

УДК 004.8

Гулаєва Н.М., к. ф.-м. н.

## Еволюційні алгоритми

*У роботі проведено комплексний аналіз стану царини практичного застосування сучасних еволюційних алгоритмів, а також визначення світових тенденцій і перспектив розвитку еволюційних технологій.*

*Ключові слова:* генетичний алгоритм, еволюційна стратегія, еволюційне програмування, генетичне програмування, алгоритм мурашки, алгоритм бджолиного рою, епідемічний алгоритм.

Національний університет «Києво-Могилянська академія», вул. Сковороди 2, Київ 04655, Україна, e-mail: ngulayeva@yahoo.com

Gulayeva N.M., PhD

## Evolutionary Algorithms

*In this paper, a comprehensive analysis of the realm of the practical application of modern evolutionary algorithms is provided. Also, global trends and prospects for the evolutionary technologies are determined.*

*Key words:* genetic algorithm, evolutionary strategy, evolutionary programming, genetic programming, ant algorithm, bees algorithm, epidemic algorithm.

National University of Kyiv-Mohyla Academy, 2 Skovorody vul., Kyiv 04655, Ukraine, e-mail: ngulayeva@yahoo.com

**Вступ.** Природа завжди вражала своєю складністю і водночас не випадковою взаємопов'язаністю. Тільки зараз людина почала детальніше вивчати великі та малі соціальні системи, імунні та нейронні системи, ланцюги зв'язків між різними видами та складні процеси життєдіяльності, що безупинно відбуваються всередині та зовні кожної живої істоти. Це лише невелика частина всіх чудес, що стали більш очевидними, коли ми глибше дослідили себе самих як біологічних істот та частину складніших систем Всесвіту.

Вивчаючи кожну нову для себе концепцію, людина застосовує унікальну здатність проводити аналогії між, на перший погляд, геть різними життєвими фактами чи законами природи. Ця здатність є корисною не тільки для запам'ятовування нового матеріалу, а й підштовхує людину до наукового поступу і відкриттів. Наприклад, теорія гравітації завдячує деякому яблуку, яке своїм падінням розвинуло хід думок у Ісаака Ньютона, а пліснява у чашці Петрі з неприбраної лабораторії Олександра Флемінга подарувала світу пеніцилін.

Зовсім не дивує той факт, що дослідники математичних та комп'ютерних задач у пошуках натхнення звернулись до теорії еволюції. Можливість функціонування обчислювальної системи з простими механізмами мінливості та

відбору по аналогії з законами еволюції в природних системах була досить привабливою.

Найскладнішим при побудові обчислювальної системи, заснованої на принципі природного відбору, і застосуванні такої системи у прикладних задачах, є те, що природні системи є дуже складними і, як на погляд пересічної людини, хаотичними. Можливо, людина ніколи не зможе досягнути розумом всі закони і принципи, за якими щомиті відбувається її життя і все, що обертається навколо нього. Фактично, будь-які комп'ютерні обчислення мають чітко визначений характер. Коли ми звертаємося до обчислювальної системи за розв'язком задачі, ми сподіваємось на максимально швидке виконання з мінімально можливими витратами (як часу, так і пам'яті). Ми маємо конкретну ціль: отримати коректний розв'язок тієї чи іншої задачі. Коли відбуваються реальні процеси відбору, схрещування, можливо, мутації, природа не має ніяких обмежень. Виживання в природі може не бути спрямованим до якоїсь фіксованої цілі. Не кожне нове створіння є життєздатним. Замість того, щоб міркувати про доцільність чи про ймовірність наступного кроку, природа просто робить цей крок у доступному для неї напрямку. Природа – це життя. Єдиною метою для життя є продовження життя за будь-яких обставин.

На відміну від природного процесу, обчислювальна система на основі генетичного алгоритму, як правило, створена для отримання можливого і прийнятного розв'язку деякої конкретної задачі (або неодноразового отримання таких розв'язків) за скінченний проміжок часу.

Тому еволюційну метафору можна застосовувати для оптимізаційних задач чи задач пошуку у будь-якій галузі, за умови, що можливі розв'язки задачі можна деяким чином закодувати у рядок символів, коректно визначити оператори вибірки, мутації та схрещування для таких рядків і описати деяку функцію, що оцінюватиме життєздатність такого розв'язку за виконання умов задачі. Оператор схрещування вносить у закодований рядок суттєві зміни, забезпечуючи пошук на всій множині можливих розв'язків, а оператор мутації потроху «налаштовує» оптимальніший розв'язок.

Генетичними алгоритмами послуговуються як математика (оптимізація функцій) і комп'ютерні науки (нейронні мережі, штучний інтелект тощо), так і майже всі науки, які мають справу з оптимізаційними обчисленнями: економіка, фізика, астрономія, хімія, екологія, а останнім часом генетичні алгоритми навіть проникли у царину творчості і допомагають в архітектурному і інтер'єрному дизайні, створенні картин і музичних композицій, пошитті нових фасонів одягу тощо.

Дослідження даної теми є безперечно актуальним, а вивчення еволюційних алгоритмів наразі відбувається у багатьох університетах і наукових центрах світу, адже розробка, аналіз та застосування ефективних і універсальних методів розв'язку задач (алгоритмів) є наразі ключовою задачею комп'ютерних наук, а тема еволюційних алгоритмів ще не достатньо досліджена. Генетичні алгоритми надають можливість швидкої генерації прийнятних розв'язків задач, які неможливо розв'язати іншими аналітичними методами, уникаючи повного перебору і значно скорочуючи часові витрати.

При застосуванні традиційних методів оптимізації і пошуку, у разі навіть незначної зміни параметрів середовища всі обчислення доводиться проводити заново. Еволюційний підхід дозволяє проводити аналіз і адаптацію вже створеної популяції до нових умов середовища,

чим скорочує час роботи алгоритму і реалізує принцип машинної адаптації і навчання.

Дана робота є дослідженням еволюційних алгоритмів, їх особливостей при розв'язуванні різних прикладних задач, і робить огляд сучасних застосувань еволюційних алгоритмів у різноманітних сферах людської діяльності – від вирішення теоретичних задач до побутового застосування результатів еволюційних обчислень. ГА є одним з найперспективніших напрямів розвитку комп'ютерних технологій, тому що охоплюють широчезне коло оптимізаційних (і не тільки) задач. Останні дослідження вчених у цій галузі продемонстрували, що ГА можна використовувати навіть у мистецтві – для формування естетично приємних зображень і складання гармонійних музичних композицій.

Метою дослідження цієї статті є комплексний аналіз сучасного стану царини практичного застосування еволюційних алгоритмів, а також визначення світових тенденцій і перспектив розвитку еволюційних технологій. Ми вважаємо, що особливу увагу слід приділити новим, неординарним сферам застосування генетичних алгоритмів, оскільки вони вказують на перспективність цього напрямку і можливість використання викладеного матеріалу у різних навчальних курсах.

#### **Різновиди еволюційних алгоритмів.**

Історія еволюційних обчислень починається з розробки різних незалежних моделей. Перші публікації «Симбіогенетичні еволюційні процеси, реалізовані штучними методами» («Symbiogenetic evolution processes realised by artificial methods» (1957)) та «Цифрова перевірка еволюційних теорій» («Numerical testing of evolution theories" (1962)), що належать Барічеллі Н.А., були спрямовані насамперед на розуміння механізмів природного феномену спадковості. Відомі піонери штучного моделювання еволюційних процесів і генетичних систем А.Фрейзер, Г.Бремерманн, Г.-Ф.Швефель, І.Рехенберг досліджували еволюційні стратегії. У 1966 році Л.Фогель, А.Оуенз та М.Волш запропонували і провели дослідження еволюції простих автоматів, які передбачали символи у числових послідовностях, заснувавши еволюційне програмування [1].

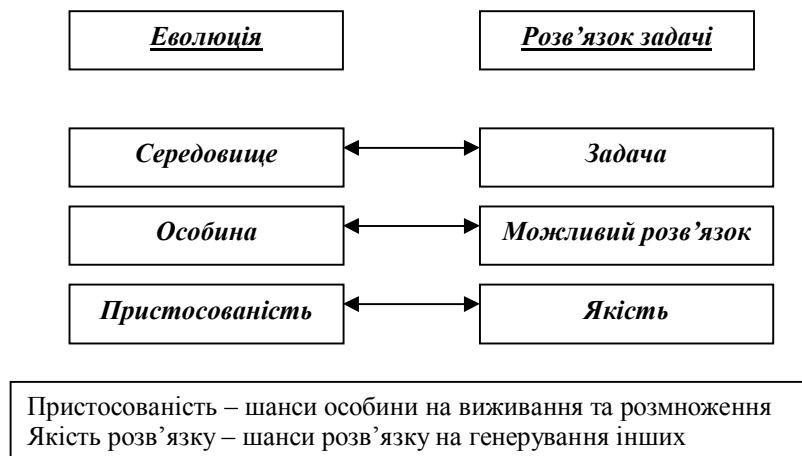


Рис. 1. Основна метафора еволюційних обчислень (за Ейбенем-Смітом)

Наразі назвою *еволюційні алгоритми* охоплюють усі обчислювальні і оптимізаційні методи, в основу яких покладена еволюційна метафора (див. рис. 1): еволюційні стратегії, еволюційне програмування, генетичні алгоритми, генетичне програмування. Ці методи відрізняються здебільшого представленнями можливих розв'язків задачі. В еволюційних стратегіях хромосоми представлені векторами дійсних чисел, в генетичних алгоритмах – векторами двійкових чисел, в еволюційному програмуванні – скінченними автоматами, а в генетичному програмуванні – деревами (списками), в перших роботах реалізованими виразами LISP. Оператори селекції еволюційних алгоритмів не залежать від представлення розв'язків, адже вони використовують лише значення функції пристосованості. Тому різниця між цими видами алгоритмів невелика, все залежить від найкращої стратегії для конкретної поставленої задачі. Під конкретну задачу підбирають відповідні представлення розв'язків задачі, а також для оперування ними – оператори мутації та схрещування.

Важливо зазначити, що різні види еволюційних алгоритмів (рис. 2) у чистому вигляді застосовують все рідше, а дедалі частіше

застосовують гібридні еволюційні алгоритми, а також еволюційні алгоритми з використанням інших підходів, які дозволяють пришвидшити розв'язання конкретної задачі і покращити отримувані результати. Це можуть бути і оператори інших алгоритмів, використовуваних для цієї задачі, і використання інформації та методів, специфічних для даної предметної області.

Правильно підібрані під задачу гібридні алгоритми, як правило, працюють на порядок швидше за відповідні їм генетичні, а також дозволяють отримати значно точніші результати. Однак, розширення генетичного алгоритму додатковими операторами не завжди сприяє зростанню його ефективності, а знаходження методів, які б покращували його результативність, порівнянне з мистецтвом.

Досить довгий час тривали пошуки алгоритму *all-purpose*, який би показував найкращий результат у порівнянні з іншими на будь-яких даних і на будь-якій задачі, однак зараз все частіше лунає думка про те, що такий єдиний найкращий алгоритм віднайти неможливо і доцільно додатково використовувати методи, розроблені під конкретну задачу.

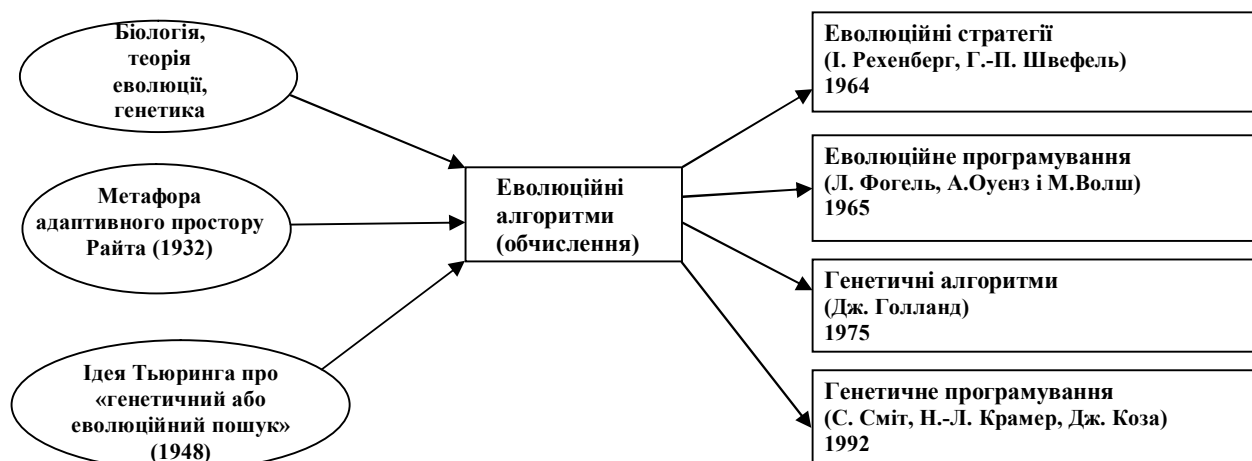


Рис. 2. Основні види еволюційних алгоритмів (за Ейбенем-Смітом)

*Еволюційне програмування* винайдене Лоуренсом Фогелем у 1960 – 1965 роках під час роботи над звітом перед Конгресом Сполучених Штатів про стан галузі штучного інтелекту і можливості інвестування в неї. Основними підходами у цій галузі на той час були моделювання роботи мозку (нейронні мережі) та моделювання розв’язання задач експертами. Л.Фогель помітив можливість ще одного, альтернативного підходу до проблем штучного інтелекту – моделювання не кінцевого результату еволюції, а моделювання самого процесу еволюції як засобу випрацювання розумної поведінки і можливостей передбачення явищ у середовищі.

Л.Фогель виконав низку експериментів, у яких скінченні автомати представляли собою особин у популяції вирішувачів задач. Ці скінченні автомати передбачали символи у цифрових послідовностях, які, еволюціонуючи, ставали все придатнішими до розв’язання поставленої задачі. Саме тоді за досліджуваною цариною закріпилася назва «еволюційне програмування».

Популяція скінченних автоматів потрапляє до експериментального середовища – отримує певну послідовність символів. Для кожного автомата-батька виконується процедура звірення кожного наступного символу з відповідним йому прогнозованим символом з виходу автомата і оцінюється значення функції втрат для цього виходу. Після останнього прогнозу обчислюється «життєздатність» такого автомата чи програми. Автомати-нащадки утворюються випадковою мутацією автоматів-батьків і теж оцінюються. Найкращі автомати відбираються до наступного покоління, і процес повторюється. Коли необхідний прогноз нових символів,

послугуються найкращим на цей час автоматом, а нове спостереження додається до старих [2].

Таким чином, еволюційне програмування, на відміну від генетичних алгоритмів, моделює еволюцію більш як процес пристосувальної поведінки особин популяції або виду, аніж процес адаптації генів.

Роботи Л.Фогеля та його учнів згодом поширили процеси еволюційного програмування від прогностичних задач на моделювання систем, розпізнавання образів, теорію ігор тощо. Сучасні вчені у царині еволюційного програмування займаються дослідженням еволюції програм, впливу різних параметрів середовища та різних видів мутації на результативність еволюційного програмування, розпізнаванням образів, тощо [1].

*Еволюційні стратегії* у багатьох аспектах подібні і до генетичних алгоритмів, і до еволюційного програмування, бо теж імітують процеси природної еволюції. Однак, вони мають суттєву відмінність на прикладному рівні. У той час як генетичні алгоритми просто створені для оптимізації дискретних або цілочисельних розв’язків задач, еволюційні стратегії застосовують для неперервних значень, які є типовішими для експериментальних задач у лабораторіях.

Відмінності еволюційних стратегій від інших оптимізаційних методів і еволюційних алгоритмів [3]: пошук від однієї популяції до іншої на протигагу від однієї особини до іншої; використання інформації про саму функцію, а не її похідних; використання імовірнісних методів при переході від популяції до популяції (а не детермінованих); кодування розв’язків векторами дійсних чисел; зосередження уваги на впливі генетичних операторів на зміни фенотипу;

адаптивний крок мутації – крок мутації еволюціонує разом з розв'язком, оскільки цей параметр «защитий» до хромосом; невелика різниця між батьками та нащадками в процесі схрещування (сильна взаємопов'язаність – маленькі зміни у перших відбиваються у маленьких змінах у останніх).

Технологія еволюційних стратегій була винайдена І.Рехенбергом, а згодом розвинута Г.-П.Швевелем та іншими вченими, які сформували теоретичну частину цієї царини.

Особливим поняттям і кульмінацією теорії є поняття «еволюційного вікна». Часом прогрес спостерігається тільки на дуже вузькому проміжку розміром із крок мутації. Таким чином, має існувати деяке правило, за яким еволюція регулює розмір кроку мутації, залежно від обставин. Знаходження оптимального кроку мутації саме по собі є складною і цікавою задачею.

Генетичне програмування було розроблене у Сполучених Штатах у 1990-их роках, серед перших дослідників – Дж.Коца. Особливістю цього виду еволюційних алгоритмів є використання нелінійних хромосом (графи, дерева – подання у вигляді списків LISP), а також (якщо популяції особливо великих розмірів) необов'язковість оператора мутації/схрещування. Здебільшого використовують або мутацію, або схрещування. Мутація полягає в заміні випадковим деревом деякого піддерева розв'язку, а схрещування – у обміні батьківських хромосом піддеревами. На відміну від лінійного представлення даних, представлення деревами дозволяє гнучкі зміни розв'язків, а також оперування хромосомами різного розміру

(деревами з різною глибиною чи шириною).

Оперування подібними структурами даних уможливило автоматичний комп'ютерний синтез фрагментів програм на основі навчання шляхом індуктивного виводу. Хромосоми, що фактично закодовують математичні вирази (функції, змінні та константи), автоматично генеруються за допомогою генетичних операторів. Початкова популяція створюється випадковим чином і складається з програм, які включають до свого складу елементи множини проблемно-орієнтованих елементарних функцій (function set: +, -, \*, %, sin, cos, log, or, and, for, do-until та ін.), а також спеціальних функцій з предметної області конкретної задачі, змінних та констант. Ці множини стають основою для еволюційного синтезу програми для розв'язку деякої поставленої задачі. Спочатку у генетичному програмуванні застосовувалась мова LISP, однак на сьогоднішній день популярні також такі мови програмування як C, Smalltalk, C++ тощо.

Як правило, генетичне програмування застосовують у машинному навчанні (прогнозування, класифікація тощо).

**Інші евристичні алгоритми, що використовують метафори природних процесів.** Останніми роками активно розвивається такий напрям наукових досліджень як *природні обчислення* (Natural Computing), який об'єднує математичні методи на основі принципів природних механізмів прийняття рішень [4]. До таких методів належать еволюційні алгоритми, імітація самоорганізації мурашиної колонії, оптимізаційні алгоритми «бджолиного рою» тощо.



Рис. 3. Метафора мурашиної колонії

Мурашина колонія [5] – це один із нових перспективних метаевристичних алгоритмів (різновид евристики рою), який імітує

колективну поведінку мурашок для досягнення певної цілі, спільної для всього мурашника. Ідея алгоритму мурашиної колонії, натхнена статтею

про колективну поведінку аргентинських мурах, вперше висловлена А.Колорні, М.Доріго і В.Маньєццо у 1991 – 1992 роках при розв'язанні задачі про оптимальні шляхи у графах відповідно до того, як мураха відшукує найкращий шлях між мурашником та деяким харчовим ресурсом (див. рис. 3). Так само вона застосовна до задачі комівояжера, задач оптимізації маршрутів вантажних перевезень, задачі розфарбування графів, оптимізації мережених графіків, задач календарного планування тощо [6].

Один-єдиний мураха має обмежені когнітивні можливості, але весь мурашник, діючи спільно в певному напрямку, може досить швидко і ефективно знайти оптимальний шлях. Відтоді ідея мурашиної колонії поширилася на значно більше коло числових задач, а також всередині цієї ідеї з'явилися численні відгалуження, які застосовують різні аспекти поведінки мурах.

Колонію мурах можна розглядати як багатоагентну систему, у якій кожен агент («мурашка») функціонує автономно за досить простими правилами. На противагу примітивній поведінці окремих агентів, поведінка всієї системи виявляється досить розумною.

Перший мураха (припустимо, що він бігає навколо мурашника випадковим чином), який знаходить ресурс харчування яким-небудь шляхом, повертається до мурашника, залишаючи на цьому шляху відповідний слід феромону, який приваблює інших мурах, що проходять повз, слідувати саме цим шляхом. Оптимальніший шлях скоро набуває більшої концентрації «запаху» феромонів, ніж інші, чим виявляється ще привабливішим. Поки цей шлях залишається найкращим, легкі сліди феромонів на інших шляхах поступово вивітрюються.

Розподіл феромонів по середовищу пересування мурах є своєрідною динамічно змінюваною глобальною пам'яттю всього мурашника. «Мурашка» на деякому вузлі графа має прийняти рішення, яким шляхом далі слідувати, при цьому він користується параметрами «привабливості» та «сліду феромонів» на тій чи іншій ланці. Будь-який мурашка у фіксований момент часу може зчитувати або вносити зміни лише до єдиної комірки глобальної пам'яті.

Алгоритм «бджолиного рою» [7] – ще один різновид евристик рою, відносно молодий метод знаходження глобального екстремуму (максимуму або мінімуму) складної багатовимірної функції (у термінах алгоритму –

цільової функції). Як видно з назви, цей алгоритм використовує особливості поведінки рою бджіл-розвідників під час пошуку нектару.

Припустимо, що бджоли живуть у багатовимірному просторі, де кожна координата означає один параметр функції, яку необхідно оптимізувати. Нехай глобальний екстремум функції – це місце, де зосереджено найбільше нектару, причому таке місце у довкіллі вулика – єдине, а на інших галявинах нектару однозначно менше. Знайдена кількість нектару – це значення цільової функції у даній точці простору. Деякий розвіданий розв'язок представлений бджолою, яка зберігає параметри місця, де можна добути нектар. Нові бджоли знаходять ділянки простору, де значення цільової функції оптимальні, причому окіл випадкового пошуку наступними бджолами від вже розвіданих точок з найкращими показниками цільової функції динамічно зменшується (тільки не можна зменшувати окіл занадто швидко, бо є шанси застрягнути на локальному екстремумі). Найуспішніша бджола запам'ятовується як проміжний розв'язок, і процес повторюється, поки не буде досягнутий відповідний критерій зупинки.

Цей тип алгоритмів вже зарекомендував себе у численних застосуваннях, наприклад: навчання нейронних мереж розпізнавання образів, оптимізація ваг у перцептронах, формування розкладу робіт на виробництві, кластеризація даних, багатоцільова оптимізація тощо.

Епідемічні алгоритми [8] також використовують природну метафору, використовуючи відносно прості правила поширення інформації, потрібної іншим елементам, у гетерогенному некогерентному середовищі, де для елемента можливе лише вузьке, локальне бачення. Для деякого елемента середовища визначаються правила, за якими він може «народитися», «вмерти від самотності», «вмерти від перенаселення» або «вижити» і перейти у наступне покоління. Елемент може набувати стану «чутливості» (susceptible: не знати, але бути готовим до отримання інформації), «інфікованості» (infective: знати і розповсюджувати інформацію) або «зцілення» (removed/recovered: знати, але вже не розповсюджувати інформацію).

Батьком сучасної теорії генетичних алгоритмів вважається Дж.Г.Голланд. Довгий час він займався здебільшого здатністю природних систем до адаптації і мріяв про створення системи, здатної адаптуватися до будь-яких умов

довкілля. Голланд побачив наявність зв'язку між біологією і обчислювальними процесами. В процесі еволюції генетичний матеріал «навчається» краще пристосовуватися до змін у навколишньому середовищі. Результатом такого «навчання» протягом численних поколінь є все живе на планеті, у всій повноті краси і життєздатності. Якщо еволюція так добре працює на живих істотах, чому б не застосувати її і до комп'ютерних обчислень?

Голланд розпочав роботу над алгоритмами, які оперували послідовностями двійкових чисел, які за аналогією до біології отримали назву хромосом. Його алгоритми виконували операції над цими хромосомами, імітуючи процеси природного відбору і еволюції. В 1975 році Дж. Голланд опублікував, мабуть, найвідомішу свою працю «Адаптація у природних і штучних системах» («Adaptation in Natural and Artificial Systems»). Після цієї публікації інтерес до новоствореної галузі безупинно зростає. Якщо у 1982 році було всього 15 публікацій на цю тему, то у 1992 році – вже понад 500. У книзі вводиться власне ідея генетичного алгоритму («репродуктивний план Голланда» [9]) і пропонується схема такого алгоритму, який згодом назвали класичним або канонічним генетичним алгоритмом (canonical GA).

Згодом поняття генетичних алгоритмів значно розширилося, і до них належать і суттєво відмінні від канонічного алгоритми. Наприклад, у техніці генетичного програмування шукані розв'язки мають форму комп'ютерних програм, а оцінка життєздатності полягає у визначенні, чи здатна програма розв'язати деяку обчислювальну задачу. В еволюційному програмуванні, навпаки, структура комп'ютерної програми фіксована, а еволюціонують параметри. У методі еволюційної стратегії замість двійкових рядків застосовують вектори дійсних чисел, а також часто використовуються самопристосовні показники риня мутації.

Учні Голланда Кеннет Де Йонг та Девід Голдберг продовжили його дослідження і зробили чималий внесок в розвиток генетичних алгоритмів. Найвідомішим доробком Голдберга є книга «Генетичні алгоритми в оптимізації пошуку та машинному навчанні» («Genetic algorithms in search optimization and machine learning» (1989)), одна з найбільш цитованих у галузі комп'ютерних наук. Завдяки цим вченим описано всі генетичні оператори і досліджено поведінку алгоритмів на групі тестових функцій,

і саме алгоритм Голдберга отримав назву «генетичний алгоритм».

Від традиційних методів оптимізації генетичні алгоритми відрізняються тим, що [10]: мають справу не зі значеннями параметрів задачі, а з їх закодованою формою (кодування параметрів); здійснюють пошук рішення не з єдиної вихідної точки, а відштовхуючись від деякої популяції (операції на популяції); використовують лише цільову функцію, а не її похідні чи іншу допоміжну інформацію (використання мінімальної інформації про задачу – не потрібна інформація про поведінку функції); застосовують ймовірнісні, а не детерміновані правила вибору (рандомізація операцій); застосовні до задач, які раніше розв'язувалися лише повним комбінаторним перебором, або лише за допомогою нейронних мереж, а також до раніше нерозв'язних задач.

**Сучасні застосування генетичних алгоритмів.** Генетичні алгоритми дозволяють працювати з широким колом теоретичних і прикладних задач. За допомогою них відшукується баланс, «золота середина» між ефективністю роботи алгоритму та якістю отримуваних розв'язків за рахунок виживання «найпристосованіших», що дозволяє досліджувати і розв'язувати задачі навіть в умовах невизначеності і з нечіткими даними.

Генетичні алгоритми застосовуються при розв'язанні задач, які полягають в оптимізації деякої функції від великої кількості параметрів. Такі функції можуть мати значну кількість екстремумів, чим унеможливають їх розв'язання іншими традиційними методами.

Генетичні алгоритми застосовують також в системах підтримки прийняття рішень, адже вони дозволяють приймати рішення як в умовах визначеності, так і в умовах невизначеності та з нечіткими даними. Нерідко генетичні алгоритми використовують у подібних задачах з нейронними мережами. Процеси глобалізації і посилення зв'язків між колись майже незалежними процесами у світі викликали потребу в створенні систем, які б здійснювали аналіз досліджуваного середовища, допомагали прийняти відповідне до поточного стану середовища рішення, а також виявляли деякі закономірності і тенденції змін та здійснювали відносний прогноз на найближче майбутнє.

У класифікації алгоритмів комбінаторних обчислень виділяють точні і наближені методи. Перші гарантовано знаходять точний оптимальний розв'язок задачі. Однак, майже всі

принципово важливі задачі є NP-повними, і для них не існує жодного точного алгоритму з поліноміальною складністю. Також є задачі з недоліками і помилками у даних, для яких невиправдані і неприпустимі значні часові витрати, оскільки потрібно часто здійснювати корекцію даних і запускати алгоритм повторно. На противагу точним методам, сучасні наближені методи, до яких відносяться і еволюційні алгоритми, дозволяють за раціональний (а також наперед визначений) час отримувати розв'язки «високої якості», дуже близькі до оптимальних і тому цілком прийнятні з практичної точки зору [11].

Генетичні алгоритми з успіхом застосовують, коли необхідно скомпонувати (розмістити на шматі матеріалу, нерідко обмеженого за розміром) задану множину елементів (інколи навіть враховуючи зв'язки між ними), мінімізуючи витрати на площу (задача крою, дизайн інтер'єрів, планування міста чи ландшафту тощо). У задачі крою одягу може також бути врахована інформація про напрям нитки у деталях. Таким чином, генетичні алгоритми з успіхом застосовують в конструктивному дизайні і створенні креслень майбутніх деталей.

Особливо популярним є використання генетичних алгоритмів у нейронних мережах, навчанні перцептронів та вирішенні складних задач штучного інтелекту [12]. При цьому використовується така властивість генетичних алгоритмів як слабкий зв'язок з конкретною структурою задачі (можливість використання одного і того ж алгоритму для різних задач) – ГА застосовні до різних нейромережових парадигм.

За допомогою стратегії на основі генетичного алгоритму можна також оптимізувати запити до баз даних, при цьому ми уникаємо комбінаторного перебору усіх таблиць і заощаджуємо час та ресурси пам'яті. У такому ГА таблиці, які розглядаються запитом, кодуються у вигляді сукупності хромосом, над якими протягом скінченної кількості ітерацій здійснюються генетичні операції. В результаті вибірки залишаються тільки такі хромосоми, які мають менше, у порівнянні з попередньою ітерацією, значення функції вартості. Найбільш оптимальний розв'язок стає результатом запиту. Базы даних невпинно зростають, тому оптимізація запитів на основі великої кількості таблиць набуває все більшої актуальності [13].

Останнім часом все більшої популярності набувають паралельні генетичні алгоритми, тому

вчені замислюються над проблемами комунікації і безпеки в розпаралеленому ГА.

**Висновки.** З наведених вище задач добре видно, що генетичні алгоритми є потужним допоміжним обчислювальним засобом у математичних дослідженнях та в теорії прийняття рішень і керування. Особливу цінність ГА становлять у обчисленнях, які не можна провести іншими традиційними методами і для яких необхідно застосовувати евристичні та наближені методи. Генетичні алгоритми успішно застосовуються при розв'язанні математичних задач, які полягають в оптимізації деякої функції від великої кількості параметрів. Таким чином, за їх допомогою можна практично досліджувати залежність поведінки функції від динаміки параметрів. Більше того, еволюційні алгоритми здійснюють пошук над дійсними числами і дозволяють отримувати прийнятні для практичного застосування рішення за скінченний і розумний час. Можливість кодування як цілих чисел, так і чисел з плаваючою крапкою і дробів за допомогою кодів Грея розширює дослідникам простір роботи [1].

За кілька останніх років необхідність у використанні ГА у математичних розрахунках настільки посилилася, що виникла потреба в імплементації відповідного модуля у середовищі MATLAB 7. Тепер можна застосовувати оптимізацію за допомогою еволюційного процесу або викликати ГА як функцію з командного рядка.

Досліджуючи деяке вузьке застосування генетичних алгоритмів, вчені водночас наводять власні результати, що стосуються самого генетичного алгоритму – динаміки процесу еволюції, самоадаптації алгоритму, залежності ефективності операторів і збіжності алгоритму від значень заданих параметрів тощо. Наприклад, Брітта Кехден та Франк Нойманн досліджують представлення та функціонування стандартних операторів мутації та схрещування у термінах реляційної алгебри [14]. Дослідники з Великобританії Михайло Семенов та Дмитро Теркель вивчають збіжність генетичного алгоритму за допомогою стохастичних функцій Ляпунова [15]. Математичний критерій збіжності еволюційних алгоритмів досліджений також у роботі Марчіна Радванського [16]. Процеси динаміки параметрів генетичного алгоритму та адаптації параметрів детально розглянуті російськими вченими [17].

Колін Г. Джонсон (Colin G. Johnson) у своїй роботі «Використання методу табу та генетичних



алгоритмів у дослідженнях математики (Using Tabu Search and Genetic Algorithms in Mathematics Research) звертає увагу на можливість застосування генетичних алгоритмів, самостійно або у поєднанні з іншими евристичними методами (наприклад, методом табу), у пошуку прикладів та контрприкладів до гіпотез, знаходженні послідовностей перетворень одного об'єкта на інший, нумерації класів еквівалентності тощо. У якості прикладів наводяться задачі про топології: представлення вузлів математичної топології, визначення, чи є конкретна структура вузлом тощо [18]. Генетичні алгоритми успішно застосовуються разом з нечіткими системами [13].

Наведемо інші приклади популярних досліджень і задач, розв'язаних за допомогою генетичних алгоритмів (список постійно доповнюється новими напрямками роботи): задачі

лінійного та нелінійного програмування [19, 20]; дослідження скінченних алгебр, нечітких алгебр [21], задачі про точки і дерева Штейнера [22, 23]; задачі на графах [24, 25].

Застосування генетичних алгоритмів у системах штучного інтелекту та нейронних мережах має суттєве значення, тому це питання розглядається окремо від інших застосувань у комп'ютерних науках.

Особливостями застосування ГА у системах III є реактивність систем і динамічна оптимізація, уможливлення навчання системи і прогнозування поведінки досліджуваного середовища, а відтак – визначення оптимальних способів поведінки у такому середовищі. Прикладами таких застосувань можуть бути системи прогнозування банкрутства [26], системи допомоги у творчому пошуку [27] та багато інших.

#### Список використаних джерел

1. **Eiben A. E.** Introduction to Evolutionary Computing / *Eiben A. E., Smith J. E.* // Natural Computing Series. – Springer. – 2007. Available at: <http://www.cs.vu.nl/~gusz/ecbook/ecbook.html> (Eng.)
2. **Fogel J.** A Retrospective View and Outlook on Evolutionary Algorithms / *Fogel J., Lawrence J. A.* // Proc. of the Int. Conf. on Computational Intelligence Theory and Applications: Lecture Notes In Computer Science. – 1997. – Vol. 1226. – P. 337–342. (Eng.)
3. **Kureichik V. M.** Evolutionary Computations: Genetic and Evolutionary Programming / *Kureichik V. M., Rodzin S.I.* // Novosti Iskusstvennogo Intellecta. – 2003. – №5(59). — P. 13–20. – RAI, Moskva. — Available at: [http://www.raai.org/library/ainews/2003/5/kureichik\\_rodzin.doc](http://www.raai.org/library/ainews/2003/5/kureichik_rodzin.doc) (Rus.)
4. **Glybovets M.** On Necessity of Medical, Biological and Computer Science Research Integration / *Glybovets M., Gorokhovskiy S., Gulayeva N., Kyriienko O.* // Naukovi Zapysky NaUKMA. – 2007. – Vol. 73: Komputerni nauki. – P. 5–9. Available at: [http://aleph.ukma.kiev.ua/e-lib/NZ/NZV73\\_2007\\_computer/01\\_glybovets\\_mm.pdf](http://aleph.ukma.kiev.ua/e-lib/NZ/NZV73_2007_computer/01_glybovets_mm.pdf) (Ukr.)
5. **Maniezzo V.** Ant Colony Optimization / *Maniezzo V., Gambardella L. M., de Luigi F.* // New Optimization Techniques in Engineering, eds. Onwubolu G. C., Babu, B. V. – Springer-Verlag, Berlin Heidelberg. – 2004. – P. 101–117. Available at: <http://www.idsia.ch/~luca/aco2004.pdf> (Eng.)
6. **Shtovba S.** Ant Algorithms / *Shtovba S.* // Exponenta Pro. Matematika v prilozenijah. – SoftLine. – 2003. – №4. — С. 70-75. Available at: <http://soft.mail.ru/journal/pdfversions/32150.pdf> (Rus.)
7. **Pham D. T.** The Bees Algorithm – A Novel Tool for Complex Optimisation Problems / *Pham D. T., Ghanbarzadeh A., Koç E., Otri S., Rahim S., Zaidi M.* // Intelligent Production Machines and Systems. – Cardiff University, Elsevier Ltd. – 2006. – P. 454–459. Available at: [http://conference.iproms.org/the\\_bees\\_algorithm\\_a\\_novel\\_tool\\_for\\_complex\\_optimisation\\_problems](http://conference.iproms.org/the_bees_algorithm_a_novel_tool_for_complex_optimisation_problems) (Eng.)
8. **Hollerung T. D.** Algorithms of the Internet' presentation. Epidemic Algorithms / *Hollerung T. D., Bleckmann P.* // – Universität Paderborn. – 2004. Available at: <http://www.wcs.uni-paderborn.de/cs/ag-madh/WWW/Teaching/2004SS/AlgInternet/Submissions/09-Epidemic-Algorithms.pdf> (Eng.)
9. **Panchenko T. V.** Genetic Algorithms: teaching aid / eds. Tarasevich Y. Y. — Astrahan: Astrahanskiy universitet. – 2007. Available at: <http://mathmod.aspu.ru/images/File/ebooks/GAfinal.pdf> (Rus.)
10. **Rutkowska D.** Neural Networks, Genetic Algorithms and Fuzzy Systems / *Rutkowska D., Pilinski M., Rutkowski L.* // Moskva: Goriachaja linija. Telekom. – 2004. – 452p. (Rus.)

11. **Sirenko S.** On Classification of approximate methods of combinatorial optimization / *Sirenko S.* // Int. Book Series "Information Science and Computing". – Vol. "Artificial Intelligence and Decision Making". – 2007. Available at: [http://www.foibg.com/ibs\\_isc/ibs-07/IBS-07-p25.pdf](http://www.foibg.com/ibs_isc/ibs-07/IBS-07-p25.pdf) (Rus.)
12. **Subbotin S. O.** Non-iterative, evolutionary and multi-agent methods of synthesis of fuzzy-logic and neural-network models: monograph / eds. *Subbotin S. O., Olijnyk A. O., Olijnyk O. O.* // Zaporozhye: ZNTU. – 2009. – 375p. (Ukr.)
13. **Bennett K. A.** Genetic Algorithm for Database Query Optimization / *Bennett K., Ferris M. C., Ioannidis Y. E.* // Proc. of the 4th Int. Conf. on Genetic Algorithms. – 1991. Available at: <ftp://ftp.cs.wisc.edu/tech-reports/reports/91/tr1004.ps> (Eng.)
14. **Kehden B.** Relation-Algebraic View on Evolutionary Algorithms for Some Graph Problems / *Kehden B., Neumann F.* // Proc. of the EvoCOP'06. – 2006. – P. 147-158. (Eng.)
15. **Semenov M. A.** Analysis of Convergence of an Evolutionary Algorithm with Self-Adaptation using a Stochastic Lyapunov function / *Semenov M. A., Terkel D. A.* // – Evolutionary Computation. – 2003. – Vol. 11, № 4. – P. 363-379. (Eng.)
16. **Radwanski M.** Convergence of nonautonomous evolutionary algorithm / *Radwanski M.* // Universitatis Iagellonicae Acta Mathematica. – 2007. Available at: <http://www.emis.de/journals/UIAM/PDF/45-197-206.pdf> (Eng.)
17. **Komartsova L. G.** Adapting the parameters of genetic algorithms to find solutions / *Komartsova L. G., Golubin A. V.* // Sbornik nauchnyh trudov. Nauchnaja sessija MIFI. – Moskva: MIFI. – 2003. – Vol. 3. – P. 124-125 (Rus.)
18. **Johnson C. G.** Using Tabu Search and Genetic Algorithms in Mathematics Research. – 2004. Available at: [http://kar.kent.ac.uk/14050/1/Using\\_Tabu\\_Search\\_and\\_Genetic\\_Algorithms\\_in\\_Mathematics.pdf](http://kar.kent.ac.uk/14050/1/Using_Tabu_Search_and_Genetic_Algorithms_in_Mathematics.pdf) (Eng.)
19. **Sarimveis H.** Solving Mixed Integer and Nonlinear Programming Problems by Genetic Algorithms / *Sarimveis H., Alexandridis A., Nikolakopoulos A., Spyropoulos A.* // THALES Project No. 1179. – 2005. Available at: [http://www.ntua.gr/eese/Vasikh\\_Ereyna/Thalis/Thalis\\_projects\\_English\\_summaries/Sarimveis.pdf](http://www.ntua.gr/eese/Vasikh_Ereyna/Thalis/Thalis_projects_English_summaries/Sarimveis.pdf) (Eng.)
20. **Wang Y.** An evolutionary algorithm for solving nonlinear bilevel programming based on a new constraint-handling scheme / *Wang Y., Jiao Y.-C., Li H.* // IEEE Trans. Syst. Man Cybern.: Applications and Reviews. – 2005. – Vol. 35, № 2. – P. 221-232. (Eng.)
21. **Spector L.** Genetic Programming for Finite Algebras / *Spector L., Clark D. M., Lindsay I., Barr B., Klein J.* // Proc. of the GECCO-2008. – ACM Press. – 2008. Available at: <http://www.genetic-programming.org/hc/2008/03-Spector/Spector-PAPER.pdf> (Eng.)
22. **Yang B.** A Hybrid Evolutionary Algorithm for the Euclidean Steiner Tree Problem Using Local Searches / *Yang B.* // KES'06 Proc. of the 10th Int. Conf. on Knowledge-Based Intelligent Information and Engineering Systems. – 2006. (Eng.)
23. **Kalashnikov R. S.** Steiner tree construction by the method of genetic search / *Kalashnikov R. S.* // Perspektivnije informacionnije tehnologii i intelektualnije systemy. – 2005. – № 2 (22). (Rus.)
24. **Greiner G.** Single- and multi-objective evolutionary algorithms for graph bisectioning / *Greiner G.* // FOGA '09 Proc. of the 10th ACM SIGEVO workshop on Foundations of GA. – 2009. (Eng.)
25. **Kureichik V. V.** Application of genetic algorithms to solve graph optimization problems / *Kureichik V. V., Stasenko L. A.* // Perspektivnije informacionnije tehnologii i intelektualnije systemy. – 2002. – № 3 (11). (Rus.)
26. **Min S.-H.** Hybrid genetic algorithms and support vector machines for bankruptcy prediction / *Min S.-H., Lee J., Han I.* // Expert Systems with Applications. – 2006. – Vol. 31 (3). – P. 652-660. Available at: <http://www.deepdyve.com/lp/elsevier/hybrid-genetic-algorithms-and-support-vector-machines-for-bankruptcy-uj4ha91K1L> (Eng.)
27. **Yang H.-L.** An Intelligent System for Supporting Personal Creativity Based on Genetic Algorithm / *Yang H.-L., Lee C.-H.* // PRICAI'06 Proc. of the 9th Pacific Rim Int. Conf. on Art. Intelligence. – 2006. (Eng.)

Надійшла до редколегії 28.02.2013