

ЕВОЛЮЦІЙНЕ ПРОГРАМУВАННЯ

Проведено комплексний аналіз стану царини практичного застосування сучасного еволюційного програмування, а також визначення світових тенденцій і перспектив розвитку еволюційних технологій.

Вступ

Природа завжди вражала своєю складністю і водночас не випадковою взаємопов'язаністю. Зовсім не дивує той факт, що дослідники математичних та комп'ютерних задач у пошуках натхнення звернулися до теорії еволюції. Можливість функціонування обчислювальної системи з простими механізмами мінливості та відбору по аналогії з законами еволюції у природних системах була досить привабливою. На відміну від природного процесу, обчислювальна система на основі еволюційного алгоритму, як правило, створена для отримання можливого і прийняттого розв'язку деякої конкретної задачі (або неодноразового отримання таких розв'язків) за скінченний проміжок часу.

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.

Дослідження даної теми є безперечно актуальним, адже розробка, аналіз та застосування ефективних і універсальних методів розв'язку задач (алгоритмів) – це ключова задача комп'ютерних наук, а тема еволюційних алгоритмів ще не достатньо досліджена. Еволюційні алгоритми надають можливість швидкої генерації прийнятних розв'язків задач, які неможливо

розв'язати іншими аналітичними методами, уникаючи повного перебору і значно скорочуючи часові витрати.

Метою цього дослідження є комплексний розгляд стану царини практичного застосування сучасного еволюційного програмування, а також визначення світових тенденцій і перспектив розвитку еволюційних технологій. Ми вважаємо, що особливу увагу слід приділити новим, неординарним сферам застосування еволюційних алгоритмів, оскільки вони вказують на перспективність цього напрямку.

1. Еволюційне програмування

Історія еволюційних обчислень починається з розробки різних незалежних моделей.

Еволюційне програмування (ЕП) винайдене Лоуренсом Фогелем у 1960 – 1965 роках під час роботи над звітом перед Конгресом Сполучених Штатів «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. Фогель помітив можливість ще одного, альтернативного підходу до проблем штучного інтелекту – не моделювання 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, потрапивши до експериментального середовища, отримує певну послідовність символів. Для кожного автомата-батька виконується процедура зіставлення кожного наступного символу з відповідним прогнозованим символом із виходу автомата та оцінюється значення функції втрат для цього виходу. Закінчивши прогнозування, визначається «життєздатність» такого автомата (програми). Автоматнащадки утворюються випадковою мутацією автоматів-батьків і теж оцінюються. Найкращі автомати відбираються до наступного покоління, і процес повторюється. За потреби прогнозування нових символів використовують найкращий на цей час автомат, а нове спостереження додається до старих [1].

Тому можна сказати, що ЕП, на відміну від генетичних алгоритмів, моделює еволюцію більш як процес пристосувальної поведінки особин популяції або виду, аніж процес адаптації генів.

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. Водночас генетичні алгоритми просто створені для оптимізації дискретних або цілочисельних розв'язків задач, еволюційні стратегії застосовують для неперервних значень, які типовіші для експериментальних задач у лабораторіях.

Спочатку ЕП було запропоновано як процедура породження машинного інтелекту. Інтелект при цьому розглядався у контексті адаптивної поведінки, а саме як здатність системи адаптувати в широкому діапазоні середовищ свою поведінку для досягнення поставленої мети; можливі розв'язки заданої проблеми представлялись скінченними автоматами [2].

В кінці 1980-х рр. ЕП було розширене для використання в задачах числової оптимізації; зокрема, були запропоновані альтернативні представлення розв'язків, включаючи дійсні вектори для оптимізації

неперервних функцій та упорядковані списки (перестановки) для задачі комівояжера [3]. Фактично, початкова організація алгоритму еволюційного програмування, який оперував скінченними автоматами, була розширена до використання довільного способу кодування, різних операторів мутації та селекції, а також до техніки самоадаптації.

Наразі прийнято говорити про дві гілки ЕП – традиційне еволюційне програмування та сучасне еволюційне програмування.

Принциповим аспектом, який відрізняє ЕП від генетичного алгоритму, є те, що ЕП представляє фенотипічний, а не генотипічний підхід до моделювання еволюції: пошук розв'язків йде на рівні фенотипу, а біологічна еволюція розглядається як процес пристосування до навколишнього середовища в першу чергу на рівні поведінки. Центральним об'єктом еволюції в ЕП є популяція. Такий рівень абстракції у природі не передбачає рекомбінації, тому в ЕП немає оператора кросинговеру, а єдиним оператором пошуку альтернативних розв'язків є мутація.

В цілому сьогодні ЕП розглядається як відкрита структура з точки зору представлення розв'язків та операторів мутації. Фундаментальним положенням тут є те, що представлення не повинно фіксуватись задалегідь, а визначається з поставленої проблеми. Аналогічно, оператори мутації розробляються у відповідності з обраним способом кодування розв'язків; ключовим моментом у конструюванні операторів мутації є те, що вони мають забезпечувати тісний зв'язок між батьком та його нащадком.

Наразі ЕП найчастіше використовується для оптимізації неперервних функцій. Тому спочатку розглянемо сучасне еволюційне програмування (СЕП), а ознайомитися з його історично першою формою можна в роботі [4].

2. Сучасне еволюційне програмування

Сучасне еволюційне програмування (Contemporary Evolutionary Programming), як вже згадувалось, розроблене для задач

числової оптимізації, зокрема, для оптимізації дійсних функцій (оптимізації дійсних параметрів – continuous parameter optimization): необхідно знайти значення аргументів, на яких досягається мінімум функції

$$F(\vec{X}) = F(x_1, x_2, \dots, x_n) : R^n \rightarrow R$$

(аргументи та значення функції є дійсними). В загальному випадку припускається існування обмеженого підпростору:

$$X^n = [u_1, v_1] \times [u_2, v_2] \times \dots \times [u_n, v_n] \subseteq R^n,$$

такого що $u_i \leq x_i \leq v_i, i = 1, \dots, n$. Насправді ці обмеження мають місце тільки на етапі ініціалізації, при встановленні початкової популяції; оператори не перевіряють виконання вказаної умови. Таким чином, простір пошуку в принципі є необмеженим: $A_x = R^n$. Обмежень на вид цільової функції немає.

2.1. Стандартна форма алгоритму сучасного еволюційного програмування. Стандартна форма алгоритму (Standard EP) ґрунтується на припущенні, що $\min(F(x_1, x_2, \dots, x_n)) = 0$.

Схема алгоритму.

0. Кодування розв'язків.
1. Ініціалізація.
2. Оцінювання.
3. Моделювання еволюційного процесу (репродукція):
 - 3.1. застосування генетичних операторів;
 - 3.2. селекція (оцінювання та відбір або перехід на п. 4);
 - 3.3. перехід на п. 3.1.
4. Завершення роботи.

Розглянемо кожен крок детальніше.

Кодування розв'язків. Одній особині A відповідає вектор дійсних змінних:

$$A = (\vec{X}) = (x_1, x_2, \dots, x_n) \in R^n.$$

Ініціалізація. Випадковим чином генерується початкова популяція, що складається з N особин $A_i, i = 1, 2, \dots, N$; рекомендоване значення розміру популяції $N \geq 200$ ([5]). Значення елемента $x_{ij}, j = 1, 2, \dots, n$, кожної особини $A_i = (x_{i1}, x_{i2}, \dots, x_{in}), i = 1, 2, \dots, N$, встановлюється випадково за допомогою рівномірного розподілу на інтервалі $[u_j, v_j] \subset R, u_j < v_j$. Границі цього інтервалу мають значення лише на етапі ініціалізації; у подальшій еволюції простір пошуку в принципі є необмеженим. Якщо границі інтервалів не обумовлені задачею, то рекомендовані значення параметрів є такими: $u_j = -50, v_j = 50, j = 1, 2, \dots, n$ ([5]).

Оцінювання. Функція пристосованості $\Phi(A)$ зазвичай збігається з цільовою функцією:

$$\Phi(A) = F(\vec{X}).$$

В загальному випадку значення функції пристосованості змінюється за допомогою випадкової величини ζ_i – елемента збурення (random noise term), після чого масштабується до додатного значення. Тобто, в загальному випадку функція здоров'я набуває вигляду:

$$\Phi(A) = \Omega(F(\vec{X}), \zeta),$$

де $\Omega : R \times \Theta \rightarrow R_+$ – функція масштабування (scaling function),

Θ – додаткова множина параметрів, включених у процес.

Застосування генетичних операторів. Єдиним оператором в еволюційному програмуванні є оператор мутації, який застосовується наступним чином:

- кожному особину $A_i, i = 1, \dots, N$, копіюють для застосування оператора мутації;
- мутація копії кожної особини A_i здійснюється шляхом додавання нормально розподіленої випадкової величини з нульовим математичним сподіванням та середньоквадратичним відхиленням, що змінюється динамічно (вираховується для кожної компоненти вектора \vec{X}_i як квадратний корінь лінійного перетворення функції здоров'я). Тобто, значення змінних $x'_{ij}, j = 1, 2, \dots, n$, нащадка A_i' особини A_i оброблюються за формулою

$$x'_{ij} = x_{ij} + \sqrt{\beta_j \cdot \Phi(A_i) + \gamma_j} \cdot N_j(0,1), \quad (1)$$

де $\Phi(A_i)$ – значення функції пристосованості батька, β_j – константа масштабуван-

ня або константа пропорційності (proportionality constant), γ_j – дисперсія або значення зсуву (offset value), $N_j(0,1)$ – стандартна нормально розподілена випадкова величина.

Як видно з наведених формул, «ширина мутації» залежить від значення функції пристосованості батька. Ідея полягає у підвищенні ефективності процесу оптимізації шляхом скорочення «ширини мутації» при наближенні до оптимуму (нагадаємо, що в стандартній формі алгоритму глобальний оптимум вважається відомим та таким, що дорівнює 0).

Константи β_j , γ_j ($j=1,2,\dots,n$) є параметрами алгоритму та встановлюються користувачем (всього $2n$ параметрів). Підбір значень параметрів створює додаткові труднощі при використанні алгоритму.

На практиці рекомендують покласти $\beta_j=1$, $\gamma_j=0$ ($j=1,2,\dots,n$). Тоді формула (1) набуває вигляду

$$x'_{ij} = x_{ij} + \sqrt{\Phi(A_i)} \cdot N_j(0,1).$$

Однак необхідно зазначити, що вказані значення параметрів були досліджені для випадку задач малої розмірності. Водночас існує суттєва чутливість процесу пошуку до розміру кроку мутації у випадку задач великої розмірності. Тому при $n > 2$ в [5] рекомендується використовувати значення $\beta_j = -\frac{1}{n^2}$;

- кожен нащадок A_i' оцінюється (за допомогою функції пристосованості $\Phi(A_i')$) та додається в популяцію, причому i -ий нащадок додається під номером $N+i$.

Легко бачити, що після генерації всіх нащадків розмір популяції збільшується вдвічі: популяція складається з N батьків та N нащадків.

Відбір. Перед початком відбору популяція містить $2N$ особин. Для відбору серед них N особин в наступне покоління влаштовуються змагання. Для визначення кількості особин, з якими має змагатись кожна особина з об'єднаної популяції ба-

тьків та дітей, вводиться додатковий параметр $h \geq 1$ ($h \in N$).

Змагання особин проводяться наступним чином. Кожна особина A_i ($i=1,\dots,2N$) попарно порівнюється з h іншими особинами, обраними випадково за законом рівномірного розподілу з об'єднаної популяції батьків та дітей. Особина A_i перемагає особину A_j ($j \in \{m_1, m_2, \dots, m_h\}$, $m_k \in \{1, 2, \dots, 2N\}$, $k=1,\dots,h$), якщо її функція пристосованості є не гіршою, ніж у супротивника: $\Phi(A_i) \leq \Phi(A_j)$. Далі, для кожної особини A_i вираховується оцінка W_i – кількість її перемог:

$$W_i = \sum_{j=1}^h res_j,$$

$$де \ res_j = \begin{cases} 1, & \text{якщо } \Phi(A_i) \leq \Phi(A_j); \\ 0, & \text{інакше} \end{cases};$$

очевидно, $W_i \in \{0,1,\dots,h\}$. Після цього всі особини сортуються, за зменшенням кількості їхніх перемог (за спаданням значення оцінки W_i), і найкращі особини утворюють нову популяцію розміром N . У випадку однакової кількості перемог, перевага надається особині з кращим значенням функції пристосованості.

Значення параметра h встановлюється користувачем. Як правило, параметр h набуває значення від $h=0.05 \cdot N$ до $h=0.01 \cdot N$, тобто від 1% до 5% від загальної кількості особин в популяції. При $N \geq 200$ в [5] рекомендується брати $h=10$.

Зауважимо, що з ростом h відбір набуває дискримінаційного елітарного характеру. Неважко також побачити, що проведена в описаний спосіб селекція може інтерпретуватись як ймовірнісна $(N+N)$ -селекція еволюційної стратегії.

Тут доречно було б навести головні аргументи дослідників ЕП, які наполягають на ймовірнісній селекції [5]. Перш за все вони звертають увагу на те, що селекція в природі є ймовірнісною. Крім того, існують емпіричні факти, згідно з якими «м'який» механізм селекції, що присвоює ненульову ймовірність відбору навіть най-

гіршим індивідуумам, дозволяє еволюційним алгоритмам вийти з локального оптимуму [6].

Завершення роботи. Умовою зупинки алгоритму ЕП може бути виконання однієї з умов: досягнення заданого числа поколінь t_{\max} ; досягнення заданого рівня якості; досягнення заданого рівня збіжності (особини в популяції стають настільки схожими між собою, що подальше покращення відбувається надто повільно).

ЕП, як було сказано вище, не передбачає оператора рекомбінації, повністю покладаючись на силу мутації. Аргументація такого підходу є наступною.

Трактування еволюції у першу чергу як механізму генетичних змін, а не фенотипічних наслідків, є неправильним, отже, роль кросинговеру в генетичному алгоритмі перебільшується [7]. На кросинговер слід дивитись скоріше як на можливість ізолювання дефектів у зародку, а не як на можливість покращення еволюційної оптимізації шляхом рекомбінації гарних розв'язків, на чому наголошують дослідники в галузі генетичних алгоритмів [8]. Полігенність та плейотропія в більшості випадків перешкоджають останній можливості, адже класичне припущення «один ген – одна властивість» в органічній еволюції не є вірним.

Д. Фогель та В. Атмар в [9] представили емпіричні дані (отримані шляхом порівняння результатів роботи алгоритмів ЕП з та без рекомбінації на послідовностях лінійних функцій з параметризованими взаємодіями між генами), згідно з якими оператор мутації Гаусса є більш ефективним, ніж однокроковий кросинговер, та навіть більш ефективним, ніж комбінація кросинговеру та мутації.

Після цього вийшла низка робіт, присвячених дослідженню ЕП та генетичних алгоритмів, автори яких намагались встановити, за яких обставин використання оператора рекомбінації покращує продуктивність алгоритму [10]. Так, в [11] отримано емпіричний результат, згідно з яким кросинговер є перешкодою для великих популяцій при вирішенні проблеми виходу популяції з локального оптимуму. Водночас автори роботи [12] дослідили

нішу кросинговеру (двоточкового та однорідного) в генетичному алгоритмі та розробили задачу, для якої кросинговер має суттєві переваги над мутацією. В роботі [13] робиться висновок, що можливості кросинговеру чи гауссівської мутації з породження нащадка кращої у порівнянні з батьками якості великою мірою залежать від стану простору пошуку; мутація виявляється кращою на початку, а кросинговер – у процесі еволюції.

2.2. Теоретичні основи сучасного еволюційного програмування. Розглядаючи ЕП як оптимізаційну процедуру, природно дослідити її математичні властивості.

Маючи на меті дослідження збіжності алгоритму ЕП, Д. Фогель провів аналіз стандартної форми алгоритму ЕП для випадку $\beta_j = 1$, $\gamma_j = 0$, $\Phi(A_i) = F(\bar{X}_i) > 0$ ($i = 1, 2, \dots, N$, $j = 1, 2, \dots, n$). Зокрема, використовуючи ланцюги Маркова, Д. Фогель показав [7, 14], що ймовірність збіжності алгоритму дорівнює 1, тобто довів асимптотичну збіжність стандартної форми алгоритму до глобального оптимуму.

В доведенні Д. Фогеля кожна можлива популяція, яка може виникнути в процесі еволюції, інтерпретується як стан скінченного ланцюга Маркова, а єдиний поглинаючий стан ланцюга Маркова – як об'єднання всіх станів популяції, які містять точку глобального оптимуму. Простір станів визначається шляхом дискретизації простору R , яка фактично використовується при будь-якому представленні комп'ютером дійсних значень.

В роботі [5] звертається увага на те, що результат Д. Фогеля ґрунтується на: властивості елітарності відбору, яка гарантує монотонну поведінку еволюції, зокрема, існування поглинаючого стану; дискретизації простору пошуку; можливості переходу в будь-яку точку простору пошуку за один або декілька кроків.

Можна зауважити, що реальний глобальний оптимум може як завгодно сильно відрізнятися від знайденого алгоритмом. Більше того, в зв'язку з дискретизацією простору пошуку, нічого не можна сказати про властивість збіжності еволюцій-

ного програмування для цільової функції $f: R^n \rightarrow R$.

Аналіз матриці ймовірностей переходів [15], яку визначає еволюційний марківський ланцюг, також вказує на те, що ймовірність покращення розв'язку зростає як геометрична прогресія, збігаючись до 1.0, та є лінійною комбінацією власних значень цієї матриці переходів.

Виходячи зі схожості підходів ЕП та еволюційної стратегії, в [15] пропонується перенести на еволюційне програмування аналіз швидкості збіжності, проведений для еволюційної стратегії в роботі [16]. Ці результати вказують на геометричну швидкість збіжності для строго опуклих функцій, і більше того, на те, що швидкість збіжності може бути покращена як логарифм кількості нащадків λ в $(1, \lambda)$ -відборі.

У роботі [5] Д. Фогель також проаналізував випадок розмірності популяції $N=1$ та елітарної селекції; $k_j=1$, $z_j=0$, $j=1,2,\dots,n$. Результуючий алгоритм виявився ідентичним алгоритму $(1+1)$ -ЕС для $\sigma = \sqrt{f(\bar{x})}$.

Загалом можна сказати, що на сьогодні еволюційне програмування є відносно мало дослідженою парадигмою моделювання еволюції.

2.3. Варіанти алгоритмів сучасного еволюційного програмування. Використання стандартної форми алгоритму ЕП супроводжується такими труднощами:

- необхідність підбору $2n$ значень параметрів еволюції β_j та γ_j ($j=1,2,\dots,n$);
- узгодження та підбір функції масштабування Ω у випадку, коли немає інформації про глобальний оптимум;
- у випадках, коли значення функції пристосованості є надто великими, пошук стає квазівипадковим;
- негативно впливає на стійкість алгоритму випадок, коли глобальний мінімум функції пристосованості відрізняється від 0; точне досягнення глобального мінімуму в цьому випадку є неможливим.

Для подолання вказаних недоліків стандартної форми ЕП було запропонова-

но низку модифікацій наведеного вище алгоритму.

Один з варіантів модифікації допускає змінний розмір популяції, а також наявність різної кількості нащадків у різних батьків.

Інший підхід полягає у зміні процедур генерації нащадків та відбору. Наприклад, обирають випадково батька, з нього отримують шляхом мутації нащадка, оцінюють нащадка та додають в популяцію; одразу після цього з популяції вилучають найслабшу особину.

Найбільш перспективним підходом видається застосування ідеї самоадаптації в ЕП. Дійсно, в алгоритмі ЕП існує низка параметрів, які керують поведінкою алгоритму: розмір популяції, розподіл мутацій, тиск відбору. Тоді, наприклад, оператори мутації (варіації, зміни) можуть бути включені до складу особин як додаткова інформація щодо способу породження нащадків; ця додаткова інформація може підлягати мутації та селекції в той же спосіб, що і можливий розв'язок задачі. Були запропоновані методи розширення еволюційного пошуку до двокрокового процесу, який включає еволюцію мутаційних змін. Зауважимо, що такі методи були розроблені для ЕП та еволюційної стратегії повністю незалежно. Самоадаптовні алгоритми застосовувались як до проблем неперервної, так і до проблем дискретної оптимізації [15].

Враховуючи сказане, на сьогодні можна виділити, згідно робіт [5, 14, 15], 5 варіантів сучасного (неперервного) еволюційного програмування:

- стандартне ЕП (Standard EP) – характеризується відсутністю будь-якого механізму самоадаптації;
- неперервне стандартне ЕП (Continuous Standard EP) – на противагу механізму, базованому на зміні популяцій, новостворений індивід одразу оцінюється та додається в популяцію;
- мета-ЕП (Meta-EP) – включає додаткові змінні (параметри мутацій) в генотип з метою проведення самоадаптації;
- неперервне мета-ЕП (Continuous Meta-EP) – мета-ЕП, за якого новостворе-

ний індивід одразу оцінюється та додається в популяцію;

- *R*-мета-ЕП (Rmeta-EP) – крім стандартних відхилень, включає в генотип коваріації (представлені за допомогою коефіцієнтів кореляції) для самоадаптації. Цей метод, який, очевидно, легко узагальнити до неперервного *R*-мета-ЕП, був реалізований та протестований Д. Фогелем тільки для двовимірного випадку.

Зупинимось детальніше на питанні самоадаптації в ЕП – розглянемо мета-ЕП та *R*-мета-ЕП. Ще раз наголосимо, що в цьому випадку кожна особина містить не лише потенційний розв’язок задачі, але й інформацію щодо того, як розподіляються нащадки.

Мета-еволюційне програмування.

Мета-ЕП (Meta-EP), як вже зазначалось, включає додаткові змінні (параметри мутацій) в генотип з метою проведення самоадаптації. Доповнимо схему стандартної форми ЕП для випадку мета-ЕП.

Кодування розв’язків. Особина кодується у вигляді

$$A = (\bar{X}, \bar{\Sigma}) \in A_x \times A_s,$$

де $\bar{X} = (x_1, x_2, \dots, x_n) \in R^n$ – вектор параметрів об’єкта (object variable vector), які оцінюються за допомогою цільової функції,

$\bar{\Sigma} = (\sigma_1, \sigma_2, \dots, \sigma_n) \in R_+^n$ – вектор параметрів мутації, вектор стандартних відхилень (mutation step sizes vector, standard deviations vector), які використовуються при генерації нащадків.

Таким чином, у випадку мета-ЕП $A_x = R^n$, $A_s = R_+^n$. Можна узагальнити наведений спосіб кодування для стандартної форми ЕП: у випадку стандартної форми $A_x = R^n$, $A_s = \emptyset$.

Ініціалізація. Випадковим чином генерується початкова популяція, що складається з *N* особин A_i , $i=1, \dots, N$; рекомендоване значення розміру популяції $N \geq 200$ ([10]). При цьому значення елемента σ_{ij} , $j=1, 2, \dots, n$, кожної особини $A_i = (x_{i1}, x_{i2}, \dots, x_{in}, \sigma_{i1}, \sigma_{i2}, \dots, \sigma_{in})$, $i=1, \dots, N$, встановлюється випадково за законом рівномірного розподілу на інтервалі $[0, c]$, де

c – параметр еволюції, $c > 0$. В момент ініціалізації рекомендується брати $c = 25$ [7]. Під час оптимізації значення σ_{ij} можуть варіюватись на всьому діапазоні додатних дійсних чисел R_+ .

Застосування генетичних операторів. Існує кілька варіантів оператора мутації; в усіх цих варіантах вектор $\bar{\Sigma}$ використовується при генерації нащадка $A' = (\bar{X}', \bar{\Sigma}')$ індивідуума $A = (\bar{X}, \bar{\Sigma})$ та відбувається самоадаптація параметрів мутації $\bar{\Sigma}$. Розглянемо п’ять основних варіантів оператора мутації.

Найпоширенішими операторами мутації у мета-еволюційному програмуванні є наступні.

1. При застосуванні оператора мутації у роботах [3, 7, 15, 16] до індивідуума $A = (\bar{X}, \bar{\Sigma})$ нащадок $A' = (\bar{X}', \bar{\Sigma}')$ утворюється наступним чином:

$$x'_i = x_i + \sqrt{\sigma_i} N(0,1),$$

$$\sigma'_i = \sigma_i + \sqrt{\zeta \sigma_i} N(0,1).$$

Тут $N(0, 1)$ – стандартна нормально розподілена випадкова величина; ζ – зовнішній параметр, завдяки якому величини σ'_i мають тенденцію залишатись додатними; рекомендоване значення $\zeta = 6$. Для гарантування додатних значень змінних σ'_i (параметрів мутації), використовується обмежувальне правило (boundary rule):

- якщо $\sigma'_i \leq 0$, то покласти $\sigma'_i = \varepsilon_0 > 0$, де ε_0 – невелике додатне, наприклад, $\varepsilon_0 = 0.001$.

Слід пам’ятати, що при надто малих значеннях змінних – параметрів мутації – може суттєво уповільнитись процес збіжності.

Легко бачити, що однією з відмінностей мета-еволюційного програмування від еволюційної стратегії є те, що в мета-еволюційному програмуванні спочатку відбувається мутація параметрів об’єкта, а потім – параметрів мутації. Робилися спроби змінити порядок мутації векторів \bar{X} та $\bar{\Sigma}$. Зокрема, в роботі [17] дається явне порівняння варіантів мутації «спочатку

сигма» та «в кінці сигма» і робиться висновок, що перший варіант має суттєві переваги.

2. Змінні нащадка $A'=(\bar{X}',\bar{\Sigma}')$ особини $A=(\bar{X},\bar{\Sigma})$ обраховуються в роботах [10, 18] наступним чином:

$$\sigma'_i = \sigma_i + \sigma_i \cdot \alpha \cdot N_i(0,1),$$

$$x'_i = x_i + \sigma'_i \cdot N_i(0,1).$$

Тут $N_i(0,1)$ – стандартна нормально розподілена випадкова величина; α – зовнішній параметр, який задається користувачем. Рекомендоване значення $\alpha = \frac{1}{6}$ ($\alpha \approx 0.2$).

Звернемо увагу на такий факт. Якщо коефіцієнт α обрати надто великим, то величина σ'_i може стати від'ємною. В цьому випадку її слід замінити на деяке мале $\varepsilon_0 > 0$, хоча це й протирічить певною мірою ідеї самоадаптації. Якщо ж обрати коефіцієнт α надто малим, то проявляється тенденція уповільнення еволюції [18]. Для попередження надто близьких до 0 відхилень також застосовується обмежувальне правило (boundary rule):

- якщо $\sigma'_i < \varepsilon_0$, то покласти $\sigma'_i = \varepsilon_0 > 0$, де ε_0 – невелике додатне.

Існують варіанти оператора мутації з використанням не адитивної, а логнормальної схеми модифікації розміру кроку мутації.

3. В роботах [10, 19] розглядається алгоритм, названий *класичним* (classical EP – CEP), з мутацією виду:

$$x'_i = x_i + \sigma_i \cdot N_i(0,1),$$

$$\sigma'_i = \sigma_i \cdot \exp(\tau' N(0,1) + \tau N_i(0,1)).$$

Тут $N(0,1)$ – стандартна нормально розподілена випадкова величина, що генерується для кожного нащадка; $N_i(0,1)$ – стандартна нормально розподілена випадкова величина, що генерується для кожної змінної; $\tau = (\sqrt{2\sqrt{n}})^{-1}$ та $\tau' = (\sqrt{2n})^{-1}$ – навчальні норми (learning rates).

4. В роботах [10, 19] також розглядається алгоритм, названий *швидким* (fast-EP – FEP), з мутацією виду:

$$x'_i = x_i + \sigma_i \cdot \delta_i,$$

$$\sigma'_i = \sigma_i \cdot \exp(\tau N(0,1) + \tau N_i(0,1)).$$

Тут δ_i – випадкова величина з розподілом Коші (параметр масштабування $t=1$).

5. В роботах [10, 19] пропонується також модифікація попереднього алгоритму, названа *покращеним швидким* алгоритмом (improved fast evolutionary programming algorithm – IFEP). В цьому випадку в кожного з батьків породжуються по два нащадки, один з використанням розподілу Гаусса, інший – розподілу Коші. Останній розподіл має товстіший «хвіст», завдяки якому з'являється більше шансів згенерувати широкі мутації; в цілому це дає алгоритму більше шансів вийти з локального мінімуму. Водночас розподіл Гаусса при використанні малих кроків мутації дає більше можливостей для регулювання існуючих батьків.

Серед інших варіантів оператора мутації у мета-еволюційному програмуванні можна виділити алгоритми оптимізації керованих мутацій в полярній системі координат [20] та алгоритми з використанням комбінації різних розподілів мутації [21].

R-мета-еволюційне програмування. У випадку R-мета-еволюційного програмування (Rmeta-EP) для самоадаптації параметрів використовується коваріаційна матриця. Наведемо лише основні відмінності R-мета-еволюційного програмування від мета-еволюційного програмування.

Кодування розв'язків. Особина кодується у вигляді

$$A = (\bar{X}, \bar{\Sigma}, \bar{P}) \in A_x \times A_s \times A_p,$$

де $\bar{P} = (\rho_1, \rho_2, \dots, \rho_m) \in [-1, 1]^{n*(n-1)/2}$ – вектор коефіцієнтів кореляції (correlation

coefficients vector), $\rho_{ij} = \frac{c_{ij}}{\sqrt{\sigma_i \sigma_j}} \in [-1, 1]$,

$i \in \{1, \dots, n-1\}$, $j \in \{i+1, \dots, n\}$, представляє

матрицю коваріацій C^{-1} . Таким чином, у випадку R -мета-еволюційного програмування $A_x = R^n$, $A_y = R_+^n$, $A_\rho = [-1, 1]^{n*(n-1)/2}$.

Ініціалізація коефіцієнтів кореляції $\rho_j \in [-1, 1]$ здійснюється аналогічно ініціалізації інших змінних.

Застосування генетичних операторів. При застосуванні оператора мутації до індивідуума $A = (\bar{X}, \bar{\Sigma}, \bar{P})$ нащадок $A' = (\bar{X}', \bar{\Sigma}', \bar{P}')$ утворюється наступним чином [5]:

$$\begin{aligned} \bar{X}' &= \bar{X} + \bar{N}(\bar{0}, C(\bar{\Sigma}, \bar{P})), \\ \sigma'_i &= \sigma_i + \sqrt{\zeta \sigma_i} N_i(0,1), \end{aligned}$$

$$\rho'_j = \begin{cases} \tilde{\rho}_j, \text{ якщо } |\tilde{\rho}_j| \leq 1 \\ \text{sign}(\tilde{\rho}_j), \text{ інакше} \end{cases},$$

де $\tilde{\rho}_j = \rho_j + \sqrt{\omega \rho_j} N_j(0,1)$, ω – масштабуючий множник коефіцієнтів кореляції; рекомендоване значення $\omega = 6$.

Для повноти викладення матеріалу наведемо (згідно з [5]) зведену таблицю рекомендованих значень зовнішніх параметрів стандартного еволюційного програмування та мета-еволюційного програмування, яка може бути корисною при практичній реалізації відповідних алгоритмів. Звернемо увагу, що вказані значення були рекомендовані для задач малої розмірності.

Таблиця. Рекомендовані значення зовнішніх параметрів

Параметр	Має місце в		Значення за замовченням
	стандартне ЕП	мета-ЕП	
Границі u_i, v_i інтервалів значень змінних x_i	+	+	$u_i = -50$ $v_i = 50$
Верхня границя c параметрів мутації σ_i		+	$c = 25$
Константи пропорційності β_i	+		$\beta_i = 1$
Константи зсуву γ_i	+		$\gamma_i = 0$
Параметр підтримки самоадаптації ζ		+	$\zeta = 6$
Параметр турнірного відбору h	+	+	$h = 10$
Розмір популяції N	+	+	$N = 200$

2.4. Порівняння сучасного еволюційного програмування та еволюційної стратегії. Спочатку зазначимо спільні властивості сучасного еволюційного програмування та еволюційної стратегії.

Обидва підходи представляють фенотипічний підхід до моделювання еволюції: увага фокусується на поведінці потенційних розв'язків у популяції, а не на генетичному зв'язку між батьками та нащадками. При розв'язку задач оптимізації дійсних функцій оперують безпосередньо з дійсними числами. Мутації параметрів задачі мають нормальний розподіл. Відбір в ЕП можна розглядати як ймовір-

нісну $(N+N)$ -селекцію в еволюційній стратегії. В мета-ЕП, як і в еволюційній стратегії, до складу генотипу включені додаткові змінні – параметри мутацій, тобто кожна особина містить потенційний розв'язок та інформацію про те, як розподілятимуться нащадки.

Серед відмінностей між СЕП та еволюційною стратегією виділимо такі. В мета-еволюційному програмуванні, на відміну від еволюційної стратегії, кількість змінних – параметрів мутації дорівнює n ($\bar{\Sigma} = (\sigma_1, \sigma_2, \dots, \sigma_n)$), тобто немає можливості обрати довільне число таких змінних з інтервалу $[1, n]$. В еволюційній

стратегії спочатку проводиться мутація параметрів стратегії, а потім – параметрів об'єкта; в мета-ЕП порядок мутацій є зворотним, що призводить до ефекту затримки в зміні параметрів еволюції. Хоча необхідно відзначити, що останнім часом з'явилися приклади використання в мета-ЕП порядку мутації $\bar{S} - \bar{X}$, що фактично згладжує відмінності між еволюційним програмуванням та еволюційною стратегією. Зокрема, в роботах Д. Фогеля [22, 23] використовується логарифмічно нормальна адаптація параметрів \bar{S} , за якою виконується мутація змінних – параметрів задачі \bar{X} . Логарифмічно нормальний розподіл, що використовується в еволюційній стратегії, автоматично гарантує додатні значення параметрів стратегії, а також відсутність відхилень у випадку нульового тиску відбору. Водночас флуктуації в мета-ЕП не є нейтральними і не гарантують додатних значень параметрів мутації. В ЕП використовується стохастичний відбір, а в еволюційній стратегії – детермінований (найгірші розв'язки викидаються з популяції). Таким чином, в еволюційній стратегії кожна особина бере участь в середньому в породженні $\frac{\lambda}{\mu}$ нащадків, але, можливо, і жодного. В еволюційній стратегії реалізовано багато форм кросинговеру, а ЕП кросинговер не використовує.

Останній пункт вимагає більш ґрунтовного пояснення.

Справа в тому, що еволюційна стратегія є абстракцією еволюції на рівні *індивідуальної поведінки*. Включена в хромосому інформація для самоадаптації параметрів фактично є генетичною, а не фенотипічною інформацією, тому деякі форми кросинговеру є доречними. ЕП є абстракцією еволюції на рівні *репродуктивних популяцій, видів*. Точка в просторі пошуку в ЕП розглядається не як особина – представник певного виду, а скоріше як абстракція самого виду. Як наслідок, рекомбінація в ЕП не має сенсу, оскільки її неможливо застосувати до різних видів: кросинговер не відбувається між видами [10].

Висновок

Роботи Л. Фогеля та його учнів поширили процеси ЕП від задач прогнозування на моделювання систем, розпізнавання образів, теорію ігор тощо.

Сучасні вчені у царині ЕП займаються дослідженням еволюції програм, впливу різних параметрів середовища та різних видів мутації на результативність ЕП, розпізнаванням образів тощо. Принципи ЕП застосовуються у плануванні та маршрутизації трафіку, військово-стратегічному плануванні, діагностиці раку, керуючих системах, обробці сигналів, навчанні у іграх і багатьох інших застосуваннях.

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.

1. Fogel L.J., Lawrence J.A. A Retrospective View and Outlook on Evolutionary Algorithms // Proceedings of the International Conference on Computational Intelligence Theory and Applications: Lecture Notes In Computer Science. – 1997. – Vol. 1226. – P. 337–342.
2. Фогель Л., Оуэнс А., Уолш М. Искусственный интеллект и эволюционное моделирование. – М.: Мир, 1969. – 230 с.
3. Fogel L.J., Fogel D.B. Artificial Intelligence through Evolutionary Programming // Final Report for U. S. Army Research Institute, Contract. – 1986.
4. Гулаєва Н.М. Еволюційні алгоритми // Вісник Київського національного університету імені Тараса Шевченка. Серія: фізико-математичні науки. – 2013. – Вип. 2. – С. 141–150.
5. Bäck T. Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms. – Oxford University Press, 1996. – 318 p.
6. Galar R. Evolutionary search with soft selection // Biological Cybernetics. – 1989. – Vol. 60. – P. 357–364.
7. Fogel D.B. Evolving Artificial Intelligence // PhD thesis. – University of California, San Diego, CA. – 1992.

8. *Atmar W.* The philosophical errors that plague both evolutionary theory and simulated evolutionary programming // In: D.B. Fogel and W. Atmar (eds.) Proceedings of the 1st Annual Conference on Evolutionary Programming. – Evolutionary Programming Society, San Diego, CA. – 1992. – P. 27–34.
9. *Fogel D.B., Atmar W.* Comparing genetic operators with Gaussian mutations in simulated evolutionary processes using linear systems // Biological Cybernetics. – 1990. – Vol. 63, N 2. – P. 111–114.
10. *Eiben A.E., Smith J.E.* Introduction to Evolutionary Computing. – Springer, Natural Computing Series, 2007. – 300 p.
11. *Galar R.* Simulation of local evolutionary dynamics of small populations // Biological Cybernetics. – 1991. – Vol. 65. – P. 37–45.
12. *Eshelman L.J., Schaffer J.D.* Crossover's niche // In: S. Forrest (editor) Proceedings of the 5th International Conference on Genetic Algorithms. – Morgan Kaufmann Publishers, San Mateo, CA. – 1993. – P. 9–14.
13. *Jain A.D., Fogel D.B.* Case studies in applying fitness distributions in evolutionary algorithms. II. Comparing the improvements from crossover and gaussian mutation on simple neural networks // In: X. Yao, D.B. Fogel (eds.) Proceedings of the 2000 IEEE Symposium on Combinations of Evolutionary Computation and Neural Networks. – IEEE Press, Piscataway, NJ. – 2000. – P. 91–97.
14. *Fogel D.B.* Asymptotic Convergence Properties of Genetic Algorithms and Evolutionary Programming: Analysis and Experiments // Cybernetics and Systems. – 1994. – Vol. 25, N 3. – P. 389–407.
15. *Fogel D.B.* An overview of evolutionary programming // In: L.A. Davis et al. (eds.) Evolutionary algorithms. – Springer. – 1999. – P. 89–110.
16. *Bäck T., Rudolph G., Schwefel H.-P.* Evolutionary Programming and Evolution Strategies: Similarities and Differences // In: D.B. Fogel and W. Atmar (eds.) Proceedings of the 2nd Annual Conference on Evolutionary Programming. – Evolutionary Programming Society, La Jolla, CA. – 1993. – P. 11–22.
17. *Gehlhaar D.K., Fogel D.B.* Tuning evolutionary programming for conformationally flexible molecular docking // In: L.J. Fogel, P.J. Angeline, T. Bäck (eds.) Proceedings of the 5th Annual Conference on Evolutionary Programming. – MIT Press, Cambridge, MA. – 1996. – P. 419–429.
18. *Курейчик В.М., Родзин С.И.* Эволюционные вычисления: генетическое и эволюционное программирование // Новости искусственного интеллекта. – 2003. – № 5. – С. 13–19. Режим доступа: www.raai.org/library/ainews/2003/5/kureichik_rodzin.doc
19. *Yao X., Liu Y., Lin G.* Evolutionary programming made faster // IEEE Transactions on Evolutionary Computing. – 1999. – Vol. 3, N 2. – P. 82–102.
20. *Ghozeil A., Fogel D.B.* A Preliminary Investigation into Directed Mutation in Evolutionary Algorithms // In: H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel (eds.) Parallel Problem Solving from Nature PPSN IV. – Springer, Berlin. – 1993. – P. 329–335.
21. *Saravanan N., Fogel D.B.* Multi-Operator Evolutionary Programming // In: P.J. Angeline, R.C. Eberhart, R.G. Reynolds, and J.R. McDonnell (eds.) Evolutionary Programming VI: Proceedings of the 6th Annual Conference on Evolutionary Programming. – Springer, Berlin. – 1997. – P. 215–221.
22. *Chellapilla K., Fogel D.B.* Evolving an expert checkers playing program without human expertise // IEEE Transactions on Evolutionary Computation. – 2001. – Vol. 5, N 4. – P. 422–428.
23. *Fogel D.B.* Blondie24: Playing at the Edge of AI. – Morgan Kauffmann, San Francisco, 2002. – 406 p.

Одержано 28.02.2013

Про авторів:

Глибовець Микола Миколайович,
доктор фізико-математичних наук,
професор,
декан факультету інформатики НаУКМА,

Гулаєва Наталія Михайлівна,
кандидат фізико-математичних наук,
доцент кафедри інформатики НаУКМА.

Місце роботи авторів:

Національний університет
«Києво-Могилянська Академія»,
04655, м. Київ,
вул. Сковороди, 2.
Тел.: (044) 463 6985.
Факс: (044) 416 4515.
E-mail: glib@ukma.kiev.ua