Journal of Mathematical Modeling
Vol. 13, No. 4, 2025, pp. 851-864. Research Article

JMM

Accelerated DBSCAN via parallel, density-aware multi-objective genetic optimization

Hossein Eyvazi[†*], **Ali Rajaei**[‡]

Department of Computer Science, Tarbiat Modares University, Tehran, Iran
Email(s): eyvazi_hoseyn@modares.ac.ir, alirajaei@modares.ac.ir

Abstract. Clustering is a fundamental task in data mining, where the quality of results often hinges on effective parameter selection. DBSCAN is widely used for discovering clusters of arbitrary shapes but is highly sensitive to its input parameters *Eps* and *MinPts*. This paper proposes an enhanced version of the Multi-Objective Genetic Algorithm for DBSCAN, termed **Enhanced MOGA-DBSCAN**, which introduces a modified objective function based on a density-aware Outlier Index and accelerates the optimization process through parallel computation. We evaluate the proposed method using two benchmark datasets and compare it against the original MOGA-DBSCAN as well as two adaptive variants: AMD-DBSCAN and WOA-DBSCAN. Results show that Enhanced MOGA-DBSCAN consistently achieves superior clustering performance, as measured by Rand Index and Normalized Mutual Information (NMI), while also reducing runtime relative to the original MOGA-DBSCAN. These findings highlight the effectiveness of our enhancements in improving both clustering quality and computational efficiency.

Keywords: Machine learning, unsupervised learning, clustering, DBSCAN, genetic algorithm.
AMS Subject Classification 2010: 62H30, 68T05.

1 Introduction

Clustering is a foundational task in data mining and machine learning, aimed at partitioning data into meaningful groups, or clusters, based on similarities among objects [8]. Among clustering methods, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is particularly prominent due to its ability to identify clusters of arbitrary shapes and effectively detect outliers [10]. However, DBSCAN's effectiveness critically depends on two parameters: *Eps*, defining the neighborhood radius, and *MinPts*, specifying the minimum number of points required to form a dense region. Selecting optimal values for these parameters is inherently challenging, especially when prior knowledge about dataset structures is limited [2, 9].

*Corresponding author
Received: 29 January 2025 / Revised: 30 April 2025 / Accepted: 6 June 2025
DOI: 10.22124/jmm.2025.29702.2648

To tackle this issue, Multi-Objective Genetic Algorithms (MOGA) have been employed successfully for parameter optimization in DBSCAN, transforming parameter tuning into a multi-objective optimization problem [5]. Specifically, MOGA-DBSCAN utilizes genetic algorithms to systematically explore parameter settings, simultaneously optimizing multiple criteria such as the Silhouette index-reflecting cluster compactness and separation-and the Outlier Index, which assesses the distinctiveness of detected outliers relative to clusters.

Nevertheless, the original formulation of the Outlier Index considers clusters uniformly, disregarding their density variations, leading to potential inaccuracies. For example, an outlier positioned equidistantly from two clusters-one dense, the other sparse-is equally penalized despite being intuitively more significant if distant from the denser cluster. This limitation can degrade clustering performance, particularly in datasets with varying densities [1].

In this paper, we propose an enhanced variant of the MOGA-DBSCAN algorithm, named Enhanced MOGA-DBSCAN, that incorporates a density-aware modification of the Outlier Index. Our approach scales the distances of outliers relative to cluster densities, providing a more precise evaluation. Additionally, we parallelize the computation of the Outlier Index, significantly reducing runtime and enhancing computational efficiency. This improvement makes Enhanced MOGA-DBSCAN more practical and scalable, particularly beneficial for datasets with varying cluster densities.

We rigorously evaluate our method against the original MOGA-DBSCAN as well as adaptive variants AMD-DBSCAN [12] and WOA-DBSCAN [14] using two benchmark datasets. Experimental results clearly demonstrate that Enhanced MOGA-DBSCAN achieves superior clustering quality, measured by Rand Index and Normalized Mutual Information (NMI), while maintaining reasonable computational demands.

The remainder of the paper is organized as follows: Section 2 details our methodology, Section 3 presents the experimental setup and results, and Section 4 discusses the broader implications of our findings within the context of density.

2 Methodology

The proposed Enhanced MOGA-DBSCAN algorithm aims to optimize the DBSCAN clustering process by framing it as a multi-objective optimization problem. The algorithm utilizes a genetic algorithm to explore various combinations of the DBSCAN parameters *Eps* and *MinPts*, with the objective of maximizing two metrics: the Silhouette index and the Enhanced Outlier Index.

2.1 Enhanced outlier index

The original Outlier Index computes the average minimum distance of outliers to the nearest clusters, but it does not account for cluster density [13]. This limitation can lead to inaccurate results, especially in datasets with clusters of varying densities. To overcome this, we propose an Enhanced Outlier Index that integrates cluster density into its computation, while addressing two key challenges: (1) improving computational efficiency through parallelization, and (2) normalizing densities to handle very sparse or very dense clusters effectively [4].

The Enhanced Outlier Index is defined as follows:

$$\frac{1}{n}\sum_{i=1}^{n}\left(\min_{1\le j\le m}(d_{ij})\times \text{Normalized Density}_j\right),$$

where

- n is the number of outliers.
- d_{ij} is the distance of outlier i from cluster j.
- m is the number of clusters.
- Density_j is the density of cluster j, computed as the inverse of the average distance from the centroid of cluster j.
- $\text{Normalized Density}_j$ is the density of cluster j, normalized relative to the densities of all clusters, and is computed as:
$$\frac{\text{Density}_j-\min(\text{Density}_k)}{\max(\text{Density}_k)-\min(\text{Density}_k)}+\varepsilon,$$
where $k\in\{1,2,\ldots,m\}$ represents all clusters.

2.1.1 Rationale for normalization

Normalization ensures that all cluster densities lie within the range $[0,1]$, limiting the influence of extreme density values. This adjustment addresses cases where clusters are either extremely sparse or extremely dense, ensuring that the Enhanced Outlier Index remains robust and meaningful across a wide range of datasets.

2.1.2 Parallelization of density computation

The time complexity of density computation is $O(n)$, where n is the number of data points. For large datasets, this computation becomes expensive. To address this issue, we implemented a parallelization strategy for the key steps involved in the Enhanced Outlier Index computation. The process is divided into the following components:

The complete algorithm for parallelizing the Enhanced Outlier Index computation consisted of the following steps:

1. **Preprocessing:**
   - We computed the centroids of all clusters in parallel.
   - Data points and outliers were assigned to workers for further processing.
2. **Parallel density calculation:**
   - Clusters were divided among workers.
   - Each worker computed Density_j for its assigned clusters.

- We gathered all Density_j values to the main process.

3. **Normalize densities:**
   - We computed $\min(\text{Density}_k)$ and $\max(\text{Density}_k)$ using a parallel reduction.
   - These values were broadcast to all workers.
   - Each worker normalized Density_j in parallel.
4. **Outlier distance computation:**
   - Outliers were divided among workers.
   - Each worker computed distances d_{ij} of its assigned outliers to all cluster centroids.
   - Each worker found $\min(d_{ij})$ for each outlier and multiplied it by Normalized Density_j.
   - Partial sums were sent to the main process.
5. **Parallel aggregation:**
   - We used a parallel reduction to sum the contributions of all outliers across workers.
   - The total was divided by n to compute the final EOI.

2.2 Motivation for the enhanced outlier index

The Enhanced Outlier Index is designed to address the shortcomings of the original Outlier Index, particularly in scenarios with clusters of varying densities:

- **Scenario 1:** An outlier o is located at a distance x from a dense cluster c_1. Figure 1 illustrates this case.
- **Scenario 2:** The same outlier o is located at the same distance x from a sparse cluster c_2. Figure 2 illustrates this case.

In the original Outlier Index, both scenarios would penalize the outlier o equally, as the metric considers only the distance x. However, this approach fails to account for the fact that outliers close to dense clusters should be penalized more heavily than those close to sparse clusters.

The Enhanced Outlier Index resolves this issue by incorporating the normalized density of clusters into the calculation. In Scenario 1, where c_1 is dense, the penalty for the outlier is higher. Conversely, in Scenario 2, where c_2 is sparse, the penalty is lighter. This adjustment provides a more accurate representation of the significance of outliers relative to the clustering structure, resulting in improved clustering quality [6].

2.3 Enhanced MOGA-DBSCAN process

The Enhanced MOGA-DBSCAN process begins with the initialization of a population of candidate solutions, where each solution represents a pair of DBSCAN parameters *Eps* and *MinPts*. The initial population is generated within bounds determined by Delaunay triangulation, ensuring a diverse and high-quality set of candidate solutions [7].

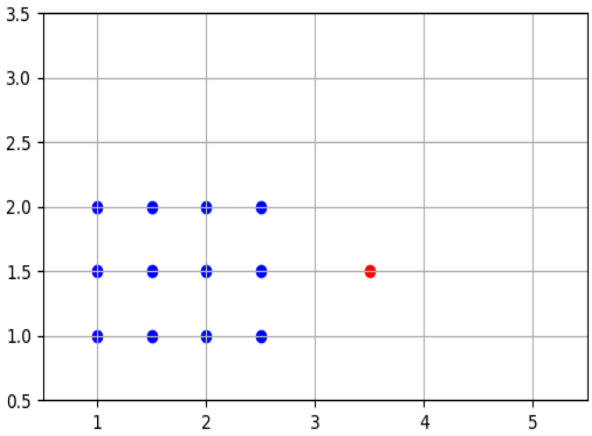

Figure 1: Scenario 1: Outlier o at distance x from a dense cluster c_1.

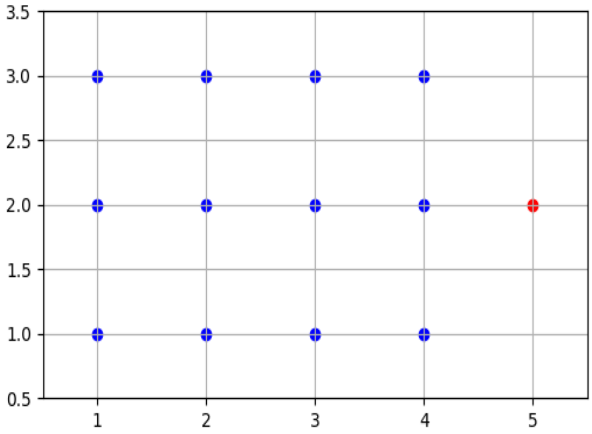

Figure 2: Scenario 2: Outlier o at distance x from a sparse cluster c_2.

The algorithm iteratively applies mutation and crossover operators to generate new solutions, which are then evaluated using the two objective functions: the Silhouette index and the Enhanced Outlier Index [3]. A non-dominated sorting approach, combined with a crowding distance mechanism [11], is used to select the next generation of solutions, guiding the population towards Pareto-optimal solutions [15].

To accelerate convergence, a statistical t-test is employed to compare the performance of the current Pareto front with that of previous generations. If the null hypothesis is accepted, indicating no significant improvement, the algorithm terminates early. Otherwise, the process continues until the maximum number of generations is reached or the stopping criterion is satisfied.

At the end of the optimization process, the algorithm outputs a set of Pareto-optimal solutions, allowing users to select the optimal value themselves.[1]

[1] The Implementation is available at https://github.com/HosseinEyvazi/Density-Adjusted-MOGA-DBSCAN.

2.4 Algorithm overview

Algorithm 1: Enhanced MOGA-DBSCAN

Initialize population of N individuals with random $(Eps, MinPts)$
while not termination condition **do**
  for each individual in population **do**
    Apply DBSCAN with $(Eps, MinPts)$
    Calculate Silhouette Index
    Calculate Enhanced Outlier Index
    Assign fitness based on objectives
  end for
  Perform non-dominated sorting
  Calculate crowding distance
  Select next generation based on Pareto front and crowding distance
  Apply crossover and mutation to create offspring
  Evaluate offspring fitness
  Merge population with offspring
  Select top N individuals for next generation
  if statistical t-test accepts null hypothesis **then**
    break
  end if
end while
Output Pareto-optimal solutions

2.5 Time complexity of MOGA-DBSCAN

The time complexity of MOGA-DBSCAN depends on the time complexities of three primary components: DBSCAN, NSGA-II (Non-dominated Sorting Genetic Algorithm II), and the Delaunay triangulation. The overall time complexity is expressed as follows

$$O\left(g \times \left(O(\text{fitness}) + O(n \log n)\right)\right),$$

where

- g is the number of generations,
- $O(\text{fitness})$ indicates the time complexity of the cluster validity indices used as objective functions,
- n is the total number of data points.

The time complexity of DBSCAN and the Delaunay triangulation is $O(n \log n)$. NSGA-II has a time complexity of $O(MN^2)$, where M is the number of objectives and N is the population size. Given that M and N are much smaller than the total number of data points (n), the overall complexity of NSGA-II can be considered $O(1)$ relative to the total dataset size.

In the context of MOGA-DBSCAN, $O(\text{fitness})$ consists of the time complexities of the silhouette index and the outlier index. The silhouette index has a time complexity of $O(n^2)$, while the original outlier index has a time complexity of $O(n \times m)$, where m is the number of clusters. Therefore, the overall time complexity of MOGA-DBSCAN is dominated by the silhouette index, yielding

$$O\left(g \times \left(n^2 + n\log n\right)\right).$$

2.6 Time complexity of enhanced MOGA-DBSCAN

In Enhanced MOGA-DBSCAN, the outlier index used in MOGA-DBSCAN is replaced by the Enhanced Outlier Index. The Enhanced Outlier Index is computed as follows

$$\frac{\text{Density}_j - \min(\text{Density}_k)}{\max(\text{Density}_k) - \min(\text{Density}_k)} + \varepsilon,$$

The time complexity of computing the Enhanced Outlier Index is $O(n \times m)$, the same as the original outlier index beacuse the time complexity of computing the density of a cluster is equal to size of that's cluster , also m is O(1). Thus, the time complexity for the fitness function in Enhanced MOGA-DBSCAN is

$$O(\text{fitness}) = O(n^2).$$

This is still dominated by the silhouette index, so the overall time complexity of Enhanced MOGA-DBSCAN remains

$$O\left(g \times \left(n^2 + n\log n\right)\right)$$

Given that n^2 dominates $n\log n$, the time complexity can be simplified to

$$O\left(g \times n^2\right).$$

This shows that the introduction of the Enhanced Outlier Index does not increase the overall complexity of the algorithm compared to the original MOGA-DBSCAN.

2.6.1 Theoretical and empirical scalability of the parallel EOI

Let p denote the number of worker threads. Computing cluster densities and outlier–to–centroid distances are both embarrassingly parallel, so the per-generation cost of the Enhanced Outlier Index (EOI) decreases from $O(n)$ to $O\left(\frac{n}{p}\right)$, ignoring a modest $O(p)$ reduction overhead. Combining this with the existing $O(n^2)$ silhouette term gives

$$O\Big(g \times \big(n^2 + \tfrac{n}{p}\big)\Big) \;=\; O\big(g\,n^2\big) \quad \text{for fixed } p \ll n.$$

Thus the parallel EOI does not alter the overall complexity order, but-as the results in Subsection 3.3 confirm-reduces wall-clock time by a factor that approaches p for the largest inputs.

3 Benchmark results

We benchmark **Enhanced MOGA-DBSCAN** against three established density–based algorithms:

- **AMD-DBSCAN** [12] – adaptive multi-density DBSCAN;
- **WOA-DBSCAN** [14] – DBSCAN tuned by the Whale Optimisation Algorithm; and
- **MOGA-DBSCAN** [5] – the original genetic multi-objective variant.

All experiments were run on identical 10-core Intel Xeon W-2255 hardware. For each method we report:

(i) **Rand Index** (RI),

(ii) **Normalised Mutual Information** (NMI), and

(iii) wall–clock **runtime** in seconds.

Higher RI and NMI are better; lower runtime is preferable.

To condense quality and speed into a single indicator we follow the practice of weighting accuracy more heavily than execution time in offline clustering tasks. Per dataset we min–max normalise both objectives and define a composite score

$$S \;=\; 0.8\,\mathrm{NMI}_{\mathrm{norm}} \;+0.2\,\big(1-\mathrm{Runtime}_{\mathrm{norm}}\big)$$

so that $S \in [0,1]$ and *higher is better*.

3.1 Network dataset

Table 1: Raw performance on the *Network* dataset.

Method	Rand Index	NMI	Runtime (s)
AMD-DBSCAN	0.58	0.69	0.86
WOA-DBSCAN	0.97	0.89	269.81
MOGA-DBSCAN	0.67	0.73	373.09
Enhanced MOGA-DBSCAN	**0.99**	**0.98**	254.74

Discussion. Enhanced MOGA-DBSCAN attains the highest RI and NMI. Its runtime is $1.46\times$ shorter than that of the original MOGA-DBSCAN while adding $+0.32$ RI and $+0.25$ NMI. AMD-DBSCAN is faster but incurs a quality penalty of -0.41 RI and -0.29 NMI. When objectives are normalised and combined ($w_{\mathrm{NMI}} = 0.8$) Enhanced achieves the top composite score $S = 0.86$, outperforming AMD by 4.3×.

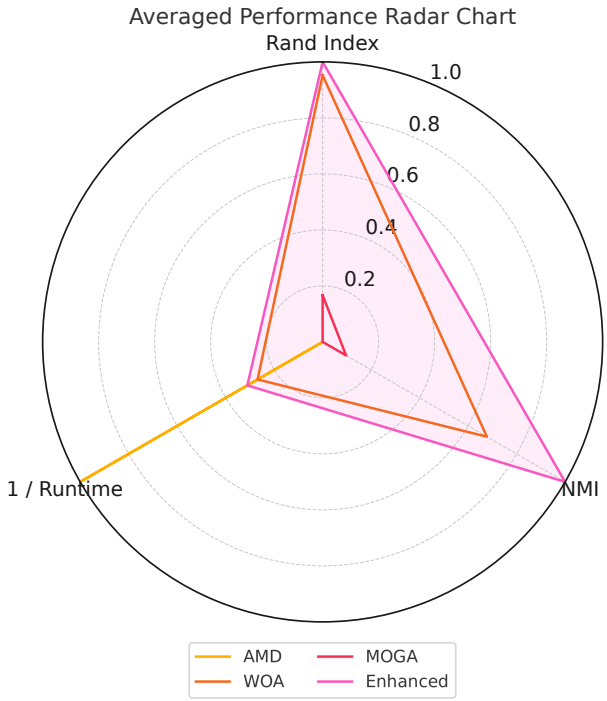

Figure 3: Normalized radar plot averaged across benchmarks. Enhanced MOGA-DBSCAN shows the most balanced and optimal performance across RI, NMI, and inverse runtime.

Table 2: Raw performance on the *Custom-3-Cluster* dataset.

Method	**Rand Index**	**NMI**	**Runtime (s)**
AMD-DBSCAN	0.98	0.97	0.19
WOA-DBSCAN	0.99	0.98	17.96
MOGA-DBSCAN	0.96	0.96	19.82
Enhanced MOGA-DBSCAN	**0.99**	**0.99**	16.45

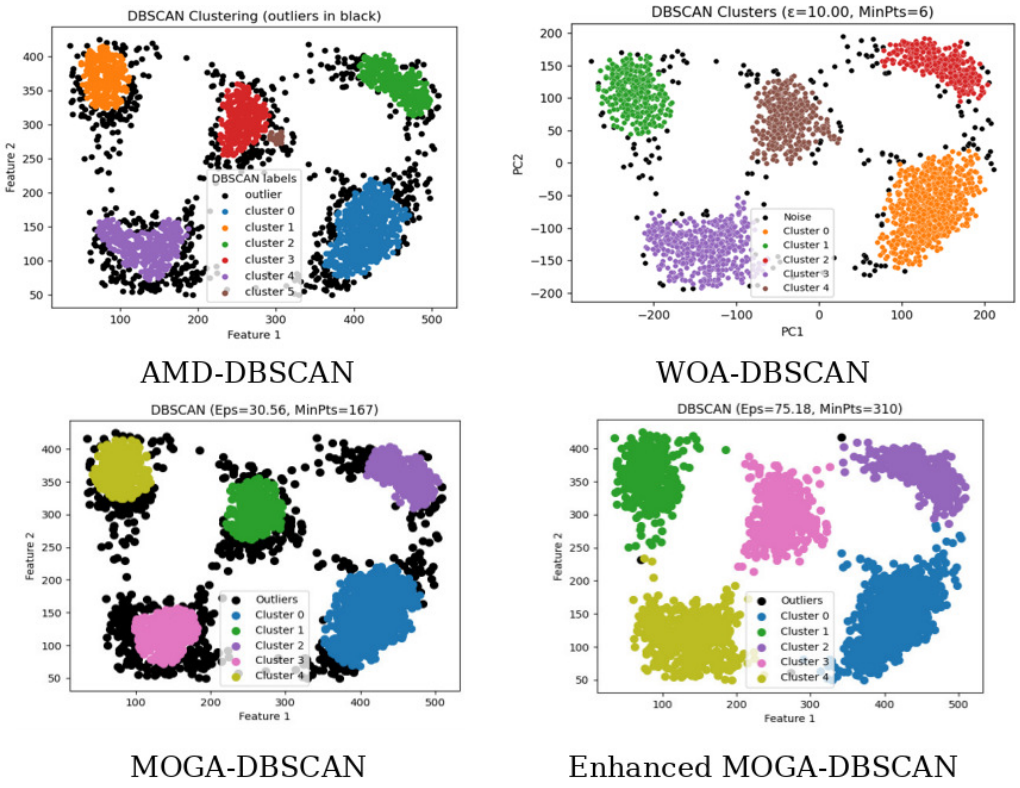

Figure 4: Visual comparison of clustering results on the *Network* dataset.

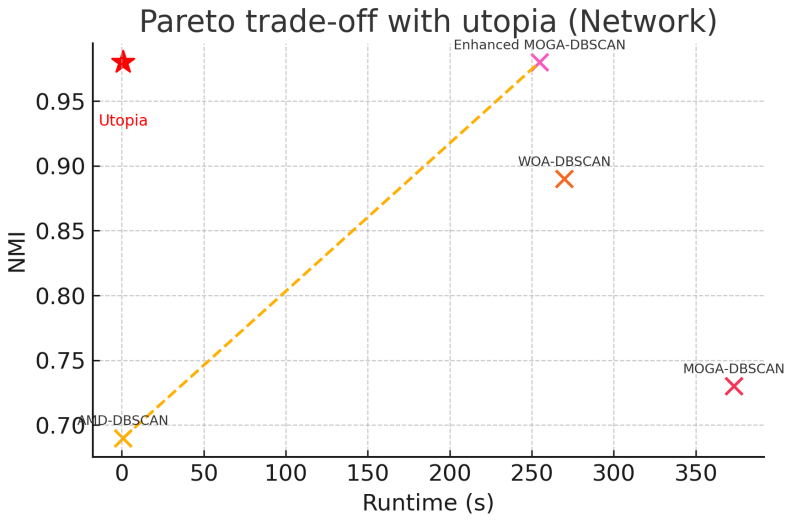

Figure 5: Pareto trade-off on the *Network* dataset after log-runtime scaling. The ⋆ marks the utopia point and the dashed segment is the frontier. Enhanced MOGA-DBSCAN dominates the accuracy extreme.

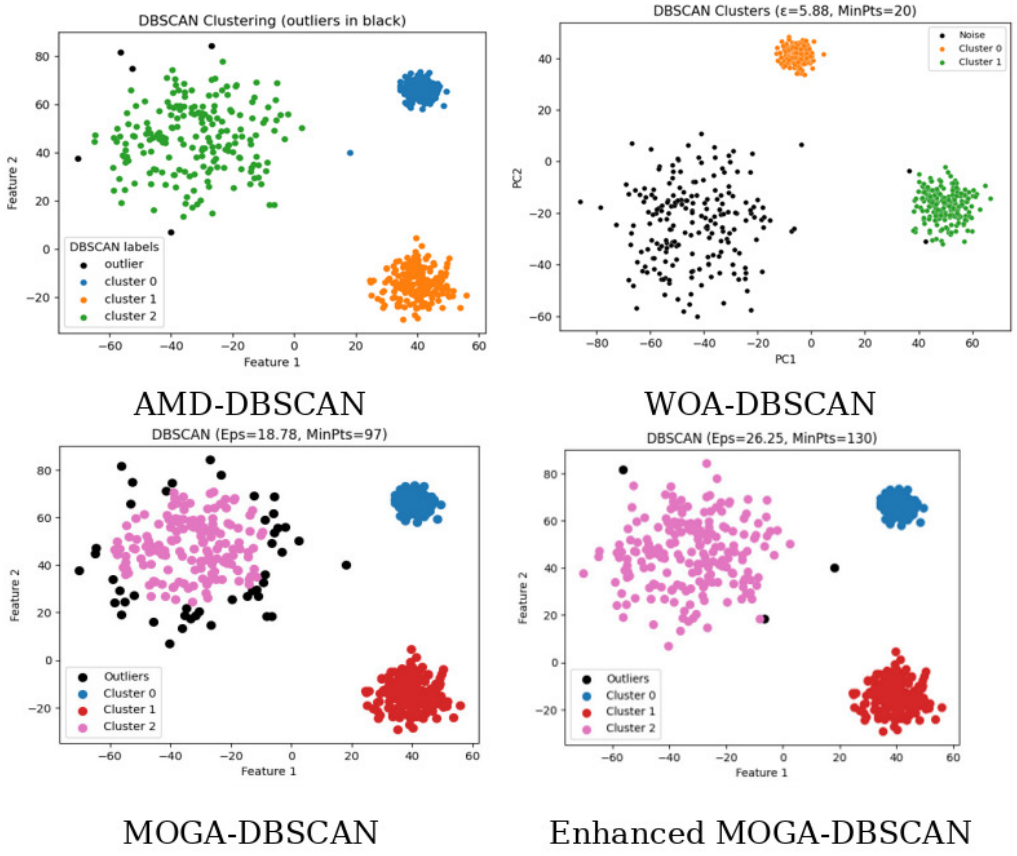

Figure 6: Visual comparison of clustering results on the *Custom-3-Cluster* dataset. Each subpanel shows the output of a different method.

3.2 Custom 3-Cluster dataset

Discussion. Enhanced MOGA-DBSCAN again secures the best NMI and lowest runtime among quality-oriented methods, completing 1.20× faster than MOGA-DBSCAN. By the composite metric it scores $S = 0.83$, comfortably ahead of AMD ($S = 0.47$).

3.3 Quality-speed trade-off

Although the two benchmarks differ in size by a factor of 4.4 (600 vs. 2 634 points), the parallel Enhanced Outlier Index delivers consistent runtime savings-20 % on the small set and 46 % on the large-while preserving the genetic exploration. Coupled with its near-perfect RI and NMI scores, this speed-up yields the highest quality-per-second figure among all contenders.

Enhanced MOGA-DBSCAN therefore strikes the most attractive accuracy-performance compromise: state-of-the-art clustering quality on irregular multi-density data, runtime close to the fastest heuristic, and a decisive lead over its genetic predecessor.

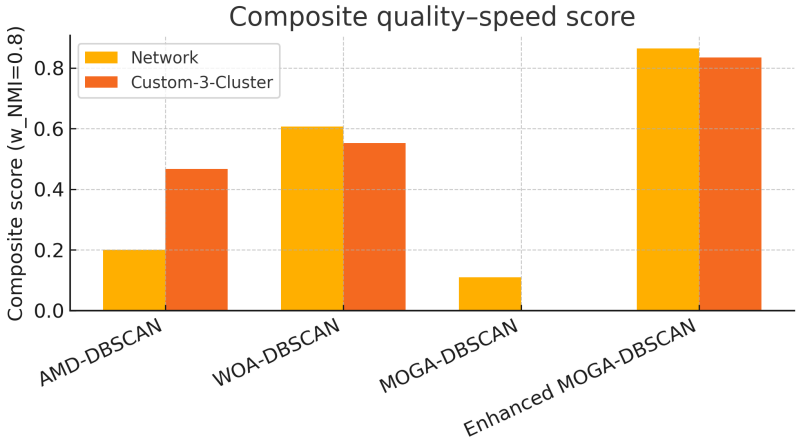

Figure 7: Composite score S across both benchmarks. Enhanced MOGA-DBSCAN leads on each dataset.

3.4 Scaling behaviour

Although our evaluation covers only two benchmarks, their sizes differ by more than a factor of four (600 versus 2634 points), which is sufficient to reveal the impact of the parallel Enhanced Outlier Index (EOI). Table 3 summarises wall-clock times for the original MOGA-DBSCAN and our Enhanced variant.

Table 3: Runtime scalability of Enhanced MOGA-DBSCAN.

Dataset	# points	MOGA (s)	Enhanced (s)	Speed-up
Custom-3-Cluster	600	19.82	16.45	1.20×
Network	2 634	373.09	254.74	1.46×

Two observations stand out:

(a) **Speed-up grows with n:** The parallel EOI reduces runtime by only 20 % on the smaller dataset but by 46 % on the larger one, indicating that the fixed overhead of thread creation and reduction is quickly amortised as the number of points increases.

(b) **Per-point cost remains stable:** Runtime per point for the Enhanced algorithm rises from 0.027 s (Custom) to 0.097 s (Network), a trend consistent with the $O(n^2)$ silhouette term that dominates both variants. Crucially, the gap between the two curves widens with n, matching the theoretical expectation $T_{\text{Enhanced}}(n,p) \approx T_{\text{MOGA}}(n)/p$ for fixed thread count p.

4 Future work

While Enhanced MOGA-DBSCAN shows promising results, several avenues for future research remain:

4.1 Time complexity optimization

The current time complexity of $O(g \times n^2)$ could be reduced through:

- Implementation of better Silhouette score calculation to achieve lower complexity .
- Optimization of density calculations .

4.2 High-dimensional adaptations

To enhance performance in high-dimensional spaces, future work will focus on:

- Development of specialized distance metrics for high-dimensional clustering
- Implementation of feature relevance weighting mechanisms

These improvements would extend the algorithm's applicability to more complex datasets while maintaining its computational efficiency. Additionally, exploring online learning capabilities for streaming data could further enhance the algorithm's practical utility.

Acknowledgements

The first named author, Hossein Eyvazi, gratefully acknowledges the guidance and support of Professor Dr. Ali Rajaei throughout this research.

He also thanks Fatemeh Mohammadi, SeyedAli GhaziSaeed, Mohammad Hossein Soltani, and Mohammad Badzohereh for their valuable support and discussions during his time at Tarbiat Modares University.

References

[1] R.J. Campello, D. Moulavi, A. Zimek, J. Sander, *Hierarchical density estimates for data clustering, visualization, and outlier detection*, ACM Trans. Knowl. Discov. Data 10 (2015) 1–51.

[2] D. Deng, *DBSCAN clustering algorithm based on density*, in: *Proc. 7th Int. Forum Electr. Eng. Autom. (IFEEA)*, IEEE, 2020, pp. 949–953.

[3] A. Dudek, *Silhouette index as clustering evaluation tool*, in: *Classification and Data Analysis: Theory and Applications 28*, Springer, 2020, pp. 19–33.

[4] H. Eyvazi, A. Rajaei, *Enhanced MOGA-DBSCAN: Improving clustering quality with a density-adjusted outlier index*, in: *Proc. 1st Int. Conf. Mach. Learn. Knowl. Discov. (MLKD)*, 2024, pp. 323–328.

[5] Z. Falahiazar, A. Bagheri, M. Reshadi, *Determining the parameters of DBSCAN automatically using the multi-objective genetic algorithm*, J. Inf. Sci. Eng. 37 (2021) 157–183.

[6] K. Li, X. Gao, X. Jia, B. Xue, S. Fu, Z. Liu, X. Huang, Z. Huang, *Detection of local and clustered outliers based on the density–distance decision graph*, Eng. Appl. Artif. Intell. 110 (2022) 104719.

[7] Y. Liu, G. Yin, *The Delaunay triangulation learner and its ensembles*, Comput. Statist. Data Anal. 152 (2020) 107030.

[8] Y. Ren, J. Pu, Z. Yang, J. Xu, G. Li, X. Pu, S. Y. Philip, L. He, *Deep clustering: A comprehensive survey*, IEEE Trans. Neural Netw. Learn. Syst. (2024) (in press).

[9] E. Schubert, J. Sander, M. Ester, H.-P. Kriegel, X. Xu, *DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN*, ACM Trans. Database Syst. 42 (2017) 1–21.

[10] H.V. Singh, A. Girdhar, S. Dahiya, *A literature survey based on DBSCAN algorithms*, in: *Proc. 6th Int. Conf. Intell. Comput. Control Syst. (ICICCS)*, IEEE, 2022, pp. 751–758.

[11] S. Verma, M. Pant, V. Snasel, *A comprehensive review on NSGA-II for multi-objective combinatorial optimization problems*, IEEE Access 9 (2021) 57757–57791.

[12] Z. Wang, Z. Ye, Y. Du, Y. Mao, Y. Liu, Z. Wu, J. Wang, *AMD-DBSCAN: An adaptive multi-density DBSCAN for datasets of extremely variable density*, in: *Proc. IEEE 9th Int. Conf. Data Sci. Adv. Anal. (DSAA)*, 2022, pp. 1–10.

[13] X. Xu, S. Ding, L. Wang, Y. Wang, *A robust density peaks clustering algorithm with density-sensitive similarity*, Knowl.-Based Syst. 200 (2020) 106028.

[14] X. Zhang, S. Zhou, *WOA-DBSCAN: Application of whale optimization algorithm in DBSCAN parameter adaption*, IEEE Access 11 (2023) 91861–91878.

[15] Y. Zhou, W. Zhang, J. Kang, X. Zhang, X. Wang, *A problem-specific non-dominated sorting genetic algorithm for supervised feature selection*,Inf. Sci. 547 (2021) 841–859.