивність до змін.** BA та PSO добре адаптуються до змін у середовищі завдяки своїм механізмам пошуку. DE та GA відносно добре адаптуються, але можуть потребувати додаткових механізмів для адаптації. Можна вважати, що SA найменш адаптивний через фіксований графік охолодження.

Проведене дослідження дозволяє узагальнити, що вибір оптимального алгоритму залежить від конкретного поставленого завдання, вимог до швидкості та точності прийняття рішення, наявності ресурсів тощо. Порівняльні характеристики алгоритмів оптимізації зведено до таблиці. У багатьох випадках використання гібридних підходів, що поєднують переваги кількох алгоритмів, можуть забезпечити найкращі результати.

Порівняльні характеристики алгоритмів оптимізації

Критерій	Генетичний алгоритм (GA)	Алгоритм рою частинок (PSO)	Диференціальна еволюція (DE)	Алгоритм імітації відпалу (SA)	Бджолиний алгоритм (BA)
Ефективність пошуку	Висока (залежить від налаштувань)	Висока	Висока	Середня	Висока
Стійкість до локальних екстремумів	Середня (залежить від селекції)	Середня	Середня	Висока	Висока
Швидкість збіжності	Середня	Висока	Висока	Низька	Середня
Чутливість до параметрів	Висока	Середня	Середня	Висока	Висока
Складність реалізації	Середня	Низька	Низька	Низька	Висока
Адаптивність до змін	Середня	Висока	Середня	Низька	Висока

Висновки

У статті проведено порівняльний аналіз п'яти сучасних алгоритмів оптимізації (генетичного алгоритму, алгоритму рою частинок, диференціальної еволюції, алгоритму імітації відпалу та бджолиного алгоритму) з метою визначення їхнього потенціалу для підвищення ефективності інформаційних систем підприємств. Кожен із розглянутих алгоритмів має свої переваги та сильні сторони, які можуть бути ефективно реалізовані у різних умовах та типах завдань оптимізації, характерних для сучасних ІС.

Генетичний алгоритм показав високу універсальність, але вимагає дуже ретельного налаштування параметрів. Алгоритм рою частинок відзначається достатньо швидкою збіжністю і простотою реалізації, що робить його привабливим для багатьох практичних застосувань. Диференціальна еволюція виявився ефективним і збалансованим методом з гарною здатністю до глобального пошуку. Алгоритм імітації відпалу, попри повільну збіжність, добре справляє-

ться з проблемами локальних мінімумів. Бджолиний алгоритм демонструє сильні сторони у комбінаторних задачах, хоча його реалізація є дещо складнішою.

Результати порівняльного аналізу показали, що жоден з алгоритмів не є універсальним для всіх типів задач, проте всі вони мають потенціал бути частиною процесу оптимізації в інформаційних системах підприємств. Найбільш перспективними в умовах бізнес-середовища, що динамічно розвивається, виглядають алгоритм рою частинок та бджолиний алгоритм, насамперед через свою здатність адаптуватися до змін і ефективно здійснювати глобальний пошук.

У подальших дослідженнях плануються наступні кроки:

- розробка та огляд гібридних моделей, які поєднують переваги кількох алгоритмів, для досягнення кращого балансу між точністю, швидкістю та стабільністю;
- експериментальна перевірка ефективності обраних алгоритмів на практичних задачах у сферах управління ресурсами, логістики, виробничого планування тощо;
- розробка адаптивних механізмів налаштування параметрів алгоритмів у реальному часі;
- створення програмних прототипів, які дозволять інтегрувати ці алгоритми в існуючі корпоративні ІС.

Таким чином, результати дослідження свідчать про доцільність і перспективність застосування алгоритмів оптимізації у процесах вдосконалення інформаційних систем підприємств та створюють підґрунтя для подальших прикладних і теоретичних досліджень у цьому напрямку.

Список літератури

1. Storn R., Price K. Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces // *Journal of Global Optimization*. – 1997. – Vol. 11, No. 4. – P. 341–359.
2. Kennedy J., Eberhart R. Particle swarm optimization // *Proceedings of ICNN'95 – International Conference on Neural Networks*. – 1995. – Vol. 4. – P. 1942–1948.
3. Kirkpatrick S., Gelatt C. D., Vecchi M. P. Optimization by Simulated Annealing // *Science*. – 1983. – Vol. 220, No. 4598. – P. 671–680.
4. Pham D. T., Ghanbarzadeh A., Koc E., Otri S. The Bees Algorithm – A Novel Tool for Complex Optimisation Problems // *Intelligent Production Machines and Systems / Ed. by D.T. Pham*. – Elsevier, 2005. – P. 454–459.
5. Goldberg D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. – Reading (MA): Addison-Wesley, 1989. – 412 p.
6. Ma Z., Tang Y. An Optimization Method for Enterprise Resource Integration Based on Improved Particle Swarm Optimization [Електронний ресурс]. – Режим доступу: <https://www.researchgate.net/publication/361004065>
7. Zhang J., et al. Particle swarm optimization-based decision-making model for enterprise digital transformation // *Journal of Innovation & Knowledge*. – 2024. – Vol. 9, No. 1. – [Електронний ресурс]. – Режим доступу: <https://www.sciencedirect.com/science/article/pii/S2772941924000802>
8. Chen X., et al. Financial optimization of enterprise systems based on swarm intelligence // *PeerJ Computer Science*. – 2024. – No. 2812. – [Електронний ресурс]. – Режим доступу: <https://peerj.com/articles/cs-2812>
9. Zhang M., et al. Optimization study on enterprise production plan based on genetic algorithm [Електронний ресурс]. – Режим доступу: <https://www.researchgate.net/publication/281666652>

M. Korepanov

COMPARISON AND USE OF OPTIMIZATION ALGORITHMS IN INFORMATION SYSTEMS

The article explores the possibilities of enhancing the efficiency of enterprise information systems (IS) through the implementation of optimization algorithms. It presents a general problem state-

ment highlighting the importance of utilizing modern and highly adaptable enterprise information systems, as well as the necessity of their improvement and optimization. The article briefly reviews recent studies and publications in this area. The aim of the article is formulated as a theoretical analysis of five modern optimization algorithms.

The following five methods are analyzed: Genetic Algorithm, Particle Swarm Optimization, Differential Evolution, Simulated Annealing, and the Bee Algorithm. Their main advantages are studied, while specific disadvantages in practical application are also highlighted. The areas of application within information systems are identified, including resource management, logistics, planning, and others. In modern enterprise information systems – where processing large volumes of data and making real-time decisions is critically important – the choice of an effective optimization algorithm plays a significant role. Results of the comparative analysis show that none of the algorithms is universal for all types of tasks; however, each has the potential to be part of the optimization process in enterprise information systems. The conducted research allows us to conclude that the choice of an optimal algorithm depends on the specific task, requirements for decision-making speed and accuracy, availability of resources, and other factors. The comparative characteristics of the optimization algorithms are given and summarized in Table 1.

In many cases, the use of hybrid approaches – combining the advantages of multiple algorithms – can provide the best results. This article also outlines future research plans concerning the use of optimization algorithms in improving enterprise information systems, including the development and review of hybrid models that integrate the strengths of several algorithms to achieve a better balance between accuracy, speed, and stability.

Keywords: enterprise information systems; information systems improvement; genetic algorithm; particle swarm optimization; differential evolution; simulated annealing; bee algorithm; optimization algorithms.
