

TEHNICĂ DE CLUSTERING DESTINATĂ PROGRAMELOR DE PROIECTARE OPTIMALĂ CU ALGORITMI GENETICI

Daniela JUCAN, Lucian TUDOSE
Universitatea Tehnică Cluj-Napoca

Cuvinte cheie: clustering, $t_{\Omega m}$ -threshold.

Abstract. In this paper, an improved clustering technique addressed to optimal design is presented. This technique is based on the assumption that any couple formed by a set of points and its envelope, in the n -dimensional space, is characterized by a unique real number, named $t_{\Omega m}$ -threshold that is able to describe the state of uniform cover.

1. NECESITATEA TEHNICILOR DE CLUSTERING IN PROIECTAREA OPTIMALĂ

În proiectarea optimală, Algoritmii Genetici reprezintă unele dintre cele mai utilizate metode de optimizare. O problemă importantă este aceea a gestionării funcțiilor obiectiv multimodale. Autorii lucrării consideră că se poate dezvolta o nouă categorie de algoritmi de optimizare care să includă module de clustering. Aceste module vor fi necesare în următoarea modificare a Algoritmului Genetic:

La un moment dat evoluția populației se oprește și se identifică aglomerările, grupurile (clusterelor) din spațiul variabilelor (genelor) cu speranța ca aceste cluster, lăsate ulterior să evolueze ca populații independente, să conducă, fiecare, la identificarea unui alt punct de extrem local al funcției obiectiv (multimodală).

Este evident că un astfel de algoritm de clustering nu va fi necesar să fie extrem de precis, ci va trebui să aibă alte calități, și anume:

- să identifice rapid numărul de cluster existente în populație;
- să fie cât mai simplu;
- să fie robust.

Autorii lucrării propun aici o nouă tehnică de clustering, ce a dat rezultate foarte bune în implementările ei în programele proprii de optimizare cu Algoritmi Genetici.

2. ANALIZA CLUSTERELOR

2.1 Generalități

Se consideră o mulțime de puncte $\Omega_m \subset \mathbf{R}^n$, adică o listă nesortată a coordonatelor celor m puncte (în spațiul real n -dimensional) supuse procesului de clustering:

$$\Omega_m = \{M_1(x_1^1, x_2^1, \dots, x_n^1), M_2(x_1^2, x_2^2, \dots, x_n^2), \dots, M_m(x_1^m, x_2^m, \dots, x_n^m)\} \quad (1)$$

Trebuie menționat aici că numărul de cluster nu se consideră cunoscut a priori (spre deosebire de multe dintre tehnicile fără ierarhizare cunoscute [1], [2], [4] – ca de exemplu K-means Algorithm). Pentru buna înțelegere a tehnicii propuse, se vor introduce câteva axiome și definiții.

Definiția 1. Învelitoarea plană a mulțimii Ω_m este hiper-paralelipipedul definit ca mulțimea:

$$P_{\Omega m} = \{(x_1, x_2, \dots, x_n) | x_1^{\min} \leq x_1 \leq x_1^{\max}, x_2^{\min} \leq x_2 \leq x_2^{\max}, \dots, x_n^{\min} \leq x_n \leq x_n^{\max}\} \quad (2)$$

unde:

$$\begin{aligned} x_k^{\min} &= \min \{x_k^1, x_k^2, \dots, x_k^m\} \quad (\forall) k \in \{1, 2, \dots, n\} \\ x_k^{\max} &= \max \{x_k^1, x_k^2, \dots, x_k^m\} \quad (\forall) k \in \{1, 2, \dots, n\} \end{aligned} \quad (3)$$

În figura 1 este prezentată învelitoarea plană (paralelipipedul dreptunghic) a unei mulțimi 3D, $\Omega_m \subset \mathbb{R}^3$.

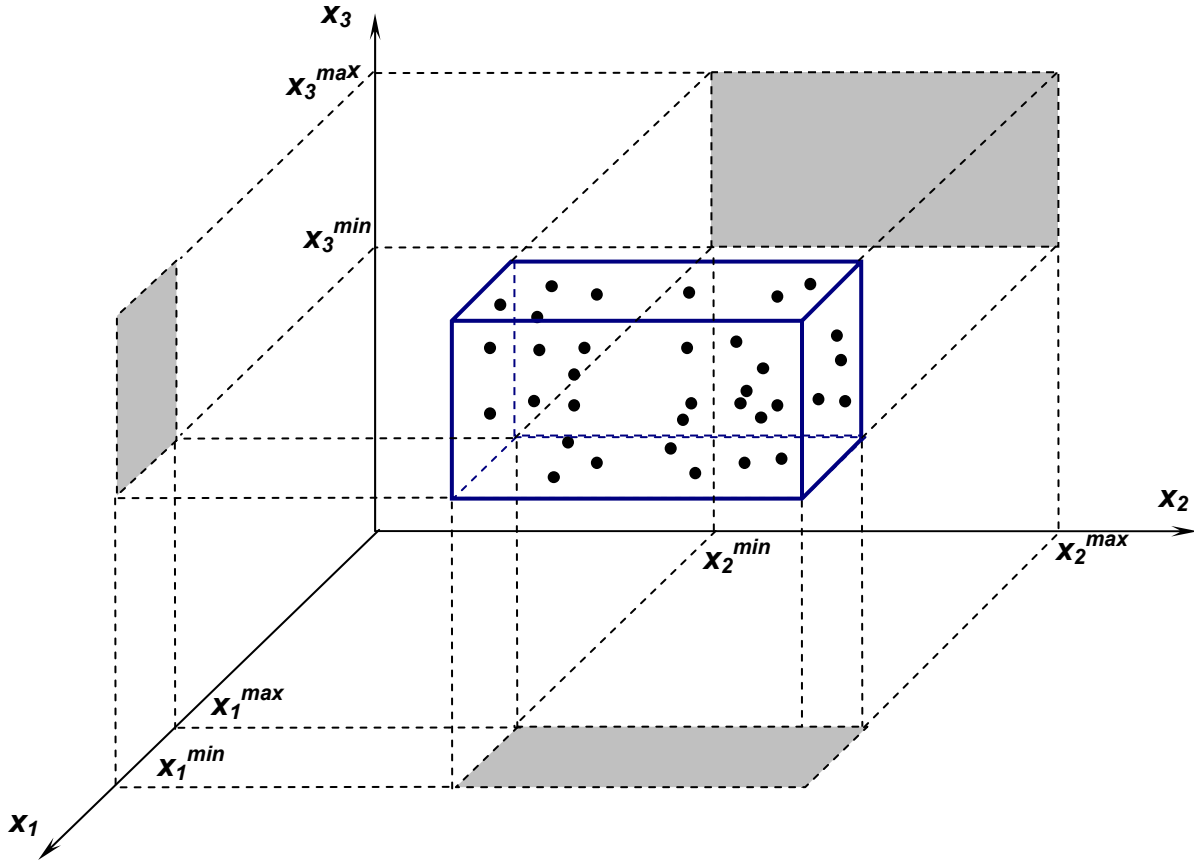


Fig. 1 Învelitoarea plană a unei mulțimi 3D

Axiomă: Pentru fiecare cuplu (Ω_m, P_{Ω_m}) există o unică valoare de prag $t_{\Omega_m} > 0$ care descrie situația în care punctele mulțimii Ω_m ocupă uniform hiper-spațiul delimitat de învelitoarea sa plană P_{Ω_m} .

Definiția 2. Fie punctul $M_i \in \Omega_m$. Punctul $M_j \in \Omega_m, j \neq i$ este numit t_{Ω_m} -vecin al punctului M_i dacă:

$$d(M_i, M_j) = \sqrt{(x_1^i - x_1^j)^2 + (x_2^i - x_2^j)^2 + \dots + (x_n^i - x_n^j)^2} \leq t_{\Omega_m}, i \neq j \quad (4)$$

Definiția 3. O submulțime $C_{\Omega_m} \subseteq \Omega_m$ este un t_{Ω_m} -cluster dacă pentru oricare punct $M_i \in C_{\Omega_m}$, există cel puțin un alt punct $M_j \in C_{\Omega_m}, j \neq i$ astfel încât M_i și M_j sunt t_{Ω_m} -vecine:

$$d(M_i, M_j) = \sqrt{(x_1^i - x_1^j)^2 + (x_2^i - x_2^j)^2 + \dots + (x_n^i - x_n^j)^2} \leq t_{\Omega_m}, M_{i,j} \in C_{\Omega_m}, i \neq j \quad (5)$$

Definiția 4. Numărul de cluster ale mulțimii Ω_m este numărul de t_{Ω_m} -cluster în care este divizată mulțimea.

2.2 Determinarea valorii de prag t_{Ω_m}

Hiper-volumul învelitorii plane P_{Ω_m} (a mulțimii Ω_m) este dat de relația:

$$V_{P_{\Omega_m}} = \prod_{k=1}^n (x_k^{\max} - x_k^{\min}) \quad (6)$$

iar lungimea muchiei unui hiper-cub corespunzător unui punct oarecare din mulțimea Ω_m este:

$$I_{P\Omega m} = \sqrt[n]{\frac{V_{P\Omega m}}{m}} = \sqrt[n]{\frac{\prod_{k=1}^n (x_k^{\max} - x_k^{\min})}{m}} \quad (7)$$

Acest lucru înseamnă că fiecare interval $[x_k^{\min}, x_k^{\max}]$, $k = \overline{1, n}$ trebuie divizat într-un număr de subintervale, fiecare cu lungimea $I_{P\Omega m}$. Numărul acestor subintervale, pentru fiecare dimensiune în parte, este dat de relația:

$$n'_k = \frac{x_k^{\max} - x_k^{\min}}{I_{P\Omega m}}, k = \overline{1, n} \quad (8)$$

Evident, pentru orice dimensiune k , numărul n'_k nu este întreg. Din acest motiv se vor lua în considerare două numere întregi:

$$\begin{cases} n_k = \left\lceil \frac{x_k^{\max} - x_k^{\min}}{I_{P\Omega m}} \right\rceil, k = \overline{1, n} \\ N_k = n_k + 1, k = \overline{1, n} \end{cases} \quad (9)$$

care reprezintă, pentru orice dimensiune k , numărul minim și, respectiv, maxim de diviziuni ale intervalului $[x_k^{\min}, x_k^{\max}]$, $k = \overline{1, n}$. În relația (9) prin $\lceil \cdot \rceil$ s-a notat partea întreagă a numărului respectiv.

Se poate, acum, scrie relația evidentă:

$$\prod_{k=1}^n n_k \leq m \leq \prod_{k=1}^n N_k \quad (10)$$

Lungimea muchiei hiper-cubului corespunzător dimensiunii k și numărului de diviziuni n_k este:

$$I_k = \frac{x_k^{\max} - x_k^{\min}}{n_k}, k = \overline{1, n} \quad (11)$$

Analog, lungimea muchiei hiper-cubului corespunzător dimensiunii k și numărului de diviziuni N_k este:

$$L_k = \frac{x_k^{\max} - x_k^{\min}}{N_k}, k = \overline{1, n} \quad (12)$$

Lungimea diagonalei hiper-cubului corespunzător numărului de diviziuni n_k este furnizată de relația:

$$t_{\max} = \sqrt{\sum_{k=1}^n I_k^2} \quad (13)$$

Analog, lungimea diagonalei hiper-cubului corespunzător numărului de diviziuni N_k este:

$$t_{\min} = \sqrt{\sum_{k=1}^n L_k^2} \quad (14)$$

În final, valoarea de prag $t_{\Omega m}$ se va determina cu relația:

$$t_{\Omega m} = t_{\min} + \frac{\left(\prod_{k=1}^n N_k - m \right) \cdot (t_{\max} - t_{\min})}{\prod_{k=1}^n N_k - \prod_{k=1}^n n_k} \quad (15)$$

Modalitatea prin care s-a determinat valoarea de prag $t_{\Omega m}$ se poate urmări și în schema din figura 2.

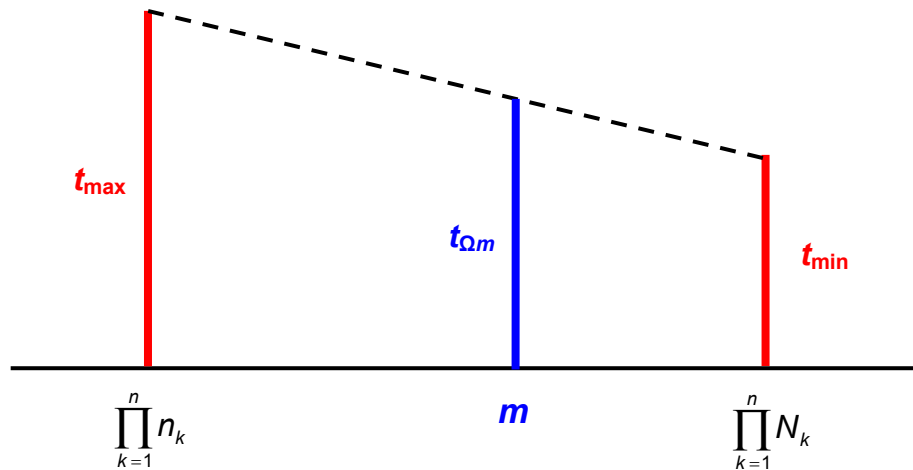


Fig. 2 Schema de calcul a valorii de prag

2.3 Algoritmul de identificare a clusterelor unei mulțimi

În cele ce urmează, se vor prezenta pe larg, pașii algoritmului propus pentru identificarea clusterelor unei mulțimi date de puncte. Acest algoritm este modificat mult față de cel propus anterior de autori [3].

- Pasul 1.** Se determină, conform paragrafului precedent, valoarea de prag $t_{\Omega m}$ a mulțimii de puncte date Ω_m (lista target, lista coordonatelor punctelor care trebuie supuse clustering-ului).
- Pasul 2.** Se inițializează o listă (lista curentă) prin primul punct al listei target (înregistrare care se șterge din lista target).
- Pasul 3.** Pentru primul punct al listei curente, se identifică toți $t_{\Omega m}$ -vecinii din lista target, se adaugă la sfârșitul listei curente și se șterg din lista target.
- Pasul 4.** Pentru următorul punct din lista curentă se identifică toți $t_{\Omega m}$ -vecinii din lista target, se adaugă la coada listei curente și se șterg din lista target.
- Pasul 5.** Se reia Pasul 4 (până când nici unul dintre punctele listei curente nu mai are nici un $t_{\Omega m}$ -vecin în lista target). Lista curentă astfel obținută este unul dintre clusterelor mulțimii date.
- Pasul 6.** Se reia secvența Pasul 2 – Pasul 5, până la epuizarea listei target.

3. MODULUL DE CLUSTERING

În cele ce urmează, se prezintă modul în care funcționează algoritmul și se realizează identificarea clusterelor dintr-o mulțime dată. Pentru ușurința expunerii, modulului care are implementat algoritmul prezentat în paragraful anterior i s-a atașat o interfață grafică. Această interfață este prezentată în figura 3.

S-au generat aleator 1137 puncte, care, evident, sunt grupate în trei cluster (fig. 4). De menționat că programul primește numai lista coordonatelor punctelor, ordinea punctelor din listă neavând nimic de a face cu clusterul din care fac parte.

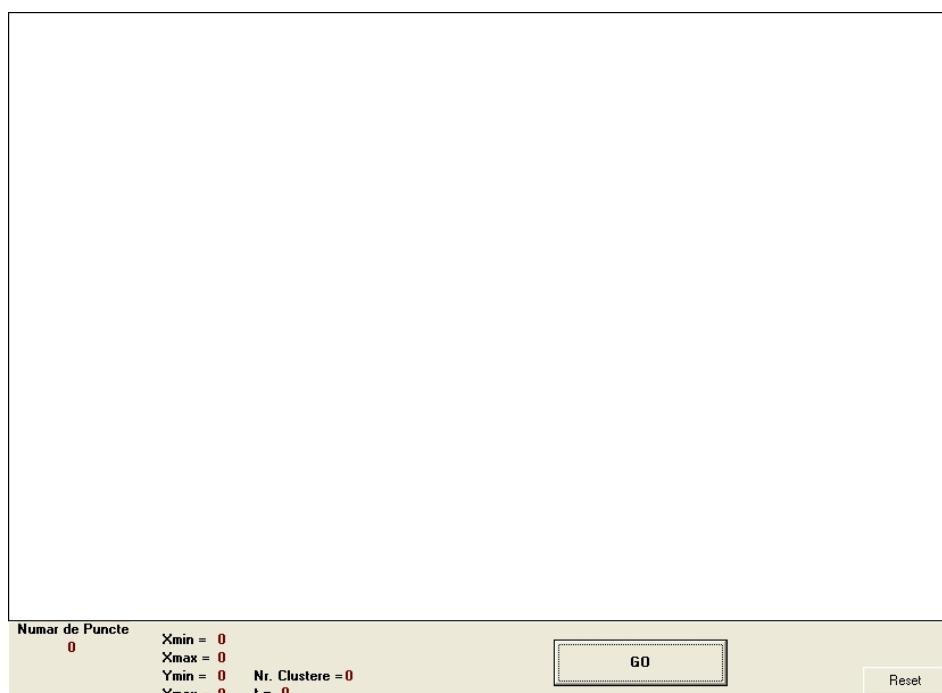


Fig. 3 Interfața grafică a modului de clustering

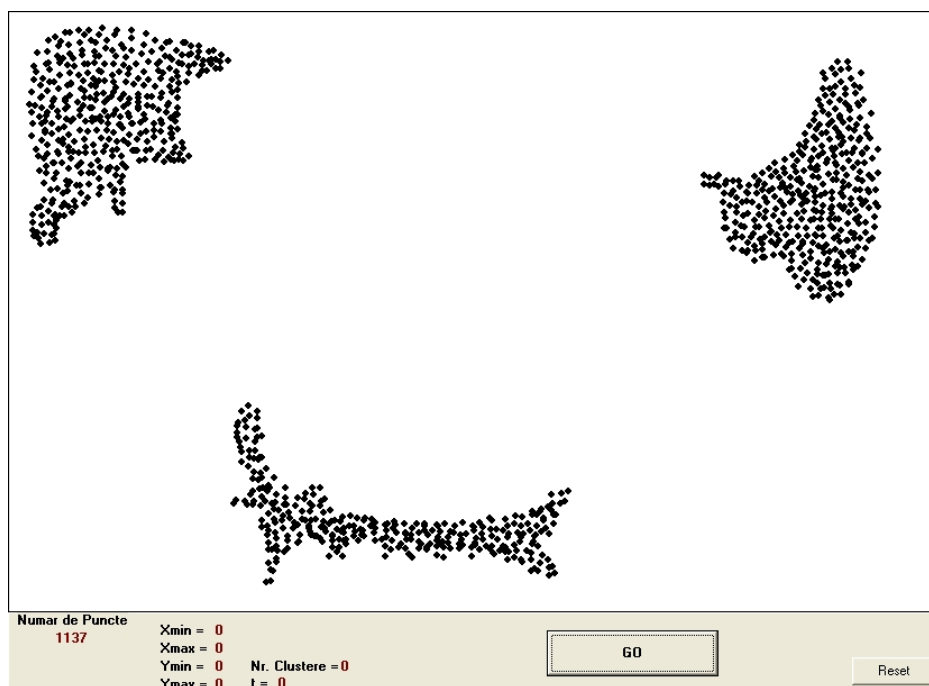


Fig. 4 Mulțimea inițială

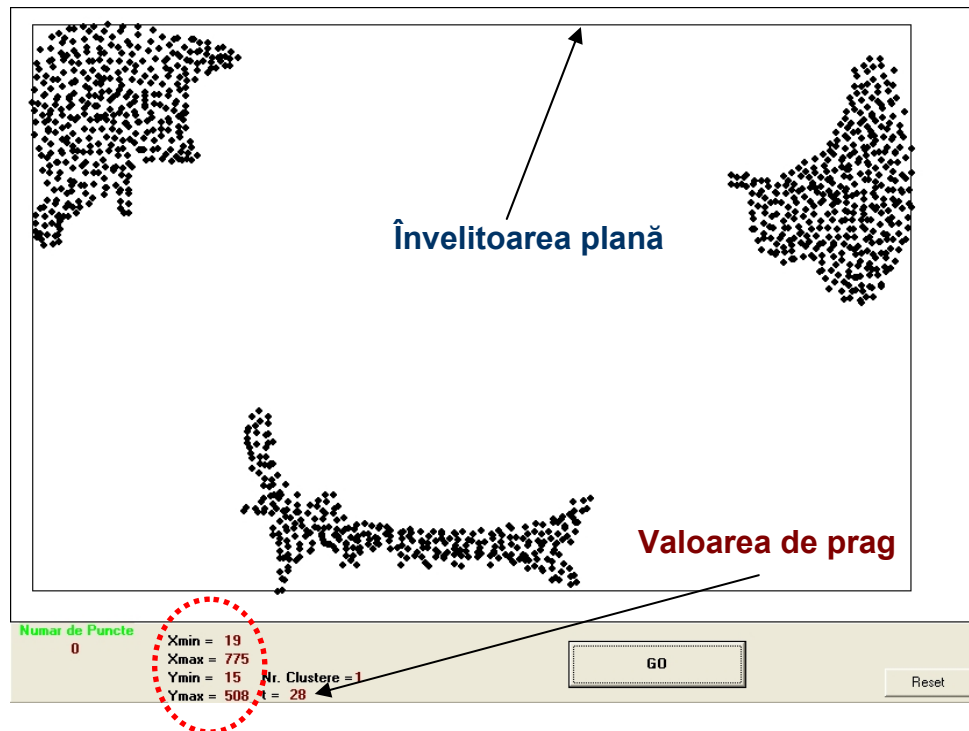


Fig. 5 Stabilirea învelitorii plane și a valorii de prag

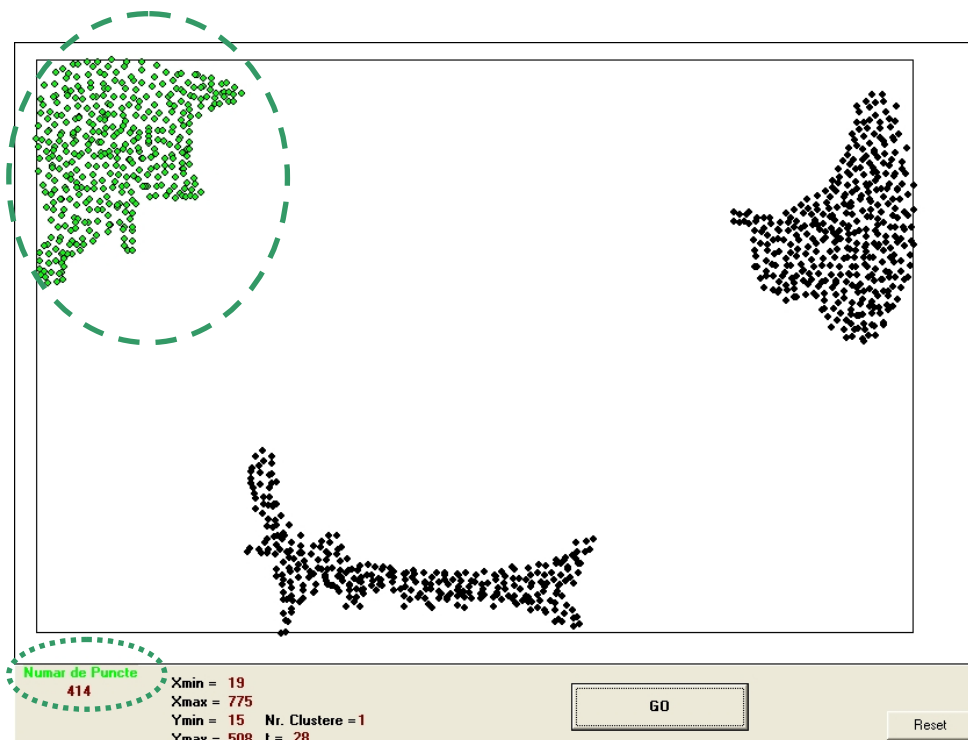
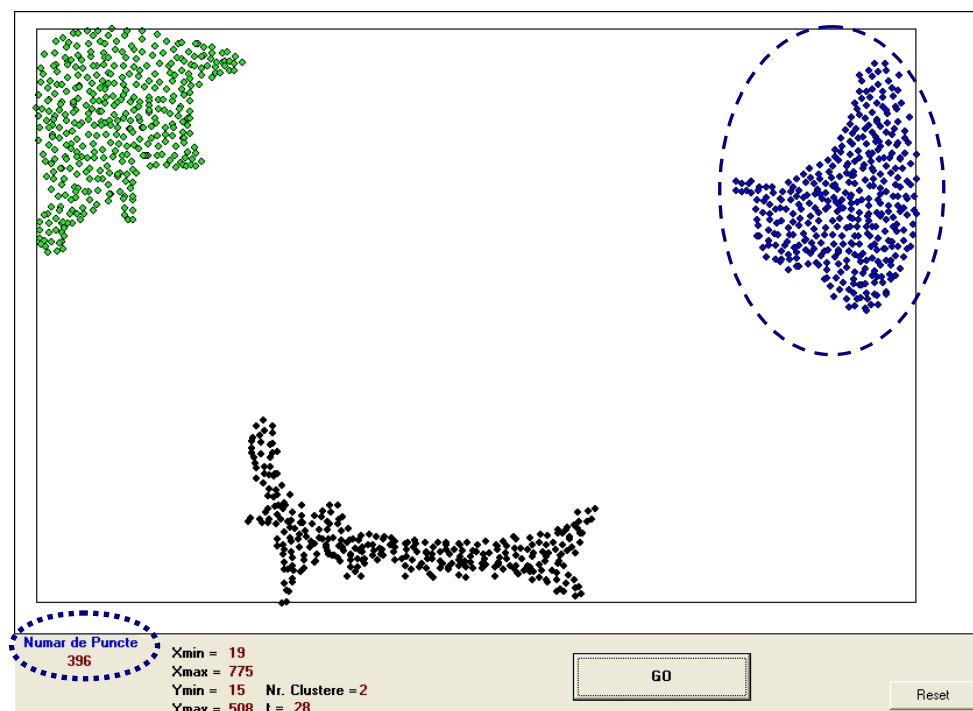
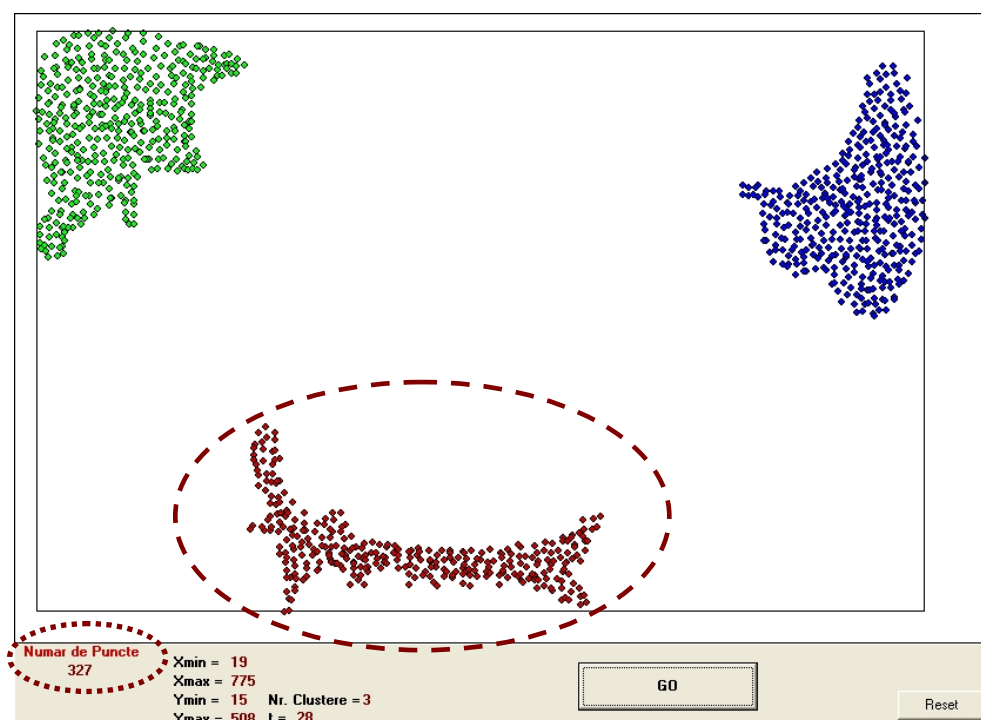


Fig. 6 Identificarea primului cluster

*Fig. 7 Identificarea celui de-al doilea cluster**Fig. 8 Identificarea celui de-al treilea cluster*

Pentru început se va identifica învelitoarea 2D a mulțimii date și se va calcula valoarea de prag (fig. 5).

Din comoditate, pentru valoarea pragului $t_{\Omega m}$ s-au afișat numai numere întregi. În acest fel, valoarea găsită a pragului este:

$$t_{\Omega m} \approx 28 \quad (16)$$

În figura 6 se prezintă momentul identificării primului cluster. După cum se vede acesta conține un număr de 414 puncte.

În figura 7 se prezintă momentul identificării celui de-al doilea cluster, care conține 396 puncte distincte.

În final, în figura 8 este prezentat momentul identificării ultimului cluster. Acest al treilea și ultim cluster conține cele 327 puncte rămase.

4. APLICAȚII ȘI CONCLUZII

Această nouă tehnică de clustering, mult îmbunătățită față de varianta precedentă prezentată în [3], a fost implementată cu succes mai întâi într-un program dedicat, numit *GetCluster v. 1.2* și apoi ca modul specializat al unor programe de optimizare cu Algoritmi Genetici (programul *ClusterGA*).

Rezultatele obținute în gestionarea funcțiilor cu mai multe extreme locale au fost excelente. Metoda a permis identificarea cu precizie a tuturor extremelor locale unor funcții multimodale, fără a avea informații despre localizarea (nici măcar aproximativă) a acestora.

BIBLIOGRAFIE

- [1] Berkhin, P. – Survey of Clustering Data Mining Techniques, Accrue Software, Inc., 1045 Forest Knoll Dr., San Jose, CA, 95129, 2002.
- [2] Jain, A. K., Murty, M.N., Flynn, P.J. – Data Clustering: A Review, ACM Computing Surveys, Vol. 31, No. 3, September 1999.
- [3] Jucan D., Tudose L. – A New Robust Clustering Technique Dedicated to Optimal Design Software Annals of the Oradea University, Annals of the Oradea University, Fascicle of Management and Technological Engineering, Vol. IV(XXV), Editura Universității din Oradea, p.25+CD, ISSN 1583-0691, 2005.
- [4] Mendonca, M., Sunderhaft, N. L. – Mining Software Engineering Data: A Survey. A DACS State-of-the-Art Report, Contract Number SPO700-98-D-4000 (Data & Analysis Center for Software), 1999.
- [5] Tudose, L., Pop, D., Haragăș, S., Nistor, G., Jucan, D., Pustan, M. – Proiectarea optimală a sistemelor complexe, Editura Mediamira, ISBN 973-713-076-6, Cluj-Napoca, 2006.