УДК519.688
DOI10.15407/fmmit2023.38.147

Еволюційний метод апроксимації функцій дійсними поліномами

КозакО.В.[1], Самотий В. В.[2,3], Павельчак А. Г.[4]

[1] магістр, аспірант, кафедра КСА, Національний університет «Львівська політехніка», вул. С.Бандери 12, 79013, Львів, e-mail:oleh.v.kozak@lpnu.ua

[2] д.т.н., професор, кафедра КСА, Національний університет «Львівська політехніка», вул. С.Бандери 12, 79013, Львів, e-mail: volodymyr.v.samotyi@lpnu.ua

[3] dr hab. inż., profesor, katedra Automatyki i Informatyki, Politechnika Krakowska im. Tadeusza Kościuszki, ul. Warszawska 24, 31155, Kraków, e-mail: vsamotyy@pk.edu.pl

[4] к.т.н., доцент, кафедра КСА, Національний університет «Львівська політехніка», вул. С.Бандери 12, 79013, Львів, e-mail: andrii.h.pavelchak@lpnu.ua

В даній роботі запропоновано гібридний метод визначення коефіцієнтів полінома, степені якого є дійсними числами з використанням генетичного алгоритму (ГА). Вхідною інформацією є набір дискретних значень аргументу і функції. Основний фокус нашого підходу полягає в апроксимації функцій за допомогою дійсних поліномів, які надають більшу гнучкість у різних сценаріях. Наш підхід передбачає двокроковий процес оптимізації. У першому кроці в якості початкового наближення вибираються коефіцієнти полінома з цілими степенями, які обчислені за допомогою ГА. Наступним кроком є знаходження дійсних степенів полінома і уточнення коефіцієнтів апроксимації, що також відбувається за допомогою ГА. Це дало можливість швидко і достатньо точно апроксимувати задану функцію поліномом, степені якого є дійсними числами. Еволюційний характер нашого методу забезпечує адаптивність і здатність долати функціональні перепони, таким чином досягаючи кращої загальної продуктивності апроксимації. Дослідження показали, що у порівнянні зі звичайними поліномами була досягнута значно вища точність апроксимації.

Ключові слова: поліном з дійсними степенями, апроксимація функцій, неперіодичні сигнали, генетичні алгоритми, алгоритми оптимізації, параметрична оптимізація.

Вступ.Існує два способи розв'язування задачі параметричної оптимізації. З математичної точки зору перший спосіб – це аналіз параметричної чутливості [1-3], який характеризує вплив на розв'язок відносно малих збурень параметрів. А другий спосіб це параметрична оптимізація для мінімізації функції розбіжності між отриманим результатом і бажаним [4-5].

Метою параметричної оптимізації є знаходження оптимальних значень змінних розв'язку. Змінні розв'язку повинні бути вибрані таким чином, щоб вони мали значний вплив на продуктивність системи та могли бути змінені в межах заданого діапазону [6]. Важливість змінних розв'язку полягає в тому, що вони є фундаментальними будівельними блоками процесу оптимізації. Правильний вибір і маніпулювання змінними розв'язку може призвести до значних покращень у продуктивності системи [7-8].

Дослідження розв'язків коефіцієнтів полінома і раніше викликало інтерес у дослідників, тому було запропоновано різні вирішення цієї задачі [9-15]. Всі вони підходять для випадків коли графік містить поліном степені якого є цілими числами, але не у випадках коли степені можуть бути дійсними числами.

1. Параметрична оптимізація за допомогою еволюційних алгоритмів

Еволюційні алгоритми (ЕА) — це клас алгоритмів оптимізації, що відтворює процес природного відбору. Вони використовуються для вирішення складних задач оптимізації, які важко розв'язати традиційними методами. ЕА довели свою високу ефективність у вирішенні складних задач оптимізації в різних областях.

Однією з ключових переваг ЕА є їх здатність досліджувати великі простори пошуку та знаходити близькі до оптимальних розв'язки для задач із великим простором пошуку та нелінійністю [16]. Також було показано, що ЕА стійкі до шуму та невизначеності цільової функції, що робить їх придатними для застосування в реальних задачах [17]. Найбільш широко використовувані ЕА включають ГА, еволюційні стратегії і метод рою часток.

Кожен із цих алгоритмів має свої сильні та слабкі сторони, а вибір алгоритму залежить від розв'язуваної задачі [18]. Однією з ключових проблем у використанні ЕА є вибір параметрів алгоритму, таких як розмір популяції, тиск відбору та частота мутації. Продуктивність ЕА може бути дуже чутливою до значень його параметрів, і пошук хороших значень параметрів може бути важким і трудомістким процесом. Для вирішення цієї проблеми було запропоновано кілька методів, наприклад налаштування параметрів і адаптивні оператори [19].

2. Порівняння еволюційних алгоритмів з іншими техніками оптимізації

Незважаючи на те, що ЕА показали вражаючі результати в багатьох реальних програмах, важливо порівняти їх продуктивність з іншими методами оптимізації, щоб краще зрозуміти їхні переваги та обмеження.

Одним із найпоширеніших методів оптимізації є градієнтний метод. Цей підхід дуже ефективний, коли цільова функція є гладкою та неперервною, але він часто не може знайти глобальний оптимум у дуже нелінійних, мультимодальних та розривних просторах задач [20]. Крім того, він вимагає аналітичного обчислення похідних цільової функції, що може бути дорогим для обчислення і іноді нездійсненним. Натомість обчислення похідних числовими методами завжди приводить до значних похибок, що робить неможливим їх використання.

Іншим широко використовуваним методом оптимізації є метод локального пошуку. Цей підхід шукає найкращий розв'язок шляхом ітеративного вдосконалення даних початкових умов за допомогою пошуку сусідства [21]. Хоча методи локального пошуку можуть бути дуже ефективними у пошуку глобального оптимуму в певних задачах, але вони можуть легко залипнути в локальних оптимумах, що призведе до неоптимальних рішень.

Порівняно з цими методами оптимізації ЕА мають кілька переваг, які роблять їх придатними для широкого кола проблем. По-перше, ЕА — це алгоритми на основі популяції, які підтримують різноманітний набір варіантів рішень, що дає можливість їм досліджувати простір пошуку ефективніше, ніж методи локального пошуку [22]. По-друге, ЕА не вимагають жодних попередніх знань про цільову функцію чи її похідні, що робить їх придатними до широкого спектру проблем, у тому числі тих, які включають функції чорної скриньки або зашумлені дані. По-третє, ЕА можуть легко обробляти багатоцільові проблеми оптимізації, де цільова функція має кілька конфліктуючих критеріїв, які потрібно оптимізувати одночасно [23].

3. Постановка задачі

В якості прикладу розглянуто згасаючу синусоїду

$$y = \exp(-x)\cos(\pi x). \tag{1}$$

графік якої наведено на рисунку 1. Необхідно знайти значення коефіцієнтів поліному, які максимально наближено відтворюють зображений графік.

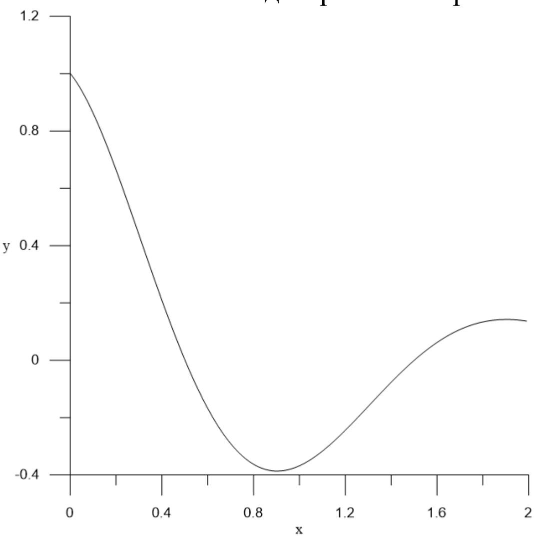

Рис. 1. Згасаюча синусоїда

Розглянемо кубічний поліном

$$y = a_0 + a_1 x + a_2 x^2 + a_3 x^3. \tag{2}$$

для апроксимації залежності (1) на інтервалі $x \in [0, 2]$. Це рівняння вносить певні обмеження і фактично не гарантує, що можливо знайти такі коефіцієнти a_0, a_1, a_2 та a_3 які би дозволили знайти рішення наближене до шуканого. Тому було вирішено змінити рівняння полінома для того, щоб можна було компенсувати цей недолік:

$$y = a_0 + a_1 x + a_2 x^2 + a_3 x^3. \tag{3}$$

У формулі (3) замінено фіксовані степені 1, 2 та 3 на n_1, n_2 та n_3 відповідно. Після цієї заміни степені перестають бути лише цілими числами, а й можуть бути дійсними. Відповідно задача зводиться тепер не тільки до пошуку коефіцієнтів a_0, a_1, a_2 та a_3, а і до пошуку степенів n_1, n_2 та n_3, які б дали можливість знайти максимально наближений розв'язок.

4. Покрокова реалізація генетичного алгоритму

ГА — це клас алгоритмів оптимізації, які імітують процес природного відбору та еволюції [24]. Нижче наведено загальні кроки виконання ГА [25]:

1. Ініціалізація: створення початкової популяції.
2. Оцінка пристосованості: оцінка придатності кожної особи в популяції на основі того, наскільки добре вона вирішує проблему.
3. Відбір: вибір найпридатніших особин із популяції, які будуть батьками для наступного покоління.
4. Схрещування: об'єднання генетичного матеріалу двох батьків для створення нових рішень для нащадків.
5. Мутація: внесення випадкових змін у генетичний матеріал нащадків для збільшення різноманітності популяції.
6. Заміна: заміна найменш придатних особин у поточній популяції новими нащадками.
7. Зупинка алгоритму: зупинка алгоритму, коли буде знайдено задовільне рішення або досягнуто максимальної кількості поколінь.

Однак потрібно враховувати, що на швидкодію ГА впливає вибір параметрів, таких як розмір популяції, шанс мутації, типи відбору та схрещування [16].

5. Використання генетичних алгоритмів для апроксимації неперіодичних сигналів

Для тестування обрано поліном з цілими степенями (2). Необхідно виконати апроксимацію функції (1) формулою (2). Вибрано чотири значення з функції (1) – перше значення ($x_1 = 0, y_1 = 1$), останнє значення ($x_4 = 2, y_4 = 0.13534$) і два екстремуми ($x_2 = 0.902, y_2 = -0.38668$), ($x_3 = 1.902, y_3 = 0.14225$). В результаті отримано систему чотирьох лінійних, алгебричних рівнянь

$$\begin{cases} y_1 = a_0 + a_1 x_1 + a_2 x_1^2 + a_3 x_1^3 \\ y_2 = a_0 + a_1 x_2 + a_2 x_2^2 + a_3 x_2^3 \\ y_3 = a_0 + a_1 x_3 + a_2 x_3^2 + a_3 x_3^3 \\ y_4 = a_0 + a_1 x_4 + a_2 x_4^2 + a_3 x_4^3 \end{cases}$$

Розв'язком цієї системи рівнянь методом Гауса є значення коефіцієнтів: a_0=1, a_1=-3.917, a_2=3.376 та a_3=-0.816. Тому поліном (2) для даного випадку набуде вигляду

$$y = 1 - 3.917x + 3.376x^2 - 0.816x^3 \tag{4}$$

Вираз (4) вибрано тестовою задачею для ГА. В подальшому приймаємо $a_0 = 1$, щоб забезпечити збіг першої точки ($x_1 = 0, y_1 = 1$). Для проведення дослідження використано python бібліотеку pygad, яка містить реалізацію ГА. Далі у виразі (4)x^2 та x^3 замінюємо на x^{n_2} та x^{n_3} відповідно

$$y = 1 - 3.917x + 3.376x^{n_2} - 0.816x^{n_3} \tag{5}$$

Завдання ГА полягає у тому, щоб знайти степені n_2 та n_3, які є невідомими у формулі (5). Як очікуване значення до якого ГА буде наближатися є кубічний поліном із формули (4), тому очікуваними значеннями є n_2=2 та n_3=3. Для оцінки придатності особи в популяції потрібно створити фітнес функцію на основі якої ГА коригуватиме наступні покоління. Запишемо фітнес функцію, яка оцінює придатність особи згідно апроксимаційної дельти між розрахованим для конкретної особи популяції та очікуваним розв'язком

$$fitness = \sum_i |y_i - z_i| \tag{6}$$

де y_i– розраховане значення на кроці *i(4)*; z_i – розраховане значеннядля особина кроці *i(5)*.

Вибрано інтервали пошуку $n_2 \in$ [1, 3] і $n_3 \in$ [2, 4]. В обох випадках використовувалися дійсні числа. Найкраще рішення, яке знайшов ГА для розв'язку рівняння вказаного у формулі (5), є n_2 =2.002 та n_3 =3.005, проти очікуваних 2 та 3 відповідно.Значення фітнес функції склало $fitness = 0.132$.Обчислено відносну похибку, де кількість точок складає 200 і максимальне значення по $y = 1$, яка склала $\delta = 0.066\%$ для розрахованого значення.

$$\delta = \frac{\sum_i |y_i - z_i|}{200}$$

Відомо, що ГА імітують процес природного відбору і еволюції з певною похибкою, тому що цей процес є доволі випадковим. Беручи до уваги отримані результати, тестування ГА вважається успішним, що дає підстави перейти до вирішення основної задачі.

6. Вирішення задачі

Апроксимація функції поліномом з дійсними степенями полягає у обчисленні коефіцієнтів полінома, які максимально точно відтворюватимуть функцію (1).Спочатку застосовано загальний підхід в якому всі коефіцієнти апроксимації a_1, a_2, a_3 та дійсні степені поліному n_1, n_2, n_3 обчислювалися за допомогою ГА. Принагідно зазначимо, що $a_0 = 1$. Для всіх коефіцієнтів задано однакові інтервали [-5, 5], а налаштування фітнес функції (6) були такими ж як і на етапі тестування. В результаті обчислено коефіцієнти апроксимації$a_1 = -1.682, a_2 =$

2.603, $a_3 = -2.312$ та дійсні степені поліному $n_1 = 3.743$, $n_2 = 3.323$, $n_3 = 0.927$.

Отримана залежність наведена на рисунку 2. Відносна похибка склала $\delta = 5.76\%$, а значення фітнес функції склало $fitness = 11.533$. Для розв'язку системи рівнянь отриманого в (4) відносна похибка склала $\delta = 9.02\%$, а значення фітнес функції $fitness = 18.054$. Таким чином розв'язок знайдений генетичним алгоритмом є кращим ніж отриманий розв'язок системи рівнянь.Однак як можна побачити на рисунку 2, найкраще рішення яке вдалося отримати таким способом має місце для покращення. Знайдене рішення на початку має дуже велике відхилення від згасаючої синусоїди. Основна проблема в тому, що для ГА в цьому випадку важко знайти одночасно всі типи коефіцієнтів, які би добре підходили під вирішення задачі, через дуже велику варіативність, враховуючи що навіть невеликі коливання степені може доволі сильно вплинути на результат.

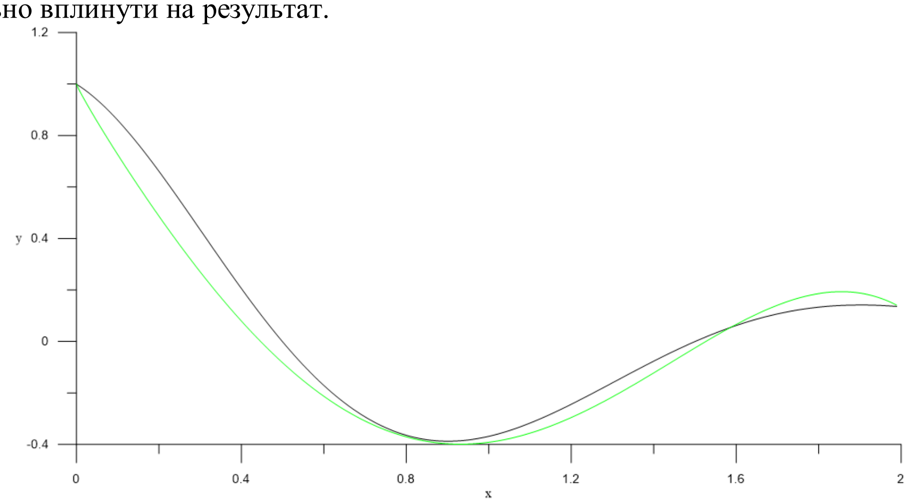

Рис. 2. Результати пошуку коефіцієнтів за допомогою ГА

Так як такий підхід не зміг досягнути бажаного результату, підхід до вирішення задачі було змінено на комбінований, який складається з двох кроків. Перший крок передбачає пошук a_1, a_2 та a_3 коефіцієнтів з формули (2), а другий крок передбачає пошук коефіцієнтів степенів n_1, n_2 та n_3 з формули (3).

Отже переходимо до першого кроку. Для пошуку коефіцієнтів використано такі ж налаштування ГА і фітнес як і під час тестування. Для коефіцієнтів a_1, a_2 та a_3 інтервал був заданий у межах [-5, 5]. Після виконання програми отримано наступний результат:

$$y = 1 - 3.782x + 3.239x^2 - 0.782x^3 \quad (7)$$

У формулі (5) записано знайдене рішення за допомогою розрахунку системи рівнянь методом Гауса і у формулі (7) за допомогою ГА. Для того, щоб вирішити, яке рішення обрати для проведення другого кроку було визначено відносну похибку обох рішень, вона склала 9.02% і 8.12% відповідно. Так як рішення яке знайшов ГА отримало меншу похибку було обрано його.

Тепер переходимо до другої частини підходу. Коли коефіцієнти a_0, a_1, a_2 та a_3 відомі, завдання зводиться до того, щоб знайти коефіцієнти степенів n_1, n_2 та n_3 з формули (3). Для цього потрібно провести наступну ітерацію пошуку. Для коефіцієнтів степенів задано наступні інтервали $n_1 \in [0.5, 1.5]$, $n_2 \in [1.5, 2.5]$ і $n_3 \in [3, 4]$. Якщо подивитися на ці інтервали тоді виникає запитання, чому у n_3 інтервал не відповідає [2.5, 3.5], так як це би узгоджувалося з іншими інтервалами коефіцієнтів степенів. Причина у тому, що під час пошуку рішень було визначено, що найкращі рішення n_3 досягали верхньої межі для n_3 коефіцієнта степені, тому межі для цього коефіцієнта було змінено, щоб покращити рішення і це спрацювало.

В результаті ми отримали наступні коефіцієнти$a_0 = 1$, $a_1 = -3.782$,$a_2 = 3.239$, $a_3 = -0.782$ і коефіцієнти степенів $n_1 = 1.335$, $n_2 = 2.459$, $n_3 = 3.546$. На рисунку 3 зображено результат проведеної оптимізації за допомогою ГА, чорним кольором позначено згасаючу синусоїду і зеленим рішення знайдене за допомогою ГА. Як бачимо після проведення оптимізації графік став плавнішим у порівнянні із минулими результатами. В результаті отримано фінальне вирішення задачі для якого відносна похибка становить $\delta = 3.01\%$ та значення фітнес функції склало $fitness = 6.037$:

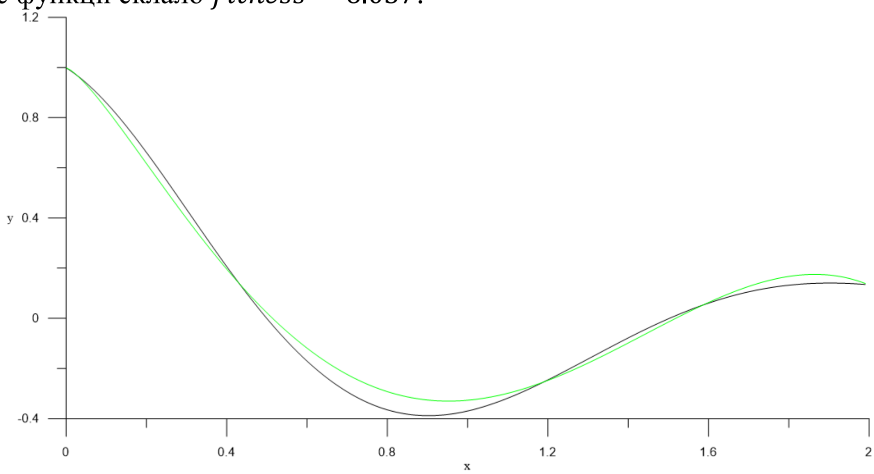

Рис. 3. Порівняння оптимізованого за допомогою ГА і очікуваного рішень

Висновки. В роботі реалізовано новий метод апроксимації нелінійних алгебричних функцій поліномами з дійсними степенями. Для обчислення коефіцієнтів апроксимації і степенів полінома використано ГА.Дослідження показали, що доцільно почергово виконувати обчислення,так як досягається більша точність у порівнянні із одночасним пошуком коефіцієнтів апроксимації і степенів полінома. На першому кроці задаємо степені, що відповідають кубічному поліному. Далі з допомогою ГА визначаємо коефіцієнти апроксимації кубічного поліному. На другому кроці знайдені коефіцієнти апроксимації вважаємо сталими. Далі застосовуючи ГА обчислюємо степені поліному як дійсні числа. Цей алгоритм можна повторити, а саме степені поліному знайдені на попередньому кроці вважаємо сталими і, застосовуючи ГА уточнюємо коефіцієнти апроксимації. Таке розпаралелення обчислень покращує збіжність ГА і підвищує точність апроксимації.

Тим не менш, в подальшому потрібно врахувати наступні шляхи удосконалення:

- Впровадження гібридизованих еволюційних алгоритмів задля покращення якості знайдених рішень та оптимізації продуктивності ЕА [26-27].
- Модифікація функції фітнесу задля зменшення кількості обчислень, що в загальному також впливає на продуктивність.
- Оптимізація налаштувань ГА, таких як: кількість популяції, кількість генерацій та критерію зупинки алгоритму у випадку якщо значення фітнесу не змінюється протягом певної кількості генерацій.

Література

[1] Chemmangat K., Ferranti F., Knockaert L., Dhaene T. (2011). Parametric Macromodeling for Sensitivity Responses From Tabulated Data. IEEE Microwave and Wireless Components Letters. vol. 21, no. 8, pp. 397-399.

[2] Gu Y., Wang X., Gao P., Li X. (2021). Mechanical Parametric Sensitivity Analysis of High-Speed Permanent Magnet Synchronous Machine. IEEE Transactions on Applied Superconductivity. vol. 31, no. 8, pp. 1-4.

[3] Chen H.and Lee C. H. T. (2019). Parametric Sensitivity Analysis and Design Optimization of an Interior Permanent Magnet Synchronous Motor. IEEE Access. vol. 7, pp. 159918-159929.

[4] Grancharova A., Johansen T.A. (2012). Explicit Nonlinear Model Predictive Control. Springer Berlin, Heidelberg, pp. 301.

[5] Pahner U., Hameyer K. and Belmans R. (1999). A parallel implementation of a parametric optimization environment-numerical optimization of an inductor for traction drive systems. IEEE Transactions on Energy Conversion, vol. 14, no. 4, pp. 1329-1334.

[6] Rao S. S. (2019). Engineering Optimization Theory and Practice. John Wiley & Sons, Inc, pp. 798.

[7] Arora J. S. (2016). Introduction to Optimum Design. (Fourth Edition), Elsevier, pp. 968.

[8] Deb K. (2012). Optimization For Engineering Design: Algorithms And Examples. (Second Edition), Phi, pp. 421.

[9] Blinn J.F. (2006). How to solve a cubic equation. Part 1. The shape of the discriminant. IEEE Computer Graphics and Applications, vol. 26, no. 3, pp. 84-93.

[10] Blinn J.F. (2006). How to Solve a Cubic Equation, Part 2: The 11 Case. IEEE Computer Graphics and Applications vol. 26, no. 4, pp. 90-100.

[11] Blinn J.F. (2006). How to Solve a Cubic Equation, Part 3: General Depression and a New Covariant. IEEE Computer Graphics and Applications, vol. 26, no. 6, pp. 92-102.

[12] Blinn J.F. (2007). How to Solve a Cubic Equation, Part 4: The 111 Case. IEEE Computer Graphics and Applications, vol. 27, no. 1, pp. 100-103.

[13] Blinn J.F. (2007). How to Solve a Cubic Equation, Part 5: Back to Numerics. IEEE Computer Graphics and Applications, vol. 27, no. 3, pp. 78-89.

[14] Strobach P. (2011). Solving cubics by polynomial fitting. Journal of Computational and Applied Mathematics, vol. 235, no. 9, pp. 3033-3052.

[15] Flocke N. (2015). Algorithm 954: An Accurate and Efficient Cubic and Quartic Equation Solver for Physical Applications. ACM Transactions on Mathematical Software, vol. 41, no. 4, no. 30, pp. 1-24.
[16] Deb K. (2009). Multi-objective optimization using evolutionary algorithms. John Wiley & Sons, pp. 544.
[17] Yang X. S. (2020). Nature-Inspired Optimization Algorithms. (Second Edition), Elsevier, pp. 310.
[18] Mirjalili S., A. Lewis. A. (2016). The Whale Optimization Algorithm. Advances in Engineering Software, vol. 95, pp. 51-67.
[19] Jin Y. (2011). Surrogate-assisted evolutionary computation: Recent advances and future challenges. Swarm and Evolutionary Computation, vol. 1, no. 2, pp. 61-70.
[20] Nocedal J., Wright S. J. (2006). Numerical optimization. (Second Edition), Springer Science & Business Media, pp. 664.
[21] Glover F., Laguna M. (1998). Tabu search. Springer Science & Business Media, pp. 382.
[22] Eiben A. E., and Smith J. E. (2015). From evolutionary computation to the evolution of things. Nature, vol. 521, 476-482.
[23] Deb K., Pratap A., Agarwal S., Meyarivan T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182-197.
[24] Holland J. H. (1992). Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. The MIT press, pp. 232.
[25] Simon D. (2013). Evolutionary optimization algorithms. John Wiley & Sons, Inc, pp. 784.
[26] Fister I., Mernik M., Brest J. (2013). Hybridization of Evolutionary Algorithms. In: Kita E (ed) Evolutionary algorithms, InTech, pp 3–26.
[27] Grosan C. and Abraham A. (2007). Hybrid Evolutionary Algorithms: Methodologies, Architectures, and Reviews. Hybrid Evolutionary Algorithms, Studies in Computational Intelligence, Springer, vol. 75. pp. 1-17.

Evolutionary method of functions approximation by real polynomials

Oleh Kozak, Volodymyr Samotyi, Andriy Pavelchak

This paper proposes a hybrid method for determining the coefficients of a polynomial whose power coefficients are real numbers using a genetic algorithm (GA). The input is a set of discrete values of the function arguments. The main focus of our approach is to approximate functions using real polynomials, which provide more flexibility compared to cubic polynomials. Our approach involves a two-step optimization process. In the first step, the power coefficients of the polynomial are equal to cubic polynomial powers. Then approximation coefficients of the cubic polynomial are calculated using GA. In the second step, instead of cubic polynomial is introduced polynomial with real powers. In this step the approximation coefficients of polynomial are set as constant and power coefficients of polynomial are calculated using GA to refine the solution. This makes it possible to quickly and accurately approximate a given function with a polynomial whose powers are real numbers. The evolutionary nature of the method ensures adaptability and the ability to overcome functional obstacles, thus achieving better overall approximation performance. Research has shown that, compared to conventional polynomials, significantly higher approximation accuracy has been achieved.

Keywords: polynomial with real powers, approximation of functions, non-periodic signals, genetic algorithms, optimization algorithms, parametric optimization.

Отримано 16.12.23