DOI https://doi.org/10.15407/usim.2018.02.068

УДК 6:004.8

О.Г. МОРОЗ, мол. наук. співр., Міжнародний науково-навчальний центр інформаційних технологій та систем НАН України та МОН України, просп. Глушкова, 40, Київ 03187, Україна, olhahryhmoroz@gmail.com

АНАЛІЗ ЗАСТОСУВАННЯ ГЕНЕТИЧНИХ АЛГОРИТМІВ В ЗАДАЧАХ ГЛОБАЛЬНОЇ ОПТИМІЗАЦІЇ

Подано загальну характеристику генетичних алгоритмів як ефективних та перспективних засобів вирішення проблеми глобальної оптимізації. Розглянуто приклади складних реальних задач оптимізації та ідентифікації, успішно розв'язаних із застосуванням генетичних алгоритмів.

Ключові слова: *глобальна оптимізація, генетичні алгоритми, ідентифікація систем.*

Вступ

Часто виникають реальні задачі оптимізації та ідентифікації моделей такої складності і розміру, для розв'язання яких застосування більшості існуючих методів не дає бажаних результатів. Найвагомішими причинами цього є застосування градієнтів, складна програмна реалізація, відсутність механізмів виходу з локального оптимуму у разі глобального пошуку екстремуму, низька швидкодія і неточність, потреба у великих обсягах обчислювальних ресурсів та в аналітичному описі цільової функції. Все це сприяло розробці метаевристичних методів обчислювального інтелекту, зокрема, генетичних алгоритмів (ГА) [1—5], позбавлених більшості з вказаних недоліків. Ці алгоритми були отримані в процесі узагальнення та імітації таких властивостей живої природи, як природний відбір, пристосованість організмів до мінливих умов довкілля, успадкування нащадками життєво важливих властивостей батьків тощо. Ефективність використання ГА підтверджена численними успішними практичними реалізаціями.

Проблема глобальної оптимізації

Загальна постановка задачі глобальної оптимізації представлена у [6]. Нехай Ω — множина всіх можливих рішень дискретної, або неперервної проблеми оптимізації. При цьому Ω визначено деякими обмеженнями виду $g_i(x) \le 0$, $i = \overline{1, m_1}$; $h_j(x) = 0$, $j = \overline{1, m_2}$; $r_k(x) \ge 0$, $k = \overline{1, m_3}$, $f : \Omega \to R^n$ — цільова функція (ЦФ), яку потрібно оптимізувати. Необхідно знайти таку точку $x^* \in \Omega$, що $\forall x \in \Omega$, $f(x^*) \le (\ge) f(x)$. Точка x^* — точка глобального мінімуму (максимуму) на Ω. Функція f ставить у відповідність кожному рішенню $x \in \Omega$ дійсне число, яке позначає цінність рішення і визначає повне відношення порядку між довільною парою рішень у просторі пошуку Ω.

Перші методи, запропоновані для розв'язання задач оптимізації, були точними. Серед них прямі і градієнтні методи, динамічне програмування, стратегії релаксації і розкладання функції Лагранжа, методи перерахування (керовані або некеровані) та ін. [6]. Ці методи призначені для пошуку глобального оптимуму і гарантують його знаходження. У цьому їх основна перевага. Проте практичне застосування цих методів істотним чином обмежене низкою суттєвих причин. Зокрема, вони зазвичай потребують неприйнятних часових і/або обчислювальних ресурсів навіть при розв'язанні задач невеликої розмірності, є чутливими не тільки до розміру, але і до структури задачі, їх збіжність залежить від початкової точки, вони мають тенденцію застрягати в локальному оптимумі тощо [7]. Пошук оптимальних розв'язків задач, ЦФ яких є недифе-

ренційовані, мультимодальні, багатокритеріальні, з застосуванням точних методів є практично неможливим. Все це сприяло розробці оптимізаційних методів для пошуку практично прийнятних розв'язків, серед яких можна виділити два підкласи: *евристичні* і *наближені*. Евристичні, у свою чергу, розділяються на *метаевристики* і *спеціальні евристики*. Останні призначені для вирішення конкретних проблем, тоді як метаевристики є алгоритмами загального призначення і можуть застосовуватися для вирішення майже будь-якої проблеми оптимізації. Це пояснює підвищення інтересу до них протягом останніх десятиліть. На відміну від точних алгоритмів оптимізації, метаевристики не гарантують оптимальність отриманих рішень. У порівнянні з алгоритмами наближення, вони не забезпечують доказову точність розв'язку та доказові межі часу виконання. Проте сьогодні метаевристики вважаються найбільш перспективними і успішними методами оптимізації, оскільки здатні забезпечити "практично прийнятні" розв'язки за оптимальний час при вирішенні великих, *NP-складних* проблем науки і практики, на що не здатні інші методи.

Розрізняють два класи метаевристик [6]: *S-метаевристики*, в основу яких покладено одне рішення (*Single–Solution Based Metaheuristics*) і *P-метаевристики*, які базуються на використанні популяції (*Population-Based Metaheuristics*).

Типовими прикладами *S-метаевристик* є ітеративний локальний пошук (*iterated local search*), табу пошук (*tabu search*), імітація відпалу (*simulated annealing*), алгоритм великого потопу (*Great Deluge Algorithm*), жадібна рандомізована адаптивна процедура пошуку (*GRASP*). Всі ці алгоритми працюють з одним потенційним розв'язком задачі, яке певним чином покращується в результаті виконання ітераційних процедур.

Одночасна робота з набором потенційних розв'язків в *P-метаевристиках* виконує суттєву роль для їх стійкості, підвищує імовірність досягнення глобального оптимуму, а також сприяє усуненню локальних стаціонарних точок. Вони значно швидше і повніше досліджують простір пошуку для виявлення глобального оптимуму. Серед *P-метаевристик* найбільш поширеними є еволюційні алгоритми, найяскравіші представники яких — генетичні алгоритми.

Генетичні алгоритми

Передумовою для виникнення ГА стали модель біологічної еволюції Ч. Дарвіна і методи випадкового пошуку. «Винахід» ГА зазвичай приписують Дж. Голланду, що не зовсім відповідає дійсності, оскільки до нього деякі учені самостійно вигадали декілька подібних «винаходів» [8, 9]. Проте саме Дж. Голланд їх популяризував. Він також першим вивчив еволюційні явища, такі як адаптація з метою розробки методів з аналогічними механізмами для комп'ютерних застосувань.

Оскільки ГА ґрунтуються на принципах *біологічної еволюції і генетики*, то для їх опису активно (і деколи некоректно) використовують біологічні терміни. Наведемо декілька таких термінів. *Особина* — потенційний розв'язок задачі; *популяція* — набір особин; *нащадок* — як правило, покращена копія потенційного розв'язку (*батька*); *придатність* — зазвичай, якісна характеристика розв'язку. *Хромосома* — закодована структура даних особини у вигляді масиву фіксованої довжини. В найпростішому випадку — бінарний рядок фіксованої довжини. *Ген* — елемент цього масиву.

Формально ГА можна описати так [5]: ГА = $= \{P_0, n, \ell, f, G, s\}$, де $P_0 = (a_{10}, a_{20}, ..., a_{n0})$ — початкова популяція; a_{i0} — можливий розв'язок задачі, поданий у вигляді хромосоми; n — розмір популяції; ℓ — довжина кожної хромосоми; f — цільова функція; G — множина генетичних операторів; s — правило завершення алгоритму.

Загалом, робота ГА полягає в одночасній обробці елементів множини альтернативних розв'язків і складається з наступних основних етапів: 1) генерація початкової популяції; 2) обчислення значень ЦФ кожної її хромосоми; 3) поки не буде виконано правило завершення ГА, застосування генетичних операторів (селе-

кції, кросинговеру, мутації та ін.); 4) формування нової популяції і перехід до п. 2).

При розробці ГА необхідно визначити якісний і кількісний склад потенційних розв'язань задачі та сформувати відповідний пошуковий простір ГА, тобто множину всіх можливих хромосом, а також задати ЦФ і правило завершення ГА (наприклад, здійснення заданого числа ітерацій, отримання розв'язку потрібної якості на деякій ітерації).

Ефективність роботи ГА істотно залежить від способу кодування генів, складу початкової популяції, використовуваних генетичних операторів, параметрів ГА, таких як розмір популяції, кількості відібраних хромосом при селекції і для кросинговеру, імовірності використання генетичних операторів. Центральне місце в ГА належить генетичним операторам, з яких найбільш важливими є селекція (відбір), кросинговер (схрещування) і мутація.

Кодування потенційних розв'язків. Цей процес може бути здійснений при використанні бітів, цифр, дерев, масивів, матриць, списків або будь-яких інших об'єктів, що головним чином залежить від розв'язуваної задачі. Найпоширенішим є *двійкове кодування* (*Binary Encoding*) [1, 2], при якому кожен ген хромосоми має значення «ноль» або «одиниця». Існує декілька інших способів кодування, наприклад, цілими або дійсними числами, вузлами дискретизації або підінтервалами (для неперервної оптимізації), логарифмічне кодування (для багатовимірної оптимізації з великим простором параметрів) та ін. [4]. Кількість символів, які використовуються для кодування, має бути мінімально необхідною згідно *принципу мінімальних алфавітів* [2].

Генетичні оператори

Вони працюють аналогічно оператору в класичному математичному розумінні, відображаючи одну множину об'єктів на іншу, і представляють один з кроків ГА. У роботах [1—5] описано велику кількість різних операторів. Розглянемо основні генетичні оператори, які використовуються найчастіше.

Селекція (відбір). За використання цього оператора зберігається деяка кількість хромосом з кращими значеннями придатності на кожній ітерації ГА. Розглянемо декілька найчастіше використовуваних механізмів відбору [4].

Елітна селекція. Полягає у виборі заданої кількості хромосом поточної популяції з кращими значеннями придатності.

Пропорційна селекція (*Fitness-Proportionate Selection*), або *Селекція на основі рулетки* (*Roulette Wheel Selection*). Спочатку обчислюється придатність кожної хромосоми в популяції. Далі кожній хромосомі популяції ставиться у відповідність сектор колеса рулетки, пропорційний величині її пристосованості. Тоді при повороті колеса рулетки хромосома з більшою пристосованістю матиме велику імовірність вибору для подальшого схрещування.

Турнірна селекція (*Tournament selection*). Ця селекція з розміром турніру t полягає у випадковому виборі з популяції t хромосом і відборі з цієї групи в нову популяцію однієї хромосоми з найкращим значенням придатності. Так повторюється n разів. Турнірна селекція стала основним методом селекції в ГА з різних причин. По-перше, вона нечутлива до особливостей обчислення придатності. По-друге, дуже проста, не потребує попередньої обробки і може бути легко розпаралелена. По-третє, вона налаштовується: розмір турніру t визначає тиск селекції. На крайній випадок, коли $t=1$ маємо випадковий відбір. За дуже великих t (значно більше, ніж навіть розмір популяції) імовірність перемоги в турнірі найбільш пристосованої хромосоми наближається до одиниці, тому турнірна селекція буде кожного разу вибирати одну і ту ж пристосовану хромосому. У ГА найбільшою популярністю користується $t=2$.

Кросинговер. Оператор кросинговеру (ОК) породжує нащадків шляхом обміну частинами батьківських хромосом. При цьому зазвичай ОК застосовується з заданою імовірністю до пари хромосом-батьків з поточної популяції, утворюючи дві хромосоми–нащадки, проте існують й інші варіанти [2, 3]. Деякі найбільш поширені ОК розглянуто далі.

Одноточковий кросинговер (*One-point crossover*). З популяції випадково вибираються дві батьківські хромосоми, і довільним чином генерується точка розриву (або точка кросинговеру) $\ell_k \in [1, \ell - 1]$, де ℓ — довжина хромосоми. Далі батьківські хромосоми обмінюються частинами, утворюючи двох нащадків: перший нащадок є хромосомою, яка на позиціях від одиниці до ℓ_k містить гени першого з батьків, а на позиціях від $\ell_k + 1$ до ℓ — гени другого з батьків; другий нащадок — це хромосома, яка, відповідно, на позиціях від одиниці до ℓ_k містить гени другого з батьків, а на позиціях від $\ell_k + 1$ до ℓ — гени першого з батьків.

Двоточковий кросинговер (Two-point crossover) відрізняється від одноточкового тим, що випадковим чином генерується не одна, а дві точки розриву, і хромосоми обмінюються ділянками між цими двома точками [4, 5].

Рівномірний кросинговер (Uniform crossover). Випадково створюється хромосома-маска — рядок з нулів і одиниць, довжина якої дорівнює довжині хромосоми. Два нащадки утворюються за таким правилом. Одиниця в *i-й* позиції маски означає, що *i-й* елемент першого батька слід розташувати на *i-му* місці першого нащадка, а ноль означає, що *i-й* елемент другого батька слід розташувати на *i-му* місці першого нащадка. Вважаючи першого батька другим, а другого — першим, за описаним правилом отримують другого нащадка. Маска може бути одна на всіх або своя для кожної пари батьків [3, 4].

Мутація. Роль мутації полягає в запобіганні передчасній збіжності ГА і його "застрягання" в локальному оптимумі. Оператор мутації (ОМ) вносить випадкові зміни до генів хромосоми, додаючи новий генетичний матеріал в популяцію. Вибір процедури залежить, перш за все, від обраного способу кодування. Цей оператор може застосовуватися з заданою ймовірністю до всієї хромосоми або до кожного її гена. Серед всього різноманіття ОМ найчастіше використовуються такі.

Бітова мутація (*bit-flip mutation*) [1, 2]. Кожен ген хромосоми мутує із заданою малою імовірністю p_m, значення якої найчастіше обирають в інтервалі [0.001;0.1].

Багатоточкова мутація (*multi-point mutation*) [1, 5]. У хромосомі змінюються значення довільно вибраних генів, кількість яких може бути заданою або випадковою, але не перевищує довжини хромосоми. У разі, коли в хромосомі змінюється тільки один випадково вибраний ген, отримаємо *одноточкову мутацію* (*one-point mutation*) [2, 4]. Слід зазначити, що у багатьох задачах одноточкова мутація не здатна вивести ГА з локального оптимуму.

Рівномірна мутація (uniform mutation) [4] є прикладом оператора мутації у разі дійсного кодування. Вона змінює значення відібраного гена на рівномірно розподілену випадкову величину (р.р.в.в.) в заданих користувачем межах: $c_i = p_i + s_i r_i (x_i^u - x_i^v)$, де c_i — *i*-та координата нащадка; p_i — *i*-та координата батька; $s_i \in \{-1, 1\}$ — напрям кроку мутації, р.р.в.в.; r_i — відносний розмір кроку мутації, р.р.в.в., $r_i \in [0,1]$; x_i^u і x_i^v — верхня і нижня допустимі межі *i*-го параметра відповідно.

Необхідно відзначити, що зі стрімким розширенням сфер застосування перші ГА швидко видозмінювалися, намагаючись максимально враховувати особливості розв'язуваних задач. Сьогодні кількість різновидів ГА вражає. Це, зокрема: квантовий ГА (*Quantum GA*) [10], багатокритеріальний ГА недомінантного сортування (*Non-dominated Sorting Genetic Algorithm II*, *NSGA-II*) [11], мультиагентний ГА (*Multi-Agent GA)* [12], пилкоподібний ГА (*Saw-Tooth GA*) [13], мутативний ГА (*Mutative GA)* [14] та ін.

Моделювання простого генетичного алгоритму

Перші ГА були створені Дж. Голландом на основі його спостережень за природною еволюцією. Хоча ці ранні ГА були дуже простими, вони сприяли кращому розумінню їх поведінки, а також спровокували низку дискусій щодо різних аспектів ГА. Одна з найбільш гострих дискусій була викликана заявою Дж. Голланда про те, що *кросинговер є важливішим за*

мутацію. При цьому всі його аргументи мали чисто емпіричний характер. Думки учених були розділені. Деякі з них підтримували мутацію, інші — кросинговер. Проте з часом більшість фахівців дійшла висновку, що для вирішення цієї дискусії одних емпіричних даних недостатньо, і необхідно розробити деяку математичну базу для моделювання, пояснення і прогнозування поведінки ГА.

Запропонована Дж. Голландом в [1] *теорема про шаблони,* була, ймовірно, першою спробою теоретичного аналізу *простого* ГА, який працює з двійковими хромосомами певної довжини ℓ і для якого створення нової популяції розміром n засноване на трьох операторах: пропорційній селекції, одноточковому кросинговері і бітовій мутації з імовірністю застосування p_c і p_m відповідно. При цьому вважалося, що $p_c \in [0{,}8;1)$, а $p_m \in (0;0{,}1]$. Для формулювання теореми було введено такі поняття та означення.

Шаблон — бінарний рядок $H = (b_1, b_2, ..., b_\ell)$, $b_i \in \{0,1,*\}$, що містить хоча б один символ "*", замість якого може бути будь-яке зі значень «ноль» або «одиниця». Позиції рядка, що містять " * ", називають вільними, а решта позицій — закріпленими. Якщо шаблон H має k $(1 < k \le \ell)$ вільних позицій, то існує 2^k рядків, які збігаються на всіх $\ell - k$ закріплених позиціях. Множину таких рядків позначимо через Y_H. Рядок h належить шаблону H, якщо $h \in Y_H$.

Основними кількісними характеристиками шаблону H є:

• *порядок* $o(H)$ — кількість закріплених позицій в рядку;

• *визначальна довжина* (d_H) — відстань між першою і останньою закріпленими позиціями в рядку. Відзначимо, що $d_H \ge o(H) - 1$.

Нехай t — номер ітерації або умовний параметр часу, $P(t)$ — поточна популяція, а $f(x)$ — цільова функція рядка x. Позначимо через $U_H(t)$ всі рядки популяції $P(t)$, які належать шаблону H $(U_H(t) = P(t) \cap Y_H)$, а через $m(H,t)$ — кількість рядків множини $U_H(t)$.

У випадку пропорційної селекції важливими показниками є:

• середнє значення ЦФ популяції в момент t:

$$\bar{f}(t) = \sum_{x \in P(t)} \frac{f(x)}{n};$$

• середнє значення ЦФ шаблону в момент t:

$$\bar{f} = (H,t) = \frac{\sum_{x \in U(H,t)} f(x)}{m(H,t)}.$$

Теорема про шаблони [1, 3]. Якщо шаблон H має низький порядок ($o(H)$ мале), є коротким (d_H близьке до $o(H) - 1$) і середнє значення придатності шаблону в момент t більше середнього значення придатності популяції $P(t)$, то він (в середньому) збільшує своє представництво в популяції $P(t+1)$.

Формально цю теорему можна записати так:

$$E\left(m(H,t+1)\right) \ge \left(\frac{\bar{f}(H,t)}{\bar{f}(t)} \cdot m(H,t)\right) \times \times \left(1 - p_c \cdot \frac{d_H}{\ell - 1}\right) \times (1 - p_m)^{o(H)}, \quad (1)$$

де E — математичне сподівання.

У нерівності (1) перший співмножник показує вплив селекції, другий — кросинговера, третій — мутації на очікуване представництво рядка шаблону H в популяції $P(t+1)$. Зокрема, з (1) випливає, що чим менші значення $o(H)$ і p_m, тим менший вплив мутації на роботу ГА. Тому здавалося, що теорема про шаблони свідчила про перевагу кросинговера над мутацією, і нерівність (1) часто записувалася без третього співмножника. Проте, недовго. Як теоретична основа для пояснення роботи ГА, ця теорема створила багато нових проблем і суперечностей, що викликало могутній потік критики на її адресу [3]. Зокрема, виявилося, що вона є придатною лише для вираження того, що відбувається (в середньому) з шаблоном H під час переходу ГА від однієї популяції до наступної, але не дає уявлення про роботу ГА протягом декількох ітерацій (можливо, протягом всього виконання ГА). При цьому абсолютно не враховується вірогідність наявності в популяції інших шаблонів і їх взаємний вплив. Отже, з'явилася потреба у створенні більш потужної математичної основи.

Точні моделі ГА. У 1991 р. була запропонована [15] точна модель простого ГА, яка фіксує кожну його характеристику у вигляді математичних операторів. Для простоти вони розглядалися ГА з нескінченною популяцією. У цій моделі популяція в момент часу t була представлена двома вектор-стовпцями дійсних чисел $\vec{p}(t)$ і $\vec{s}(t)$:

$$\vec{p}(t)=\left(p_0(t),p_1(t),\ldots,p_{r-1}(t)\right)^T,$$

де $p_i(t)$ — частка i-рядків у популяції у момент часу t, r — кількість усіх двійкових рядків довжини ℓ, що представляють можливі розв'язки задачі $(r=2^\ell)$. Під i-рядком розуміють двійковий рядок, представлений цілим числом, яке кодується цим двійковим рядком (наприклад, якщо $\ell=5$, то сьомий рядок — це бінарний рядок 00111).

При пропорційному відборі отримаємо:

$$\vec{s}(t)=\left(s_0(t),s_1(t),\ldots,s_{r-1}(t)\right)^T=\frac{F\cdot p_i(t)}{\sum_{j=0}^{r-1}F_{j,j}\cdot p_j(t)}.$$

Тут $F=(F_{ij})$ є $(r\times r)$-матрицею, в якій $F_{ij}=0$ при $i\neq j$ і $F_{ij}=f(i)$ при $i=j$, де $f(i)$ — ЦФ i-рядка. Компонента $s_i(t)$ вектора $\vec{s}(t)$ рівна імовірності вибору i-рядка як батька поточної популяції. Зазначимо, що F можна розглядати як оператор, який представляє селекцію.

Задача полягала у пошуку оператора G, що має властивість: $\vec{s}(t+1)=G\cdot\vec{s}(t)$. Тоді, починаючи з нульової ітерації $s(0)$, ітеративно застосовуючи оператор G, можна отримати точний опис очікуваної поведінки ГА у будь-який момент часу t. Ефект кросинговеру і мутації імітується в математичній структурі $\mathcal{M}(\cdot)$, яка складається з таких $r_{ij}(k)$, що

$$E\left[p_k(t+1)\right]=\sum_{\forall i,j}s_i(t)\cdot s_j(t)\cdot r_{i,j}(k),$$

де $r_{ij}(k)$ — імовірність того, що рядок k, отриманий внаслідок рекомбінації між рядками i та j. Об'єднаний ефект матриць $\mathcal{M}$ і F такий, що $G=\mathcal{M}\circ F$, де $\circ$ — композиція $\mathcal{M}$ і F. У роботі [17] було показано, що:

$$r_{i,j}(\mathrm{O})=\frac{1-p_m}{i}\left[\eta^{d(i)}\left(1-p_c+\frac{p_c}{i-1}\sum_{c=1}^{i-1}\eta^{-\Delta_{i,j,c}}\right)+\right.$$

$$\left.+\eta^{d(j)}\left(1-p_c+\frac{p_c}{i-1}\sum_{c=1}^{i-1}\eta^{\Delta_{i,j,c}}\right)\right]. \qquad (2)$$

Тут $\eta=p_m/(1-p_m)$, $d(x)$ — кількість одиничних бітів у двійковому рядку, який представляє ціле число x, c — точка кросинговеру $\Delta_{i,j,c}=d$ ("найправіші c бітів i") — d ("найправіші c бітів j"). Маніпулюючи математичним рівнянням (2), були отримані вирази для решти $r_{ij}(k)$.

Створення цієї математичної моделі є одним з найбільш значних і важливих етапів у визначенні ролі кросинговеру і мутації. Проте вона має серйозний недолік, оскільки передбачає нескінченність популяції. У більшості реальних ситуацій чисельність популяції завдяки великій кількості практичних обмежень є скінченною. Тому залишалася необхідність розробки точної моделі ГА для скінченної популяції.

Історичні передумови для вирішення цієї проблеми було закладено в [16, 17], де обґрунтовано можливість використання ланцюгів Маркова для моделювання ГА. Ланцюг Маркова першого порядку є випадковим процесом, при якому імовірність того, що він перебуває у певному стані в даний момент часу t, залежить лише від стану процесу у момент часу $t-1$. Якщо n — розмір популяції $(n<r)$, то кількість всіх можливих популяцій дорівнює $N=C_{n+r-1}^{r-1}$. Генетичний алгоритм моделюється ланцюгом Маркова, що має N можливих станів, які представляють різні популяції. Ланцюг Маркова задається своїми станами і $(N\times N)$-матрицею $Q=(Q_{ij})$ перехідних імовірностей із стану в стан, де Q_{ij} — імовірність того, що на k-й ітерації буде сформовано популяцію P_j за умови, що на $(k-1)$-й ітерації була популяція P_i.

У роботі [18] отримано точну модель ГА для скінченної популяції. Для цього всі можливі популяції були представлені $(r\times N)$-матрицею $Z=(z_{i,j})$, у якій i-й стовпець $\varphi_i=\langle z_{0,i},\ldots,z_{r-1,i}\rangle$ є вектором інцидентності для i-ї популяції. Іншими словами, $z_{i,j}$ — кількість входжень y-рядків в i-ту популяцію. Наприклад, якщо $\ell=2$ і $n=2$, то $r=4$, $N=10$, а матриця Z матиме такий вигляд:

Бінарні рядки (y)	Популяції									
	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9
00	2	1	1	1	0	0	0	0	0	0
01	0	1	0	0	2	1	1	0	0	0
10	0	0	1	0	0	1	0	2	1	0
11	0	0	0	1	0	0	1	0	1	2

Точна модель простого ГА є кінцевим ланцюгом Маркова, в якій стани задаються індексами стовпців матриці Z, а матриця перехідних імовірностей Q складається з елементів

$$Q_{i,j} = n! \prod_{y=0}^{r-1} \frac{\left\{ \mathcal{M}\left(\frac{F\varphi_i}{|F\varphi_i|} \right)_y \right\}^{z_{y,j}}}{z_{y,j}!}.$$

Тут $|F\varphi_i|$ позначає суму координат вектора $F\varphi_i$. Подібні моделі є найбільш детальними і можуть використовуватися для прогнозування будь-якого аспекту поведінки простого ГА. Якщо швидкість мутації не дорівнює нулю, то всі елементи матриці Q відмінні від нуля, а отже, ланцюг Маркова буде ергодичним і має граничний розподіл. Цей факт дозволив запропонувати гіпотезу, в якій йдеться про те, що хоча короткострокова поведінка популяцій ГА залежна від початкової популяції, довгострокова поведінка популяцій від неї не залежить [18].

Недолік цієї моделі полягає в тому, що її можна використовувати тільки для простих ГА. Проте це дозволило зробити докладний і точний аналіз деяких фундаментальних механізмів, наприклад, вона дала більш повне уявлення про те, як оптимальні ймовірності мутацій залежать від чисельності популяції; показала, як взаємопов'язані важливість імовірностей кросинговеру і мутації; підтвердила одні й відкинула інші з багатьох емпіричних претензій щодо кросинговеру і мутації, а також значно зменшила роль кросинговеру в ГА.

Застосування генетичних алгоритмів

Задача ідентифікації. Спосіб побудови моделей, який використовує експериментальні дані, отримані в реальних умовах функціонування системи, називається ідентифікацією, а сама задача ідентифікації формулюється так: за результатами спостережень над вхідними і вихідними сигналами системи потрібно побудувати в певному сенсі оптимальну формальну модель цієї системи. При цьому якщо структура моделі невідома, то говорять про структурно-параметричну ідентифікацію, інакше — про параметричну ідентифікацію [19]. Проблему ідентифікації можна сформулювати у вигляді двох взаємопов'язаних оптимізаційних задач: дискретної (для пошуку оптимальної структури моделі) і неперервної (для оцінки параметрів моделі). Тому ефективність вирішення проблеми суттєво залежить від вдалого вибору методу оптимізації.

Одним із сучасних підходів до моделювання складних нелінійних систем є блок-орієнтований підхід, в якому система розділяється на декілька певним чином з'єднаних підсистем, серед яких є, принаймні, одна нелінійна і одна лінійна підсистеми [19]. При моделюванні системи, що складається з двох послідовно з'єднаних підсистем, залежно від того, яка підсистема (лінійна або нелінійна) приймає вхідні сигнали, розрізняють моделі Гаммерштейна і Вінера. Структурно-параметрична ідентифікація таких систем є непростим завданням і часто потребує застосування методів глобальної оптимізації, зокрема ГА. Наприклад, у [20] здійснюється ідентифікація моделі Гаммерштейна, яка складається з одного вхідного статичного нелінійного блоку і динамічної лінійної стаціонарної підсистеми. У цій моделі невідома нелінійна статична частина, яку слід оцінити, представлена мережею радіально базисних функцій, структура якої оптимізується з застосуванням ГА.

В [21] розглянуто ідентифікацію моделі Вінера, в якій лінійна динамічна частина передує нелінійній статичній частині, представленій зворотною кусково-лінійною функцією, параметри якої оцінюються алгоритмом ГА. В [22] моделюється сила пальця на основі даних сигналів поверхневої електроміографії (ЕМГ) для покращення управління протезом руки. Сила пальця і дані сигналів ЕМГ генеруються, якщо суб'єкт виконує випадкові рухи безіменним

пальцем для імітації різних рівнів сили. Подальша обробка сигналів ЕМГ виконується з використанням просторової фільтрації. Деякі маски просторового фільтру для фільтрації сигналів ЕМГ оптимізуються з використанням ГА.

Успішне застосування ГА в онлайн-ідентифікації для прямої оцінки полюсів і нулів динамічної системи стає можливим, оскільки ГА не потребує лінійності за цими параметрами [23]. ГА використовується для ідентифікації параметрів лінійних і нелінійних систем для мінімізації помилки апроксимації за наявності шуму [24]. Під час моделювання було виявлено, що конвергенція процесу дуже чутлива до механізму селекції і таких параметрів ГА, як розмір популяції, імовірність мутації і кросинговеру. Для подолання цієї проблеми деякі з цих параметрів розглянуто як конструктивні параметри ГА, закодовані в хромосомі, які оптимізуються шляхом еволюційного процесу.

Застосування ГА часто істотно збільшує ефективність інших методів гібридизацією для досягнення синергетичного ефекту. У [25] представлено наявні підходи до застосування ГА для проектування архітектури МГУА нейронних мереж (МГУАНМ), основою яких є ітераційний багаторядний алгоритм БІА МГУА. Розглянуто основні наявні способи кодування структур МГУАНМ і відповідні генетичні оператори. Показано спектр реальних проблем, які можуть бути успішно вирішені гібридними алгоритмами МГУАНМ-ГА.

***Управління ядерним паливом* (*УЯП*)**. Це недетермінована *NP-складна* задача оптимізації. Мета УЯП полягає в тому, щоб мінімізувати витрати на виробництво електроенергії, враховуючи експлуатаційні обмеження і безпеку [26]. Основними характеристиками проблем, пов'язаними з УЯП, є велика розмірність, велика кількість обмежень і можливих рішень, відсутність інформації про похідні ЦФ і висока складність її обчислення [27].

Оптимізація УЯП — багатокритеріальна задача, метою якої є ідентифікація вектора розв'язку, яка дає найкращий компроміс між різними цільовими функціями. Застосовувати ГА в управлінні ядерним паливом почали у 90-х роках минулого століття [28]. Сьогодні ГА є одним з основних методів, які використовуються в УЯП, оскільки вони не застрягають в локальних оптимумах, мають високу швидкодію і легко розпаралелюються, на відміну від інших раніше використовуваних методів. Основною перевагою ГА є можливість з мінімальними змінами використовувати їх для різних типів реакторів [29].

Деталі реалізації ГА, такі як спосіб кодування хромосом, оператори ГА, критерії збіжності варіюються залежно від вибраного типу проблеми оптимізації УЯП. Найбільш поширеними є бінарне і цілочисельне кодування; турнірна і пропорційна селекції; одноточковий, двоточковий і частково підібраний кросинговери; рівномірна мутація.

Регресивне тестування. При розробці нової версії програмного забезпечення (ПЗ) вона обов'язково піддається так званому регресивному тестуванню (РТ), щоб гарантувати відсутність внесення нових помилок (помилок регресу) в раніше перевірений код попередньої версії ПЗ. Тільки після такого тестування випускається нова версія ПЗ. Регресивне тестування потребує великих витрат, які часто складають майже половину всіх витрат на розробку ПЗ [30]. Задача РТ полягає в тому, щоб визначити *оптимальний набір тестів* (ОНТ), які мають як прямий, так і опосередкований стосунок до змін в ПЗ; і в умовах ресурсно-часових обмежень організувати процес запуску тестових випадків так, щоб виявити якомога більше внесених прямих і опосередкованих помилок. Зазначимо, що навіть для невеликих програмних систем множина допустимих тестових випадків може бути величезною, що дає підставу зарахувати цю проблему до *NP-складної.*

Одним з важливих методів РТ є метод пріоритизації (тобто встановлення оптимальних пріоритетів) тестів. Пріоритизація упорядковує набір тестів так, щоб в процесі РТ тести з вищим пріоритетом завжди виконувалися раніше, ніж тести з нижчим пріоритетом, який встановлюється згідно з деякою метрикою, наприклад, *APFD* (*Average of the Percentage of Faults Detected*) або її модифікації [31]. У роботі

[32] представлено різні підходи до розв'язання задачі пріоритизації. Проте останнім часом все частіше для розв'язання цієї проблеми залучають ГА. Так, в [33] був запропонований метод на основі ГА для встановлення пріоритетів тестів з урахуванням часу їх виконання. Експеримент проводився для програм *Gradebook i JDepend*. Були застосовані інструменти відстежування процесу *Emma* і *Linux*, і результати, визначені кількісно, з використанням метрики *APFD*. Зрештою ОНТ, отриманий методом, основаним на ГА, виявився кращим, порівняно з ОНТ, отриманим іншими методами пріоритизації. Інший вид пріоритизації тестів на основі ГА запропоновано в [34]. У роботі подано широкий спектр мутацій, кросинговерів і селекцій, використаних для зміни порядку тестів. Було здійснено експериментальне дослідження на восьми реальних застосунках, з використанням однієї і тієї ж метрики. Метод на основі ГА показав кращі результати порівняно з іншими методами пошуку екстремуму.

Задача кластеризації. Кластеризація сьогодні є однією з найважливіших задач інтелектуального аналізу даних і розглядається як особливий вид *NP-складної* задачі оптимізації. Кластеризація — це неконтрольована класифікація набору об'єктів, згрупованих в кластери так, щоб виконувалися умови: гомогенність (*однорідність*) *всередині кластерів:* об'єкти, що належать одному і тому ж кластеру, мають бути якомога більш схожими, і *гетерогенність між кластерами*: об'єкти, що належать різним кластерам, мають максимально відрізнятися [35].

Формально задачу кластеризації можна сформулювати, як у [36]. Припустимо, що $X=\{x_1,x_2,...,x_N\}$ — набір об'єктів, і між ними задано функцію відстані $\rho(x,x')$. Завдання полягає в пошуку K кластерів $C_1,C_2,...,C_K$ для яких: $C_i\neq\varnothing$ для $i=\overline{1,K}$; $C_i\bigcap C_j=\varnothing$ для $i=\overline{1,K}$ та $i\neq j$; $\bigcup_{i=1}^{K}C_i=X$. Кожен кластер при цьому має містити об'єкти, близькі за метрикою ρ, тоді як об'єкти різних кластерів мають істотно відрізнятися за цією метрикою. Як функції ρ найчастіше використовують Евклідову відстань (*Euclidian Distance*) та відстань Махаланобіса (*Mahalanobis Distance*).

Існує багато алгоритмів кластеризації [35], зокрема *K*-середніх (*k-means*), нечітких-с-середніх (*Fuzzy-c-means*), *EM*-алгоритм ("*expectation-maximization*") на основі відстані тощо.

У 1992 р. у [37] вперше повідомлено оптимістичні результати досліджень з використанням ГА для кластеризації. З того часу на основі ГА розроблено багато алгоритмів розв'язання різних складних задач кластерного аналізу. Так, однією з основних цілей алгоритмів кластеризації є пошук «природних» груп в наборі об'єктів з одночасним їх розподілом на ці групи [37]. Проте жоден з не базованих на ГА алгоритмів не здатний ефективно і автоматично формувати «природні» групи, особливо коли кількість кластерів, які входять до складу набору даних, може бути великою. Огляд існуючих методів кластеризації на основі ГА і приклади їх практичного застосування подано у [36]. Показано, що використання ГА у поєднанні з традиційними методами дуже ефективне і усуває велику кількість проблем.

Завдання про призначення. Це один з найбільш представницьких класів задач комбінаторної оптимізації. Серед них особливе місце посідає *NP-складна* узагальнена задача про призначення (УЗП) [38]. Хай J — множина робіт, а I — множина агентів. Кожен агент $i\in I$ має обмежений граничний ресурс b_i і може бути призначений для виконання будь-якої кількості робіт з J. Виконуючи роботу, j — агент i витрачає свій ресурс на величину a_{ij} і отримує прибуток c_{ij}. Необхідно знайти матрицю призначень $X=(x_{ij})$, де $x_{ij}=1$, якщо агент i виконує роботу j, та $x_{ij}=0$ в іншому випадку, яка максимізує загальний прибуток

$$c(x)=\sum_{i\in I}\sum_{j\in J}c_{ij}x_{ij}\to\max$$

за обмежень

$$\sum_{j\in J}a_{ij}x_{ij}\le b_i,\forall\, i\in I \qquad (3)$$

$$\sum_{i\in I}x_{ij}\le 1,\forall\, j\in J; \qquad (4)$$

$$x_{ij} \in \{0,1\}, \forall i \in I, \forall j \in J .$$

Обмеження (3) гарантує, що загальні витрати ресурсу агентом *i* на виконання всіх призначених йому робіт не перевищує його граничний ресурс. Обмеження (4) гарантує, що кожну роботу виконує лише один агент.

Багато інших задач комбінаторної оптимізації є окремими випадками цієї задачі. Наприклад, якщо всі $b_i = 1$ і $a_{ij} = 1$, то УЗП перетворюється на задачу про *максимальне поєднання пар*. Якщо $c_{ij} = a_{ij}$ для всіляких комбінацій агентів і робіт, то задача зводиться до *мультиплікативного ранця*. У разі, коли є всього один агент, який виконує всі *n* робіт і при цьому $b_1 = n$, всі $a_{1j} = 1$, а c_{ij} — прибуток, який отримує агент, якщо він після завершення роботи *i* виконуватиме роботу *j*, то маємо *задачу про комівояжера*.

Останніми роками в більшості сфер управління виробництвом і при оптимізації управління ланцюгами постачань (*Supply Chain Management*) широко і ефективно використовуються підходи на основі ГА, аналітичний огляд яких представлено в [39]. Механізм покращення якості розв'язання УЗП на основі заданої евристики (наприклад, жадібної) шляхом налаштування її параметрів на основі ГА досліджено у [40]. У завданні цільового призначення групи безпілотних літальних апаратів такий підхід зменшує участь експерта, необхідного для розв'язання поставленої задачі, що приводить до ефективнішого використання наявних ресурсів. Автоматизований інструмент на основі ГА для оптимального призначення людей на виконання завдань проекту відносно його різноманітних конфігурацій вартості і тривалості створено у [41]. Генетичний алгоритм здійснює перебір сценаріїв проектування програмного забезпечення. Завдяки такому генератору примірників можна провести структуровані дослідження про вплив найбільш важливих атрибутів проблеми на її рішення.

Висновки

Генетичні алгоритми є потужним засобом розв'язання різноманітних складних задач глобальної оптимізації і моделювання, які часто або не можуть бути розв'язані іншими методами, або розв'язуються ними з незадовільною точністю та/або швидкістю. Такі задачі характеризуються неповнотою інформації, багатокритеріальністю, великою розмірністю, нелінійністю, відсутністю аналітичного опису ЦФ тощо.

Ефективність ГА істотно залежить від того, наскільки якісно буде проаналізовано проблему, що безпосередньо впливає на подальший вибір виду ГА, його операторів, способу кодування потенційних розв'язків та інших характеристик. Їх можна застосовувати самостійно, адаптуючись під конкретне завдання, або удосконалювати існуючі методи розв'язання, утворюючи гібридні системи. Одним з певних недоліків ГА є багатократне обчислення значення ЦФ, проте за зростаючої потужності обчислювальних засобів цей недолік практично не впливає на його ефективність. Теоретичні аспекти добре розвинені для простих ГА. Генетичні алгоритми мають велику перспективу і потребують подальшого удосконалення, розроблення і більш загального теоретичного обґрунтування.

REFERENCES

1. *Hollannd J.H.* Adaptation in natural and artificial systems, University of Michigan, 1975, 210 p.
2. *Goldberg D.E.* Genetic algorithms in search, optimization, and machine learning, Addison-Wesley, Reading, 1989, 432 p.
3. *Reeves C.R.*, *Rowe J.E.* Genetic algorithms: principles and perspectives. A Guide to GA Theory, Kluwer Acad. Publ., 2003, 332 p.
4. *Глибовець М.М.*, *Гулаєва Н.М.* Еволюційні алгоритми, К.: НаУКМА, 2013, 828 с.
5. *Курейчик В.М.* Генетические алгоритмы и их применение, Таганрог: Изд-во Таганрогского РТУ, 2002, 244 с.
6. *Talbi E.-G.* Metaheuristics: from design to implementation, John Wiley & Sons, Inc., 2009, 593 p.
7. *Deb K.* Multi-objective genetic algorithms: Problem difficulties and construction of test problems, EvolutionaryComputation, 1999, **7**, N 1, P. 205–230.
8. *Bremermann H.J.* Optimization through evolution and recombination. Yovits M.C., Jacobi G.T., Goldstein G.D., eds., Self-Organizing Systems. Spartan Books, 1962, P. 93–106.

9. *Reed J.*, *Toombs R.*, *Barricelli N.A.* Simulation of biological evolution and machine learning, J. of Theoretical Biology, 1967, **17**, N 3, P. 319–342.
10. *Zhang G.X.* Quantum-inspired evolutionary algorithms: a survey and empirical study, J. Heuristics, 2011, **17**, N 3, P. 303–351.
11. *Fast* and Elitist Multiobjective Genetic Algorithm: NSGA-II, K. Deb, A. Pratap, S. Agarwal et al. IEEE Transactions on evolutionary computation, 2002, **6**, N 2, P. 182–197.
12. *A multiagent* genetic algorithm for global numerical optimization / W. Zhong, J. Liu, M. Xue et al., IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2004, **34**, N 2, P. 1128–1141.
13. *Koumousis V.K.*, *Katsaras C.P.* A saw-tooth genetic algorithm combining the effects of variable population size and reinitialization to enhance performance, IEEE Trans. Evol. Comput., 2006., **10**, N 1, P. 19–28.
14. *Eldos T.* Mutative Genetic Algorithms, J. of Computations & Modelling, 2013, **3**, N 2, P. 111–124.
15. *Vose M.D.*, *Liepins G.E.* Punctuated Equilibria In Genetic Search, Complex Systems, 1991, **5**, N 1, P. 31–44.
16. *De Jong K.A.* An analysis of the behavior of a class of genetic adaptive systems, Doctoral Thesis, Department of Computer and Communication Sciences, University of Michigan, Ann Arbor, 1975.
17. *Goldberg D.E.*, *Segrest P.* Finite Markov chain analysis of genetic algorithms, Proc. of the 2nd Int. Conf. on Genetic Algorithms, Cambridge, 1987, P. 1–8.
18. *Nix A.E.*, *Vose M.D.* Modelling genetic algorithms with Markov chains, Annals of Mathematics and Artificial Intelligence, 1992, **5**, N 1, P. 79–88.
19. *Ljung L.* System Identification — Theory for the User, 2nd Edition, Prentice-Hall, Upper Saddle River, N J, 1999, 607 p.
20. *Гаращенко Ф.Г.*, *Мороз О.Г.* Ідентифікація нелінійної системи Гаммерштейна за допомогою генетичного алгоритму, Вісн. КНУ ім.Т. Шевченка. Серія: фізико-математичні науки, 2012, № 1, С. 145–150.
21. *Hatanaka T.*, *Uosaki K.*, *Koga M.* Evolutionary Computation Approach to Wiener Model Identification, Proc. of the 2002 Cong. on Evolutionary Computation, CEC'2002, Honolulu, Hawaii: IEEE, 2002, P. 914–919.
22. *Sebastian A.*, *Kumar P.*, *Schoen M.P.* Spatial filter masks optimization using genetic algorithm and modeling dynamic behavior of sEMG and finger force signals, Int. J. of Circuits, Systems, and Signal Processing, 2011, **5**, N 6, P. 597–608.
23. *Kristinsson K.*, *Dumont G.* System identification and control using genetic algorithms, IEEE Transactions on Systems, Man and Cybernetics, 1992, **22**, N 5, P. 1033–1046.
24. *Duong V.*, *Stubberud A.R.* System identification by genetic algorithm, Proc. IEEE Aerospace Conf., Big Sky, Montana, 2002, **5**, N 3, P. 2331–2337.
25. *Мороз О.Г.*, *Степашко В.С.* Огляд гібридних структур МГУА-подібних нейронних мереж та генетичних алгоритмів, Індуктивне моделювання складних систем: Зб. наук. праць, вип. 7, К.: МННЦ ІТС НАНУ, 2015, С. 173–191.
26. *Chapot J.L.C.*, *Da Silvab F.C.*, *Schirrub R.* A New Approach to the Use of Genetic Algorithms to Solve the Pressurized Water Reactor's Fuel Management Optimization Problem, Annals of Nuclear Energy, 1999, **26**, N 7, P. 641–655.
27. *Pereira C.M.N.A.*, *Schirru R.*, *Martinez A.S.* Basic Investigations Related to Genetic Algorithms in Core Designs, Annals of Nuclear Energy, 1999, **26**, N 3, P. 173–193.
28. *DeChaine M.D.*, *Feltus M.A.* Nuclear Fuel Management Optimization Using Genetic Algorithms, Nuclear Technology, 1995, **111**, N 1, P. 109–114.
29. *Karahroudi M.R.*, *Shirazi S.A.M.*, *Sepanloo K.* Optimization of Designing the Core Fuel Loading Pattern in AVVER-1000 Nuclear Power Reactor Using the Genetic Algorithm, Annals of Nuclear Energy, 2013, **57**, N 7, P. 142–150.
30. *Leung H.*, *White L.* Insights into regression testing, Proc. of the Conf. on Software Maintenance, 1989, P. 60–69.
31. *Elbaum S.*, *Malishevsky A.G.*, *Rothermel G.* Test case prioritization: A family empirical studies, IEEE Transactions on Software Engineering, 2002, **28**, N 2, P. 159–182.
32. *Мороз Г.Б.*, *Плис А.В.* Регрессивное тестирование: методы и будущие направления исследований, Проблеми програмування, 2014, № 2–3, С. 133–142.
33. *Time* aware test suite Prioritization, K.R. Walcott, M.L. Soffa, G.M. Kapfhammer et aql., Proc. of Int. Symp. on software Testing and Analysis (ISSTA 06), New York, USA, 2006, P. 1–12.
34. *Conrad A.P.*, *Roos R.S.* Empirically Studying the role of selection operators during search based test suite prioritization, Proc. of the ACM SIGEVO Genetic and Evolutionary Comp. Conf., Portland, Oregon, 2010, P. 1373–1380.
35. *Freitas A.A.* A review of evolutionary algorithms for data mining, Soft Computing for Knowledge Discovery and Data Mining, Springer, 2008, P. 79–111.
36. *Sheikh R.H.*, *Raghuwanshi M.M.*, *Jaiswal A.N.* Genetic algorithm based clustering: A Survey, First International Conference on Emerging Trends in Engineering and Technology (ICETET '08), IEEE, 2008, P. 314–319.
37. *Krovi R.* Genetic algorithms for clustering: a preliminary investigation, Proc. of the Twenty-Fifth Hawaii Int. Conf., 1992, **IV**, P. 540–544.
38. *Feltl H.*, *Raidl G.R.* An Improved Hybrid Genetic Algorithm for the Generalized Assignment Problem, The 19th Annual ACM Symp. on Applied Computing (SAC '04), March 14–17, 2004, Nicosia, Cyprus, P. 990–995.
39. *Chakravarthy V.K.*, *Ramana V.V*, *Umashankar C.* Genetic Algorithmic Approach to Generalized Assignment Problems in Conjunction with Supply Chain Optimization: A Survey, pdfs.semanticscholar.org/0774/ 5e749118a21c4d4d8818 bab29c01116c2ec2.pdf

40. *Juell P.*, *Perera A.*, *Nygard K.E.* Application of a Genetic Algorithm to Improve an Existing Solution for the Generalized Assignment Problem, www.cs.ndsu.nodak.edu/~nygard/research/GaIctpFinal2.pdf
41. *Alba E.*, *Chicano F.* Software project management with GAs, Information Sciences: an Int. J., 2017, **177**, N 11, P. 2380–2401.

Поступила 28.03.2018

O.G. Moroz, junior research scientist, Departament for Technologies of Inductive Modelling,
International Research and Training Center for Information Technologies and Systems
of the NAS and MES of Ukraine, Glushkov ave., 40, Kyiv, 03187, Ukraine, olhahryhmoroz@gmail.com

ANALYSIS AND APPLICATION OF GENETIC ALGORITHMS FOR GLOBAL OPTIMIZATION PROBLEMS

Introduction. Most of the existed traditional methods of optimization and identification models of complex systems are not effective in solving the problems of finding a globally optimal solution for the undifferentiated, multimodal, nonlinear, and multi-objective tasks. This leads to the development of approximate methods, in particular the meta-heuristic global search methods such as the genetic algorithm. The effectiveness of their application is confirmed by the numerous successful practical implementations.

Purpose. The purpose of the research is to examine more comprehensively the theoretical and practical aspects of the genetic algorithms and their capabilities for solving optimization and system identification problems.

Methods. The goal of this article is achieved by presenting a comprehensive survey of the main publications in the area of genetic algorithms theory and their application to the complex optimization tasks.

Results. The theoretical and applied aspects of genetic algorithms are considered in detail. Some examples of modern global optimization and model identification problems successfully solved by genetic algorithms are presented.

Conclusion. The genetic algorithms are a powerful tool for solving various complex global optimizations and modeling tasks that are characterized by incompleteness of input information, multi-objectiveness, large dimensionality, nonlinearity, lack of analytical description of objective function etc. The effectiveness of GA depends on its type, the choice of genetic operators, encoding method of potential solutions, etc. The theoretical aspects are well developed for simple GAs. Genetic algorithms have a great perspective and require further improvements, developments and more general theoretical justification.

Keywords: *global optimization, genetic algorithm, system identification.*

О.Г. Мороз, мл. науч. сотр., отдел информационных технологий индуктивного моделирования,
Международный научно-учебный центр информационных технологий и систем
НАН Украины и МОН Украины, просп. Глушкова, 40, Киев 03187, Украина, olhahryhmoroz@gmail.com

АНАЛИЗ ПРИМЕНЕНИЯ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ В ЗАДАЧАХ ГЛОБАЛЬНОЙ ОПТИМИЗАЦИИ

Введение. Большинство существующих традиционных методов оптимизации и идентификации моделей сложных систем неэффективны при решении проблем поиска глобально оптимального решения для недифференцированных, мультимодальных нелинейных и многоцелевых задач. Это привело к разработке приближенных методов, в частности метаэвристических методов глобального поиска, таких как генетический алгоритм (ГА). Эффективность их применения подтверждается многочисленными практическими реализациями.

Цель. Подробно рассмотреть теоретические и практические аспекты ГА и их возможности для решения задач оптимизации и идентификации систем.

Методы. Цель статьи достигается путем представления всестороннего обзора основных публикаций в области теории генетических алгоритмов и их применения для эффективного решения сложных задач оптимизации.

Результаты. Рассмотрены теоретические и прикладные аспекты ГА. Приведены примеры современных глобальных задач оптимизации и идентификации моделей, успешно решаемых генетическими алгоритмами.

Выводы. Генетические алгоритмы служат мощным инструментом для решения различных сложных задач глобальной оптимизации и моделирования, которые характеризуются неполнотой входной информации, многокритериальностью, большой размерностью, нелинейностью, отсутствием аналитического описания целевой функции и пр. Эффективность ГА зависит от его типа, выбора генетических операторов, метода кодирования потенциальных решений. Теоретические аспекты хорошо разработаны для простых ГА, в перспективе требующих усовершенствования, разработки и более глубокого теоретического обоснования.

Ключевые слова: *глобальная оптимизация, генетический алгоритм, идентификация системы.*