

УДК 681.5.015

ПОРІВНЯЛЬНИЙ АНАЛІЗ ГЕНЕРАТОРІВ СТРУКТУР МОДЕЛЕЙ У ПЕРЕБІРНИХ АЛГОРИТМАХ МГУА

О.Г. Мороз, В.С. Степашко

Міжнародний науково-навчальний центр інформаційних технологій та систем НАН та МОН України, Київ, пр-т Академіка Глушкова, 40

olga_moroz@irtc.org.ua, stepashko@irtc.org.ua

В статті представлені наявні способи повного та неповного перебору моделей в алгоритмах МГУА та їх порівняльний аналіз.

Ключові слова: Комбінаторний алгоритм МГУА, Перебірний алгоритм МГУА, Генератор структур моделей.

The article presents the available techniques of exhaustive and partial search of models in GMDH algorithms and their comparative analysis.

Keywords: Combinatorial GMDH algorithm, Sorting-out GMDH algorithm, Model structures generator.

В статье представлены имеющиеся способы полного и неполного перебора моделей в алгоритмах МГУА и их сравнительный анализ.

Ключевые слова: Комбинаторный алгоритм МГУА, Переборный алгоритм МГУА, Генератор структур моделей.

Вступ

Перебірні алгоритми МГУА є ефективним засобом розв'язання задач структурно-параметричної ідентифікації, прогнозування, побудови моделей об'єктів і процесів за даними спостережень в умовах невизначеності та низького рівня апріорних знань про модельований об'єкт. Особливість цих алгоритмів МГУА полягає в тому, що вони знаходять оптимальну структуру моделі в результаті мінімізації помилки прогнозування на скінченній множині моделей різної структури.

Першим перебірним алгоритмом є комбінаторний алгоритм МГУА (СОМВІ) [1-4] з повним перебором варіантів структур моделей для пошуку найкращої. Цей алгоритм обмежений у практичному застосуванні, оскільки потребує великих витрат обчислюваних ресурсів і часу та без використання спеціальних засобів не дозволяє розв'язувати задачі з більш ніж 30-ма вхідними змінними (аргументами) навіть на сучасних комп'ютерах. Кількість перебраних моделей за цим алгоритмом є показниковою функцією від кількості незалежних змінних, тобто зростає експоненційно.

Найбільший вплив на швидкість та обсяг обчислень у комбінаторному алгоритмі має вибір генератора структур моделей, тобто механізму їх перебору. Додатковими засобами прискорення обчислень в алгоритмі СОМВІ є розпаралелювання операцій у процесі перебору варіантів моделей та застосування рекурентних процедур оцінювання параметрів, проте вони дозволяють збільшити граничну складність задач лише приблизно до 40 аргументів.

Існує також значна група перебірних алгоритмів, розроблених для прискорення пошуку моделі оптимальної складності з метою розв'язування задач великої розмірності. Вони ефективно застосовуються на практиці й базуються здебільшого на тих чи інших ідеях скорочення обсягу перебору за рахунок різних процедур спрямованого пошуку глобального екстремуму заданого критерію. Недоліком усіх пошукових алгоритмів різного типу, порівняно з комбінаторним, є ризик пропуску моделі оптимальної складності, але це компенсується їх значними перевагами у швидкодії та знаходженням моделі, близької до оптимальної за значенням заданого критерію.

Перебірні алгоритми МГУА дають можливість ефективно розв'язувати актуальні проблеми штучного інтелекту, оскільки автоматично знаходять закономірності, що діють в системі, та дозволяють зменшити суб'єктивний вплив користувача на результати моделювання.

В цій роботі виконано аналіз та класифікацію генераторів структур моделей найбільш відомих перебірних алгоритмів МГУА.

1. Постановка задачі

Згідно з [5, 6], для побудови моделі оптимальної складності розв'язується така задача: за даними матриці спостережень вхідних змінних X розміру $[n \times m]$, де n – число спостережень, m – число вхідних змінних, і вектора спостережень вихідної змінної y розмірності $[n \times 1]$ необхідно оцінити залежність (структуру моделі та її параметри) такого вигляду:

$$y = X\theta^\circ + \xi = y + \xi, \quad (1)$$

де y – точний або істинний вихід об'єкта (сигнал); θ° – невідомий істинний вектор параметрів, структуру і значення якого треба оцінити; ξ – вектор випадкових завад (шум); X – матриця вхідних змінних.

При цьому слід сформулювати деяку множину \mathfrak{F} моделей різної структури у вигляді $\hat{y}_f = f(X, \hat{\theta}_f)$ і знайти оптимальну модель f^* з мінімальним значенням заданого критерію $CR(\cdot)$:

$$f^* = \arg \min_{f \in \mathfrak{F}} CR(y, f(X, \hat{\theta}_f)), \quad (2)$$

де $\hat{\theta}_f$ – вектор оцінок параметрів для кожної $f \in \mathfrak{F}$, що визначається як розв'язок неперервної екстремальної задачі

$$\hat{\theta}_f = \arg \min_{\theta_f \in R^{s_f}} Q(y, X, \theta_f), \quad (3)$$

де s_f – складність моделі f , $Q(\cdot) \neq CR(\cdot)$ – критерій якості розв’язку задачі параметричної ідентифікації кожної частинної моделі, що генерується в процесі розв’язування задачі структурної ідентифікації (1).

В цій роботі якість моделі будемо визначати за значенням критерію регулярності $AR(f)$, який будується на основі розбиття вибірки X на дві підвибірки X_A та X_B . На *навчальній* підвибірці X_A оцінюємо параметри моделі, а на *перевірній* підвибірці X_B обчислюємо помилку:

$$AR(f) = \| y_B - f(X_B, \hat{\theta}_{Af}) \|^2, \quad (4)$$

де $\hat{\theta}_{Af}$ – вектор параметрів моделі f , визначений на підвибірці X_A .

Якість моделі f^* вважається оптимальною, коли значення критерію $AR(f)$ буде найменшим, тобто:

$$f^* = \arg \min_{s=1,m} AR(f). \quad (5)$$

При використанні комбінаторного алгоритму виконується перебір усіх можливих моделей з вибором найкращої з них за критерієм селекції. За невеликої кількості аргументів можна виконати повний перебір, при цьому загальна кількість P_m усіх можливих моделей, які містять від 1 до m аргументів, обчислюється за формулою:

$$P_m = \sum_{j=1}^m C_m^j = 2^m - 1. \quad (6)$$

Це показникова функція, і повний перебір моделей на сучасних однопроцесорних комп’ютерах є практично можливим, коли кількість аргументів m не більше 30.

2. Генератори структур у перебірних алгоритмах МГУА з вичерпним перебором моделей

У структурі комбінаторного алгоритму можна виділити такі основні блоки [5, 6, 7]: 1) перетворення початкових даних відповідно до вибраної системи опорних (базисних) функцій, в якій шукається модель (базис може бути поліноміальним, тригонометричним, у формі різницевих рівнянь тощо); 2) генерування і перебір моделей різної структури у вибраному базисі; 3) обчислення значень деякого критерію селекції, який має властивості зовнішнього доповнення, і послідовний відбір частинних моделей, кращих за цим критерієм.

Найбільш істотний вплив на швидкодію алгоритму має другий блок *генерування (перебору) частинних моделей*. Основні операції, які здійснюються в цьому блоці, є такими: формування структури чергової частинної моделі та відповідної системи нормальних рівнянь, а також її розв'язання (оцінка коефіцієнтів моделі). Розглянемо особливості другого блоку у разі поліноміальних опорних функцій.

Структури частинних моделей формалізуються за допомогою двійкового структурного вектора $d = (d_1, d_2, \dots, d_m)$: якщо елемент d_i приймає значення 1, то відповідний i -й аргумент включається в модель, якщо 0 – не включається ($i=1, \dots, m$).

Найчастіше використовуваною при здійсненні повного перебору моделей є схема зміни вектора d за принципом *двійкового лічильника*. Вона дозволяє істотно розширити можливості комбінаторного алгоритму, а також побудувати алгоритм з універсальною структурою перебору та простою реалізацією [1, 3]. При її використанні в частинну модель включаються аргументи, що відповідають стану двійкового регістра, встановленому після додавання одиниці в останній розряд (число розрядів m рівне числу аргументів). При цьому кількість і склад аргументів у частинних моделях постійно змінюються, наприклад: 001, 010, 011, 100, ... , та будується взаємно однозначна відповідність між порядковим номером поточної частинної моделі та станом структурного вектора, що відповідає послідовним десятковим числам.

Ця схема дозволяє організувати рекурентну процедуру перебору, при якій досягається найкраще узгодження схеми перебору моделей і методу обчислення оцінок коефіцієнтів [3, 5]. Також вона дозволяє досить просто виключити з розгляду моделі з лінійно залежними членами.

При використанні рекурентних методів оцінювання параметрів досить ефективним є *лічильник Гарсайда*, в якому моделі, що йдуть підряд, відрізняються одним членом. Він дозволяє рекурентно обчислювати коефіцієнти моделей, що економить пам'ять, але виконує більшу кількість операцій і є повільнішим для розрахунків порівняно з двійковим лічильником з методом обрамлення [5].

Простим за ідеєю, але складнішим для реалізації, є *лічильник з послідовним ускладненням моделей або послідовний лічильник*, який доцільніше застосовувати в алгоритмах повного перебору з обмеженням складності моделей [6]. При цьому спершу розглядаються всі можливі варіанти розміщення у векторі d однієї одиниці ($C_m^1 = m$ варіантів), потім –

двох ($C_m^2 = \frac{m(m-1)}{2}$ варіантів), і так далі, до m одиниць ($C_m^m = 1$ варіант) .

Загальна кількість варіантів дорівнює P_m , див. формулу (6), тобто виконується повний перебір різних структур.

Якщо при такому лічильнику ввести обмеження на максимальну складність генерованих моделей, то одержимо генератор структур з повним перебором моделей обмеженої складності алгоритму COMBIS [3].

3. Генератори структур у перебірних алгоритмах МГУА з неповним перебором моделей

Відомі алгоритми часткового перебору моделей для пошуку оптимальної з них використовують специфічні способи послідовної генерації їх структур і показують поліноміальну складність обчислень в експериментальних дослідженнях. Вони здатні суттєво зменшити час та обсяг обчислювальних ресурсів, необхідних для знаходження оптимальної моделі в порівнянні з комбінаторним алгоритмом МГУА, і за прийнятний час дозволяють розв'язувати задачі з сотнями аргументів.

3.1. Пошукові алгоритми МГУА з детермінованим пошуком оптимальної моделі

Комбінаторно-селекційний алгоритм MULTI для пошуку моделі оптимальної складності. Першим алгоритмом спрямованого послідовного пошуку оптимальної моделі, тобто результату повного перебору, був так званий «багатоетапний комбінаторно-селекційний алгоритм» MULTI [8-10]. Він реалізує процедуру поетапного формування структур та оцінювання параметрів моделей з послідовним збільшенням складності частинних моделей тільки на один аргумент. При цьому моделі, що генеруються, залишаються в початковому базисі, а число аргументів, що містяться в них, відповідає номеру етапу.

Загальна схема алгоритму: спочатку оцінюються всі моделі, що містять тільки один аргумент, і вибирається $F \leq m$ кращих із них; на кожному наступному етапі додаються по одному різні аргументи до кожної з кращих моделей попереднього етапу і відбираються ті, що покращують величину критерію попереднього етапу. Ця процедура продовжується доти, поки зменшується величина критерію якості моделей, а її зупинка визначається автоматично за умовою збільшення мінімального значення критерію на певному етапі обчислень. При цьому модель оптимальної складності визначається мінімальним значенням критерію на попередньому етапі.

Алгоритм MULTI, як і багаторядні алгоритми, є селекційним, проте має скінченне число етапів – не більше кількості незалежних змінних m – і дозволяє шукати модель у тому ж заданому функціональному базисі, який використовується і в комбінаторному алгоритмі. Така схема скороченого перебору відрізняється від відомих покрокових схем перш за все застосуванням принципу неостаточних рішень, що істотно збільшує імовірність знаходження результату повного перебору. Загальна кількість порівнюваних за MULTI моделей у разі постійної свободи вибору на етапі $F_s = t$ є степеневою (кубічною) функцією числа аргументів m на відміну від показникової залежності 2^m при повному переборі [8, 5]. Можна сказати, що MULTI об'єднує переваги комбінаторних і багаторядних алгоритмів: він призначений для вирішення комбінаторних завдань моделювання, але за

обчислювальних витрат, звичних для багаторядних, і при цьому має велику імовірність отримання результату повного перебору.

Перебірні алгоритми на основі кореляційного ранжування послідовності аргументів. У роботах [11, 12] ступінь релевантності аргументів пропонується оцінювати за величиною модуля коефіцієнта парної кореляції аргумента з вихідною змінною. Модель будується тільки для відібраних 20-25 найбільш істотних аргументів.

У роботі [13] пропонувалося розбивати послідовність аргументів, ранжованих за модулем парного коефіцієнта кореляції, на групи по 20-25 змінних. За допомогою повного перебору для кожної групи знаходиться краща модель за заданими критеріями. Всі аргументи, які увійшли до кращих моделей, об'єднуються і визначають собою нову вибірку. Далі на цій вибірці формуються нові групи по 20-25 аргументів і знову будуються моделі – така послідовність дій повторюється доти, поки вибірка стане такою, щоб можна було виконати повний перебір – тобто не більше 25 аргументів.

У іншій модифікації комбінаторного алгоритму [14-18] запропоновано схожий спосіб, коли з ранжованих за модулем коефіцієнта кореляції аргументів формується дві підмножини, перша з яких містить близько 25 аргументів, інша – решту. За комбінаторним алгоритмом знаходиться оптимальна модель, і аргументи, які не увійшли до її структури, видаляються з вибірки. Далі вибірка даних поповнюється до 25 змінних з іншої підмножини аргументів, які мають найбільше значення модуля коефіцієнта кореляції, і знову будується модель. Алгоритм продовжується доти, поки не будуть використані всі аргументи вхідної вибірки.

Отже, основні ідеї модифікації багатоетапного алгоритму МГУА [14-18] на основі кореляційного ранжування послідовності аргументів запропонованої багатоетапної процедури такі: 1) ранжування змінних за істотністю з подальшою генерацією повної структури базової моделі; 2) пошук оптимальної моделі, використовуючи обмежену частину базисних функцій; 3) повторення заміни невикористаних базисних функцій у вхідному наборі новими, доки не будуть розглянуті всі базисні функції.

Ця процедура, як і реалізована в [13], виконує повний пошук серед обмеженого числа змінних багато разів. Для додаткового прискорення обчислень в цьому алгоритмі може також застосовуватися розпаралелювання операцій. Перевагою є можливість розв'язання задач з великою розмірністю.

У роботі [19] запропоновано замість *парної* кореляції використовувати *множинну*, що є значно коректнішим. Відповідно до цього підходу, спочатку за багаторядним алгоритмом МГУА, що має велику швидкодію, відбираються F кращих моделей, які складають множину \mathfrak{F}_F . Ступінь інформативності кожного аргумента визначається числом його потраплянь у кращі моделі. Тобто тут використовується *частотний критерій* визначення ступеня інформативності аргументів, який встановлює, як часто конкретний аргумент присутній в побудованих моделях з множини \mathfrak{F}_F :

$$FC_j = \frac{q_j}{F} . \quad (4)$$

Тут q_j – число моделей з множини \mathfrak{S}_F , у яких присутній j -й аргумент. Аргументи, що мають більше значення критерію FC , можна вважати більш інформативними.

Перебірні алгоритми з використанням процедур послідовного відсіювання найменш інформативних аргументів. Індуктивний підхід на основі МГУА передбачає перебір багатьох можливих варіантів розв’язання задачі моделювання і вибір найкращого варіанта моделі. Щоб побудувати найбільш адекватну модель, у вибірку включають якомога більше аргументів. Але за дуже великої їх кількості перебір усіх варіантів займає багато часу і часто стає неможливим. Тому виникає проблема прискорення комбінаторного алгоритму МГУА, що можна здійснити за рахунок відбору інформативних аргументів, які мають найбільший рівень впливу на вихідну змінну, і видалення решти неінформативних із вибірки.

В роботах [20-23] запропоновано новий багатоетапний алгоритм перебірного типу, в якому інформативність аргументів оцінюється за критерієм FC (4) і на кожному етапі відсіюються найменш інформативні.

У роботі [24] проведено аналіз ефективності цього критерію спільно з методами послідовної селекції інформативних аргументів FSS (Forward Successive Selection), BSS (Backward Successive Selection) і CSS (Combined Successive Selection).

Всі три методи розбивають процес пошуку кращої моделі на декілька етапів, на кожному з яких за допомогою алгоритму обмеженого перебору типу COMBIS будується множина \mathfrak{S}_F моделей прийнятної (за часом обчислень) складності, потім за допомогою частотного критерію оцінюється ступінь інформативності аргументів. Найменш інформативні аргументи вилучаються з вибірки. Метод FSS розглядає моделі малої складності ($s = 1, \dots, s_{\max}$), збільшуючи на кожному етапі s_{\max} . У методі BSS процес виконується в зворотному порядку, починаючи з моделей більшої складності ($s = m - s_{\max}, \dots, m$). Метод CSS є комбінацією двох попередніх підходів, в якому виконуються перебір моделей, як малої, так і великої складності.

Метод BSS показав найкращі результати вирішення завдань з великою кількістю аргументів. Такий результат обґрунтований тим, що BSS на кожному етапі розглядає моделі великої складності, тим самим враховуючи не попарну і не часткову кореляцію, а множинну кореляцію всіх істотних аргументів з вихідною змінною.

Як показали експерименти, при використанні критерію FC на вибірках з великим числом аргументів задача структурної ідентифікації розв’язується неоднозначно, оскільки часто істотні аргументи некоректно відсіюються на початкових етапах. Ймовірною причиною цього є те, що не враховується якість $CR(f)$ моделей, які входять до множини \mathfrak{S}_F , де оцінюється

інформативність аргумента. Деякі істинні аргументи мають гірше значення критерію FC , ніж неістинні, через що виникає можливість втрати істинних.

Неінформативні аргументи мають невеликі коефіцієнти в моделях, внаслідок чого вони мають слабкий вплив на значення вихідної змінної. Отже, при оцінюванні ступеня інформативності аргумента доцільно враховувати також ступінь його впливу на вихідну величину в побудованій моделі. Цього можна досягти за допомогою спільного врахування норми вектора аргумента у вибірці і його коефіцієнта в цій моделі. Виходячи з цих міркувань, у [25] пропонується використовувати так званий *ваговий критерій* WC оцінки ступеня інформативності аргумента на множині \mathfrak{F}_F :

$$WC(x_j, \theta_{jf}, CR(f)) = \frac{\|x_j\|^2}{nF} \sum_{k=1}^F \frac{\theta_{jf_k}^2}{CR(f_k)}. \quad (6)$$

Тут $\|x_j\|$ – норма вектора вимірювань аргумента у вибірці W , F – свобода вибору, θ_{jf_k} – коефіцієнт при аргументі x_j у моделі f_k .

На відміну від критерію FC , який враховує тільки *присутність* аргумента в поточній моделі, ваговий критерій WC враховує також його *вагу* в кожній моделі та якість $CR(f)$ моделі, в яку він входить.

В роботі [25] також показано, що ваговий критерій WC , на відміну від частотного FC , здатний виділяти істинні аргументи незалежно від величини свободи вибору. Ці результати дозволяють говорити про ефективність критерію WC при розв'язанні задачі структурної ідентифікації.

Перебірний алгоритм з багаторядним принципом формування структури моделі. У роботі [26] запропоновано процедуру організації скороченого перебору за принципом багаторядного алгоритму: на першому ряді будуються звичайні частинні моделі від усіх пар початкових аргументів, на наступному ряді формуються частинні моделі вже від чотирьох початкових аргументів, які містяться в моделях від проміжних аргументів, і так далі – тобто від ряду до ряду число аргументів у частинних моделях подвоюється. Таким чином відбувається неповний (скорочений) перебір ускладнених моделей. Складність частинних описів (кількість доданків у поліномі) з кожним рядом селекції збільшується і одночасно оцінюються всі параметри розгорнутої моделі, щоб зменшити можливість втрати істотних аргументів. При цьому з ряду в ряд передаються списки аргументів, що увійшли в кращі частинні описи, а не оцінки вихідної величини, і зберігається принцип селекції, типовий для багаторядної процедури.

Рекурентний адитивно-мультиплікативний багатоступінчастий алгоритм. У [27, 28] запропоновано підхід до розпізнавання об'єктів, заданих множинами спостережень (підмножинами рядків матриці об'єкт-властивості). Він передбачає побудову процедур розпізнавання в просторі параметрів

моделей об'єктів класифікації, де кожен об'єкт представлений точкою в просторі параметрів своєї моделі.

Для знаходження такої структури розроблено рекурентний адитивно-мультиплікативний багатоетапний алгоритм РАМБА МГУА. Структури, що перебираються, утворюють послідовно ускладнюванні описи (їх відслідковується F), до кожного з яких, на кожному k -тому етапі, додається найкраща за зовнішнім критерієм узагальнена змінна (УЗ), що генерується як член повного полінома.

Організація перебору вкладеними структурами. Для збільшення точності шуканих моделей передбачено розширення початкової множини змінних. Для генерації моделей використовується дерево перебірного алгоритму з використанням гілок. Гілки дерева відповідають утвореним раніше УЗ. На кожному ряді алгоритму реалізується рух по вузлах дерева зліва направо і зверху вниз. Для того, щоб пересуватися по даному дереву зліва направо і зверху вниз, необхідно рухатися по відповідному табличному масиву знизу вгору – спочатку по рядках, де сума значень змінних розширеної множини вхідних змінних (вузлів) дорівнює 1, потім де дорівнює 2 і так далі до $p + 1$, де p – максимальний степінь полінома.

Для першого і всіх наступних рівнів дерева в алгоритмі введено ступені свободи F_1 і F_2 , які визначають максимальну кількість кращих за зовнішнім критерієм вузлів на даному рівні, з яких триває пошук. Параметр F визначає кількість моделей, які відбираються в процесі виконання алгоритму. Організована таким чином послідовність вкладених структур дозволяє для оцінювання параметрів моделей застосувати рекурентні формули на основі розв'язування систем нормальних рівнянь методом «обрамлення» [3, 5], що істотно збільшує швидкодію алгоритму та зменшує час обчислень.

3.2. Пошукові алгоритми МГУА з випадковим пошуком оптимальної моделі

Оптимізаційний метод скорочення перебору моделей. Оптимізаційний метод (ОМ) [29] ґрунтується на розгляді задачі (5) пошуку оптимальної структури як комбінаторної задачі цілочисельного програмування, яку пропонується розв'язувати дискретним алгоритмом глобального пошуку (ДАГП), описаного в [30]. Тобто необхідно знайти структурний вектор d^* такий, що

$$f(d^*) = \min_{d \in \Sigma} f(d) = \|Y_B - X_B S \hat{\theta}_A\|^2 \quad (7)$$

де S – діагональна $m \times m$ матриця, така, що $S = \text{diag}(d_1, \dots, d_m)$; $\hat{\theta}_A$ – оцінка вектора параметрів для структури d , що визначається на навчальній вибірці (X_A, Y_A) , Σ – множина усіх структурних векторів. Вектор θ_A оцінюється для кожної структури $d \in \Sigma$ за методом найменших квадратів.

Ідейною основою методу ДАГП є інформаційно-статистичний підхід, в якому невідома функція f розглядається як реалізація заданого випадкового процесу, завдяки чому здійснюється випадковий пошук [31]. Цей метод призначений для мінімізації одновимірних багатоекстремальних функцій, заданих у вузлах скінченного інтервалу, тобто для вирішення завдання вигляду: знайти $i^* \in I$ таке, що виконується рівність:

$$f(i^*) = \min_{i \in I} f(i), \quad (8)$$

де I – задана множина номерів вузлів, $I = \{1, \dots, l\}$, а функція $f(i)$ обмежена зверху константою L і задовольняє умову: $|f(i) - f(j)| \leq L|i - j|$, де $i, j \in I$.

Для застосування методу ДАГП багатовимірною задачею (7) зводиться до одновимірної (8), яка розв'язується на десяткових числах, еквівалентних двійковим числам – векторам $d \in \Sigma$. Для обчислення коефіцієнтів за МНК використовується метод сингулярного розкладу.

Гібридний перебірний алгоритм COMBI-GA. В цьому алгоритмі для знаходження глобально оптимальної структури моделі використовується біологічно-інспірований генетичний алгоритм випадкового глобального пошуку [31, 32].

Генетичний алгоритм – це один з метаевристичних алгоритмів глобальної оптимізації, сконструйований в результаті узагальнення та імітації в штучних системах таких властивостей живої природи, як природний відбір, пристосовність до змінних умов середовища, успадкування нащадками життєво необхідних властивостей від батьків і тому подібне. Основні принципи GA були сформульовані Джоном Голландом [33] і описані в багатьох роботах.

Довільний генетичний алгоритм має такі характеристики: $P_0 = (a_1^0, \dots, a_M^0)$ – початкова популяція розміру M ; a_i^0 – претендент на розв'язок задачі оптимізації, представлений у вигляді хромосоми довжиною L ; F – цільова функція (ЦФ); G – множина генетичних операторів; s – правило завершення алгоритму. Хромосоми складаються з генів, які зазвичай кодуються значеннями з двійкового алфавіту $\{0, 1\}$, проте можна використовувати й інші алфавіти – буквені, десяткові тощо. У цій роботі розглядається лише двійкове кодування хромосом.

Вхідними даними для будь-якого GA є початкова популяція – скінченна множина хромосом (елементів, особин, екземплярів тощо) P_0 , кожна з яких представляє можливий варіант розв'язку задачі. Далі з батьківських хромосом в P_0 за допомогою певних генетичних правил утворюється перша популяція нащадків P_1 , аналогічно з популяції P_1 утворюється наступна популяція P_2 і так далі. Процес продовжується доти, поки не буде виконано задане правило завершення алгоритму. Важливою особливістю роботи GA є те, що з кожним кроком певні популяційні характеристики алгоритму, зокрема, середнє значення ЦФ популяції, покращуються.

Алгоритм COMBI-GA [34-38] поетапно формує множину найбільш перспективних структур частинних моделей і знаходить оптимальну з них використовуючи генетичні оператори селекції, кросинговеру та мутації, які визначають механізм їх перебору. Формально цей алгоритм можна описати так:

$$\text{COMBI-GA} = \{Z, y, f, D, CR, M, H, G, k, F\},$$

де $Z[n \times r]$ – матриця вимірювань вектора вхідних змінних модельованого об'єкта, тут r – кількість вхідних змінних, n – кількість точок вимірювань;

$y[n \times 1]$ – вектор вимірювань вихідний змінної модельованого об'єкту;

$f[m \times 1]$ – вектор базисних (опорних) функцій від вхідних змінних, де m – кількість базисних функцій;

D – задане правило розбиття матриці вимірювань $X[n \times m]$ опорного набору аргументів (елементів вектора базисних функцій) і вектора $Y[n \times 1]$ на навчальну і перевірку частини;

CR – зовнішній критерій селекції (як цільова ЦФ або функція пристосовності), заснований на вказаному вище розбитті вибірки (X, Y);

M – розмір популяції двійкових хромосом, тобто закодованих структур частинних моделей;

H – розмір поточної популяції з кращих моделей, $H < m$;

G – множина генетичних операторів;

k – критерій зупинки GA;

F – кількість кращих частинних моделей (свобода вибору), які відстежуються на всіх етапах роботи алгоритму, $1 \leq F \leq H$.

Детально цей алгоритм описано в [39], де подаються й обговорюються результати дослідження його ефективності на тестових і реальних задачах.

4. Класифікаційна схема наявних перебірних алгоритмів МГУА

Усі розглянуті способи генерування структур моделей перебірних алгоритмів можна подати у вигляді схеми, зображеної на рисунку.



Рисунок. Класифікаційна схема перебірних алгоритмів

Зі схеми видно, що переважна більшість генераторів структур перебірних алгоритмів є пошуковими. Такі перебірні алгоритми усувають головний недолік комбінаторного алгоритму – повний перебір моделей, здатні розв’язувати задачі великої розмірності, є значно швидшими і потребують, як правило, менше обчислювальних ресурсів.

Висновок

На сьогодні розроблено значну кількість алгоритмів з повним та неповним перебором моделей для підвищення ступеня прикладної ефективності роботи COMBI МГУА. Проте наявних результатів недостатньо для того, щоб вважати повністю вирішеною проблему пошуку найкращого «генератора моделей», тобто такого, якому для успішного розв’язання заданої задачі потрібно перебрати найменшу кількість частинних моделей.

Однією з важливих причин цього є те, що більшість розглянутих алгоритмів неповного перебору побудовані на детермінованих процедурах

пошуку оптимальної моделі. Ці алгоритми часто мають доволі складні схеми пошуку кращої структури моделі, потребують значних зусиль для їх практичної реалізації, область їх успішного застосування зазвичай обмежена задачами з меншою кількістю вхідних змінних у порівнянні з ітераційними алгоритмами МГУА. Проте створення нових, більш ефективних детермінованих пошукових алгоритмів варто продовжувати.

Ефективною альтернативою детермінованим алгоритмам можуть стати пошукові алгоритми з випадковим пошуком, що об'єднують СОМВІ з еволюційними алгоритмами, зокрема з ГА. Такі гібридні алгоритми позбавлені більшості вказаних недоліків детермінованих алгоритмів. Перші результати в цьому напрямку підтверджують ефективність гібридного перебірного алгоритму СОМВІ-ГА. Тому подальші дослідження доцільно спрямувати на конструювання нових гібридних алгоритмів на основі поєднання СОМВІ з еволюційними та іншими алгоритмами обчислювального інтелекту.

Література

1. Степашко В.С. Оптимизация и обобщение схем перебора моделей в алгоритмах МГУА. – Автоматика. – 1979. – № 4. – С. 28-47.
2. Степашко В.С. Комбинаторная программа для структурной идентификации объектов и процессов управления / В кн.: Справочник по типовым программам моделирования / Под ред. А.Г. Ивахненко. – Киев: Техніка, 1980. – С.80-86.
3. Степашко В.С. Комбинаторный алгоритм МГУА с оптимальной схемой перебора моделей // Автоматика. – 1981. – № 3. – С. 31-36.
4. Степашко В.С. Потенциальная помехоустойчивость моделирования по комбинаторному алгоритму МГУА без использования информации о помехах. – Автоматика, 1983. – № 3. – С. 18-27.
5. Ивахненко А.Г., Степашко В.С. Помехоустойчивость моделирования. – Киев: Наук. думка, 1985. – 216 с.
6. Степашко В.С., Єфіменко С.М., Савченко Є.А. Комп'ютерний експеримент в індуктивному моделюванні. – Київ: Наукова думка. – 2014. – 222 с.
7. Степашко В.С. Алгоритмы МГУА как основа автоматизации процесса моделирования по экспериментальным данным. – Автоматика, 1988. – № 4. – С. 44-55.
8. Степашко В.С. Конечная селекционная процедура сокращения полного перебора моделей // Автоматика. – 1983. – № 4. – С. 84-88.
9. Степашко В.С. Костенко Ю.В. Исследование свойств комбинаторно-селекционного (многоэтапного) алгоритма МГУА // Моделирование и

управление состоянием эколого-экономических систем региона. – Сб. науч. тр. – Киев: МНУЦ ИТС НАНУ, 2001. – С. 96–100.

10. Степашко В.С., Костенко Ю.В. Комбинаторно-селекционный алгоритм последовательного поиска модели оптимальной сложности // Праці І Міжн. конф. з індуктивного моделювання, Львів, 20–25 травня, 2002. – Т. 1., ч. 1. – Львів: ДНДП, 2002. – С. 72–76.

11. Ивахненко А.Г., Ивахненко Г.А., Савченко Е.А. Концепция последовательных алгоритмических приближений (спусков) к точному решению интерполяционных задач искусственного интеллекта // Кибернетика и вычислительная техника, 2000. – № 127. – С. 47-58.

12. Ivakhnenko A.G., Ivakhnenko G.A., Savchenko E.A., and Wunsch D. Problems of Further Development of GMDH Algorithms: Part 2 // Pattern Recognition and Image Analysis, Vol. 12, № 1, 2002, pp. 6-18.

13. Ivahnenko A.G., Ivahnenko G.A., Savchenko E.A. GMDH Algorithm for Optimal Model Choice by the External Error Criterion with the Extension of Definition by Model Bias and its Applications to the Committees and Neural Networks // Pattern Recogn. and Image Analysis. – 2002. – **12**, N 4. – P. 347–353.

14. Koshulko O., Koshulko A. Adaptive parallel implementation of the Combinatorial GMDH algorithm // Proceedings of the 2nd International Workshop on Inductive Modelling (IWIM2007), , Czech Rep., 2007. – Prague: Czech Technical University, 2007. – P. 71–74.

15. Koshulko O., Koshulko A. Acceleration of GMDH Combinatorial search with HPC clusters, in Proceedings of the 2nd International Workshop on Inductive Modelling (IWIM2008), Kyiv, Ukraine, 2008. – Kyiv: International Research and Training Centre for Information Technologies and Systems, 2008. – P.164–167.

16. Кошулько О.А., Кошулько Г.А. Багатоетапний комбінаторний алгоритм МГУА для моделювання об'єктів з великою розмірністю // Матеріали міжнар. конф. «Інтелектуальні системи прийняття рішень та проблеми обчислювального інтелекту», Євпаторія, 2009 р. – Херсон: Вид-во ХНТУ, 2009. – Т. 2. – С. 331–332.

17. Koshulko O., Koshulko A. Multistage Combinatorial GMDH algorithm for parallel processing of high-dimensional data // Proceedings of the 3rd International Workshop on Inductive Modelling (IWIM2009), Sep. 14–19th, Krynica, Rzeszow, Poland, 2009. – Rzeszow: University of Technology Publisher. – P. 114–116.

18. Koshulko, O., Koshulko, G.: Validation Strategy Selection in Combinatorial and Multilayered Iterative GMDH Algorithms. In: Proc. Intern. Workshop on Inductive Modeling (IWIM20011), Kyiv, Ukraine, 2011. – Kyiv: International Research and Training Centre for Information Technologies and Systems, 2011. – P. 51–54.

19. Степашко В.С., Коппа Ю.В. Опыт применения системы АСТРИД для моделирования экономических процессов по статистическим данным // Кибернетика и вычислительная техника, 1998. – Вып.117. – С. 24-31.
20. Samoilenko O.A., Stepashko V.S. Combinatorial GMDH algorithm with successive selection of arguments // Proc. of the II Int. Workshop on Inductive Modelling IWIM–2007, 19–23 Sept. 2007, Prague, Czech Republic. – Prague: Czech Techn. Univ., 2007. – P. 139–143.
21. Samoilenko O., Stepashko V. A method of Successive Elimination of Spurious Arguments for Effective Solution of the Search-Based Modelling Tasks. – Proceedings of the II International Conference on Inductive Modelling ICIM-2008, 15–19 September 2008, Kyiv, Ukraine. – Kyiv: IRTC ITS NANU, 2008. – P. 36–39.
22. Самойленко О.А., Степашко В.С. Оптимізація структури моделей методом послідовного відсіювання аргументів // Матеріали міжнар. конф. «Інтелектуальні системи прийняття рішень та проблеми обчислювального інтелекту», 14-19 травня 2008 р., Євпаторія. – Херсон: Вид-во ХНТУ, 2008. – Т. 3., Част. 2. – С. 83–86.
23. Самойленко О.А. Проектування перебірних алгоритмів МГУА як основних компонентів підсистеми моделювання // Індуктивне моделювання складних систем: Зб. наук. пр. – К.: МННЦ ІТС НАН та МОН України, 2011. – Вип. 3. – С. 191-208.
24. Самойленко О.А., Степашко В.С. Аналіз ефективності застосування частотного критерію в алгоритмі послідовного відсіювання неінформативних аргументів // Індуктивне моделювання складних систем: Зб. наук. праць. – Випуск 4. – К.: МННЦ ІТС НАНУ, 2012. – С. 198–205.
25. Самойленко А.А. Весовой критерий определения информативности аргументов в методах последовательной селекции переменных // УСиМ. – 2013. – № 2. – С. 46-33.
26. Светальский Б.К., Ковальчук П.И. Многорядный алгоритм МГУА с селекцией первичных аргументов. – Автоматика. – 1979. – № 4. – С. 31-34.
27. Кнышов Г.В., Настенко Е.А., Кондрашова Н.В., Носовец Е.К., Павлов В.А. Комбинаторный алгоритм построения параметрического пространства признаков для классификации многомерных моделей // Кибернетика и системный анализ. – 2014. – Т. 50. – № 4. – С. 162-169.
28. Павлов В.А., Коновал А.О. Рекуррентный аддитивно-мультипликативный многоэтапный алгоритм МГУА для задачи классификации объектов, заданных множествами наблюдений // Індуктивне моделювання складних систем: Зб. наук. праць. – Випуск 7. – К.: МННЦ ІТС НАНУ, 2015.– С. 220-231.
29. Голованов И.Н., Пономаренко В.С. Кузьменко Ю.И. Оптимизационный метод сокращения перебора моделей в комбинаторных алгоритмах МГУА // Автоматика. – № 3. – 1983. – С. 12-17.

30. Стронгин Р.Г. Численные методы в многоэкстремальных задачах. – М.: Наука, 1978. – 240 с.
31. Растрингин Л. А. Случайный поиск // Новое в жизни, науке и технике. Серия «Математика, кибернетика». – М.: Знание, 1979. – №1. – С. 64.
32. Jansen T. Analyzing Evolutionary Algorithms: The Computer Science Perspective // Springer: Natural Computing Series, 2013. – 255 p.
33. Holland J.H. Adaptation in natural and artificial systems. An introductory analysis with application to biology, control, and artificial intelligence. – Michigan: University of Michigan, 1975. – 210 p.
34. Мороз О.Г., Степашко В.С. Огляд гібридних структур МГУА-подібних нейронних мереж та генетичних алгоритмів // Індуктивне моделювання складних систем: Зб. наук. праць. – Випуск 7. – К.: МННЦ ІТС НАНУ, 2015.– С. 173–191.
35. Мороз О.Г., Степашко В.С. Перебірний алгоритм МГУА з генетичним пошуком моделі оптимальної складності // Матеріали Міжнар. конф. «Інтелектуальні системи прийняття рішень і проблеми обчислювального інтелекту», Залізний Порт, 2016. – Херсон: Вид-во ПП Вишемирський В.С., 2016. – С. 297–299.
36. Мороз О.Г. Порівняння ефективності операторів кросинговеру при застосуванні генетичного пошуку кращих моделей в перебірному алгоритмі МГУА // Матеріали VI Всеукраїнської школи–семінару молодих вчених і студентів «Сучасні комп'ютерні інформаційні технології» (АСІТ–2016). – Тернопіль: Вид-во ТНЕУ, 2016. – С. 77–79.
37. Stepashko V., Moroz O. Hybrid Searching GMDH-GA Algorithm for Solving Inductive Modeling Tasks // The 1th IEEE International Conference on Data Stream Mining & Processing. – Lviv, Ukraine, 23–27 August , 2016. – Lviv Polytechnic National University, 2016 – С. 350-355.
38. Мороз О.Г. Ефективність операторів мутації в задачі генетичного пошуку оптимальних моделей в перебірному алгоритмі МГУА // Матеріали VIII Українсько-польської науково-практичної конф. «Електроніка та інформаційні технології» (ЕЛІТ-2016), Львів-Чинадієво, 27-30 серпня 2016 р. – Львів: Видавн. центр ЛНУ ім. Ів. Франка, 2016.– С. 331–332.
39. Мороз О.Г. Переборный алгоритм МГУА с генетическим поиском оптимальной модели // УСиМ, 2016 – №6. – С. 73-79.