УДК 004.8:004.032.26

КЛАСТЕРИЗАЦІЯ МАСИВІВ ДАНИХ НА ОСНОВІ МОДИФІКОВАНОГО АЛГОРИТМУ СІРОГО ВОВКА

Шафроненко А. Ю. – канд. техн. наук, доцент, доцент кафедри інформатики, Харківський національний університет радіоелектроніки, Харків, Україна.

Бодянський Є. В. – д-р техн. наук, професор, професор кафедри штучного інтелекту, Харківський національний університет радіоелектроніки, Харків, Україна.

Головін О. О. – д-р техн. наук, с.н.с., заступник начальника Центрального науково-дослідного інституту озброєння та військової техніки Збройних Сил України, Київ, Україна.

АНОТАЦІЯ

Актуальність. Задача кластеризації масивів багатовимірних даних, основною метою якої є знаходження однорідних у сенсі прийнятої метрики класів спостережень, є важливою частиною інтелектуального аналізу даних Data Mining. З обчислювальної точки зору задача кластеризації перетворюється у проблему пошуку локальних екстремумів багатоекстремальної функції, які багатократно запускаються з різних точок вихідного масиву даних. Пришвидшити процес пошуку цих екстремумів можна, скориставшись ідеями еволюційної оптимізації, що включає в себе алгоритми, інспіровані природою, ройові алгоритми, популяційні алгоритми, тощо.

Мета. Мета роботи полягає у запровадженні процедури кластеризації масивів даних на основі покращеного алгоритму сірого вовка.

Метод. Введено метод кластеризації масивів даних на основі модифікованого алгоритму сірого вовка. Перевагою запропонованого підходу є скорочення часу вирішення оптимізаційних задач в умовах коли кластери перетинаються. Особливістю запропонованого методу є обчислювальна простота і висока швидкість, пов'язана з тим, що весь масив обробляється тільки один раз, тобто виключається необхідність в багатоепоховому самонавчанні, що реалізується в традиційних алгоритмах нечіткої кластеризації.

Результати. Результати експериментів підтверджують ефективність запропонованого підходу в задачах кластеризації за умов перетинних кластерів та дозволяють рекомендувати запропонований метод для використання на практиці для вирішення проблем автоматичної кластеризації великих даних.

Висновки. Введено метод кластеризації масивів даних на основі покращеного алгоритму сірого вовка. Перевагою запропонованого підходу є скорочення часу вирішення оптимізаційних задач. Результати експериментів підтверджують ефективність запропонованого підходу в задачах кластеризації за умов перетинних кластерів.

КЛЮЧОВІ СЛОВА: нечітка кластеризація, багатоекстремальна оптимізація, еволюційний метод.

АБРЕВІАТУРИ

FCM – метод нечітких с-середніх;

HSCI – гібридні системи обчислювального інтелекту;

PCO – алгоритм «рій частинок»;

CSO – алгоритм зграї котів;

NIC – природньо – інспіровані обчислення;

GWO – алгоритм сірого вовка.

НОМЕНКЛАТУРА

X – матриця набору даних;

E – цільова функція;

N – кількість спостережень;

R – вектор атрибутів;

n – кількість атрибутів;

k – номер вектору-спостереження;

$x(k)$ – вектор-спостереження;

$x_i(k)$ – значення вектора-спостереження за i-м атрибутом;

$x_{i,i2}(k)$ – значення вектора-спостереження за i-м та $i2$-м атрибутами при нормуванні даних в гіперкуб;

i – номер атрибуту вектора-спостереження;

j – номер кластеру;

m – кількість неперетинних класів;

w – вага вовка;

c – центроїд кластера;

c_j – центроїд j-го кластеру;

t – ітерація пошуку;

T – максимальна кількість ітерацій, що задана;

φ – фаззіфікатор;

α – контрольний параметр;

r – випадкове число;

U – рівень належності спостереження до кластеру;

U_j – рівень належності спостереження до j кластеру;

$U_j(k)$ – рівень належності k-го вектора – спостереження до j-го кластера;

$U_j^{\varphi}(k)$ – рівень належності k-го вектора-спостереження до j-го кластера при заданому рівні розмитості (фаззіфікатора) φ кластерів, що перетинаються;

A, B, C – коефіцієнти поведінки оточення;

GW – вектор позиції сірого вовка;

DOI 10.15588/1607-3274-2023-1-7

$GW(t)$ – вектор позиції сірого вовка в поточній ітерації t;

α - , β - та δ – вовки-домінанти.

ВСТУП

Задача кластеризації масивів багатовимірних даних, основною метою якої є знаходження однорідних у сенсі прийнятої метрики класів спостережень, є важливою частиною інтелектуального аналізу даних Data Mining [1–3]. В рамках традиційного кластерного аналізу апріорі передбачається, що кожен вектор спостереження може належати тільки одному класу-кластеру, хоча в реальних даних досить часто виникає ситуація, коли це спостереження з різними рівнями належності (можливості, ймовірності) відноситься відразу до кількох кластерів, що взаємно перетинаються. Подібна ситуація є предметом розгляду нечіткого (фаззі -) кластерного аналізу [4–5], в рамках якого необхідно оцінити не тільки факт належності кожного спостереження до конкретних класів, але і дати кількісну оцінку рівня цієї належності. З обчислювальної точки зору можна відзначити, що найбільш адекватним математичним апаратом для вирішення задач кластеризації є методи штучного інтелекту [6–8] і, перш за все, нейронні мережі, нечіткі системи, еволюційна оптимізація та, так звані, гібридні системи обчислювального інтелекту.

Об'єкт дослідження кластеризація даних на основі покращеного алгоритму сірого вовка.

Предмет дослідження процедура онлайн кластеризації даних в умовах кластерів що перетинаються на основі модернізованої еволюційної оптимізації.

Мета роботи полягає у запровадженні процедури кластеризації масивів даних на основі покращеного алгоритму сірого вовка.

1 ПОСТАНОВКА ЗАВДАННЯ

Вихідною інформацією для вирішення задачі кластеризації традиційно є матриця спостережень

$$X = \{x(1), x(2), ..., x(k), ..., x(N)\},$$

$x(k) = \{x_i(k)\} \in R^n$, при цьому дані попередньо відцентровано на гіперкуб так, що $x(k) = \{x_{i,i_2}(k)\} \in R^{n_1 \times n_2}$. Така ситуація може виникати у випадку обробки масивів зображень.

2 ОГЛЯД ЛІТЕРАТУРИ

На сьогодні крім методів нечіткої кластеризації таких як FCM, розроблено безліч методів і алгоритмів нечіткої класифікації зі своїми достоїнствами і недоліками, всі вони дозволяють відшукати тільки локальний екстремум прийнятої цільової функції [5, 9], що веде до того, що використання процедур оптимізації (нелінійного програмування) на основі похідних прийнятого критерію в загальному випадку не дозволяє отримати найкраще шукане рішення. Подолати цю проблему можна, багаторазово вирішуючи задачу за різних початкових умов і вибираючи найкращий варіант із безлічі отриманих. Зрозуміло, що подібний підхід суттєво збільшує час вирішення задачі.

Подолати зазначені труднощі можна, скориставшись апаратом гібридних систем обчислювального інтелекту (HSCI) [7, 9–11], що поєднують в собі навчання штучних нейронних мереж, інтерпретованість результатів і можливість роботи в умовах класів, що перетинаються, систем непарного виведення і високу швидкість відшукання глобального екстремуму, що забезпечується еволюційними алгоритмами оптимізації, заснованими на «роях частинок» (PCO).

Традиційно алгоритми оптимізації поділяються на дві частини: детерміновані алгоритми та стохастичні алгоритми [13]. Доведено, що детерміновані алгоритми легко потрапляють в локальні оптимуми, в той час як стохастичні алгоритми здатні уникати локальних розв'язків випадковим чином. Таким чином стохастичні алгоритми набули широкого розвитку, зокрема презентацій, удосконалень і застосувань природньо-інспірованих обчислень (NIC).

Однією з найважливіших частин алгоритмів NIC є так звані біонічні алгоритми, і більшість яких є метаевристичними [13–15]. Вони можуть вирішувати проблеми з паралельними обчисленнями та глобальним пошуком. Метаевристичні алгоритми поділяють рої на глобальний і локальний пошук за допомогою деяких методів. NIC алгоритми не можуть гарантувати глобальні оптимальні рішення; таким чином, більшість метаевристичних алгоритмів вводять випадковість, щоб уникнути локальних оптимумів. Індивідуумами в зграях керують, щоб розділяти, вирівнювати та об'єднувати за допомогою випадковості; їх поточні швидкості складаються з попередніх швидкостей, випадкових множників частоти [16] або евклідових відстаней положень конкретних індивідів [17–21]. Деякі покращення зроблено за допомогою модифікації ваг інерції, хаосу та бінарних векторів, тощо. Більшість із цих удосконалень призводить до трохи кращої продуктивності конкретних алгоритмів, але загальні структури залишаються незмінними.

Більшість метаевристичних алгоритмів та їх удосконалення наразі базуються безпосередньо на поведінці організмів, таких як пошук, полювання [18], запилення [19] та спалах [20].

Метаевристичні алгоритми працюють за схожими цільовими функціями, та досягають кращої продуктивності та зменшення ймовірності потрапити в пастку локальних оптимумів, уникнути випадкових блукань або польотів за допомогою введення додаткових умов для індивідів. Здебільшого це означає, що зграї поводитимуться більш неконтрольованими способами. Крім того, як організми, що живуть у зграях в природі, більшість із них мають соціальну ієрархію. Наприклад, у мурашиній колонії королева є командиром, незважаючи на її репродуктивну роль; динергати – це солдати, які займаються садівництвом колонії, тоді як

DOI 10.15588/1607-3274-2023-1-7

ергати займаються будівництвом, збиранням і розведенням.

3 МАТЕРІАЛИ І МЕТОДИ

В основі поширеного алгоритму ймовірної нечіткої кластеризації лежить процедура мінімізації цільової функції

$$E\left(U_j(k), c_j\right) = \sum_{k=1}^{N}\sum_{j=1}^{m} U_j^{\varphi}(k)\left\|x(k)-c_j\right\|^2 \qquad (1)$$

при обмеженнях

$$\sum_{j=1}^{m} U_j(k) = 1, \quad 0 \le \sum_{j=1}^{m} U_j(k) \le N, \qquad (2)$$

(тут φ – невід'ємний параметр фазифікації (фаззифікатор), що задає розмитість границь між кластерами), в основі якого лежать стандартні методи нелінійного (при $\varphi = 2$ – квадратичного) програмування.

Вирішуючи задачу нелінійного програмування, отримуємо імовірнісний алгоритм нечіткої кластеризації

$$\begin{cases} U_j(k) = \dfrac{\left(\left\|x(k)-c_j\right\|^2\right)^{-1}}{\sum\limits_{l=1}^{m}\left(\left\|x(k)-c_j\right\|^2\right)^{-1}}, \\ c_j = \dfrac{\sum\limits_{k=1}^{N} U_j^2(k)\,x(k)}{\sum\limits_{k=1}^{N} U_j^2(k)}. \end{cases} \qquad (3)$$

В [5] була показана збіжність процесу (3) до локального мінімуму, при цьому досягнення глобального екстремуму в загальному випадку не гарантується.

В роботах [22–23] задача умовної оптимізації (1), (2) була переформульована в задачу безумовної оптимізації цільової функції виду

$$E\left(c_j\right) = \sum_{k=1}^{N}\left(\sum_{j=1}^{m}\left\|x(k)-c_j\right\|^{2(1-\varphi)}\right)^{1-\varphi}, \qquad (4)$$

при $\varphi = 2$

$$E\left(c_j\right) = \sum_{k=1}^{N}\left(\sum_{j=1}^{m}\left\|x(k)-c_j\right\|^{-2}\right)^{-1}. \qquad (5)$$

Таким чином, задача нечіткої кластеризації може бути зведена до пошуку глобального екстремуму цільових функцій (4), (5).

Для вирішення задачі можуть бути використані еволюційні біоінспіровані «роєві» процедури оптимізації [9–11], серед яких в якості одного з найбільш швидкодіючих можна відзначити, так званий, алгоритм сірого вовка (GWO) [24].

За даними Мірджалілі [24], сірі вовки живуть разом і полюють групами. Процес пошуку та полювання можна описати так: (6) якщо видобуток знайдено, вони спочатку вистежують, переслідують і наближаються до неї; (7) якщо здобич біжить, тоді сірі вовки переслідують, оточують і спостерігають за здобиччю, поки вона не перестане рухатися; (8) нарешті починається атака.

Стандартний алгоритм GWO. Алгоритм імітує поведінку пошуку і полювання на здобич сірих вовків в зграї. В математичній моделі найкращій результат вовка в зграї називається альфа (α), а другий найкращий є бета (β), і, отже, третій найкращій називається дельта (δ). Інші рішення кандидатів зграї омегами (ω). Всі омеги будуть керуватися цими трьома сірими вовками під час пошуку (оптимізації) та полювання.

Коли жертва знайдена, починається ітерація (t=1). Згодом α- , β- та δ-вовки керуватимуть ω, щоб переслідувати здобич і, зрештою, оточити її. Три коефіцієнти A, B і C пропонуються для опису поведінки оточення:

$$\begin{aligned} C_\alpha &= \left|B_1 * GW_\alpha - X(t)\right|, \\ C_\beta &= \left|B_2 * GW_\beta - X(t)\right|, \\ C_\delta &= \left|B_3 * GW_\delta - X(t)\right|, \end{aligned} \qquad (6)$$

де t вказує на поточну ітерацію, GW вектор позиції сірого вовка, GW_1, GW_2 і GW_3 – є векторами положення α- , β- та δ-вовків, що обчислюється наступним чином:

$$\begin{aligned} GW_1 &= GW_\alpha - A_1 * C_\alpha, \\ GW_2 &= GW_\beta - A_2 * C_\beta, \\ GW_3 &= GW_\delta - A_3 * C_\delta, \end{aligned} \qquad (7)$$

$$GW(t) = \frac{GW_1 + GW_2 + GW_3}{3}. \qquad (8)$$

Параметри A та B є комбінаціями керуючого параметра α та випадкових чисел r_1 та r_2 [24]:

$$\begin{aligned} A &= 2\alpha r_1 - \alpha, \\ B &= 2r_2. \end{aligned} \qquad (9)$$

Контрольний параметр α замінюється значенням параметра A і, нарешті, змушує омега-вовків наближатися або тікати від домінуючих вовків, таких як альфа, бета та дельта. Якщо $|A| > 1$, сірі вовки втікають від домінантів, а це означає, що омега-вовки вте-

DOI 10.15588/1607-3274-2023-1-7

чуть від здобичі та досліджуватимуть більше простору, що в оптимізації називається глобальним пошуком. Та якщо $|A|<1$ вони наближаються до домінант, а значить δ-вовки будуть слідувати за домінантами, які наближаються до здобичі, і це називається локальним пошуком в оптимізації.

Контрольний параметр α визначається як лінійне зниження від максимального значення 2 до 0 під час ітерацій:

$$\alpha = 2\left(1-\frac{t}{T}\right),$$

де t – номер ітерації, T – максимальна кількість ітерацій, що задана.

Схематично представити роботу алгоритму можна наступним чином (рис. 1).

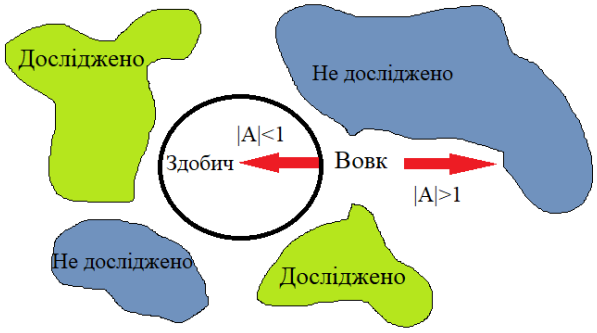

Рисунок 1 – Схема роботи алгоритму GWO

Блок-схема алгоритму сірих вовків наведена на рис. 2.

Багато алгоритмів ройового інтелекту імітують поведінку полювання та пошуку деяких тварин. Однак GWO моделює внутрішню ієрархію керівництва вовків, таким чином, в процесі пошуку позиція найкращого рішення може бути комплексно оцінена-трьома рішеннями. Але для інших алгоритмів ройового інтелекту, найкраще рішення шукається лише на основі одного рішення – локального оптимума.

Отже, GWO може значно зменшити ймовірність передчасного потрапляння в локальний оптимум. Щоб досягти належного компромісу між розвідкою та полюванням, пропонується покращений GWO.

Розглядаючи рівняння (8) видно, що в процесі пошуку, однакову роль відіграють домінанти. Кожен із сірих вовків зграї наближається або тікає в пошуку здобичі. Однак, слід зауважити, що найближче до здобичі домінанти із середньою вагою альфа, ніж бета і дельта. Таким чином, на початку процедури пошуку в рівнянні (8) слід враховувати лише положення альфа, або його вага має бути набагато більшою, ніж ваги інших домінант. Таким чином, рівняння (8) можна переписати у вигляді:

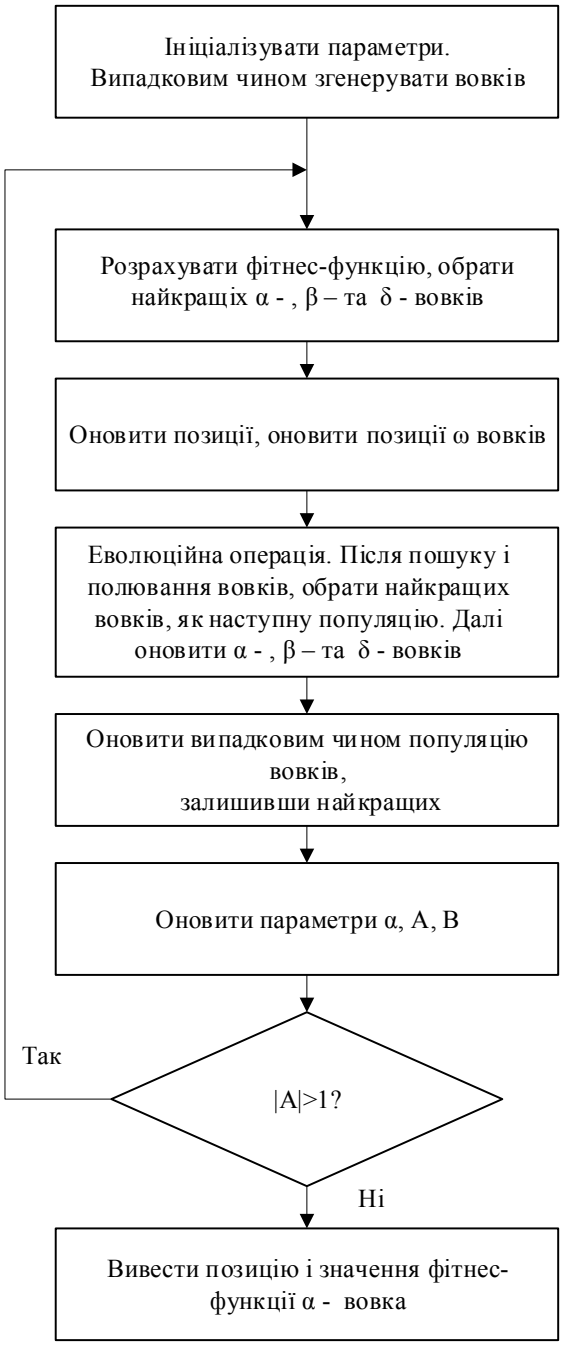

Рисунок 2 – Блок-схема алгоритму GWO

$$GW(t+1)=\frac{w_1GW_1+w_2GW_2+w_3GW_3}{3}, \qquad (10)$$

де $w_1+w_2+w_3=1$, при w_1 – вага α – вовка, w_2 – вага β-вовка, w_3 – вага δ-вовка, при цьому $w_1 \geq w_2 \geq w_3$. На першій (або $t=0$) ітерації пропонується задати ваги результатами алгоритму кластеризації за рівнянням (3), де:

$$\begin{aligned} c_1 &= w_\alpha; \\ c_2 &= w_\beta; \text{ при } t=0 \\ c_3 &= w_\delta; \end{aligned} \qquad (11)$$

DOI 10.15588/1607-3274-2023-1-7

Тоді ми можемо визначити, що ваги змінних задовольняють гіпотезі про соціальну ієрархію функцій сірих вовків та їх пошукову поведінку.

4 ЕКСПЕРИМЕНТИ

Дослідження методу кластеризації масивів даних на основі покращеного алгоритму сірого вовка (FGWO) проводились на двох багатоекстремальних функціях, наведених в табл. 1.

Таблиця 1 – Тестові функції

Назва функції	Формула	Інтервал
Растрігін	$f(x) = 20 + x^2 + y^2 - 10\cos(2\pi x) + \cos(2\pi y)$	$[-5.12; 5.12]$
Гріванг	$f(x) = \frac{1}{4000}x + \frac{1}{4000}y - \cos\left(\frac{x}{\sqrt{1}}\right)\cos\left(\frac{x}{\sqrt{2}}\right) + 1$	$[-30; 30]$

Якість роботи запропонованого метода (FGWO) порівнювалось із декількома класичними алгоритмами кластеризації, еволюційними процедурами, а також модифікованими методами кластеризації на основі оптимізаційних процедур, а саме алгоритм оптимізації рою частинок (PSO), алгоритм зграї котів (CSO), класичний алгоритм сірого вовка (GWO) та модифікованого алгоритму кластеризації на основі зграї котів (FCSO) [25–26]. Для кожного метода, задано 30 агентів, що шукають оптимум в багатоекстремальній функції.

5 РЕЗУЛЬТАТИ

Перш за все перевіримо роботу запропонованого метода з його модифікацією, тобто використання вагів для кожного вовка. Результат зміни ваг продемонстровано на Рисунку 3. Аналізуючи отриманий графік залежності зміни ваг кожного вовка від кількості ітерацій, можна зробити висновок, що запропонований підхід є сприятливим для подальшого аналізу методу кластеризації масивів даних на основі покращеного алгоритму сірого вовка.

На рис. 4 та рис. 5 показане графічне порівняння методів та їх збіжності за функціями Растригіна та Грінварга відповідно.

6 ОБГОВОРЕННЯ

Аналізуючи результати отриманих експериментальних досліджень та порівняльного аналізу роботи методу кластеризації масивів даних на основі покращеного алгоритму сірого вовка із методами кластеризації, що базуються як на класичному підході до кластеризації даних, так і більш екзотичних, запропонований метод демонструє достатньо високі результати.

Основними перевагами запропонованого методу полягає в простоті математичних розрахунків, швидкості роботи з даними, незалежно від виду, розміру та якості вибірки, що аналізується. Слід відзначити точність роботи метода кластеризації даних на основі покращеного алгоритму сірого та отриманих результатів кластеризації, що досягається за допомогою оптимізаційної процедури еволюційного алгоритму.

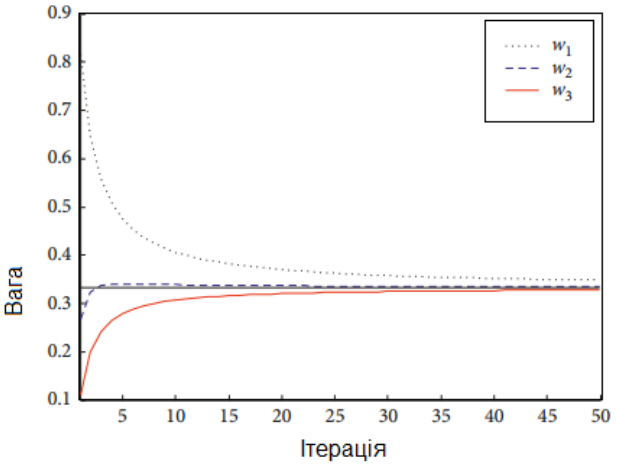

Рисунок 3 – Залежність зміни ваги вовків від кількості ітерацій

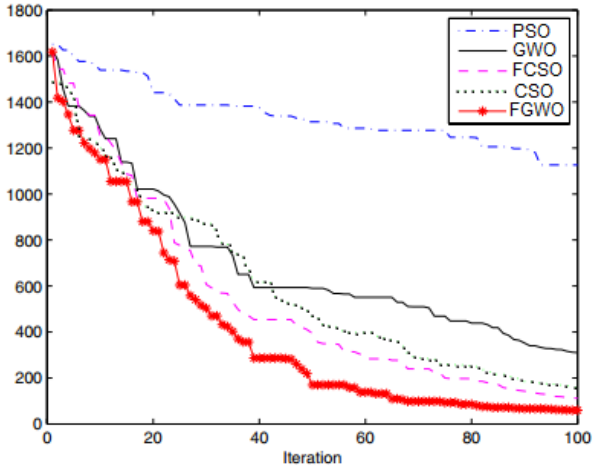

Рисунок 4 – Криві збіжності функції Растригіна

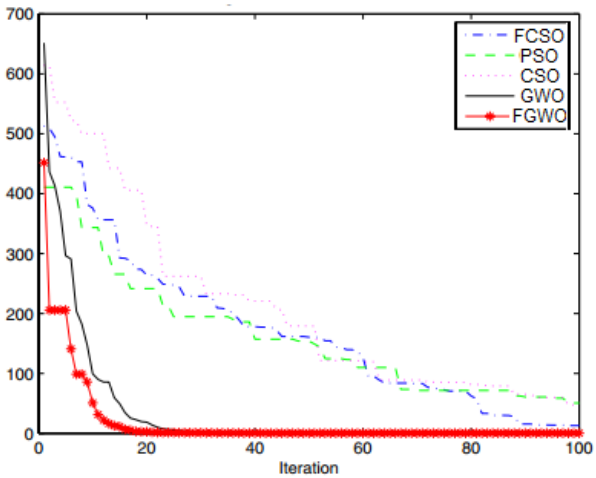

Рисунок 5 – Криві збіжності функції Грінванга

DOI 10.15588/1607-3274-2023-1-7

ВИСНОВКИ

Введено метод кластеризації масивів даних на основі покращеного алгоритму сірого вовка. Перевагою запропонованого підходу є скорочення часу вирішення оптимізаційних задач. Результати експериментів підтверджують ефективність запропонованого підходу в задачах кластеризації за умов перетинних кластерів.

Наукова новизна: вперше запропонований метод кластеризації масивів даних на основі покращеного алгоритму сірого вовка.

Практичне значення: результати експерименту дозволяють рекомендувати запропонований метод для використання на практиці для вирішення проблем автоматичної кластеризації багатоекстремальних даних різної природи.

Перспективи подальших досліджень методи нечіткої кластеризації даних для широкого класу практичних проблем.

ПОДЯКА

Робота виконана в рамках науково-дослідного проєкту державного бюджету Харківського національного університету радіоелектроніки «Розробка методів та алгоритмів комбінованого навчання глибинних нейро-нео-фаззі систем за умов короткої навчальної вибірки».

ЛІТЕРАТУРА

1. Gan G. Data Clustering: Theory, Algorithms and Applications/ G. Gan, Ch. Ma, J. Wu. – Philadiphia, Pensilvania: SIAM: 2007. – 455 p. doi: https://doi.org/10.1137/ 1.9780898718348
2. Abonyi J. Cluster Analisis for Data Mining and System Identification / J. Abonyi, D. Feil. – Basel : Birlhause, 2007 – 303p.
3. Xu R. Clustering/ R. Xu, D. C. Wunsch. – Hoboken N.J. : John Wiley & Sons, Inc., 2009. – 398p.
4. Fuzzy Clustering Analysis: Methods for Classification, Data Analisys and Image Recognition / [Höppner F., Klawonn F., Kruse R., Runkler T.] – Chichester: John Wiley &Sons, 1999. – 300 p.
5. Fuzzy models and algorithms for pattern recognition and image processing / Bezdek J. C. et al. – Springer Science & Business Media, 1999. – T. 4.
6. Engelbrecht A. Computational intelligence: an introduction / A. Engelbrecht. – Sidney: John Wiley & Sons, 2007. – 597 p.
7. Rutkowski L. Computational Intelligence Methods and Techniques / L Rutkowski. - Berlin Heidelberg : Springer-Verlag, 2008. – 514 p.
8. Kroll A. Computational Intelligence. Eine Einfürung in Problelme, Methoden and Tchnische Anwendungen / A. Kroll. – München : Oldenbourg Verlag, 2013. – 428 p.
9. Bezdek J. C. Fuzzy Models and Algorithms for Pattern Recognition and Image Processing / [J. C. Bezdek, J. Keller, R. Krishnapuram, N. R. Pal]. – N.Y. : Springer Science + Business Media, Inc., 2015. – 776 p.
10. Mumford C. L. Computational Intelligence/ C. L. Mumford, L.C. Jain. – Berlin: Springer-Verlag, 2009. – 729 p.
11. Kroll A. Computational Intelligence. Eine Einfürung in Problelme, Methoden and Tchnische Anwendungen / A. Kroll. – München : Oldenbourg Verlag, 2013 – 428 p.
12. Mirjalili S. The ant lion optimizer / S. Mirjalili // Advances in Engineering Software. – 2015. – Vol. 83. – P. 80–98. doi: https://doi.org/10.1016/j.advengsoft.2015.01.010.
13. Bio-inspired Computation in Telecommunications / [X. S. Yang, S. F. Chen, and T. O. Ting]. –Morgan Kaufmann, Boston, MA, USA, 2015.
14. Syberfeldt A. Real-world simulation-based manufacturing optimizations using cuckoo search / A. Syberfeldt, S. Lidberg // Proceedings of the 2012 Winter Simulation Conference (WSC): Berlin, Germany, December 2012: proceedings. – P. 1–12. doi: 10.1109/WSC.2012.6465158
15. Coelho L. D. S. Improved firefly algorithm approach applied to chiller loading for energy conservation / L. D. S. Coelho and V. C. Mariani // Energy and Buildings. – 2013. – Vol. 59. – P. 273–278. doi: https://doi.org/10.1016/j.enbuild. 2012.11.030
16. Juan Z. The Bat Algorithm and Its Parameters, Electronics, Communications and Networks IV / Z. Juan, G. Zheng-Ming. – CRC Press, Boca Raton, FL, USA, 2015.
17. Yu. J. J. Q. A social spider algorithm for global optimization/ J. J. Q. Yu and V. O. K. Li// Applied Soft Computing. – 2015. – Vol. 30. – P. 614–627. doi: https://doi.org/10.48550/ arXiv.1502.02407
18. Azizi R. Empirical study of artificial fish swarm algorithm / R. Azizi // International Journal of Computing, Communications and Networking. – 2014. – Vol. 3, No. 1–3. – P. 1–7.
19. Yan-Xia L. Improved ant colony algorithm for evaluation of graduates/ L. Yan-Xia, L. Lin, and Zhaoyang//Physical conditions, measuring technology and mechatronics automation (ICMTMA): proceedings of the 2014 Sixth International Conference on Measuring Technology and Mechatronics Automation: Zhangjiajie, China, January 2014. – P. 333–336.
20. A modification of artificial bee colony algorithm applied to loudspeaker design problem/ [Z. Xiu, Z. Xin, S. L. Ho, and W. N. Fu] // IEEE Transactions on Magnetics. – 2014. –Vol. 50, No. 2. – P. 737–740. doi: 10.1109/TMAG.2013. 2281818.
21. Marichelvam M. K. A discrete firefly algorithm for the multiobjective hybrid flowshop scheduling problems/ M. K. Marichelvam, T. Prabaharan, and X. S. Yang // IEEE Transactions on Evolutionary Computation. – 2014. – vol. 18, No. 2. – P. 301–305.
22. Hathaway R. J. Optimization of clustering criteria by reformulation/ R. J. Hathaway, J. C Bezdek// IEEE Transactions Fuzzy Systems. – 1995. – No. 3. – P. 241–245.
23. Pal N. R. Sequental competitive learning algorithm / N. R. Pal, J. C. Bezdek, R. J. Hathaway // Neural Networks. – 1996. – Vol. 9, № 5. – P.787–796.
24. Mirjalili S. M. Grey wolf optimizer / S. M. Mirjalili and A. Lewis // Advances in Engineering Software. – 2014. – Vol. 69. – P. 46–61.
25. Бодянський Є. В. Кластеризація масивів даних на основі комбінованої оптимізації функцій щільності розподілу та еволюційного методу котячих зграй/ Є. В. Бодянський, І. П. Пліс, А. Ю. Шафроненко // Радіоелектроніка, інформатика, управління. – 2022. – № 4. – С. 61–70. doi: 10.15588/1607-3274-2022-4-5
26. Bodyanskiy Y. V. Credibilistic fuzzy clustering based on evolutionary method of crazy cats / Y. V. Bodyanskiy, A. Y. Shafronenko, I. P. Pliss // System Research and Information Technologies. – 2021 (3). – P. 110–119.

Стаття надійшла до редакції 09.01.2023.
Після доробки 13.02.2023.

UDC 004.8:004.032.26

CLUSTERIZATION OF DATA ARRAYS BASED ON THE MODIFIED GRAY WOLF ALGORITHM

Shafronenko A. Yu. – PhD, Assosiated Professor at the Department of Informatics, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

DOI 10.15588/1607-3274-2023-1-7

Bodyanskiy Ye. V. – Dr. Sc., Professor at the Department of Artificial Intelligence, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

Holovin O. O. – Dr. Sc., Senior Reseacher, Deputy Chief of Central Scientific Research Institute of Armament and Military Equipment of Armed Forces of Ukraine, Kyiv, Ukraine.

ABSTRACT

Context. The task of clustering arrays of multidimensional data, the main goal of which is to find classes of observations that are homogeneous in the sense of the accepted metric, is an important part of the intelligent data analysis of Data Mining. From a computational point of view, the problem of clustering turns into the problem of finding local extrema of a multiextreme function, which are repeatedly started from different points of the original data array. To speed up the process of finding these extrema using the ideas of evolutionary optimization, which includes algorithms inspired by nature, swarm algorithms, population algorithms, etc.

Objective. The purpose of the work is to introduce a procedure for clustering data arrays based on the improved gray wolf algorithm.

Method. A method of clustering data arrays based on the modified gray wolf algorithm is introduced. The advantage of the proposed approach is a reduction in the time of solving optimization problems in conditions where clusters are overlap. A feature of the proposed method is computational simplicity and high speed, due to the fact that the entire array is processed only once, that is, eliminates the need for multi-era self-learning, implemented in traditional fuzzy clustering algorithms.

Results. The results of the experiments confirm the effectiveness of the proposed approach in clustering problems under the condition of classes that overlap and allow us to recommend the proposed method for use in practice to solve problems of automatic clustering big data.

Conclusions. A method of clustering data arrays based on the modified gray wolf algorithm is introduced. The advantage of the proposed approach is the reduction of time for solving optimization problems. The results of the experiments confirm the effectiveness of the proposed approach in clustering problems under the conditions of overlapping clusters.

KEYWORDS: fuzzy clustering, multi-extremal optimization, evolutionary method.

REFERENCES

1. Gan G., Ma Ch., Wu J. Data Clustering: Theory, Algorithms and Applications. Philadiphia, Pensilvania, SIAM, 2007, 455 p. DOI: https://doi.org/10.1137/ 1.9780898718348
2. Abonyi J., Feil D. Cluster Analisis for Data Mining and System Identification. Basel, Birlhause, 2007, 303 p.
3. Xu R., Wunsch D. C. Clustering. Hoboken N. J., John Wiley & Sons, Inc., 2009, 398 p.
4. Höppner F., Klawonn F., Kruse R., Runkler T. Fuzzy Clustering Analysis: Methods for Classification, Data Analisys and Image Recognition. Chichester, John Wiley &Sons, 1999, 300 p.
5. Bezdek J. C. et al. Fuzzy models and algorithms for pattern recognition and image processing. Springer Science & Business Media, 1999, T. 4.
6. Engelbrecht A. Computational intelligence: an introduction. Sidney, John Wiley & Sons, 2007, 597 p.
7. Rutkowski L. Computational Intelligence Methods and Techniques. Berlin Heidelberg, Springer-Verlag, 2008, 514 p.
8. Kroll A. Computational Intelligence. Eine Einfürung in Problelme, Methoden and Tchnische Anwendungen. München, Oldenbourg Verlag, 2013, 428 p.
9. Bezdek J. C., Keller J., Krishnapuram R., Pal N. R. Fuzzy Models and Algorithms for Pattern Recognition and Image Processing. N.Y., Springer Science + Business Media, Inc., 2015, 776 p.
10. Mumford C. L., Jain L. C. Computational Intelligence. Berlin, Springer-Verlag, 2009, 729 p.
11. Kroll A. Computational Intelligence. Eine Einfürung in Problelme, Methoden and Tchnische Anwendungen. München, Oldenbourg Verlag, 2013, 428 p.
12. Mirjalili S. The ant lion optimizer, *Advances in Engineering Software*, 2015, Vol. 83, pp. 80–98. doi: https://doi.org/10.1016/j.advengsoft.2015.01.010.
13. Yang X. S., Chen S. F., and Ting T. O. Bio-inspired Computation in Telecommunications. Morgan Kaufmann, Boston, MA, USA, 2015.
14. Syberfeldt A., Lidberg S. Real-world simulation-based manufacturing optimizations using cuckoo search, *Proceedings of the 2012 Winter Simulation Conference (WSC)*. Berlin, Germany, December 2012, proceedings, pp. 1–12. doi: 10.1109/WSC.2012.6465158
15. Coelho L. D. S. and Mariani V. C. Improved firefly algorithm approach applied to chiller loading for energy conservation, *Energy and Buildings,* 2013, Vol. 59, pp. 273–278. doi: https://doi.org/10.1016/j.enbuild. 2012.11.030
16. Juan Z., Zheng-Ming G. The Bat Algorithm and Its Parameters, Electronics, Communications and Networks IV. CRC Press, Boca Raton, FL, USA, 2015.
17. Yu. J. J. Q., Li V. O. K. A social spider algorithm for global optimization, *Applied Soft Computing*, 2015, Vol. 30, pp. 614–627. doi: https://doi.org/10.48550/ arXiv.1502.02407
18. Azizi R. Empirical study of artificial fish swarm algorithm, *International Journal of Computing, Communications and Networking*, 2014, Vol. 3, No. 1–3, pp. 1–7.
19. Yan-Xia L., Lin L., and Zhaoyang Improved ant colony algorithm for evaluation of graduates, *Physical conditions, measuring technology and mechatronics automation (ICMTMA): proceedings of the 2014 Sixth International Conference on Measuring Technology and Mechatronics Automation*. Zhangjiajie, China, January 2014, pp. 333–336.
20. Xiu Z., Xin Z., Ho S. L., and Fu W. N. A modification of artificial bee colony algorithm applied to loudspeaker design problem, *IEEE Transactions on Magnetics*, 2014, Vol. 50, No. 2, pp. 737–740. doi: 10.1109/TMAG.2013. 2281818.
21. Marichelvam M. K., Prabaharan T., and Yang X. S. A discrete firefly algorithm for the multi-objective hybrid flowshop scheduling problems, *IEEE Transactions on Evolutionary Computation*, 2014, Vol. 18, No. 2, pp. 301–305.
22. Hathaway R. J., Bezdek J. C. Optimization of clustering criteria by reformulation, *IEEE Transactions Fuzzy Systems*, 1995, No. 3, P.241–245.
23. Pal N. R., Bezdek J. C., Hathaway R. J. Sequental competitive learning algorithm, *Neural Networks,* 1996, Vol. 9, № 5 pp. 787–796.
24. Mirjalili S. M. and Lewis A. Grey wolf optimizer, *Advances in Engineering Software*, 2014, Vol. 69, pp. 46–61.
25. Bodyanskiy Ye. V., Pliss I. P., Shafronenko A. Yu. Clusterization of data arrays based on combined optimization of distribution density functions and the evolutionary method of cat swarm, *Radio Electronics, Computer Science, Control,* 2022, №4, pp. 61–70. doi: 10.15588/1607-3274-2022-4-5
26. Bodyanskiy Y. V., Shafronenko A. Y. , Pliss I. P. Credibilistic fuzzy clustering based on evolutionary method of crazy cats, *System Research and Information Technologies,* 2021 (3), pp. 110–119.

DOI 10.15588/1607-3274-2023-1-7