УДК 004.8+004.94+338.2
DOI https://doi.org/10.32838/2663-5941/2020.2-1/29

Снитюк В.Є.
Київський національний університет імені Тараса Шевченка

Сорока П.М.
Київський національний університет імені Тараса Шевченка

Ткаченко О.В.
Київський національний університет імені Тараса Шевченка

ПРОБЛЕМА РОЗБИТТЯ РЕГІОНІВ УКРАЇНИ НА КЛАСТЕРИ З МЕТОЮ ПРОВЕДЕННЯ РЕГІОНАЛЬНО ОРІЄНТОВАНОЇ ЕКОНОМІЧНОЇ ПОЛІТИКИ

Задача кластеризації сьогодні є однією з найпопулярніших задач машинного навчання. Вона являє собою задачу розбиття об'єктів на групи так, щоб об'єкти в одній групі були максимально подібні, а об'єкти з різних кластерів істотно відрізнялися. Ця задача полегшує обробку даних та має багато застосувань у реальному житті.

У статті розглядається проблема розбиття регіонів України на кластери з метою проведення регіонально орієнтованої економічної політики. Зроблено математичну постановку задачі, у якій кожен вектор описує конкретний регіон України за певними ознаками. В роботі вважають, шо всі ознаки є рівноважними, тому перед початком розв'язання проводиться нормалізація даних з використанням лінійного перетворення. Наведено функцію, яку мінімізують у задачі кластеризації з використанням нормалізованого набору даних (датасету).

У статті розглядаються такі алгоритми кластеризації: еволюційна стратегія, алгоритм диференціальної еволюції, генетичний алгоритм, імунний алгоритм, K-means, метод найближчого сусіда. Вони використані під час розв'язання задачі розбиття регіонів України на групи за соціально-економічними показниками. Для експерименту було взято 10 показників для кожного із регіонів. У розрахунках використано нормалізований набір даних, побудований за даними з веб-сайту Державної служби статистики України за 2017 рік. У дослідженнях використовуються двомірні проекції цього багатовимірного набору даних та згенерований допоміжний набір даних.

Побудовано графіки залежності результатів мінімального значення функції f_min від кількості ітерацій для алгоритму диференціальної еволюції і генетичного алгоритму. Показано, що при ітераціях більших 250 для алгоритму диференціальної еволюції і більших 80 для генетичного алгоритму мінімальне значення функції f_min стабілізується. Зроблено порівняння та аналіз результатів досліджень від кількості кластерів, одержаних різними алгоритмами. Показано, що за кількості кластерів рівної 5 еволюційні алгоритми дають досить пристойні результати.

Ключові слова: *регіон, кластер, кластеризація, економічна політика, алгоритм, ітерація.*

Постановка проблеми. Задача кластеризації – це задача розбиття об'єктів на групи так, щоб об'єкти в одній групі були максимально подібні, а об'єкти з різних кластерів істотно відрізнялися. Такого роду задача сьогодні є однією з найпопулярніших задач машинного навчання, бо полегшує обробку даних та має багато застосувань у реальному житті.

У цій статті буде розглянуто декілька алгоритмів кластеризації для розбиття регіонів України на групи за соціально-економічними показниками. Таке розбиття полегшує ведення політики та дозволяє визначати напрями розвитку не окремо для кожного регіону України, а для групи регіонів одночасно. Це дозволяє економити людські ресурси, спрямовані на розробку стратегій для кожного регіону, та розробити більш детальну стратегію для групи регіонів. У роботі використано набір даних (датасет), побудований за даними з веб-сайту Державної служби статистики України за 2017 рік [1].

Постановка завдання. Метою статті є дослідження проблеми розбиття регіонів України на кластери з метою проведення регіонально орієнтованої економічної політики.

Виклад основного матеріалу дослідження. Математична постановка задачі. Нехай є N

векторів $x_1, x_2, \ldots, x_n$, де $x_i \in \mathcal{R}^d$ для кожного i з множини $\overline{1,N}$ та d – це розмірність вектору. Кожен вектор описує конкретний регіон України за певними ознаками. Потрібно розбити ці вектори на k груп. В статті будемо вважати, що всі ознаки є рівноважними, тому перед початком розв'язання необхідно провести нормалізацію даних, тобто перевести значення кожної ознаки з $\mathcal{R}$ у інтервал $[a, b]$.

Для цього використаємо наступне лінійне перетворення. $\forall x_i, i \in \overline{1,N}$ побудуємо вектор $\hat{x}_i$ за формулою:

$$\hat{x}_{i,j} = \frac{(x_{ij} - \mathrm{a}_{\mathrm{now_j}})(b-a)}{\mathrm{b}_{\mathrm{now_j}} - a_{\mathrm{now_j}}} + \mathrm{a},$$

де $j \in \overline{1,d}$, $a_{\mathrm{now_j}} = \min\limits_{i \in \overline{1,N}} x_{i,j}$ та $\mathrm{b}_{\mathrm{now_j}} = \max\limits_{i \in \overline{1,N}} x_{i,j}$.

Далі будемо використовувати тільки нормалізований датасет. Щоб не перевантажувати статтю індексами та позначеннями, далі будемо використовувати позначення $\{x_1, x_2, \ldots, x_N\}$ для нормалізованої множини $\{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_N\}$.

Розглянемо функцію, для мінімізації якої буде застосовано алгоритми, що описані далі:

$$f_k(X, y) = \sum_{p=1}^{k} d_{cp}(X_p),$$

де k – це кількість кластерів;

y – це вектор-результат кластеризації, $y \in \left(\overline{1,k}\right)^N$;

X_p – це підмножина множини X, що складається з векторів, які потрапили в p-ий кластер;

$d_{cp}(X_p)$ – середня відстань між векторами у кластері p та центроїдом кластеру, яка обчислюється за формулою:

$$d_{cp}(X_p) = \frac{1}{|X_p|} \sum_{h=1}^{|X_p|} \|x_h^p - c_p\|,$$

де c_p – центроїд кластеру X_p;

$|X_p|$ – кількість елементів у кластері X_p;

x_h^p – h-ий елемент кластеру X_p.

Іноді в задачах кластеризації намагаються не тільки мінімізувати функцію, наведену вище, а також максимізувати функцію відстаней між кластерами, або навіть тільки її.

Норма $\|x_h^p - c_p\|$ – це довільна норма. У статті буде розглянута звичайна евклідова норма, але в загальному випадку норма може бути довільна.

Опис використаних методів кластеризації. Для методів кластеризації еволюційної стратегії, диференціальної еволюції, генетичного та імунного алгоритмів функція $f_k(X, y)$ буде переписана в іншому вигляді. Ми будемо мінімізувати $f(X, c) = f_k(X, y)$, де c – це множина центрів відповідних кластерів. Тоді кожна точка початкової множини буде належати тому кластеру, до якого вона знаходиться найближче всього. Кожен елемент популяції $p_i^j \in \mathbb{R}^{d*k}$ являє собою множину центрів кластерів, розгорнуту в один вектор, тобто перші d компонент вектору – це центр 1-го кластеру, другі d компонент вектору – це центр 2-го кластеру і так далі. Зауважимо, що $f(X, c)$ – це невід'ємна функція. Деякі з формул, що наведені нижче, правильні тільки для такого роду функцій.

Зробимо опис декількох алгоритмів кластеризації, використаних для розбиття регіонів України на групи за соціально-економічними показниками.

Еволюційна стратегія. У ньому використані такі позначення: T – кількість ітерацій; μ – кількість батьків; λ – кількість дітей; d – розмірність простору; k – кількість кластерів; p_i^j – i-ий елемент популяції P_j.

Використаний алгоритм еволюційної стратегії [2] виглядає таким чином:

1: Початкова ініціалізація:

$P_0 = \{p_1^0, \ldots, p_\mu^0\}$,

де $p_i^0 \in [a,b]^{d*k}$ – випадковий вектор

2: **for** $t = 1..T$ **do**

3: Обрати λ індивідуумів з P_t рівноймовірно $\{p_1^{(t)}, \ldots, p_\lambda^{(t)}\}$,

4: Згенерувати множину потомків $\{c_1^t, \ldots, c_\lambda^t\}$, де $c_i^t = p_i^{(t)} + N(0, \sigma^2)$,

5: Обчислимо значення f для потомків та їх батьків

6: P_{t+1} = множина $\propto$ найкращих з батьків та дітей

7: **end for**

Алгоритм диференціальної еволюції. Тут використано такі позначення: T – кількість ітерацій; d – розмірність простору; k – кількість кластерів; N – розмір популяції; $CR \in (0;1)$ – константа; F – константа; $v^{(d_{now})}$ – координата d_{now} вектора v; p_i^j – i-ий елемент популяції P_j.

Використаний алгоритм диференціальної еволюції [3] виглядає таким чином:

1: Початкова ініціалізація:

$P_0 = \{p_1^0, \ldots, p_N^0\}$,

де $p_i^0 \in [a,b]^{d*k}$ – випадковий вектор

2: **for** $t = 1..T$ **do**

3: **for** $j = 1..\mathrm{N}$ **do**

4: Випадково обрати 3 різні представника з P_t : r_0, r_1, r_2

5: dim_random = randint(1, d)
6: **for** d_now = 1..d **do**
7: **if** $\left(\text{rand}[0,1) < \text{CR}\ \textbf{or}\ \text{d}_{\text{now}} == \text{dim}_{\text{random}}\right)$:
8: $u_{\text{d\_now}} = r_1^{(\text{d\_now})} + F\left(r_2^{(\text{d\_now})} - r_0^{(\text{d\_now})}\right)$
9: **else**:
10: $u_{\text{d\_now}} = p_j^{t(d_{now})}$
11: **end for**
12: **if** $\left(f(u) < f\left(p_{\text{j}}^t\right)\right)$:
13: В множину P_t додамо елемент u
14: **else**:
15: В множину P_t додамо елемент p_{j}^t
16: **end for**
17: **end for**

Генетичний алгоритм. У ньому використані такі позначення: $\boldsymbol{T}$ – кількість ітерацій; $\boldsymbol{d}$ – розмірність простору; $\boldsymbol{k}$ – кількість кластерів; $\boldsymbol{N}$ – розмір популяції; $\boldsymbol{\mu}$– точність, використовується для знаходження кількості бітів, необхідних для кодування; $\boldsymbol{\delta}$ – константа, яка описує ймовірність мутації.

Використаний генетичний алгоритм [2] виглядає таким чином:

1: Початкова ініціалізація:
$P_0 = \left\{p_1^0, \ldots, p_N^0\right\}$,
де $p_i^0 \in [a,b]^{d*k}$ – випадковий вектор
2: **for** $t = 1..T$ **do**
3: Обчислимо значення f для елементів популяції
4: Обчислимо значення ймовірності бути обраним для оператора кросовера залежно від якості функції f:

$$probability\_array_i^{temporary} = 1 - \frac{f\left(p_i^t\right)}{\sum_{h=1}^{N} f\left(p_h^t\right)}$$

$$probability_i = \frac{probability\_array_i^{temporary}}{\sum_{h=1}^{N} probability\_array_h^{temporary}}$$

5: **for** $\text{d}_{\text{now}} = 1..\frac{\text{N}}{2}$ **do**

6: Візьмемо з урахуванням ймовірності $probability\_array$ два вектора з P_t r_0, r_1
7: Отримаємо $r_0^{'}$, $r_1^{'}$ після використання оператора кросинговера до r_0, r_1.
8: Застосуємо оператор мутації з ймовірністю δ до $r_0^{'}$, $r_1^{'}$
9: Додамо $r_0^{'}$, $r_1^{'}$ у множину дітей
10: **end for**
11:Обчислимо значення f для потомків та їх батьків
12: P_{t+1} = множина N найкращих з батьків та дітей
13: **end for**

Зауваження: Для використання операторів мутації та кросинговера до p_i^j завжди застосовується двійковий запис числа.

Імунний алгоритм. Тут використано такі позначення: $\boldsymbol{T}$ – кількість ітерацій; d – розмірність простору; k – кількість кластерів; N – розмір популяції; ε – точність, використовується для знаходження кількості бітів, необхідних для кодування; δ – константа, яка описує ймовірність мутації; $\boldsymbol{max}_{\text{вклад унікальності}}$ – константа; $\boldsymbol{k}_{\text{унікальність}}$ та $\boldsymbol{k}_{\text{якість}}$ – константи, відношення між якими залежить від важливості унікальності елемента популяції відносно важливості якості функції.

Використаний імунний алгоритм на основі [4], [5], [6], [7] виглядає таким чином:

1: Початкова ініціалізація:
$P_0 = \left\{p_1^0, \ldots, p_N^0\right\}$,
де $p_i^0 \in [a,b]^{d*k}$ – випадковий вектор
2: **for** $t = 1..T$ **do**
3: Обчислимо значення f для елементів популяції
4: Обчислимо якість кожного з елементів популяції:

$$f_{якість}\left(p_i^t\right) = f\left(p_i^t\right) - f_{\min+1},$$

де $f\_\min$ – мінімальне значення функції для елементів популяції
5: Обчислимо значення унікальності для кожного з елементів популяції:
u_i – унікальності комбінації нулів та одиниць для p_i^t відносно інших елементів популяції обчислюється за формулою:

$$u_i = Pr_1^1 Pr_2^1 * \ldots * Pr_R^1 * \ldots * Pr_1^d Pr_2^d * \ldots * Pr_R^d$$

(R – кількість бітів використано для запису числа у двійковій системі;
Нехай *val*=значення *i*-го біту *j*-го виміру для p_i^t,
а $h = \sum_{q=1}^{N}$значення *i*-го біту *j*-го виміру для p_q^t),
Тоді $Pr_i^j = val * \frac{h}{N} + (1 - val)\left(1 - \frac{h}{N}\right)$

6: Обчислимо значення кількості клонів для кожного елементу з популяції. Кількість клонів прямо пропорціональна якості функції f від цього елементу та унікальності комбінації нулів та одиниць відносно інших елементів популяції:

$$clones_i = \min\left(\frac{k_{унікальність}}{0.001 + u_i}, max_{вклад\ унікальності}\right) + \frac{k_{якість}}{f_{якість}\left(p_i^t\right)}$$

7. Побудуємо P_{temp} – множина, де кожен елемент p_i^t популяції P_t зустрічається $clones_i$ разів

8: **for** $d_{now} = 1..\frac{\text{кількість елементів в } P_{temp}}{2}$ **do**

9: Візьмемо з рівною ймовірністю два елемента з множини P_{temp} два вектори з P_t r_0, r_1

10:Отримаємо r_0', r_1' після використання оператора кросинговера до r_0, r_1.

11:Застосуємо оператор мутації з ймовірністю δ до r_0' та r_1'

12:Додамо r_0', r_1' у множину дітей

13:**end for**

14:Обчислимо значення f від для потомків та батьків

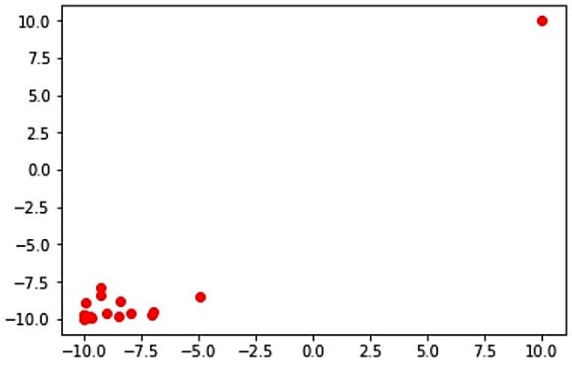
15: P_{t+1} = множина N найкращих з батьків та дітей

16: **end for**

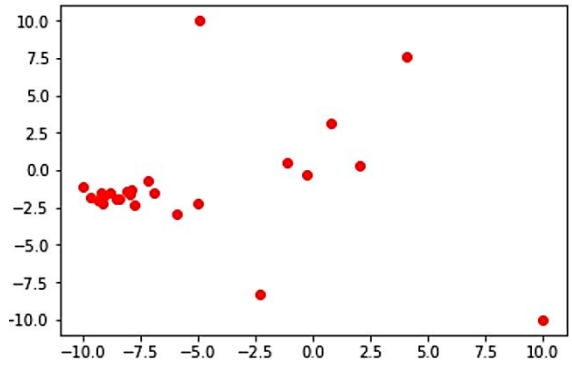
Зауваження: Для використання операторів мутації та кросинговера до p_i^j завжди застосовується двійковий запис числа.

K-means. Для цього алгоритму був використаний стандартний метод із бібліотеки SciPy для мови програмування Python [8]. Згідно з цим алгоритмом на його початку обираються випадковим чином стільки точок у просторі, скільки буде кластерів $C=\{c_1, c_2, ..., c_k\}$. Далі для кожної точки початкової множини, знаходимо найближчу точку з множини С. Після цього початкову множину точок ділимо на k підмножин, кластерів, за найближчою точкою з множини С. Далі кожну точку з множини С замінюємо на центроїд точок відповідного кластеру. Такі дії виконуємо доки точки в множині С не будуть змінюватися, або майже не будуть [9].

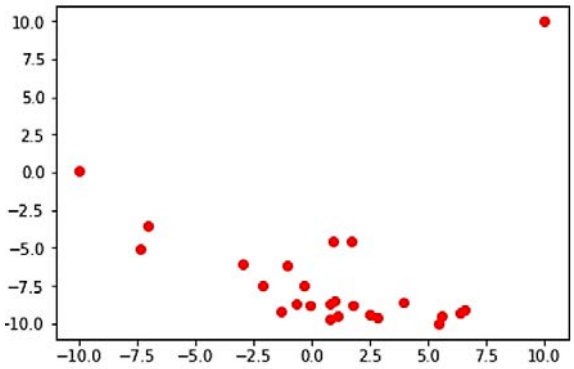
Метод найближчого сусіда або метод одиничного зв'язку. Тут використано стандартний метод



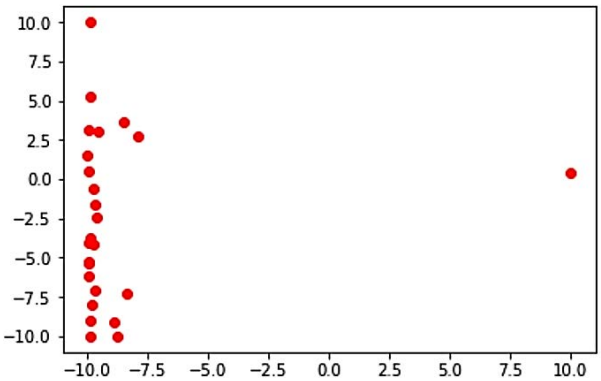
По осі X – Експорт (тис. доларів США)
По осі У – Імпорт (тис. доларів США)



По осі X – Середня чисельність населення (осіб)
По осі У – Міграційний приріст (осіб)



По осі X – Природний приріст (осіб)
По осі У – Доходи (млн. грн.)



По осі X – Імпорт (тис. доларів США)
По осі У – Кількість неформально зайнятого населення (тис. осіб)

Рис. 1. Двовимірні проекції набору даних

із бібліотеки SciPy для мови програмування Python [10]. За цим алгоритмом на його початку кожна точка являє собою окремий кластер. Далі на кожному кроці, кластери з найменшою відстанню між ними поєднуються в один кластер. Такі дії виконуються доки не буде досягнуто бажаної кількості кластерів [11].

Гіперпараметри для експерименту

Гіперпараметри для алгоритмів кластеризації. Для всіх алгоритмів було виконано 2000 ітерацій, тобто $T = 2000$. Кількість кластерів k змінювалосьвід 2 до 9. Розмірністьпростору $d = 10$. Розмір популяції N = 100.

Гіперпараметри для еволюційної стратегії: кількість батьків μ=**N**; кількість дітей λ=**7*N**.

Гіперпараметри для алгоритму диференціальної еволюції: ***CR*=0,5**; ***F*=0,04**.

Гіперпараметри для генетичного алгоритму: розмір популяції **N=100**; точність використовується для знаходження кількості бітів, необхідних для кодування ε=1e-3; ймовірність мутації δ=0,001.

Гіперпараметри для імунного алгоритму. Точність використовується для знаходження кількості бітів, необхідних для кодування ε=1e-3. Ймовірність мутації δ=0,001; $max_{\text{вклад унікальності}}$=**1**; $k_{\text{унікальність}}$=**1**; $k_{\text{якість}}$=**10**.

Зауваження: Значення гіперпараметрів було знайдено експериментальним шляхом.

Результати дослідження.

Формування набору даних. Для кожного регіону України було взято 10 політико-економічних показників: середня чисельність постійного населення у 2017 році (осіб); міграційний приріст, скорочення (осіб); природний приріст, скорочення (осіб); доходи населення по регіонах України (млн грн.); наявний дохід у розрахунку на одну особу (грн.); заборгованість із виплати заробітної плати станом на початок 2018 року (млн грн.); капітальні інвестиції (млн грн); експорт (тис. доларів США); імпорт (тис. доларів США); кількість неформально зайнятого населення віком 15–70 років (тис. осіб). Після нормалізації значення кожного з них стало в інтервалі [-10;10].

З метою проведення розрахунків у роботі використано набір даних, побудований за даними з веб-сайту Державної служби статистики України за 2017 рік [1]. Оскільки цей набір даних багатовимірний, то розглянемо декілька двовимірних його проекцій, отриманих після нормалізації (рис. 1).

Окрім основного набору даних, наведеного вище, було згенеровано двовимірний набір даних з явно вираженими центроїдами. Для цього було випадково згенеровано 7 випадкових двовимірних векторів, причому кожна з координат цих векторів знаходиться в інтервалі [-30;30]. Потім кожен з цих 7 векторів використовується як математичне сподівання для нормального розподілу з σ = 4,5. Генеруємо по 50 точок на кожен з них. Отриманий набір даних є допоміжним, і будемо його використовувати для порівняння властивостей алгоритмів. На рис. 2 й зображено цей допоміжний набір даних.

На рис. 3 показані залежності мінімального значення функції f_min від кількості ітерацій для алгоритму диференціальної еволюції (3а) і генетичного алгоритму (3б). З цього рисунка ми бачимо, що при ітераціях, більших 250 для алгоритму диференціальної еволюції і більших 80

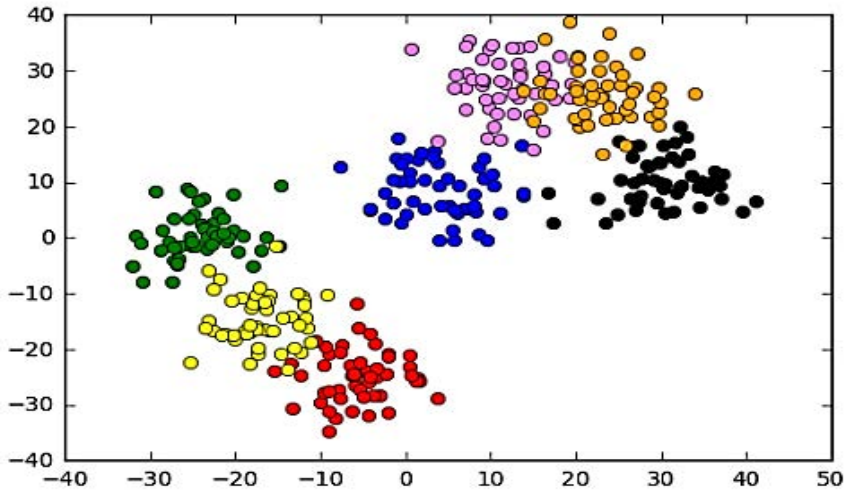


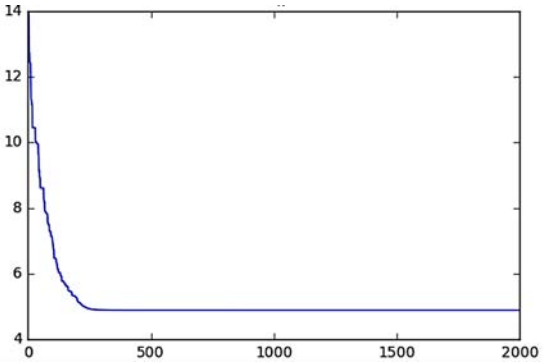
Рис. 2. Допоміжний набір даних

для генетичного алгоритму, мінімальне значення функції f_min стабілізується.

На рис. 4 наведено залежності відношення приросту

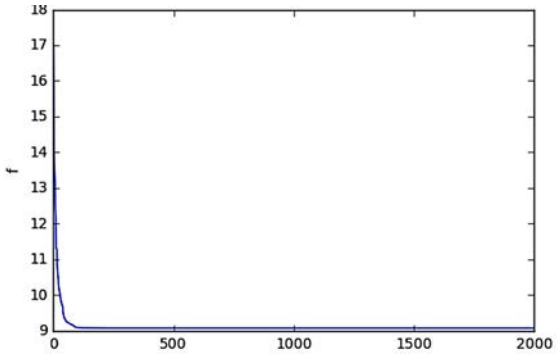
$$\left(\frac{f_{\min(num\_clusters+1)}}{f_{\min(num\_clusters)}} \right)$$

від кількості кластерів, отримане 6 різними алгоритмами. Коли приріст менше за 1, то f_{min} зменшується зі збільшеннями кількості кластерів, тобто розв'язок покращується. Коли приріст більше за 1, то розв'язок погіршується. Коли він рівний 1, то зі збільшенням кількості кластерів розв'язок не покращується, зазвичай це проявляється, коли є кластери без елементів.



По осі X – кількість ітерацій
По осі У – f_min

Рис. 3а. Алгоритм диференціальної еволюції, k=9



По осі X – кількість ітерацій
По осі У – f_min

Рис. 3б. Генетичний алгоритм, k=2

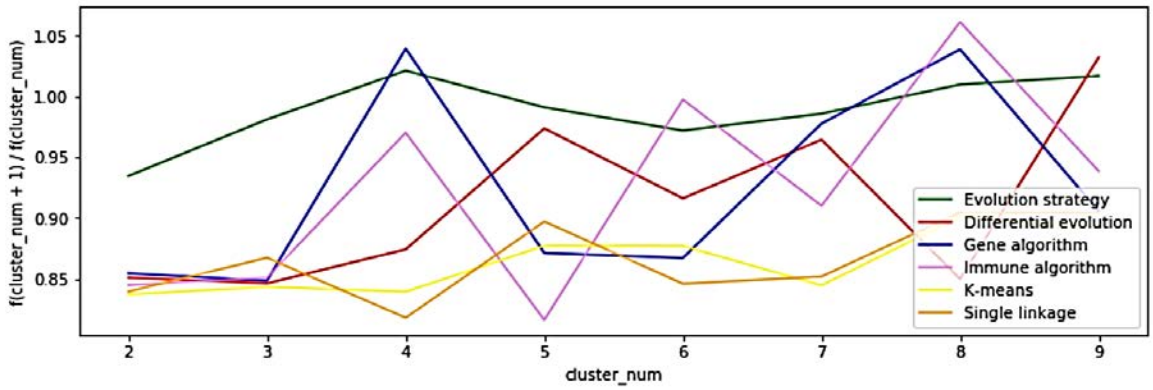


Рис. 4. Відношення приросту від кількості кластерів

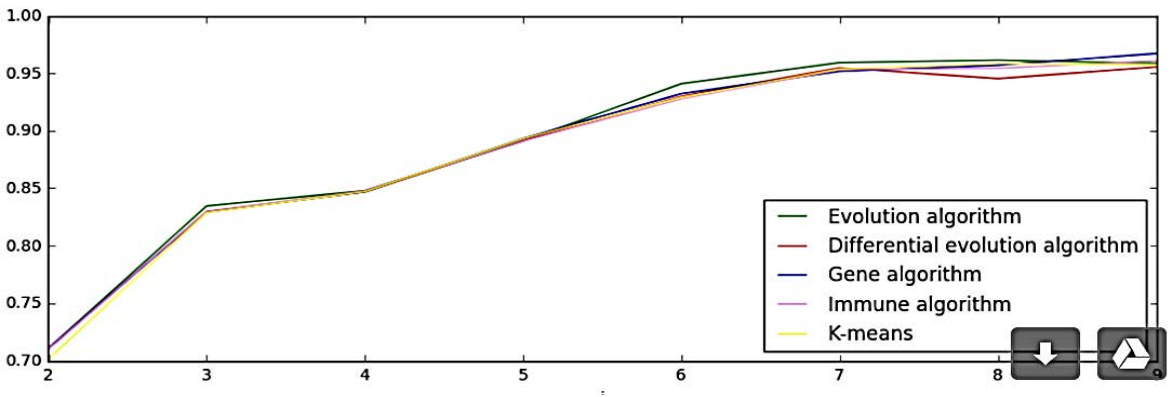


Рис. 5. Приріст на допоміжному наборі даних

Приріст на рис. 4 «стрибає» через те, що набір даних складний для алгоритмів кластеризації з відносно великою кількістю вимірів і невеликою кількістю даних. Розглянемо залежності приросту

$$\left(\frac{f_{\min(num\_clusters+1)}}{f_{\min(num\_clusters)}} \right)$$

від кількості кластерів на допоміжному наборі даних для декількох алгоритмів (рис. 5).

На рис. 6 наведено залежності f_min від кількості кластерів, отримані 6 різними алгоритмами.

Як можна бачити з рис. 5–6, еволюційні методи не встигають знайти максимум. Також про це свідчить те, що з якогось моменту з'являються кластери без елементів. Але для невеликих k (k " 5) результати еволюційних алгоритмів дають досить пристойні результати. Особливо добре себе показав метод диференціальної еволюції. Він є швидким та дає хороші результати.

Загалом розподіли на кластери були хороші, бо регіони, які потрапили в один кластер, і справді були подібні за економічною ситуацією, а ще часто навіть знаходяться поряд. Також це свідчить про те, що датасет, сформований на початку, є якісним. Наприклад, один з розподілів, сформований імунним алгоритмом для кількості кластерів k = 5, виглядатиме таким чином:

1. {Донецька, Луганська}
2. {Дніпропетровська, Київська, Харківська}
3. {Волинська, Житомирська, Закарпатська, Кіровоградська, Миколаївська, Полтавська, Сумська, Тернопільська, Херсонська, Хмельницька, Черкаська, Чернівецька, Чернігівська}
4. {Вінницька, Запорізька, Івано-Франківська, Львівська, Одеська, Рівненська}
5. {м. Київ}

Розглянемо ще декілька двовимірних проекцій отриманих кластеризацій (рис. 7).

Відмітимо, що візуально важко оцінювати якість кластеризації в багатовимірному просторі, тому розглянемо один з результатів на допоміжному наборі даних для k=4 та використаємо генетичний алгоритм (рис. 8).

З рис. 7–8 видно, що генетичний алгоритм показав хороші результати як на основному наборі даних, так і на допоміжному. Причому на допоміжному вже можна оцінити якість і візуально.

При розрахунках було здійснено 2 000 ітерацій, що є, наскільки нам відомо, достатньо багато. Це було зроблено, щоб оцінити можливості алгоритмів на такому специфічному датасеті, коли мало даних та багато вимірів. У протилежному випадку, на інших датасетах, де вимірів було, наприклад, лише 2, але точок більше 1000, еволюційні алгоритми зійшлися за 100 ітерацій. Саме тому одне з покращень результатів, що пропонується в роботі, це використовувати алгоритм лише для одного вектору центрів на одній ітерації, а інші координати вважати фіксованими. На наступній ітерації зафіксувати наступний вектор центрів. Ще одне покращення – це зменшення розмірності простору та використання алгоритмів для отриманих векторів меншої розмірності. Слід звернути увагу, що результати при великій кількості кластерів погані, бо еволюційні алгоритми намагаються відійти від розв'язку при великій кількості напрямків. Але результати хороші вже майже для всіх точок, що вказує на те, що алгоритм рідко може «вгадати» правильний напрямок та покращити результат. Саме тому вектор результатів рідко змінюється, хоча деякі кластери все ще порожні.

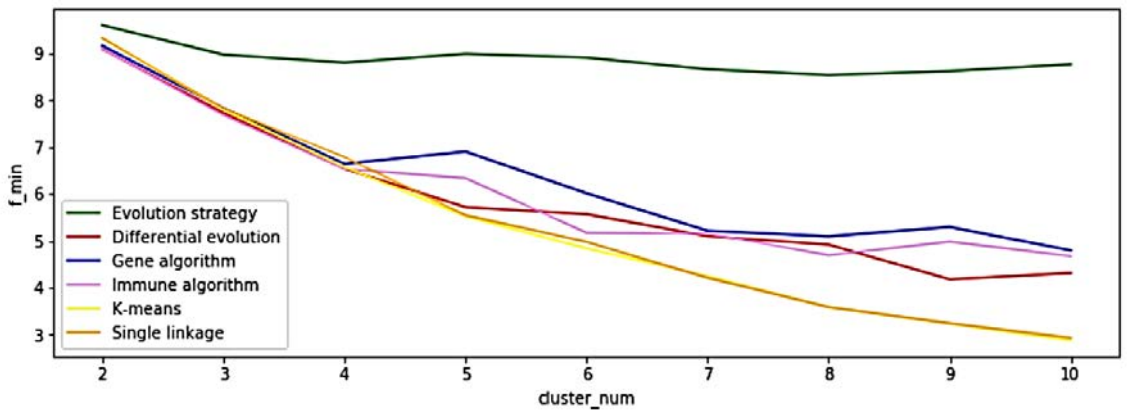
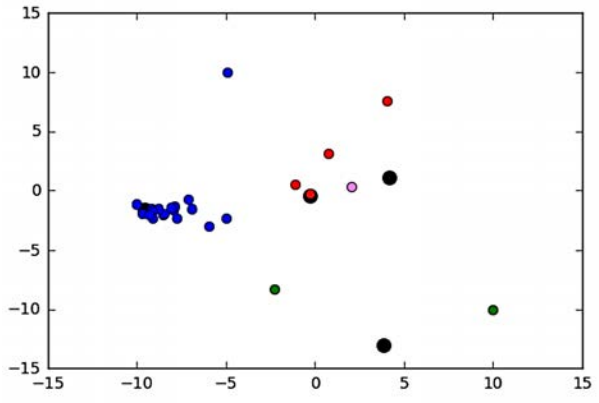
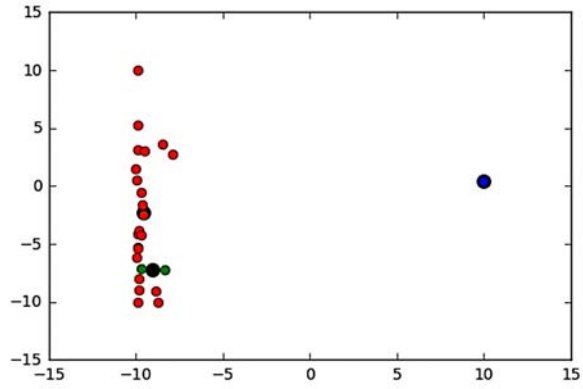


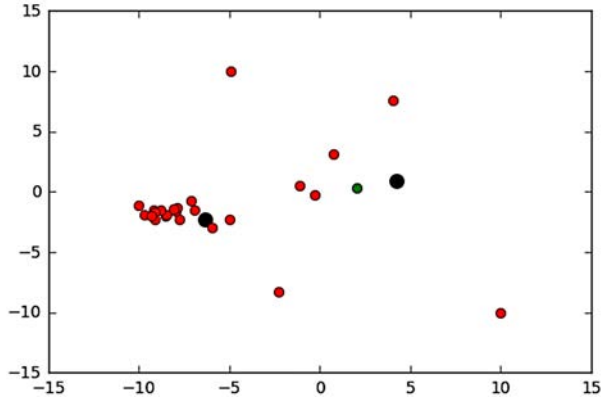
Рис. 6. Зменшення цільової функції



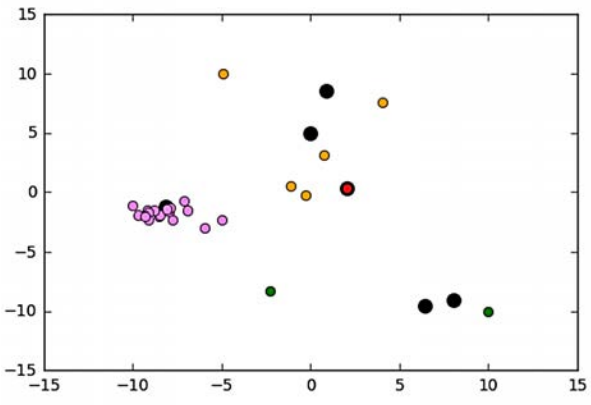
Алгоритм диференціальної еволюції, k=4
По осі X – Середня чисельність населення (осіб)
По осі У – Міграційний приріст (осіб)



Імунний алгоритм, k=3
По осі X – Імпорт(тис. доларів США)
По осі У – Кількість неформально зайнятого населення (тис. осіб)



Еволюційний алгоритм,k=2
По осі X – Середня чисельність населення (осіб)
По осі У – Міграційний приріст (осіб)



Генетичний алгоритм, k=6
По осі X – Середня чисельність населення (осіб)
По осі У – Міграційний приріст (осіб)

Рис. 7. Результати кластеризації

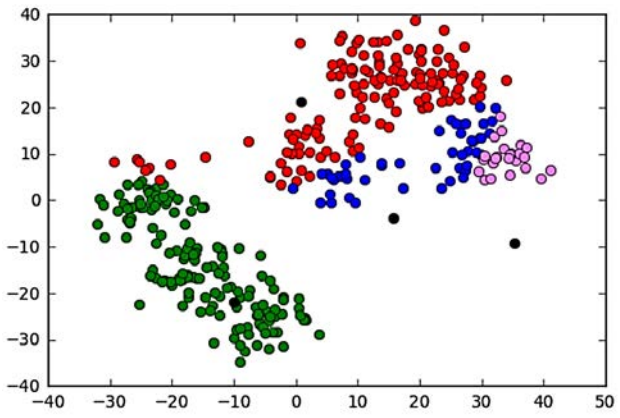


Рис. 8. Результати кластеризації на допоміжному наборі даних

Висновки. В статті розглянута задача кластеризації регіонів України з метою проведення ефективної регіонально орієнтованої економічної політики. Зроблена математична постановка задачі. Наведено опис декількох алгоритмів кластеризації для розбиття регіонів України на групи за соціально-економічними показниками та використано їх при розв'язанні задачі. У статті для експерименту було взято 10 показників для кожного із регіонів. Побудовано графіки залежності результатів від ітерації. Зроблено порівняння та аналіз результатів, одержаних різними алгоритмами, залежно від числа кластерів. Результати досліджень свідчать про те, що датасет, сформований на початку експерименту, є якісним.

Список літератури:

1. Веб-сайт Державної служби статистики України. 2017. URL: http://www.ukrstat.gov.ua/.
2. Прогнозирование. Модели, методы, алгоритмы / В.Е. Снитюк. Киев, 2008. 367 с. ISBN 978-966-2200-09-6
3. Multi-Objective Optimization using Differential Evolution: A Survey of the State-of-the-Art / [Efr´en Mezura-Montes, Margarita Reyes-Sierra, Carlos A. Coello Coello] 28 с.
4. Искусственные иммунные системы и их применение / Под ред. Д. Дасгупты. Пер. с англ. под ред. А.А. Романюхи. Москва: ФИЗМАТЛИТ, 2006. 344 с. ISBN 5-9221-0706-2.
5. Implementation of Immunological Algorithms in Solving Optimization Problems / [Petar Čisar, Cara Dušana, Sanja Maravić Čisar, Branko Markoski] – Acta Polytechnica Hungarica. Vol. 11, No. 4, 2014 P: 225-239. URL: https://www.uni-obuda.hu/journal/Cisar_Maravic-Cisar_Markoski_50.pdf.
6. Using Genetic Algorithms to Explore Pattern Recognition in the Immune System / [Stephanie Forrest, Brenda Javornik, Robert E. Smith, Alan S. Perelson]. 1993 P. 31. URL: https://pdfs.semanticscholar.org/ f392/c26 fd80a321d773eff40437fcec0edbf5982.pdf.
7. Immune Network: An Example of Complex Adaptive Systems, Debashish Chowdhury. 1998. P. 20. URL: https://arxiv.org/pdf/cond-mat/9803033.pdf.
8. Веб-сайт з документацією до використання sklearn.cluster. K-Means. URL: http://scikit-learn.org/ stable/modules/generated/sklearn.cluster. KMeans.html.
9. Clustering using K-means algorithm, Firdaouss Doukkali. URL: https://towardsdatascience.com/ clustering-using-k-means-algorithm-81da00f156f6.
10. Веб-сайт з документацією до використання scipy.cluster.hierarchy.linkage. URL : https://docs.scipy. org/doc/scipy/reference/generated/scipy.cluster.hierarchy.linkage.html#scipy.cluster.hierarchy.linkage.
11. Метод ближайшего соседа или метод одиночной связи. URL: http://www.aiportal.ru/articles/ autoclassification/single-link.html.

Snytyuk V.Ye., Soroka P.M., Tkachenko O.V. **THE DISTRIBUTION PROBLEM OF REGIONS OF UKRAINE INTO CLUSTERS FOR THE PURPOSE OF CONDUCTING REGIONALLY ORIENTED ECONOMIC POLICY**

The problem of clustering is one of the most popular tasks of machine learning today. It is a task to divide objects into groups so that the objects in the same group are as similar as possible and the objects from different clusters differ significantly. This task facilitates data processing and has many applications in real life.

The article deals with the problem of the partitioning of regions of Ukraine into clusters for the purpose of conducting regionally oriented economic policy. A mathematical statement of the problem is made in which each vector describes a specific region of Ukraine according to certain characteristics. The paper considers that all the features are equilibrium, so before starting the solution, the data is normalized using linear transformation. A function that is minimized in the clustering problem using a normalized dataset is presented.

The article deals with the following clustering algorithms: evolutionary strategy, differential evolution algorithm, genetic algorithm, immune algorithm, K-means, nearest neighbor method. They were used to solve the problem for the distribution of the regions of Ukraine into groups by social and economic indicators. For the experiment, 10 indicators for each region were taken. The calculations used a normalized dataset, based on data from the website of the State Statistics Service of Ukraine for 2017. The studies use two-dimensional projections of this multidimensional data set and ancillary dataset generated.

Graphs of the results of the minimum value of the function f_min on the number of iterations for the differential evolution algorithm and the genetic algorithm are constructed. It is shown that at iterations greater than 250 for the differential evolution algorithm and greater than 80 for the genetic algorithm, the minimum value of the function f_min stabilizes. Comparison and analysis of the results of studies by the number of clusters obtained by different algorithms are made. It is shown that for the number of clusters equal to 5 evolutionary algorithms give quite decent results.

Key words: *region, cluster, clustering, economic policy, algorithm, iteration.*