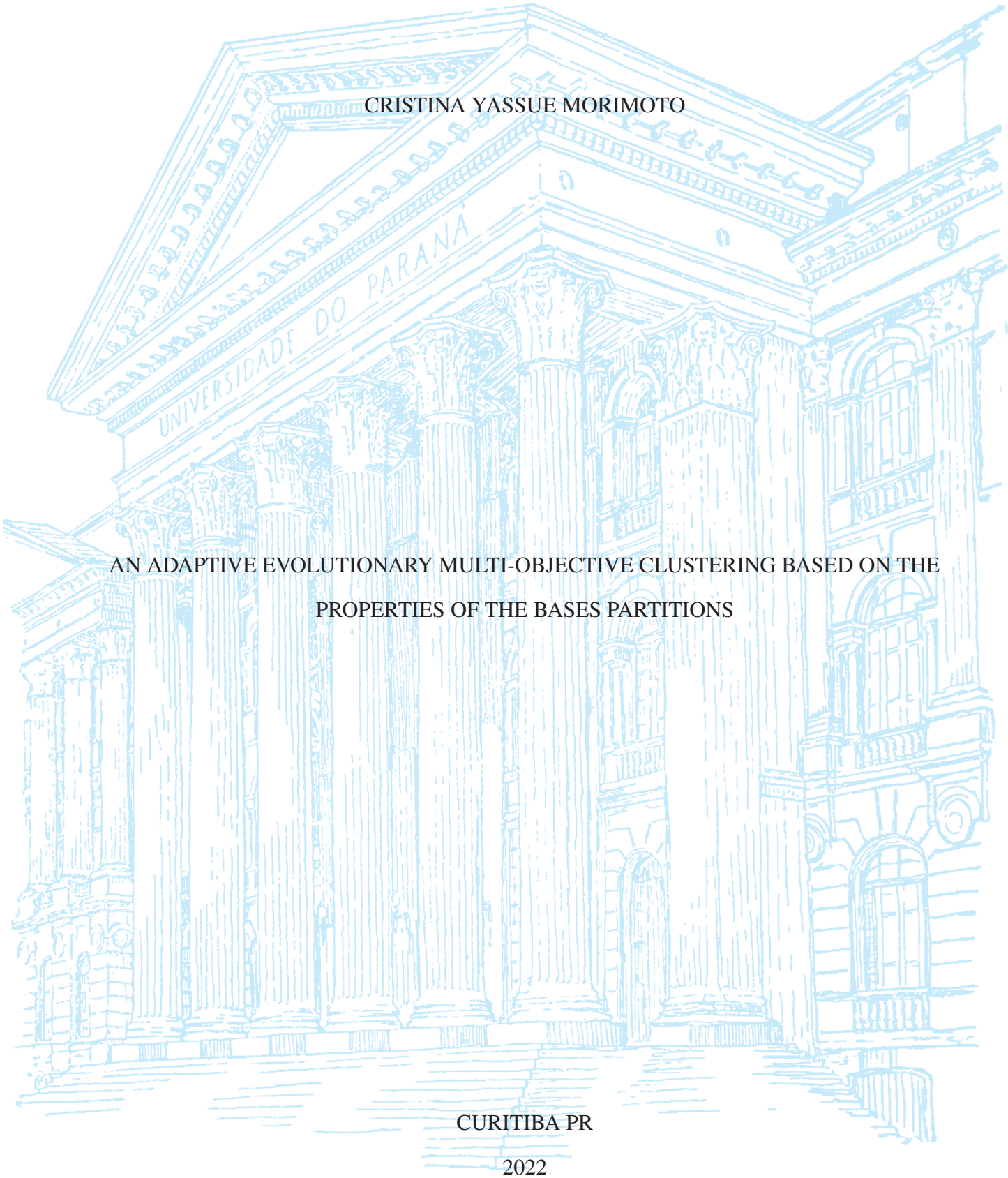
UNIVERSIDADE FEDERAL DO PARANÁ

CRISTINA YASSUE MORIMOTO



AN ADAPTIVE EVOLUTIONARY MULTI-OBJECTIVE CLUSTERING BASED ON THE PROPERTIES OF THE BASES PARTITIONS

CURITIBA PR

2022

CRISTINA YASSUE MORIMOTO

AN ADAPTIVE EVOLUTIONARY MULTI-OBJECTIVE CLUSTERING BASED ON THE PROPERTIES OF THE BASES PARTITIONS

Tese apresentada como requisito parcial à obtenção do grau de Doutor em Ciência da Computação no Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Aurora Trinidad Ramirez Pozo.

Coorientador: Marcílio Carlos Pereira de Souto.

CURITIBA PR

2022

Catalogação na Fonte: Sistema de Bibliotecas, UFPR
Biblioteca de Ciência e Tecnologia

M857a Morimoto, Cristina Yassue

An adaptive evolutionary multi-objective clustering based on the properties of the bases partitions [recurso eletrônico] / Cristina Yassue Morimoto – Curitiba: UFPR, 2022.

Tese (Doutorado) apresentada como requisito parcial à obtenção do grau de Doutor em Ciência da Computação no Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, da Universidade Federal do Paraná. Área de concentração: Ciência da Computação.

Orientador: Aurora Trinidad Ramirez Pozo.
Coorientador: Marcílio Carlos Pereira de Souto

1. Ciência da Computação. 2. Agrupamento de dados. I. Pozo, Aurora Trinidad Ramirez. II. Souto, Marcílio Carlos Pereira de. III. Universidade Federal do Paraná. IV. Título.

CDD 006.3

Bibliotecária: Vilma Machado CRB-9/1563

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação INFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da tese de Doutorado de **CRISTINA YASSUE MORIMOTO** intitulada: **An Adaptive Evolutionary Multi-objective Clustering Based on the Properties of the Bases Partitions**, sob orientação da Profa. Dra. AURORA TRINIDAD RAMIREZ POZO, que após terem inquirido a aluna e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de doutora está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

CURITIBA, 10 de Dezembro de 2022.

Assinatura Eletrônica
12/12/2022 13:26:04.0
AURORA TRINIDAD RAMIREZ POZO
Presidente da Banca Examinadora

Assinatura Eletrônica
12/12/2022 13:42:16.0
RENATO TINÓS
Avaliador Externo (UNIVERSIDADE DE SÃO PAULO)

Assinatura Eletrônica
12/12/2022 17:56:25.0
CARMEM SATIE HARA
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica
15/12/2022 11:15:06.0
MARCILIO CARLOS PEREIRA DE SOUTO
Coorientador(a)

Assinatura Eletrônica
13/12/2022 14:02:53.0
FRANCISCO DE ASSIS TENORIO DE CARVALHO
Avaliador Externo (UNIVERSIDADE FEDERAL DE PERNAMBUCO)

This thesis is dedicated to everyone who supported me throughout my education.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my advisors, Aurora Pozo and Marcílio C. P. de Souto, for their patience and faith in me. Thanks for their invaluable advice on the research subject.

I am sincerely grateful to my family, friends, and labmates for their support in helping me finish this study. Thank you for your understanding, caring, and continued support for the completion of this research work.

RESUMO

O agrupamento de dados evolutivo multiobjetivo (AEM) é uma técnica moderna de agrupamento de dados em que os conceitos gerais de otimização evolutiva de multiobjetivos são aplicados no problema de agrupamento. O projeto e definição de algoritmos de agrupamento de dados é um problema difícil, no qual a escolha das funções objetivo e definição dos parâmetros de configuração ainda são desafios. Neste estudo, visando compreender esse campo, mapeamos e analisamos as abordagens existentes e avaliamos suas principais características. Esta análise demostrou que, em geral, a escolha das funções objetivo considera apenas as propriedades de agrupamento desejadas, e a maioria das abordagens AEM presentes na literatura não considera aspectos de otimização multiobjetivo, como a direção de busca, em seu projeto. Visando apoiar uma melhor escolha e definição dos objetivos nas abordagens AEM, neste manuscrito propomos uma análise da admissibilidade dos critérios de agrupamento para examinar a direção de busca e avaliar seu potencial em encontrar resultados ótimos. Para tanto, consideramos os fundamentos associados a avaliação de uma função heurística para analisar os critérios de agrupamento de dados e demonstrar como eles podem influenciar a otimização. Como resultado, apresentamos uma análise detalhada das principais funções objetivo encontradas na literatura e avaliamos como a inicialização interfere na sua admissibilidade. Além disso, observamos alguns problemas no projeto de algoritmos estabelecidos, os quais não consideram como a estratégia de inicialização pode impactar na busca em termos das funções objetivo aplicadas. Podendo limitar ou piorar os resultados encontrados na inicialização. Para tratar esta questão propomos o AEMOC (Adaptive Evolutionary Multi-objective Clustering approach based on data properties). Essa abordagem considera a propriedades das partições base para determinar se a otimização é necessária ou não. Para isso, propomos uma métrica para medir o grau de separação dos dados, que estima a qualidade relativa da população inicial gerada pelo agrupamento de árvores geradoras mínimas. Além disso, esta avaliação permite definir uma seleção offline de funções objetivas e configurações de parâmetros do algoritmo multiobjetivo. O AEMOC apresentou resultados promissores considerando um conjunto diversificado de conjuntos de dados artificiais e reais, considerando dois aspectos: obteve sucesso na definição da qualidade relativa das partições de base e forneceu melhores resultados de agrupamento do que as abordagens AEM de referência.

Palavras-chave: Agrupamento de dados. Otimização multiobjetivo. Agrupamento de dados multiobjetivo.

ABSTRACT

Evolutionary multi-objective clustering (EMOC) is a modern clustering technique in which the general concepts of evolutionary multi-objective optimization are applied to the clustering problem. The design and definition of the clustering are difficult problems in which the choice of the objective functions and parameter setting of the algorithms are still challenges. In our study, aiming to understand this field, we mapped and analyzed the existing approaches and evaluated their main characteristics. This analysis showed that many different objective functions and initialization strategies have been applied in EMOC approaches. In general, the choice of the objective functions only considers the desired clustering properties, and most EMOC approaches present in the literature do not consider aspects of multi-objective optimization, such as the search direction, in their design. Aiming to support a better choice and definition of the objectives in the EMOC approaches, we introduce an analysis of the clustering criteria admissibility to examine the search direction and evaluate their potential for finding optimal results. We consider the fundamentals of the evaluation of a heuristic function to analyze the clustering criteria and demonstrate how they can influence the optimization. As a result, this study provides a detailed analysis of the main objective functions found in the literature and evaluates how the initialization interferes with their admissibility. Besides that, we observed some issues in the design of established algorithms that do not consider the impact of the initialization strategy on the search when determined objective functions are applied. This aspect can limit the clustering or worsen the results found in the initialization. To amend this matter, in this manuscript, we propose the AEMOC (adaptive evolutionary multi-objective clustering approach based on data properties). This approach considers the properties of the base partitions to determine whether optimization is required or not. For that, we propose a metric to measure the data separation degree that estimates the relative quality of the initial population generated by minimum spanning tree clustering. Furthermore, this evaluation makes it possible to define an offline selection of objective functions and parameter settings for the multi-objective algorithm. AEMOC presented promising results considering a diverse set of artificial and real-life datasets, considering two aspects: it succeeded in the definition of the relative quality of the base partitions, and it provided better clustering results than reference EMOC approaches.

Keywords: Clustering. Multi-objective optimization. Multi-objective clustering.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

ABGSS	Average Between Group Sum of Squares
ACM	Association for Computing Machinery
AEMOC	Adaptive evolutionary multi-objective clustering approach based on data proprieties
AI	Artificial Intelligence
AL	Average-Linkage
ARI	Adjusted Rand Index
AUC	Area under the curve
AUCC	Area Under the Curve for Clustering
AWGSS	Average Within Group Sum of Squares
CC	Cheng and Church's Algorithm
CCDG-K	Constrained Decomposition with Grids
CCI	Cluster Cardinality Index
CDCS	Categorical Data Clustering with subjective factors
CBO	Constraint-Based Overlap value
CH	Calinski-Harabasz index
CoL	Complete-Linkage
COSEC	Compactness and Separation Measure of Clusters
CL	Cannot-link (CL)
Con	Connectivity Index
ConP	Connectivity index based on the Pearson Correlation
CVI	Clustering Validity Index
DCD	Data Continuity Degree
DSD	Data Separation Degree
Dev	Overall Deviation
DB	Davies-Bouldin index
DI	Degree of Interestingness
DM	Decision Maker
Dunn	Dunn Index
ICONIP	International Conference of Neural Information Processing
IN	inadmissible
IMOCLE	Improvement of the Multi-Objective Clustering Ensemble algorithm
EA	Evolutionary Algorithm
EMCOC	Evolutionary Multi-objective Clustering for detecting overlapping clusters

EMOC	Evolutionary Multi-Objective Clustering
EMO-KC	Evolutionary Multi-objective k-clustering
Ent	Intra-cluster Entropy
EvoApplications	International Conference on the Applications of Evolutionary Computation
EWCD	Expected Weighted Coverage Density
FCM	Fuzzy C-Means
FRC	Fuzzy Relational Clustering
FTS	Frequent Terms Set
H	Homogeneity
HBGF	Hybrid Bipartite Graph Formulatation
HCM	H-Confidence Metric
HT-MOC	Hierarchical topology-based MOC
IEEE	Institute of Electrical and Electronic Engineers
J_m	Fuzzy compactness
J_{add}	Addition feature weight
JGGA	Jumping Genes Genetic Algorithm
KM	K-Means
Km_{ed}	K-Mode external distance
Km_{id}	K-Mode internal distance
$Km_w ed$	K-Mode weighted external distance
$Km_w id$	K-Mode weighted internal distance
KNN	k-Nearest Neighbours
LAG	Locus adjacency graph
MaOEA	Many-Objectives Evoluctionary Algorithm
MCLA	Meta Clustering Algorithm
ML	Must-link
MOCA	Multi-Objective Clustering Algorithm
MIE-MOCK	Multiple Information Exchange Multi-Objective Clustering with automatic k-determination
MOAC-L	locus-based multi-objective automatic clustering
MOC	Multi-objective clustering
MOCK	Multi-Objective Clustering with automatic K-determination
MOCLE	Multi-Objective Clustering Ensemble
Mod	Modularity index
MOEA	Multi-Objective Evoluctionary Algorithm
MOEASSC	Multi-Objective Evolutionary Approach-based Soft Subspace Clustering
MOEA/D	Multi-Objective Evolutionary Algorithm Based on Decomposition

MOEA/DD	Multi-Objective Evolutionary Algorithm based on Dominance and Decomposition
MOECDM	Multi-objective Evolutionary Clustering Based on Combining Multiple Distance Measures
MOEACDM	Multi-objective Evolutionary Automatic Clustering based on Combining Multiple Distance Measures
MOGA	Multi-Objective Genetic Algorithm
MOGGC	Multi-Objective Genetic Graph-based Clustering Algorithm
MOO	Multi-objective optimization
MOKCW	Multi-Objective Kernel Clustering algorithm
MOP	Multi-Objective Optimization Problem
MOSSC	Multi-Objective evolutionary algorithm-based Soft Subspace Clustering
MST	Minimum Spanning Tree
NPGA	Niched Pareto Genetic Algorithm
NSGA-II	Non-dominated Sorting Genetic Algorithm version 2
NSGA-III	Non-dominated Sorting Genetic Algorithm version 3
OP	Optimal solution
PBM	Pakhira, Bandyopadhyay and Maulik index
PC	Person Correlation
PF	Pareto front
PESA-II	Pareto Envelope-based Selection Algorithm version 2
PSVIndex	Projection Similarity Validity Index
RQ	Research Question
RVEA	Reference Vector Guided Evolutionary Algorithm
ROC	Receiver Operating Characteristics
SBX	Simulated Binary Crossover
SDE	Shift-Based Density Estimation
Sep_{AL}	intra-cluster average Separation
Sep_{CL}	Separation index
Sep_{fuzzy}	Fuzzy Separation index
Sep_{nfuzzy}	Fuzzy Overlap Separation index
Sep_{graph}	Graph-based Separation index
Sim	Similarity
Sil	Silhouette index
SIVID	Sum of Internal Validity Indices with Diversity
SL	Single Linkage
SNN	Shared Nearest Neighbor
SP	Sparsity

SPC	Spectral Clustering
SPEA2	Strength Pareto Evolutionary Algorithm version 2
SSD	Sum of Squared Distance
SSXB	Soft Subspace Xie-Beni index
TWCV	Total Within-Cluster Variance
RE	Reconstruction Error
Var	Intra-cluster Variance
VRJGGA	Variable-length Real Jumping Genes Genetic Algorithm
XB	Xie-Beni index
WSN	Wireless Sensor Network

LIST OF SYMBOLS

$\mathbf{c}_i$	i-th cluster of a partition π
$d(\mathbf{x}_a, \mathbf{x}_b)$	distance between two objetcts
$d(\mathbf{x}_a, \mathbf{z}_i)$	distance between a object to its centroid
$d(\mathbf{z}_i, \mathbf{z}_j)$	distance between two centroids
$\mathbf{F}(\pi)$	a set of objective functions
$f(\pi)$	objective function
$g(\pi)$	restriction (inequality function)
$h(\pi)$	heuristic function
$k*$	number of clusters in the true partition
k_i	number of the clusters of a partition π_i
$k\Pi_{max}$	maximum number of the clusters in the partitions of Π
$k\Pi_{min}$	minimal number of the clusters in the partitions of Π
$k\Pi_{mean}$	mean number of the clusters in the partitions of Π
$l(\pi)$	restriction (equality function)
L	number of the nearest neighbors (neighborhood size)
m	fuzzy exponent
$\mathbf{m}_i$	cluster mode
n	number of the objects in a dataset $\mathbf{X}$
n_i	number of the objects in the c_i
nn_{ab}	b-th nearest neighbor of the object $\mathbf{x}_a$
p	total number of restriction (inequality functions)
q	total number of restriction (equality functions)
round()	function to get the closest integer value of a float value
s	size of the initial population
s'	size of the selected set of individual to be presented to the final user
$sep_{CL}\Pi_{max}$	maximum separation result in the partitions of Π
$sep_{CL}\Pi_{min}$	minimal separation result in the partitions of Π
$sep_{CL}\Pi_{mean}$	mean separation result in the partitions of Π
$\mathbf{z}_i$	centroid of the i-th cluster of the solution
$\overline{\mathbf{z}}$	centroid of the dataset
u_{ia}	e membership degree of the ath data point to the ith cluster
$\mathbf{X}$	a set of points (dataset)
$\mathbf{x}_a$	a-th point or object of the data-set
ϵ_i	maximum distance between the object $\mathbf{x}_i$ and the k-nearest neighbor in a particular neighborhood

π	a partition or solution
Π	a set of partitions
Π_0	base partitions (initial population)
Π_{nonD}	a set of non-dominated partitions
μ_{ij}	fuzzy membership
τ	fuzzy weighting index

CONTENTS

1 INTRODUCTION

The use of knowledge discovery techniques has become essential to analyze and understand large volumes of data generated in different fields of application (e.g. marketing, medicine, bioinformatics). Clustering analysis has been widely studied and adopted for several purposes, including pattern analysis, image segmentation, data mining, and decision-making. Clustering is a type of unsupervised learning whose goal is to find the underlying structures that compose finite sets of data (clusters), in which the objects or observations belonging to a cluster should share some relevant property (similarity) regarding the data domain. In other words, in clustering, there is an absence of category information that distinguishes data clustering (unsupervised learning) from classification (supervised learning) (Aggarwal and Reddy, 2014).

There are several clustering algorithms that have been proposed in different fields of research. However, in spite of that, clustering remains a difficult problem. As described by Jain (2010), this can be attributed to the inherent vagueness in the definition of a cluster, and, in particular, the difficulty in defining an appropriate objective function.

In recent years, multi-objective evolutionary algorithms (MOEAs) have become a popular method applied for clustering. In general, clustering studies that consider this kind of approach are referred to as evolutionary multi-objective clustering (EMOC). They are capable of obtaining a set of solutions that represent the trade-off between different objectives. They use multiple criteria (e.g., compactness and connectedness) as objective functions to deal with datasets with different types of clusters. However, the evolutionary-based clustering methods are still under-explored in the literature, deserving more attention and investigation (Zhu et al., 2020). In particular, the design and definition of the clustering problem are still challenges, in which issues related to the definition of the objective functions and initialization strategy emerge in evolutionary multi-objective optimization, in addition to other difficulties.

Some studies, such as Hruschka et al. (2009); Mukhopadhyay et al. (2015), introduced some approaches present in the literature but were limited in listing their components and main features. In recent studies presented by Wang et al. (2018, 2020), the authors provide analysis regarding the generation and maintenance of diversity of solutions in EMOC approaches. Other studies have evaluated the objective functions for evolutionary multi-objective data clustering. Handl and Knowles (2012) present a comparison of four criteria pairs for multi-objective clustering in datasets with different types of clusters. Barton and Kordík (2015) investigated some clustering criteria and analyzed their correlation with the ground truth to develop an evolutionary multi-objective clustering algorithm. However, these studies do not consider evaluating the search direction of objective functions and the impact of the initial population on the evolutionary optimization.

1.1 RESEARCH QUESTIONS AND GOALS

Aiming to improve the research in this field and provide insights regarding the design of EMOC, we raised the following research questions (RQ):

> **RQ.1**: How to evaluate the objective functions and define the best combination of the clustering criteria applied in EMOC?
>
> **RQ.2**: How does the initialization strategy affect the optimization?

RQ.3: How to improve the design and the parameter setting of the EMOC?

These RQs guided our general studies and analysis. Regarding **RQ.1** and **RQ.2**, they motivated the mapping of the existing approaches to analyze the general clustering criteria applied in the EMOC approaches. However, it demonstrated a lack of studies that analyze both the clustering and optimization features in order to determine EMOC objective functions and initialization strategy. In the literature, there are a variety of clustering criteria being applied in the EMOC approaches, but most studies do not provide any analysis regarding the choice and combination of the objective functions. Fundamentally, they are selected concerning some desired proprieties in the data clustering. The same occurs in terms of the initialization strategy. Many approaches make use of established clustering algorithms without a prior study of the impact of the search. This superficial analysis, which only concerns the clustering aspects, can promote a wrong usage of multi-objective optimization with the application of inadequate objective functions. Thus, we rely on the fundamentals of artificial intelligence in defining heuristic functions to evaluate the clustering criteria and determine whether a function can lead the search to the optimal results based on the admissibility. The proposed analysis of the inadmissibility of the clustering criteria, presented in Chapter 4, unifies the fundamentals of clustering and optimization to promote good practices that contribute to the improvement of this research field. Furthermore, this analysis considers the main objective functions and initialization strategies applied in the literature. This broad view of the EMOC allows new practionaries and students to be more observant to fundamentals and do not reproduce the same mistakes found in some popular approaches.

Here, we characterize admissible objective functions as having the property of detecting the "natural" ground-truth clustering, that is, the one with the optimal value. In contrast, the inadmissibility analysis refers to evaluating the search direction to define a non-admissible objective function, where the search does not lead to finding the ground-truth clustering. In other words, our study introduces an investigation into the inadmissibility of the objective functions applied to evolutionary multi-objective clustering that supports defining whether the objectives are worth optimizing. It is important to note that we evaluate the (in)admissibility regarding the search direction. This is different from the study presented by Fisher and Ness (1971), where the authors consider the admissibility of the clustering algorithms by considering the evaluation of the structure of the data essentially.

In particular, the analysis of the inadmissibility demonstrated issues regarding the design of some established EMOC approaches. Some initialization strategies limit the search or provide the optimal results in respect of some clustering criteria, and the optimization is inadequate or not required in terms of the objective functions applied. In particular, approaches that use traditional clustering algorithms (such as k-means (MacQueen, 1967), average-linkage (Sokal, 1958), among others) to generate the initial population (base partitions or candidate solutions) do not evaluate the impact of using high-quality partitions in the initial population, or even how they affect the optimization. However, our analysis showed that, depending on the data structure nature and criterion applied in the initialization strategy, the optimal result can be found in the base-partitions and the optimization is not required.

In terms of **RQ.3**, we verified that it is essential to distinguish the data properties of base partitions to avoid unnecessary processing, as observed in the admissibility analysis. Thus, we considered the general capabilities of minimal spanning tree (MST) clustering (Handl et al., 2007) (a clustering algorithm) in detecting well-separated arbitrary shaped clusters and proposed an evaluation method to estimate the relative quality of candidate solutions generated by MST-clustering based on the general separation and overlap of the data. For that, a new metric to measure the data separation degree (DSD) was introduced to evaluate the general data separation

considering some observed aspects of the initial population. This metric was used along with a semi-supervised metric called the constrained-based overlap value (CBO) (Adam and Blockeel, 2017) to obtain the necessary information to explore specific configurations in the optimization. In particular, CBO is applied to measure the overlap of the data. In Chapter 5, we present DSD and CBO, presenting some general features that were used in the definition of the evaluation method described in Section 6.2. The estimated quality results provide information to define whether the optimization is required or not, avoiding unnecessary data processing.

Based on the studies regarding the RQs, we generated a new approach, AEMOC - adaptive evolutionary multi-objective clustering approach based on data proprieties, as present in Chapter 7. AEMOC provides a new view of the modeling of multi-objective clustering approaches. The main idea of this approach is the use of an evaluation method to estimate the relative quality of the base partitions and determine the objective functions and parameter settings of the multi-objective algorithm. Here, the relative quality refers to the data proprieties in which the initialization strategy has good (or poor) clustering performance.

This approach presented promising results considering a diverse set of artificial and real-life datasets in two aspects: it succeeded in the definition of the relative quality of the base partitions generated by MST-clustering and it provided better clustering results than reference EMOC approaches. In general, as the clustering topic is studied in several research areas, we consider that our analysis and results promote good practices that contribute to the improvement of this research field.

1.2 CONTRIBUTIONS

The contributions of this work are described as follows:

- The introduction of the admissibility analysis in the clustering problem, which evaluates the search direction of the objective functions. Many existing approaches only consider clustering aspects in the choice of the objective functions, which does not observe the influence of the other aspects, such as the initialization, in the optimization.

- A broad analysis of a variety of clustering criteria applied as objective functions in existing EMOC approaches. We observe the general impact of the initialization in the admissibility, including common issues found in the design of established algorithms.

- The introduction of a new EMOC approach, called AEMOC, provides new features in the design of multi-objective clustering that estimate the relative quality of the initial population to determine whether an optimization is required or not, and performs a selection of objective functions and parameter setting of the multi-objective algorithm.

- The introduction of a new metric, DSD, that is applied to evaluate the separation of the base partitions generated by MST-clustering.

- The analysis and comparison of established EMOC approaches, MOCK (Handl et al., 2007), MOCLE (Faceli et al., 2006), and EMO-KC (Wang et al., 2018), with EAMOC was conducted in order to illustrate and explain how our approach amends some common issues observed in these algorithms.

The results of these studies were reported in different conferences and journals. The first one, as co-authors of the analysis of the established EMOC approaches, in “Multi-objective clustering: A data-driven analysis of MOCLE, MOCK and Δ-MOCK” published in the annals

of ICONIP'21 (*International Conference of Neural Information Processing*). Furthermore, we proposed an improvement to an objective function, and it was applied to improve the detection of nested structures in "Detecting nested structures through evolutionary multi-objective clustering", published in the annals of EvoApplications'22 (*International Conference on the Applications of Evolutionary Computation*). Additionally, the proposed analysis of the admissibility applied to evaluate the objective functions was published in the journal Information Sciences (Morimoto et al., 2022a).

Currently, we are working on two journal publications: one refers to the literature review considering a detailed explanation of the general EMOC architecture; the other refers to the new approach, AEMOC.

1.3 THESIS ORGANIZATION

The remainder of this manuscript is organized into six chapters. The first two chapters provide the main concepts and features found in the EMOC literature, followed by two chapters that introduce methods and experiments applied to generate a new EMOC approach. In particular, this manuscript is structured as follows:

- **Chapter 2 - Background** presents the main concepts and theoretical background related to clustering and multi-objective optimization problem. Furthermore, in this chapter, we present the general architecture of EMOC, considering the general features found in the literature and the main aspects applied to the clustering problem.

- **Chapter 3 - Literature Review** provides a literature revision of the EMOC approaches. We present a profile of this field by considering an extensive mapping of the literature to identify the main methods and concepts that have been adopted to design the EMOC approaches.

- **Chapter 4 - Analysis of the inadmissibility of the objective functions in EMOC approaches** presents the analysis of the admissibility applied to several clustering criteria, which points out some general issues found in established EMOC approaches.

- **Chapter 5 - Measuring the separation and the overlapping of the data** introduces the metrics applied to measure the separation and overlap of the data, which are used to compose the evaluation method applied to determine the relative quality of base partitions generated by MST-clustering.

- **Chapter 6 - Proposed multi-objective clustering approach** describes the proposed approach: AEMOC - Adaptive evolutionary multi-objective clustering approach based on data proprieties.

- **Chapter 7 - Experiments** presents the results regarding the relative quality estimation of the base partitions, and the clustering results of the experiments that compare the clustering performance of the AEMOC with other reference approaches.

- **Chapter 8 - Conclusion** presents a summary of the thesis and future work.

2 BACKGROUND

In this section, we introduce basic concepts in clustering and multi-objective optimization. In particular, we describe the main features and concepts considered in the general EMOC architecture, along with the main elements applied in designing EMOC algorithms found in the literature.

2.1 PRELIMINARIES

The convention adopted in this thesis considers the terms, clustering criteria, objective functions, fitness functions, and heuristic functions, interchangeably to represent the multi-objective clustering problem's goals or objectives (i.e., distinct mathematical functions):

- ***Clustering Criteria***: A clustering criterion (function) guides the selection of features and clustering schemes in a clustering algorithm.
- ***Objective Functions***: The objective function refers to a criterion that should be maximized or minimized in an optimization problem.
- ***Fitness Functions***: A fitness function is a particular type of objective function that quantifies the optimality of a solution. The fitness functions are used in evolutionary approaches to guide the search towards optimal design solutions.
- ***Heuristic Functions***: A heuristic is a term adopted in artificial intelligence (AI) that works by guiding search, suggesting behavior, making decisions, or transforming the problem. A heuristic function guides the decision, as a strategy or simplification, to limit the search for solutions in large problem spaces.

In this study, instead of using the term fitness function, we rely on the general term used in evolutionary multi-objective optimization: the objective function. However, each term is important to facilitate the general understanding of the content of this thesis, considering that they relate to different fields of study.

2.2 CLUSTERING AND MULTI-OBJECTIVE OPTIMIZATION

Data clustering consists of the decomposition of finite and unlabeled data into subgroups based on similar attributes, or naturally occurring trends, patterns, or relationships in the data (Jain and Dubes, 1988). There is not a unique and formal definition of a cluster since the clustering methods and algorithms were proposed for researchers in different fields and applied to a variety of problems and distinct goals. In general, some general properties for cluster analysis are considered (Hruschka et al., 2009; Rai and Singh, 2010; Faceli et al., 2011):

(a) **Well-separated clusters** represent clusters where each object is closer (more similar) to all of the objects in its cluster than to any object in another cluster;

(b) **Connected or contiguous clusters** refer to clusters in which each object is closer to at least one object in its cluster than to any object in another cluster;

(c) **Compact clusters** represent clusters with small intra-cluster variation, considering the variation between same-cluster data items or between data items and clusters;

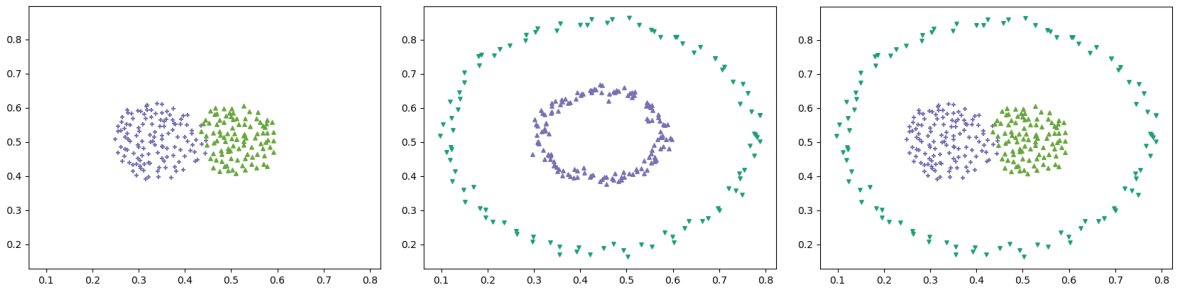
(d) **Center-based clusters** represent clusters in which each object is closer to the center of its cluster than to the center of any other cluster;

(e) **Density-based clusters** denote clusters in which regions of high density are separated by regions of low density.

In terms of the clustering process, in this chapter and in the literature review (Chapters 2 and 3), we consider two general types: hard and soft clustering. Our proposal and analysis, on the other hand, are centered on hard clustering. Formally, given a set of objects $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, an hard (exclusive) partition of $\mathbf{X}$ in k clusters can be defined as $\pi = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k\}$, where $k < n$, such that: $\mathbf{c}_i \neq \emptyset$, for $(i = 1, \ldots, k)$, $\bigcup_{i=1}^{k} \mathbf{c}_i = \mathbf{X}$ and $\mathbf{c}_i \cap \mathbf{c}_j = \emptyset$ for $(i, j = 1, \ldots, k)$ and $i \neq j$. If the condition of mutual disjunction ($\mathbf{c}_i \cap \mathbf{c}_j = \emptyset$, for $(i, j = 1, \ldots, k)$ and $i \neq j$) is relaxed, then the corresponding data partitions are said to be of the soft (fuzzy) type (Hruschka et al., 2009).

It is important to note that we use the term "overlap" to define overlapping areas among categories. In the literature, some works refer to overlap clusters or overlapping clustering as soft clustering.

Regarding the taxonomy of the algorithms, traditional clustering algorithms can be divided into two general categories: partitional and hierarchical. Hierarchical methods produce a nested series of partitions, while partitional methods produce only one (Jain et al., 1999). For example, k-means (KM) (MacQueen, 1967) is a partitional algorithm; single linkage (SL) (Sneath, 1957), average linkage (AL) (Sokal, 1958), and complete linkage (CoL) (Sorensen, 1948) are hierarchical algorithms. In general, traditional clustering algorithms optimize only one clustering criterion and are often very effective for this purpose. However, they may not find all clusters in the datasets with different data structures, or clusters with shapes hidden in sub-spaces of the original feature space.



(a) globular clusters (b) ring shape clusters (c) heterogeneous data structure

Figure 2.1: Different data structures (objects with the same color represent a cluster in each sub-figure)

In contrast, EMOC, a modern clustering type of algorithm, considers the simultaneous optimization of multiple objectives to solve a variety of clustering problems considering different data properties. An EMOC that considers two criteria, compactness-based and connectedness-based, for example, can detect all of the data structures in Fig. 2.1, whereas algorithms that use only the compactness-based criterion, such as KM, can detect globular clusters, as shown in Fig. 2.1(a), but KM cannot find the ring-shaped clusters in Fig. 2.1(b) and the heterogeneous structures in Fig. 2.1(c). In contrast, a connectedness-based algorithm, such as shared nearest neighbor (SNN) (Ertöz et al., 2002), can detect the ring shapes in Fig. 2.1(b), but SNN cannot find the clusters in Fig. 2.1(a) and Fig. 2.1(c).

EMOC applies the concepts of multi-objective optimization (MOO) to the clustering problem. In MOO, the goal is to find a vector of decision variables, π, that satisfies the inequality and equality constraints ($g_i(\pi)$ and $l_j(\pi)$) presented in Eq. 2.2, and optimizes the vector $\mathbf{F}(\pi)$ of z objective functions, Eq. (2.1) (Coello et al., 2006). In particular, identifying a solution π that is feasible and optimizes the objective at hand is notably challenging when restrictions and objectives have a non-linear, non-convex, discrete, or non-differentiable nature. One common approach to dealing with the restrictions is to treat those restrictions as objective functions. For example, in bi-objective optimization, a constraint can be used as a second objective subjected to multi-objective optimization for the formation of a Pareto front (PF), in which the optimization can be focused on the main objective function.

$$\text{minimize/maximize } \mathbf{F}(\pi) = (f_1(\pi), f_2(\pi), \ldots, f_z(\pi)) \tag{2.1}$$

$$\begin{aligned} \text{subjected to } g_i(\pi) \leq 0, \quad & i = \{1, \ldots, p\}, \text{and} \\ l_j(\pi) = 0, \quad & j = \{1, \ldots, q\} \end{aligned} \tag{2.2}$$

Evolutionary algorithms (EAs) are considered well-suitable to MOO because they address both search and multi-objective decision making (while some approaches focus on search and others on multi-criteria decision making) and can search partially ordered spaces for several alternative trade-offs (Fonseca and Fleming, 1995). EA uses a heuristic solution-search or optimization technique based on the principle of evolution through selection. Most multi-objective evolutionary algorithms select solutions using the Pareto dominance relation, in which given two candidate solutions π_i and π_j, π_i dominates π_j (denoted as $\pi_i \prec \pi_j$), if and only if: i) π_i is strictly better than π_j in at least one of all the objectives considered, and ii) π_i is not worse than π_j in any of the objectives considered. The goal of this process is to find the set of all non-dominated solutions, that is, the PF. For example, Fig. 2.2 shows a Pareto set of two objective functions that should be minimized. Points A and B are the non-dominated solutions and hence lie on the Pareto front. Point C is dominated by points A and B, so it does not lie on the frontier (Li et al., 2015).

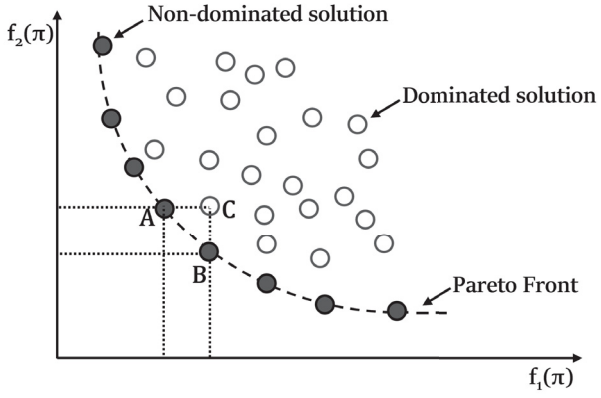


Figure 2.2: Pareto Dominance Relation

Due to their population-based nature, evolutionary algorithms are able to approximate the whole PF of a given multi-objective problem in a single run. Consequently, they have been a popular choice for the design of multi-objective data clustering techniques (Hruschka et al., 2009; Mukhopadhyay et al., 2015). In this context, the multi-objective evolutionary algorithms (MOEA) are applied to solve a multi-objective optimization problem (MOP) with $z \geq 2$. However, the traditional techniques based on Pareto dominance have their effectiveness degraded (convergence

and diversity difficulties) when applied to problems with more than three objectives, and the computational complexity of non-dominated sorting considerably increases. Many-objective evolutionary algorithms (MaOEA) have been proposed to deal with this scalability issue, in which the Many Objectives Problem can be defined as a MOP with $z \geq 4$ (Li et al., 2015).

In terms of the evaluation of the EMOC results, there are two types of assessment: one considering aspects of clustering quality, and the other considering MOO performance, as presented in the following.

2.3 CLUSTERING VALIDATION

The clustering approaches are evaluated regarding clustering validity indices (CVIs), which define how well a partition fits the structure underlying the data. There are three types of criteria (Brun et al., 2007): relative, internal, and external. Relative criteria are based on comparisons of partitions generated by the same algorithm with different parameters or different subsets of the data. Internal criteria refer to quality measures based on calculating properties of the resulting clusters, establishing the validity of a cluster-based exclusively on the dataset itself, for example, how much a cluster is justified by means of the proximity matrix. External criteria lie in prior knowledge of structures in the dataset to evaluate the given partitions generated by an algorithm in contrast with a model partition or labeled data, denominated True Partition, provided by specialists. In Section 2.4.2.1, we present some CVIs and their application in EMOC approaches.

2.4 A GENERAL ARCHITECTURE OF EVOLUTIONARY MULTI-OBJECTIVE CLUSTERING

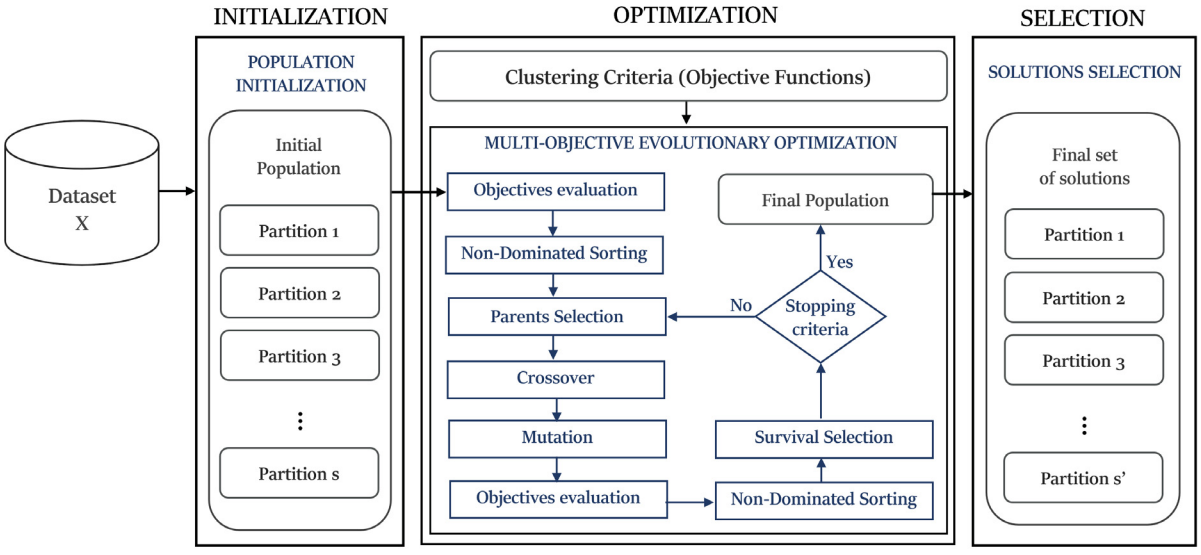


Figure 2.3: A general architecture of Evolutionary Multi-objective Clustering

In this section, we introduce this general architecture of EMOC to describe the main elements applied in designing EMOC algorithms. In the literature, we did not find other studies that provide a clear definition of the main components and their relationships in EMOC approaches. Thus, we illustrate the general architecture of the EMOC in Fig. 2.3, considering 3 modules:

1. **Initialization**: Given a dataset, traditional data clustering algorithms (or random generator methods) are applied to build the partitions (individuals) that compose the initial population. Each partition is a clustering solution with a specific encoding or representation. In section 2.4.1, we detail the types of representations and initialization strategies applied in EMOC.

2. **Optimization**: The initial population is taken as an input to multi-objective evolutionary optimization, in which iteratively the objective functions are minimized (or maximized) to generate a final population. In general, the existing EMOC algorithms rely on general-purpose MOEAs in the optimization flow. Most approaches consider the standard features of a particular MOEA, while using a specific set of objective functions and different combinations of crossover and mutation operators. In section 2.4.2, we detail the optimization phase. We present some traditional MOEAs and introduce other types of multi-objective approaches that consider other aspects in the selection besides Pareto dominance. Furthermore, we point out the main aspects of the objective functions and the evolutionary operators applied in EMOC.

3. **Selection**: MOO approaches may generate large sets of efficient solutions using Pareto dominance. Thus, this module is applied to determine the final set of solutions to be presented to the data experts. According to prior criteria, a suitable number of solutions, s', is selected from the final population in this phase. Partition selection is a specific subject in clustering, in which it is possible to find studies focused on this subject. Therefore, this module is not considered mandatory in the design of EMOC approaches. In Section 2.4.3, we present some strategies applied to EMOC partition selection.

In the following, we present the main concepts and elements of each module of evolutionary multi-objective clustering by introducing the main features of the EMOC approaches described in Section 3.

2.4.1 Initialization Module: Representation and Initialization strategies

The solution representation or chromosome encoding denotes an individual (candidate solution) in the evolutionary algorithm. The choice of the representation should consider the information necessary to be manipulated by the evolutionary operators to generate new feasible solutions. In general, the most popular types of clustering representation solutions for EMOC are (Hruschka et al., 2009):

(a) **Label-based** representation, which takes into account labels for each object in the partition. The length of an encoding of the solution is equal to the number of objects in the dataset, and each position denotes the cluster label of the respective object.

(b) **Prototype-based** encoding is usually applied in centered-based clustering, in which cluster prototypes, such as centroids, medoids, or modes, are used in partition representation. In the centroid-based encoding, the chromosomes are denoted by the coordinates of the cluster centers. In medoid-based encoding, the chromosomes are represented by the coordinates that define the smallest average dissimilarity of the cluster to all other objects. In mode-based encoding, the chromosome can denote the frequency of the attribute. In general, in the prototype-based representation, one can have k chosen centers, in which the objects at each point are associated with the closest chosen center measure.

(c) **Locus-based adjacency graph** (LAG) representation corresponds to a graph containing a vertex for each data point, and the links between two data points represent the edges. The linked objects represent the clusters in the solution.

In particular, some approaches use a binary representation to define the labels or prototypes instead of using numerical values. In Sert et al. (2011, 2012), each chromosome includes $n \cdot k$ bits, and each reserved k bits provides the cluster number of the corresponding instance. In Ripon and Siddique (2009), each data point is a candidate center, and a binary encoding is applied to define whether a data point is a center or not. Besides that, it is possible to consider other aspects of the clustering problem in the representation. For example, in Di Nuovo et al. (2007), Fuzzy C-Means (FCM) parameters and feature weights are applied to represent the solution. In Zhu et al. (2012); Xia et al. (2013); Z. Zhou (2018), the authors used the center information associated with a center weight to encode the solutions. In Dong et al. (2018) and Zhu and Xu (2018), the fuzzy membership matrix and the center information are designed to represent each solution. In Luo et al. (2015), the authors consider an input as a linear combination of base elements (e.g., parameters or coefficients), which are chosen from an over-complete dictionary to design the sparse-based representation.

Regarding the initialization, a common practice in EMOC approaches is to use random generators to assign labels or choose the initial centers of the clusters in the partition. The random initialization generally provides unfavorable partitions since the clusters are likely to be mixed up to a high degree. However, this strategy is very popular because of its simplicity and effectiveness in testing the algorithms against hard evaluation scenarios (Hruschka et al., 2009).

In contrast, some relevant EMOC algorithms use high-quality individuals in the initial population, in which clustering algorithms are applied to generate the base partitions. For example, KM, AL, SL, CL, MST-clustering, SNN, Spectral Clustering (SPC) (Shi and Malik, 2000) are applied in the initialization of some EMOC approaches presented in Chapter 3.

In the literature, most prototype-based encoding approaches use random generators in the initialization. On the other hand, the label-based encoding takes advantage of not requiring decoding of the solutions, making it possible to apply most of the traditional clustering algorithms in the initialization. The LAG representation can rely on a graph-based method in the initialization, such as MST-clustering, taking advantage of its data structure.

2.4.2 Optimization Module: Multi-objective Evolutionary Optimization

In general, the EMOC algorithms rely on general-purpose MOEAs in the optimization module. The choice of the multi-objective approach should consider the number of objective functions and the characteristics of the application, in which it is possible to explore some aspects, such as user preference, diversity of solutions, among other features.

The most traditional category of multi-objective algorithms is **Pareto-based**, where the solutions are evaluated and compared by considering the Pareto dominance. For example, the NPGA - Niched Pareto Genetic Algorithm (Horn et al., 1994) is designed along with the natural analogy of the evolution of distinct species exploiting different niches or resources in the environment, in which the main strategy relies on tournament selection among a population's individuals and Pareto dominance. The PESA-II - Pareto Envelop-based Selection Algorithm version 2 (Corne et al., 2000) is an elitist method (the selection considers the best one or more solutions, called the elites, in each generation, which are inserted into the next), where the diversity mechanism is cell-based density. The NSGA-II - Non-dominated Sorting Genetic Algorithm version II (Deb et al., 2000) is an elitism method that employs a ranking based on non-domination sorting associated with crowding distance. The SPEA-2 - Strength Pareto

Evolutionary Algorithm version 2 (Zitzler et al., 2001) is also an elitism method that applies the concept of the strength of dominators as a fitness assignment, employing a density based on the kth nearest neighbor to preserve the diversity.

Beyond that, Li et al. (2015) defined other categories by considering other aspects beyond the Pareto front to evaluate and compare the solutions in MOEAs/MaOEAs:

(a) **Relaxed dominance-based** algorithms use a variant of dominance, such as value-based (that changes the objective values by modifying the Pareto dominance of the solutions when comparing them) or number-based dominance (that compares a solution to another by counting the number of objectives where it is better than, the same as, or worse than the other);

(b) **Diversity-based** algorithms apply a customized diversity-based approach, for example, the SDE (Shift-Based Density Estimation), where the diversity is taken as the first criterion instead of the convergence; it is possible because SDE shifts the positions of the solutions to measure the density of the neighborhood of the solution, allowing both the distribution and the convergence information to be used in the comparison of the solutions;

(c) **Aggregation-based** algorithms apply aggregation functions to evaluate the solutions, which can be divided into two categories: aggregation of objective values and aggregation of objective ranks.

(d) **Indicator-based** algorithms aim to maximize the value of a specific indicator, which can be divided into three classes: hypervolume driven, distance-based indicator driven, and R2 indicator driven;

(e) **Preference set-based** algorithms consider the user's preferences in the optimization process. This kind of algorithms can be divided into three classes based on the timing of the set of preferences being used: a priori (selection before the search), interactive (selection during the search), and a posteriori (selection after the search);

(f) **Reference-based** algorithms consider a set of reference solutions, which are applied to measure the quality of the solutions and guide the search during the evolutionary optimization process, such as in NSGA-III (Deb and Jain, 2014) and RVEA (Cheng et al., 2016);

(g) **Dimensionality reduction** algorithm seeks to simplify the problem by reducing its complexity, where the number of objectives can be reduced gradually during the search process (online) or the dimensionality reduction is carried out after obtaining a set of Pareto-optimal solutions (offline).

Additionally, it is possible to consider another category, a **Hybrid-based**, that combines two or more approaches to overcome their particular problems, for example, the MOEA/DD - Multi-Objective Evolutionary Algorithm based on Dominance and Decomposition approaches (Li et al., 2015) combines two categories of strategies: Pareto dominance and aggregation.

As mentioned above, in general, MOEAs are applied to clustering problems, considering specific objective functions (clustering criteria), and different combinations of crossover and mutation operators. Thus, we detail them in the following sub-sections.

2.4.2.1 Objective Functions

In general, CVIs (see Section 2.3) that consider internal and relative criteria are used as clustering objective functions. On the other hand, specific objective functions designed for multi-objective clustering, such as the sparsity (SP) and reconstruction error (RE) designed for spectral clustering, can be used in EMOC approaches (Luo et al., 2015).

In the following, we introduce objective functions categorized by criteria (cluster properties). These objective functions denote the clustering criteria adopted in the approaches presented in Chapter 3:

(a) **Compactness criteria**: average within group sum of squares ($AWGSS$) (Kirkland et al., 2011), overall deviation (Dev) (Handl and Knowles, 2005a), K-Mode internal distance (Km_{id}) (Sert et al., 2011), K-Mode weighted internal distance (Km_{wid}) (Sert et al., 2011), intra-cluster entropy (Ent) (Ripon et al., 2006a), homogeneity (H) (Dutta et al., 2012a), intra-cluster variance (Var) (Garza-Fabre et al., 2018), and total within-cluster variance ($TWCV$) (Du et al., 2005), and fuzzy compactness (J_m) (Bezdek, 2013), are criteria based on intra-cluster similiarity.

(b) **Connectedness criteria**: connectivity index (Con) (Handl and Knowles, 2005a), and data continuity degree (DCD) (Menéndez et al., 2013), are criteria based on neighborhood relationship.

(c) **Separation criteria**: average between-group sum of squares ($ABGSS$) (Kirkland et al., 2011), inter-cluster average separation (Sep_{AL}) (Ripon and Siddique, 2009), K-Mode external distance (Km_{ed}) (Sert et al., 2011), K-Mode weighted external distance (Km_{wed}) (Sert et al., 2011), separation index (Sep_{CL}) (Dutta et al., 2012b), and graph-based separation (Sep_{graph}) (Menéndez et al., 2013), fuzzy separation (Sep_{fuzzy}) (Mukhopadhyay et al., 2007), and fuzzy overlap separation (Sep_{nfuzzy}) (Wikaisuksakul, 2014), are criteria based on inter-cluster similarity.

(d) **Separation and Compactness criteria**: categorical data clustering with subjective factors ($CDCS$) (Zhu and Xu, 2018), Calinski-Harabasz (CH) (Zhu and Xu, 2018), Davies-Bouldin (DB) (Zhu and Xu, 2018), Dunn (Dutta et al., 2019), modularity (Mod) (Liu et al., 2018), silhouette (Sil) (Mukhopadhyay and Maulik, 2007), $\mathcal{I}$ (Dong et al., 2018), addition feature weight (J_{Add}) (Xia et al., 2013), Pakhira, Bandyopadhyay and Maulik (PBM) (Pakhira et al., 2004), Xeni-Beni (XB) (Di Nuovo et al., 2007), soft subspace Xie-Beni ($SSXB$) (Zhu et al., 2012), are criteria that take into account both intra-cluster and inter-cluster similarity.

(e) **Other criteria**: cluster cardinality index (CCI) (Zhu and Xu, 2018) and expected weighted coverage density ($EWCD$) (Sert et al., 2011) consider the relation of the occurrence of the objects in a categorical dataset. The similarity index (Sim) (Li et al., 2017) is the only relative CVI that compares partitions used as the objective function, while the other CVIs consider the data properties of each partition.

It is a common practice in the literature to apply two or more different categories of clustering criteria as objective functions, where the approach will be able to optimize multiple characteristics of the evolved clusters. For example, a popular pair of objective functions, (Var, Con), consider the compactness and connectedness criteria. In Chapter 3 other combinations of objective functions are presented. Due to the large number of clustering criteria and considering that some objective functions may have different names in the literature, we present a detailed description of each of these objective functions in Appendix A.

2.4.2.2 Crossover and Mutation Operators

Evolutionary optimization relies on crossover and mutation operators to generate new solutions. In the literature, we can find approaches using traditional evolutionary operators and clustering designed operators. The most popular traditional operators used in EMOC approaches are:

(a) **One-Point crossover**: one crossover point is considered along the length of the parents' chromosomes, and the genes following the crossover point in one parent are swapped with the genes in the other parent (Hruschka et al., 2009).

(b) **Two-Point crossover**: two crossover points along the length of the chromosome of each parent, such that the interval of genes between these two points are swapped (Hruschka et al., 2009).

(c) **Shuffle crossover**: this operator is similar to one-point crossover, in which a single crossover position is selected, and before the variables are exchanged, they are randomly shuffled in both parents (Sert et al., 2011).

(d) **Uniform crossover**: for each position on the chromosome, a random decision is made on whether the swapping of genes should be done or not (Handl and Knowles, 2007).

(e) **Simulated binary crossover (SBX)**: this operator uses a probability density function that simulates the One-Point Crossover in binary-coded representation (Wikaisuksakul, 2014).

(f) **Polynomial mutation**: a polynomial probability distribution is applied to perturb a solution (Ripon et al., 2006b).

(g) **Uniform mutation**: this operator replaces the value of the chosen particular slot position with a uniform random value selected considering a specified upper and lower bounds for that position (Dong et al., 2018).

In terms of the clustering-designed operators, the representation and clustering criteria are taken into consideration. For example, the perturbation or replacement of center, centroid, or medoid is applied in the algorithms that use a prototype-based encoding to shift a randomly selected center slightly from its current position or replace the position of the cluster prototype according to a criterion; the exchange of the prototypes considers two parents in which there is an exchange of centroids to generate a new solution. Also, there are operators designed to split the objects of a cluster or merge two or more clusters to generate new solutions. Handl and Knowles (2005a,b); Handl and Knowles (2007) presented the neighborhood-based mutation that is applied to the graph-based representation, replacing an existing link in the graph with another link to one of the randomly selected nearest neighbors. In Bousselmi et al. (2017) and Bechikh et al. (2019), Cheng and Church's (CC) algorithm was adapted to be applied as a mutation operator. The CC algorithm considers three steps (multiple node deletion, single node deletion, and node addition) to iteratively perform the removal and addition of rows and columns in a data expression matrix. As a mutation operator, only row operations are performed to preserve specific data properties. Besides that, Faceli et al. (2006) introduced the use of clustering ensembles as a crossover operator. A clustering ensemble is a technique applied to combining multiple different clustering results (generated by different clustering algorithms or the same algorithm with different iterations) into a single partition (Boongoen and Iam-On, 2018). As a crossover operator, pairs of partitions are combined with a consensus function to generate new individuals.

2.4.3 Selection Module: Partitions Selection

The Selection module is applied to restrict the number of clustering solutions presented to the decision-maker or data specialist. In the literature, most EMOC approaches select the final set of solutions by applying CVIs (see Section 2.3). For example, in Tsai et al. (2012), PBM, and DB were used to single out the optimal solution. In Menéndez et al. (2013, 2014), the solution with the highest value of the Sep_{graph} in the Pareto front was considered the best solution to be selected. In Xia et al. (2013), a new indicator called the projection similarity validity index (PSVIndex) was designed to select the best solution and cluster number. In Dutta et al. (2019), the EMOC approach uses an overall rank of nine CVIs to determine the final set of solutions: C index (Baker and Hubert, 1976), COSEC - Compactness and Separation Measure of Clusters (Rahman and Islam, 2014), DB, $Dunn$, Dev, Ent, XB, Purity (Schütze et al., 2008) and F-Measure (Larsen and Aone, 1999). In particular, in Luo et al. (2015), the non-dominated solutions are used to construct a standard adjacency matrix, and the measurement Ratio Cut (Wei and Cheng, 1991) provides a way to select a final trade-off solution.

Another way to select final solutions is by applying the knee-based approaches that are usually applied in determining the number of clusters in a data set. For example, the knee method presented by Handl and Knowles (2005a,b); Handl and Knowles (2007) compares the final set of solutions and a control front. The solution corresponding to the largest distance between the actual non-dominated front and the control fronts is chosen to be the final solution, corresponding to the "knee" (the point of inflection) of the non-dominated front. In Wang et al. (2018) and Du et al. (2005), the best clustering result is defined by the "elbow" method, which consists of picking the "elbow" or "knee" of the curve in the non-dominated front.

Besides that, clustering ensemble methods are used to select the final solutions. The non-dominated solutions are used as base partitions to generate the consensual partition by applying a consensual function to combine the base partitions.

2.4.4 Evaluation of the EMOC algorithms

In terms of evaluating clustering results, most EMOC approaches consider an external validity index, such as the adjusted Rand index (ARI) (Rand, 1971), to evaluate the set of final solutions. ARI is a corrected-for-chance version of the Rand index (Hubert and Arabie, 1985), computes the probability of two objects of two partitions belong to the same cluster or different clusters, as defined in Equation (2.3), where n_{ij} is the number of common objects between the clusters $\mathbf{c}_i$ in π_a and $\mathbf{c}_j$ in π_b, n_i is the number of objects in the cluster $\mathbf{c}_i$ in p_a, e n_j is the number of objects in the cluster $\mathbf{c}_j$ in π_b, k_a and k_b are the number of clusters in the partitions π_a and π_b.

$$ARI = \frac{\sum_{i=1}^{k_a}\sum_{j=1}^{k_b}\binom{n_{ij}}{2} - \left[\sum_{i=1}^{k_a}\binom{n_i}{2}\cdot\sum_{i=1}^{k_b}\binom{n_j}{2}\right]/\binom{n}{2}}{\frac{1}{2}\cdot\left[\sum_{i=1}^{k_a}\binom{n_i}{2} + \sum_{i=1}^{k_b}\binom{n_j}{2}\right] - \left[\sum_{i=1}^{k_a}\binom{n_i}{2}\cdot\sum_{i=1}^{k_b}\binom{n_j}{2}\right]/\binom{n}{2}} \tag{2.3}$$

Besides that, the analysis of internal criteria can also be applied to investigate specific data structures. For example, in Ripon et al. (2006a,b), H, Sep_{AL}, $Dunn$, and Dev are evaluated to analyze the general behavior of the EMOC approaches regarding each criterion. In Dutta et al. (2012b,c), the authors compare their approaches with other ones based on the DB, H, and Sep_{AL}.

2.5 CHAPTER REMARKS

In this chapter, we presented the general concepts applied in our study by describing the concepts and properties associated with clustering and multi-objective optimization. In particular, we present an abstraction of the main components of multi-objective clustering algorithms, introducing a general architecture of EMOC. We dealt with all the components of this architecture to support the implementation of EMOC algorithms. Furthermore, we use this architecture to support the identification of the main components and features of the existing EMOC studies, presented in the next chapter.

3 LITERATURE REVIEW

In this chapter, we present a general view of multi-objective clustering, considering an extensive mapping of the literature to identify the main methods and concepts that have been adopted to design the EMOC approaches.

3.1 OVERVIEW OF MULTI-OBJECTIVE CLUSTERING STUDIES

This review considers papers related to multi-objective clustering (MOC) from IEEE Xplore[1], ACM Digital Library[2] and Scopus[3]. These article repositories contain the most important journal papers and conference proceedings, in the computer science and engineering domains. We used the terms "multi-objective", "multiobjective", and "many-objective" as keywords related to optimization with multiple objectives, along with the term "clustering" to search by title for articles about multi-objective clustering. The article mapping considered English-language papers that were published before the year 2021. The search result is 231 papers from IEEE Xplore, 30 papers from the ACM Digital Library, and 533 papers from Scopus, totaling 794 papers. Then, duplicated papers were removed. After that, we analyzed the main contents of the resulting set of documents, resulting in 358 papers.

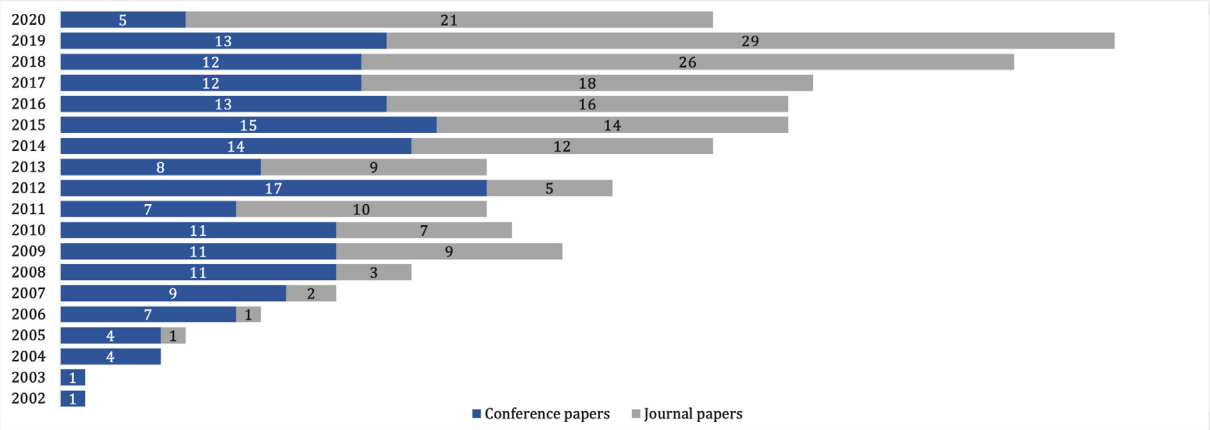


Figure 3.1: The number of publications related to MOC from 2002 to 2020.

Fig. 3.1 shows the number of publications related to MOC that appeared in both journals and conferences over the years. It provides information on how the MOC field is evolving, based on the number of papers published. The first indexed article found was published in 2002 (Zwir et al., 2002), a conference paper in the Annals of the New York Academy of Sciences. In the same way, most of the articles published between 2002 and 2008 were published at conferences. In 2009, we observed a substantial increase in journal papers. Between 2008 and 2016, we verified a certain equilibrium in the number of articles published in conferences and journals, except in 2012, when the number of conference papers increased abnormally, without a specific explanation. Finally, in the last four years, the number of articles published in journals has substantially increased. In particular, in 2019, the number of publications in journals was

[1] https://ieeexplore.ieee.org
[2] https://dl.acm.org/
[3] https://www.scopus.com

almost three times greater than the number of papers presented at conferences. In 2020, we can notice that the total number of papers significantly decreased compared to 2018 and 2019. One reasonable motivation is the indexing time before the papers appear in the search, considering that the mapping was performed in the first trimester of 2021. Another reasonable motivation was the COVID pandemic, which motivated periods of suspension of non-essential activities and caused some conferences worldwide to be canceled or postponed.

Regarding the optimization approach, considering the general classification of the metaheuristics presented by Siarry (2016), we observed that most studies applied evolutionary optimization. Fig. 3.2 presents the relationship between the number of articles and the evolutionary optimization articles, including memetic and hybrid approaches that include other methods associated with the evolutionary approach. In the early years, almost all MOC papers relied on the evolutionary approach. In the middle years, the use of other optimization methods was observed, such as Artificial Immune system-inspired (Timmis et al., 2008), Differential Evolution-based (Eltaeib and Mahmood, 2018), Simulated Annealing-based (Bertsimas and Tsitsiklis, 1993), and Particle swarm-based (Rana et al., 2011). In the mapped articles, the first occurrence of these approaches was between 2007 and 2009. In recent years, the use of a variety of other optimization methods has also been verified, such as other nature-inspired algorithms (Siarry, 2016), among others.

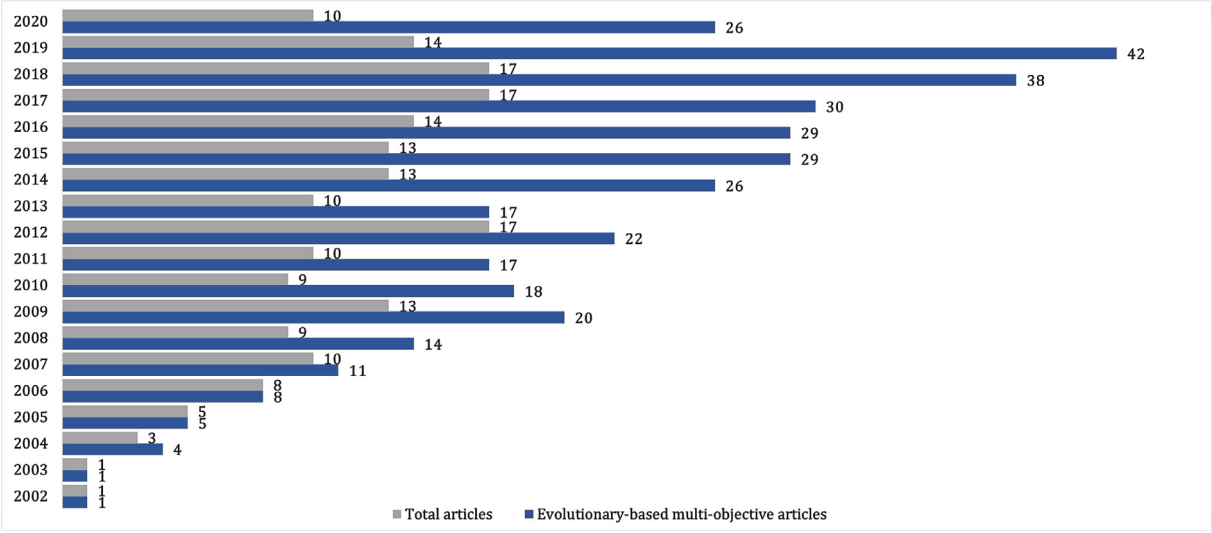


Figure 3.2: Total articles vs. evolutionary-based optimization articles.

In the following, we list the most relevant works. They were selected by considering two general indices: h-index and Scopus-percentile. We filtered the articles by h-index greater than 10 to filter the conference papers and by Scopus-percentile greater than 50% to obtain the list of the most relevant journal papers. These values were selected to cover the A-rank papers in the CORE - Computing Research and Education Association of Australasia and Qualis (a Brazilian official system to classify scientific production). These algorithms were grouped based on some shared characteristics that highlight the main features or applications of these approaches. The general concepts and methods applied in these EMOC approaches were introduced in Section 2.4.

3.2 GENERAL-PURPOSE EMOC ALGORITHMS

First, we present general-purpose EMOC approaches divided in: MOCK-based works, EMOC for categorical data, EMOC for bi-clustering, EMOC for subspace clustering, ensemble-based

EMOC, fuzzy clustering-based EMOC, spectral clustering-based EMOC, multiple distance measures-based EMOC, multi-k-clustering-based EMOC, EMOC with specific MOEA, and other EMOC approaches.

3.2.1 MOCK-based works

One of the most popular algorithms is MOCK - Multi-Objective Clustering with automatic k-determination (Handl and Knowles, 2005a,b; Handl and Knowles, 2007). The MOCK algorithm uses LAG representation, initialization with MST-clustering and KM, and two objective functions: Dev and Con. The PESA-II was the MOEA used in this approach. The adjacency graph representation promoted the use of specific operators for the clustering problem, such as the neighborhood-based mutation operator, which manipulates the links over the MST, in which each vertex can only be linked to one of its nearest neighbors. After the optimization process and the generation of final clustering solutions, MOCK uses an automatic k-determination scheme to choose the best clustering solution from a set of solutions with a knee-based strategy.

Other studies were derived from the analysis of MOCK, as follows. Matake et al. (2007) provided an approach, MOCK-Scalable, to improve the final selection of solutions in large-scale data based on a scaling filter to reduce the solutions in the Pareto front. Tsai et al. (2012) proposed the MIE-MOCK - Multiple Information Exchange Multi-Objective Clustering with automatic k-determination. The MIE-MOCK algorithm uses a pool of crossover and mutation operators selected by a random method and also provides a new final selection of solutions based on two CVIs: PMB and DB. In Handl and Knowles (2012), the authors analyzed four pairs of objective functions for multi-objective clustering, including an analysis of the original objective functions of MOCK. Also, Handl and Knowles (2013) presented an analysis of the use of evidence accumulation to support the post-processing of the clustering solutions returned by the MOCK. In Garza-Fabre et al. (2017, 2018), the authors proposed the Δ-MOCK, providing a new encoding to improve the MOCK scalability and other specific modifications to improve the convergence of the solutions. Zhu et al. (2018) provided the Δ-EMaOC - Evolutionary Many-Objective Optimization Clustering, improving the general architecture of the Δ-MOCK to optimize five objective functions. The Δ-EMaOC algorithm considers the use of MaOEAs (SPEA-II-SDE (Li et al., 2014), NSGA-III (Yuan et al., 2016), MOEA/DD (Li et al., 2015) and RVEA (Cheng et al., 2016)) instead of MOEA (NSGA-II). In general, these approaches are applied to detect clusters in heterogeneous structured data, considering a continuous data type and crisp clustering. Zhu et al. (2020) proposed the MOAC-L - locus-based multi-objective automatic clustering. The MOAC-L algorithm applies CVIs and ensemble-clustering to improve the encoding and the selection of solutions in the optimization process.

3.2.2 EMOC for Categorical Data

In particular, some EMOC approaches were designed for categorical data clustering, where the data objects are defined over categorical attributes (instead of using the continuous data type that is applied in most of the other approaches). For example, Handl and Knowles (2005c) presented the MOCK-medoid, a MOCK extension for multi-objective clustering around medoids for categorical data. Mukhopadhyay and Maulik (2007) also introduced a medoid-based EMOC, the MOGA-medoid, to deal with categorical data. The MOGA-medoid algorithm uses the NSGA-II to optimize the Sil and Dev (computed in terms of the medoid instead of the centroids), applying the one-point crossover and a medoid-based replacement mutation designed to consider a center-based solution encoding.

Dutta et al. (2012b) provided a specific MOEA, the Hybrid MOGA, to optimize H and Sep_{AL}. The main contribution of this work relies on the use of this new MOEA with the Pairwise Crossover (Fränti et al., 1997), the replacement (substitution) mutation, and the local searching power of K-modes (or KM) to deal with continuous and categorical features in the dataset.

Mukhopadhyay et al. (2007) presented a multi-objective genetic fuzzy clustering of categorical attributes (MOGA-fuzzy), considering a uniform crossover and a center-based replacement mutation in NSGA-II to optimize the global compactness (a normalized J_m index for categorical data (Tsekouras et al., 2004)) and Sep_{fuzzy}. They applied a specific selection method to obtain the final solution, in which the points assigned to the same cluster by at least 50% of the clustering solutions are taken as the training set, and the remaining points are assigned a class label using k-nearest neighbor (k-nn) classification in order to select a single solution from the set of the non-dominated solutions. In Mukhopadhyay et al. (2009), the authors provide a new version of the MOGA-fuzzy, MOGA-fuzzy2, considering modifications in the evolutionary operators, in which the One-Point Crossover and Mode replacement were applied.

Zhu and Xu (2018) introduced the MaOFcentroids, a many-objective fuzzy centroid clustering algorithm for categorical data. MaOFcentroids algorithm uses fuzzy membership matrix encoding (a matrix with the degree of membership of each object), and adapted operators that consider the number of the clusters and the membership of the solutions in the NSGA-III. It simultaneously optimizes five CVIs ($CDCS$, DB, CH, CCI, and XB). In terms of the selection, this approach uses a specific clustering ensemble for categorical data, the SIVID - Sum of Internal Validity Indices with Diversity (Zhao et al., 2017).

The most recent work of Dutta et al. (2019) introduces the MOGA-KP, an approach with automatic k-determination applied to deal with different types of features (continuous, categorical, and missing feature values). It considers some common aspects of the previous works (Dutta et al., 2012b,c), while improving some aspects, such as the use of other evolutionary operators, and the local search operators. Besides that, the MOGA-KP algorithm uses a ranking of nine CVIs to determine the final set of solutions.

3.2.3 EMOC for Bi-Clustering

One specific line of study in EMOC is Bi-clustering, which consists of simultaneous partitioning of the set of samples and the set of their attributes into subsets (classes). The goal of this kind of algorithm is to find one or all (possibly overlapping) sub-matrices of a given matrix, each of which shares a pre-defined property over the elements across all its columns (or rows). Each such sub-matrix is called a bi-cluster. Bousselmi et al. (2017) presented the BI-MOCK, which extends MOCK to the case of bi-clustering by adding a subset of columns (conditions) to each chromosome in the representation. BI-MOCK algorithm uses the two-points crossover adapted for variable-size chromosomes and the CC algorithm as a mutation operator in the PESA-II to optimize Var, and the size of the bi-cluster. Bechikh et al. (2019) presented the MOBICK - Multi-Objective BI-Clustering with automated k deduction, that extends Bousselmi et al. (2017) study. MOBICK algorithm uses the Δ-MOCK reduced encoding, the uniform crossover adapted for bi-clustering conditions, and the CC algorithm as the mutation operator in the PESA-II to also optimize Var and the size of the bi-cluster.

3.2.4 EMOC for Subspace Clustering

Another line of studies considers Subspace Clustering, an extension of traditional clustering that seeks to find clusters in different subspaces within a dataset. Zhu et al. (2012) introduced the MOSSC - Multi-Objective evolutionary algorithm-based Soft Subspace Clustering, which

optimizes the $SSBX$ and J_{wm} in the NSGA-II. This approach uses a center-based encoding with weights to avoid trapping in local minima, aiming to obtain more stable clustering results. Xia et al. (2013) presented the MOEASSC - Multi-Objective Evolutionary Approach-based Soft Subspace Clustering, which also uses a mixed encoding (center and weight-based). MOEASSC differs from the MOSSC in terms of the pair of objectives (J_{wm} and J_{Add}), and the use of a local search operator based on the KM. Z. Zhou (2018) introduced the MOKCW - Multi-Objective Kernel Clustering algorithm with automatic attribute Weighting. In general, MOKCW extends MOSSC and MOEASSC by considering kernel clustering. For example, MOKCW used the MOSSC objective functions adapted to consider kernel distance. The authors also improved the final selection method of the MOEASSC by applying a clustering ensemble method (MCLA - Meta Clustering Algorithm (Strehl, 2002) and HBGF - Hybrid Bipartite Graph Formulation (Fern and Brodley, 2004)) associated with the PSVIndex.

3.2.5 Ensemble-based EMOC

Another specific approach was proposed by Faceli et al. (2006), the MOCLE - Multi-Objective Clustering Ensemble. The main idea behind this approach is the use of clustering ensemble methods as crossover operators to combine partitions and extract agreed patterns to generate new solutions in the evolutionary optimization process. MOCLE is a framework that uses a label-based representation; the initial population is generated with various clustering methods to detect different cluster formats, such as SL, AL, KM, and SNN. The original implementation of the MOCLE (Faceli et al., 2006) provides two MOEAs: NSGA-II and SPEA-II, to optimize the Dev and Con; and two crossover operators: MCLA and HBGF; however, it does not use any mutation operator.

This general concept of using clustering ensemble methods as crossover operators has been used in other studies as well. Faceli et al. (2009) introduced the MOCLE in the context of gene expression datasets, applying an additional objective, $ConP$ (the connectivity index based on the Pearson Correlation), and a new set of clustering methods to generate the initial population (AL, CoL, KM, and SPC). Liu et al. (2012) introduced the IMOCLE - Improvement of the Multi-Objective Clustering Ensemble algorithm, in which a relative CVI, Sim, was added along with the three objective functions defined by Faceli et al. (2009) to improve the clustering. In general, these approaches are also applied to detect clusters in heterogeneous structured data, considering both continuous data type and crisp clustering.

3.2.6 Fuzzy Clustering-based EMOC

Another line of studies considers the integration of the general concepts of the existing fuzzy clustering algorithms, such as FCM and FRC - Fuzzy Relational Clustering, with a multi-objective evolutionary approach (NSGA-II). Di Nuovo et al. (2007), Wikaisuksakul (2014) and Dong et al. (2018) presented fuzzy approaches integrating the NSGA-II with the FCM (Bezdek, 2013). Di Nuovo et al. (2007) introduced the NSGA-II & FCM that optimizes the number of features and the XB index to discover the best number of groups while pruning the features to reduce the dimensionality of the dataset. NSGA-II & FCM algorithm uses a specific solution encoding that considers the FCM parameters (number of the clusters k and FCM fuzzyfier m) and the feature weights. Wikaisuksakul (2014) introduced the FCM-NSGA, which optimizes the J_m and Sep_{nfuzzy} in NSGA-II, considering SBX and polynomial mutation operators. Dong et al. (2018) introduced the ADNSGA2-FCM that optimizes the DB and $\mathcal{I}$ indexes. ADNSGA2-FCM uses a center-based and fuzzy membership matrix (a matrix with the degree of membership of each object) as an encoding. In terms of the evolutionary operators, it considers the uniform

mutation with two new crossover operators, the Nearest Neighbor Matching Crossover Operation (an exchange of centers in the nearest neighbor to produce solutions with the same number of clusters) and the Truncation and Stitching Crossover Operation (an exchange of a set of center positions is performed to produce solutions with a different number of clusters). Moreover, they introduced an adaptive mechanism that is applied to compute the crossover and mutation probabilities that are changed according to the fitness of the current population. On the other hand, Paul and Shill (2018) propose the FRC-NSGA/IFRC-NSGA, hybrid methods that combine the FRC algorithm (Skabar and Abdalgader, 2013) and the NSGA-II to optimize the J_m and Sep_{nfuzzy}.

3.2.7 Spectral Clustering-based EMOC

Some works use spectral clustering as a foundation for designing EMOC approaches. MOGGC - Multi-Objective Genetic Graph-based Clustering Algorithm (Menéndez et al., 2013) considers optimizing the computation of graph similarity features in SPC to achieve lower memory consumption and increase the clustering quality. For that, this approach provided a new objective function pair, the separation of clusters (Sep_{Graph}) and a graph continuity metric (DCD). MOGGC was extended by the CEMOG - CoEvolutionary Multi-Objective Genetic Graph-based Clustering (Menéndez et al., 2014), a partitional k-adaptive spectral clustering algorithm that uses a strategy based on island-model and a graph topology to migrate individuals from sub-populations. This last approach does not require input of the initial number of clusters required in the MOGGC. In this context, Luo et al. (2015) introduced the framework SRMOSC, which uses sparse representation for sparse spectral clustering. SRMOSC uses SP and RE as objective functions to be optimized in the NSGA-II (or MOEA/DD) with a specific pair of operators that consider the sparsity properties.

3.2.8 Multiple Distance Measures-based EMOC

Other approaches consider the use of different distance functions in the objective functions. Liu et al. (2018) introduced the MOECDM - Multi-objective Evolutionary Clustering Based on Combining Multiple Distance Measures and the MOEACDM - Multi-objective Evolutionary Automatic Clustering based on Combining Multiple Distance Measures. Both these approaches consider a single CVI computed with distinct distance functions to define objective functions to be optimized. They use a label-based encoding and an NCUT pre-clustering (Shi and Malik, 2000) in the initialization, but in the MOECDM, a portion of the individuals are generated by a random generator. They also adapted the crossover and mutation operators, in which the probabilities are adjusted along with the generations. MOECDM was designed to detect the desirable cluster number automatically, using Sep_{CL} index computed with Euclidean distance (Sep_{CL1}) and Path distance (Sep_{CL2}) as objective functions. MOEACDM was designed to detect compact clusters, using Mod also computed with Euclidean distance (Mod_1) and Path distance (Mod_2) as objective functions.

3.2.9 Multi-k-clustering-based EMOC

Other approaches consider multi-k-clustering with the a posteriori method, where k is taken as an objective function, differing from the automatic data clustering methods, such as MOCK, that consider k an inner aspect of the decision variable, obtained by the optimization of clustering criteria. For that, Du et al. (2005) introduced a specific solution representation, the linked-list based encoding. The authors used the fellowship between the objects instead of the label-based

relationship to define the clusters, in which each cluster has all its elements linked, similar to the relationship of the nodes presented by Handl et al. (2007). This representation was applied in the MOGA-LL (Du et al., 2005), an EMOC approach that optimizes the $TWCV$ and k as objective functions in the NPGA, considering two particular operators: (i) an adapted one-point crossover, which allows different clusters to exchange partial contents and may split a cluster into two; (ii) link-replacement mutation, in which a sub-group of objects is associated with another cluster instead of just a different node.

Wang et al. (2018) proposed the EMO-KC (Evolutionary Multi-objective k-clustering) to demonstrate the importance of the conflict between the objective functions to obtain a diverse set of final solutions with a different number of clusters. They showed evidence that sum of squared distances (SSD) and k are not always conflicting between two individuals and introduced a transformation of SSD. SSD can be denoted by $(Var \cdot n)$, and the adapted SSD (Var'), considers the following transformation: $(1 - exp^{-1 \cdot SSD}) - k$. In Wang et al. (2020), this same pair of objective functions was explored in a new MOEA that considers a constrained decomposition with grids (CCDG-K). Both EMO-KC and CCDG-K define the best clustering result (the optimal k) by the "elbow" method (Hancer and Karaboga, 2017).

3.2.10 Specific MOEA for EMOC

As previously presented, Dutta et al. (2012b,c) provided a specific MOEA, the Hybrid MOGA designed for categorical data. Besides that, another particular approach is the VRJGGA - Variable-length Real Jumping Genes Genetic Algorithm introduced by Ripon et al. (2006a). The VRJGGA is an EMOC algorithm that extends the Jumping Genes Genetic Algorithm (JGGA) (Man et al., 2004) and applies the survival selection of the NSGA-II. The JGGA considers jumping gene operations before evolutionary operators to improve the diversity of solutions. VRJGGA uses a centroid-based encoding associated with the modulo crossover (Srikanth et al., 1995) (an adapted one-point crossover, where each child is a set of completely specified sub-solutions) and the polynomial mutation, to optimize the Ent and Sep_{AL}. In Ripon et al. (2006b), the authors provided new features to VRJGGA, introducing two local search methods, probabilistic cluster merging, and splitting for clustering improvement. Ripon and Siddique (2009) also applied the extended version of the JGGA to EMOC, introducing the EMCOC - Evolutionary Multi-objective Clustering for detecting overlapping clusters. EMCOC introduces a new chromosome representation and cluster-assignment method in which each data point is a candidate center and a binary encoding is applied to define whether a data point is a center or not.

3.2.11 Other MOC approaches

Some papers consider other objective functions and provide other features in the design of the EMOC approaches. For example, Kirkland et al. (2011) presented the Multi-Objective Clustering algorithm (MOCA), that optimizes three objective functions, $AWGSS$, $ABGSS$, and Con in the NSGA-II. Sert et al. (2011, 2012) presented the MOC-HCM, which uses five objective functions: Km_{id}, Km_{ed}, Km_{wid}), Km_{wed}, and EWCD. The MOC-HCM algorithm uses a binary representation, a local search operator (k-mode-based operator) that reassigns the instances to the closest clusters in terms of their frequencies, and a new final selection method based on a new metric, the H-Confidence Metric (HCM).

Besides the above-mentioned works, we also found specific approaches, in which their main features consider some particular methods, as follows. Özyer and Alhajj (2009) applied the divide and conquer approach in an iterative way to handle the clustering process and improve the performance of the evolutionary algorithm. Zheng et al. (2012) extended algebraic

operations of gene expression to propose a multi-objective gene expression programming for clustering. Garcia-Piquer et al. (2017), focused on reducing the impact of the volume of data in the EA by means of the stratification of the complete data set into disjoint strata and alternating them in each cycle of the genetic algorithm. Liu et al. (2019) improved the performance of multi-objective soft subspace clustering algorithms for clustering high-dimensional data by using a transfer learning-assisted multi-objective evolutionary clustering framework with MOEA/D.

3.2.12 Summary of the EMOC approaches

Here, we summarize the components of the main presented EMOC algorithm. We considered the publishing chronology to list each EMOC to make it possible to observe the variations of components over time.

In Table 3.1, we present the main features (components) applied in the initialization and optimization of each approach. In this table, we used acronyms and abbreviations for some words: Ad. for Adapted, Repl. for Replacement, and Mod. for Modified, NA for not assigned, and FM for Fuzzy membership-based.

It is possible to note that there are a variety of representations being applied in the EMOC approaches. In particular, from the year 2017, the use of representations concerning the reduction of the size of the chromosome has emerged. In contrast, most EMOC approaches use a random strategy in the initialization, without introducing a relevant novelty in recent years.

Regarding the optimization phase, the NGSA-II has been the most applied MOEA over the years. In particular, from the year 2018, the use of MaOEAs considering the optimization of more than 3 objective functions has emerged. In terms of the objective functions, over the years, new combinations of clustering criteria have been applied. A common practice considers at least one compactness-based criterion associated with a connectedness-based criterion for clustering heterogeneous structured data. In the case of the centered-based clustering optimization, it is common to see other schemes for the objectives: (i) a compactness-based criterion and the number of the clusters, (ii) a combination of the two compactness-based criteria, (iii) a compactness-based criterion and a spatial separation-based criterion. In this last case, these different configurations of objective functions are mostly related to specific classes of clustering studies, such as bi-clustering (i), categorical data clustering (ii and iii). The same occurs with the crossover and mutation operators, in which we can observe a diversity of combinations of operators.

Table 3.2 summarizes the selection methods applied to each approach that provides this component in their design. As this component is not mandatory in the EMOC design, almost half of the presented algorithms do not provide it. The existing selection methods are, in general, as follows: ensemble-based, which provides the best solution (consensual partition); knee-based, which provides the best k-solution; and CVIs-based, which considers specific criteria (as ranking) to define the best set of solutions.

EMOC algorithms: initialization and optimization components

Year	Article	Representation	Initialization	MOEA	Objectives	Crossover/Mutation
2005	MOCK (Handl and Knowles, 2005a,b; Handl and Knowles, 2007)	LAG	MST, KM	PESA-II	(Con, Dev)	Uniform/Neighborhood-based
2005	MOCK-medoid (Handl and Knowles, 2005c)	LAG	KM	PESA-II	(Var, Dev)	Uniform/Neighborhood-based
2006	VRJGGA (Ripon et al., 2006a,b)	Centroid-based	Random	JGGA-based	(Ent, Sep_{AL})	Modulo/Polynomial
2006	MOCLE (Faceli et al., 2006)	Label-based	KM, AL, SL, SNN	SPEA2/NSGA-II	(Con, Dev)	HBGF or MCLA/—
2007	MOGA-medoid (Mukhopadhyay and Maulik, 2007)	Medoid-based	Random	NSGA-II	(Dev, Sil)	One-point/Medoid Repl.
2007	MOCK-scalable (Matake et al., 2007)	LAG	MST , KM	SPEA2	(Con, Dev)	Uniform/Neighborhood-based
2007	MOGA-fuzzy (Mukhopadhyay et al., 2007)	Mode-based	Random	NSGA-II	$(J_m, Sep_{\mathrm{fuzzy}})$	Uniform/Mode Repl.
2007	NSGA-II&FCM (Di Nuovo et al., 2007)	FCM parameters and features weights	FCM	NGSA-II	(Number of features, XB)	SBX/Polynomial
2009	MOGA-fuzzy2 (Mukhopadhyay et al., 2009)	Mode-based	Random	NSGA-II	(J_m, Sep_{fuzzy})	One-Point/Mode Repl.
2009	EMCOC (Ripon and Siddique, 2009)	Binary center-based	Random	JGGA-based	(Ent, Sep_{AL})	Center exchange /NA
2011	MOCA (Kirkland et al., 2011)	Medoid-based	Random	NSGA-II	$(AWGSS, ABGSS, Con)$	Centroids exchange/ Split, Merge, Centroid Repl.
2011	MOC-HCM (Sert et al., 2011, 2012)	Binary-based	NA	NSGA	$(Km_{\mathrm{id}}, Km_{\mathrm{ed}}, Km_{\mathrm{wid}}, Km_{\mathrm{wed}}, EWCD)$	Shuffle/NA
2012	IMOCLE (Liu et al., 2012)	Label-basaed	KM, AL, CoL, SPC	NSGA-II	$(Con, ConP, Dev, Sim)$	MCLA/—
2012	Hybrid MOGA (Dutta et al., 2012b,c)	Centroid-based	Random	specific MOEA	(Sep_{CL}, H)	Pairwise/Centroid Repl.
2012	MOSSC (Zhu et al., 2012)	Center and weight-based	Random	NSGA-II	$(SSXB, J_{wm})$	SBX/Polynomial
2012	MIE-MOCK (Tsai et al., 2012)	LAG	MST , KM	PESA-II	(Con, Dev)	Uniform, One-Point and Two-Point/ Neighborhood-based, Split and Merge

Table 3.1 – continued from previous page

Year	Articles	Representation	Initialization	MOEA	Objectives	Crossover/Mutation
2013	MOGGC (Menéndez et al., 2013)	Label-based	Random	SPEA2	(sep_{graph}, DCD)	Labels Exchange/Adaptive
2013	MOEASSC (Xia et al., 2013)	Center and weight-based	Random	NSGA-II	(J_{wm}, J_{Add})	One Point/Uniform
2014	CEMOG (Menéndez et al., 2014)	Label-based	Random	SPEA2	$(sep_{graph}, DCD$	Labels Exchange/Adaptive
2014	FCM-NSGA (Wikaisuksakul, 2014)	Center-based	Random	NSGA-II	(J_m, Sep_{nfuzzy})	SBX/Polynomial
2016	SRMOSC (Luo et al., 2015)	Sparse-based	Neighbor-based	NSGA-II/ MOEA/DD	(RE, SP)	specific operators based on the sparsity property
2017	Δ-MOCK (Garza-Fabre et al., 2017, 2018)	Reduced LAG	MST	NSGA-II	(Con, Var)	Uniform/Neighborhood-based
2017	BI-MOCK (Bousselmi et al., 2017)	LAG with conditions	MST, KM	PESA-II	(Var, size of the bi-cluster)	Ad. Two-Points/ CC
2018	MOKCW (Z. Zhou, 2018)	Center and weight-based	Random	NSGA-II	(Ad. J_{wm}, Ad. $SSBX$)	One-Point/Uniform
2018	EMO-KC (Wang et al., 2018)	Centroid-based	Random	NSGA-II	(Mod. Var, k)	SBX/ Polynomial
2018	FRC-NSGA (Paul and Shill, 2018)	Center-based	Random	NSGA-II	J_m and Sep_{nfuzzy}	SBX/Polynomial
2018	Δ-EMaOC (Zhu et al., 2018)	Reduced LAG	MST	NSGA-III/ RVEA/ MOEA/DD/ SPEA-II-SDE	$(Con, Var, Dunn, DB, CH)$	Uniform/Neighborhood-based
2018	MOECDM (Liu et al., 2018)	Label-based	Random, NCUT	NSGA-II	(Sep_{CL1}, Sep_{CL2})	Ad. Uniform/Ad. Uniform
2018	MOEACDM (Liu et al., 2018)	Label-based	NCUT	NSGA-II	(Mod_1, Mod_2)	Ad. Uniform/ Ad. Uniform
2018	ADNSGA2-FCM (Dong et al., 2018)	Center and FM-based	Random	NSGA-II	DB and $\mathcal{I}$	Neighborhood-based, Truncation and Stitching/ Uniform
2018	MaOFcentroids (Zhu and Xu, 2018)	FM-based	Random	NSGA-III	$(CDCS, DB, CH, CCI, XB)$	Uniform/specific Mutation for Membership Repl.
2019	MOBICK (Bechikh et al., 2019)	Reduced LAG with conditions	MST, KM	PESA-II	(Var, size of the bi-cluster)	Ad. Uniform/CC
2019	MOGA-KP (Dutta et al., 2019)	Centroid-based	Random	specific MOEA	(Sep_{CL}, H)	One-Point/Polynomial
2020	MOAC-L (Zhu et al., 2020)	Reduced LAG	MST	adapted NSGA-II	Con, Var	Uniform/ Neighborhood-based

Table 3.2: EMOC algorithms: selection strategies

Year	Article	Final Selection
2005	MOCK (Handl and Knowles, 2005a,b; Handl and Knowles, 2007)	Knee-based
2005	MOCK-medoid (Handl and Knowles, 2005c)	Knee-based
2007	MOCK-scalable (Matake et al., 2007)	Knee-based
2007	MOGA-fuzzy (Mukhopadhyay et al., 2007)	Specific approach (k-nn-based)
2009	MOGA-fuzzy2 (Mukhopadhyay et al., 2009)	Specific approach (k-nn-based)
2011	MOC-HCM (Sert et al., 2011, 2012)	Ensemble-based (H-confidence)
2012	MIE-MOCK (Tsai et al., 2012)	PBM and DB
2012	MOSSC (Zhu et al., 2012)	Ensemble-based (HBGF)
2013	MOGGC (Menéndez et al., 2013)	sep_{graph}
2013	MOEASSC (Xia et al., 2013)	PSVIndex
2014	CEMOG (Menéndez et al., 2014)	sep_{graph}
2016	SRMOSC (Luo et al., 2015)	Ratio cut-based
2018	MOKCW (Z. Zhou, 2018)	PSVIndex and ensemble-based (HBGF or MCLA)
2018	EMO-KC (Wang et al., 2018)	Elbow-based and DB
2018	ADNSGA2-FCM (Dong et al., 2018)	Ensemble-based (Majority vote)
2018	MaOFcentroids (Zhu and Xu, 2018)	Ensemble-based (SIVID)
2019	MOGA-KP (Dutta et al., 2019)	DB, Dev, $Dunn$, C, $COSEC$, Ent, F-Measure, $Purity$ and XB

3.3 MOCK, Δ-MOCK, MOCLE AND EMO-KC

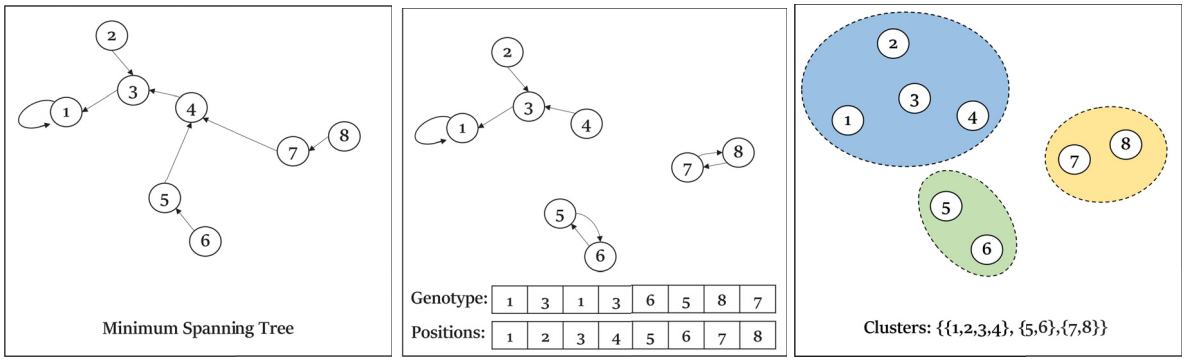
In this section, we present more details of four approaches: MOCK, Δ-MOCK, MOCLE and EMO-KC. These approaches were used in our experiments and they are compared to the proposed approach in Chapter 7.

3.3.1 MOCK

MOCK (Multi-Objective Clustering with automatic K-determination) is a well-known algorithm for multi-objective clustering (Handl and Knowles, 2005a; Handl et al., 2007).

To encode the solutions (partitions), MOCK uses a graph-based encoding called locus-based adjacency representation (Handl et al., 2007): a solution is represented as a vector of genes, and each gene g_i can take an integer value between 1 and n, where n is the number of objects in the dataset. If a value j is assigned to the ith gene, it can be interpreted as a link between the data points i and j, i.e., i and j belong to the same cluster. Figure 3.3 illustrate a partition encoding applied in MOCK.

In terms of the generation of the initial population, MOCK uses the partitions derived from the Minimum Spanning Tree (MST) clustering and k-means. In particular, the MST-clustering implemented in MOCK considers the use of a measure called *degree of interestingness* (DI) to define the most relevant links that are removed to obtain the clusters. Besides that, a link removed at position i is subsequently replaced by a link to a randomly chosen neighbor. These procedures are applied to amend the separation of outliers (Handl et al., 2007). For both MST-clustering and k-means, partitions with different numbers of clusters are generated to compose the initial population. The partitions generated by k-means are converted to the locus-based encoding by removing all MST links crossing cluster boundaries in the partitions.



(a) MST (b) One possible genotype (c) Clusters in the partition

Figure 3.3: MOCK representation. Adapted from Handl and Knowles (2007)

Regarding the multi-objective algorithm, MOCK uses PESA-II (Corne et al., 2000). It is applied to optimize two objective functions: Dev and Con. The evolutionary operators used in this algorithm are standard uniform crossover and a neighborhood-biased mutation scheme. In particular, links selected by random are removed and they are replaced by a link randomly chosen neighbor, in the same way of the initialization.

At last, MOCK has a model selection applied to select the best partitions in the Pareto Front. It considers a comparison of the shape of the curve (knee) obtained in the optimization with a null model, produced by clustering random data.

3.3.2 Δ-MOCK

Δ-MOCK (Garza-Fabre et al., 2017) was developed to improve the scalability of MOCK (Handl et al., 2007) considering modifications applied to: (i) the initialization and representation schemes, (ii) the multi-objective optimization algorithm, (iii) the objective functions.

In terms of the initialization procedure, according to Garza-Fabre et al. (2017) the use of two approaches to generate the base partitions affects the general scalability of MOCK, specifically k-means. Thus, Δ-MOCK uses only one approach to generate the base partitions, considering the one that removes the links of the MST.

Furthermore, according to (Garza-Fabre et al., 2017), one of the main limiting factors regarding MOCK's scalability is the length of the genotype in the locus-based adjacency representation, which is equal to the number n of objects in the dataset (see Section 3.3.1). To address this issue, Garza-Fabre et al. (2017) introduced two alternative representations: the Δ-locus and the Δ-binary encodings. These schemes are based on the original representation of MOCK. However, they can significantly reduce the length of the genotype by making use of information from the MST and DI. More specifically, based on a user-defined parameter, δ ($0 \leq \delta \leq 100$), the MST links are classified either into the set of relevant links, Γ, or into the set of non-relevant, fixed links, Δ. Only the relevant links are used in the optimization, i.e., the new encoding has a $|\Gamma|$-length genotype.

Fig. 3.4 illustrate an decode of the full-length representation to Δ-locus. By considering a dataset with 12 objects and $\delta = 80$, Δ-locus encoding has a size equal to 3, while a full-encoding has 12. In this case, only the most relevant links of the MST (linked to positions 3, 7, and 10) are operated by the evolutionary operators in this new encoding.

On the other hand, Fig. 3.5 illustrate the decode Δ-locus to the full-length representation, in which the relevant links are replaced by new links.

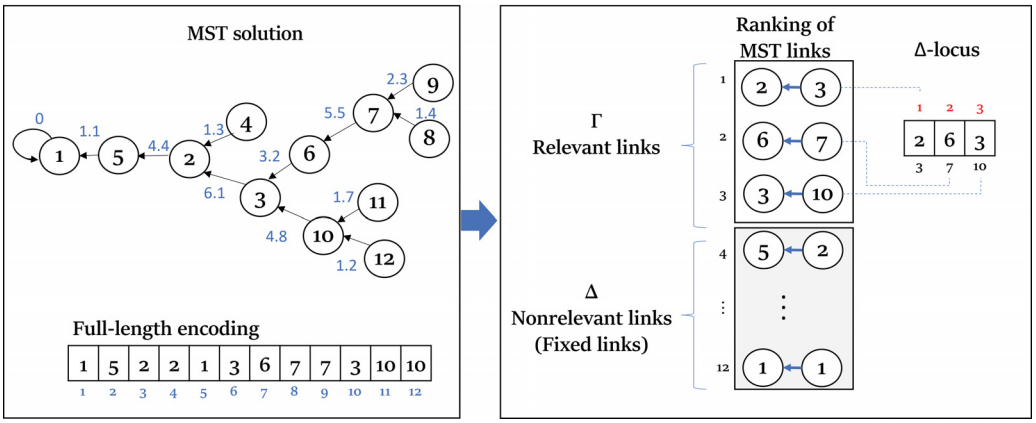


Figure 3.4: Encoding the full-length representation to Δ-locus. In the Δ-locus representation, the numbers above the encoding, in red font, represent the rank of the relevant links, and the numbers below the encoding represent the position in the full-length encoding. Adapted from Garza-Fabre et al. (2017)

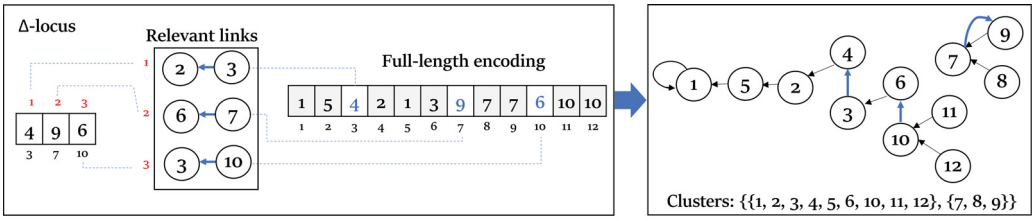


Figure 3.5: Decoding the Δ-Locus to full-length representation. The numbers in red font refer to the rank of the relevant links, and the numbers in blue font denote the modified links. Adapted from Garza-Fabre et al. (2017)

Regarding the search strategy and objective functions, Δ-MOCK replaces MOCK's PESA-II (Corne et al., 2000) with NSGA-II (Non-dominated Sorting Genetic Algorithm II) (Deb et al., 2000) to optimize the *Var* and *Con*, in which *Var* was used instead of *Dev* to support their pre-computation of the fixed links applied in new representation schemes.

3.3.3 MOCLE

MOCLE (Multi-Objective Clustering Ensemble) is a clustering algorithm proposed by Faceli et al. (2006) that combines characteristics from both cluster ensemble techniques and multi-objective clustering methods.

The ensemble clustering generates a consensual partition, π^* according to the basic process of the cluster ensemble presented in Fig. 3.6 and explained in the following. Let $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ be a set of n data points, and $\Pi = \{\pi_1, \ldots \pi_M\}$ be a set of partitions generated by one or more clustering algorithms, a consensus function combines these partitions to obtain the final clustering result π^*, and to improve the quality of the clustering results (Faceli et al., 2006).

Like in traditional ensemble clustering, starting with a diverse set of base partitions, MOCLE employs the multi-objective evolutionary algorithm to generate an approximation of the Pareto optimal set. It optimizes the same criteria as MOCK and uses a special crossover operator, which combines pairs of partitions using an ensemble method. No mutation is employed.

MOCLE uses the label-based representation, in which each position denotes the cluster label of the respective point. This representation supports the use of different clustering algorithms in the initialization. In contrast to MOCK and Δ-MOCK, in which the links should be evaluated

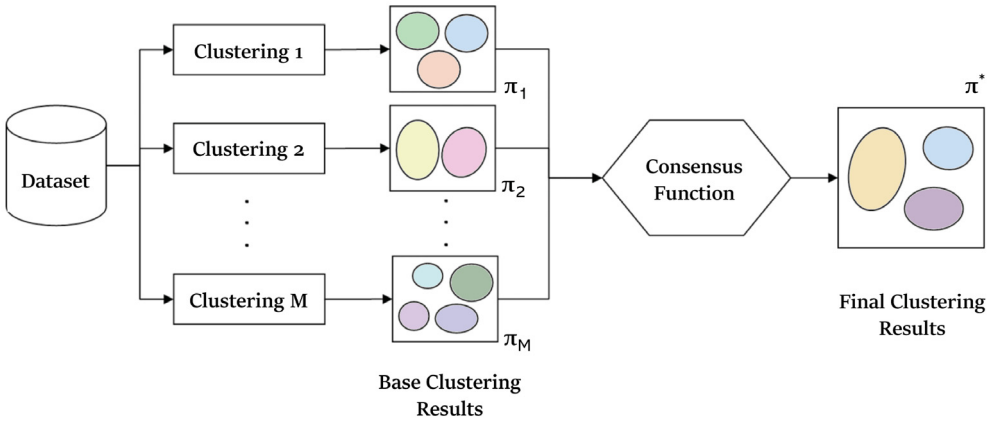


Figure 3.6: Ensemble Clustering.

to determine the clusters (linked points), label-based representation does not require any extra effort to obtain the clusters since the cluster labels is direct given in the encoding.

Finally, it is worth noting that the core ideas of MOCLE, as well as those of MOCK and Δ-MOCK, are not linked to specific objective functions, crossover operators, and search algorithms. For instance, in terms of objective functions, like MOCK, MOCLE has been mainly implemented using Dev and Con (Faceli et al., 2006, 2009; Antunes et al., 2020). Concerning the crossover operators, the software available at `http://lasid.sor.ufscar.br/mocleproject/` implements the MCLA (Strehl, 2002) and the cluster ensemble method HBGF (Fern and Brodley, 2004). The optimization process, like in Δ-MOCK, has been mainly performed by using NSGA-II (Deb et al., 2000).

3.3.4 EMO-KC

EMO-KC (multi-objective optimization-k-clustering) was introduced by Wang et al. (2018). This algorithm uses a centroid-based representation, in which the chromosomes consist of real numbers that represent the coordinates of the cluster centroid. To generate the initial population, it considers a random choice of the points in the dataset to define the initial centroids, and the clusters consist of objects in which each point is associated with the closest centroid.

EMO-KC relies on the NSGA-II with its standard operators (simulated binary crossover and polynomial mutation) to optimize Var' and k (number of clusters). According to the authors, this approach was proposed to harness the implicit parallelism of EMOC, for that they introduced the adapted SSD (Var'), to improve the conflict of any two solutions having different k values.

To select the best solution EMO-KC considers the elbow method (Hancer and Karaboga, 2017). The final population could present more than one elbow or no elbow for some datasets, thus the DB is further considered to select the final solution.

3.4 EMOC APPROACHES DESIGNED FOR SPECIFIC APPLICATIONS

In this section, we present approaches designed for specific applications. Each algorithm considers the particularities of problem application to define the representation of the solutions, the objective functions, or/and the evolutionary operators. It promotes the generation of a variety of configurations, so we will limit ourselves to listing some algorithms designed for each following application.

3.4.1 Association rule learning

Association rule learning is a rule-based machine learning method for discovering interesting relations between variables in large databases. Kaya and Alhajj (2004) and Alhajj and Kaya (2008) provided an EMOC approach for fuzzy association rules mining to automatically cluster values of a given quantitative attribute to obtain a large number of itemsets in a short period of time.

3.4.2 Document clustering

Document clustering is a data/text mining technique that makes use of text clustering to divide documents according to various topics. Lee et al. (2014) proposed a method of enhancing multi-objective genetic algorithms for document clustering with parallel programming. Wahid et al. (2015) presented a new approach for document clustering based on SPEA-II, that explores the concept of multiple views to generate multiple clustering solutions with diversity.

3.4.3 Gene/micro-array analysis

The Gene/Micro-array clustering analysis is applied to discover groups of correlated genes potentially co-regulated or associated with the disease or conditions under investigation. Romero-Zaliz et al. (2008) provided an EMOC to identify conceptual models in structured datasets that can explain and predict phenotypes in the immune inflammatory response problem, similar to those provided by gene expression or other genetic markers. Li et al. (2017) provided a new ensemble operator to improve the data clustering in gene expression datasets in IMOCLE (Liu et al., 2012). Mukhopadhyay et al. (2010) provide an approach that simultaneously selects relevant genes and clusters the input dataset. Mukhopadhyay et al. (2013) presented an interactive approach to multi-objective clustering of gene expression patterns considering an adapted NSGA-II, in which inputs from the human decision-maker (DM) are taken to learn which objective functions are more suitable for the datasets. Dutta and Saha (2017) presented an EMOC approach to identify gene clusters from a given expression dataset; in which apart from utilizing the gene expression values of the individual genes, the corresponding protein-protein interaction scores are also used while clustering the set of genes.

3.4.4 Image Segmentation

Image segmentation consists of the process by which a digital image is partitioned into various subgroups (multiple parts or regions), often based on the characteristics of the pixels in the image. Qian et al. (2008) presented a multi-objective evolutionary ensemble algorithm to perform texture image segmentation. Shirakawa and Nagao (2009) introduced a variation of the MOCK (Handl and Knowles, 2007) improving its general features for its application in image segmentation. Zhang et al. (2016) provided a multi-objective evolutionary fuzzy clustering for image segmentation, considering the original FCM energy function to preserve image details and a function based on local information to restrain noise, both minimized by MOEA/D. Zhao et al. (2018, 2019) introduced the use of the concepts of intuitionistic fuzzy set (IFS) and multiple spatial information to generate an EMOC approach to overcome the effect of noise in image segmentation.

3.4.5 Software module clustering

Software module clustering refers to the problem of automatically organizing software units into modules to improve program structure. Praditwong et al. (2010) provided a multi-objective formulation of the software module clustering problem considering a two-archive Pareto optimal genetic algorithm. Barros (2012) provided an analysis of the effects of composite objectives in multi-objective software module clustering.

3.4.6 Network community detection

Network community detection refers to the procedure of identifying groups of interacting vertices in a network depending upon their structural properties to unveil the dynamic behaviors of networks. Folino and Pizzuti (2010) provided an approach for the detection of communities with temporal smoothness formulated as an EMOC. Attea et al. (2016) reformulate the community detection problem as an EMOC model that can simultaneously capture the intra and inter-community structures based on functions inspired by different types of node neighborhood relations. Shang et al. (2017) introduced an EMOC approach based on k-nodes update policy and a similarity matrix for mining communities in social networks. Pizzuti and Socievole (2019) provided a framework for detecting community structure in attributed networks, introducing a post-processing local search procedure that identifies those communities that can be merged to provide higher quality community divisions.

3.4.7 Web recommendation

Web topic mining and web recommendation consider the problem of extracting web navigation patterns, based on the interests of a user, to be applied in the recommender systems to guide users during their visit to a Web site. Demir et al. (2010) presented EMOC approaches to clustering Web user sessions in a Web page recommender system. Morik et al. (2012) investigated the problem of finding alternative high-quality structures for (Web) navigation in a large collection of high-dimensional data, and they provided a formulation of FTS (Frequent Terms Set) clustering as a multi-objective optimization problem.

3.4.8 WSN - Wireless Sensor Network topology management

There are several challenges in designing WSN because the sensor nodes have limited resources of energy, processing power, and memory. In this context, the clustering technique can organize nodes into a set of groups based on a set of pre-defined criteria to improve their usage. Peiravi et al. (2013) provided an EMOC approach whose goal was to obtain clustering schemes in which the network lifetime was optimized for different delay values. Hacioglu et al. (2016) presented an EMOC approach that can extend network lifetime while enabling high coverage and data.

3.4.9 Other applications

Wang et al. (2015) proposed an approach to solve the circuit clustering problem in field-programmable gate array computer-aided design flow. Mukhopadhyay and Maulik (2009) introduced a multi-objective genetic clustering approach for pixel classification in remote sensing imagery. Wang et al. (2014) and Li et al. (2016) provided a multi-objective fuzzy clustering approach for change detection in Synthetic Aperture Radar (SAR) images. Liu et al. (2017) presented an approach to automatic clustering of shapes considering a multi-objective optimization with decomposition and improvement in the shape descriptor and diffusion process (that was

applied to transform the similarity distance matrix among total shapes of a dataset into a weighted graph).

3.5 CHAPTER REMARKS

In this chapter, we presented a review of the EMOC studies, focused on a general architecture of evolutionary multi-objective clustering (see Chapter 2), considering the chromosome representation, initialization strategies, MOEAs (or MaOEAs), objective functions, evolutionary operators (crossover and mutation), and final solution selection. Furthermore, in this manuscript, we presented some applications of EMOC and the most relevant related papers that can be useful to researchers that are exploring EMOC for a specific purpose.

This mapping of EMOC approaches allows us to observe some patterns and obtain some insights regarding the evolutionary multi-objective clustering algorithms. For example, the choice of the objective functions is one of the most critical factors in the optimization process. In general, there is no consensus around the ideal number and the best combination of objective functions among researchers because of the difficulty in defining appropriate clustering criteria. In this way, more studies on the objective functions are required to improve the composition of objective functions and provide more information on the limitations of the existing ones.

In terms of an evolutionary multi-objective approach, we can note the wide use of the NSGA-II as MOEAs over the years. In recent years, the use of MaOEAs has been verified (Zhu et al., 2018; Zhu and Xu, 2018), in contrast to other works (Sert et al., 2011, 2012; Liu et al., 2012) that considered the optimization of more than three objective functions in MOEAs (NSGA/NSGA-II).

Other multi-objective clustering works were published recently (between 2021 and 2022), but they do not provide novelty in the analysis of the EMOC approaches, as described in this manuscript. For example, Zhu et al. (2021) proposed HT-MOC - hierarchical topology-based MOC. HT-MOC is a MOCK-based algorithm, that uses a hierarchical topology-based cluster representation to improve the time and memory usage. Besides that, this approach uses a specific MOEA to optimize Con and Var, which considers an ensemble-based operation after the crossover and mutation operations, aiming to improve the quality of the solutions. Like in the other cited works, HT-MOC uses a clustering algorithm to generate the base partitions (MST-clustering), but the impact of using high-quality partitions in the initialization is not evaluated.

The literature review presented in this chapter was submitted to the journal Computer Science Review and is still under review. A preprint is available in Morimoto et al. (2021).

In the next chapter, we introduce the admissibility analysis applied to evaluate the search direction and the potential of finding the optimal solutions, in which we evaluate the impact of the initial population in the optimization.

4 ANALYSIS OF THE INADMISSIBILITY OF THE OBJECTIVE FUNCTIONS IN EMOC APPROACHES

In this chapter, we introduce the analysis of the inadmissibility applied to the objective functions and the influence of the initialization strategy in the optimization, in order to answer **RQ.1** and **RQ.2**. This chapter is divided into four sections. In the first section, we present the admissibility and inadmissibility concept applied in the analysis of the objective functions. In sequence, we present objective functions and the clustering algorithms applied in most of the EMOC approaches that consider high-quality base partitions in the initialization to evaluate their impact in the optimization. In the next two sections, we present the experimental setup and results of experiments considering the analysis of 17 objective functions as for their inadmissibility in 24 artificial datasets. Finally, we present the general discussion and analysis of the results.

4.1 ADMISSIBILITY AND INADMISSIBILITY OF OBJECTIVE FUNCTIONS

An admissible heuristic function can be characterized as a function that does not overestimate the cost of reaching the goal (Russell and Norvig, 2002).

In our study, we consider this general admissibility concept in evolutionary optimization. Thus, we verified the potential of the objective functions, as heuristic functions, in finding the optimal results based on the search direction. Here, we considered the **optimal value** as the underlying structure of the data, called the **true partition** or **ground-truth**. In other words, the true partition represents the ideal model partition. As our analysis considered artificial datasets, the true partition was known in advance, making it possible to perform a detailed examination of the underlying structure of the data and relate it to the clustering criteria.

In practice, our analysis consists of evaluating the inadmissibility of the objective functions. An objective function is inadmissible if for each $f(\pi)$, $\pi \in \Pi_0$ (initial population), $\exists f(\pi) \geq f(\pi^*)$ for the maximization problem, or $\exists f(\pi) \leq f(\pi^*)$ for a minimization problem, where $f(\pi)$ denotes an objective function result for each candidate solution π, and π^* represent the optimal solution.

It is important to note that our analysis does not ensure that the objective functions are admissible or that they can reach optimal values. However, it is possible to clearly visualize the inadmissible objective functions and the potential for optimization of the other objective functions.

Fig. 4.1 illustrates the general case of the inadmissibility applied in our analysis, considering the optimization (minimization) of one objective function over a period of time. In this figure, the term "cost" refers to the best result of the objective function in each iteration, here called "time". In this figure, the red point represents the optimal value (optimal cost), and the blue arrow indicates the search direction. The objective function is inadmissible in order to find the optimal solution ($\exists f(\pi) \leq f(\pi^*)$), in which the initial cost is 7 for the solutions in the initial population, that overestimates the optimal cost (9). Furthermore, the optimization of this objective function worsens this aspect over time (final cost = 0.8).

Inadmissible functions are not adequate to be optimized; however, they can be applied as restrictions to the search space (see Eq. 2.2 in Section 2.2) to define the feasible region of solutions. In particular, the inadmissible functions can be applied as objective functions to constrain the search in a specific direction.

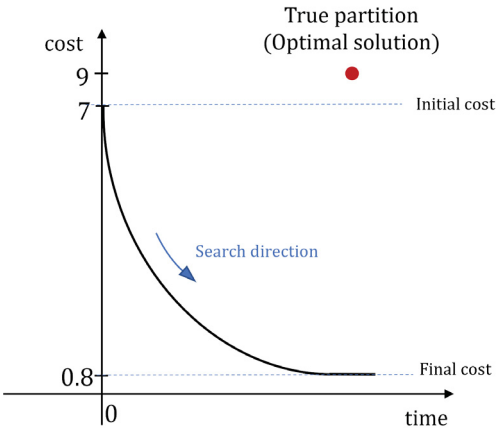


Figure 4.1: Example of an inadmissible objective function, considering the best results of the objective function over a period of time

In the following, we present the objective functions, and the algorithms applied to generate the base partitions used in the admissibility analysis.

4.1.1 Objective Functions

In our study, we analyzed different objective functions:

- Four compactness criteria: intra-cluster entropy (Ent), overall deviation (Dev), intra-cluster variance (Var), and total within-cluster variance ($TWCV$).
- Two connectedness criteria: connectivity (Con) and data continuity degree (DCD).
- Four separation criteria: average between-group sum of squares ($ABGSS$), average separation (Sep_{AL}), separation index (Sep_{CL}), and graph-based separation (Sep_{graph}).
- Seven compactness and separation criteria: Calinski-Harabasz index (CH), Davies-Bouldin index (DB), Dunn index ($Dunn$), modularity (Mod), silhouette (Sil), Pakhira-Bandyopadhyay-Maulik (PBM), and Xeni-Beny (XB).

These objective functions were extracted from the evolutionary multi-objective clustering approaches detailed in Section 3. We selected clustering criteria that can be applied as objective functions to clustering continuous data and that differ in their general computation. To delimit the number of the clustering criteria, we do not consider variations of popular indices, or specific scope objective functions. Details of each objective function are presented in Appendix A.

4.1.2 Clustering Algorithms applied in initialization of EMOC approaches

Here, we present five clustering algorithms: k-means, average linkage, single linkage, shared nearest neighbor-based clustering, and minimum spanning tree clustering. These clustering algorithms provide different strategies that allow us to evaluate how they can affect the optimization.

4.1.2.1 k-means

k-means (KM) (MacQueen, 1967) is a partitional clustering algorithm applied to detect compact clusters. Its objective is to minimize the distance between the centroid and their respective

instances. The k-means starts by choosing a k set of centroids randomly (or based on prior knowledge and associating each object with the nearest centroid), where k is a user-given parameter. After that, the centroids are recomputed based on the current cluster data, followed by a new association of each instance with the nearest centroid; this operation is successively repeated until there is no change in the groups or the stopping criterion is met.

4.1.2.2 Average linkage and single linkage

Average linkage (AL) and single linkage (SL) are hierarchical algorithms applied to detect nested or hierarchical data structures. Each instance starts out standing as an individual cluster in both algorithms, and a sequence of merge operations is executed until it reaches a single cluster with all the instances. The core difference between AL and SL is the distance measure used to compute proximity between pairs of clusters. This measure is used to define the closest pair of sub-sets that are merged. SL uses the minimal distance between two instances of a cluster pair, and AL applies the average distance of all observations of the cluster pairs (Xu and Wunsch, 2005).

4.1.2.3 Shared nearest neighbor-based clustering

Shared nearest neighbor-based clustering (SNN) (Ertöz et al., 2002) is a density-based algorithm. SNN can detect clusters of different sizes, shapes, and densities. The main idea behind this algorithm is to use the concept of similarity based on the shared nearest neighbor. The objects are assigned to a cluster that shares a large number of their nearest neighbors (the density-based on the neighborhood). This algorithm begins with the computation of the similarity matrix, which is sparsified by retaining only the k-nearest neighbors (KNN). In the following, the shared nearest neighbor graph is constructed, in which links are created between pairs of objects that have each other in their KNN lists. Then, SNN computes the number of shared neighbors between vertices, considering the links coming from each point in the graph, providing the density factor. This factor is used to identify the noise or core points based on the user-defined thresholds. Then, noisy points are discarded, and the clusters are formed by the core points and the border points (non-noise non-core points), considering all the connected components.

4.1.2.4 Minimum spanning tree clustering

The minimum spanning tree (MST) clustering is a graph-based algorithm that can identify clusters of arbitrary shapes. Among a variety of versions of this algorithm, we consider here the MST-clustering described in Handl et al. (2007). This algorithm uses the concept of *degree of interestingness* (DI) and the properties of the MST to find the clusters. DI defines the neighborhood relationship between the nodes in the MST, where a link between two nodes is considered interesting if neither of them is a part of the other node's set of nearest neighbors. Thus, the clusters are generated by removing interesting links in the MST that split it into sub-graphs in which the connected elements represent a cluster.

4.2 EXPERIMENTAL DESIGN

4.2.1 Goals of the experiments

Besides the general goals of our research, the specific goals of these experiments are to answer the following research questions: (i) "Which evaluated objective functions are inadmissible and which ones have potential (search space) for optimization?", and (ii) "Are there specific features

or optimizing scenarios that should be considered in the choice and combination of the objective functions to obtain better clustering results?".

4.2.2 Experimental setup

In order to answer the first research question, we evaluated the admissibility of each objective function presented in Section 4.1.1, considering the base partitions obtained from different initialization strategies. Therefore, we used the initialization algorithm of the MOCK and MOCLE, both established and popular approaches, that consider different clustering criteria. Thus, we generated five initial populations using the clustering algorithm: KM, AL, SL, SNN, and MST-clustering. The general setting applied in the KM, AL, SL, and SNN is the same as reported in Faceli et al. (2006). Regarding the MST-clustering, we employed the general setting presented in Handl and Knowles (2007). Furthermore, we adjusted such algorithms to produce partitions containing clusters in the range $\{2, 2k^*\}$, where k^* is the number of clusters in the true partition representing the dataset. This setting is commonly used in MOCK/Δ-MOCK's to define the number of clusters in the partitions of the initial population. In this initial experiment, in particular, we analyze the admissibility by comparing the individuals of the initial populations with the optimal value (true partition) of each dataset. The results of this experiment provide us with information about which objective function and initialization strategy could improve the optimization. In particular, we demonstrated in MOCLE this impact in terms of ARI, in which we consider the MOCLE general setting present in Faceli et al. (2006).

Regarding the second research question, we evaluated different combinations of objective functions and analyzed which conditions could lead the EMOC approach to provide better results. In particular, we analyze the clustering performance of some promising objective functions found in the first experiment. Experiments were carried out in the new MOCK version, Δ-MOCK (Garza-Fabre et al., 2018). We select this algorithm, among others, because it is a recently established approach in which the present features (as the use of MST-clustering in the initialization) contribute to the evaluation in terms of the search direction, demonstrating how admissibility supports the choice of objective functions. Regarding this algorithm setting, we employed the one reported in Garza-Fabre et al. (2018), considering the Δ-locus scheme with δ settled heuristic $\sim 5/\sqrt{n}$, where n is the number of objects in the dataset.

4.2.3 Datasets

As previously stated, our analysis takes into account the use of the true partition. Thus, we selected 24 artificial datasets, in which we can analyze the relationship between their data structures or cluster shapes and the optimization of the objective functions. Table 4.1 summarizes the main characteristics of these datasets, in which n is the number of objects, d refers to the number of attributes (dimensions), and $k*$ is the number of clusters in the true partition. These datasets were obtained from 4 repositories: Clustering benchmarks[1] and Clustering basic benchmark[2], UCI Machine Learning Repository[3] and Clusters Evaluation Benchmark[4].

We divided these datasets into four groups (column **G** in Table 4.1), considering similar data structures evaluated in our analysis. In the first group (G1), Fig. 4.2, we have 8 datasets with gaussian-like clusters and 4 datasets with hyper-spherical shaped clusters. `R15`, `D31`, `Engytime`, `Sizes5`, `Square1`, `Square4`, `Twenty`, and `Fourty` have gaussian-like

[1] `https://github.com/deric/clustering-benchmark`
[2] `http://cs.joensuu.fi/sipu/datasets/`
[3] `https://archive.ics.uci.edu/ml/datasets.php`
[4] `http://lasid.sor.ufscar.br/clustersEvaluationBenchmark/`

G	Dataset	n	d	k^*
G1	R15	600	2	15
	D31	3.100	2	31
	Engytime	4.096	2	2
	Sizes5	1.000	2	4
	Square1	1.000	2	4
	Square4	1.000	2	4
	Twenty	1.000	2	20
	Fourty	1.000	2	40
	Sph_5_2	250	2	5
	Sph_6_2	300	2	6
	Sph_9_2	900	2	9
	Sph_10_2	500	2	10

G	Dataset	n	d	k^*
G2	ds2c2sc13_S1	588	2	2
	ds2c2sc13_S2	588	2	5
	ds2c2sc13_S3	588	2	13
G3	Long1	1.000	2	2
	Pat2	417	2	2
	Spiral	1.000	2	2
G4	3MC	400	2	3
	DS-850	850	2	5
	Aggregation	788	2	7
	Complex9	3.030	2	9
	Pat1	557	2	3
	Spiralsquare	2.000	2	6

Table 4.1: Dataset characteristics

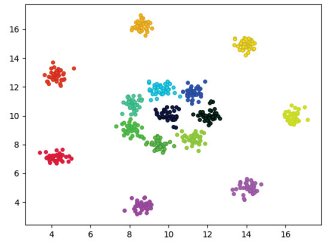
clusters. R15 consists of 15 identical-sized clusters with some overlapping points. D31 has 31 clusters that are slightly overlapping and distributed randomly. Engytime has two highly overlapping clusters with different variances. Size5 has five clusters of varying sizes and the same inter-cluster distance over all clusters. Square1 and Square4 consist of four clusters of equal size and spread that vary in the degree of overlap and the relative size of clusters. Fourty and Twenty consist of well-separated small clusters distributed into 40 and 20 clusters, respectively. Sph_5_2, Sph_6_2, Sph_9_2, Sph_10_2 have hyper-spherical shaped clusters with different proximity between the clusters. Algorithms based on cluster compactness, such as KM, can detect well-separated hyper-spherical shaped clusters; they can also detect gaussian-like clusters when they contain globular (no oblong) and well-separated data structures.

In the second group (G2), Fig. 4.3, we have the ds2c2sc13 dataset, which contains three different structures: S1, S2, and S3. These structures represent three levels of structures in a nested dataset. In this example, S1 represents two well-separated clusters, which can be found by techniques based on optimizing connectedness or compactness; in contrast, S2 and S3 combine distinct types of clusters that could be hard to find with techniques based only on connectedness or compactness. Hierarchical clustering algorithms, such as SL and AL, are usually applied to detect nested structures.

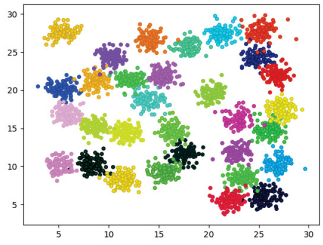
In the third group (G3), Fig 4.4, we have datasets that contain well-separated and elongated cluster shapes that are hard to identify for algorithms based on cluster compactness: Long1, Spiral, and Pat2.

In the last group (G4), Fig. 4.5, we have shaped datasets that combine different types of clusters: 3MC, DS-850, Aggregation, Complex9, Pat1, Spiralsquare. Aggregation contains 6 clusters with a uniform and compact distribution, and they also have different sizes, and two clusters are linked by a line of points. 3MC contains symmetrical shaped clusters (e.g., ring-shape, ellipsoidal clusters, etc.). Pat1 and Complex9 present clusters surrounding other ones, among other data structures. Spiralsquare combines spirals and square shapes into clusters.

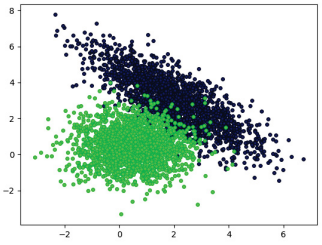
It is important to observe that the use of artificial datasets that provide the true partition and also have well-known data structures makes it possible to analyze in detail the conditions that can affect the optimization in our study.



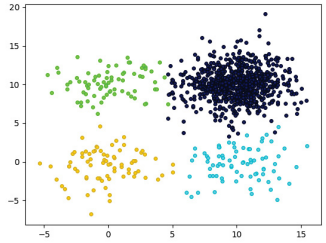
(a) R15



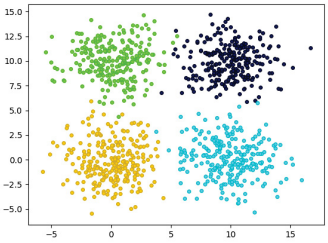
(b) D31



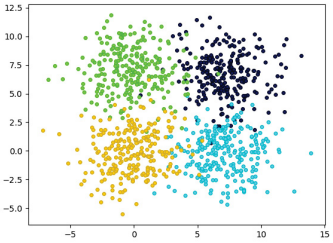
(c) engytime



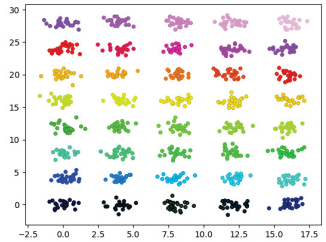
(d) Sizes5



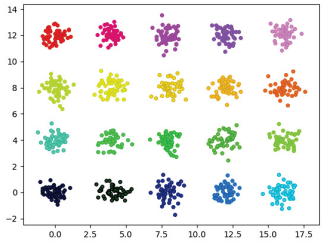
(e) Square1



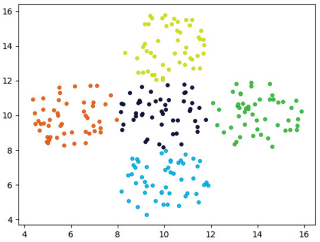
(f) Square4



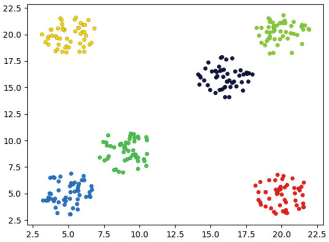
(g) Fourty



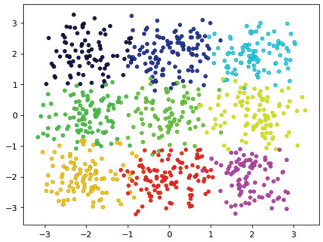
(h) Twenty



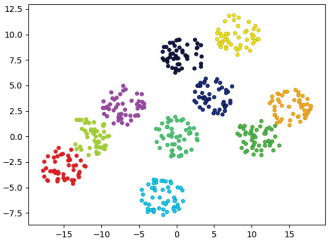
(i) Sph_5_2



(j) Sph_6_2

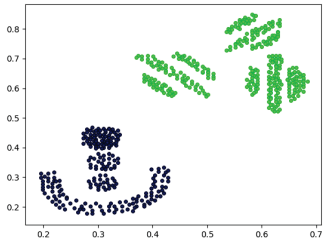


(k) Sph_9_2

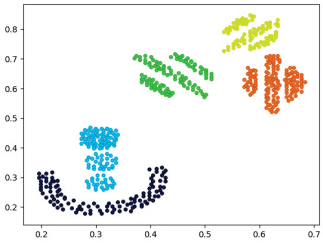


(l) Sph_10_2

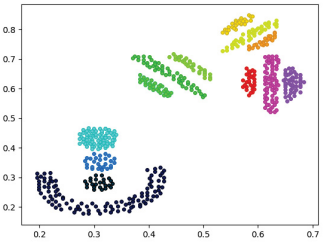
Figure 4.2: Datasets with gaussian-like and hyper-spherical shaped clusters



(a) S1

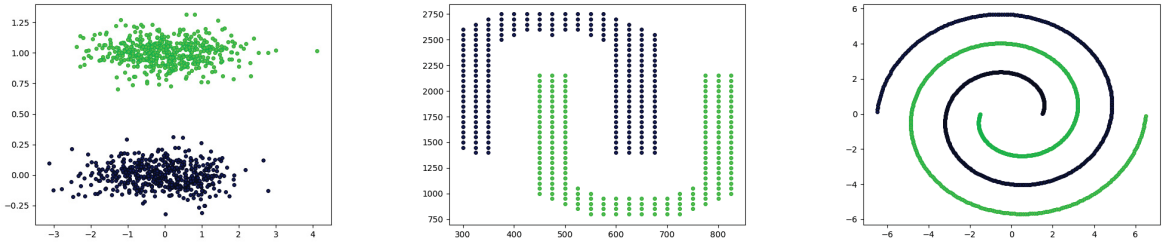


(b) S2



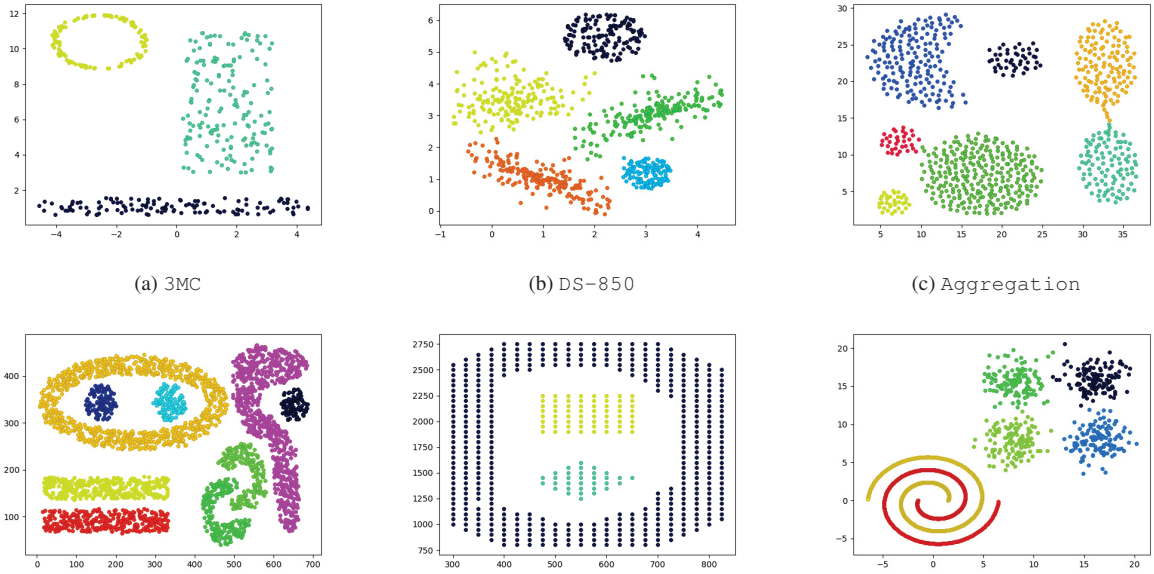
(c) S3

Figure 4.3: ds2c2sc13 data structures



(a) Long1 (b) Pat2 (c) spiral

Figure 4.4: Datasets with elongated cluster shapes



(a) 3MC (b) DS-850 (c) Aggregation

(d) complex9 (e) Pat1 (f) spiralsquare

Figure 4.5: Datasets with distinct types of clusters

4.2.4 Performance assessment

In terms of optimization, we evaluated the objective function's inadmissibility and analyzed the solutions regarding the dominance of the true partition. The general concepts of both these items (admissibility and dominance) are presented in Section 2.

Finally, as the main indicator of clustering performance, we used the ARI, Eq. 2.3 (see Section 2.4.4).

4.3 EXPERIMENTAL RESULTS

As described in Section 4.2, for every individual in each population, we computed the objective function presented in Appendix A and compared their results with the respective values of the true partition to determine their inadmissibility and the potential of the optimization.

Table 4.2 and Table 4.3 present the detailed results of the inadmissibility in terms of the initialization with MST-clustering and KM, respectively. In these tables, we point out the objective functions that are inadmissible (×) for each dataset. Also, in terms of the potentially admissible objective function, we consider two other classes: (i) the true partition was found in

G	Datasets	Ent	Dev	Var	$TWCV$	CH	DB	$Dunn$	Mod	Sil	PBM	XB	$ABGSS$	Sep_{AL}	Sep_{CL}	Sep_{graph}	Con	DCD
G1	R15	×						×	×		×		×	×		✓	×	×
	D31	×						×	×	×	×		×	×		×	×	×
	Engytime	×					×	×	×	×	×			×		×	×	×
	Sizes5	×						×	×								×	×
	Square1	×						×	×		×						×	×
	Square4	×						×	×		×					×	×	×
	Twenty	×	×	×	×	✓	✓	✓	×	✓	✓	✓	×	×	×	×	×	×
	Fourty	×	×	×	×	✓	✓	✓	×	✓	✓	✓	×	×	×	✓	×	×
	Sph_5_2	×	×	×	×			×	×		×		×	×	×	×	×	×
	Sph_6_2	×	×	×	×	✓	✓	×	×	✓	×	×	×	×	×	×	✓	×
	Sph_9_2	×						×	×		×		×	×		×	×	×
	Sph_10_2	×	×	×	×		×	×	×		×		×	×	×	×	×	×
G2	ds2c2sc13_S1	×	×	×	×	✓	✓	✓	×	✓	✓	✓	✓	✓	×	✓	✓	✓
	ds2c2sc13_S2	×	×	×	×	×	×	×	×	✓	×	×	×	×	×	×	✓	×
	ds2c2sc13_S3	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
G3	Long1	×	×	×	×	×	×	✓	✓	×	✓	×	✓	×	×	✓	✓	✓
	Pat2	×	×	×	×	✓	×	✓	×	×	✓	×	✓	×	×	×	✓	×
	Spiral	×	×	×	×	✓	×	✓	×	×	✓	✓	✓	×	×	✓	✓	×
G4	3MC	×	×	×	×	×	✓	✓	×	✓	✓	✓	×	✓	×	✓	✓	×
	DS-850	×							×					×			×	×
	Aggregation	×						×	×		×		×	×		×	×	×
	Complex9	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
	Pat1	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
	Spiralsquare	×				×	×	×	×	×	×		×	×		✓	×	×

Table 4.2: Results of the analysis of the admissibility of the objective functions considering an initialization with MST-clustering

the initial population, where optimization is not required (✓), and (ii) the objective function has space to be optimized and potential to be admissible (blank cells).

Table 4.4 summarizes the results of all initialization strategies. Since the initialization of the AL, SL, and SNN are comparable with the results of the MST-clustering and KM, we compiled the results by counting the number of datasets in which the objective functions are inadmissible. Column **IN** denotes the total number of the datasets where each objective function is inadmissible, and column **OP** refers to the total number of the datasets where the optimal solution is provided in the initial population. For example, the fields **IN** fulfilled with 24 mean that a specific objective function is inadmissible for any of the analyzed datasets.

4.4 DISCUSSION

By analyzing Table 4.2, we observed that every connectedness criterion (Con and DCD) provides results in which there is no space to be optimized since they are inadmissible, or the optimal result was found in the initial population (cells marked with × and ✓). In contrast, some objective functions that take into account the compactness or/and separation criteria could be used in the optimization of the datasets that include the Gaussian-like clusters and hyper-spherical clusters that have some degree of overlap in G1, or heterogeneous data structures with close objects between the clusters in G4 (blank cells). In general, MST-clustering fails in detecting close or overlapping clusters, and the use of a complementary objective function that considers a

G	Datasets	*Ent*	*Dev*	*Var*	*TWCV*	*CH*	DB	*Dunn*	*Mod*	*Sil*	*PBM*	*XB*	*ABGSS*	Sep_{AL}	Sep_{CL}	Sep_{graph}	*Con*	*DCD*
G1	R15	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
	D31	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
	Engytime	×	×	×	×	×	×	×	×	×	×	×	×	×	×		×	
	Sizes5	×	×	×	×	×			×	×			×		×			×
	Square1	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
	Square4	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
	Twenty	×	×	×	×	✓	✓	✓	×	✓	✓	✓	×	×	×	×	✓	×
	Fourty	×	×	×	×	✓	✓	✓	×	✓	✓	✓	×	×	×	✓	×	×
	Sph_5_2	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
	Sph_6_2	×	×	×	×	✓	✓	×	×	✓	×	×	×	×	×	×	✓	×
	Sph_9_2	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
	Sph_10_2	×	×	×	×	×	×	×	×		×	×	×	×	×	×	×	×
G2	ds2c2sc13_S1	✓	×	×	×	✓	✓	✓	×	✓	✓	✓	✓	✓	×		✓	✓
	ds2c2sc13_S2	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	✓	×
	ds2c2sc13_S3	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
G3	Long1	×	×	×	×	×	×			×		×	×	×	×			×
	Pat2	×	×	×	×	×	×	×		×	×	×	×	×	×			×
	Spiral	×	×	×	×	×	×			×		×	×	×	×			×
G4	3MC	×	×	×	×	×	×		×			×	×	×	×	×		×
	DS-850	×	×	×	×	×			×			×	×	×	×	×		×
	Aggregation	×	×	×	×	×		×	×	×	×	×	×	×	×	×		×
	Complex9	×	×	×	×	×	×		×	×		×	×	×	×	×		×
	Pat1	×	×	×	×	×	×	×		×	×	×	×	×	×	×		×
	Spiralsquare	×	×	×	×	×	×		×	×		×	×	×	×			×

Table 4.3: Results of the analysis of the admissibility of the objective functions considering an initialization with KM

Type	Objectives	MST		SNN		SL		AL		KM	
		IN	OP	IN	OP	IN	OP	IN	OP	IN	OP
Compactness	*Ent*	24	-	20	3	24	-	23	1	23	1
	Dev	14	-	11	3	12	-	24	-	24	-
	Var	14	-	11	3	12	-	24	-	24	-
	TWCV	14	-	17	1	15	-	24	-	24	-
Compactness and Separation	*CH*	7	6	5	9	4	8	13	3	20	4
	DB	10	5	9	8	14	4	18	4	17	4
	Dunn	16	7	16	7	17	6	16	4	14	3
	Mod	23	1	22	2	21	1	22	-	20	-
	Sil	7	5	6	7	7	6	14	3	20	3
	PBM	9	6	5	10	7	8	18	4	17	4
	XB	15	7	16	7	17	6	15	4	14	3
Separation	*ABSS*	16	3	13	5	13	4	21	1	23	1
	Sep_{AL}	19	2	13	5	20	3	23	1	22	1
	Sep_{CL}	14	-	11	3	12	-	24	-	24	-
	Sep_{graph}	14	7	6	11	13	4	19	1	16	1
Connectedness	*Con*	17	7	17	7	17	6	15	4	10	4
	DCD	22	2	24	-	24	-	24	-	22	1

Table 4.4: A summary of the results regarding the analysis of the admissibility of the objective functions

search in different criteria (direction) covered in the initialization can lead the EMOC to obtain better results.

In terms of the results shown in Table 4.3, we observe that for most of the objective functions, there is no space for optimization (cells marked with ✓or ×) when KM is applied in the initialization. Only the objective functions $Dunn$, PBM, Sep_{graph}, and Con have at least five datasets in which it is possible to improve the results (blank cells). In particular, we observed that Con has space to be optimized in ten datasets, including all datasets present in G3 and G4. In this case, KM fails to detect the elongated clusters, and the use of Con could lead the EMOC to find this kind of structure present in G3 and G4. In general, these results point out that the initialization with KM and AL provide limitations in optimizing most of the compactness or separation criteria for most datasets.

In the initialization with MST-clustering, SNN, and SL, the objective functions present similar behavior. All objective functions are inadmissible, or the initial population has the best results for datasets with well-separated clusters. Consequently, there is no space for optimizing any evaluated criteria for these features (well-separated clusters and initialization with MTS-clustering, SNN, and SL).

In terms of the EMOC approaches, we verified an issue in the design of the approaches that consider the same clustering criteria in the initialization strategy and the objective functions, in which the evolutionary optimization could not be adequate. For example, in Handl and Knowles (2005c), KM is applied in the initialization along with the pair of objectives (Var and Dev). In this case, the initial population has solutions that either reach the optimal results or exceed the boundaries of feasible search space to find compacted clusters. Therefore, optimization in this direction would not be necessary. Furthermore, these objective functions are very similar in their formulation, which limits the capabilities of the algorithm in generating a diverse set of solutions. MOCLE, beyond other approaches, also presents a similar design, in which every objective function is inadmissible for all the datasets in terms of at least one method used in the initialization.

To demonstrate this impact in terms of ARI, Table 4.5 presents the best ARI results of the partitions generated by each algorithm applied in the initialization of MOCLE (AL, KM, SL and SNN), and MOCLE results. Column Π_0 presents the best ARI in the initial population, and in column MOCLE, the best average ARI and its standard deviation in the results of MOCLE in 30 executions. The "optimization" of the base partition worsened the clustering results, because the use of inadmissible objective functions can move away from the goal. In comparison with the results of the initial population, MOCLE promoted a slight but not significant ARI improvement in 3 datasets. On the other hand, we can observe a significant worsening of ARI in 5 datasets without an improvement in the remaining datasets. In the case where the initial population has a diverse set of partitions, the use of ensemble clustering methods or even selection methods may provide better results than MOCLE. For example, according to the mean ARI (last row in Table 4.5), in the case of a selection method picking the best partitions in Π_0 it could provide a better mean result than MOCLE (best mean ARI found in Π_0 equals to 0.9653 and the MOCLE mean result equals to 0.8459).

It is important to note that, in general, EMOC approaches present in the literature do not use restrictions as defined in Eq. 2.2 (see Section 2). They usually apply the restrictions as objective functions to maintain good solutions found in the initialization or restrict the search in some direction. This case is different from the above scenario, in which the initialization strategy limits the search to all the objective functions used in the multi-objective approach.

These results, presented in Table 4.4, show that for every analyzed initialization, there is no objective function that is admissible in all the datasets. Furthermore, we presented which objective functions have space to be optimized and determined the inadmissible ones, answering our first research question.

G	Dataset	AL	KM	SL	SNN	Π_0	MOCLE	
G1	R15	0.9893	0.9928	0.8955	0.9928	**0.9928**	**0.9928**	7.90E-16
	D31	0.9307	0.9529	0.2124	0.5807	0.9529	**0.9530**	1.93E-04
	Engytime	0.6807	0.8151	0.0000	0.0000	**0.8151**	**0.8151**	0.00E+00
	Sizes5	0.9435	0.9197	0.0307	0.4067	**0.9435**	**0.9435**	3.39E-16
	Square1	0.9501	0.9735	0.0000	0.3285	0.9735	**0.9764**	1.44E-03
	Square4	0.7047	0.8348	0.0000	0.0000	**0.8348**	**0.8348**	5.65E-16
	Twenty	1.0000	1.0000	1.0000	1.0000	**1.0000**	**1.0000**	0.00E+00
	Fourty	1.0000	1.0000	1.0000	1.0000	**1.0000**	**1.0000**	0.00E+00
	Sph_5_2	0.8635	0.8688	0.6949	0.5877	**0.8688**	**0.8688**	5.65E-16
	Sph_6_2	1.0000	1.0000	1.0000	1.0000	**1.0000**	**1.0000**	0.00E+00
	Sph_9_2	0.7731	0.8313	0.0007	0.0001	**0.8313**	**0.8313**	5.65E-16
	Sph_10_2	0.9782	0.9911	0.7968	0.8804	0.9911	**0.9935**	7.48E-03
G2	ds2c2sc13_S1	1.0000	1.0000	1.0000	1.0000	**1.0000**	**1.0000**	0.00E+00
	ds2c2sc13_S2	1.0000	0.8752	1.0000	1.0000	**1.0000**	**1.0000**	0.00E+00
	ds2c2sc13_S3	0.6648	0.6475	0.8724	1.0000	**1.0000**	0.7771	2.26E-16
G3	Long1	0.0152	0.2907	0.9940	1.0000	**1.0000**	**1.0000**	0.00E+00
	Pat2	0.2161	0.2446	1.0000	1.0000	**1.0000**	0.2446	1.41E-16
	Spiral	0.0221	0.0518	1.0000	1.0000	**1.0000**	0.0518	2.12E-17
G4	3MC	1.0000	0.8003	1.0000	1.0000	**1.0000**	**1.0000**	0.00E+00
	DS-850	0.9657	0.9018	0.3927	0.7159	**0.9657**	**0.9657**	1.13E-16
	Aggregation	1.0000	0.7906	0.8089	0.8089	**1.0000**	**1.0000**	0.00E+00
	Complex9	0.4954	0.4921	0.9988	1.0000	**1.0000**	0.5119	1.34E-03
	Pat1	0.0788	0.0684	1.0000	1.0000	**1.0000**	**1.0000**	0.00E+00
	Spiralsquare	0.5410	0.4962	0.9283	0.9971	**0.9971**	0.5410	3.39E-16
	MEAN	**0.7422**	**0.7433**	**0.6927**	**0.7625**	**0.9653**	**0.8459**	

Table 4.5: MOCLE initial population vs. final population. The boldface values denote the best ARI found in Π_0 and generated by MOCLE.

In the following sub-section, we analyzed the clustering results considering the optimization of the selected objective functions, extending our analysis. We picked the objective functions that presented the lowest results of the inadmissibility considering the initialization with MST-clustering.

4.4.1 Analysis of the objective functions in the optimization

Aiming to answer the second research question presented at the beginning of this chapter, we analyzed one initialization strategy, considering different scenarios of the combination of objective functions. In particular, we analyze the behavior of the objective functions in order to improve the detection of no well-separated clusters and close clusters in the heterogeneous data structures in terms of the results presented in Table 4.2. Hence, we selected one objective function per criterion that presented the lowest number of datasets in which they are inadmissible: Var, CH, Sep_{CL}, and Con. Moreover, as described in Section 4.2, Δ-MOCK was chosen because it is a recent approach based on an established algorithm that provides features that allow us to explore the use of MST-clustering in the initialization.

It should be noted that in this section we demonstrate how to perform the analysis of the objective functions while considering a particular EMOC approach and specific goals. Different scenarios, considering other initialization (or even other EMOC algorithms), can lead to different admissibility results and different clustering performance (ARI).

Table 4.6 presents the average ARI and standard deviation of 30 runs for each dataset generated by the Δ-MOCK considering different combinations of the selected objective function. The MST column refers to the best ARI found in the partitions generated by the MST-clustering.

G	Dataset	MST	(Var, Sep_{CL})	(Ch, Sep_{CL})	(Var, CH)	(CH, Con)	(Var, Con)	(Con, Sep_{CL})
G1	R15	0.7203	0.4312 ± 1.58E-02	0.9844 ± 8.49E-03	0.9857 ± 1.60E-03	**0.9928** ± 7.90E-16	0.9914 ± 9.62E-02	0.9892 ± 5.11E-03
	D31	0.4708	0.5515 ± 1.86E-02	0.7358 ± 3.13E-02	**0.7620** ± 1.60E-03	0.4327 ± 8.95E-02	0.7455 ± 2.15E-02	0.7383 ± 1.81E-02
	Engytime	0.0076	0.0369 ± 2.50E-03	0.4945 ± 1.12E-01	0.4849 ± 1.60E-03	0.7638 ± 8.95E-02	0.7847 ± 3.39E-02	**0.8247** ± 2.00E-02
	Sizes5	0.5642	0.0333 ± 1.26E-03	0.3464 ± 1.16E-01	0.3027 ± 6.52E-02	**0.9638** ± 3.27E-03	0.9622 ± 7.63E-03	0.9488 ± 1.35E-02
	Square1	0.3711	0.1315 ± 6.13E-03	0.9457 ± 1.73E-02	0.9504 ± 1.74E-02	**0.9761** ± 3.39E-16	0.9728 ± 4.82E-03	0.9714 ± 6.46E-03
	Square4	0.4570	0.1389 ± 4.60E-03	0.7282 ± 6.85E-02	0.7101 ± 7.78E-02	**0.7847** ± 2.05E-02	0.7739 ± 4.82E-03	0.7678 ± 2.56E-02
	Twenty	**1.0000**	0.4496 ± 4.60E-03	**1.0000** ± 0.00E+00	**1.0000** ± 0.00E+00	**1.0000** ± 0.00E+00	**1.0000** ± 0.00E+00	**1.0000** ± 0.00E+00
	Fourty	**1.0000**	0.6833 ± 1.27E-02	**1.0000** ± 0.00E+00	**1.0000** ± 0.00E+00	**1.0000** ± 0.00E+00	**1.0000** ± 0.00E+00	**1.0000** ± 0.00E+00
	Sph_5_2	0.7315	0.1325 ± 5.45E-03	**0.9239** ± 3.95E-02	0.9072 ± 1.37E-01	0.9205 ± 2.58E-02	0.9019 ± 2.15E-02	0.8932 ± 3.72E-02
	Sph_6_2	**1.0000**	0.1665 ± 6.86E-03	**1.0000** ± 0.00E+00	**1.0000** ± 0.00E+00	**1.0000** ± 0.00E+00	**1.0000** ± 0.00E+00	**1.0000** ± 0.00E+00
	Sph_9_2	0.3057	0.2396 ± 7.92E-03	0.7079 ± 4.70E-02	0.7230 ± 4.00E-02	0.5799 ± 1.48E-01	**0.7228** ± 2.25E-02	0.7110 ± 2.85E-02
	Sph_10_2	0.8651	0.2765 ± 8.36E-03	0.7008 ± 2.69E-01	0.8313 ± 2.06E-01	**0.9857** ± 3.85E-03	0.9743 ± 1.08E-02	0.9732 ± 8.89E-03
G2	ds2c2sc13_S1	**1.0000**	0.0361 ± 1.71E-03	0.0622 ± 7.16E-03	0.0707 ± 5.08E-02	**1.0000** ± 0.00E+00	0.3520 ± 2.82E-16	0.3520 ± 2.82E-16
	ds2c2sc13_S2	**1.0000**	0.1444 ± 6.64E-03	0.2323 ± 2.39E-02	0.2409 ± 3.05E-02	0.8455 ± 1.60E-01	0.9518 ± 5.65E-16	0.9518 ± 5.65E-16
	ds2c2sc13_S3	**0.9952**	0.3043 ± 8.35E-03	0.4517 ± 3.89E-02	0.4623 ± 3.90E-02	0.5672 ± 1.37E-02	0.8713 ± 4.31E-03	0.9139 ± 1.19E-02
G3	Long1	**1.0000**	0.0426 ± 1.89E-03	0.0850 ± 1.28E-02	0.0870 ± 1.25E-02	**1.0000** ± 0.00E+00	**1.0000** ± 0.00E+00	**1.0000** ± 0.00E+00
	Pat2	**1.0000**	0.0334 ± 1.64E-03	0.0638 ± 4.13E-03	0.0642 ± 6.17E-03	**1.0000** ± 0.00E+00	**1.0000** ± 0.00E+00	**1.0000** ± 0.00E+00
	Spiral	**1.0000**	0.7122 ± 9.11E-04	0.7363 ± 2.83E-03	0.7363 ± 2.22E-03	**1.0000** ± 0.00E+00	**1.0000** ± 0.00E+00	**1.0000** ± 0.00E+00
G4	3MC	**1.0000**	0.0764 ± 1.80E-03	0.1268 ± 1.37E-02	0.1263 ± 1.42E-02	**1.0000** ± 0.00E+00	**1.0000** ± 0.00E+00	**1.0000** ± 0.00E+00
	DS-850	0.4503	0.1722 ± 7.00E-03	0.8059 ± 7.75E-02	0.7590 ± 1.61E-01	**1.0000** ± 0.00E+00	0.9994 ± 1.97E-03	0.9990 ± 2.88E-03
	Aggregation	0.8089	0.1395 ± 5.50E-03	0.5008 ± 5.61E-02	0.4812 ± 8.18E-02	0.7722 ± 1.05E-01	**0.9388** ± 1.79E-02	0.9302 ± 2.01E-02
	Complex9	0.9314	0.1308 ± 4.81E-03	0.2186 ± 2.21E-02	0.2151 ± 2.72E-02	0.6262 ± 2.91E-03	0.9660 ± 3.47E-02	**0.9707** ± 3.33E-02
	Pat1	**1.0000**	0.0153 ± 8.23E-04	0.0564 ± 1.23E-02	0.0595 ± 1.90E-02	0.7438 ± 2.26E-16	**1.0000** ± 0.00E+00	**1.0000** ± 0.00E+00
	Spiralsquare	0.9290	0.9292 ± 3.39E-04	0.9978 ± 1.60E-03	**0.9983** ± 1.09E-03	0.7744 ± 9.62E-02	0.9980 ± 1.18E-03	0.9976 ± 1.56E-03
	MEAN	**0.7753**	**0.2503**	**0.5794**	**0.5816**	**0.8637**	**0.9128**	**0.9139**

Table 4.6: Average ARI and standard deviation of different pairs of objective functions in Δ-MOCK

The underlined results point out the objective functions in which the optimization generated solutions that dominate the true partition.

The results point out that, in general, the use of the pairs of objective functions (Var, Sep_{CL}), (Ch, Sep_{CL}), and (Var, CH) does not provide reliable results, because they lose the relation of connectedness in the solutions when it is not applied any restriction. Besides that, these pairs of objective functions dominate the true partition in most of the datasets.

In contrast, the pairs of the objectives (CH, Con), (Sep_{CL}, Con) and (Var, Con) provide the mean ARI of all datasets above 0.85, as shown in the row Mean in Table 4.6. The use of Con as an objective function preserves the continuity property of clusters and restricts the search to providing solutions that correspond to the trade-off between this objective and the other objectives (CH, Var, or Sep_{CL}). The best results are provided by the pairs (Sep_{CL}, Con) and (Var, Con), both with a mean ARI above 0.91.

In general, the results relating to the use of (Sep_{CL}, Con) and (Var, Con) show that the ARI was improved in the no well-separated clusters present in G1 and heterogeneous data structures in G4. Besides that, most of the good solutions found in the initialization were preserved in most datasets. However, we can observe a loss of the ARI for the datasets in G2 when compared with the initial population (MST column). As shown in Table 4.2, the objective functions (Sep_{CL}, Con) or (Var, Con) were inadmissible ($\times$) or obtained the optimal result ($\checkmark$) for the datasets in G2 and G3, thus the optimization is not required. In this context, the results in Table 4.6 confirm our previous results, in which the optimization of these objective functions could only provide a general cost, and it did not afford any improvement in the clustering results. Moreover, the optimization of these objective functions could worsen the clustering performance, as observed in the G2. Furthermore, the optimization of these objective functions has another issue, the domination of the true partitions in most of the datasets in the G1 and G2. In G1, it occurred mainly in the datasets with overlapping clusters. In this case, the size of the neighborhood used in the computation of the Con and the distribution of the points in the boundaries of the overlapping clusters may determine the domination of the true partition. In particular, Con computes the continuity of the data based on the neighborhood; however, an overlapping region might have several nearest neighbors in common, making it difficult to determine which cluster each point in the boundaries belongs to. Regarding G2, as reported in (Kultzak et al., 2021), the optimization of the dataset `ds2c2sc13` in Δ-MOCK can produce several solutions with optimal Con; as a consequence, for these solutions, the decisions around the evolutionary multi-objective optimization will be taken essentially based on the other criteria, in which the true partition is dominated.

In summary, we observe that the initialization strategy should be correlated with the restrictions applied in the EMOC approaches. For example, in Δ-MOCK, the objective function Con takes on this role in order to maintain the high-quality partitions found in the initialization. Furthermore, optimizing some groups of datasets is not required because the initialization provides the optimal result. However, in Δ-MOCK there are no criteria to prevent the "optimizing" of the partitions. This general view demonstrates conditions regarding the choice of the objective functions, and we presented a scenario where optimization is not required, answering our second research question.

4.5 CHAPTER REMARKS

In this chapter we proposed and presented an analysis of the (in)admissibility of clustering criteria in support of defining objective functions in evolutionary multi-objective clustering approaches. Furthermore, we highlighted the importance of aligning the choice of the objective function and

the initialization strategy in designing the EMOC. In general, the use of a traditional clustering algorithm in the initialization provides solutions that reach the boundaries of the search space in terms of some criteria. Thus, optimizing the objective functions that consider such criteria is not required, thus other complementary criteria should be applied in the optimization. In contrast, the criteria applied in the initialization can be taken as "restrictions", to determine the feasible search region. It is important to note that, in general, the EMOC approaches do not use explicit restrictions (see Eq. 2.2 in Section 2). In many cases, the "restrictions" are represented as objective functions without prior notice, which could lead to a mistake regarding the understanding of which objectives are optimized. Thus, our study helps the understanding of the concept of admissibility to support the better choice of the objective functions, considering the different roles that the objective function can perform in the evolutionary multi-objective optimization, answering the **RQ.1** and **RQ.2** (see Chapter 1).

The study and analysis presented in this chapter was published in the journal Information Sciences, in Morimoto et al. (2022a).

In the next chapter, we present metrics applied to measure the relative quality of the base partitions generated by MST-clustering that are applied to design a new EMOC approach. These metrics consider the data proprieties of this initialization strategy and the criteria applied to the objective functions, that were observed in the analysis presented in this chapter.

5 MEASURING THE SEPARATION AND OVERLAPPING OF DATA

In this chapter, we present a new metric to measure separation of data. To our knowledge, there is not an unsupervised metric that measures the separation or the overlapping of the data. Thus, we propose a Data Separation Degree (DSD) that considers the base partitions generated by MST-clustering to determine the data separation.

Furthermore, we present the Constraint-Based Overlap value (CBO) (Adam and Blockeel, 2017), a semi-supervised metric that measures the overlapping of the data. In general, CBO is applied to select algorithms that should be applied according to the data overlap.

Based on the analysis of the admissibility and the ARI results presented in the previous Chapter, we verified that using clustering algorithms in the initialization provides high-quality solutions or is inadmissible in the context of some objective functions. However, the existing EMOC approaches do not have a criterion to define when the optimization should be (or should not be) performed. Both CBO and DSD, are used in our study to deal with this issue, in which we consider the general properties of the MST-clustering to estimate the relative quality of the base partitions generated by this algorithm and define whether the optimization should be performed in EMOC. Here, the relative quality refers to the data proprieties in which the initialization strategy has good (or poor) clustering performance.

5.1 DATA SEPARATION DEGREE

Zahn (1971) introduced the general concept of MST-clustering, presenting it as a method to deal with the problem of detecting inherent separations between subsets (clusters) of a given dataset. Furthermore, Xu and Tian (2015) characterized MST-clustering as capable of detecting clusters of different shapes and sizes. Thus, based on the literature and an analysis of the MST-clustering presented by Handl and Knowles (2007), we assume that it can detect well-separated clusters with arbitrary shapes (heterogeneous nature) but fails in detecting close or overlapping data structures.

Taking into account these characteristics, we developed a metric to estimate the degree of separation of the partitions in the population generated by MST-clustering, denoting the general separation of the data. In our data analysis, we observed that the initial population generated by MST-clustering presents a pattern: the results for the separation index, Sep_{CL}, (Liu et al., 2018) have a high variation between the minimal, mean, and maximal results when the partitions are generated from datasets with overlapping structures, while this variation is near to zero when considering datasets with well-separated data structures.

In other words, we verified that the initial population generated by the MST-clustering can provide information regarding the data separation, which is applied to compute the DSD, a new measure to determine the separation of the data.

5.1.1 Computation of Data Separation Degree

The Data Separation Degree (DSD) considers the variation of the results of Sep_{CL} for the base partitions generated by MST-clustering to define a degree of separation of the data. In our analysis, the relations of minimal ($sep_{CL}\Pi_{min}$), mean ($sep_{CL}\Pi_{mean}$), and maximal ($sep_{CL}\Pi_{max}$) results of Sep_{CL} in the base partitions generated by MST-clustering have low variation (near to zero) when the dataset has well-separated clusters and an inverse relation for overlapping data structures. Thus, the relation between ($sep_{CL}\Pi_{min}/sep_{CL}\Pi_{mean}$) or ($sep_{CL}\Pi_{mean}/sep_{CL}\Pi_{max}$)

or ($sep_{CL}\Pi_{min}/sep_{CL}\Pi_{max}$) is equal to one to well-separated data structures and near to zero in the case of overlapping, providing the degree of separation of the data. For example, `Spiral` dataset (that contains two well-separated clusters with a regular dispersion of the objects into the spiral shape) have the same outcome for these three relations considering two decimals, ($sep_{CL}\Pi_{min}/sep_{CL}\Pi_{mean}$) = ($sep_{CL}\Pi_{mean}/sep_{CL}\Pi_{max}$) = ($sep_{CL}\Pi_{min}/sep_{CL}\Pi_{max}$) = 0.99. However, we also verified an asymmetric distribution in terms of the number of clusters in datasets with arbitrary shaped clusters, presenting different relations between the median and mean values under different skewness, as illustrated in Fig. 5.1. In this case, the relation of results of Sep_{CL} should consider the skewed direction (positive or negative).

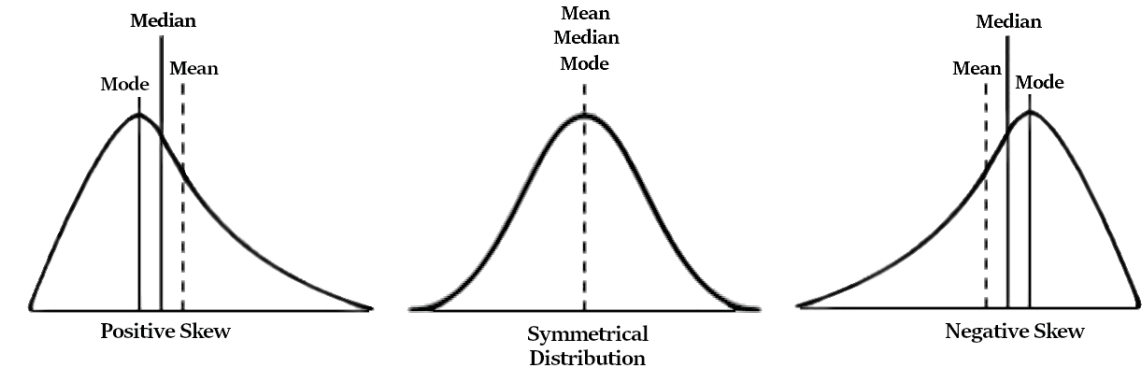


Figure 5.1: Relation between mean and median under different skewness

In Algorithm 1, we present the steps applied to compute the data separation degree. First, we compute the Sep_{CL} and obtain the number of the clusters (k) for each solution in the initial population (lines 1-3). After that, it is obtained with $statistics()$ the maximum, minimum, mode, mean, and median values for Sep_{CL} and k in the initial population (lines 5-6). These data are used to define the DSD according to the skew direction in the number of clusters distribution: ($sep_{CL}\Pi_{min}/sep_{CL}\Pi_{mean}$) for negative skew or ($sep_{CL}\Pi_{mean}/sep_{CL}\Pi_{max}$) for positive skew (lines 7-11). The DSD provides results in the range of 0 and 1, in which 1 indicates well-separated data and 0 high overlapping data. In our computation, we consider a closest integer value of $k\Pi_{mean}$ and $k\Pi_{median}$, obtained by the function $round()$.

5.2 CBO - CONSTRAINT-BASED OVERLAP VALUE

The CBO was proposed by Adam and Blockeel (2017). As a semi-supervised metric, the CBO metric uses some information available about the desired solution. This information takes the form of constraints: must-link (ML) and cannot-link (CL) constraints.

In particular, the CBO considers a short CL and two parallel constraints to measure the degree of overlap of the clusters in a dataset. The short CL, illustrated in Fig. 5.2(a), considers that: if two objects are close (they belong to a defined neighborhood) and have different labels, it indicates an overlap between two clusters, in which ϵ_1 indicates a maximum distance between the object $\mathbf{x}_1$ and the k-nearest neighbor in a particular neighborhood. In terms of the two parallel constraints, illustrated in Fig. 5.2(b), it considers two pairs of objects, in which the objects of each pair are close (they belong to a defined neighborhood); in this case if it is observed a ML and CL relationship between the objects of each pair, it also implies an overlap region.

Algorithm 1 Data Separation Degree

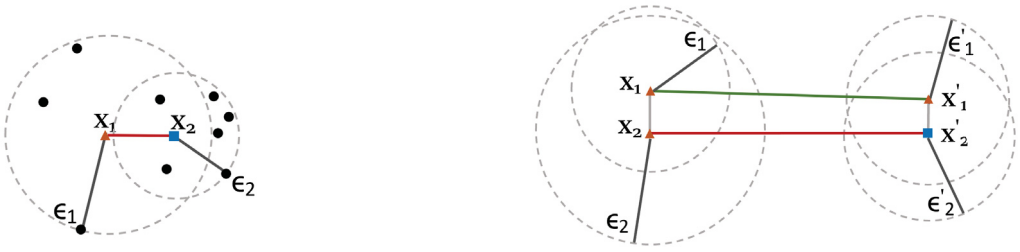
Input: Π_0
Output: DSD

for each $\pi_i \in \Pi_0$ **do**
  $sepv[i] \leftarrow Sep_{CL}(\pi_i)$
  $kv[i] \leftarrow K(\pi_i)$
end for
$k\Pi_{max}, k\Pi_{min}, k\Pi_{mean}, k\Pi_{mode}, k\Pi_{median} \leftarrow$ statistics(kv)
$sep_{CL}\Pi_{max}, sep_{CL}\Pi_{min}, sep_{CL}\Pi_{mean} \leftarrow$ statistics$(sepv)$
if $(k\Pi_{mode} >$round$(k\Pi_{mean}))$ and (round$(k\Pi_{median}) \geq$round$(k\Pi_{mean}))$ **then**
  $DSD \leftarrow sep_{CL}\Pi_{mean}/sep_{CL}\Pi_{max}$
else
  $DSD \leftarrow sep_{CL}\Pi_{min}/sep_{CL}\Pi_{mean}$
end if
return DSD

(a) Short cannot-link pattern

(b) Parallel and close must-link and cannot-link pattern

Figure 5.2: Overlapping patterns of constraints. Red links denote CL and green links denote ML patterns, in which the objects of the same color belong to the same cluster.



(a) Score of a single constraint for L=6

(b) Score of a pair of constraints

Figure 5.3: An illustration of the relationship between two objects and the more distant objects in their neighborhoods with L = 6. The dashed circles point out the objects included in the neighborhoods. The black points denote the objects whose labels or patterns are unknown.

The relationship of short (close) link between the objects consider the distance, $d(,)$, between the objects. As above-mentioned the CBO considers a defined neighborhood size (L), a user-parameter, to compute a score that indicates the relationship of closeness between the objects. Eq. 5.1 presents the score considering two close objects.

$$score(s) = \begin{cases} 1 - \frac{d(\mathbf{x}_1,\mathbf{x}_2)}{max(\epsilon_1,\epsilon_2)}, \text{if } d(\mathbf{x}_1,\mathbf{x}_2) \leq max(\epsilon_1,\epsilon_2), \\ 0, \text{otherwise} \end{cases} \tag{5.1}$$

According with the authors, we can assume that $d(\mathbf{x}_1,\mathbf{x}_2) + d(\mathbf{x}'_1,\mathbf{x}'_2) \leq d(\mathbf{x}_1,\mathbf{x}'_2) + d(\mathbf{x}'_1,\mathbf{x}_2)$ without loss of generality. Thus, the parallel relationship can be scored according to Eq. 5.2.

$$score(p) = score(s_1) \cdot score(s_2) \tag{5.2}$$

In both scores, higher scores indicate an overlap of clusters. The CBO aggregates these scores in Eq. 5.3. This metric compares the number of short CL constraints, direct (single pattern) or by propagation (double pattern), to the total number of constraints, both ML and CL.

$$CBO = \frac{\sum_{s \in CL} score(s) + \sum_{\substack{s_1 \in CL, \\ s_2 \in ML}} score(p)}{\sum_{s \in CL \cup ML} score(s) + \sum_{\substack{s_1 \in ML, \\ s_2 \in CL \cup ML}} score(p)} \tag{5.3}$$

CBO takes the interval between 0 and 1, in which values equal to zero indicate well-separated data structures, while results near to 1 indicate total overlap.

5.3 EXPERIMENTAL DESIGN

To evaluate DSD we considered two experiments. The first one compares DSD with traditional metrics applied to measure the relative quality of clusters, such as PBM, $Dunn$ and DB. Also, we compare our results with a recent published measure of clustering quality, AUCC - Area Under the Curve for Clustering presented by Jaskowiak et al. (2022). The AUCC explores the features of AUC/ROC - Area under the curve/Receiver Operating Characteristics (Spackman, 1989), a performance measure usually applied in the supervised learning domain to the unsupervised domain. These experiments were performed to demonstrate that general clustering metrics do not provide the required information to support a configuration or determine whether a dataset should be optimized or not. In the second experiment, we compare the results of DSD and CBO with ARI, aiming to demonstrate the general features of these metrics.

Considering that MST-clustering (Handl and Knowles, 2007) has a random choice of the neighbor node to link the nodes to other ones when the interesting link is removed, and the initial node applied in the construction of the MST can also be chosen by random, the initial population can have a slight variation in the number of clusters with different random seeds, that can affect the results of the DSD. In order to amend this matter and obtain a consistent result for DSD, in our experiments, we use the average DSD results of 10 initial populations.

Regarding the CBO setting, we applied the original neighborhood size ($L_O = 10 + n/20$) provided by Adam and Blockeel (2017), and three others: ($L_{25} = \sqrt{n} \cdot 25\%$), ($L_{50} = \sqrt{n} \cdot 50\%$) and ($L_{75} = \sqrt{n} \cdot 75\%$), in which n is the number of objects in the dataset. Furthermore, similar to Adam and Blockeel (2017), we used 20 known short link patterns that were applied to obtain the parallel patterns (totaling 40 objects), i.e., we considered about 200 constraints (close and parallel must-link and cannot-link patterns), applied to compute both $score(s)$ and $score(p)$. The points applied to define the short link patterns are selected at random, thus the results provided consider the average of 30 runs.

5.3.1 Datasets

In terms of the datasets, we used the same ones from the previous experiments, presented in Section 4.2. For the purpose of facilitating the visualization of the correlated metrics, we ordered the datasets considering the ARI results, in which we do not use the groups applied in Section 4.2.

5.3.2 Performance assessment

In general, the correlations between relative and external criteria are given by the Pearson correlation coefficient (Jaskowiak et al., 2022). Thus, we consider this coefficient to evaluate DSD and compare it with other metrics.

5.4 RESULTS

Table 5.1 presents the results of DSD, DB, $Dunn$, PBM, AUCC and their correlation with ARI. The datasets in this table and the next one are not grouped by the data structures as presented in the previous chapter. In this table, we can observe that the results of DSD have the highest absolute correlation with ARI. Besides that, it is important to note that DB, $Dunn$ and PBM were computed considering the best partitions in the initialization, that requires a selection of the best partition in the base partitions to make it possible to use these metrics. In contrast, DSD measures the quality in terms of the initial population generated by MST-clustering.

Datasets	**ARI**	**DSD**	**DB**	**DUNN**	**PBM**	**AUCC**
3MC	1.0000	0.6807	0.7076	0.1613	49.0	0.9262
Fourty	1.0000	0.9539	0.3306	0.4943	167.5	1.0000
Long1	1.0000	0.8720	1.5709	0.0647	0.4	0.7036
Pat1	1.0000	0.2696	2.3157	0.0358	53556.2	0.4450
Pat2	1.0000	0.6709	1.0907	0.0723	220488.0	0.6991
Sph_6_2	1.0000	0.7717	0.3555	0.5150	626.4	0.9995
Spiral	1.0000	0.9997	4.5302	0.1424	0.8	0.5204
Twenty	1.0000	0.9291	0.3236	0.3679	122.9	1.0000
ds2c2sc13_S1	1.0000	0.9015	0.4900	0.4592	0.2	0.9895
ds2c2sc13_S2	1.0000	0.9305	0.7258	0.1520	0.2	0.9740
ds2c2sc13_S3	0.9951	0.9461	1.6246	0.0451	0.1	0.9303
Complex9	0.9361	0.8218	1.8153	0.0351	20961.5	0.8364
Spiralsquare	0.9287	0.6872	1.7825	0.0368	33.5	0.8320
Sph_10_2	0.8588	0.5547	0.7624	0.1025	177.4	0.9872
Aggregation	0.8089	0.8239	0.6249	0.1078	195.9	0.9441
R15	0.7275	0.6049	0.7718	0.0349	24.7	0.9713
Sph_5_2	0.7127	0.4553	0.8807	0.0931	10.9	0.9309
Sizes5	0.5032	0.0861	0.8980	0.0121	16.4	0.8078
Square4	0.4694	0.2579	1.0345	0.0149	14.2	0.9468
D31	0.4568	0.4170	1.3983	0.0166	13.9	0.8049
DS-850	0.4505	0.5462	0.9474	0.0235	2.1	0.7068
Square1	0.3797	0.2681	0.8470	0.0192	34.9	0.7713
Sph_9_2	0.3056	0.1015	1.5381	0.0241	1.1	0.7869
Engytime	0.0076	0.1552	1.2201	0.0062	3.6	0.5721
Person Correlation		**0.8062**	**0.0815**	**0.5058**	**0.2147**	**0.2569**

Table 5.1: Best ARI found in the MST-clustering and CVIs relationship (average results of 10 populations generated by MST-clustering)

Table 5.2 presents the results for CBO and DSD. In CBO, the negative correlation refers to the inverse relation with ARI, in which the well-separated data is denoted with results near to

zero, while the best results of ARI are near to one. The result on Table 5.2, shows that DSD has the highest absolute correlation with ARI. However, in the datasets with an ARI near to 1, some CBO results provided better correlation. In particular, in the case of the datasets with an ARI greater than 0.99, CBO has an absolute correlation of 1 for L_{25} and 0.90 for L_{50}, while the DSD has a correlation of 0.21.

Datasets	**ARI**	**CBO**				**DSD**
		L_O	$L_{25\%}$	$L_{50\%}$	$L_{75\%}$	
3MC	1.0000	0.0000	0.0000	0.0000	0.0000	0.6807
Fourty	1.0000	0.2440	0.0000	0.0000	0.0016	0.9539
Long1	1.0000	0.0000	0.0000	0.0000	0.0000	0.8720
Pat1	1.0000	0.0353	0.0000	0.0000	0.0039	0.2696
Pat2	1.0000	0.0846	0.0000	0.0013	0.0164	0.6709
Sph_6_2	1.0000	0.0000	0.0000	0.0000	0.0000	0.7717
Spiral	1.0000	0.0507	0.0000	0.0000	0.0002	0.9997
Twenty	1.0000	0.0263	0.0000	0.0000	0.0000	0.9291
ds2c2sc13_S1	1.0000	0.0000	0.0000	0.0000	0.0000	0.9015
ds2c2sc13_S2	1.0000	0.0023	0.0000	0.0000	0.0000	0.9305
ds2c2sc13_S3	0.9951	0.1257	0.0004	0.0030	0.0164	0.9461
Complex9	0.9361	0.0627	0.0000	0.0065	0.0022	0.8218
Spiralsquare	0.9287	0.1383	0.0009	0.0027	0.0028	0.6872
Sph_10_2	0.8588	0.0359	0.0015	0.0131	0.0156	0.5547
Aggregation	0.8089	0.0221	0.0094	0.0046	0.0031	0.8239
R15	0.7275	0.0065	0.0116	0.0021	0.0134	0.6049
Sph_5_2	0.7127	0.0816	0.0072	0.0227	0.0282	0.4553
Sizes5	0.5032	0.0175	0.0126	0.0031	0.0122	0.0861
Square4	0.4568	0.0943	0.0656	0.0848	0.0954	0.1096
D31	0.4694	0.2194	0.0183	0.0410	0.0350	0.4170
DS-850	0.4505	0.0148	0.0019	0.0034	0.0021	0.5462
Square1	0.3797	0.0160	0.0104	0.0140	0.0172	0.2681
Sph_9_2	0.3056	0.1335	0.0992	0.1038	0.1003	0.1015
Engytime	0.0076	0.0618	0.0468	0.0561	0.0451	0.1552
PersonCorrelation(PC)		**-0.1357**	**-0.6995**	**-0.7062**	**-0.6521**	**0.8062**
PCfordatasetswithARI≥0.99		**-0.3233**	**-1.0000**	**-0.9091**	**-0.6583**	**-0.2111**
PCfordatasetswithARI<0.99		**-0.0828**	**-0.6018**	**-0.5848**	**-0.5260**	**0.8056**

Table 5.2: Best ARI found in the MST-clustering and the relation between CBO and DSD

5.5 CHAPTER REMARKS

In this section, we introduce a new metric, DSD, to measure the separation of the data in the clusters present in the base partitions generated by MST-clustering. Also, we present the main features of the CBO (Adam and Blockeel, 2017), an existing semi-supervised metric applied to measure the data overlapping.

These metrics are analyzed in order to verify their potential in defining the relative quality of the solutions generated by MST-clustering. Our experiments demonstrated that both DSD and CBO have a higher correlation with ARI in comparison with other metrics.

In the next chapter, we present a new EMOC approach that considers both DSD and CBO to support an adaptive parameter setting and choice of objective functions. As discussed in the previous section, both CBO and DSD have features to determine the relative quality of the base partitions generated by MST-clustering, supporting the definition of a new EMOC.

6 PROPOSED MULTI-OBJECTIVE CLUSTERING APPROACH

In this chapter, we present the proposed EMOC approach, AEMOC - Adaptive evolutionary multi-objective clustering approach based on data proprieties. We consider the findings described in Chapters 4 and 5 to improve the design of the multi-objective clustering approach in comparison to existing approaches.

Fig. 6.1 presents the general architecture of the proposed approach that is composed of 4 modules: Initialization, Evaluation, Configuration, and Optimization. The main difference between this approach and the EMOC approaches present in the literature is the introduction of the Evaluation and Configuration modules. The Evaluation Module verifies whether the initialization strategy could provide optimal results in terms of the evaluated criteria. In particular, the proposed evaluation method measures the separation and overlapping of the data to analyze the potential of MST-clustering in detecting the clusters. Based on these data properties, the AEMOC strategy consists of deciding whether an optimization step should be performed. In the case where optimization is applied, the configuration module determines the parameter setting of the multi-objective optimizer according to the results of the evaluation module. In the following, we present details of each module.

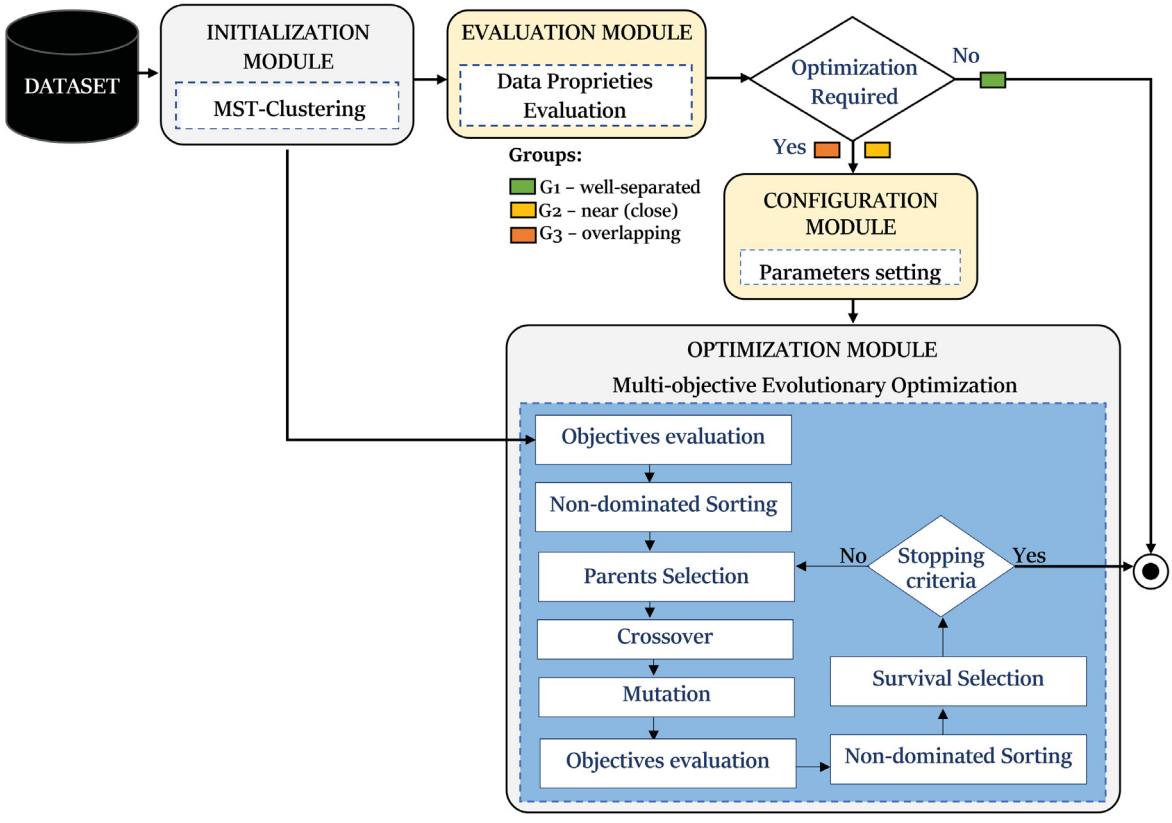


Figure 6.1: Proposed approach: AEMOC

6.1 INITIALIZATION MODULE

According to the admissibility analysis (Chapter 4), the initialization strategy and the choice of the objective functions should be complementary in terms of the clustering criteria adopted. We

verified the potential of using MST-clustering, which considers a connectedness criterion to apply other kinds of clustering criteria (such as compactness). Thus, the choice of the representation of the solutions also follows this initialization strategy, in which we applied the LAG encoding (locus) proposed by Handl et al. (2007). This representation considers the general structure of the MST to denote the partitions, in which each node represents a locus (variable) manipulated by the evolutionary operators.

Another well-known representation that makes use of the MST data structure is the reduced LAG (called Δ-locus) introduced by Garza-Fabre et al. (2018). However, we do not make use of this representation because additional experiments presented in Appendix B demonstrated that Δ-locus (Garza-Fabre et al., 2018) has lower clustering performance in diverse datasets in comparison with locus representation.

6.2 EVALUATION MODULE

The evaluation module analyzes the capabilities of the MST-clustering in detecting the clusters based on separation and/or overlapping of the data. In other words, we measure the potential quality (relative quality) of the base partitions in order to define whether the optimization should (or should not) be performed. Furthermore, we use the outcome of this module to determine the parameter setting of the optimization module.

In general, the evaluation method measures the data properties (or attributes) that contribute (or obstruct) to the initialization strategy in detecting the clusters. As above mentioned, MST-clustering can detect well-separated clusters, but has difficulties detecting near and overlapping clusters; therefore, in this case, this module evaluates the relation of the separation and overlapping of the data that is applied to define whether the initialization strategy can detect the clusters in a dataset.

In particular, in our approach, we combined CBO and DSD, aiming to inherit their strong points in defining separation and overlapping of the data, in which well-separated clusters are indicated by CBO results near zero and DSD results near one, and the opposite relationship denotes overlapping clusters.

In our method, we consider three groups related to the data properties to measure the potential of the MST-clustering in detecting the cluster: G1, which considers well-separated data and has a high potential for MST-clustering to detect clusters; G2, which denotes near (close) data and has a middle potential for MST-clustering to detect clusters; and, G3, which refers to overlapping data, where, in general MST-clustering fails in detecting clusters.

To determine these different groups, we analyzed the following datasets[1] to define the range of CBO and DSD results: `Long`, `Spiral`, `Twenty`, `Complex9`, `R15`, `Sph_5_2`, `Square4`, `D31`, and `Sph_9_2`. These datasets were selected because they present different data structures with distinct data separation (or overlap), making it possible to define the ranges of each group.

In Table 6.1, we present the CBO and DSD results for each dataset. Regarding CBO, we show the results considering three different sizes of neighborhood: ($L_{25} = \sqrt{n} \cdot 25\%$), ($L_{50} = \sqrt{n} \cdot 50\%$) and ($L_{75} = \sqrt{n} \cdot 75\%$), where n is the number of objects in the dataset.

It is important to note that the range of well-separated data to close data is very short in the CBO. CBO measures the degree of overlap of the data according to the intersection of the objects between different clusters based on the neighborhood. Moreover, in the case of a few objects being close or in the intersection, the CBO presents a result near zero, but it is still

[1]Datasets repository: `https://github.com/deric/clustering-benchmark`

different from zero. Therefore, in our evaluation, we also consider this aspect in the definition of the range of the groups.

Datasets	CBO			DSD
	$L_{25\%}$	$L_{50\%}$	$L_{75\%}$	
Long1	0.0000	0.0000	0.0000	0.8720
Spiral	0.0000	0.0000	0.0002	0.9997
Twenty	0.0000	0.0000	0.0000	0.9291
Complex9	0.0000	0.0065	0.0022	0.8218
R15	0.0065	0.0021	0.0134	0.6049
Sph_5_2	0.0072	0.0227	0.0282	0.4553
D31	0.0183	0.0410	0.0350	0.4170
Square4	0.0656	0.0848	0.0954	0.1096
Sph_9_2	0.0992	0.1038	0.1003	0.1015

Table 6.1: Best ARI found in the MST-clustering and the relation between CBO and DSD

By analyzing the results of CBO and DSD for the datasets with well-separated clusters (Long, Spiral, Twenty), we observe that the CBO presents results between 0 and 0.0002, and the DSD results between 0.87 and 0.99. Based on these results, we defined a range of CBO and DSD to describe datasets in G1: $[0.00, 0.001[$ for CBO and $[1.0, 0.85]$ for DSD. In terms of the datasets with data overlap (Square4, D31, and Sph_9_2), we verified results above 0.018 for CBO and below 0.417 for DSD; therefore, we determine the following ranges for the dataset in G3: $[0.015, 1.00]$ for CBO and $]0.45, 0.0]$ for DSD. Finally, to define the datasets in G2, we consider a range between G1 and G3: $[0.001, 0.015[$ for CBO and $]0.85, 0.45]$ for DSD.

Considering the three groups of data properties, we introduce an evaluation method in Fig. 6.2, that takes into account the strengths of each metric (DSD and CBO). At first, this method evaluates the CBO in order to determine the groups (G1, G2, and G3).

As can be seen in Table 6.1 the CBO results of the datasets Complex9 and Sph_5_2, are highly dependent on the size of the neighborhood, which can lead to mistakes in the boundaries of the groups. Therefore, the merit of this metric is evaluated by considering different sizes for this item: L_{25}, L_{50}, and L_{75}. Only when the CBO provides a consensus result for the various L are the groups determined by this metric. In the case where this metric presents a weak indication of the group (no consensus), the DSD is applied instead of the CBO.

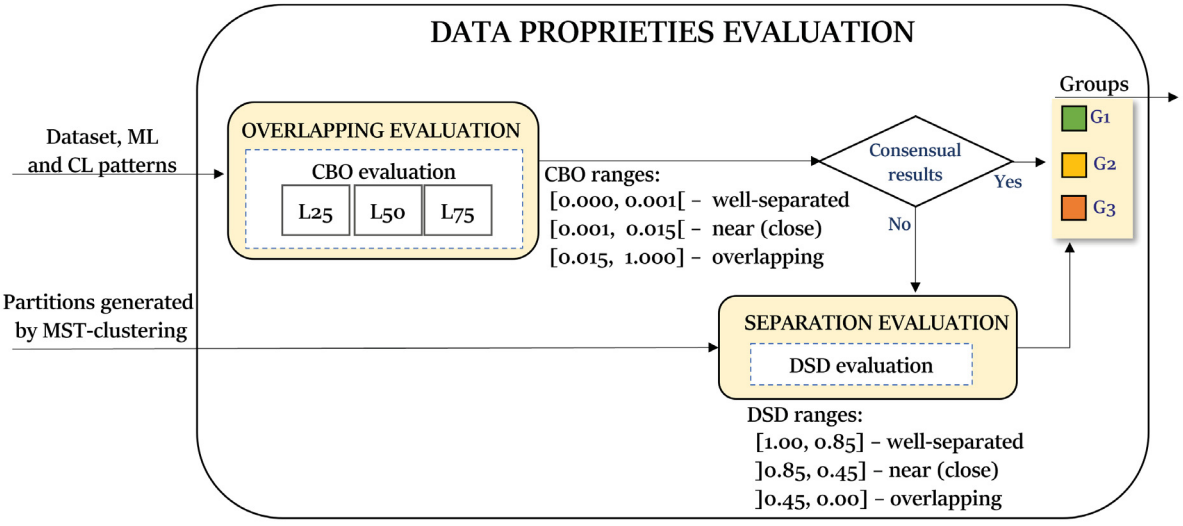


Figure 6.2: Quality evaluation.

6.3 CONFIGURATION AND OPTIMIZATION MODULES

The configuration module is used to define the general settings of the optimizer, including the selection of the objective functions according to the groups defined in the evaluation method.

In terms of the MOEA, we used the NSGA-II in the Optimization Module. This is a well-known MOEA that has been shown to be effective in the clustering problem. In the NSGA-II, we considered the uniform crossover and neighborhood mutation (Handl et al., 2007) with two different configurations for the number of iterations: 100 iterations (the same general setting applied to Δ-MOCK (Garza-Fabre et al., 2017)) to optimize the base partitions related to the datasets classified in G2; (ii) 250 iterations (the same used in EMO-KC (Wang et al., 2018)) to optimize base partitions related to G3.

With respect to the objective functions, we defined the use of one connectedness associated with a compactness criterion or separation criterion. The connectedness is applied to restrict the search space and support the maintenance of the continuous structures provided in the initialization. The compactness (or separation) criterion is a complementary clustering criterion to the ones applied in the initialization, that will guide the search.

Our approach considers Var as a compactness criterion, Sep_{CL} as a separation criterion, and a modified connectivity index, Con' as a connectedness criterion. In particular, for the datasets with near data (clusters), classified in G2, we consider that Var can provide good results, in line with the general results of other approaches that use this criterion in the optimization.

In terms of the dataset that presents overlapping data, classified in G3, we observe that Sep_{CL} could provide better results than Var, being more promising for this kind of structure (see experiments in Appendix B.2). Thus, we applied the Sep_{CL} as a separation criterion associated with the Con' as an objective function for optimizing this group of data. The Con' refers to an improved Con (Handl et al., 2007) that we introduced in Morimoto et al. (2022b), which is detailed in the following.

6.3.1 An Improved Connectivity Index

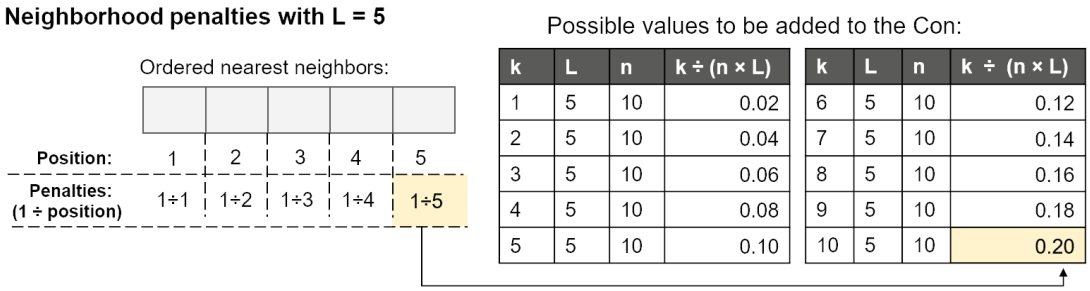
By observation, we verified that Con does not distinguish solutions with different numbers of clusters. Thus, different solutions could have the same outcome for this metric. For example, in different solutions with optimal connectivity ($Con = 0$), the decision is taken by the other objective function. In the case of a compactness criterion, such as Var, only the solution with a lower Var (in general, the solution with the highest number of clusters) will be selected to compose the next generation.

Since we aim to improve the general clustering performance, including finding solutions with different granularities, we propose in Eq. (6.1) a slight but effective modification of the definition of the Con (see Eq. A.10 Appendix A):

$$Con'(\pi) = Con(\pi) + \left(\frac{k}{n \cdot L}\right) \tag{6.1}$$

where k is the number of clusters in partition π, n is the number of objects in the dataset, and L is the number of nearest neighbors that contribute to connectivity. This modification takes k as a secondary criterion of connectivity that differentiates solutions with the same outcome for Con but a different number of clusters, which can be found in nested clusters or hierarchical data structures. The term $(n \cdot L)$ ensures that the number of clusters k will be mapped to a value lower or equal to $\frac{1}{L}$, resulting in values in the interval $]0, \frac{1}{L}]$. This is required to maintain the ordinal relationship between the best and the worst connectivity results. Thus, this modification will only affect solutions that have the same outcome for Con.

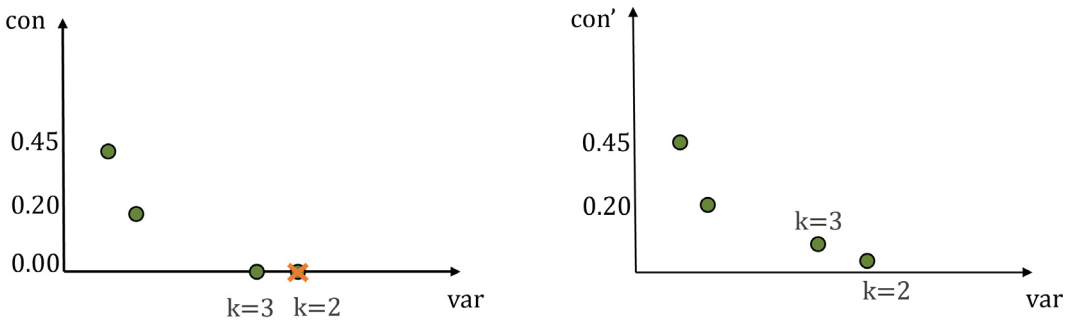
Figure 6.3 illustrates how the term $(n \cdot L)$ maintains the ordinal relationship when k is added to Con. In this example, we consider $n = 10$, and $L = 5$. In this figure, we can observe that the penalties take intervals based on their position in the neighborhood (the penalty is equal to 1 divided by the neighborhood position of the evaluated object). Thus, by dividing k by the term $(n \cdot L)$, the k is mapped to the interval $]0, \frac{1}{L}]$, in which the values summed to Con do not overtake the interval of the penalties of the connectedness. In this example, the maximum value of the mapped k is 0.20 (or $1 \div 5$). This specific case occurs when we have each object standing for an individual cluster, in which we have the maximum penalties of the connectedness for the partition, and the addition of any value will not affect its evaluation. The general effect of not using the term $(n \cdot L)$ can be observed in the following example: considering $Con'' = Con + k$, a partition π_A with $Con = 3$ and $k = 6$, and a partition π_B with $Con = 4$ and $k = 5$, for both partitions, the result of the Con'' is 9. In contrast, if we consider the Con' and the data in Figure 6.3, it is obtained two different results: $Con' = 3.12$ for π_A and $Con' = 4.1$ for π_B , where the information the connectivity is maintained, and new information about the number of clusters is associated.



k	L	n	k ÷ (n × L)	k	L	n	k ÷ (n × L)
1	5	10	0.02	6	5	10	0.12
2	5	10	0.04	7	5	10	0.14
3	5	10	0.06	8	5	10	0.16
4	5	10	0.08	9	5	10	0.18
5	5	10	0.10	10	5	10	0.20

Figure 6.3: Example of the relation of $k/(n. \cdot L)$ and the interval of neighborhood penalties

Fig. 6.4 illustrates the general effect of Con' in the selection considering a Pareto front of (Var, Con'). Fig. 6.4(a) illustrates four solutions, in which the solution with the highest outcome for Var is discarded in the selection. This solution has the optimal connectivity result ($Con = 0$); however, other solutions with a lower Var (and a higher number of clusters) dominates it. By using Con', we create a differentiation of the solutions with same outcome for Con and different number of clusters, thus the solutions with this kind of relation are maintained, Fig. 6.4(b).



(a) Pareto Front of (Con, Var)

(b) Pareto Front of (Con′, Var)

Figure 6.4: An example of the effect of the new connectivity in the Pareto Front

It is important to note that Con' only uses the k to support the decision as a secondary criterion, in which case the automatic k-determination prevails instead of the the multi-k-clustering

(see Section 3.2.9). In other words, the solutions with a different number of clusters are naturally obtained in the optimization of different clustering criteria, different from the approaches that use k, as restrictions in the form of objective functions, to optimize clustering criteria to multiple k.

6.4 CHAPTER REMARKS

In this chapter, we present a new multi-objective clustering approach, AEMOC—Adaptive Evolutionary Multi-Objective Clustering Approach based on data proprieties. The main feature that differ this approach from the others is the use of the information of the data proprieties in the initialization in order to adapt the parameter setting and the choose of the objective function. For that, we introduced an evaluation module that was designed with the data proprieties and features of MST-clustering in mind to define the relative quality of base partitions and devise the optimization strategy.

Besides that, we introduce an improved connectivity index, Con', applied as an objective function in our approach. This index was designed to improve the selection in the Pareto front and improve the clustering in datasets with sub-clusters (sub-sets), as presented in nested clusters, when applied with Var or Sep_{CL}.

In Morimoto et al. (2022b), we demonstrated that Con' can improve the clustering in different EMOC approaches, in particular for nested clusters, among other hierarchical data structures. This paper is presented in Appendix C.

In the next chapter, we present the results of the experiments that demonstrate the potential of the evaluation method to define the relative quality of the base partitions. Furthermore, we compare the results of AEMOC with other established approaches.

7 EXPERIMENTS

In this chapter, we evaluate the new approach. We present two different experiments: the first one was applied to assess the evaluation method and the DSD as relative quality evaluators, followed by experiments considering the comparison of AEMOC and other approaches: MOCK, MOCLE, Δ-MOCK and EMO-KC. These approaches provide different strategies and proprieties that allow us to analyze different aspects of multi-objective clustering.

7.1 EXPERIMENTAL DESIGN

In this section, we present the experimental setup applied to each approach considered in our analysis. Furthermore, we present the datasets and performance assessment used in the evaluation of the results.

7.1.1 Experimental setup

In terms of the AEMOC, we applied the general setting presented in Chapter 6. For MOCK, we used the settings reported by Handl et al. (2007). Regarding Δ-MOCK, we applied the general parameter setting presented by Garza-Fabre et al. (2017). In terms of the representation of the Δ-MOCK, we applied the Δ-locus scheme with δ defined as a function of $\sim 5/\sqrt{n}$, where n is the number of objects in the dataset — this function is one of the heuristics employed in Garza-Fabre et al. (2017). Concerning the MOCLE, we used the general setting as in Faceli et al. (2006), considering the NSGA-II as MOEA and HBGF as the crossover operator. For EMO-KC, we applied the same general setting presented in Wang et al. (2018). Furthermore, we applied the Euclidean distance as a distance function, and we adjusted the other parameters required to produce partitions containing clusters in the range $\{2, 2k^*\}$. Finally, as such algorithms are non-deterministic, we executed the experiments 30 times.

We summarized the main components of each EMOC in Table 7.1, in which we applied an acronyms: NB to denote the neighborhood-based mutation; L refers to the neighborhood-size applied in Con, that also is applied in the initialization and mutation operator of MOCK, Δ-MOCK and AEMOC; and n the number of objects in the dataset.

FEATURES	MOCK	MOCLE	Δ-MOCK	EMO-KC	AEMOC
Initialization	MST and KM	AL, KM, SL, SNN	MST	Random	MST
Encoding	Locus	Label	Δ-locus	Centroid	Locus
MOEA	PESA-II	NSGA-II	NSGA-II	NSGA-II	NSGA-II
Crossover	Uniform	HBGF	Uniform	SBX	Uniform
Mutation	NB	-	NB	Polynomial	NB
N. Generations	1000	50	100	250	100 or 250
Objective Functions	(Dev, Con)	(Dev, Con)	$(Var$, Con$)$	(Var', k)	(Var, Con') or (Sep_{CL}, Con')
L	10	5% of n	10	-	10

Table 7.1: Parameters and configuration of MOCK, MOCLE, Δ-MOCK and AEMOC

7.1.2 Datasets

In the experiments, we applied a diverse set of datasets that have different data structures and cluster sizes. Table 4.1 presents the main characteristics of these datasets, considering the number of objects n, the number of clusters k^* in the true partition and the number of dimensions dim. These datasets are divided into 4 groups (D). These datasets were obtained from the same repositories listed in Section 4.2. In D1, D2 and D3, we grouped the artificial datasets. In D1, we gathered datasets with well-separated clusters. In D2, we included datasets that present heterogeneous data structures and/or datasets with close clusters. In D3, we have datasets with overlapping data or/and close clusters and a high spread of data. Finally, in D4, we have the real-life datasets.

D	Dataset	n	dim	k^*
D1	3MC	400	2	3
	Pat1	557	2	3
	Pat2	417	2	2
	Fourty	1.000	2	40
	Sph_6_2	300	2	6
	ds2c2sc13_S1	588	2	2
	ds2c2sc13_S2	588	2	5
	ds2c2sc13_S3	588	2	13
D2	Aggregation	788	2	7
	Complex8	2551	2	8
	Spiralsquare_S1	1.500	2	2
	Spiralsquare_S2	1.500	2	6
	2d_10c_no9	3580	2	10
	2d_4c_no2	1064	2	4
	Sph_10_2	500	2	10
	Sizes5	1.000	2	4
D3	ds4c2sc8	485	2	8
	DS-850	850	2	5
	Flame	240	2	2
	Patbased	300	2	3
	Engytime	4.096	2	2
	Square1	1.000	2	4
	Triangle2	1000	2	4
	Twodiamonds	800	2	2
D4	Glass	214	9	2
	Iris	150	4	3
	Libra	360	90	15
	Optdigits	5620	62	10
	Thyroid	215	5	3
	Soybeans	47	35	4
	Wine	178	13	2
	Zoo	101	17	7

Table 7.2: Datasets Information - dataset applied to analyze the performance of the proposed EMOC approach

7.1.3 Performance assessment

We used the ARI, Eq. 2.3, to evaluate the clustering performance and the definition of the groups with regard to the data properties. In the evaluation of the groups, we verified the correlation of the data properties along with the potential quality of the base partitions generated by MST-clustering. We consider 3 ranges of the ARI: $[1.00, 0.95]$, $]0.95, 0.50]$, and $]0.50, 0]$, that are applied to evaluate G1, G2, and G3, respectively.

The ARI is one of the most popular clustering validity indexes applied to evaluate EMOC approaches. Most of the approaches presented in Section 3 make use of this index to evaluate the clustering results.

Furthermore, we use a non-parametric test to analyze the ARI results, the Friedman and Bergmann-Hommel Post Hoc hypothesis test (Pohlert, 2018) with alpha=0.05. This test is applied to compare the overall performance of the algorithms.

7.2 RESULTS OF THE EVALUATOR MODULE

Table 7.3 presents the results of the Evaluator module, in which, in the last two columns, the cells marked with ✓ or × denote whether CBO or DSD was taken in the definition of the relative quality of the partitions, according to the method described in Section 6.2. In particular, ✓

denotes the groups (G1, G2, G3) correctly assigned, and × indicates that the evaluation module defined a different class than the expected one. Furthermore, in this table, we present the best ARI found in the initial population and the individual results of CBO and DSD.

D	Dataset	ARI	Results				Q. Groups	
			CBO L_{25}	CBO L_{50}	CBO L_{75}	DSD	CBO	DSD
D1	3MC	1.0000	0.0000	0.0000	0.0000	0.6807	✓	
	Pat1	1.0000	0.0000	0.0000	0.0039	0.2696		×
	Pat2	1.0000	0.0000	0.0013	0.0164	0.6709		×
	Fourty	1.0000	0.0000	0.0000	0.0016	0.9539		✓
	Sph_6_2	1.0000	0.0000	0.0000	0.0000	0.7717	✓	
	ds2c2sc13_S1	1.0000	0.0000	0.0000	0.0000	0.9015	✓	
	ds2c2sc13_S2	1.0000	0.0000	0.0000	0.0000	0.9305	✓	
	ds2c2sc13_S3	0.9951	0.0004	0.0030	0.0164	0.9461		✓
D2	Aggregation	0.8089	0.0094	0.0046	0.0031	0.8239	✓	
	Complex8	0.9361	0.0008	0.0102	0.0142	0.7836		✓
	Spiralsquare_S1	0.5711	0.0000	0.0000	0.0000	0.6726	×	
	Spiralsquare_S2	0.9287	0.0009	0.0027	0.0028	0.6872		✓
	2d_10c_no9	0.5685	0.0028	0.0083	0.0052	0.7373	✓	
	2d_4c_no2	0.7586	0.0035	0.0023	0.0045	0.7643	✓	
	Sph_10_2	0.8588	0.0015	0.0131	0.0156	0.5547		✓
	Sizes5	0.5032	0.0126	0.0031	0.0122	0.0861	✓	
D3	Triangle2	0.4931	0.0063	0.0157	0.0047	0.1867		✓
	DS-850	0.4505	0.0019	0.0034	0.0021	0.5462		×
	ds4c2sc8	0.4503	0.0525	0.0682	0.0580	0.1263	✓	
	Square1	0.3797	0.0104	0.0140	0.0172	0.2681		✓
	Pathbased	0.1537	0.0002	0.0102	0.0160	0.2399		✓
	Flame	0.0328	0.0000	0.0096	0.0113	0.1977		✓
	Twodiamonds	0.0296	0.0009	0.0000	0.0025	0.1961		✓
	Engytime	0.0076	0.0468	0.0561	0.0451	0.1552	✓	

Table 7.3: Results of the data proprieties (CBO and DSD) evaluation considering the initial population of the artificial datasets.

These results, which present 83% agreement with the groups defined in terms of the ARI, point out that our method is promising to evaluate the data properties. In artificial datasets, only 4 (of 24) of them were wrongly classified. In Spiralsquare_S1, MST-clustering fails in detecting the clusters. However, CBO presents results that define this dataset with well-separated clusters (CBO=0). For DS-850, the DSD defines that the dataset is in G2, while the ARI is lower than 0.45, denoting the group G3. Also, DSD defined that Pat1 and Pat2 should be optimized, however these datasets present the optimal partitions in the initial population.

Furthermore, we consider that the DSD could be used as a single metric to evaluate the data properties, in cases where the ML and CL patterns are not provided or are difficult to obtain. DSD missed 6 classes (with 75% of agreement with the classes defined in terms of the ARI), and CBO missed 9 (62% of agreement), 8 (66% of agreement) and 10 (58% of agreement) by using L_{25}, L_{50} and L_{75} respectively. Moreover, DSD does not require any additional information besides the initial population.

In terms of the real-life datasets, Table 7.4 presents the ARI, CBO and DSD. In real-life datasets, only Iris was classified wrongly by the evaluation module.

Dataset	ARI	Results				Groups	
		CBO L_{25}	CBO L_{50}	CBO L_{75}	DSD	CBO	DSD
Iris	0.8755	0.0275	0.0640	0.0289	0.8950	×	
Optdigits	0.5781	0.0127	0.0287	0.0375	0.7107		✓
Soybeans	0.9211	0.0000	0.0051	0.0026	0.6301		✓
Zoo	0.7123	0.0000	0.0292	0.0249	0.7234		✓
Glass	0.4996	0.0474	0.0501	0.0521	0.4038	✓	
Libras	0.3255	0.1495	0.2379	0.3292	0.5597	✓	
Wine	0.3826	0.1826	0.2797	0.3172	0.6288	✓	
Thyroid	0.2698	0.0279	0.0612	0.0713	0.2285	✓	

Table 7.4: Results of the data proprieties (CBO and DSD) evaluation considering the initial population of real-life datasets

7.3 RESULTS OF DIFFERENT EMOC APPROACHES

Finally, Table 7.5 presents results of MOCK, MOCLE, Δ-MOCK and two versions of AEMOC, AEMOC$_Q$, and AEMOC$_D$, for artificial datasets. AEMOC$_Q$ uses the complete evaluation method (see Fig. 6.2) while AEMOC$_D$ uses only DSD to estimate the groups of data properties. The cells with a green background indicate the datasets in which the base partitions are classified in G1 and the optimization was not performed, because the evaluation module correctly assigned them. In contrast, the gray background denotes the datasets in which the evaluation module or DSD made a mistake in the definitions of the groups.

Regarding the clustering results, it is important to observe that, considering the datasets in D1, MOCK, Δ-MOCK, and MOCLE have optimal solutions in the initial population, but for some datasets, they lose them by trying to optimize base partitions. In particular, this loss occurs because Con does not distinguish solutions with optimal connectivity (Con=0) and different numbers of clusters, in which the setting of the neighborhood size (L) becomes an important factor in determining this difference. In MOCLE, this factor explains the use of 5% of the number of objects in the dataset to set L instead of $L = 10$ applied in MOCK and Δ-MOCK. However, this configuration can highly impact the clustering results in some datasets. For example, it caused in MOCLE the worst results in Pat2, and the maintenance of the optimal results in ds2c2sc13_S1 in comparison with MOCK and Δ-MOCK. In contrast, the general design of AEMOC allowed the preservation of the optimal results found by MST-clustering in most of the datasets.

In D2, a similar behavior to D1 occurs in SpiralSquare_S1, in which MOCLE and MOCK have the high-quality partitions in the initial population (base partitions generated by AL and KM with ARI equal to 1, and 0.96, respectively). In contrast, Δ-MOCK does not have a best partition in the initial population but generates it in the optimization. Nonetheless, MOCK and Δ-MOCK lose the optimal solution in the selection, because when there is more than one solution with Con=0 the decision is taken by the other objective function. In AEMOC$_D$, the use of Con' ensures a distinction of solutions with the same outcome for the connectivity with a different number of clusters, avoiding this kind of problem. In general, the results of the EMO-KC can be attributed to centroid-based representation, which presents a limitation in detecting heterogeneous and elongated data structures associated with the limitation of the objective functions in detecting these kinds of clusters. In particular, in AEMOC$_Q$, the evaluation method classified SpiralSquare_S1 in G1 because this dataset had well-separated clusters, thus the optimization was not performed. However, MST-clustering did not detect the clusters, and the best ARI in the initial population was 0.5711.

D	Dataset	MOCK	MOCLE	Δ-MOCK	EMO-KC	AEMOC$_Q$	AEMOC$_D$
D1	3MC	**1.0000**	**1.0000**	**1.0000**	0.7864	**1.0000**	**1.0000**
	Pat1	0.9374	**1.0000**	**1.0000**	0.0841	**1.0000**	**1.0000**
	Pat2	0.7355	0.2446	0.7052	0.3337	**1.0000**	**1.0000**
	Fourty	0.9999	**1.0000**	**1.0000**	0.7845	**1.0000**	**1.0000**
	Sph_6_2	**1.0000**	**1.0000**	**1.0000**	0.9224	**1.0000**	**1.0000**
	ds2c2sc13_S1	**0.3860**	**1.0000**	0.3520	**1.0000**	**1.0000**	**1.0000**
	ds2c2sc13_S2	**1.0000**	**1.0000**	0.9518	0.8721	**1.0000**	**1.0000**
	ds2c2sc13_S3	0.8703	0.7771	0.8724	0.5893	**0.9951**	**0.9951**
D2	Aggregation	0.9925	**1.0000**	0.9658	0.7767	0.9941	0.9941
	Complex8	0.8556	0.6614	0.9219	0.4803	**0.9335**	**0.9335**
	Spiralsquare_S1	0.5711	**1.0000**	0.5711	0.9690	0.5711	**1.0000**
	Spiralsquare_S2	0.9978	0.5410	0.8001	0.4641	**0.9986**	**0.9986**
	2d_10c_no9	0.9749	0.8484	0.9668	0.7727	**0.9762**	**0.9762**
	2d_4c_no2	0.9894	0.9068	0.9557	0.8837	**0.9904**	**0.9904**
	Sph_10_2	0.9805	**0.9935**	0.9782	0.0362	0.9798	0.9798
	Sizes5	0.9624	0.9435	**0.9692**	0.8297	**0.9692**	0.9554
D3	ds4c2sc8	0.9016	0.8267	0.9111	0.7878	**0.9124**	**0.9124**
	DS-850	0.9982	0.9657	**1.0000**	0.8305	0.9985	0.9985
	Flame	0.9712	0.6902	0.9568	0.5653	**0.9722**	**0.9722**
	Pathbased	0.7273	0.4851	0.7236	0.4834	**0.8240**	**0.8240**
	Engytime	0.8096	0.8151	0.7707	0.7687	**0.8236**	**0.8236**
	Square1	**0.9777**	0.9764	0.9761	0.8871	0.9748	0.9748
	Triangle2	**0.9878**	0.9246	0.9866	0.8150	0.9865	0.9865
	Twodiamonds	**1.0000**	**1.0000**	**1.0000**	0.9834	0.9998	0.9998
	Mean D1	**0.8661**	**0.8777**	**0.8602**	**0.6716**	**0.9994**	**0.9994**
	Mean D2	**0.9155**	**0.8618**	**0.8911**	**0.6615**	**0.9266**	**0.9785**
	Mean D3	**0.9216**	**0.8355**	**0.9156**	**0.7651**	**0.9365**	**0.9365**
	MEAN	**0.9011**	**0.8583**	**0.8889**	**0.6961**	**0.9542**	**0.9714**

Table 7.5: Best average ARI of MOCK, MOCLE, Δ-MOCK, EMO-KC and two versions of AEMOC: AEMOC$_Q$ and AEMOC$_D$ (Average of 30 executions). AEMOC$_Q$ uses the complete evaluation method with CBO and DSD, and AEMOC$_D$ uses only DSD to estimate the relative quality of the initial population.

In D3, both AEMOC$_Q$ and AEMOC$_D$ provided a better mean ARI for overall the group of datasets (0.93 for both AEMOC$_Q$ and AEMOC$_D$) (see the antepenultimate row in Table 7.5). In AEMOC, this result can be attributed to the use of Sep_{CL} instead of Var. The use of Var with 250 generations results in a lower ARI than the Sep_{CL} for these datasets (see Table B.1 in B.1).

Furthermore, in AEMOC, the mean ARI result of each group of datasets was significantly better in the proposed approach than in all the compared approaches, and the overall mean ARI (0.95 and 0.97 for AEMOC$_Q$ and AEMOC$_D$, respectively) was significantly better than MOCK (0.90), MOCLE (0.85), Δ-MOCK (0.88), and EMO-KC (0.69).

Table 7.6 presents the average ARI results of MOCK, MOCLE, Δ-MOCK, EMO-KC, AEMOC$_Q$ and AEMOC$_D$ for real-life datasets. In this table, the dataset in D4.1 refers to the ones classified in G2, and the dataset in D4.2 refers to the ones defined in G3. The gray background denotes the datasets in which DSD made a mistake in the definitions of the data properties groups. By analyzing these results, it is possible to observe a similar behavior to the artificial datasets, in which our approach provided the best mean results.

It is important to note that in this study, we consider the general case in which the separation and overlapping properties define whether the base partition should not be optimized, but the optimization of some data structures with (Var, Con') or (Sep_{CL}, Con') could be inappropriate. For example, the dataset Iris seems to be in the wrong group with DSD, however the optimization of the base partitions of Iris in AEMOC can cause a loss of ARI, where the

best partition found in initialization has an ARI of 0.87. In this case, not optimizing the partitions of the `Iris` dataset, as defined by the DSD, results in the best ARI for AEMOC$_D$ in comparison with the other approaches. In contrast, in AEMOC$_Q$, MOCK, Δ-MOCK, and EMO-KC, the optimization of the base partitions caused the loss of the ARI. In MOCLE, the best partition found in the initialization (ARI = 0.823) was maintained in the final population.

In terms of datasets in D4.2, AEMOC mean results were higher than the other approaches (0.49 for both AEMOC$_Q$ and AEMOC$_D$). However, it still requires a wide investigation of these datasets and their properties to improve these results.

G4	Datasets	MOCK	MOCLE	Δ-MOCK	EMO-KC	AEMOC$_Q$	AEMOC$_D$
G4.1	`Iris`	0.7700	0.8232	0.7769	0.7421	0.7611	**0.8755**
	`Optdigits`	**0.8976**	0.7461	0.8278	0.4049	0.8973	0.8973
	`Soybeans`	0.9296	0.9169	0.9348	0.7035	**0.9365**	**0.9365**
	`Zoo`	**0.8753**	0.8651	**0.8753**	0.7522	**0.8753**	**0.8753**
G4.2	`Glass`	0.5402	**0.6468**	0.5635	0.6149	0.5669	0.5669
	`Libras`	0.3927	0.3346	0.3843	0.2717	**0.4013**	0.3890
	`Wine`	**0.4025**	0.3879	**0.4025**	0.3929	**0.4025**	**0.4025**
	`Thyroid`	0.5917	0.5791	0.5836	0.4365	**0.6105**	**0.6105**
	Mean G4.1	**0.8681**	**0.8378**	**0.8537**	**0.6507**	**0.8676**	**0.8862**
	Mean G4.2	**0.4818**	**0.4871**	**0.4834**	**0.4290**	**0.4953**	**0.4922**
	MEAN	**0.6749**	**0.6624**	**0.6685**	**0.5398**	**0.6814**	**0.6941**

Table 7.6: Best average ARI of MOCK, MOCLE, Δ-MOCK, EMO-KC and two versions of AEMOC: AEMOC$_Q$ and AEMOC$_D$ in real-life datasets (Average of 30 executions).

In general, the presented results demonstrate that AEMOC is more robust than the other approaches, and the selection of objective functions and the specific parameter setting are promising for both artificial and real-life datasets. That is also pointed out in the Critical Difference Diagram, Fig. 7.1, which shows the performance comparison of EMOC approaches according to the Friedman and Bergmann-Hommel Post Hoc hypothesis test.

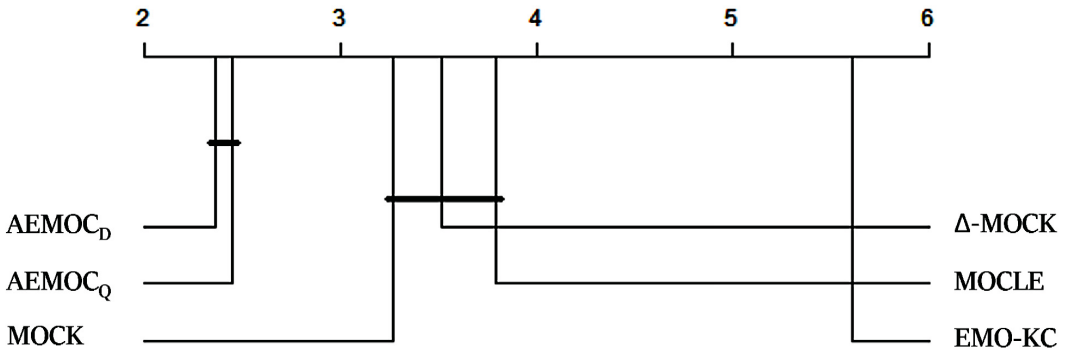


Figure 7.1: Critical Difference Diagram. The bold horizontal lines link the strategies that had statistically equivalent performance among them at a confidence level of 95%, and the lower the rank the better performance of an approach.

In the critical difference diagram, we can observe that both AEMOC$_Q$ and AEMOC$_D$ have no significant differences in their results. That emphasizes that the use of the DSD instead of the complete evaluation method is promising.

7.4 CHAPTER REMARKS

In this chapter, we present the results of the AEMOC. These results show that the use of the evaluation module can be promising considering CBO and DSD to define the relative quality of the base partitions generated by the MST-clustering. Also, it points out that DSD is robust enough to be applied as a single metric to evaluate the relative quality of base partitions.

Furthermore, the clustering results show that the proposed approach is significantly better than the MOCK, Δ-MOCK, MOCLE and EMO-KC. It shows that the evaluation and analysis of objective functions regarding the base partitions proprieties is important to designing an EMOC approach, as introduced in chapter 4.

Currently, we are working on the publication of a paper concerning the proposed AEMOC and the results of the experiments presented in this chapter in the journal Expert Systems and Applications.

In the next chapter, we present our final notes regarding the whole study described in this thesis and outline the future work direction.

8 CONCLUSION

Clustering analysis is an important research field in which emerge a variety of techniques to improve the finding of underlying structures that compose finite sets of data (clusters), where the use of evolutionary multi-objective approaches is still under-explored, requiring more attention and investigation in order to improve clustering results considering the optimization features. Thus, in this manuscript we illustrate some general issues found in established approaches regarding the choice of the objective functions. For that, we introduced the analysis of the admissibility of clustering criteria in support of defining objective functions in evolutionary multi-objective clustering approaches. In particular, we demonstrated the importance of aligning the objective function and the initialization strategy in designing the EMOC approaches.

In general, the use of a traditional clustering algorithm in the initialization provides solutions that reach the boundaries of the search space in terms of some criteria. Thus, optimizing the objective functions that consider such criteria is not required, thus other complementary criteria should be applied in the optimization.

In order to amend this common issue found in the existing EMOC approaches that use clustering algorithms in the initialization, and do not observe the search aspects in the modeling of their multi-objective strategies, we proposed the AEMOC, a new multi-objective approach that provides a new conceptual design of multi-objective optimization applied to a clustering problem.

In our approach, the initial population quality is evaluated to determine the general aspects of the optimizer. In other words, this approach introduces the analysis and the use of the base partition features to apply an offline selection of the objective functions and parameter settings in the multi-objective clustering algorithm.

In general, AEMOC provided promising results in comparison with established approaches, such as MOCK, MOCLE, Δ-MOCK and EMO-KC with a significant statistical difference, in which we verified a general gain in the clustering performance in different quality groups of the base partitions in the artificial and real-life datasets.

Furthermore, we introduced a new metric to measure the data separation degree. To our knowledge, there is not an unsupervised metric that measures the separation or the overlapping of the data. This metric was applied with the CBO, a semi-supervised metric that measures the data overlapping, to define the general separation and overlapping degree in the data. The combined method (CBO and DSD) presented robust results in defining the relative quality of the base partitions generated by MST-clustering. On the other hand, DSD is robust enough to be considered as a single metric to measure the relative quality of these partitions.

8.1 FUTURE WORKS

In terms of future work, one interesting research direction is to consider other objective functions and parameter settings to refine the clustering according to other data proprieties or application domain features. For example, a wide analysis of the real-life datasets in G4.2 could be applied to verify specific data proprieties that could support the improvement in the parameter setting or in the choice of the objective functions. Even though AEMOC provided the best average clustering results for this group, it provided ARI values below 0.5.

Furthermore, we considered the general case of the quality of the base partitions to determine whether the optimization should be performed or not. However, other cases of

inadmissibility should be examined in order to avoid “optimizing” objective functions that could worsen the clustering results. Thus, improvements in the relative quality measures to consider other initialization strategies could raise the potential of AEMOC.

Another interesting direction of research is the definition of new objective functions that could be widely admissible in different datasets. As demonstrated in the admissibility analysis, there is a lack of objective functions that can be widely admissible in different datasets, making the development of EMOC approaches for generalized clustering difficult.

Finally, the improvement of DSD or even the generation of other metrics to classify the data quality makes it possible to make better use of the MOEAs in the clustering problem, avoiding unnecessary data processing or providing a fine adjustment of the parameters.

REFERENCES

Adam, A. and Blockeel, H. (2017). Constraint-based measure for estimating overlap in clustering. In *Benelearn 2017: Proceedings of the Twenty-Sixth Benelux Conference on Machine Learning*, pages 54–61.

Aggarwal, C. C. and Reddy, C. K. (2014). Data clustering. *Algorithms and applications. Chapman&Hall/CRC Data mining and Knowledge Discovery series, Londra*.

Alhajj, R. and Kaya, M. (2008). Multi-objective genetic algorithms based automated clustering for fuzzy association rules mining. *Journal of Intelligent Information Systems*, 31(3):243–264.

Antunes, V., Sakata, T. C., Faceli, K., and de Souto, M. C. (2020). Hybrid strategy for selecting compact set of clustering partitions. *Applied Soft Computing*, 87:105971.

Attea, B. A., Hariz, W. A., and Abdulhalim, M. F. (2016). Improving the performance of evolutionary multi-objective co-clustering models for community detection in complex social networks. *Swarm and Evolutionary Computation*, 26:137–156.

Baker, F. B. and Hubert, L. J. (1976). A graph-theoretic approach to goodness-of-fit in complete-link hierarchical clustering. *Journal of the American Statistical Association*, 71(356):870–878.

Barros, M. O. (2012). An analysis of the effects of composite objectives in multiobjective software module clustering. In *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*, GECCO Í2, page 1205–1212, New York, NY, USA. Association for Computing Machinery.

Barton, T. and Kordík, P. (2015). Evaluation of relative indexes for multi-objective clustering. volume 9121, pages 465–476.

Bechikh, S., Elarbi, M., Hung, C., Hamdi, S., and Said, L. B. (2019). A hybrid evolutionary algorithm with heuristic mutation for multi-objective bi-clustering. In *2019 IEEE Congress on Evolutionary Computation (CEC)*, pages 2323–2330, Wellington, New Zealand. IEEE.

Bertsimas, D. and Tsitsiklis, J. (1993). Simulated annealing. *Statistical science*, 8(1):10–15.

Bezdek, J. C. (2013). *Pattern recognition with fuzzy objective function algorithms*. Springer Science & Business Media, New York, N. Y.

Boongoen, T. and Iam-On, N. (2018). Cluster ensembles: A survey of approaches with recent extensions and applications. *Computer Science Review*, 28:1–25.

Bousselmi, M., Bechikh, S., Hung, C., and Said, L. B. (2017). Bi-mock: A multi-objective evolutionary algorithm for bi-clustering with automatic determination of the number of bi-clusters. In Liu, D., Xie, S., Li, Y., Zhao, D., and El-Alfy, E. M., editors, *Neural Information Processing*, pages 366–376, Cham. Springer International Publishing.

Brun, M., Sima, C., Hua, J., Lowey, J., Carroll, B., Suh, E., and Dougherty, E. R. (2007). Model-based evaluation of clustering validation measures. *Pattern Recogn.*, 40(3):807–824.

Cheng, R., Jin, Y., Olhofer, M., and Sendhoff, B. (2016). A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 20(5):773–791.

Coello, C. A. C., Lamont, G. B., and Veldhuizen, D. A. V. (2006). *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*. Springer-Verlag, Berlin, Heidelberg.

Corne, D. W., Knowles, J. D., and Oates, M. J. (2000). The pareto envelope-based selection algorithm for multiobjective optimization. In Schoenauer, M., Deb, K., Rudolph, G., Yao, X., Lutton, E., Merelo, J. J., and Schwefel, H., editors, *Parallel Problem Solving from Nature PPSN VI*, pages 839–848, Berlin, Heidelberg. Springer Berlin Heidelberg.

Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In Schoenauer, M., Deb, K., Rudolph, G., Yao, X., Lutton, E., Merelo, J. J., and Schwefel, H., editors, *Parallel Problem Solving from Nature PPSN VI*, pages 849–858, Berlin, Heidelberg. Springer Berlin Heidelberg.

Deb, K. and Jain, H. (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601.

Demir, G. N., Uyar, A. Ş., and Gündüz-Öğüdücü, Ş. (2010). Multiobjective evolutionary clustering of web user sessions: a case study in web page recommendation. *Soft Computing*, 14(6):579–597.

Di Nuovo, A. G., Palesi, M., and Catania, V. (2007). Multi-objective evolutionary fuzzy clustering for high-dimensional problems. In *2007 IEEE International Fuzzy Systems Conference*, pages 1–6, London, UK. IEEE.

Dong, Z., Jia, H., and Liu, M. (2018). An adaptive multiobjective genetic algorithm with fuzzy-means for automatic data clustering. *Mathematical Problems in Engineering*, 2018.

Du, J., Korkmaz, E. E., Alhajj, R., and Barker, K. (2005). Alternative clustering by utilizing multi-objective genetic algorithm with linked-list based chromosome encoding. In Perner, P. and Imiya, A., editors, *Machine Learning and Data Mining in Pattern Recognition*, pages 346–355, Berlin, Heidelberg. Springer Berlin Heidelberg.

Dutta, D., Dutta, P., and Sil, J. (2012a). Clustering by multi objective genetic algorithm. In *2012 1st International Conference on Recent Advances in Information Technology (RAIT)*, pages 548–553, Dhanbad, India. IEEE.

Dutta, D., Dutta, P., and Sil, J. (2012b). Data clustering with mixed features by multi objective genetic algorithm. In *2012 12th International Conference on Hybrid Intelligent Systems (HIS)*, pages 336–341, Pune, India. IEEE.

Dutta, D., Dutta, P., and Sil, J. (2012c). Simultaneous feature selection and clustering for categorical features using multi objective genetic algorithm. In *2012 12th International Conference on Hybrid Intelligent Systems (HIS)*, pages 191–196, Pune, India. IEEE.

Dutta, D., Sil, J., and Dutta, P. (2019). Automatic clustering by multi-objective genetic algorithm with numeric and categorical features. *Expert Systems with Applications*, 137:357–379.

Dutta, P. and Saha, S. (2017). Fusion of expression values and protein interaction information using multi-objective optimization for improving gene clustering. *Comput. Biol. Med.*, 89(C):31–43.

Eltaeib, T. and Mahmood, A. (2018). Differential evolution: A survey and analysis. *Applied Sciences*, 8(10).

Ertöz, L., Steinbach, M., and Kumar, V. (2002). A new shared nearest neighbor clustering algorithm and its applications. In *Workshop on Clustering High Dimensional Data and its Applications at 2nd SIAM International Conference on Data Mining*, pages 105–115, Arlington, VA, USA. SIAM.

Faceli, K., de Carvalho, A. C. P. L. F., and de Souto, M. C. P. (2006). Multi-objective clustering ensemble. In *2006 Sixth International Conference on Hybrid Intelligent Systems (HIS06)*, pages 51–51, Rio de Janeiro, Brazil. IEEE.

Faceli, K., de Souto, M. C. P., de Araújo, D. S. A., and de Carvalho, A. C. P. L. F. (2009). Multi-objective clustering ensemble for gene expression data analysis. *Neurocomput.*, 72(13-15):2763–2774.

Faceli, K., Lorena, A. C., Gama, J., and Carvalho, A. C. P. d. L. F. d. (2011). *Inteligência artificial: uma abordagem de aprendizado de máquina*. LTC.

Fern, X. Z. and Brodley, C. E. (2004). Solving cluster ensemble problems by bipartite graph partitioning. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML 04, pages 36–, New York, NY, USA. ACM.

Fisher, L. and Ness, J. W. V. (1971). Admissible clustering procedures. *Biometrika*, 58(1):91–104.

Folino, F. and Pizzuti, C. (2010). A multiobjective and evolutionary clustering method for dynamic networks. In *2010 International Conference on Advances in Social Networks Analysis and Mining*, pages 256–263, Odense, Denmark. IEEE.

Fonseca, C. M. and Fleming, P. J. (1995). An overview of evolutionary algorithms in multiobjective optimization. *Evol. Comput.*, 3(1):1–16.

Fränti, P., Kivijärvi, J., Kaukoranta, T., and Nevalainen, O. (1997). Genetic Algorithms for Large-Scale Clustering Problems. *The Computer Journal*, 40(9):547–554.

Garcia-Piquer, A., Bacardit, J., Fornells, A., and Golobardes, E. (2017). Scaling-up multiobjective evolutionary clustering algorithms using stratification. *Pattern Recognition Letters*, 93:69–77. Pattern Recognition Techniques in Data Mining.

Garza-Fabre, M., Handl, J., and Knowles, J. (2017). A new reduced-length genetic representation for evolutionary multiobjective clustering. In Trautmann, H., Rudolph, G., Klamroth, K., Schütze, O., Wiecek, M., Jin, Y., and Grimme, C., editors, *Evolutionary Multi-Criterion Optimization*, pages 236–251, Cham. Springer International Publishing.

Garza-Fabre, M., Handl, J., and Knowles, J. (2018). An improved and more scalable evolutionary approach to multiobjective clustering. *IEEE Transactions on Evolutionary Computation*, 22(4):515–535.

Hacioglu, G., Kand, V. F. A., and Sesli, E. (2016). Multi objective clustering for wireless sensor networks. *Expert Systems with Applications*, 59:86–100.

Hancer, E. and Karaboga, D. (2017). A comprehensive survey of traditional, merge-split and evolutionary approaches proposed for determination of cluster number. *Swarm and Evolutionary Computation*, 32:49–67.

Handl, J., Kell, D. B., and Knowles, J. (2007). Multiobjective optimization in bioinformatics and computational biology. *IEEE/ACM Transactions on computational biology and bioinformatics*, 4(2):279–292.

Handl, J. and Knowles, J. (2005a). Exploiting the trade-off - the benefits of multiple objectives in data clustering. In Coello Coello, C. A., Hernández Aguirre, A., and Zitzler, E., editors, *Evolutionary Multi-Criterion Optimization*, pages 547–560, Berlin, Heidelberg. Springer Berlin Heidelberg.

Handl, J. and Knowles, J. (2005b). Improvements to the scalability of multiobjective clustering. In *2005 IEEE Congress on Evolutionary Computation*, volume 3, pages 2372–2379 Vol. 3, Edinburgh, UK. IEEE.

Handl, J. and Knowles, J. (2005c). Multiobjective clustering around medoids. In *2005 IEEE Congress on Evolutionary Computation*, volume 1, pages 632–639 Vol.1, Edinburgh, UK. IEEE.

Handl, J. and Knowles, J. (2007). An evolutionary approach to multiobjective clustering. *IEEE Transactions on Evolutionary Computation*, 11(1):56–76.

Handl, J. and Knowles, J. (2012). Clustering criteria in multiobjective data clustering. In Coello, C. A. C., Cutello, V., Deb, K., Forrest, S., Nicosia, G., and Pavone, M., editors, *Parallel Problem Solving from Nature - PPSN XII*, pages 32–41, Berlin, Heidelberg. Springer, Springer Berlin Heidelberg.

Handl, J. and Knowles, J. (2013). Evidence accumulation in multiobjective data clustering. In Purshouse, R. C., Fleming, P. J., Fonseca, C. M., Greco, S., and Shaw, J., editors, *Evolutionary Multi-Criterion Optimization*, pages 543–557, Berlin, Heidelberg. Springer Berlin Heidelberg.

Horn, J., Nafpliotis, N., and Goldberg, D. (1994). A niched pareto genetic algorithm for multiobjective optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*, pages 82–87 vol.1, Orlando, FL, USA. IEEE.

Hruschka, E. R., Campello, R. J. G. B., and andA. C. P. L. F. de Carvalho, A. A. F. (2009). A survey of evolutionary algorithms for clustering. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 39(2):133–155.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2(1):193–218.

Jain, A. (2010). Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31:651–666.

Jain, A. K. and Dubes, R. C. (1988). *Algorithms for Clustering Data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: A review. *ACM Comput. Surv.*, 31(3):264–323.

Jaskowiak, P. A., Costa, I. G., and Campello, R. J. G. B. (2022). The area under the roc curve as a measure of clustering quality. *Data Min. Knowl. Discov.*, 36(3):1219–1245.

Kaya, M. and Alhajj, R. (2004). Integrating multi-objective genetic algorithms into clustering for fuzzy association rules mining. In *Fourth IEEE International Conference on Data Mining (ICDḾ04)*, pages 431–434, Brighton, UK. IEEE.

Kirkland, O., Rayward-Smith, V. J., and de la Iglesia, B. (2011). A novel multi-objective genetic algorithm for clustering. In Yin, H., Wang, W., and Rayward-Smith, V., editors, *Intelligent Data Engineering and Automated Learning - IDEAL 2011*, pages 317–326, Berlin, Heidelberg. Springer Berlin Heidelberg.

Kultzak, A., Morimoto, C. Y., Pozo, A., and de Souto, M. C. P. (2021). Multi-objective clustering: A data-driven analysis of MOCLE, MOCK and Δ-MOCK. In Mantoro, T., Lee, M., Ayu, M. A., Wong, K. W., and Hidayanto, A. N., editors, *Neural Information Processing*, pages 46–54, Cham. Springer International Publishing.

Larsen, B. and Aone, C. (1999). Fast and effective text mining using linear-time document clustering. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ́99, page 16–22, New York, NY, USA. Association for Computing Machinery.

Lee, J. S., Park, S. C., Lee, J. J., and Ham, H. H. (2014). Document clustering using multi-objective genetic algorithms with parallel programming based on cuda. In *2014 11th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, volume 01, pages 280–287, Vienna, Austria. IEEE, IEEE.

Li, B., Li, J., Tang, K., and Yao, X. (2015). Many-objective evolutionary algorithms: A survey. *ACM Computing Surveys*, 48(1):13:1–13:35.

Li, H., Gong, M., Wang, Q., Liu, J., and Su, L. (2016). A multiobjective fuzzy clustering method for change detection in sar images. *Applied Soft Computing*, 46:767–777.

Li, J., Liu, R., Zhang, M., and Li, Y. (2017). Ensemble-based multi-objective clustering algorithms for gene expression data sets. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pages 333–340, Donostia, Spain. IEEE.

Li, K., Deb, K., Zhang, Q., and Kwong, S. (2015). An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Transactions on Evolutionary Computation*, 19(5):694–716.

Li, M., Yang, S., and Liu, X. (2014). Shift-based density estimation for pareto-based algorithms in many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 18(3):348–365.

Liu, C., Liu, J., Peng, D., and Wu, C. (2018). A general multiobjective clustering approach based on multiple distance measures. *IEEE Access*, 6:41706–41719.

Liu, C., Zhao, Q., Yan, B., Elsayed, S., and Sarker, R. (2019). Transfer learning-assisted multi-objective evolutionary clustering framework with decomposition for high-dimensional data. *Information Sciences*, 505:440–456.

Liu, R., Liu, Y., and Li, Y. (2012). An improved method for multi-objective clustering ensemble algorithm. In *2012 IEEE Congress on Evolutionary Computation*, pages 1–8, Brisbane, QLD, Australiac. IEEE.

Liu, R., Wang, R., Yu, X., and An, L. (2017). Shape automatic clustering-based multi-objective optimization with decomposition. *Machine Vision and Applications*, 28(5):497–508.

Liu, Y., Li, Z., Xiong, H., Gao, X., and Wu, J. (2010). Understanding of internal clustering validation measures. In *2010 IEEE international conference on data mining*, pages 911–916. IEEE.

Luo, J., Jiao, L., and Lozano, J. (2015). A sparse spectral clustering framework via multi-objective evolutionary algorithm. *IEEE Transactions on Evolutionary Computation*, 20:1–1.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, Berkeley, Calif. University of California Press.

Man, K., Chan, T., Tang, K., and Kwong, S. (2004). Jumping-genes in evolutionary computing. In *30th Annual Conference of IEEE Industrial Electronics Society, 2004. IECON 2004*, volume 2, pages 1268–1272 Vol. 2, Busan, Korea (South). IEEE.

Matake, N., Hiroyasu, T., Miki, M., and Senda, T. (2007). Multiobjective clustering with automatic k-determination for large-scale data. In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, GECCO '07, page 861–868, New York, NY, USA. Association for Computing Machinery.

Menéndez, H. D., Barrero, D. F., and Camacho, D. (2013). A multi-objective genetic graph-based clustering algorithm with memory optimization. In *2013 IEEE Congress on Evolutionary Computation*, pages 3174–3181, Cancun, Mexico. IEEE, IEEE.

Menéndez, H. D., Barrero, D. F., and Camacho, D. (2014). A co-evolutionary multi-objective approach for a k-adaptive graph-based clustering algorithm. In *2014 IEEE Congress on Evolutionary Computation (CEC)*, pages 2724–2731, Beijing, China. IEEE.

Morik, K., Kaspari, A., Wurst, M., and Skirzynski, M. (2012). Multi-objective frequent termset clustering. *Knowledge and information systems*, 30(3):715–738.

Morimoto, C. Y., Pozo, A., and de Souto, M. C. (2022a). An analysis of the admissibility of the objective functions applied in evolutionary multi-objective clustering. *Information Sciences*, 610:1143–1162.

Morimoto, C. Y., Pozo, A., and de Souto, M. C. P. (2021). A review of evolutionary multi-objective clustering approaches.

Morimoto, C. Y., Pozo, A., and de Souto, M. C. P. (2022b). Detecting nested structures through evolutionary multi-objective clustering. In *Applications of Evolutionary Computation: 25th European Conference, EvoApplications 2022, Held as Part of EvoStar 2022, Madrid, Spain, April 20–22, 2022, Proceedings*, page 369–385, Berlin, Heidelberg. Springer-Verlag.

Mukhopadhyay, A. and Maulik, U. (2007). Multiobjective approach to categorical data clustering. In *2007 IEEE Congress on Evolutionary Computation*, pages 1296–1303, Singapore. IEEE.

Mukhopadhyay, A. and Maulik, U. (2009). Unsupervised pixel classification in satellite imagery using multiobjective fuzzy clustering combined with svm classifier. *IEEE transactions on geoscience and remote sensing*, 47(4):1132–1138.

Mukhopadhyay, A., Maulik, U., and Bandyopadhyay, S. (2007). Multiobjective genetic fuzzy clustering of categorical attributes. In *10th International Conference on Information Technology (ICIT 2007)*, pages 74–79, Rourkela, India. IEEE.

Mukhopadhyay, A., Maulik, U., and Bandyopadhyay, S. (2009). Multiobjective genetic algorithm-based fuzzy clustering of categorical attributes. *IEEE Transactions on Evolutionary Computation*, 13(5):991–1005.

Mukhopadhyay, A., Maulik, U., and Bandyopadhyay, S. (2010). Simultaneous informative gene selection and clustering through multiobjective optimization. In *IEEE Congress on Evolutionary Computation*, pages 1–8, Barcelona, Spain. IEEE.

Mukhopadhyay, A., Maulik, U., and Bandyopadhyay, S. (2013). An interactive approach to multiobjective clustering of gene expression patterns. *IEEE Transactions on Biomedical Engineering*, 60(1):35–41.

Mukhopadhyay, A., Maulik, U., and Bandyopadhyay, S. (2015). A survey of multiobjective evolutionary clustering. *ACM Computing Surveys (CSUR)*, 47(4):61:1–61:46.

Özyer, T. and Alhajj, R. (2009). Parallel clustering of high dimensional data by integrating multi-objective genetic algorithm with divide and conquer. *Applied Intelligence*, 31(3):318.

Pakhira, M. K., Bandyopadhyay, S., and Maulik, U. (2004). Validity index for crisp and fuzzy clusters. *Pattern Recognition*, 37(3):487–501.

Paul, A. K. and Shill, P. C. (2018). New automatic fuzzy relational clustering algorithms using multi-objective nsga-ii. *Information Sciences*, 448-449:112–133.

Peiravi, A., Mashhadi, H. R., and Hamed Javadi, S. (2013). An optimal energy-efficient clustering method in wireless sensor networks using multi-objective genetic algorithm. *International Journal of Communication Systems*, 26(1):114–126.

Pizzuti, C. and Socievole, A. (2019). Multiobjective optimization and local merge for clustering attributed graphs. *IEEE transactions on cybernetics*, 50(12):4997–5009.

Pohlert, T. (2018). PMCMR: Calculate Pairwise Multiple Comparisons of Mean Rank Sums.

Praditwong, K., Harman, M., and Yao, X. (2010). Software module clustering as a multi-objective search problem. *IEEE Transactions on Software Engineering*, 37(2):264–282.

Qian, X., Zhang, X., Jiao, L., and Ma, W. (2008). Unsupervised texture image segmentation using multiobjective evolutionary clustering ensemble algorithm. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pages 3561–3567, Hong Kong, China. IEEE.

Rahman, M. A. and Islam, M. Z. (2014). A hybrid clustering technique combining a novel genetic algorithm with k-means. *Knowledge-Based Systems*, 71:345–365.

Rai, P. and Singh, S. (2010). A survey of clustering techniques. *International Journal of Computer Applications*, 7(12):1–5.

Rana, S., Jasola, S., and Kumar, R. (2011). A review on particle swarm optimization algorithms and their applications to data clustering. *Artificial Intelligence Review*, 35(3):211–222.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850.

Ripon, K. S. N. and Siddique, M. N. H. (2009). Evolutionary multi-objective clustering for overlapping clusters detection. In *2009 IEEE Congress on Evolutionary Computation*, pages 976–982, Trondheim, Norway. IEEE.

Ripon, K. S. N., Tsang, C., Kwong, S., and I., M.-K. (2006a). Multi-objective evolutionary clustering using variable-length real jumping genes genetic algorithm. In *18th International Conference on Pattern Recognition (ICPR06)*, volume 1, pages 1200–1203, Hong Kong, China. IEEE.

Ripon, K. S. N., Tsang, C.-H., and Kwong, S. (2006b). Multi-objective data clustering using variable-length real jumping genes genetic algorithm and local search method. In *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, pages 3609–3616, Vancouver, BC, Canada. IEEE.

Romero-Zaliz, R. C., Rubio-Escudero, C., Cobb, J. P., Herrera, F., Cordón, Ó., and Zwir, I. (2008). A multiobjective evolutionary conceptual clustering methodology for gene annotation within structural databases: a case of study on the gene ontology database. *IEEE Transactions on Evolutionary Computation*, 12(6):679–701.

Russell, S. and Norvig, P. (2002). *Artificial intelligence: a modern approach*. Prentice Hall, Upper Saddle River, NJ, USA.

Schütze, H., Manning, C. D., and Raghavan, P. (2008). *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge, Cambridge, England.

Sert, O. C., Dursun, K., Özyer, T., Jida, J., and Alhajj, R. (2012). The unification and assessment of multi-objective clustering results of categorical datasets with h-confidence metric. *J. UCS*, 18(4):507–531.

Sert, O. C., Dursun, K., and Özyer, T. (2011). Ensemble of multi-objective clustering unified with h-confidence metric as validity metric. In *2011 International Conference on Advances in Social Networks Analysis and Mining*, pages 537–541, Kaohsiung, Taiwan. IEEE.

Shang, R., Liu, H., and Jiao, L. (2017). Multi-objective clustering technique based on k-nodes update policy and similarity matrix for mining communities in social networks. *Physica A: Statistical Mechanics and its Applications*, 486:1–24.

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905.

Shirakawa, S. and Nagao, T. (2009). Evolutionary image segmentation based on multiobjective clustering. In *2009 IEEE Congress on Evolutionary Computation*, pages 2466–2473, Trondheim, Norway. IEEE.

Siarry, P. (2016). *Metaheuristics*. Springer, Cham,Switzerland.

Skabar, A. and Abdalgader, K. (2013). Clustering sentence-level text using a novel fuzzy relational clustering algorithm. *IEEE Transactions on Knowledge and Data Engineering*, 25(1):62–75.

Sneath, P. H. (1957). The application of computers to taxonomy. *Microbiology*, 17(1):201–226.

Sokal, R. R. (1958). A statistical method for evaluating systematic relationships. *Univ. Kansas, Sci. Bull.*, 38:1409–1438.

Sorensen, T. A. (1948). A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on danish commons. *Biol. Skar.*, 5:1–34.

Spackman, K. A. (1989). Signal detection theory: Valuable tools for evaluating inductive learning. In *Proceedings of the sixth international workshop on Machine learning*, pages 160–163. Elsevier.

Srikanth, R., George, R., Warsi, N., Prabhu, D., Petry, F. E., and Buckles, B. P. (1995). A variable-length genetic algorithm for clustering and classification. *Pattern Recogn. Lett.*, 16(8):789–800.

Strehl, A. (2002). *Relationship-based Clustering and Cluster Ensembles for High-dimensional Data Mining*. PhD thesis, The University of Texas. AAI3088578.

Timmis, J., Hone, A., Stibor, T., and Clark, E. (2008). Theoretical advances in artificial immune systems. *Theoretical Computer Science*, 403(1):11–32.

Tsai, C., Chen, W., and Chiang, M. (2012). A modified multiobjective ea-based clustering algorithm with automatic determination of the number of clusters. In *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2833–2838, Seoul, Korea (South). IEEE.

Tsekouras, G. E., Papageorgiou, D., Kotsiantis, S. B., Kalloniatis, C., and Pintelas, P. E. (2004). Fuzzy clustering of categorical attributes and its use in analyzing cultural data. In *International Conference on Computational Intelligence*, pages 202–206. Citeseer.

Wahid, A., Gao, X., and Andreae, P. (2015). Multi-objective clustering ensemble for high-dimensional data based on strength pareto evolutionary algorithm (spea-ii). In *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–9, Paris, France. IEEE.

Wang, L., Cui, G., Zhou, Q., and Li, K. (2020). A multi-clustering method based on evolutionary multiobjective optimization with grid decomposition. *Swarm and Evolutionary Computation*, 55:100691.

Wang, Q., Li, H., Gong, M., Su, L., and Jiao, L. (2014). A multiobjective optimization method based on moea/d and fuzzy clustering for change detection in sar images. In *2014 IEEE Congress on Evolutionary Computation (CEC)*, pages 3024–3029, Beijing, China. IEEE.

Wang, R., Lai, S., Wu, G., Xing, L., Wang, L., and Ishibuchi, H. (2018). Multi-clustering via evolutionary multi-objective optimization. *Information Sciences*, 450:128–140.

Wang, Y., Walker, J. A., Bale, S. J., Trefzer, M. A., and Tyrrell, A. M. (2015). Two-phase multiobjective genetic algorithm for constrained circuit clustering on fpgas. In *2015 IEEE Congress on Evolutionary Computation (CEC)*, pages 1183–1190, Sendai, Japan. IEEE.

Wei, Y.-C. and Cheng, C.-K. (1991). Ratio cut partitioning for hierarchical designs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 10(7):911–921.

Wikaisuksakul, S. (2014). A multi-objective genetic algorithm with fuzzy c-means for automatic data clustering. *Applied Soft Computing*, 24:679 – 691.

Xia, H., Zhuang, J., and Yu, D. (2013). Novel soft subspace clustering with multi-objective evolutionary approach for high-dimensional data. *Pattern Recogn.*, 46(9):2562–2575.

Xu, D. and Tian, Y. (2015). A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2(2):165–193.

Xu, R. and Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678.

Yuan, Y., Xu, H., Wang, B., and Yao, X. (2016). A new dominance relation-based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 20(1):16–37.

Z. Zhou, S. Z. (2018). Kernel-based multiobjective clustering algorithm with automatic attribute weighting. *Soft Computing*, 22(11):3685–3709.

Zahn, C. (1971). Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers*, C-20(1):68–86.

Zhang, M., Jiao, L., Ma, W., Ma, J., and Gong, M. (2016). Multi-objective evolutionary fuzzy clustering for image segmentation with moea/d. *Applied Soft Computing*, 48:621–637.

Zhao, F., Fan, J., Liu, H., Lan, R., and Chen, C. W. (2018). Noise robust multiobjective evolutionary clustering image segmentation motivated by the intuitionistic fuzzy information. *IEEE Transactions on Fuzzy Systems*, 27(2):387–401.

Zhao, F., Li, C., Liu, H., and Fan, J. (2019). A multi-objective interval valued fuzzy clustering algorithm with spatial information for noisy image segmentation. *Journal of Intelligent & Fuzzy Systems*, 36(6):5333–5344.

Zhao, X., Liang, J., and Dang, C. (2017). Clustering ensemble selection for categorical data based on internal validity indices. *Pattern Recogn.*, 69(C):150–168.

Zheng, Y., Jia, L., and Cao, H. (2012). Multi-objective gene expression programming for clustering. *Information Technology and Control*, 41(3):283–294.

Zhu, L., Cao, L., and Yang, J. (2012). Multiobjective evolutionary algorithm-based soft subspace clustering. In *2012 IEEE Congress on Evolutionary Computation*, pages 1–8, Brisbane, QLD, Australia. IEEE.

Zhu, S. and Xu, L. (2018). Many-objective fuzzy centroids clustering algorithm for categorical data. *Expert Systems with Applications*, 96:230–248.

Zhu, S., Xu, L., and Cao, L. (2018). A study of automatic clustering based on evolutionary many-objective optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '18, page 173–174, New York, NY, USA. Association for Computing Machinery.

Zhu, S., Xu, L., and Goodman, E. D. (2020). Evolutionary multi-objective automatic clustering enhanced with quality metrics and ensemble strategy. *Knowledge-Based Systems*, 188:105018.

Zhu, S., Xu, L., and Goodman, E. D. (2021). Hierarchical topology-based cluster representation for scalable evolutionary multiobjective clustering. *IEEE Transactions on Cybernetics*, pages 1–15.

Zitzler, E., Laumanns, M., and Thiele, L. (2001). Spea2: Improving the strength pareto evolutionary algorithm. *TIK-report*, 103.

Zwir, I., Zaliz, R., and Ruspini, E. (2002). Automated biological sequence description by genetic multiobjective generalized clustering. *Annals of the New York Academy of Sciences*, 980:65–82. cited By 16.

APPENDIX A – OBJECTIVE FUNCTIONS

A.1 CLUSTERING CRITERIA

In this section, we present the CVIs applied as objective functions in the literature, as introduced in the Section 2.4.2.1. We considered a common notation in the equations, where n refers to the number of objects in the dataset $\mathbf{X}$, π denotes a partition, k denotes the number of clusters in π, $\mathbf{c}_i$ refers to the ith cluster that belongs to π, $\mathbf{x}_a$ denotes a generic object, n_i denotes the number of objects in $\mathbf{c}_i$, $\mathbf{z}_i$ refers to the centroid of cluster $\mathbf{c}_i$, and $\bar{\mathbf{z}}$ represents the centroid of the dataset. Furthermore, $d(.,.)$ denotes the chosen distance function.

A.1.1 Compactness criteria

The **Average Within Group Sum of Squares** ($AWGSS$) is computed by the average of the distance between each object in the cluster and its centroid, as present in Eq. A.1. It should be minimized to obtain compact clusters (Kirkland et al., 2011).

$$AWGSS(\pi) = \sum_{i=1}^{k} \frac{\sum_{\mathbf{x}\in\mathbf{c}_i} d(\mathbf{x_a}, \mathbf{z}_i)}{n_i} \tag{A.1}$$

The **overall Deviation** (Dev) is computed as the overall summed distance between data points and their corresponding cluster center, as defined in Eq. A.2. It should be minimized in order to obtain compact clusters (Handl and Knowles, 2005a).

$$Dev(\pi) = \sum_{\mathbf{c}_i\in\pi} \sum_{\mathbf{x}_a\in\mathbf{c}_i} d(\mathbf{x}_a, \mathbf{z}_i) \tag{A.2}$$

Sert et al. (Sert et al., 2011, 2012) considered the **K-Mode internal distance** ($\mathrm{Km_{id}}$) and **K-Mode weighted internal distance** ($\mathrm{Km_{wid}}$) as objective functions. These indices are computed in a similar way to Dev, but the mode is used instead of the centroid. $\mathrm{Km_{id}}$ and $\mathrm{Km_{wid}}$ should be minimized as objective functions.

The **intra-cluster Entropy** (Ent) measures the degree of similarity between each cluster center and the data objects that belong to that cluster, as the probability of grouping all the data objects into that particular cluster. A larger value of this index implies better clustering (Ripon et al., 2006a,b; Ripon and Siddique, 2009). This index is defined by Eq. A.3, where $g(\mathbf{z}_i)$ is the average similarity between $\mathbf{z}_i$ and the data object belong to cluster $\mathbf{c}_i$, and the $cos(.,.)$ represents the cosine distance.

$$\begin{gathered} Ent(\pi) = \sum_{i=1}^{k} \left[(1 - h(c_i))g(\mathbf{z}_i)\right]^{1/k}, \text{ where} \\ h(c_i) = -\left[(g(\mathbf{z}_i)\log_2 g(\mathbf{z}_i) + (1 - g(\mathbf{z}_i))\log_2(1 - g(\mathbf{z}_i))\right], \text{ and} \\ g(\mathbf{z}_i) = \frac{1}{n_i}\sum_{a=1}^{n_i}\left(0.5 + \frac{\cos(\mathbf{z}_i, \mathbf{x}_a)}{2}\right) \end{gathered} \tag{A.3}$$

The **Homogeneity** (H) index is computed by the sum of the average minimal intra-cluster distance, according to Eq. A.4, where $\min(d(\mathbf{z}_i, \mathbf{x}_a))$ denotes the lowest distance between the

points $\mathbf{x}_a$ in the cluster $\mathbf{c}_i$ and the cluster mode $\mathbf{m}_i$. H should be maximized to obtain homogeneous clusters (Dutta et al., 2012a).

$$H(\pi) = \sum_{i=1}^{k} \left[\frac{\sum_{a=1}^{n_i} \min(d(\mathbf{m}_i, \mathbf{x}_a))}{n_i} \right] \tag{A.4}$$

The **intra-cluster Variance** (Var) is conceptually similar to Dev, as shown in Eq. A.5, and it also should be minimized to obtain compact clusters (Garza-Fabre et al., 2018).

$$Var(\pi) = \frac{1}{n} \sum_{\mathbf{c}_i \in \pi} \sum_{\mathbf{x}_a \in \mathbf{c}_i} d(\mathbf{x}_a, \mathbf{z}_i) \tag{A.5}$$

The **Total Within-Cluster Variance** ($TWCV$) is also applied to identify sets of compact clusters, as defined in Eq. A.6, where f is the size of the dimensional feature space, $\mathbf{x}_{ar}$ denotes the rth feature value of the ath data point, $\mathbf{z}_{ir}$ is the centroid of the ith cluster of the rth feature, and $w_{ai} \in [0, 1]$ and $\sum_{i=1}^{k} w_{ai} = 1$. The goal is to minimize $TWCV$ to obtain compact clusters (Du et al., 2005).

$$\begin{gathered} TWCV(\pi) = \sum_{i=1}^{k} \sum_{a=1}^{n} w_{ai} \sum_{r=1}^{f} (\mathbf{x}_{ar} - \mathbf{z}_{ir})^2, \text{ where} \\ \mathbf{z}_{ir} = \frac{\sum_{a=1}^{n} w_{ai} \mathbf{x}_{ar}}{\sum_{a=1}^{n} w_{ai}}, \text{ and } w_{ai} \begin{cases} 1, \text{if } a^{th} \text{object belongs to the } i^{th} cluster \\ 0, \text{otherwise} \end{cases} \end{gathered} \tag{A.6}$$

The **Fuzzy Compactness** (J_m) represents the global fuzzy cluster variance, as defined in Eq. A.7, where u_{ia} is the membership degree of the ath data point to the ith cluster, and m is the fuzzy exponent. The smaller value of J_m corresponds to more compact clusters (Bezdek, 2013).

$$J_m = \sum_{i=1}^{k} \sum_{a=1}^{n} u_{ia}^m d(\mathbf{z}_i, \mathbf{x}_a) \tag{A.7}$$

Zhu et al. introduced an adapted J_m that considers the cluster weighting subspace, the **Fuzzy weighting subspace clustering** (J_{wm}). This index is defined in Eq. A.8, where f is the number of attributes (or vector of features), $\mathbf{x}_a r$ denotes rth feature of the ath object, and $\mathbf{z}_{ir}$ is the centroid of the ith cluster of the rth feature. w_{ir} is defined in Eq. A.9, where m is the fuzziness exponent, and τ is the fuzzy weighting index. J_{wm} should be minimized to improve the clustering (Zhu et al., 2012).

$$J_{wm} = \sum_{i=1}^{k} \sum_{a=1}^{n} u_{ia}^m \sum_{r=1}^{f} w_{ir}^{\tau} d(\mathbf{x}_{ar} - \mathbf{z}_{ir})^2 \tag{A.8}$$

$$w_{ir} = \frac{\left(\sum_{a=1}^{n} u_{ia}^m d(\mathbf{x}_{ar} - \mathbf{z}_{ir})^2\right)^{1/\tau - 1}}{\sum_{r=1}^{f} \left(\sum_{a=1}^{n} u_{ia}^m d(\mathbf{x}_{ar} - \mathbf{z}_{ir})^2\right)^{1/\tau - 1}}, \text{ where } u_{ia} = \frac{(\sum_{r=1}^{f} w_{ia}^{\tau} d(\mathbf{x}_{ar} - \mathbf{z}_{ir})^2)^{-1/m-1}}{\sum_{i=1}^{k} (\sum_{r=1}^{f} w_{ia}^{\tau} d(\mathbf{x}_{ar} - \mathbf{z}_{ir})^2)^{-1/m-1}} \tag{A.9}$$

A.1.2 Connectedness criteria

The **Connectivity** (Con) index (Handl and Knowles, 2005a) evaluates the degree to which neighboring data points have been placed in the same cluster. This index is computed according to Eq. (A.10), where L is the parameter that determines the number of nearest neighbors that contribute to the connectivity, nn_{ab} is the bth nearest neighbor of object $\mathbf{x}_a$. Con as objectives should be minimized.

$$Con(\pi) = \sum_{a=1}^{n} \sum_{b=1}^{L} f(\mathbf{x}_a, nn_{ab}), \text{ where } f(\mathbf{x}_a, nn_{ab}) \begin{cases} \frac{1}{b}, \text{if } \nexists \mathbf{c}_k : \mathbf{x}_a, nn_{ab} \in \mathbf{c}_k \\ 0, \text{otherwise} \end{cases} \tag{A.10}$$

The **Data Continuity Degree** (DCD) measures the connectedness of the data in terms of the connectivity factor (the total edges sum for each minimum spanning tree) in a similarity graph. In general, it can be computed in two steps. First, a similarity function is applied in order to generate a similarity graph, the k_{size}-Graph. In this graph, a vertex v_a is connected with the vertex v_b if v_b is among the k-nearest neighbors of v_a. After that, the total minimal spanning tree edges are computed considering all nodes connected within the neighborhood of the current node and internally — this process is repeated with each connected component due to the graph not being fully connected. The average arithmetic value of the metric (the connectivity factor divided by the number of clusters) is the result of this objective, which should be maximized in the optimization (Menéndez et al., 2013).

A.1.3 Separation criteria

The **Average Between-Group Sum of Squares** ($ABGSS$) is computed as the average distance between the clusters' centroids and the centroid of the data, as defined in Eq. A.11. It should be maximized to obtain well-separated clusters (Kirkland et al., 2011).

$$ABGSS(\pi) = \frac{\sum_{i=1}^{k} n_i . d(\mathbf{z}_i, \overline{\mathbf{z}})}{k} \tag{A.11}$$

The inter-cluster distance **Average Separation** (Sep_{AL}) measures the average separation distance between all clusters, according to Eq. A.12. Sep_{AL} should be maximized to obtain better clustering (Ripon and Siddique, 2009).

$$Sep_{AL}(\pi) = \frac{1}{k(k-1)/2} \sum_{i \neq j}^{k} d(\mathbf{z}_i, \mathbf{z}_j), \tag{A.12}$$

Sert et al. (Sert et al., 2011, 2012) introduce the use of **K-Mode external distance** (Km_{ed}) and **K-Mode weighted external distance** (Km_{wed}) as objective functions. These measures are similar to Sep_{AL}, however considering the mode instead of the centroid. Km_{ed} and Km_{wed} should be maximized as objective functions.

The **Separation Index** (Sep_{CL}) is computed by the sum of the distance between every two tuples (data points) in different clusters, according to Eq. A.13. It should be maximized to get well-separated clusters (Dutta et al., 2012b).

$$Sep_{CL}(\pi) = \sum_{\mathbf{c}_i \mathbf{c}_j \in \pi, i \neq j} \sum_{\mathbf{x}_a \in \mathbf{c}_i, \mathbf{x}_b \in \mathbf{c}_j} d(\mathbf{x}_a, \mathbf{x}_b) \tag{A.13}$$

The **graph-based separation** index (Sep_{graph}) measures the separation between the clusters in terms of a similarity graph. As in the DCD index, it considers the generation of a K_{size}-Graph as the first step in computing this index. The Sep_{graph} is calculated as the arithmetic average value of the edge weights between the different clusters, as defined in Eq. A.14, where $\mathbf{c}$ is a cluster, $\mathbf{G}$ is the K_{size}-Graph, $\mathbf{v}_a$ is the vertex a, and w_{ab} is the edge weight value from node a to node b. Sep_{graph} should be maximized to improve cluster separation (Menéndez et al., 2013).

$$Sep_{graph} = \left(\frac{\sum_{\mathbf{v}_a \in \mathbf{G}} \{ w_{ab} | \mathbf{v}_a \notin \mathbf{c} \}}{\mathbf{G} - \mathbf{c}} \right) / \mathbf{c} \tag{A.14}$$

The **Fuzzy Separation** (Sep_{fuzzy}) index (Mukhopadhyay et al., 2007) measures the inter-cluster fuzzy separation. This index is computed according to Eq. A.15, where the fuzzy membership is defined by μ_{ij}, $d(\mathbf{z}_j, \mathbf{z}_i)$ is the distance between two centroids $\mathbf{z}_i$ and $\mathbf{z}_j$. To get well-separated clusters, the Sep_{fuzzy} should be maximized.

$$Sep_{fuzzy} = \sum_{\substack{i,j=1, \\ i \neq j}}^{k} \mu_{ij}^{m} d(\mathbf{z}_i, \mathbf{z}_j), \text{ where } \mu_{ij} = 2 / \left(\sum_{\substack{l=1, \\ l \neq j}}^{k} \left(\frac{d(\mathbf{z}_j, \mathbf{z}_i)}{d(\mathbf{z}_j, \mathbf{z}_l)} \right)^{1/(m-1)} \right) \tag{A.15}$$

The **Fuzzy Overlap Separation** (Sep_{nfuzzy}) considers the combination of the l-order overlap and inter-cluster separation, composed of a t-normal function $\top$ and t-conorm $\perp$ to formulate the Fuzzy Overlap Separation (Wikaisuksakul, 2014; Paul and Shill, 2018). Sep_{nfuzzy} is defined in Eq. A.16, where u_{ai} is the membership degree of the ath data point to the ith cluster, $O\perp(\mathbf{u}_a(\mathbf{x}_a), k)$ is the overlapping degree that considers triplets of clusters up to a k-tuple of clusters combinations. Sep_{nfuzzy} index measures the isolation of clusters, which is preferred to be large.

$$Sep_{nfuzzy} = \frac{1}{n} \sum_{a=1}^{n} \frac{O\perp(\mathbf{u}_a(\mathbf{x}_a), k)}{\max\limits_{i=1,k} \{u_{ai}\}} \tag{A.16}$$

A.1.4 Separation and Compactness criteria

The **Categorical Data Clustering with Subjective factors** ($CDCS$) index is computed by the ratio of the intra-cluster cohesion and inter-cluster similarity for the categorical data clustering. This index is defined by Eq. A.17, where $\mathbf{A}_r$ is a set of attribute values, a_r denotes the number of attribute values for the rth attribute, $P(\mathbf{A}_r = a_r^i | \mathbf{c}_i)$ is the probability of a_r^i for the rth attribute in cluster $\mathbf{c}_i$, $S(\mathbf{c}_p, \mathbf{c}_q)$ denotes a similarity of two clusters, where $S(\mathbf{c}_p, \mathbf{c}_q) = \prod_{r=1}^{f} \left[\sum_{i}^{t_r} \min P(\mathbf{A}_r = a_r^i | \mathbf{c}_p), P(\mathbf{A}_r = a_r^i | \mathbf{c}_q) + \varepsilon \right]$, and ε is a small value in case that each component is 0 (Zhu and Xu, 2018).

$$CDCS = \frac{intra}{inter}, \text{ where}$$
$$intra = \sum_{i=1}^{k} \frac{|\mathbf{c}_k|}{n} \sum_{r=1}^{f} \frac{1}{f} (\max_{i=1}^{n_r} P(\mathbf{A}_r = a_r^i | \mathbf{c}_i))^3, inter = \frac{\sum_{p=1}^{k} \sum_{q=1}^{k} S(\mathbf{c}_p, \mathbf{c}_q)^{1/f} \cdot |\mathbf{c}_p \cup \mathbf{c}_q|}{(k-1) \cdot n} \tag{A.17}$$

The **Calinski-Harabasz** (CH) index, also known as the variance ratio criterion, is based on the degree of dispersion between clusters. It can take values in [0, ∞] with higher values indicating better clustering. CH is computed by the ratio of the sum of between-cluster dispersion and inter-cluster dispersion for all clusters, as defined in Eq. A.18 (Zhu et al., 2018).

$$CH(\pi) = \frac{\sum_{i=1}^{k} n_i \cdot d(\mathbf{z}_i, \bar{\mathbf{z}})}{\sum_{i=1}^{k} \sum_{\mathbf{x} \in \mathbf{c}_i} d(\mathbf{x}, \mathbf{z}_i)} \frac{(n-k)}{(k-1)} \tag{A.18}$$

The **Davies-Bouldin** (DB) index is computed as the ratio of the sum of within-cluster scatter to between-cluster separation (R_i), as defined in Eq. A.19. The minimum value of this DB is zero, with lower values indicating a better clustering (Tsai et al., 2012; Zhu et al., 2018; Dong et al., 2018; Dutta et al., 2019).

$$\begin{gathered} DB(\pi) = \frac{1}{k} \sum_{i=1}^{k} R_i, \text{ where} \\ R_i = \max_{j, j \neq i} \left\{ \frac{S_i + S_j}{d(\mathbf{z}_i, \mathbf{z}_j)} \right\}, \text{ and } S_i = \frac{1}{|n_i|} \sum_{\mathbf{x_a} \in \mathbf{c}_i} d(\mathbf{x_a}, \mathbf{z}_i) \end{gathered} \tag{A.19}$$

The **Dunn** index is computed as the ratio between the minimum inter-cluster distance ($\delta(\mathbf{c}_i, \mathbf{c}_j)$) to the maximum cluster diameter ($\max_{j \leq i \leq k} \Delta(\mathbf{c}_i)$), as defined in Eq. (A.20). It is considered that compact and well-separated clusters have a small diameter and a large distance between them. The Dunn index can take values between zero and infinity, and it should be maximized to obtain a well-separated and compact cluster (Liu et al., 2010).

$$\begin{gathered} Dunn(\pi) = \min_{1 \leq i \leq k} \left\{ \min_{\substack{1 \leq j \leq k, \\ j \neq i}} \left\{ \frac{\delta(\mathbf{c}_i, \mathbf{c}_j)}{\max_{j \leq i \leq k} \Delta(\mathbf{c}_i)} \right\} \right\}, \text{ where} \\ \delta(\mathbf{c}_i, \mathbf{c}_j) = \min_{\substack{\mathbf{x}_a \in \mathbf{c}_i, \\ \mathbf{x}_b \in \mathbf{c}_j}} \{d(\mathbf{x}_a, \mathbf{x}_b)\}, \text{ and } \Delta(\mathbf{c}_i) = \max_{\mathbf{x}_a, \mathbf{x}_b \in \mathbf{c}_i} \{d(\mathbf{x}_a, \mathbf{x}_b)\} \end{gathered} \tag{A.20}$$

The **Modularity** (Mod) was initially proposed as a measure of the strength of the network's module division. This index is computed as the total difference between the sum of distances of the objects in the same cluster $\mathbf{c}_i$ (that indicates how closely similar data is with others in the same cluster) and the sum of distances considering the objects in the dataset $\mathbf{X}$ (that determines how closely similar data is with others in different clusters), as defined in Eq. A.21 (Liu et al., 2018).

$$\begin{gathered} Mod(\pi) = \sum_{i=1}^{k} (cd - od^2), \text{ where} \\ cd = \frac{\sum_{\mathbf{x}_a, \mathbf{x}_b \in \mathbf{c}_i} d(\mathbf{x}_a, \mathbf{x}_b)}{\sum_{\mathbf{x}_a, \mathbf{x}_b \in \mathbf{X}} d(\mathbf{x}_a, \mathbf{x}_b)}, \text{ and } od = \frac{\sum_{\mathbf{x}_a \in \mathbf{c}_i, \mathbf{x}_b \in \mathbf{X}} d(\mathbf{x}_a, \mathbf{x}_b)}{\sum_{\mathbf{x}_a, \mathbf{x}_b \in \mathbf{X}} d(\mathbf{x}_a, \mathbf{x}_b)} \end{gathered} \tag{A.21}$$

The **Silhouette** (Sil) index measures how much each point in the data is similar to its own cluster compared to other clusters, based on the relation of the mean similarity of the objects within a cluster and the mean distance to the objects in the other clusters. Sil is defined in Eq. A.22, in which ad_a refers to the mean distance between a sample $\mathbf{x}_a$ and all other points in the same cluster. Moreover, bd_a is the mean distance between a sample $\mathbf{x}_a$ and the nearest cluster that $\mathbf{x}_a$ is not a part of. Thus, Sil produces values between -1 and 1. A higher value corresponds to a better clustering result (Mukhopadhyay and Maulik, 2007).

$$Sil(\pi) = \frac{1}{n}\sum_{a=1}^{n} S(\mathbf{x}_a), \text{ where } S(\mathbf{x}_a) = \frac{bd_a - ad_a}{max\{ad_a, bd_a\}} \tag{A.22}$$

The $\mathcal{I}$ index measures separation based on the maximum distance between cluster centers, and measures compactness based on the sum of distances between objects and their cluster centers. This index is computed according to Eq. A.23, in which E_k stands for within cluster scatter, D_k stands for between-cluster separation, $E1$ and P are correlation coefficients, u_{ia} is the membership degree of the ath object to the ith cluster. A larger value of this index implies better clustering. (Dong et al., 2018).

$$\begin{gathered}\mathcal{I} = \left(\frac{1}{k} \cdot \frac{E_1}{E_k} \cdot D_k\right)^P, \text{ where} \\ D_k = \max_{i,j=1}^{k}(\mathbf{z}_i - \mathbf{z}_j), \text{ and } E_k = \sum_{i=1}^{k}\sum_{a=1}^{n} u_{ia}(\mathbf{x}_a - \mathbf{z}_i)\end{gathered} \tag{A.23}$$

The **Addition feature weight** (J_{Add}) index is applied to minimize both the negative weight entropy and the separation between clusters. This index is defined in Eq. A.24, where f is the number of attributes, and w_{ir} takes the value in [0, 1], which corresponds to a soft partition of features. It is composed by Sep_i, that is computed according to Eq. (A.15), σ a present value that prevents the denominator from becoming zero, and A_{wi} denotes the average value of the important weights, which are more than or equal to the mean value $(1/f)$ for the ith cluster (Xia et al., 2013).

$$J_{Add} = \sum_{i=1}^{k}\left(\frac{Aw_i}{(Sep_i + \sigma)} + \sum_{r=1}^{f} w_{ir}\log w_{ir}\right), \text{ where} \tag{A.24}$$

The **Pakhira-Bandyopadhyay-Maulik** index (PBM) is defined in Eq. A.25, where E measures the total within-cluster scatter, E_0 is the total scatter considering all the samples belonging to one single cluster, and D is the maximum distance between cluster centers. Furthermore, μ_{ai} denotes the membership degree of the objects in a cluster, which can take values between 0 and 1. In our experiments, we considered a hard clustering, in which each object either belongs to a cluster completely ($\mu_{ai} = 1$) or not ($\mu_{ai} = 0$). The PBM must be maximized as objective function.

$$\begin{gathered}PBM = \frac{1}{k} \cdot \frac{E_0}{E_k} \cdot D_k \text{ where } E_0 = \sum_{a=1}^{n} d(\mathbf{x}_a, \overline{\mathbf{z}}), E_k = \sum_{i=1}^{k} E_i, \\ E_i = \sum_{a=1}^{n}\sum_{i=1}^{k} \mu_{ai} \cdot d(\mathbf{x}_a, \mathbf{c}_i)^2, \text{and } D_k = \max_{i,j=1,i\neq j}^{k} d(\mathbf{z}_i, \mathbf{z}_j)\end{gathered} \tag{A.25}$$

The **Xeni-Beny** (XB) index is defined as a function of the ratio of the total fuzzy cluster variance (J_m) to the minimum separation of the clusters (Sep), as presented in Eq. A.26, where u_{ia} is the membership degree of the ath data point to the ith cluster, and m is the fuzzy exponent. It should be minimized to obtain well-separated and compact clusters (Di Nuovo et al., 2007; Zhu and Xu, 2018).

$$XB(\pi) = \frac{J_m}{n \cdot sep} = \frac{\sum_{i=1}^{k} \sum_{a=1}^{n} u_{ia}^{m} d(\mathbf{z}_i, \mathbf{x}_a)}{n \cdot (\min\limits_{i \neq j} \{d(\mathbf{z}_i, \mathbf{z}_j)\})} \tag{A.26}$$

The **Soft Subspace Xie-Beni** ($SSXB$) index was extended from the XB, and defined as the ratio of the fuzzy weighting within-cluster compactness (J_{wm}) to the fuzzy minimum weighting between-cluster separation (J_{wsep}). This index is computed according to Eq. A.27, and it should be minimized as an objective function (Zhu et al., 2012).

$$SSBX(\pi) = \frac{J_{wm}}{n \cdot J_{wsep}} = \frac{\sum\limits_{i=1}^{k} \sum\limits_{a=1}^{n} u_{ia}^{2} \sum_{r=1}^{f} w_{ir}^{\tau} d(\mathbf{x}_{ar} - \mathbf{z}_{ir})^2}{n \cdot \min_{i \neq j} \{d^2(\mathbf{z}_{ir}, \mathbf{z}_{jr})\}} \tag{A.27}$$

where $d^2(\mathbf{z}_{ir}, \mathbf{z}_{jr}) = (\sum\limits_{r=1}^{f} w_{ij}^{\tau} d(\mathbf{z}_{ir} - \mathbf{z}_{jr})^2 + \sum\limits_{r=1}^{f} w_{ij}^{\tau} d(\mathbf{z}_{ir} - \mathbf{z}_{jr})^2)/2$, f is the number of attributes. w_{ir} and u_{ia} are defined in Eq. A.9.

A.1.5 Other criteria

Here, we present the other criteria applied as objective functions. Cluster cardinality and expected weighted coverage density indices consider the relation between the occurrence of objects in a categorical dataset. The similarity index is the only relative CVI used as the objective function, while the other CVIs consider the data properties of each partition. The sparsity and reconstruction error are two particular objective functions designed for spectral clustering.

The **Cluster Cardinality Index** (CCI) considers a set of operations to describe the property and structure of categorical data (Zhu and Xu, 2018). It is computed according to Eq. A.28, where $\mathbf{A}_{lr}$ and $\mathbf{A}_{ir}$ are the set of categorical values of rth attribute within the clusters $\mathbf{c}_i$ and $\mathbf{c}_l$. A larger value of CCI implies better clustering.

$$\begin{aligned} CCI &= \frac{1}{k} \sum_{i=1}^{k} \max_{l, i=1, l \neq i} \left(\frac{CI(i) + CI(l)}{CI(i,l)} \right), \text{ where} \\ CI(i) &= \frac{1}{f} \sum_{r=1}^{f} \frac{|\mathbf{A}_{ir}|}{\mathbf{c}_i}, \text{ and } CI(i,l) = \frac{1}{f} \sum_{r=1}^{f} \frac{|\mathbf{A}_{ir} \cap \mathbf{A}_{lr}| - |\mathbf{A}_{ir} \cup \mathbf{A}_{lr}| + 1}{|\mathbf{A}_{ir} \cap \mathbf{A}_{lr}| + 1} \end{aligned} \tag{A.28}$$

The **intra-cluster Expected Weighted Coverage Density** ($EWCD$) considers the relation between the objects in a transational dataset. The transational dataset is composed of n transactions considering the set of items $\mathbf{I} = \{\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_m\}$, where the transaction $\mathbf{t}_j (1 \leq j \leq n)$ is a set of items $\mathbf{t}_j = \{\mathbf{I}_{j1}, \mathbf{I}_{j2}, \ldots, \mathbf{I}_{jl}\}$, such that $\mathbf{t}_j \subseteq \mathbf{I}$. In this context, the WCD-Weighted Coverage Density of one cluster is defined as the sum of occurrences of all items in a cluster divided by the number of distinct items and the total number of items in this cluster. Thus, the EWCD of the partition π is defined as a average sum of the WCD in all clusters, as presented in the Eq. A.29, where $\mathbf{I}_{ij}$ is the jth item set in the cluster $\mathbf{c}_i$, $occur(\mathbf{I}_{ia})$ define the number of occurrences of the ath item in cluster $\mathbf{c}_i$, and S_i is the sum occurrences of all items in cluster $\mathbf{c}_i$ (Sert et al., 2011, 2012).

$$EWCD(\pi) = \sum_{i=1}^{k} \frac{n_i}{n} WCD = \frac{1}{n} \sum_{i=1}^{k} \left[\frac{\sum_{a=1}^{n_i} occur(\mathbf{I}_{ia})^2}{S_i} \right] \tag{A.29}$$

Li et al. (Li et al., 2017) introduced the **Similarity** (*Sim*) index to evaluate the similarity of one partition to others with a similarity matrix, as defined in Eq. A.30. This index can be used to evaluate the diversity of the solutions in an evolutionary approach. It should be minimized as an objective (Li et al., 2017).

$$Sim = \frac{1}{n} \sum_{j=1}^{n} similarity(\pi_i, \pi_j), \tag{A.30}$$

Luo et al. (Luo et al., 2015) modeled the similarity matrix for spectral clustering into objective functions. They assume that $\mathbf{y} = \mathbf{Ax}$ is a linear equation of an under-determined system, where $\mathbf{A} \in \mathbb{R}^{M \cdot N}$ is a full-rank and over-complete matrix, which is called an over-complete dictionary, $\mathbf{y} \in \mathbb{R}^M$ is called a measurement vector, and $\mathbf{x} \in \mathbb{R}^N$ is a sparse vector. Thus, they use $\mathbf{x}$ and $\mathbf{A}$ to reconstruct $\mathbf{y}$. For that, the **SParsity** (*SP*), Eq. A.31, and **Reconstruction Error** (*RE*), Eq. A.32, should be minimized.

$$SP = \|\mathbf{x}\|_0 , \tag{A.31}$$

where l_0 norm $\|.\|_0$ counts the number of nonzero values in a vector.

$$RE = \|\mathbf{Ax} - \mathbf{A}\|_2^2 , \tag{A.32}$$

where $\|.\|_2^2$ is the Euclidean norm on signals of a square matrix.

APPENDIX B – ADDITIONAL EXPERIMENTS

B.1 ANALYSIS OF LOCUS END Δ-LOCUS ENCODING

In this section, we analyzed the impact of the increase in iterations when the locus (Handl et al., 2007) and Δ-locus (Garcia-Piquer et al., 2017) representations are used in the optimization of (Var, Con') in the NSGA-II with neighborhood-based mutation and uniform crossover. For that, we considered the datasets in G2 and G3 presented in Section 7, and included other datasets (`Long`, `Spiral`, `Twenty`, `Complex9`, `R15`, `Sph_5_2`, `Square4`, `D31`, and `Sph_9_2`) presented in Section 4.2.3. We do not present the results of G1 because the MST-clustering provides the optimal results for the datasets in this group, and the optimization is not required.

Table B.1 present the best average ARI of each dataset, in which the use of the Δ-locus generated a loss of ARI in some datasets. Even with the use of $\delta = 0$, in which the encoding is not reduced, there is a general loss of the ARI. By analyzing the general data structure of the locus and Δ-locus with $\delta = 0$, we observed that the main difference between them is the arrangement of the edges in the initialization. In Δ-locus the encoding is configured to determine the fix and relevant edges, thus all edges are ordered considering the DI and the MST structure is modified. Since this is the only difference between these representations, we consider that it contributed to the loss of ARI. However, this modification was great enough to affect the ARI results. Thus, we consider the locus representation in our approach. That allows us to analyze the impact of using different objective functions and evolutionary operators without the inference of the Δ encoding in the results.

G	Gen. / Dat.	Locus				Δ-locus ($\delta = 0$)				Δ-locus ($\delta =\sim \sqrt{5}$)			
		100	**150**	**200**	**250**	**100**	**150**	**200**	**250**	**100**	**150**	**200**	**250**
G2	Aggregation	**0.9945**	0.9941	0.9943	0.9943	0.9871	0.9911	0.9927	**0.9941**	0.9291	0.9911	0.9927	**0.9941**
	Complex8	**0.9292**	0.9221	0.9195	0.9225	**0.9066**	0.8959	0.8960	0.8899	0.8992	0.9071	0.9088	**0.9112**
	Complex9	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9363	**0.9431**	0.9231	0.9090	**0.9596**	0.9431	0.9231	0.9090
	2d_10c_no9	0.9762	0.9760	0.9753	**0.9766**	0.9415	0.9414	0.9445	**0.9472**	0.9579	**0.9609**	0.9551	0.9608
	2d_4c_no2	0.9906	**0.9908**	**0.9908**	**0.9908**	0.9851	0.9870	0.9887	**0.9890**	**0.9884**	**0.9888**	**0.9888**	**0.9888**
	R15	**0.9924**	**0.9924**	**0.9924**	**0.9924**	**0.9569**	0.9611	0.9605	0.9657	**0.9860**	0.9611	0.9605	0.9657
	Sph_5_2	0.9150	0.9145	**0.9172**	0.9146	0.8771	0.8824	0.8828	**0.8873**	**0.8989**	0.8824	0.8828	0.8873
	Sph_10_2	**0.9790**	**0.9790**	**0.9790**	0.9782	0.9425	0.9431	0.9425	**0.9437**	**0.9643**	0.9431	0.9425	0.9437
	Spiralsquare_S1	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.7856	**0.8284**	**0.8284**	**0.8284**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	Spiralsquare_S2	0.9985	**0.9986**	**0.9986**	0.9985	0.9599	0.9635	**0.9640**	0.9637	**0.9970**	0.9635	0.9640	0.9637
	Sizes5	**0.9692**	0.9650	0.9634	0.9619	0.9448	0.9521	0.9549	**0.9573**	**0.9649**	0.9521	0.9549	0.9573
G3	D31	0.6894	0.7625	0.7999	**0.8428**	0.6893	0.7139	0.7350	**0.7418**	0.7202	0.7139	0.7350	**0.7418**
	ds4c2sc8	0.9003	**0.9088**	0.9065	0.9041	0.8408	0.8452	0.8559	**0.8663**	0.8737	0.8826	0.8848	**0.8886**
	DS-850	0.9987	0.9987	**0.9990**	**0.9990**	0.9767	0.9869	0.9849	**0.9947**	**0.9993**	0.9869	0.9849	0.9947
	Square1	0.9758	**0.9762**	0.9758	0.9757	0.9688	0.9685	**0.9695**	0.9691	**0.9744**	0.9685	0.9695	0.9691
	Square4	0.7897	0.7903	0.7916	**0.7939**	0.7795	0.7822	**0.7865**	0.7852	0.7686	0.7822	**0.7865**	0.7852
	Flame	**0.9644**	0.9610	0.9561	0.9506	**0.9771**	0.9717	0.9746	0.9713	0.9622	**0.9720**	0.9639	0.9551
	Pathbased	**0.7263**	0.7213	0.7184	0.7148	0.7094	0.7041	**0.7147**	0.7070	0.7091	**0.7094**	0.7046	0.6925
	Engytime	0.8056	0.8137	0.8117	**0.8170**	0.8032	0.7979	**0.8047**	0.8033	0.7935	0.7979	**0.8047**	0.8033
	Sph_9_2	0.7514	0.7563	0.7615	**0.7634**	0.6771	0.6999	0.7070	**0.7045**	**0.7296**	0.6999	0.7070	0.7045
	Triangle2	**0.9878**	0.9871	0.9868	0.9869	0.9562	0.9784	**0.9791**	0.9785	0.9828	**0.9838**	0.9830	0.9825
	Twodiamonds	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9965	0.9973	0.9978	**0.9980**	**0.9993**	0.9985	0.9985	0.9985
	Mean G2	**0.9768**	**0.9757**	0.9755	0.9754	0.9294	**0.9354**	0.9344	0.9341	**0.9587**	0.9539	0.9521	0.9529
	Mean G3	0.8718	0.8796	0.8825	**0.8862**	0.8522	0.8587	0.8645	**0.8654**	0.8648	0.8632	**0.8657**	0.8651

Table B.1: Best average ARI considering locus, Δ-locus=$\sim \sqrt{5}$ and Δ-locus= 0, different number of generations and (Var, Con') as objective function (Average of 10 executions).

B.2 ANALYSIS OF DIFFERENT CROSSOVERS AND OBJECTIVE FUNCTIONS

In this section, we present the results of experiments considering different crossover operators (One Point, Two Points, and Uniform), associated with the two pairs of objective functions, (Var, Con') or (Sep_{CL}, Con') to evaluate how different components of the MOEA can contribute the optimization of low quality base partitions.

In Tables B.2 and B.3 present the best average ARI considering different crossover operators. In these tables, OP denote the one point crossover, TP represent the two points crossover, UN denote the Uniform crossover, and PO represent these three crossover operator in a pool selected randomly.

Datasets	(Var, Con')				(Sep_{CL}, Con')			
	OP	**TP**	**UN**	**PO**	**OP**	**TP**	**UN**	**PO**
D31	0.8891	**0.8936**	0.8364	0.8894	0.8819	0.8895	0.8729	**0.8921**
ds4c2sc8	0.8908	0.8953	**0.9041**	0.9015	0.9101	0.9113	**0.9148**	0.9147
DS-850	**1.0000**	0.9997	0.9997	**1.0000**	0.9993	0.9993	0.9993	**0.9994**
Engytime	0.8153	**0.8166**	0.8148	0.8111	0.8141	0.8196	0.8210	**0.8256**
Flame	0.9495	0.9544	0.9506	**0.9565**	**0.9717**	0.9678	0.9697	0.9710
Pathbased	0.7119	0.7087	**0.7148**	0.7101	0.7855	0.7991	**0.8227**	0.8133
Sph_9_2	0.7447	0.7491	0.7630	**0.7658**	0.7347	0.7507	**0.7663**	0.7422
Square1	0.9760	0.9761	0.9754	**0.9767**	0.9732	0.9748	**0.9767**	0.9752
Square4	**0.7961**	0.7956	0.7949	0.7918	0.7786	0.7825	**0.7925**	0.7832
Triangle2	**0.9874**	0.9852	0.9869	0.9866	0.9854	0.9841	0.9864	**0.9866**
Twodiamonds	**1.0000**	0.9995	**1.0000**	0.9995	0.9995	0.9995	**1.0000**	**1.0000**
MEAN	0.8873	0.8885	0.8855	**0.8899**	0.8940	0.8980	**0.9021**	0.9003

Table B.2: Best Average ARI considering different crossovers with (Var, Con') or (Sep_{CL}, Con') (Average of 10 executions).

In particular, Table B.2 present the result of the different crossover considering (Var, Con') and (Sep_{CL}, Con') as objective functions. The boldface values indicate the best ARI results found considering each pair of objective functions. For (Var, Con'), we can observe that the Uniform crossover provided the worst Mean result considering all the datasets. The result for D31 contributes for this overall result. Besides that, the pool of crossover operators provided the best mean result.

In contrast, (Sep_{CL}, Con') have higher mean ARI results for all the datasets, indicating that this pair of objective functions is more adequate than (Var, Con') for this group of datasets. It is confirmed by the Friedman test, that point out that the (Sep_{CL}, Con') with the uniform crossover is significantly better than (Var, Con') with the pool of crossover operators.

Considering that (Sep_{CL}, Con') obtained the best ARI results, in Table B.3 we analyzed this pair of objective function with of different size of the neighborhood (L) applied mutation operator and the computation of Con'.

We can observe that using L equals to $\sqrt{n} \cdot 50\%$ instead 10 cause a increase of the ARI. Demonstrating that, a refine of this parameter setting can impact in the results of some datasets, such as D31. However, considering the Friedman test, for the analyzed datasets, there is not significant difference between using $\sqrt{n} \cdot 50\%$ or 10.

In the following experiments we consider (Sep_{CL}, Con') for the low quality partitions. Thus, we define the selection of different objective functions for each quality group, in which (Var, Con') is applied in optimizing middle quality base partitions and (Sep_{CL}, Con') in low quality base partitions. Furthermore, we applied $L = 10$ for all the experiments of the QEMOC.

Datasets	L=10		L=$\sqrt{n} \cdot 25\%$		L=$\sqrt{n} \cdot 50\%$		L=$\sqrt{n} \cdot 75\%$	
	UN	**PO**	**UN**	**PO**	**UN**	**PO**	**UN**	**PO**
D31	0.8729	0.8921	0.8774	0.8894	0.8429	**0.9122**	0.8296	0.9119
ds4c2sc8	**0.9148**	0.9147	0.9023	0.8841	0.9140	0.9087	0.9097	0.9130
DS-850	0.9993	0.9994	0.9987	0.9994	**1.0000**	**1.0000**	0.9965	0.9965
Engytime	0.8210	**0.8256**	0.8205	0.8043	0.8168	0.8237	0.8104	0.8186
Flame	0.9697	0.9710	0.9666	0.8397	0.9718	**0.9736**	0.9592	0.9705
Pathbased	**0.8227**	0.8133	0.7716	0.6876	0.8217	0.8224	0.7088	0.7616
Sph_9_2	0.7663	0.7422	0.7484	0.7486	0.7707	0.7661	0.7917	**0.7729**
Square1	0.9767	0.9752	0.9722	0.9765	0.9782	0.9779	0.9782	**0.9790**
Square4	0.7925	0.7832	0.7799	0.7871	0.7991	0.7965	**0.8046**	0.8015
Triangle2	0.9864	0.9866	0.9873	0.9868	0.9891	**0.9900**	0.9864	0.9855
Twodiamonds	**1.0000**	**1.0000**	0.9998	0.9950	0.9975	0.9968	0.9978	0.9965
MEAN	0.9021	0.9003	0.8931	0.8726	0.9002	**0.9062**	0.8884	0.9007

Table B.3: Best Average ARI considering Uniform Crossover and Pool Crossovers with different size of neighborhood (Average of 10 executions).

APPENDIX C – IMPROVED CONNECTIVITY INDEX

In this section, we present the publication regarding the application of the improved connectivity index: Detecting Nested Structures Through Evolutionary Multi-objective Clustering.

Detecting Nested Structures Through Evolutionary Multi-objective Clustering *

Cristina Y. Morimoto(✉)[1][0000−0003−4122−5698], Aurora Pozo[1][0000−0001−5808−3919], and Marcílio C. P. de Souto[2][0000−0002−7033−8328]

[1] Federal University of Paraná, Curitiba-PR, Brazil cristina.morimoto@ufpr.br, aurora@inf.ufpr.br
[2] LIFO/University of Orléans, Orléans, France
marcilio.desouto@univ-orleans.fr

Abstract. The evolutionary multi-objective algorithms have been widely applied for clustering. However, in general, the detection of heterogeneous nested clusters remains challenging for clustering algorithms. This paper proposes an adaptation of the connectedness criterion used as an objective function in established Evolutionary Multi-Objective Clustering approaches (EMOCs). This adaptation can improve the conflict between the objective functions, and then it promotes the detection of nested clusters. We performed experiments with four EMOCs (MOCK, MOCLE, Δ-MOCK, and EMO-KC) that provide different features. These different EMOCs have different initialization methods and representation schemes, allowing us to analyze how the proposed objective function can contribute to detecting nested clusters. Our results show that our adapted objective function promotes a general gain in the performance of all these algorithms.

Keywords: Multi-objective clustering · Nested data clustering · Evolutionary multi-objective optimization · Clustering methods · Data mining

1 Introduction

Complex data allow multiple data interpretations in which multiple clustering approaches can describe alternative aspects that characterize the data in different views [14]. The Evolutionary Multi-Objective Clustering approaches (EMOCs) have been widely applied to extract patterns and provide these multiple views, allowing to analyze alternative aspects that characterize the data [9,6,13,8]. However, the use of EMOCs to detect nested structures is still under-explored in the literature, especially to detect heterogeneous data structures.

Some EMOCs were applied to detect heterogeneous data structures with the generation of solutions with multiple partitions [9,6,8]. They used multiple criteria (e.g., compactness and connectedness) as objective functions to deal with datasets with different types of clusters. However, no studies have widely evaluated them to detect nested data structures and analyze how their objective functions impact this task.

In this study, we propose a modification of the connectedness criterion adopted for established EMOCs to improve the detection of a different number of clusters, especially in nested clusters. The connectivity index used by these approaches has limitations to detect some multi-level solutions, such as nested clusters, in a single run. This modified objective function was evaluated in four EMOCs: MOCK [9], MOCLE [6], Δ-MOCK [8], and EMO-KC [16], in which we analyze how the different strategies adopted in these algorithms can contribute (or hamper) to detect nested clusters. We performed experiments on fifteen datasets, which yielded promising results using the modified connectivity in all these algorithms.

The remainder of this paper is organized as follows. In Section 2, we present the main concepts concerning MOCK, Δ-MOCK, MOCLE, and EMO-KC, considering their representation, initialization strategy, optimization strategy, objective functions, crossover, and mutation operators. In Section 3, we describe some general issues around the connectedness criterion used as an objective function in MOCK, Δ-MOCK, MOCLE and introduce the proposed modification in this index. Section 4 presents the datasets used in the experiments, the specific configuration and settings of the compared methods, and the performance assessment adopted. Then, in Section 5, we present and discuss the results of our experimental evaluation of the use of this modified connectivity index. Finally, Section 6 highlights our main findings and discusses future works.

* This work was partially supported by the National Council for Scientific and Technological Development (CNPq).

2 Background

A nested cluster refers to a cluster that is composed of sub-clusters or multi-level data structures. Formally, given a partition $\pi = \{\mathbf{c}_1, \ldots, \mathbf{c}_k\}$, for any $\mathbf{c}_a, \mathbf{c}_b \in \pi$ either they are non-overlapping ($\mathbf{c}_a \cap \mathbf{c}_b = \emptyset$) or one of them includes the other ($\mathbf{c}_a \subseteq \mathbf{c}_b$ or $\mathbf{c}_b \subseteq \mathbf{c}_a$), which is equivalent to assert that $\mathbf{c}_a \cap \mathbf{c}_b \in \{\emptyset, \mathbf{c}_a, \mathbf{c}_b\}$ [1]. For example, Fig. 1 depicts the Venn diagram of the nested data structures presented in the set $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\}$, where $\pi_1 = \{\{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\}\}$ and $\pi_2 = \{\{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4, \mathbf{x}_5\}\}$ represent the solutions at different levels in the hierarchy, in which π_1 has two clusters and π_2 has three clusters.

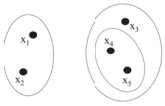


Fig. 1: Venn diagram of the nested data structures

Hierarchical clustering is the most traditional nested clustering strategy applied to produce a sequence of clusterings in which each cluster is nested into the next cluster in the sequence. This kind of approach presents a hierarchical grouping of the objects, which can be viewed as finding multiple partitions. However, the different clustering solutions obtained at different hierarchical levels differ only in their granularity [11,17].

Two well-known hierarchical clustering algorithms used in our analysis are the Single-Linkage (SL) and Group Average-Linkage (AL). In both the SL and AL algorithms, each object starts out standing as an individual cluster, and a series of merge operations is followed until it reaches the top with a single cluster. The main difference between these algorithms is the distance measure used to compute proximity between the pairs of clusters used to define the closest pair of sub-sets that are merged. SL considers the minimal distance between two objects of a cluster pair to define the closest sub-sets, and AL considers the average distance of all observations of pairs of clusters [17].

In contrast to hierarchical clustering, the EMOCs provide a diverse set of solutions considering different aspects of the data structures. This study analyzes the capabilities of four EMOCs to provide a diverse set of solutions that include nested clusters by using a modified connectedness criterion. The EMOCs analyzed are: MOCK [9], MOCLE [6], Δ-MOCK [8], and EMO-KC [16].

2.1 MOCK, Δ-MOCK, MOCLE and EMO-KC

MOCK (Multi-Objective Clustering with automatic K-determination) [9] and MOCLE (Multi-Objective Clustering Ensemble) [6] are well-known EMOCs. Δ-MOCK was introduced by [8] to improve the scalability of MOCK [9]. EMO-KC (Evolutionary Multi-objective Optimization-k-clustering) was described in [16], introducing an adapted sum of squared distances (SSD) to improve the generation of multiple solutions with a different number of clusters. These approaches present different representation encodings, initialization strategies, and/or evolutionary operators to optimize clustering criteria. In Section 5, we analyze how some of these different features can contribute to detecting different data structures, including nested clusters, based on a modified connectedness criterion (Section 3). In the following, we present more details of these EMOCs, considering the initialization strategy, representation encoding, optimization strategy, objective function, crossover, and mutation operators. Our analysis focuses on the ability of these algorithms to generate a set of solutions containing high-quality partitions. Thus, we will not be concerned with the selection of a final solution to be presented to the data expert.

Initialization Strategy. The generation of the initial population in MOCK consists of two methods: (i) Minimum Spanning Tree (MST) derived partitions, based on a measure called *degree of interestingness* (DI), and (ii) k-means (KM) [12] derived partitions. Δ-MOCK only uses one method to generate the initial population, the MST-derived partitions. In [6], MOCLE considered partitions generated by Single-Linkage (SL), Average-Linkage (AL), KM, and Shared Nearest Neighbor-based clustering (SNN) [5]. In contrast, EMO-KC considers a random choice of the points in the dataset to define the initial centroids.

Representation. MOCK introduced the locus-based adjacency graph representation, in which a solution is described as a vector of genes, and each gene g_i can take an integer value between 1 and n; if a value j is assigned to the ith gene, it can be interpreted as a link between the data points i and j, i.e., i and j belong to the same cluster. Δ-MOCK introduced two reduced locus-based adjacency graph representations, Δ-locus and Δ-binary; these schemes can significantly reduce the length of the genotype by using the concepts of MST and DI according to the length of the encoding defined by a user-defined parameter (δ). MOCLE uses a label-based encoding that considers labels for each object

in the partition. At last, EMO-KC uses a centroid-based encoding, in which the genes represent the coordinates of the cluster centroids.

Optimization Strategy. In terms of the optimization strategy, MOCK [9], MOCLE [6], Δ-MOCK [8], and EMO-KC [16] use traditional multi-objective evolutionary algorithms (MOEA) to optimize clustering criteria as objective functions. MOCK relies on the Pareto envelope-based selection algorithm version II (PESA-II) [3] in the optimization; in contrast, MOCLE, Δ-MOCK[8] and EMO-KC [16] use the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [4]. Both these MOEA, PESA-II and NSGA-II, use the Pareto dominance relation to rank and select the solutions in evolutionary optimization. The Pareto dominance is an important concept used in our analysis, that can be defined as follows: Let $\mathbf{x}_1$ and $\mathbf{x}_2$ be two feasible solutions; $\mathbf{x}_1$ is said to dominate $\mathbf{x}_2$ (denoted as $\mathbf{x}_1 \prec \mathbf{x}_2$), if the following two conditions are satisfied [2]: (i) $\forall i \in \{1, 2, \ldots, z\}$: $f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2)$, (ii) $\exists j \in \{1, 2, \ldots, z\}$: $f_j(\mathbf{x}_1) < f_j(\mathbf{x}_2)$.

Objective Functions. The analyzed EMOCs use two objective functions. MOCK and MOCLE optimize the overall deviation (*dev*) and connectivity index (*con*) as objective functions [9,6]. The *dev* is computed according to (1), where π represents a partition, $\mathbf{x}_i$ denotes an object in the cluster $\mathbf{c}_k$, $\boldsymbol{\mu}_k$ is the centroid of cluster $\mathbf{c}_k$, and $d(.,.)$ refers to the selected distance function.

$$dev(\pi) = \sum_{\mathbf{c}_k \in \pi} \sum_{\mathbf{x}_i \in \mathbf{c}_k} d(\mathbf{x}_i, \boldsymbol{\mu}_k), \tag{1}$$

The *con* is defined according to (2), where n is the number of objects in the dataset, L is the parameter that denotes the number of nearest neighbors that contributes to the connectivity, a_{ij} is the jth nearest neighbor of the object $\mathbf{x}_i$, and $\mathbf{c}_k$ is a cluster in the partition π.

$$\begin{aligned} con(\pi) &= \sum_{i=1}^{n} \sum_{j=1}^{L} f(\mathbf{x}_i, a_{ij}), \text{where} \\ f(\mathbf{x}_i, a_{ij}) &= \begin{cases} \frac{1}{j}, \text{if } \nexists \mathbf{c}_k : \mathbf{x}_i, a_{ij} \in \mathbf{c}_k \\ 0, \text{otherwise} \end{cases} \end{aligned} \tag{2}$$

Δ-MOCK also optimizes the *con*, but employs the intra-cluster variance (*var*) instead of the *dev* as an objective function. The *var* is defined according to (3), where π denotes a partition, n is the number of objects in the dataset, $\mathbf{x}_i$ is an object in the cluster $\mathbf{c}_k$, $\boldsymbol{\mu}_k$ is the centroid of the cluster $\mathbf{c}_k$, and $d(.,.)$ is the selected distance function [8].

$$var(\pi) = \frac{1}{n} \sum_{\mathbf{c}_k \in \pi} \sum_{\mathbf{x}_i \in \mathbf{c}_k} d(\mathbf{x}_i, \boldsymbol{\mu}_k)^2 \tag{3}$$

At last, EMO-KC optimizes an adapted sum of squared distances (SSD) and the number of clusters (k) as objective functions. None of the other EMOCs use the number of clusters as an objective function. In terms of the SSD, it is computed in the same way as (3) multiplied by n (the number of objects in the dataset). The adapted SSD, here denoted as var', is computed according to (4) [16].

$$var' = (1 - exp^{-1 \times (SSD)}) - k \tag{4}$$

All these objective functions should be minimized in the optimization.

Crossover and Mutation Operators. MOCK and Δ-MOCK use the standard uniform crossover and a neighborhood-biased mutation scheme [9]. MOCLE uses the Hybrid Bipartite Graph Formulation (HBGF) [7] as crossover operator, and no mutation is employed [6]. EMO-KC relies on the standard operators of the NSGA-II: simulated binary crossover and polynomial mutation [16].

3 An Improved Connectivity Index

In our studies, we verified that, according to the setting of the neighborhood size parameter (L), the connectivity index formulation could limit the detection of some data structures. For example, a dataset with well-separated nested data structures, as `ds2c2sc13` (Fig 2), can produce several solutions with optimal *con*. Consequently, when EMOCs select which solutions to keep, the decision will be taken based on the other criteria. For instance, if we consider optimizing two objective functions, where the *var* or *dev* is applied along with *con*, the partitions with a lower number of clusters would be discarded in the selection. The reason is that this solution is dominated by other solutions with lower *var* or *dev* because those solutions have a higher number of clusters. In term of the dataset `ds2c2sc13`, the algorithms that use these pairs of objective functions, such as [9,6,8], may not find the true partition[3] of the `S1` (when they use L=10), because it is dominated by `S2` — the true partition of the `S1`, Fig. 2a, has ($dev = 63.038$, $con = 0$) or ($var = 0.013$,

[3] The True Partition or ground truth is the labeled data that forms the real partition, the underlying structure of the data; and `S` denotes the hierarchy level of the partitions, in which `S1` represents the partition with the lowest number of clusters (a high-level partition), and a higher `S` refers to a partition with a low level of hierarchy.

$con = 0$); S2, Fig. 2b, has ($dev = 33.457$, $con = 0$) or ($var = 0.004$, $con = 0$); and S3, Fig. 2c, has ($dev = 24.075$, $con = 7.519$) or ($var = 0.002$, $con = 7.519$). A behavior that could also occur in other datasets with well-separated but no compact clusters.

It is important to note that the use of another setting for L can lead to the detection of the S1. However, it can generate the dominance of the other clustering levels. Thus, in order to deal with this problem, we propose in (5) a slight but effective modification of the definition of the *con* (2):

$$con'(\pi) = con(\pi) + \left(\frac{k}{n \times L}\right) \tag{5}$$

where k is the number of clusters in partition π, n is the number of objects in the dataset, and L is the number of nearest neighbors that contribute to connectivity. The term $(n \times L)$ ensures that the number of clusters k will be mapped to a value lower or equal to $\frac{1}{L}$, taking values in the interval $]0, \frac{1}{L}]$. That it is required to maintain the ordinal relationship between the best and the worst connectivity results. So, this modification will only affect the solutions that have the same outcome for the sum of the penalties of connectedness (2).

Intuitively, the new term added to *con* to yield con' will produce a new dominance relation that distinguishes the partitions with the same value of *con* but with a different number of clusters — a scenario that can occur in nested clusters, as in ds2c2sc13 dataset.

In other words, con' contains the information regarding the sum of the penalties of connectedness (the primary criterion), in the same way as the original *con*, added to the information about the number of clusters (the secondary criterion). The added information will affect the dominance evaluation of the solutions with the same outcome for the sum of the penalties of connectedness. Since the order relation regarding connectivity will only be modified in this group of solutions.

4 Experimental Design

This section presents the methodology employed to evaluate the adapted objective function, the used datasets, the experimental setup of the EMOCs, and the indicator applied for the performance assessment.

4.1 Datasets

Regarding the datasets, Table 1 summarizes the main characteristics of the fifteen datasets used in our experiments. In this table, n is the number of objects, d is the dimension of the dataset (number of attributes), S is the number of true partitions, i.e., the number of different levels of the (nested) data structures, and k* is the number of clusters of each data structure. S is also applied as an identifier for each true partition, where the associated number refers to the hierarchy level of the partitions, in which S1 represents the partition with the lowest number of clusters, and S4 the partition with the highest number of clusters. These datasets were divided into five groups (G1, G2, G3, G4, and G5), considering the general features evaluated in our analysis.

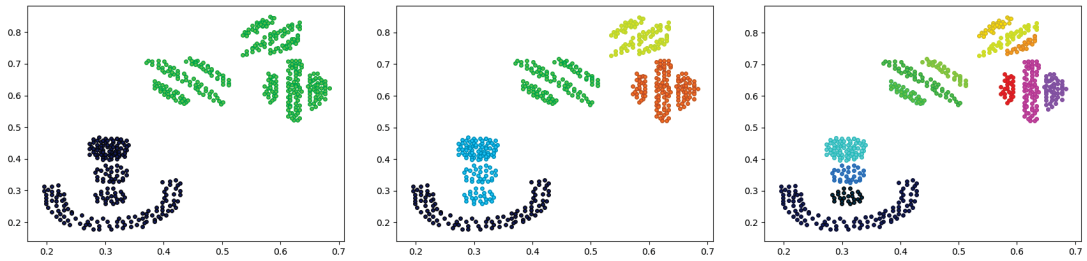
G1 and G2 contain datasets with several different properties, such as different data structures, number of observations, and distribution. G1 contains artificial datasets (20d-60c, Aggregation, D31), and G2 contains real datasets (Iris, Libras, UKC1). These groups are used to verify the general impact of the con' in comparison with *con* when applied in different datasets with a single true partition.

G3 contains artificial datasets with nested data structures and well-separated clusters (ds2c2sc13 and Spiralsquare). In this group, besides comparing the use of *con* and con', we will analyze the capabilities of the con' in the EMOCs in relation to the hierarchical clustering algorithms SL and AL.

G4 also contains artificial datasets with nested clusters, but they have several different properties, such as cluster shapes, distributions, and proximity between the clusters. In this group, we have the Monkey dataset and three new datasets, Bear, Glassesman and Stomata, Fig. 3. These three datasets were used for the first time here[4]. In particular, Bear and Glassesman contain different types of sub-sets, in which the lowest (hierarchy) level of structures (S3) contains nested clusters along with other sub-sets. For example, the overlapping clusters that represent the nose and mouth in the S3 of the dataset Glassesman are sub-sets of one general cluster, but such clusters are not nested at this level of data. The same occurs in the clusters that represent the eyes in the dataset Bear, they are sub-sets in the S3, but these clusters are not nested structures.

At last, G5 contains real datasets, (Golub, Glass, and Leukemia), that present more than one specified true partition, and may present nested data structures. The analysis of these groups will provide a general view of how the con' impacts the clustering performance of the EMOCs in complex datasets. As well, we will analyze the results of the EMOCs with regard to the hierarchical clustering algorithms SL and AL.

[4] Available at https://github.com/cymorimoto/newdatasets.



(a) `S1`, $k = 2$ (b) `S2`, $k = 5$ (c) `S3`, $k = 13$

Fig. 2: True partitions of the artificial dataset ds2c2sc13

#	Dataset	n	d	S	k	Description
G1	`20d-60c`	4,395	20	1	60	`20d-60c` has 60 ellipsoidal clusters with arbitrary elongation and orientation distributed in a 20-dimensional space.
	`Aggregation`	788	2	1	7	`Aggregation` consists of heterogeneous structures with clusters of varied sizes and shapes.
	`D31`	3,100	2	1	31	`D31` contains 31 equal sizes and spread clusters that are slightly overlapping and distributed randomly in a 2-dimensional space.
G2	`Iris`	150	4	1	3	`Iris` contains 3 clusters (types of iris plant) that contain an equal number of observations.
	`Libras`	360	90	1	15	`Libras` is composed of representations of different hand movements in the Brazilian Sign Language (LIBRAS).
	`UKC1`	29,463	2	1	11	`UKC1` is a dataset with a very large number of objects related to street-level crime in the U.K.
G3	`ds2c2sc13`	588	2	3	2, 5, 13	`ds2c2sc13` contains three different structures: S1 represents two well-separated clusters; S2 and S3 combine different types of clusters.
	`Spiralsquare`	1,500	2	2	2, 6	`Spiralsquare` contains two true partitions: S1 represents two well-separated clusters, and S2 contains 2 spirals and 4 Gaussian-like clusters.
G4	`Monkey`	4,000	2	4	2, 3, 5, 8	`Monkey` has a set of clusters with different sizes and shapes that represent a monkey head. S1 contains two major clusters. S2 and S3 present clusters with different granularities of the `S1`.
	`Bear`	1,480	2	3	2, 5, 11	`Bear` contains clusters with different dispersion and distributions, considering clusters obtained from the datasets `Pathbase` and `ds3c3sc6`.
	`Glassesman`	5,878	2	3	3, 4, 5	`Glassesman` contains heterogeneous structures with clusters of varied sizes and shapes, including clusters presented in the datasets `Engytime` and `twoDiamonds`.
	`Stomata`	2,376	2	3	2, 8, 16	`Stomata` was designed and inspired by the cells found in the epidermis of leaves, named stomata. It contains three data structures: S1 considers two internal cells surrounded by the other cells, S2 represents each cell as a cluster, S3 distinguishes the cells and their nucleus.
G5	`Glass`	214	9	3	2, 5, 6	`Glass` is a benchmark dataset, that contains glass attributes used to identify the type of glass.
	`Golub`	72	3,571	2	2, 3	`Golub` refers to gene expression data from the leukemia micro-array study.
	`Leukemia`	327	271	2	3, 7	`Leukemia` also refers to gene expression. Both `Golub` and `Leukemia` have a small number of objects (distributed in clusters of very different sizes), but a large number of attributes, typical of bio-informatics data.

Table 1: Dataset characteristics

The datasets `20d-60c` and `UKC1` were introduced in [8]. `ds2c2sc13`, `Glass`, `Golub`, `Iris`, `Leukemia`, `Libras`, `Monkey`, and `Spiralsquare` were obtained from the Clusters evaluation benchmark repository[5]. `D31` and `Aggregation` were obtained from the Clustering basic benchmark repository[6]. Besides that, the datasets that compose the `Bear` and `Glassesman` were obtained from these two repositories.

4.2 Experimental Setup

We employed the same general settings as reported in [9,8] to execute MOCK and Δ-MOCK. Regarding the Δ-MOCK representation, in this paper, we used the Δ-locus scheme with δ defined as a function of $\sim 5/\sqrt{n}$, where n is the number of objects in the dataset — this function is one of the heuristics employed in [8]. Concerning the MOCLE, we used the general setting as in [6], considering the NSGA-II as MOEA and HBGF as a crossover operator. At last, for EMO-KC, we applied the same general setting presented in [16]. Furthermore, for every approach, including the hierarchical algorithms SL and AL, we applied the Euclidean distance as a distance function, and we adjusted the other parameters required to produce partitions containing clusters in the range $\{2, 2k^*\}$, in the same way as MOCK and Δ-MOCK.

For the EMOCs, the L parameter applied in the *con* and con' was set $L = 10$ for all the experiments. Finally, as such algorithms are non-deterministic, we executed the experiments 30 times.

4.3 Performance Assessment

In this work, we use the adjusted Rand index (ARI) [10] as the indicator to measure the clustering performance. This indicator measures the similarity between two partitions. Thus, ARI is applied to compare the EMOCs results with the true partitions. ARI results close to 0 mean no correspondence between the partitions, and results close to 1 point out a high similarity between the partitions.

Besides, we use a non-parametric test to analyze the ARI results, the Kruskal-Wallis test with the Tukey-Kramer-Nemenyi post-hoc test [15] with significance level alpha=0.01. Such a test is applied to analyze the behavior of each algorithm on a different problem (dataset). Furthermore, we applied the Friedman and Bergmann-Hommel Post Hoc hypothesis test [15] with alpha=0.05. This last combination of tests is applied to compare the overall performance of the algorithms in the datasets with nested clusters.

5 Results and Discussion

In this section, we present the results of the performed experiments considering the comparison of the *con* and con' applied along with the original compactness index (dev, var, or var') in the EMOCs and compare them with the results of the hierarchical clustering methods SL and AL. Since EMO-KC originally did not use a connectedness index, we also performed experiments with its original objective functions (var', k) and associated them with the connectedness criterion. As described by [16], the var' was designed to provide more conflict around the number of the clusters; we consider that a general analysis of these objective functions can provide insights about the performance of the purpose modification of the *con* to produce conflict around the compactness criterion.

Table 2 presents the ARI of the best partition found by SL and AL, and the average ARI of the best partitions of MOCLE, MOCK, Δ-MOCK, and EMO-KC found in experiments using 2 objective functions, considering their original compactness criterion (dev, var, or var') associated with *con* and con', as objective functions. For EMO-KC, it also presents the results regarding the original objective functions presented in [16], (var', k). The ARI highlighted in boldface represents the best values found for each evolutionary multi-objective approach, considering the comparison of the different objective functions. Furthermore, underlined ARI points out the results with a significant difference according to the Kruskal-Wallis test. In the case of the SL and AL results, the ARI highlighted in boldface represents the result where these algorithms found the best ARI compared to the EMOCs.

The results point out that the con' improves the general performance of all the EMOCs, as shown in Table 2. The row **Significant Wins** presents the number of the datasets in which one pair of objective functions win over the other pair (as indicated by the statistical test), considering the objective functions with con' or *con*. For instance, for MOCLE, the objective functions (dev, con) has 2 significant wins while the pair (dev, con') has 3 significant wins. In general, Δ-MOCK and MOCK are algorithms that have the major significant wins with the con', where MOCK has 6 significant wins and Δ-MOCK 8 significant wins.

By analyzing each group of datasets, we obtained more details about how the con' impacted the results of the studied EMOCs. For example, in general, the use of the con' does not impact the results in the datasets with a single true partition, as the datasets present in G1 and G2, in which the results of all EMOCs were very close to that present with *con*. Only for the dataset `UKC1` (present in G2), the use of the con' provided a significant gain of ARI in MOCK and Δ-MOCK.

On the other hand, for the datasets with nested data structures and well-separated clusters, as presented in G3, we have the greatest improvement of the clustering results by using the con'. For example, the well-separated structures `S1` in the datasets `ds2c2sc13` and `SpiralSquare` were detected in all studied approaches, and Δ-MOCK was able to detect `S2` in the datasets `ds2c2sc13`.

A particular case occurred in the `S3` of the datasets `ds2c2sc13`, where the use of the con' caused a significant loss in the ARI in MOCLE and MOCK, when compared with the results of *con*. In this case, the parameter L is still impacting the dominance around the true partition. However, in MOCLE, our general results for this dataset are higher than others reported in the related work, as in [6] or results provided by hierarchical methods SL and AL.

In contrast, the results in the G4 and G5 were diverse. For example, we obtain an ARI gain in the `S1` of the dataset `Leukemia` in MOCK and Δ-MOCK. However, the dominance of the true partition and the influence of the L parameter are still impacting the results in the datasets `Monkey` and `Stomata`, in which we obtained gain of the ARI in some partitions and an ARI decrease in other ones. An analysis of the Pareto front (PF) of these datasets is presented in Section 5.1, to detail how the con' impacts the optimization and to explain these results. At last, for the other datasets in these two groups (G4 and G5) there are not any significant differences by using *con* or con'. Regarding the results of the SL and AL, in general, the best results found by them for G4 and G5 were worse or equal to the results found by MOCLE.

In general, this minor loss in MOCLE, MOCK, and Δ-MOCK by using con' is not so significant when compared to the general ARI gain in the datasets as presented in Table 2 (**Significant Wins** row). Furthermore, it promotes a significant increase in the performance of the EMO-KC without any loss.

Furthermore, it is important to observe that the Δ-MOCK provides the highest ARI for datasets with multiple true partitions. That also is pointed in the Critical Difference Diagram, Fig. 4, which shows the performance comparison of the strategies according to the Friedman and Bergmann-Hommel Post Hoc hypothesis test, in which Δ-MOCK has the best rank with con'.

Additionally, we also performed experiments with EMO-KC using three objectives, (var', *con*, k) and (var', con', k), that produces equivalent ARI results to the pair (var', con'). Since there is not a significant difference between the overall performance of the EMO-KC using two or three objective functions, we do not display these last results. Nevertheless, it is important to note that, based on these results, in EMO-KC, the use of the con' provides evidence that the conflict around the number of clusters is improved, even though the general ARI gain is not so robust.

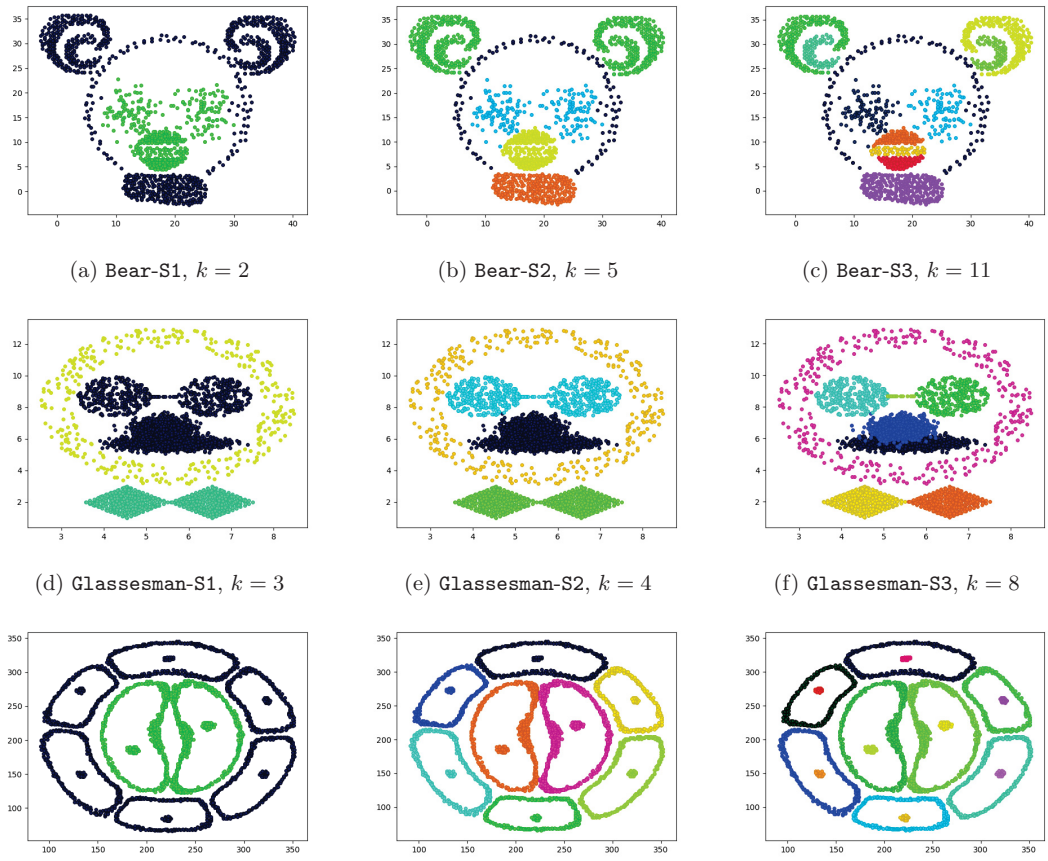
5.1 The impact of the con' in the optimization

As above-mentioned, the use of the con' promoted a general gain in the ARI; however, it also caused a loss in the ARI in some datasets. In this context, to analyze how the con' impact the optimization, we look over the Pareto front of the datasets `Monkey` and `Stomata`.

Fig. 5 presents the Pareto front of the datasets `Monkey` generated by MOCLE, in which the red points represent each true partition at different levels, as a reference for comparison. In Fig. 5b

[5] Available at http://lasid.sor.ufscar.br/clustersEvaluationBenchmark.

[6] Available at http://cs.uef.fi/sipu/datasets.



(a) `Bear-S1`, $k = 2$ (b) `Bear-S2`, $k = 5$ (c) `Bear-S3`, $k = 11$

(d) `Glassesman-S1`, $k = 3$ (e) `Glassesman-S2`, $k = 4$ (f) `Glassesman-S3`, $k = 8$

(g) `Stomata-S1`, $k = 2$ (h) `Stomata-S2`, $k = 8$ (i) `Stomata-S3`, $k = 16$

Fig. 3: New artificial datasets with nested data structures

we can observe an increase of the solutions in the region of the true partition of the `S1`, `S2`, and `S3` when compared with Fig. 5a. For the `S1`, we observed that the solution that ARI=0.5131 was dominated by other solutions when con' is applied, but it generates new solutions near the true partition, making it possible to apply other methods (local search) to improve the results. In particular, to detect the partition `S4` in `Monkey` requires further exploration of the region with smaller var in MOCLE. It is important to note that MOCLE has an inner property that reduces the number of solutions while producing new solutions generated by the ensemble-based crossover. It is the main reason for the small number of solutions in the Pareto front when compared to the other EMOCs.

Fig. 6 presents the Pareto front of this same dataset generated by MOCK. In the sub-figures, we also observe similar behavior to the MOCLE, in which using con' promoted the increase of solutions in the region of the high-level structures that are close to the true partition. However, for the `S4` in `Monkey`, the L parameter is still affecting the general performance of the MOCK, in which the use of the con' improved the convergence of the solutions.

For the dataset `Stomata`, in Δ-MOCK con' promote a better distinction of the solutions around the true partition of the `S2`, `S3`. Since the con for these structures is the same (con=9.8115), the con' generates more diversity of solutions in which the convergence is better than in con. In this context, this distinguishing of solutions improve of the ARI in the `S2`. For `S1` it promoted the increase of solutions in the region of the high-level structures that are close to the true partition, as illustrated in Fig. 7. The general loss in the ARI of the `S1` and `S3` using con' occurred because we have new solutions in the front with better convergence but still need some local exploitation to get a better ARI. It is important to note that in the Δ-MOCK, besides using the con', the initialization strategy also had an important role in its general results. We observe that initialization strategies that include KM, as in MOCK and MOCLE, could generate solutions that dominate other promising solutions with nested data structures. Besides that, the reduced encoding used in Δ-MOCK did not affect the clustering performance, in which Δ-MOCK is the more scalable approach with good ARI results.

On the other hand, in EMO-KC, the use of the random initialization and centroid-based representation had difficulties in detecting concentric clusters, such as the two spirals in the `Spiralsquare`, or clusters with close centroid and elongated data structures.

In summary, by using con' the optimization of the solutions was improved, which promoted more diversity of the solutions, including the regions of the high level of the nested data structure, and increased the convergence of the solution; however, some aspects of the EMOCs, such as initialization and representation, can impact the detection of the nested clusters.

6 Conclusion

In this study, we provide an analysis regarding the use of EMOCs for nested data structures. Furthermore, we deal with a problem in the definition of the connectivity index, in which several different partitions could present the same optimal value ($con = 0$) depending on the considered neighborhood size (L). In this scenario, the decision would be essentially taken based on the other

Table 2: The ARI of the best partition found by SL and AL, and the average ARI of the best partition found by MOCLE, MOCK, Δ-MOCK and EMO-KC. Average of 30 executions for the EMOCs

			SL	**AL**	**MOCLE**		**MOCK**		**Δ-MOCK**		**EMO-KC**		
#	**Datasets**	**S**	—	—	dev, con	dev, con'	dev, con	dev, con'	var, con	var, con'	var, k	var', con	var', con'
	20d-60c	-	0.0007	0.2601	**0.8989**	**0.8989**	**0.7819**	0.7810	**0.9003**	0.8998	0.5350	**0.5554**	0.5525
G1	Aggregation	-	0.8089	**1.0000**	**1.0000**	**1.0000**	**0.9935**	0.9908	**0.9671**	0.9656	0.7898	0.9346	**0.9535**
	D31	-	0.2124	0.9307	**0.9523**	**0.9523**	0.9030	**0.9044**	**0.7456**	0.7291	0.8136	0.8274	**0.8313**
	Iris	-	0.5681	0.7592	**0.8284**	**0.8284**	0.7707	**0.7891**	0.7709	**0.7869**	0.7370	**0.7543**	0.7437
G2	Libras	-	0.0224	0.3346	**0.3346**	**0.3346**	0.3942	**0.3973**	0.3865	**0.3886**	0.2762	**0.2949**	0.2933
	UKC1	-	**1.0000**	0.9415	**1.0000**	**1.0000**	0.9985	**1.0000**	0.9962	**0.9995**	0.9574	**0.9577**	0.9498
	ds2c2sc13	S1	**1.0000**	**1.0000**	0.6840	**1.0000**	0.3810	**1.0000**	0.3520	**1.0000**	**1.0000**	0.6828	**1.0000**
		S2	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9520	**1.0000**	0.8775	0.8777	**0.8841**
G3		S3	0.8724	0.6648	**1.0000**	0.9690	**0.8710**	0.8380	**0.8720**	**0.8720**	0.5889	0.6613	**0.6622**
	Spiralsquare	S1	**1.0000**	**1.0000**	0.8888	**1.0000**	0.5711	**1.0000**	0.5711	**1.0000**	0.9663	**1.0000**	**1.0000**
		S2	0.9283	0.5410	**0.9971**	**0.9971**	**0.9980**	0.9973	0.9986	**0.9987**	0.4742	0.5340	**0.5341**
	Monkey	S1	0.5122	0.4479	**0.5131**	0.4566	0.3377	**0.5653**	0.4544	**0.8654**	0.3737	**0.6124**	0.6076
		S2	0.8551	0.2279	0.8267	**0.8551**	0.7776	**0.8351**	0.7776	**0.9292**	0.2881	0.4468	**0.4629**
		S3	**0.8341**	0.5305	**0.8341**	**0.8341**	**0.7610**	**0.7640**	**0.7960**	**0.7960**	0.4272	**0.5197**	0.5037
		S4	0.8708	0.6713	**0.8707**	**0.8707**	**0.8628**	0.8404	**0.8737**	0.8719	0.6078	**0.6997**	0.6622
	Bear	S1	0.0042	0.1266	**0.2858**	**0.2858**	**0.4252**	0.4181	**0.4142**	0.4135	0.2179	**0.2565**	0.2431
		S2	0.0675	0.7194	**0.7194**	**0.7194**	0.7138	**0.7195**	**0.7220**	0.7219	0.5883	0.6752	**0.7048**
G4		S3	0.3895	0.6842	**0.6842**	**0.6842**	0.8061	**0.8097**	**0.7798**	0.7793	0.6349	0.6760	**0.6781**
	Glassesman	S1	**0.8775**	0.8269	**0.8775**	**0.8775**	**0.7920**	0.7857	0.7889	**0.7890**	0.7909	0.8077	**0.8097**
		S2	0.5944	0.5048	**0.9691**	**0.9691**	0.9271	**0.9291**	**0.9549**	0.9529	0.9034	**0.9273**	0.9228
		S3	0.2403	0.5048	**0.8428**	**0.8428**	**0.8505**	0.8467	0.8791	**0.8798**	0.7954	0.8155	**0.8214**
	Stomata	S1	0.0214	0.0382	**0.0382**	**0.0382**	**0.5986**	0.5946	**0.8635**	0.8311	0.0005	**0.0124**	0.0108
		S2	0.7233	0.2966	**0.7233**	**0.7233**	0.6987	**0.7025**	0.7970	**0.8368**	0.3269	**0.3708**	0.3574
		S3	0.7783	0.2620	**0.9190**	**0.9190**	**0.7356**	0.6805	**0.8952**	0.8573	0.2992	0.3368	**0.3455**
	Glass	S1	0.0536	0.0536	**0.6468**	**0.6468**	**0.5418**	**0.5424**	0.5620	**0.5663**	0.6099	**0.6086**	0.6077
		S2	0.1057	0.4918	**0.5043**	**0.5043**	0.4338	**0.4359**	**0.4605**	0.4552	0.4608	0.4951	**0.4999**
		S3	0.0403	0.2488	**0.2980**	**0.2980**	**0.2060**	**0.2030**	**0.2050**	0.2020	0.2205	0.2295	**0.2307**
G5	Golub	S1	-0.0026	-0.0139	**0.4193**	**0.4193**	0.7884	**0.8054**	0.5410	**0.5469**	0.4630	0.7203	**0.7406**
		S2	-0.0108	0.6473	**0.6473**	**0.6473**	**0.8816**	0.8714	0.5569	**0.5676**	0.5615	0.7055	**0.7126**
	Leukemia	S1	-0.0037	0.3346	**0.3295**	**0.3295**	0.3049	**0.4133**	0.3040	**0.4097**	0.2352	0.2945	**0.3004**
		S2	0.0224	0.3346	**0.7589**	**0.7589**	**0.7767**	**0.7767**	0.7706	**0.7708**	0.5922	**0.7201**	0.7180
Significant Wins			—	—	2	**3**	3	**6**	2	**8**	—	0	**4**

objective function in evolutionary multi-objective optimization. To tackle this problem, we presented a modified version of the connectivity index called con'. The results obtained with con', in terms of ARI and the ability to find nested cluster structures, are promising. In particular, there is a significant increase of the ARI in artificial datasets that present well-separate nested structures.

Besides the meaningful advantages in the scalability described by [8], Δ-MOCK demonstrated to be the best option of the studied EMOCs for nested clustering by using the con' as an objective function. In this context, we observe that the initialization strategy also contributes to the Δ-MOCK results, where other initialization strategies, like KM, could generate partitions that dominate other ones with nested structures.

Furthermore, we demonstrate how this modification impacts the optimization process by presenting the plot of the Pareto Front of the EMOCs, evidence that the con' improves the generation of a more diverse and convergent set of solutions.

Our results also showed that there are still some open problems regarding the L parameter still impacting the optimization, in which the true partition is dominated, deserving more studies. For future work, we consider that an analysis of different values of L can provide the extent of the results that depend on L.

We also introduce three new datasets (`Bear`, `Glassesman`, `Stomata`) that present a great challenge for the studied EMOCs, that could be explored in future works.

References

1. Bertrand, P., Diatta, J.: Multilevel clustering models and interval convexities. Discrete Applied Mathematics **222**, 54–66 (2017)
2. Coello, C.A.C., Lamont, G.B., Veldhuizen, D.A.V.: Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation). Springer-Verlag, Berlin, Heidelberg (2006)
3. Corne, D.W., Jerram, N.R., Knowles, J.D., Oates, M.J.: PESA-II: Region-based selection in evolutionary multiobjective optimization. In: Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation. pp. 283–290. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2001)
4. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE transactions on evolutionary computation **6**(2), 182–197 (2002)
5. Ertoz, L., Steinbach, M., Kumar, V.: A new shared nearest neighbor clustering algorithm and its ap-

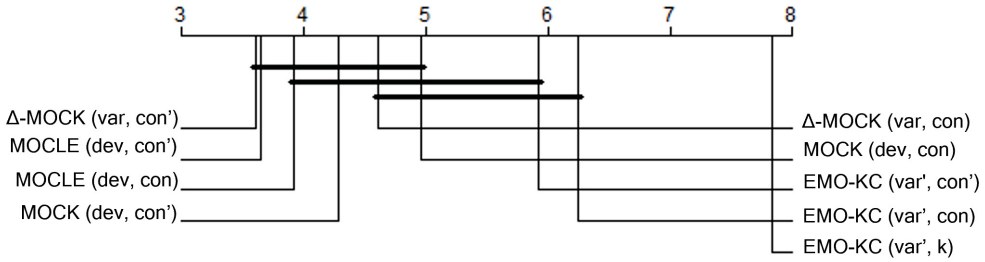
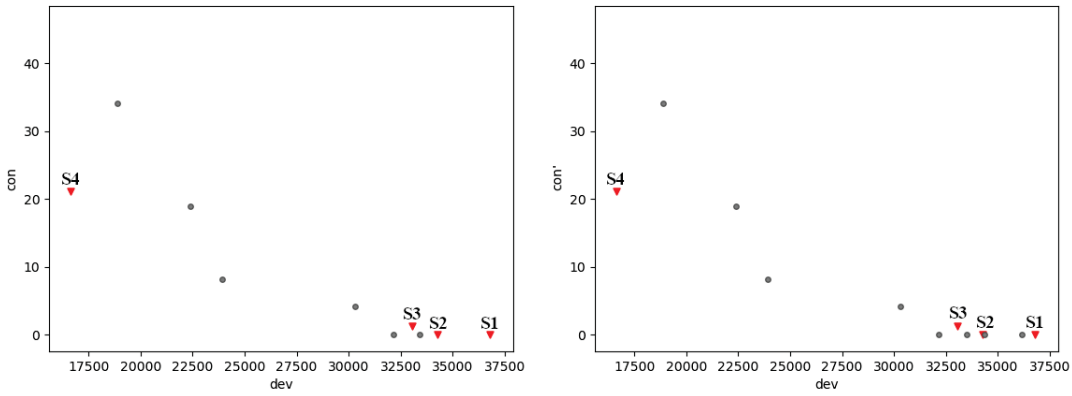


Fig. 4: Critical Difference Diagram of the EMOCs considering the different pairs of objective functions. The bold horizontal lines link the strategies that had statistically equivalent performance among them at a confidence level of 95%, and the lower the rank the better performance of an approach.



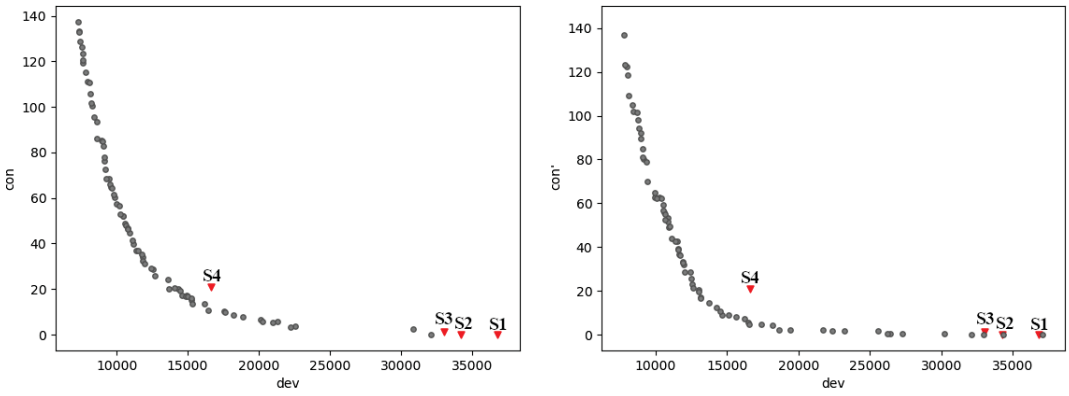
(a) PF of the objectives (*con*, *dev*)

(b) PF of the objectives (*con'*, *dev*)

Fig. 5: Front of the final population of the dataset `Monkey` generated by MOCLE

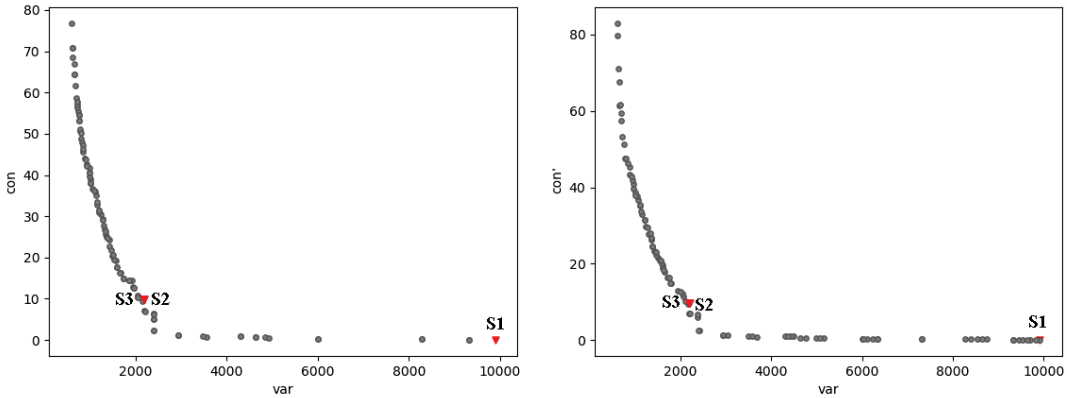
plications. In: Workshop on clustering high dimensional data and its applications at 2nd SIAM international conference on data mining. pp. 105–115 (2002)

6. Faceli, K., de Leon Ferreira de Carvalho, A.C.P., de Souto, M.C.P.: Multi-objective clustering ensemble. International Journal of Hybrid Intelligent Systems **4**(3), 145–156 (2007), http://content.iospress.com/articles/international-journal-of-hybrid-intelligent-systems/his00047
7. Fern, X.Z., Brodley, C.E.: Solving cluster ensemble problems by bipartite graph partitioning. In: Proceedings of the twenty-first international conference on Machine learning. p. 36. ACM (2004)
8. Garza-Fabre, M., Handl, J., Knowles, J.: An improved and more scalable evolutionary approach to multiobjective clustering. IEEE Transactions on Evolutionary Computation **22**(4), 515–535 (2017)
9. Handl, J., Knowles, J.: An evolutionary approach to multiobjective clustering. IEEE Transactions on Evolutionary Computation **11**(1), 56–76 (2007)
10. Hubert, L., Arabie, P.: Comparing partitions. Journal of classification **2**(1), 193–218 (1985)
11. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: A review. ACM Comput. Surv. **31**(3), 264–323 (Sep 1999). https://doi.org/10.1145/331499.331504, http://doi.acm.org/10.1145/331499.331504
12. MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. vol. 1, pp. 281–297. Oakland, CA, USA (1967)
13. Mukhopadhyay, A., Maulik, U., Bandyopadhyay, S.: A survey of multiobjective evolutionary clustering. ACM Computing Surveys (CSUR) **47**(4), 61 (2015)
14. Muller, E., Gunnemann, S., Farber, I., Seidl, T.: Discovering multiple clustering solutions: Grouping objects in different views of the data. In: 2012 IEEE 28th International Conference on Data Engineering. pp. 1207–1210. IEEE (2012)
15. Pohlert, T.: PMCMR: Calculate Pairwise Multiple Comparisons of Mean Rank Sums (may 2018), https://CRAN.R-project.org/package=PMCMR
16. Wang, R., Lai, S., Wu, G., Xing, L., Wang, L., Ishibuchi, H.: Multi-clustering via evolutionary multi-objective optimization. Information Sciences **450**, 128–140 (2018). https://doi.org/https://doi.org/10.1016/j.ins.2018.03.047
17. Xu, R., Wunsch, D.: Survey of clustering algorithms. IEEE Transactions on Neural Networks



(a) PF of the objectives (*con*, *dev*) (b) PF of the objectives (*con*′, *dev*)

Fig. 6: Front of the final population of the dataset `Monkey` generated by MOCK



(a) PF of the objectives (*con*, *var*) (b) PF of the objectives (*con*′, *var*)

Fig. 7: Front of the final population of the `Stomata` generated by Δ-MOCK

16(3), 645–678 (May 2005). https://doi.org/10.1109/TNN.2005.845141