European Journal of Operational Research 314 (2024) 1065–1077

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/eor

Decision support

Cluster ensemble selection and consensus clustering: A multi-objective optimization approach

Dilay Aktaş [a], Banu Lokman [b,*], Tülin İnkaya [c], Gilles Dejaegere [d]

[a] Centre for Industrial Management, KU Leuven, Celestijnenlaan 300, 3001 Leuven, Belgium
[b] School of Organisations, Systems and People, Centre for Operational Research and Logistics, University of Portsmouth, PO1 3DE, Portsmouth, United Kingdom
[c] Department of Industrial Engineering, Bursa Uludağ University, 16240 Nilüfer, Bursa, Turkey
[d] Service de Mathématiques de la Gestion, Université libre de Bruxelles, Boulevard du Triomphe CP 210-01, 1050 Brussels, Belgium

ARTICLE INFO

Keywords:
Multiple objective programming
Cluster ensembles
Ensemble selection
Consensus clustering

ABSTRACT

Cluster ensembles have emerged as a powerful tool to obtain clusters of data points by combining a library of clustering solutions into a consensus solution. In this paper, we address the cluster ensemble selection problem and design a multi-objective optimization-based solution framework to produce consensus solutions. Given a library of clustering solutions, we first design a preprocessing procedure that measures the agreement of each clustering solution with the other solutions and eliminates the ones that may mislead the process. We then develop a multi-objective optimization algorithm that selects representative clustering solutions from the preprocessed library with respect to size, coverage, and diversity criteria and combines them into a single consensus solution, for which the true number of clusters is assumed to be unknown. We conduct experiments on different benchmark data sets. The results show that our approach yields more accurate consensus solutions compared to full-ensemble and the existing approaches for most data sets. We also present an application on the customer segmentation problem, where our approach is used to segment customers and to find a consensus solution for each segment, simultaneously.

1. Introduction

Clustering is an unsupervised learning method aiming to reveal the true nature of data in the absence of any external knowledge of labels (Jain, Murty, & Flynn, 1999). The objective is to discover the patterns and structures in high-dimensional data and divide them into groups of similar data points (Berkhin, 2006). It has been widely used in a variety of application domains including healthcare, bioinformatics, marketing, and image segmentation (Reutterer & Dan, 2020; Saxena et al., 2017; Tomar & Agarwal, 2013). However, there is no single proven algorithm or technique that performs well for all data sets according to the well-known *no-free-lunch* theorem of Wolpert, Macready, et al. (1995). The performance of the clustering algorithms is highly dependant upon the characteristics of the data set as well as the parameter settings at the outset (Kuncheva & Hadjitodorov, 2004; Wolpert et al., 1995).

Considering that different clustering algorithms or even the same algorithm with different parameters can produce very distinct partitions of the data set, Fred and Jain (2005) introduce the concept of evidence accumulation for combining different clustering solutions into a single consolidated clustering. This process is known as *cluster ensembles* and shown to provide more robust results across a very wide range of data sets (Boongoen & Iam-On, 2018). Given a data set, the process starts with generating an initial library of clustering solutions by applying different algorithms or using different parameters. Then, a clustering combination strategy is used to produce a consensus solution. While combining all clustering solutions in the library, *the full-ensemble*, is typically employed in the earlier studies of cluster ensembles, Fern and Lin (2008) introduce cluster ensemble selection problem and show that using the full-ensemble does not only increase the computational effort but also masks the true nature of data and misleads the consensus function. The aim of the cluster ensemble selection is to find a smaller subset of solutions such that the resulting partition performs as well or better than the solution obtained by the full-ensemble. Fern and Brodley (2003) point out that the ensemble should be of good quality and diversity to obtain such consensus solutions. Although there are variations, quality mainly stands for the capability of the ensemble to reflect the clustering trend in the library whereas diversity stands for the ability to attain diverse consensus solutions from the ensemble.

* Corresponding author.
E-mail addresses: dilay.aktas@kuleuven.be (D. Aktaş), banu.lokman@port.ac.uk (B. Lokman), tinkaya@uludag.edu.tr (T. İnkaya), gilles.dejaegere@ulb.be (G. Dejaegere).

https://doi.org/10.1016/j.ejor.2023.10.029
Received 21 February 2022; Accepted 25 October 2023
Available online 29 October 2023
0377-2217/

In this paper, we develop an innovative approach to cluster ensemble selection problem and design a multi-objective optimization-based solution framework to find a consensus clustering given a library of various partitions of the data set. The aim is to select a smaller cluster ensemble (*size*) from a library of clustering solutions that represents the full-ensemble well (*quality*) without any redundancies (*diversity*) and performs as well as or better than the full ensemble. In contrast to existing literature, we utilize exact multi-objective and compromise-programming approaches to deal with these multiple criteria simultaneously. We first design a novel algorithm, Representative Clusterings Algorithm (RCA), that finds a subset of solutions representing the library well based on these three criteria. To evaluate the quality of the representative set, we utilize a new metric, coverage gap, that measures how well each solution in the full-ensemble is represented by the subset. Since RCA aims to minimize the size of the representative set while maximizing the quality and diversity, there does not exist a unique solution. Therefore, RCA is designed to find *efficient ensembles* (EEs) for which it is not possible to improve one criterion without sacrificing from another one. Each EE of clustering solutions is then used to generate an *efficient consensus solution* (ECS). Since there may exist some clustering solutions in the initial library which may mislead the representative selection process, we also develop a preprocessing algorithm (PPA) that detects and removes such outlier solutions from the full-ensemble before applying RCA.

This paper contributes to the relevant literature by not only designing a novel approach that embeds multi-objective optimization and decision making tools into the ensemble selection process but also addressing the case where the number of clusters is unknown. Without using any external knowledge on the true number of clusters, the cluster ensemble method generates *efficient consensus clustering solutions* with respect to quality, diversity and size criteria by characterizing the pattern and structure of the data. This is particularly important in customer or patient segmentation and design of recommendation systems and personalized treatment plans where having different customer or patient segments can serve different purposes.

The organization of the paper is as follows. We give background information in Section 2 and develop our algorithms in Section 3. We present computational results of experiments on the benchmark data sets in Section 4 and apply our approach to the customer segmentation problem in Section 5. We then conclude in Section 6.

2. Background and related work

Cluster analysis is the task of grouping a set of data points. By applying different clustering methods or with different parameter settings, distinct clustering solutions for the same data set can be obtained. *Cluster ensembles* have emerged as a powerful tool to overcome the difficulty of choosing one of these solutions. Instead, cluster ensembles combine a set of clustering solutions into a single consensus solution.

2.1. The cluster ensemble problem

For a set of N data points $X = \{x_1, x_2, \dots, x_N\}$, let $\Pi = \{\pi_1, \dots, \pi_L\}$ be a cluster ensemble with L clustering solutions where each ensemble member, $\pi_i = \{C_1^{\pi_i}, \dots, C_{k_i}^{\pi_i}\}$, represents a partition of the data set into k_i clusters. The aim is to combine these L ensemble members into a new clustering solution, π^*, that does not only summarize the information from the cluster ensembles but also outperforms its members in terms of the correspondence with the true labels. Boongoen and Iam-On (2018) and more recently Golalipour, Akbari, Hamidi, Lee, and Enayatifar (2021) present an overview of cluster ensemble generation and selection strategies with a discussion on theoretical details of state-of-the-art methods and extensions.

2.2. Ensemble generation

In the cluster ensemble problem, several strategies have been proposed to generate the cluster ensemble, the initial library of clustering solutions. While a homogeneous ensemble is obtained by using a single clustering algorithm with different parameter settings (Fred & Jain, 2005; Gionis, Mannila, & Tsaparas, 2007), multiple algorithms are run to generate a heterogeneous ensemble (Ayad & Kamel, 2003; Azimi & Fern, 2009; Hadjitodorov, Kuncheva, & Todorova, 2006; Law, Topchy, & Jain, 2004). Since the output of clustering algorithms is also dependant on the choice of the number of clusters and features, there are also studies that generate the cluster ensemble by changing the number of clusters (Fern & Lin, 2008; Fred & Jain, 2005; Kuncheva & Vetrov, 2006) and/or by choosing different subsets of features (Fern & Lin, 2008; Strehl & Ghosh, 2002; Yu, Wong, & Wang, 2007). It has been shown that the quality of the ensemble members as well as the diversity among them have a strong impact on the performance of the consensus solution (Fern & Brodley, 2003; Kuncheva & Vetrov, 2006).

2.3. Cluster ensemble selection

The cluster ensemble framework typically produces a large number of clustering solutions using different algorithms, different parameter settings or different representations of the data. Although it is common to use the *full-ensemble* in producing a final clustering solution, identifying a smaller subset can perform as effectively as or better than the full ensemble (Golalipour et al., 2021).

There are several studies on cluster ensemble selection in the literature (Alizadeh, Minaei-Bidgoli, & Parvin, 2014; Fern & Lin, 2008; Hadjitodorov et al., 2006; Naldi, Carvalho, & Campello, 2013; Wang & Liu, 2018; Yang, Li, Zhou, & Xiao, 2017). Although Hadjitodorov et al. (2006) earlier generated multiple cluster ensembles and selected the one with moderate diversity to use in producing a final consensus clustering, the cluster ensemble selection problem has been introduced by Fern and Lin (2008). They show that both quality and diversity are instrumental in producing a smaller cluster ensemble performing as well as or better than the full ensemble. The quality of an ensemble is defined as the total similarity of its members to the other clustering solutions in the full-ensemble while the diversity of an ensemble is the total pairwise-similarity of solutions in the ensemble. Although their strategies jointly consider both quality and diversity criteria, they do not generate all efficient ensembles with respect to the two criteria. Their Joint Criterion (JC) method reduces the problem into the single criterion by aggregating the two criteria while the Cluster and Select (CAS) method utilizes a lexicographic approach that considers both criteria in a prioritized manner. The hierarchical cluster ensemble selection (HCES) method of Akbari, Dahlan, Ibrahim, and Alizadeh (2015) can be considered as a generalized version of CAS as it follows a lexicographic approach to deal with the two criteria but also considers the size of ensemble.

While the methods developed by Hadjitodorov et al. (2006) and Fern and Lin (2008) are generic and applicable for all data sets, Azimi and Fern (2009) introduce an adaptive cluster ensemble selection method that takes the characteristics of initial library into account. They first classify the initial library of the solutions into two categories, *stable* or *non-stable*, based on the similarity between the full-ensemble members and their consensus solution, and apply a selection strategy for each category. If the initial library is stable, they select the full-ensemble. Otherwise, they identify the most dissimilar solutions to the consensus solution generated by full-ensemble and include them in the subset. Although this approach works well in terms of the accuracy of the resulting consensus solution, the ensemble selected by this approach does not represent the characteristics of the initial library well. Jia, Xiao, Liu, and Jiao (2011) generalize this approach and generate subsets using spectral clustering and bagging technique.

$$\underset{\Pi}{\text{Initial Library}} \xrightarrow{\text{PPA}} \underset{\rho \subseteq \Pi}{\text{Preprocessed Library}} \xrightarrow{\text{RCA}} \underset{\forall \tau \in R_{EE},\, \tau \subseteq \rho}{\text{Set of Efficient Representative Ensembles}} \xrightarrow{\text{CGM}} \underset{\pi^*_{\tau,k} \in \Pi^*}{\text{Efficient Consensus Solutions}}$$

Fig. 1. Multi-objective Optimization Framework to select ensembles and obtain consensus solutions.

Naldi et al. (2013) investigate the use of relative validation indexes in evaluating the quality in cluster ensemble selection strategies and compare the results against those of Fern and Lin (2008). They also investigate the impact of diversity and show that combining different relative indexes in cluster ensemble selection produces a final consensus solution that tends to be more robust and accurate than solutions obtained by using individual indexes. Abbasi, Nejatian, Parvin, Rezaie, and Bagherifard (2019) later propose a new validity measure for each cluster in a solution where in case a threshold for this measure is satisfied, the cluster can be included in the final consensus solution.

Different from prior approaches, Yang et al. (2017) and Wang and Liu (2018) take the characteristics of the original data and partitioning of objects into account in ensemble selection. Yang et al. (2017) propose a greedy approach that searches for the ensemble that maximizes a weighted sum score based on quality, diversity, and consistency. They define *must-link* and *cannot-link* constraints indicating that a pair of objects should be in the same cluster, or they should be in different clusters. They measure the consistency of a clustering solution by the fraction of constraints satisfied while the consistency of an ensemble corresponds to the average consistency of the ensemble members. Similarly, Wang and Liu (2018) use multi-modal metrics to evaluate the quality and diversity of each individual clustering solution and then use a weighted-sum method to select the solutions that will participate in the clustering ensemble process.

Given a cluster ensemble, Li, Qian, Wang, Dang, and Jing (2019) introduce a stability metric to differentiate the objects that remain consistently in one cluster from those that frequently move from one cluster to another. They then propose a cluster ensemble method that uses these stability metrics in producing a consensus solution. These stability metrics could also be utilized in cluster ensemble selection.

These studies propose techniques and methods to select a smaller better-performing subset from the full-ensemble based on diversity and quality. As diversity, quality and size compete against each other, the ensemble selection problem is a multi-objective optimization problem by its nature for which there is no unique solution that simultaneously optimizes each objective. There does not exist any algorithm in the literature which is capable of generating efficient subsets for a multi-objective ensemble selection problem.

2.4. Consensus functions

After the cluster ensemble is generated and a subset is selected, the next task is to combine these cluster ensemble members by means of a *consensus function*, which is a mapping from the set of clustering solutions to a single final solution. Boongoen and Iam-On (2018) classify the methods used in combining ensembles into four categories as direct, feature-based, pairwise-similarity based, and graph-based approaches. Direct approaches develop a voting model for the consensus label of each point (Ayad & Kamel, 2010; Topchy, Law, Jain, & Fred, 2004) while feature-based methods cluster data points based on the information obtained from an ensemble, without searching for correspondence amongst labels (Nguyen & Caruana, 2007; Yu et al., 2014). Another approach to generate a consensus solution is to create a pairwise similarity/dissimilarity matrix of data points based on the ensemble members and apply a clustering algorithm to obtain consensus labels (Fred & Jain, 2005; Yan, Chen, & Jordan, 2013). Lastly, graph-based approaches construct a network of data points representing the cluster ensemble and solve a graph partitioning problem to find a consensus solution (Fern & Brodley, 2004; Strehl & Ghosh, 2002; Xiao, Yang, Wang, Li, & Xing, 2016).

Finding the appropriate size for a cluster ensemble is a challenging problem and it is still uncertain, according to the recent and comprehensive review of Golalipour et al. (2021) on cluster ensembles and ensemble selection. Moreover, consensus functions are typically sensitive to size, quality, and diversity of the ensemble, while some are more sensitive to one criterion than the others (Golalipour et al., 2021). Therefore, integrating multi-objective optimization methods into the cluster ensemble selection process is useful in generating better-performing consensus solutions and analyzing the trade-offs in these multiple criteria.

3. Development of the algorithms

Given a library of clustering solutions, $\Pi = \{\pi_1, \dots, \pi_L\}$, we design a methodology that combines the strengths of these solutions and produces a single consensus solution, π^*. We first develop a preprocessing algorithm (PPA) that filters out the "weak" clustering solutions from the initial library and obtain a subset of the full-ensemble, $\rho \subseteq \Pi$. We then design a Representative Clusterings Algorithm (RCA), to find a subset of clustering solutions, τ, that represents the surviving solutions in the preprocessed library. Since RCA aims to minimize the size of the subset while maximizing the diversity and quality, the problem is multi-objective by its nature and there does not exist a unique optimal representative subset. Therefore, RCA generates the set of efficient representatives, R_{EE}, each of which is a subset of the clustering solutions in the preprocessed library, $\tau \subseteq \rho$, for which it is not possible to improve one criterion without a deterioration in another one. Although RCA can be used to generate a single representative consensus solution by setting the size of the subset to one, a set of efficient consensus solutions, Π^*, could also be obtained applying Consensus Generation Method (CGM) to the efficient ensembles. In CGM, a consensus function, $\Phi(\tau, k)$, is applied to map the solutions in an efficient subset τ into a final clustering solution $\pi^*_{\tau,k}$ with the desired number of clusters, k. Fig. 1 presents the solution framework of our multi-objective approach.

3.1. Preprocessing algorithm (PPA)

Given a library of clustering solutions, PPA is developed to identify "outliers" that are dissimilar to the remainder solutions. In order to detect these clustering solutions that might mislead the ensemble selection process, PPA uses an external cluster validation metric, Normalized Mutual Information (NMI), that measures the mutual information shared by two clustering solutions as given in Definition 1 (Strehl & Ghosh, 2002).

Definition 1. Let $\Pi = \{\pi_1, \dots, \pi_L\}$ be a set of clustering solutions for a set of N data points, $X = \{x_1, x_2, \dots, x_N\}$. A clustering solution, $\pi_i = \{C_1^{\pi_i}, \dots, C_{k_i}^{\pi_i}\} \in \Pi$, partitions the data set into k_i clusters, such that $X = \bigcup_{u=1}^{k_i} C_u^{\pi_i}$ where $C_u^{\pi_i} \cap C_v^{\pi_i} = \emptyset, \forall u \neq v$. Then, the normalized mutual information between two clusterings $\pi_i, \pi_j \in \Pi$ is defined as:

$$NMI(\pi_i, \pi_j) = \frac{\sum_{u=1}^{k_i} \sum_{v=1}^{k_j} n_{uv} \log\left(\frac{N.n_{uv}}{n_u^{(i)}.n_v^{(j)}}\right)}{\sqrt{\left(\sum_{u=1}^{k_i} n_u^{(i)} \log \frac{n_u^{(i)}}{N}\right)\left(\sum_{v=1}^{k_j} n_v^{(j)} \log \frac{n_v^{(j)}}{N}\right)}} \tag{1}$$

where $n_u^{(i)}$ and $n_v^{(j)}$ represent the number of data points in cluster $C_u^{\pi_i}$ and cluster $C_v^{\pi_j}$, respectively. Similarly, n_{uv} denotes the number of data points that are in cluster u according to π_i as well as in cluster v according to π_j.

Using Definition 1, we measure the similarity between two clustering solutions, π_i and π_j, by the normalized mutual information shared. Note that $NMI(\pi_i, \pi_j)$ takes the value of 1 when π_i and π_j are identical solutions.

In order to measure how a clustering solution is similar to other clusterings in the library, we next define *a measure of agreement* that is calculated for each solution based on its pairwise similarity values with the rest of the solutions.

Definition 2. Let $\Pi = \{\pi_1, \dots, \pi_L\}$ be a set of clustering solutions. The *agreement* of $\pi_i \in \Pi$ with the rest of solutions in Π is denoted by κ_i and quantified based on pairwise similarity values:

$$\kappa_i = \frac{\sum_{\pi_{j\neq i}\in\Pi} NMI(\pi_i, \pi_j)}{L-1} \tag{2}$$

Since PPA aims to identify outlier solutions in the initial library based on agreement values, we use standardized agreement values to make them comparable for different data sets:

$$Z_{\kappa_i} = \frac{\kappa_i - \mu_\kappa}{\sigma_\kappa} \tag{3}$$

where μ_κ and σ_κ are mean and standard deviation values of κ_i, $i = 1, \dots, L$.

As a rule of thumb, data points that lie beyond ±3 standard deviations from the mean signals anomaly in data collection if data is either normally distributed or the Central Limit Theorem (CLT) is applicable (Hines, Montgomery, & Borror, 2008). Exploiting the rule of thumb, we define clustering solutions that are having agreement values less than three standard deviations of the mean as outlier solutions. However, the solutions that are in agreement with the library more than three standard deviations of the mean are not considered as outliers as they represent the library well. In each iteration, PPA checks whether there exists any outlier solution. If the number of solutions in the library is greater than 30, for CLT to be applicable, the solution having the least agreement with the rest of the library is eliminated. PPA then recalculates the agreement values and repeats the process until no other solution is considered as outlier. Algorithm 1 presents the steps of PPA.

Algorithm 1: Preprocessing Algorithm (PPA)

Input: Π
Output: ρ

Initialize $\rho = \Pi$
foreach *solution* $\pi_i \in \rho$ **do**
  Calculate Z_{κ_i}
end
Find $\pi_{i^*} = \underset{\pi_i\in\rho}{\text{argmin}}\, Z_{\kappa_i}$
while $Z_{\kappa_{i^*}} \leq -3$ *and* $|\rho| > 30$ **do**
  Update $\rho \leftarrow \rho \setminus \{\pi_{i^*}\}$
  foreach *solution* $\pi_i \in \rho$ **do**
    Calculate Z_{κ_i}
  end
  Find $\pi_{i^*} = \underset{\pi_i\in\rho}{\text{argmin}}\, Z_{\kappa_i}$
end
return ρ

3.2. Ensemble selection method: Representative clusterings algorithm (RCA)

Aiming to select good cluster ensembles that represent the preprocessed library well, we adapt the measures proposed by Sayın (2000) and develop a multi-objective optimization algorithm, RCA. Given a set of clustering solutions, ρ, we next discuss how RCA measures the quality, diversity, and size of a representative ensemble, $\tau \subseteq \rho$.

Definition 3. Let ρ be a set of clustering solutions and $\tau \subseteq \rho$ denote a representative ensemble. Then, $\pi_{j*} \in \tau$ is said to be representative of $\pi_i \in \rho$ if $\pi_{j*} = \text{argmax}_{\pi_j\in\tau} NMI(\pi_i, \pi_j)$ and the representation error of π_i is measured by $\beta_{\pi_i} = 1 - NMI(\pi_i, \pi_{j*})$.

Definition 4. Let ρ be a set of clustering solutions and $\tau \subseteq \rho$ denote a representative ensemble. Then, the *coverage gap* value of τ is denoted by α_τ, determined by the representation error of the worst represented clustering solution in ρ:

$$\alpha_\tau = \max_{\pi_i\in\rho} \beta_{\pi_i} \tag{4}$$

Since we use normalized mutual information values to measure the quality of a representative ensemble, its coverage gap, α_τ, takes a value between 0 and 1. The best quality level, $\alpha_\tau = 0$, is achieved when all clustering solutions are involved in the representative ensemble, $\tau = \rho$. We next define a metric to measure how diverse the clustering solutions are in the representative ensemble.

Definition 5. Let ρ be a set of clustering solutions and $\tau \subseteq \rho$ denote a representative ensemble. The *diversity* of τ is denoted by γ_τ and determined by the quantity:

$$\gamma_\tau = \max_{\pi_i,\pi_j\in\tau} NMI(\pi_i, \pi_j) \tag{5}$$

The best diversity value is obtained when the ensemble is composed of the pair of most dissimilar solutions. Hence, the smaller γ_τ value is, the more diverse the representative ensemble is.

In order to produce a diverse but high quality set of efficient ensembles that does not include any redundancies, RCA solves a three-objective optimization problem:

$$\begin{aligned}
&[\boldsymbol{P}]\\
&\text{“}Min\text{”} z(\tau) = (z_1(\tau), z_2(\tau), z_3(\tau))\\
&\text{subject to}\\
&(z_1(\tau), z_2(\tau), z_3(\tau)) = (\alpha_\tau, \gamma_\tau, |\tau|)\\
&\tau \subseteq \rho
\end{aligned}$$

where $\tau \subseteq \rho$ denotes a representative ensemble and $z(\tau)$ is the corresponding objective vector. $|\tau| \geq 2$ is assumed since the diversity metric is not defined for $|\tau| = 1$. As the three objectives of $[\boldsymbol{P}]$ are known to have different ranges, coverage gap, diversity, as well as size values are normalized for each representative such that the best values take the value of 0 and the worst values take the value of 1.

Definition 6. For any τ^1, $\tau^2 \subseteq \rho$, if $z_p(\tau^1) \leq z_p(\tau^2)$ holds for all objectives $p = 1, 2, 3$ and $z_p(\tau^1) < z_p(\tau^2)$ for at least one objective, then τ^1 is said to *dominate* τ^2. If there does not exists any solution that dominates τ^1, then τ^1 is called an *efficient ensemble* and $z(\tau^1)$ is said to be a *nondominated point*.

Due to the multi-objective nature of $[\boldsymbol{P}]$, there does not exist a unique representative ensemble that optimizes the three criteria simultaneously. In order to generate efficient ensembles (EEs), RCA fixes the size of the representative ensemble at S and reduces $[\boldsymbol{P}]$ to a bi-objective optimization problem, $[\boldsymbol{P}(S)]$, for which all nondominated points, $z(\tau) \in Z(S)$, are iteratively generated using the modified epsilon constraint method of Steuer (1986).

$$\begin{aligned}
&[\boldsymbol{P}(S)]\\
&\text{“}Min\text{”} z(\tau) = (z_1(\tau), z_2(\tau))\\
&\text{subject to}\\
&(z_1(\tau), z_2(\tau), z_3(\tau)) = (\alpha_\tau, \gamma_\tau, |\tau|)\\
&|\tau| = S\\
&\tau \subseteq \rho
\end{aligned}$$

For any given $S \geq 2$ value, RCA solves $[\boldsymbol{P^{RCA}}(S, \gamma_{ub})]$ to generate a representative ensemble, which minimizes the coverage gap of the representative subset, α, while keeping the diversity at a certain level, i.e. $\gamma \leq \gamma_{ub}$. It then systematically changes the upper bound on the diversity measure, γ_{ub}, and generates new representative ensembles. After all nondominated points of $[\boldsymbol{P}(S)]$ are found, RCA changes the size of the representative subset, S, and repeats the same process. The representative ensembles generated throughout the process are kept in a set, R. These ensembles are guaranteed to be nondominated in terms of quality and diversity but not in terms of all the three criteria. Therefore, RCA applies a nondominance check on R at the end of the algorithm and efficient ensembles are involved in R_{EE}. We next present $[\boldsymbol{P^{RCA}}(S, \gamma_{ub})]$ and summarize the steps of RCA in Algorithm 2.

$[\boldsymbol{P^{RCA}}(S, \gamma_{ub})]$

$$\text{Minimize} \quad \alpha + \epsilon_\gamma \gamma \tag{6}$$

subject to;

$$\sum_{\pi^j \in \rho} y_{ij} = 1 \qquad \forall \pi^i \in \rho \tag{7}$$

$$y_{ij} \leq y_{jj} \qquad \forall \pi^i, \pi^j \in \rho \tag{8}$$

$$\sum_{\pi^j \in \rho} y_{jj} = S \tag{9}$$

$$\alpha \geq y_{ij}\,(1 - NMI(\pi_i, \pi_j)) \qquad \forall \pi^i, \pi^j \in \rho \tag{10}$$

$$r_{ij} \leq y_{ii} \qquad \forall \pi^i, \pi^j \in \rho,\ i \neq j \tag{11}$$

$$r_{ij} \leq y_{jj} \qquad \forall \pi^i, \pi^j \in \rho,\ i \neq j \tag{12}$$

$$r_{ij} \geq y_{ii} + y_{jj} - 1 \qquad \forall \pi^i, \pi^j \in \rho,\ i \neq j \tag{13}$$

$$\gamma \geq r_{ij}\, NMI(\pi_i, \pi_j) \qquad \forall \pi^i, \pi^j \in \rho,\ i \neq j \tag{14}$$

$$\gamma \leq \gamma_{ub} \tag{15}$$

$$y_{ij}, r_{ij} \in \{0, 1\} \qquad \forall \pi^i, \pi^j \in \rho \tag{16}$$

$$\alpha, \gamma \geq 0 \tag{17}$$

The objective function (6) of $[\boldsymbol{P^{RCA}}(S, \gamma_{ub})]$ consists of two components. While the coverage gap of the representative ensemble, α, is minimized in the first part, the second part, $\epsilon_\gamma \gamma$, breaks the ties in favor of more diverse representative ensembles using a sufficiently small positive constant $\epsilon_\gamma > 0$. Therefore, it guarantees an efficient ensemble in terms of quality and diversity measures for a given S value.

In $[\boldsymbol{P^{RCA}}(S, \gamma_{ub})]$, y_{ij} takes the value of 1 if clustering solution π^i is represented by clustering solution π^j and 0 otherwise. Constraints (7) ensure that each solution is represented by exactly one solution while y_{jj} is forced to take the value of 1 by constraints (8) if π^j is a representative solution. Constraint (9) sets the number of representative clustering solutions to S.

Since the coverage gap is measured by the maximum representation error, constraints (10) ensure that $\alpha \geq (1 - NMI(\pi_i, \pi_j))$ is satisfied if π^i is represented by π^j (i.e. $\alpha \geq \beta_{\pi_i}$). Otherwise, if $y_{ij} = 0$, the corresponding constraint becomes redundant. Constraints (11)–(14) ensure that the diversity is measured by the maximum similarity between representatives since r_{ij} is forced to take value of 1 only when π^i and π^j are both representative solutions. Constraint (15) sets an upper bound, γ_{ub}, on the diversity measure.

If $[\boldsymbol{P^{RCA}}(S, \gamma_{ub})]$ is feasible, the point corresponding to the optimal representative ensemble, $\tau_{S,\gamma_{ub}}$, is denoted as $z(\tau_{S,\gamma_{ub}}) = (\alpha_{S,\gamma_{ub}}, \gamma_{S,\gamma_{ub}}, S)$ and included in R. In order to search for more diverse representative ensembles at the same size, RCA updates the upper bound on the diversity measure and solve $[\boldsymbol{P^{RCA}}(S, \gamma_{ub})]$ using $\gamma_{ub} = \gamma_{S,\gamma_{ub}} - \epsilon$ where ϵ is a sufficiently small positive constant.

If there does not exist a representative ensemble with a size of S satisfying the upper bound on the diversity measure, $[\boldsymbol{P^{RCA}}(S, \gamma_{ub})]$ turns out to be infeasible. In this case, RCA stops changing γ_{ub} and sets the number of representatives in the ensemble to $S - 1$. For any $S \geq 2$, RCA restarts the same process with $\gamma_{ub} = 1$. The algorithm is terminated when $S = 1$ is achieved since the diversity metric is not measurable.

Algorithm 2: Representative Clusterings Algorithm (RCA)

```
Input: ρ, ε, ε_γ
Output: R_EE
Initialize ensemble size, S = |ρ|, the set of efficient ensembles
  R_EE = ∅ and the set of ensembles generated throughout the
  algorithm R = ∅.
while S ≥ 2 do
    Initialize γ_ub = 1
    Solve [P^RCA(S, γ_ub)]
    while a feasible solution exists do
        Denote the point corresponding to the optimal
          representative ensemble, τ_{S,γ_ub}, as
          z(τ_{S,γ_ub}) = (S, α_{S,γ_ub}, γ_{S,γ_ub}).
        Update R ← R ∪ {τ_{S,γ_ub}} and γ_ub ← γ_{S,γ_ub} − ε
        Solve [P^RCA(S, γ_ub)]
    end
    Update S ← S − 1
end
Initialize R_EE = R
foreach τ^e ∈ R do
    foreach τ^d ∈ R do
        if τ^d dominates τ^e then
            Update R_EE ← R_EE \ {τ^e}
        end
    end
end
return R_EE
```

Proposition 1. *Let $[\boldsymbol{P^{RCA}}(S, \gamma_{ub})]$ be feasible and $z(\tau_{S,\gamma_{ub}}) = (\alpha_{S,\gamma_{ub}}, \gamma_{S,\gamma_{ub}}, S)$ denote the point corresponding to the optimal representative ensemble, $\tau_{S,\gamma_{ub}}$. If $\epsilon_\gamma > 0$, then $z(\tau_{S,\gamma_{ub}})$ is nondominated with respect to problem $[\boldsymbol{P}(S)]$.*

Proof. To get a contradiction, suppose that $z(\tau_{S,\gamma_{ub}})$ is dominated with respect to problem $[\boldsymbol{P}(S)]$. Then, by Definition 6, there should exist an ensemble, $\tau' \subseteq \rho$, for which $z(\tau') = (\alpha', \gamma', S) \in Z(S)$ dominates $z(\tau_{S,\gamma_{ub}})$. Thus, one of the following conditions should be satisfied:

i. $\alpha' < \alpha_{S,\gamma_{ub}}$ and $\gamma' \leq \gamma_{S,\gamma_{ub}}$
ii. $\alpha' \leq \alpha_{S,\gamma_{ub}}$ and $\gamma' < \gamma_{S,\gamma_{ub}}$

Since $\epsilon_\gamma > 0$, any of these conditions implies $\alpha' + \epsilon_\gamma \gamma' < \alpha_{S,\gamma_{ub}} + \epsilon_\gamma \gamma_{S,\gamma_{ub}}$, which is a contradiction to $\tau_{S,\gamma_{ub}}$ being optimal to $[\boldsymbol{P^{RCA}}(S, \gamma_{ub})]$. □

Proposition 2. *Let $0 < \epsilon_\gamma < |\frac{\alpha_e - \alpha_d}{\gamma_d - \gamma_e}|$ and $0 < \epsilon \leq |\gamma_d - \gamma_e|$ for any $\tau^d, \tau^e \subseteq \rho$. For any given $S \in \{2, 3, \ldots, |\rho|\}$, RCA then generates all nondominated points of problem $[\boldsymbol{P}(S)]$.*

Proof. Let $S \in \{2, 3, \ldots, |\rho|\}$ and $R(S) = \{\tau^1, \tau^2, \ldots \tau^E\}$ be the set of ensembles with size of S generated by RCA solving $[\boldsymbol{P^{RCA}}(S, \gamma_{ub})]$ such that $z(\tau^1) = z(\tau_{S,\gamma_{ub}=1}) = (S, \alpha_1, \gamma_1)$, $z(\tau^2) = z(\tau_{S,\gamma_{ub}=\gamma_1-\epsilon}) = (S, \alpha_2, \gamma_2)$, $z(\tau^3) = z(\tau_{S,\gamma_{ub}=\gamma_2-\epsilon}) = (S, \alpha_3, \gamma_3)$, and so on. Then, using Proposition 1, any $\tau^d \in R(S)$ is nondominated with respect to $[\boldsymbol{P}(S)]$ satisfying $\alpha_1 < \alpha_2 < \cdots < \alpha_E$ and $\gamma_1 > \gamma_2 > \cdots > \gamma_E$ since the upper bound on the diversity metric, γ, is reduced systematically while coverage gap of the representative ensemble, α, is minimized throughout the process.

In order to get a contradiction, suppose that there exists a nondominated point of problem $[\boldsymbol{P}(S)]$ such that $z(\tau^e) \in Z(S)$ but $z(\tau^e) \notin R(S)$. Since RCA starts solving $[\boldsymbol{P^{RCA}}(S, \gamma_{ub})]$ with $\gamma_{ub} = 1$ and continues the process by reducing γ_{ub} until it becomes infeasible, there should exist $\tau^d, \tau^{d+1} \in R(S)$ for some $1 \leq d \leq E$ such that $\alpha_d < \alpha_e < \alpha_{d+1}$ and $\gamma_d > \gamma_e > \gamma_{d+1}$ hold. Then, τ^e is feasible to $[\boldsymbol{P^{RCA}}(S, \gamma_{ub} = \gamma_d - \epsilon)]$

since $0 < \varepsilon \leq \gamma_d - \gamma_e$ by definition, hence $\gamma_e \leq \gamma_d - \varepsilon$. In addition, $\alpha_e + \varepsilon_\gamma \gamma_e < \alpha_{d+1} + \varepsilon_\gamma \gamma_{d+1}$ is satisfied since $0 < \varepsilon_\gamma < \frac{\alpha_{d+1}-\alpha_e}{\gamma_e-\gamma_{d+1}}$. Then, this is a contradiction to τ^{d+1} being optimal to $[\boldsymbol{P^{RCA}}(S, \gamma_{ub} = \gamma_d - \varepsilon)]$. □

Proposition 3. *RCA generates all nondominated points of problem* $[\boldsymbol{P}]$.

Proof. Suppose that $z(\tau) = (S, \alpha_\tau, \gamma_\tau) \in Z$ is a nondominated point to $[\boldsymbol{P}]$ but $\tau \notin R(S)$. Then, by Proposition 2, there should exist an ensemble $\tau^e \in R(S)$ with a size of S that dominates τ with respect to problem $[\boldsymbol{P}(\boldsymbol{S})]$ and one of the following conditions will hold.

i. $\alpha_e < \alpha_\tau$ and $\gamma_e \leq \gamma_\tau$
ii. $\alpha_e \leq \alpha_\tau$ and $\gamma_e < \gamma_\tau$

Any of these conditions imply that $z(\tau) = (S, \alpha_\tau, \gamma_\tau)$ is dominated by $z(\tau^e) = (S, \alpha_e, \gamma_e)$. This is a contradiction to $z(\tau)$ being a nondominated point. □

Proposition 3 shows that RCA generates all nondominated points of problem $[\boldsymbol{P}]$ but not all points generated by RCA are nondominated with respect to $[\boldsymbol{P}]$. This is because RCA fixes the size of the ensembles and generates all nondominated points with respect to the quality and diversity metrics. If there are cases for which decreasing the size of an ensemble does not result in a deterioration in the quality or diversity of the ensemble, then the resulting ensemble will dominate the ensemble. This is why RCA conducts a nondominance check at the end of the process.

3.3. Consensus Generation Method (CGM)

RCA generates efficient ensembles, subsets of clustering solutions, which represent the preprocessed library well with respect to the coverage gap, diversity and cardinality measures. In this section, we develop three methods to generate a consensus clustering solution. While the first method is purely based on RCA and utilizes a consensus function to find a single clustering solution for each efficient ensemble, the second and third methods do not necessitate finding all efficient solutions. The second method generates a compromise EE to problem $[\boldsymbol{P}]$ and applies a consensus function on the compromise EE whereas the third method finds the clustering solution that best represents the library well.

3.3.1. Method 1 - A Consensus Generation Method using RCA (CGM-RCA)

Our first method, CGM-RCA, takes the set of EEs produced by RCA, R_{EE}, and the number of clusters, k^*, as an input and generates an *efficient consensus solution* (ECS) for each $\tau^e \in R_{EE}$. CGM-RCA uses a graph-based consensus function, Hybrid Bipartite Graph Formulation (HBGF) of Fern and Brodley (2004), to combine the clustering solutions in τ^e into a single clustering solution, $\pi^e = \Phi(\tau^e, k^*)$. While most of the graph-based approaches either focus on the relation between data points or between clusters, HBGF makes use of both associations in the graph representation and it is proven to work well.

3.3.2. Method 2 - A Consensus Generation Method using Compromise Programming (CGM-CP)

RCA produces a set of cluster ensembles since there does not exist a single subset of clustering solutions optimizing quality, diversity and the cardinality simultaneously. The best output would be the ideal point, τ^I, which is composed of the best possible values for each criterion, i.e. $(z_1(\tau^I), z_2(\tau^I), z_3(\tau^I)) = (0, 0, 0)$ but it is not possible to obtain under conflicting objectives. Therefore, instead of generating all EEs, CGM-CP uses τ^I as a reference point and solves $[\boldsymbol{P^{CGM-CP}}]$ in order to find a compromise ensemble $\tau^* = \min_{\tau \subseteq \rho} \max_{p=1,2,3} \lambda_p (z_p(\tau) - z_p(\tau^I))$ that minimizes a weighted Tchebycheff distance from τ^I (Ehrgott, 2005).

$[\boldsymbol{P^{CGM-CP}}]$

$$\text{Minimize} \quad D \tag{18}$$

subject to;

$$\sum_{\pi^j \in \rho} y_{ij} = 1 \qquad \forall \pi^i \in \rho \tag{19}$$

$$y_{ij} \leq y_{jj} \qquad \forall \pi^i, \pi^j \in \rho \tag{20}$$

$$\sum_{\pi^j \in \rho} y_{jj} = S \tag{21}$$

$$S \geq 2 \tag{22}$$

$$\alpha \geq y_{ij} \left(1 - NMI(\pi_i, \pi_j)\right) \qquad \forall \pi^i, \pi^j \in \rho \tag{23}$$

$$r_{ij} \leq y_{ii} \qquad \forall \pi^i, \pi^j \in \rho, \, i \neq j \tag{24}$$

$$r_{ij} \leq y_{jj} \qquad \forall \pi^i, \pi^j \in \rho, \, i \neq j \tag{25}$$

$$r_{ij} \geq y_{ii} + y_{jj} - 1 \qquad \forall \pi^i, \pi^j \in \rho, \, i \neq j \tag{26}$$

$$\gamma \geq r_{ij} \, NMI(\pi_i, \pi_j) \qquad \forall \pi^i, \pi^j \in \rho, \, i \neq j \tag{27}$$

$$D \geq \lambda_1 \alpha \tag{28}$$

$$D \geq \lambda_2 \gamma \tag{29}$$

$$D \geq \lambda_3 S \tag{30}$$

$$y_{ij}, r_{ij} \in \{0, 1\} \qquad \forall \pi^i, \pi^j \in \rho \tag{31}$$

$$S, \alpha, \gamma \geq 0 \tag{32}$$

In $[\boldsymbol{P^{CGM-CP}}]$, $\lambda = (\lambda_1, \lambda_2, \lambda_3)$ represent the weight vector that is used to scale the three criteria such that all criteria will take a value between [0,1]. Constraints (28)–(30) are included to measure the weighted Tchebycheff distance, D, from the scaled ideal point $\tau^I = (0, 0, 0)$. The efficient ensemble which is optimal to $[\boldsymbol{P^{CGM-CP}}]$ is denoted as τ^*. CGM-CP then uses HBGF of Fern and Brodley (2004) and applies its consensus function, $\Phi(\tau^*, k^*)$, to combine all clustering solutions in τ^* into a single clustering solution, π_{τ^*, k^*} where k^* represents the desired number of clusters.

3.3.3. Method 3: A Consensus Generation Method using a specialized version of RCA (CGM-sRCA)

For any given $S \geq 2$ value, RCA solves a bi-objective optimization problem and generates a number of clustering ensembles that minimizes coverage gap as well as diversity measure. However, the problem reduces to a single objective optimization problem when $S = 1$ since the diversity measure is not applicable. Therefore, our third method considers this special case and generates a consensus solution by solving $\pi^* = \text{argmin}_{\pi_i \in \rho} \max_{\pi_j \in \rho} (1 - NMI(\pi_i, \pi_j))$. That is, since the size of the clustering ensemble is set to 1, CGM-sRCA finds the clustering solution in the preprocessed library, $\pi^* \in \rho$, which best represents the other solutions in the ensemble by minimizing the coverage gap, α. Different from existing approaches, the process does not utilize a consensus function and any other external input in combining the clusterings into a single solution as CGM-sRCA also serves as a consensus function and a method for estimating the number of clusters, k^*.

4. Computational experiments

In this section, we first apply our algorithms on three small benchmark data sets (Iris, Wine, Glass) to evaluate the impact of our preprocessing algorithm, PPA, and to test the performance of our ensemble selection and consensus generation methods, CGM-RCA, CGM-CP, CGM-sRCA. We then apply one of our methods, CGM-CP, which is capable of generating a single efficient consensus solution that perform well on each criterion, to five larger benchmark data sets (Ecoli, Breast Cancer, Vehicle Silhouette, Segmentation (Image Segmentation), Satimage (Statlog)). We evaluate the overall performance of the proposed algorithms and compare its performance with the full-ensemble as well as other state-of-the-art ensemble selection methods. Lastly, we test the CGM-CP on a real-life sized data set (KDD99CUP) and compare its performance with that of the full-ensemble consensus solution. All data sets are obtained from UCI Machine Learning Repository (Kelly, Longjohn, & Nottingham, 2023). Table 1 presents the number of data points, features, and true class labels in the benchmark data sets.

4.1. Initial library generation

In our computational experiments, we create the initial library of clustering solutions using repeated runs of k-means algorithm because of its simplicity, computational efficiency and performance on large data sets (Akbari et al., 2015; Alizadeh et al., 2014; Azimi & Fern, 2009; Fern & Lin, 2008; Naldi et al., 2013; Pividori, Stegmayer, & Milone, 2016; Yang et al., 2017). The k-means algorithm iteratively constructs clusters by defining a centroid for each cluster and assigning each data point to the nearest centroid. The algorithm calculates the distance between each data point and its cluster center and aims at minimizing total distance. It starts with randomly selected k cluster centroids and relocates their positions to improve total distance. It stops when it converges to a solution where the centroids no longer move (MacQueen et al., 1967). We generate the initial library by carrying out the algorithm for all k values from 2 to $\sqrt{N}$ as a rule of thumb (Fred & Jain, 2002). Because of the randomness in the positioning of the initial centroids, we replicate the process five times for each k value and eliminate identical clustering solutions, if any (Boongoen & Iam-On, 2018; Nguyen & Caruana, 2007).

4.2. Evaluating clustering solutions

In order to evaluate the performance of our methods and compare the output with respect to the initial ensemble, we utilize both internal and external cluster validity metrics (Liu, Li, Xiong, Gao, & Wu, 2010). While NMI is used as an external validation metric to measure the extent to which cluster labels match the true labels, internal validation measures including Davies–Bouldin Index (Davies & Bouldin, 1979), Silhouette Index (Rousseeuw, 1987), and Dunn's Index (Dunn, 1974) are used to evaluate how well the clustering solution fits the data without reference to external information. We next define the measures we use to evaluate the performance of our clustering solutions.

Definition 7. Let $\pi \in \Pi$ be a clustering solution for data points in X and π^{true} be the true clustering. Then, *accuracy* of π with respect to π^{true} is given by the following expression:

$$\psi_\pi = NMI(\pi, \pi^{true}) \tag{33}$$

Please note that ψ values close to 1 are desired so as to have a better match with the true clustering solution.

Definition 8. Let $\pi = \{C_1^\pi, \dots, C_k^\pi\} \in \Pi$ be a clustering solution for a set of N data points $X = \{x_1, x_2, \dots, x_N\}$ that partitions the data set into k clusters, such that $X = \bigcup_{u=1}^k C_u^\pi$ where $C_u^{\pi_i} \cap C_v^{\pi_i} = \emptyset, \forall u \neq v$. Then, *Davies–Bouldin Index* of a clustering solution π is calculated as follows:

$$DBI_\pi = \frac{1}{k}\sum_{u=1}^{k} \max_{v \neq u} \left\{ \frac{\frac{1}{n_u}\left(\sum_{x_o \in C_u^\pi} d(x_o, \mu_u)\right) + \frac{1}{n_v}\left(\sum_{x_o \in C_v^\pi} d(x_o, \mu_v)\right)}{d(\mu_u, \mu_v)} \right\} \tag{34}$$

where n_u represents the number of data points in cluster C_u^π, μ_u denotes the cluster center and $d(.,.)$ is the distance measure.

For each cluster u, DBI finds the most similar cluster v measuring the separation between the clusters and the similarity within the clusters. The lower the value, the better separation of the clusters and the compactness within the cluster.

Table 1
Properties of benchmark data sets.

Data set	# of data points (n)	# of features (f)	# of true labels (k)
Iris	150	4	3
Wine	178	13	3
Glass	214	10	6
Ecoli	336	7	8
Breast Cancer	683	9	2
Vehicle Silhouette	946	18	4
Segmentation	2310	19	7
Satimage	6435	36	7
KDD99CUP	494,021	41	23

Definition 9. Let $\pi = \{C_1^\pi, \dots, C_k^\pi\} \in \Pi$ be a clustering solution for $X = \{x_1, x_2, \dots, x_N\}$ where $x_o \in X$ is assigned to cluster C_u^π. Then, *Silhouette Index* of π is given by the following expression:

$$SI_\pi = \frac{\sum_{x_o \in X} s(x_o)}{N} \tag{35}$$

where

$$s(x_o) = \begin{cases} 1 - a(x_o)/b(x_o) & \text{if } a(x_o) < b(x_o) \\ 0 & \text{if } a(x_o) = b(x_o) \\ b(x_o)/a(x_o) - 1 & \text{if } a(x_o) > b(x_o) \end{cases} \tag{36}$$

$$a(x_o) = \frac{\sum_{x_\omega \in C_u^\pi, x_\omega \neq x_o} d(x_o, x_\omega)}{|C_u^\pi| - 1} \tag{37}$$

$$b(x_o) = \min_{v=1,\dots,k \; v\neq u} \left\{ \frac{\sum_{x_\omega \in C_v^\pi} d(x_o, x_\omega)}{|C_v^\pi|} \right\} \tag{38}$$

In Definition 9, $a(x_o)$ is the average distance between data point x_o and the other data points in the same cluster, C_u^π, while $b(x_o)$ is the minimum average distance between x_o and all the data points in the other clusters, C_v^π $v \neq u$. Therefore, SI_π measures how similar a data point is to the other data points in its cluster compared to the data points in the other clusters and defines a silhouette value for each data point. The higher the SI value, the better the match between the data point and its cluster.

Definition 10. Let $\pi = \{C_1^\pi, \dots, C_k^\pi\} \in \Pi$ be a clustering solution for a set of N data points $X = \{x_1, x_2, \dots, x_N\}$. *Dunn's Index* of π is calculated as follows:

$$DI_\pi = \frac{\min_{\forall (u,v): u \neq v} \left\{ \min_{\forall x_o \in C_u^\pi, \forall x_\omega \in C_v^\pi} d(x_o, x_\omega) \right\}}{\max_{u=1,\dots,k} \left\{ \max_{\forall x_o, x_\omega \in C_u^\pi} d(x_o, x_\omega) \right\}} \tag{39}$$

For a clustering solution, DI metric measures the ratio between:

- the minimum (the worst) inter-cluster distance, i.e. the minimal distance between any two clusters
- the maximum (the worst) intra-cluster distance, i.e. the maximal distance between any two objects in the same cluster

Since the aim in clustering algorithms is to identify sets of clusters that are compact but well-separated, a higher DI value indicates a better clustering.

4.3. Estimating the number of clusters

In consensus generation process, the desired number of clusters is required as an input if a consensus function is to be used. However, most of the time this information is not available for unsupervised learning methods and the number of clusters varies between 2 and $\sqrt{N}$ in the initial library. Therefore, we evaluate the performance of our methods assuming the true number of clusters, k, is not known to the analyst a priori.

Table 2
Average initial and preprocessed library characteristics.[a]

Data set	Library	Size	κ_{min}	κ_{avg}	κ_{max}	ψ_{min}	ψ_{avg}	ψ_{max}	DBI_{avg}	SI_{avg}	DI_{avg}
Iris	Π	52	0.55	0.75	0.80	0.58	0.66	0.76	0.98	0.52	0.08
	ρ	37	0.61	0.75	0.80	0.59	0.67	0.76	0.95	0.55	0.08
Wine	Π	52	0.51	0.77	0.81	0.34	0.37	0.44	0.57	0.68	0.03
	ρ	30	0.54	0.76	0.80	0.36	0.38	0.44	0.55	0.69	0.03
Glass	Π	64	0.41	0.68	0.74	0.20	0.38	0.45	1.09	0.41	0.04
	ρ	33	0.47	0.68	0.75	0.21	0.36	0.45	1.09	0.48	0.04

[a] The values over 10 libraries are reported per cell.

Table 3
Performance results on data sets.

	Iris				Wine				Glass			
	DBI	SI	DI	ψ	DBI	SI	DI	ψ	DBI	SI	DI	ψ
True Clustering	0.75	0.66	0.06	1.00	1.52	0.25	0.01	1.00	3.74	−0.25	0.02	1.00
LibraryBest[a,d]	0.71	0.71	0.11	0.76	0.49	0.80	0.02	0.44	1.00	0.55	0.07	0.45
FullEnsemble[b,d]	0.50	0.79	0.27	0.76	0.76	0.60	0.04	0.41	1.09	0.55	0.04	0.33
CGM-RCA[c]	0.66	0.74	0.10	**0.76**	0.53	0.73	0.02	**0.43**	1.02	0.56	0.04	**0.39**
CGM-CP[d]	0.79	0.67	0.10	0.72	0.50	0.81	0.03	**0.43**	1.05	0.59	0.05	0.36
CGM-sRCA[d]	0.90	0.57	0.06	0.67	0.53	0.72	0.03	0.38	0.99	0.61	0.05	0.38

[a] The values are reported for the clustering solution with the maximum accuracy, ψ, in the initial library, $|\Pi|$.
[b] The full-ensemble solution is obtained by applying the consensus function to the initial library, $\Phi(\Pi, k^*)$.
[c] The values over 30, 31, 34 efficient consensus solutions are reported for Iris, Wine, and Glass data sets, respectively.
[d] The values over 10 libraries are reported.

As generating efficient consensus solutions using CGM-RCA and CGM-CP requires the number of clusters as an input, we find an estimate, k^*, using a combination of clustering validity indices to capture different aspects of clustering solutions in the ensemble as suggested in Kryszczuk and Hurley (2010). We first generate a compromise efficient ensemble using CGM-CP, then for the given cluster ensemble $\tau \subseteq \rho$, we find the clustering solution, $\hat{\pi} \in \tau$, that maximizes a linear combination of the internal validation measures,

$$\hat{\pi} = \underset{\pi \in \tau}{\operatorname{argmax}}\, SI_\pi + DI_\pi - DBI_\pi \tag{40}$$

where all indices are normalized to the range of $[0, 1]$. Since higher values of SI and DI but lower values of DBI are desired to have a more compact and well-separated clustering, we set the value of k^* based on this clustering solution, i.e. $k^* = k^*_{\hat{\pi}}$.

4.4. Results

In this section, we first present the results on Iris, Wine and Glass data sets and evaluate the contribution of our preprocessing algorithm, PPA, and ensemble selection methods, CGM-RCA, CGM-CP and CGM-sRCA, to the consensus generation process. As CGM-CP is designed as a practical tool to generate a compromise solution, we then evaluate the performance of CGM-CP on larger data sets, Ecoli, Breast Cancer and Vehicle data sets, and compare its performance against those of the state-of-the-art ensemble selection methods.

4.4.1. Experiments with Iris, Wine and Glass data sets

Aiming to evaluate the contribution of PPA to the consensus generation process, we perform experiments with Iris, Wine and Glass data sets and compare the initial and preprocessed library characteristics which are presented in Table 2. The results show that PPA improves minimum agreement levels, κ_{min}, hence works well in detecting the outlier solutions. In addition, PPA yields better average DBI and average SI values in all data sets whereas DI values stay the same. Therefore, overall performance of the clustering solutions improve in terms of the internal measures. The results in Table 2 also show that the minimum accuracy, ψ_{min}, is improved by PPA while ψ_{max} values remain the same. This implies that PPA eliminates the worst-performing clustering solutions while keeping the solutions with high accuracy. The average accuracy value, ψ_{avg}, also improves for Iris and Wine data sets, whereas ψ_{avg} decreases for Glass data set. This is because finding true partitioning using internal measures is hard for the Glass data set where the clusters are not easily separable and the true number of clusters is relatively higher compared to the other data sets as shown in Fig. 2.

We also test the performance of our ensemble selection and consensus generation algorithms on Iris, Wine and Glass data sets for each of which the results are presented in Table 3. In order to assess how well their resulting clustering solutions are, we compare their cluster validity indices against those of the best-performing clustering solution in the initial library (LibraryBest) as well as the full-ensemble solution which is obtained by applying the consensus function to the initial library (FullEnsemble). Table 3 shows that CGM-RCA generates on average the most accurate solutions among the three methods for all three data sets. Furthermore, consensus solutions produced by CGM-RCA are at least as good as the full ensemble solutions for all three benchmark data sets. This is particularly important as it demonstrates the contribution of our ensemble selection method. The overall performance of CGM-CP is also promising in terms internal and external cluster validity indices considering that it does not necessitate the generation of all efficient ensembles as in CGM-RCA. It could not converge to the most accurate solutions in the library but it yields more accurate solutions than the full-ensemble solution for Wine and Glass data sets while generating only a compromise efficient ensemble. When compared to CGM-RCA and CGM-CP, CGM-sRCA demonstrates a moderate performance but it has the unique capability of consolidating the set of clustering solutions into a single solution without selecting a cluster ensemble. That is, it serves as a novel consensus function, which does not require an estimate on the number of clusters.

In Fig. 2, we illustrate examples of consensus solutions produced by CGM-RCA, CGM-CP, and CGM-sRCA as well as the true clustering solutions in two dimensions using Principal Component Analysis (PCA). They correspond to the average-performing solutions for which the accuracy values are 0.70, 0.40, and 0.39 for Iris, Wine and Glass data sets, respectively. Fig. 2 shows that the final clusters generated by CGM-RCA and CGM-CP are more compact and separate compared to those obtained by CGM-sRCA for all benchmark data sets. However, CGM-sRCA provides more promising results for Glass data set where the true clusters are less compact and separate than the true clusters for Wine and Glass data sets.

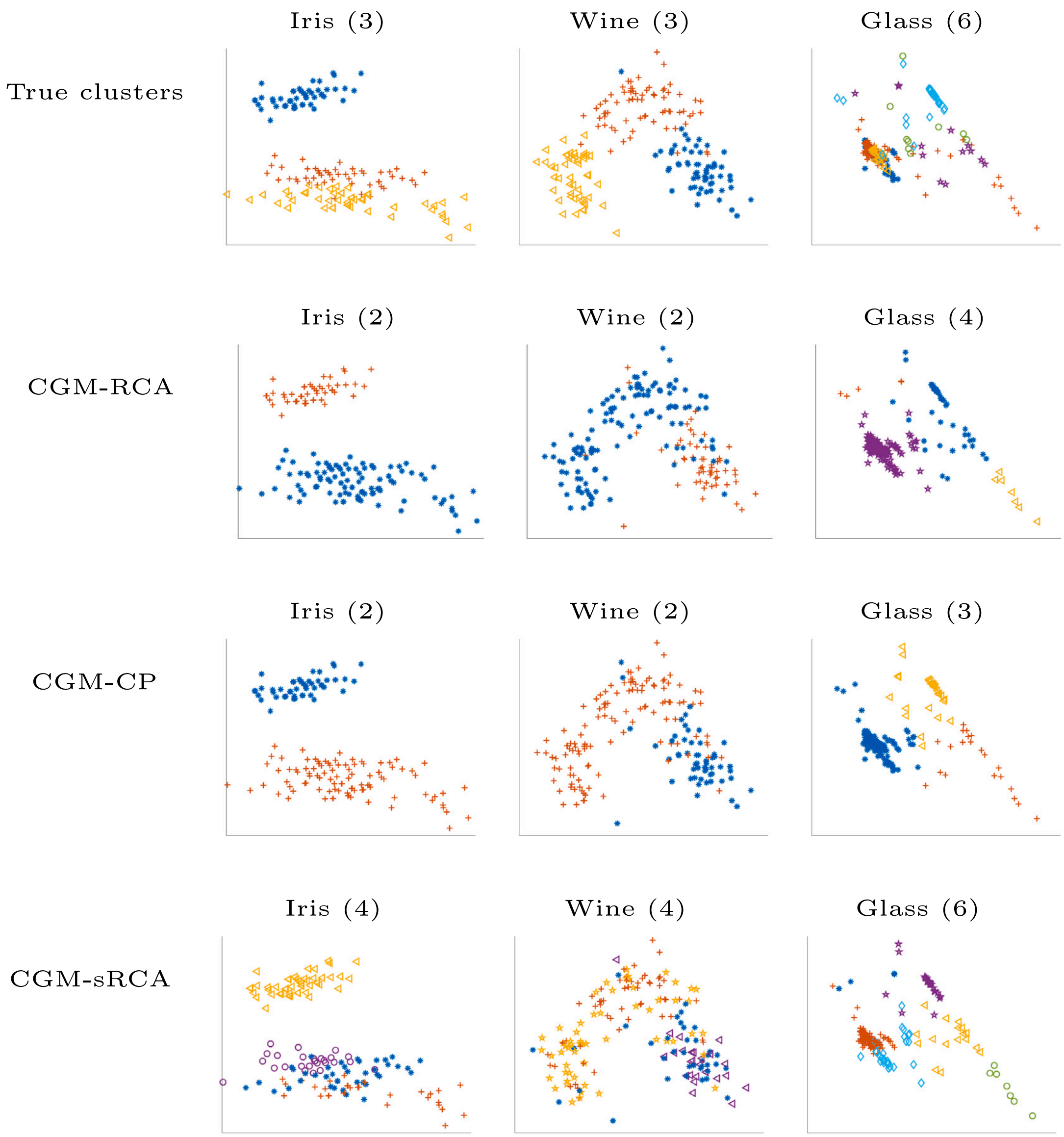

Fig. 2. Distribution of data points on the respective two principal components after PCA.

In addition to the average-performing clustering solutions, we also present the most accurate efficient consensus solutions obtained by any of the proposed methods in Fig. 3. The accuracy values of these solutions are 0.81, 0.44, and 0.43 for Iris, Wine and Glass data sets while the estimated number of clusters are 3, 2, and 6, respectively. The most accurate solutions for all three benchmark data sets are obtained by CGM-CP. CGM-CP is also able to identify the true number of clusters for Iris and Glass data sets while it estimates 2 clusters for Wine data set for which the true number of clusters is 3.

While CGM-RCA generates all efficient ensembles with respect to the multiple criteria, which is helpful to characterize the trade-offs in between, CGM-CP serves as a practical and effective tool to generate a compromise efficient solution. As there does not exist any other algorithm which is capable of finding all efficient ensembles as in CGM-RCA and the most accurate solutions for Iris, Wine, and Glass data sets are found by CGM-CP in our experiments, we conduct further experiments using CGM-CP on larger data sets and compare our results with other state of the art ensemble selection methods in the following section.

4.4.2. Experiments with Ecoli, Breast Cancer, Vehicle Silhouette, Segmentation, and Satimage data sets

In this section, in addition to the Iris, Wine, and Glass data sets, CGM-CP is applied to five larger data sets: Ecoli, Breast Cancer, Vehicle Silhouette, Segmentation and Satimage from the UCI Machine Learning Repository (Kelly et al., 2023). The experiments are conducted on these data sets so that the accuracy of the final consensus solutions can be compared against the following state-of-the-art cluster ensemble selection methods: Cluster and Select (CAS) (Fern & Lin, 2008), Adaptive Cluster Ensemble Selection (ACES) (Azimi & Fern, 2009), Hierarchical Cluster Ensemble Selection (HCES) (Akbari et al., 2015).

Similar to the previous experiments on Iris, Wine, and Glass data sets, an initial library of clustering solutions is generated by five repeated runs of k-means algorithm, where k values range between 2 and $\sqrt{N}$. Then, PPA is applied to eliminate the outlier solutions from the initial library and CGM-CP is applied to the preprocessed library to generate a compromise consensus solution.

Table 4 presents the accuracy values of the final consensus solutions generated by the full-ensemble and the compromise consensus solution

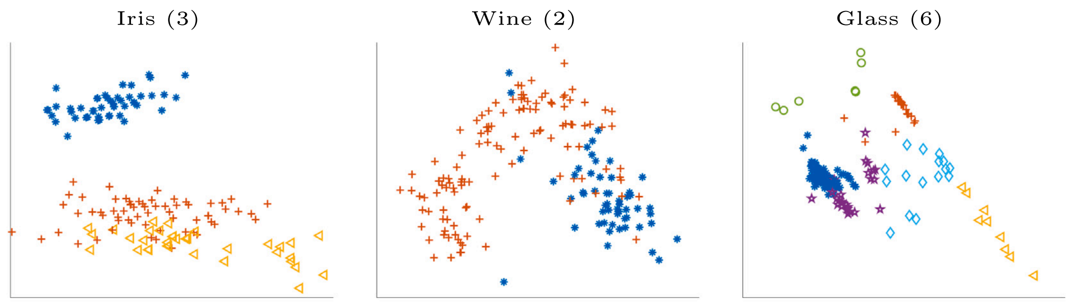

Fig. 3. Distribution of data points according to the most accurate solutions obtained by CGM-CP.

Table 4
Comparison of accuracy (ψ) values.

	Full ensemble[a,b]	CGM-CP[b]	Full ensemble[c]	HCES[c]	Full ensemble[d]	ACES[d]	Full ensemble[d]	CAS[d]
Iris	0.761	0.761	**0.919**	**0.919**	0.744	0.744	0.744	0.613
Wine	0.411	0.433	0.313	0.478	0.474	**0.680**	0.474	0.612
Glass	**0.410**	**0.410**	0.386	0.402	0.269	0.269	0.269	0.301
Ecoli	**0.695**	0.649	0.513	0.536	–	–	–	–
Breast Cancer	**0.492**	**0.493**	0.435	0.485	–	–	–	–
Vehicle Silhouette	0.203	**0.226**	–	–	0.146	0.146	0.146	0.122
Segmentation	0.569	**0.584**	–	–	0.406	0.406	0.406	0.550
Satimage	0.543	0.546	0.426	**0.565**	–	–	–	–

[a] The full-ensemble solution is obtained by applying the consensus function to the initial library.
[b] The values for a single replication are reported.
[c] As reported in Akbari et al. (2015).
[d] As reported in Azimi and Fern (2009).

obtained by CGM-CP. For CAS and ACES, the corresponding accuracy values are obtained from the work of Azimi and Fern (2009) where the authors compare the performance of ACES with their implementation of CAS, and for HCES, the accuracy values are obtained from the work of Akbari et al. (2015).

According to the results presented in Table 4, when the performance of CGM-CP is compared with the performance of the full-ensemble, it is observed that except for Ecoli data set, CGM-CP produces a compromise consensus solution that is at least as good as the one of the full-ensemble for all the other data sets. CGM-CP also provides one of the most accurate solutions for Glass, Breast Cancer, Vehicle Silhouette and Segmentation data sets, where the difference between the full-ensemble and CGM-CP is very small for Glass and Breast Cancer.

The consensus solutions obtained by HCES are at least as good as the one of its full-ensemble and outperforms the other ensemble selection methods on Iris and Satimage data sets. For Iris data set, the final consensus solution obtained by HCES has a superior accuracy compared to all three other methods. However, it should be noted that the consensus solution obtained by the full-ensemble for Iris in HCES experiments has already a high accuracy value. This means that the superior performance of HCES for Iris can also be due to its library generation and consensus generation steps. Similarly, the full-ensemble consensus generated for Ecoli in CGM-CP has the highest accuracy value and it can also be due to the library generation or the consensus generation methods.

When CAS is considered, the performance of the final consensus solutions over the full-ensemble is improved for Wine, Glass, and Segmentation data sets. Yet, for none of these data sets, CAS provides the most accurate solutions.

Unlike CAS and HCES, ACES focuses only on the diversity of the ensembles by proposing the full-ensemble for stable data sets and a high diversity subset of solutions for non-stable data sets. Among the experimented data sets, only Wine is recognized as non-stable and thus a different subset of ensemble than the full-ensemble is proposed by ACES. It should be noted that ACES improves the performance of the final consensus solution significantly for Wine data set compared to its full-ensemble and provides the most accurate solution. However, it does not provide the most accurate solutions for the other experimented data sets.

4.4.3. Experiments with KDD Cup 1999 Data

In order to illustrate the scalability of our approach, further experiments have been carried out on a data set called *KDD Cup 1999 Data* (KDDCUP99). While only 10% of this data set is used, it remains as a significantly large data set with 494,021 data points of 41 dimensions belonging to 23 classes (Stolfo & Lee, 1999). This data set has been used for the purpose of training supervised learning algorithms, which are capable of differentiating bad connections (i.e. intrusions or attacks) from legitimate connections on a computer network. Due to the large amount of data points, we have generated an initial library of 245 clustering solutions by applying the k-means algorithm for all k values between 2 and 50 and replicating this process five times.

The structure of the KDDCUP99 data set is highly unbalanced with more than 98% of the data points belonging to only three classes while hyper-graph based consensus functions usually perform better for nearly balanced clusters (Golalipour et al., 2021; Topchy, Jain, & Punch, 2004). Therefore, we have applied two different consensus functions provided by the machine learning package in Python (Pedregosa et al., 2011), HBGF of Fern and Brodley (2004) and meta-clustering algorithm (MCLA) of Strehl and Ghosh (2002). In Table 5, we report the accuracy values of the final consensus solutions obtained by CGM-CP and the full ensemble solution. While the final consensus solution obtained by the full ensemble with HBGF performs slightly better than the one of CGM-CP, CGM-CP with MCLA improves the cluster ensemble performance and produces a compromise consensus solution that is about 2.5% more accurate than the full-ensemble solution.

5. An application to the customer segmentation problem

Customer segmentation is the process of dividing customers in a given market into homogeneous groups, based on their shared characteristics or preferences with the purpose of improving the efficiency of the marketing strategies. This process also helps retailers understand

Table 5
Comparison of accuracy (ψ) values for KDDCUP99.

Consensus function	Full ensemble[a,b]	CGM-CP[b]
HBGF	**0.560**	0.546
MCLA	0.744	**0.763**

[a] The full-ensemble solutions are obtained by applying the consensus functions to the initial library.

[b] The values for a single replication are reported.

the heterogeneity in consumer preferences and tailor their assortments to specific consumer segments (Smith, 2009).

Santi, Aloise, and Blanchard (2016) develop a model for heterogeneous clustering problem (HCP) where the aim is to partition the individuals/customers into groups with similar preference structures and to come up with a consensus clustering solution for each group. The authors apply their algorithm on a real-life application problem, which aims to understand differences in the perceptions of assortments of chocolate candy. They collect data from 189 graduate students through an online sorting study where each participant is asked to partition 20 different chocolate candies into piles using as many piles as they want. The chocolate candies in each pile is considered as similar according to their own perception. In order to segment these customers and their clusterings, Santi et al. (2016) turn the pile information of clusters provided by each individual into a dissimilarity/distance matrix using the procedure of Takane (1980). This distance matrix is then used to assign customers/clusterings to g number of groups and simultaneously find a consensus clustering solution for each customer segment to minimize the sum of dissimilarities between each object and its assigned median, conditional on the group membership. By using the traditional 'elbow in the curve' method, Santi et al. (2016) come up with the best number of groups as $g = 2$, which means the customers are segmented in two groups (Experts and Novices) where a consensus partitioning of the chocolate candies found for each group separately.

We model this problem as a multi-objective ensemble selection problem where the partitions made by each customer are considered as the initial library of clustering solutions. We then apply our algorithms to find a customer segmentation as well as a consensus clustering solution for each customer segment. After identifying/removing the outlier customers' solutions using PPA, CGM-CP is applied to the preprocessed library and an efficient ensemble of customers representing all customers well in terms of quality, diversity and size criteria is produced. The size of this efficient ensemble determines the number of customer segments, g, while clustering solutions in the efficient ensemble correspond to the representative clusterings/customers of each segment. Instead of combining these ensemble members into a single consolidated clustering solution, we consider these representatives as the consensus solutions for each customer segment as in Santi et al. (2016).

Since Santi et al. (2016) propose to group customers into two groups: Novices and Experts, we apply CGM-CP with $s = 2$, as well. Table 6 presents the results when the customers are segmented in two groups with CGM-CP and HCP as well as the consensus solutions obtained by HCP according to Santi et al. (2016). When the quality and diversity measures are considered only, it is observed that CGM-CP finds a much better set of representative/consensus solutions than the consensus solutions found by HCP, with respect to both criteria. This is rather expected as the objective of HCP is not exactly the same as the quality and diversity metrics used in CGM-CP. Therefore, to be able to make a fair comparison for the representation capabilities of the solutions obtained by both methods, we also measure the intra-cluster similarity between a representative/consensus solution and the customers it represents by the normalized mutual information (NMI). For each group, we report the minimum, average, and maximum intra-cluster similarity values between the representative/consensus and the customers in the same group. Lastly, we report the overall average of the NMI values between all customers and their representative/consensus clustering solutions. The results show that the customers share overall more mutual information with the representative/consensus solutions found by CGM-CP with $s = 2$ compared to the consensus solutions obtained by HCP with $g = 2$.

6. Conclusions

This paper develops a multi-objective optimization-based approach to the cluster ensemble selection problem and designs a solution framework to find a consensus clustering solution without knowing the true number of clusters a priori. Given an initial library of clustering solutions, the framework starts with the application of a preprocessing algorithm, PPA, that eliminates outlier solutions to obtain more reliable ensembles and thus more accurate and robust consensus solutions. Then, a multi-objective optimization model is developed to find efficient subsets of clustering solutions that represent the preprocessed library well in terms of size, quality, and diversity measures. Lastly, the clustering solutions in the efficient representative ensemble are combined into a single consensus solution. To the best of our knowledge, this is the first study that develops and applies exact multi-objective optimization techniques to the ensemble selection and consensus generation problems.

Our first method, CGM-RCA, is designed to generate all efficient representative ensembles and find efficient consensus solutions for each representative ensemble while CGM-CP only finds a compromise efficient ensemble for the multi-objective ensemble selection problem. Both CGM-RCA and CGM-CP use the estimated number of clusters and then utilize a consensus function to combine the clusterings in the ensemble into a single consensus solution. Our third method, CGM-sRCA, introduces a novel consensus generation method that finds the clustering solution that best represents the clusterings in the preprocessed library. That is, CGM-sRCA does not apply a consensus function hence does not necessitate the estimation of the true number of clusters. Our experimental results show that selecting representative ensembles using multi-objective optimization methods improve the cluster ensemble performance as they help the analyst select a representative library of clustering solutions, producing better consensus solutions when compared to full-ensemble solution. Although CGM-sRCA shows a moderate performance in terms of accuracy when compared to CGM-RCA and CGM-CP, it also serves as a practical tool for selecting a representative ensemble and finding a consensus solution.

We conduct experiments on different-sized data sets on which we evaluate the contribution of our preprocessing and ensemble selection strategies. Since there does not exist any other algorithm that guarantees finding efficient consensus solutions with respect to multiple criteria, we compare the performance of CGM-CP against those of existing algorithms and show that it produces promising results. We also apply our algorithm on a customer segmentation problem to show its practical contribution. Our results show that utilizing multi-objective optimization methods in cluster ensembles helps to produce better performing consensus solutions and preference-based cluster ensembles in practical applications.

Overall, the approach proposed in this paper provides not only a multi-objective cluster ensemble selection approach but also a method to find a set of efficient *representative* solutions. As a future research, a rule-based approach utilizing the characteristics of the data set in the efficient ensemble generation process could be studied. Moreover, an interactive multi-objective optimization approach can be designed to incorporate the preferences of the decision-makers (DMs) into the ensemble selection and consensus generation process. By eliciting the preference information from the DMs, weights or aspiration levels could be attached to the criteria used in ensemble selection problem. Another next step could be performing a case study where the multi-objective cluster ensemble selection approach is applied to a large scale real-life instance.

Table 6
Chocolate candy application: Customer segmentation results for two groups.

			CGM-CP (s = 2)	HCP (g = 2)
Ensemble/representatives (τ) selection[a]		Coverage gap, α_τ	**0.58**	0.78
		Diversity, γ_τ	**0.59**	0.62
Customer segmentation[b]	Group 1	Min. NMI	**0.49**	0.29
		Avg. NMI	0.56	0.56
		Max. NMI	0.74	**0.76**
	Group 2	Min. NMI	**0.42**	0.22
		Avg. NMI	**0.64**	0.54
		Max. NMI	**0.87**	0.79
	Overall	Avg. NMI	**0.60**	0.54

[a] The smaller coverage gap α_τ values indicate better quality in representation. The smaller γ_τ value is, the more diverse the representative ensemble is.
[b] For each customer segment, minimum, average and maximum intra-cluster similarity values are reported using Normalized Mutual Information (NMI) between its representative/consensus clustering and clusterings of the customers it represents.

Acknowledgments

Computational resources have been provided by the Consortium des Équipements de Calcul Intensif (CÉCI), funded by the Fonds de la Recherche Scientifique de Belgique (F.R.S.-FNRS) under Grant No. 2.5020.11 and by the Walloon Region.

References

Abbasi, S.-o., Nejatian, S., Parvin, H., Rezaie, V., & Bagherifard, K. (2019). Clustering ensemble selection considering quality and diversity. *Artificial Intelligence Review*, *52*(2), 1311–1340.

Akbari, E., Dahlan, H. M., Ibrahim, R., & Alizadeh, H. (2015). Hierarchical cluster ensemble selection. *Engineering Applications of Artificial Intelligence*, *39*, 146–156.

Alizadeh, H., Minaei-Bidgoli, B., & Parvin, H. (2014). To improve the quality of cluster ensembles by selecting a subset of base clusters. *Journal of Experimental & Theoretical Artificial Intelligence*, *26*(1), 127–150.

Ayad, H., & Kamel, M. (2003). Finding natural clusters using multi-clusterer combiner based on shared nearest neighbors. In *International workshop on multiple classifier systems* (pp. 166–175). Springer.

Ayad, H. G., & Kamel, M. S. (2010). On voting-based consensus of cluster ensembles. *Pattern Recognition*, *43*(5), 1943–1953. http://dx.doi.org/10.1016/j.patcog.2009.11.012, URL https://www.sciencedirect.com/science/article/pii/S0031320309004312.

Azimi, J., & Fern, X. (2009). Adaptive cluster ensemble selection. In *Twenty-first international joint conference on artificial intelligence*.

Berkhin, P. (2006). A survey of clustering data mining techniques. In *Grouping multidimensional data* (pp. 25–71). Springer.

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

Davies, D. L., & Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2), 224–227.

Dunn, J. C. (1974). Well-separated clusters and optimal fuzzy partitions. *Journal of Cybernetics*, *4*(1), 95–104.

Ehrgott, M. (2005). *Multicriteria Optimization, vol. 491*. Springer Science & Business Media.

Fern, X. Z., & Brodley, C. E. (2003). Random projection for high dimensional data clustering: A cluster ensemble approach. In *Proceedings of the 20th international conference on machine learning* (pp. 186–193).

Fern, X. Z., & Brodley, C. E. (2004). Solving cluster ensemble problems by bipartite graph partitioning. In *Proceedings of the twenty-first international conference on machine learning* (p. 36). ACM.

Fern, X. Z., & Lin, W. (2008). Cluster ensemble selection. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, *1*(3), 128–141.

Fred, A. L., & Jain, A. K. (2002). Data clustering using evidence accumulation. In *Object recognition supported by user interaction for service robots: vol. 4*, (pp. 276–280). IEEE.

Fred, A. L., & Jain, A. K. (2005). Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(6), 835–850.

Gionis, A., Mannila, H., & Tsaparas, P. (2007). Clustering aggregation. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *1*(1), 4–es.

Golalipour, K., Akbari, E., Hamidi, S. S., Lee, M., & Enayatifar, R. (2021). From clustering to clustering ensemble selection: A review. *Engineering Applications of Artificial Intelligence*, *104*, Article 104388.

Hadjitodorov, S. T., Kuncheva, L. I., & Todorova, L. P. (2006). Moderate diversity for better cluster ensembles. *Information Fusion*, *7*(3), 264–275.

Hines, W. W., Montgomery, D. C., & Borror, D. M. G. C. M. (2008). *Probability and statistics in engineering*. John Wiley & Sons.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. *ACM Computing Surveys (CSUR)*, *31*(3), 264–323.

Jia, J., Xiao, X., Liu, B., & Jiao, L. (2011). Bagging-based spectral clustering ensemble selection. *Pattern Recognition Letters*, *32*(10), 1456–1467.

Kelly, M., Longjohn, R., & Nottingham, K. (2023). UCI machine learning repository. URL https://archive.ics.uci.edu.

Kryszczuk, K., & Hurley, P. (2010). Estimation of the number of clusters using multiple clustering validity indices. In *International workshop on multiple classifier systems* (pp. 114–123). Springer.

Kuncheva, L. I., & Hadjitodorov, S. T. (2004). Using diversity in cluster ensembles. In *2004 IEEE international conference on systems, man and cybernetics (IEEE Cat. No. 04CH37583), vol. 2* (pp. 1214–1219). IEEE.

Kuncheva, L. I., & Vetrov, D. P. (2006). Evaluation of stability of k-means cluster ensembles with respect to random initialization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(11), 1798–1808.

Law, M. H., Topchy, A. P., & Jain, A. K. (2004). Multiobjective data clustering. In *Proceedings of the 2004 IEEE computer society conference on computer vision and pattern recognition, 2004, vol. 2* CVPR 2004, (p. II). IEEE.

Li, F., Qian, Y., Wang, J., Dang, C., & Jing, L. (2019). Clustering ensemble based on sample's stability. *Artificial Intelligence*, *273*, 37–55.

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

MacQueen, J., et al. (1967). 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.

Naldi, M. C., Carvalho, A., & Campello, R. J. (2013). Cluster ensemble selection based on relative validity indexes. *Data Mining and Knowledge Discovery*, *27*, 259–289.

Nguyen, N., & Caruana, R. (2007). Consensus clusterings. In *Seventh IEEE international conference on data mining* (pp. 607–612). IEEE.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pividori, M., Stegmayer, G., & Milone, D. H. (2016). Diversity control for improving the analysis of consensus clustering. *Information Sciences*, *361*, 120–134.

Reutterer, T., & Dan, D. (2020). Cluster analysis in marketing research. In C. Homburg, M. Klarmann, & A. Vomberg (Eds.), *Handbook of market research* (pp. 1–29). Cham: Springer International Publishing, http://dx.doi.org/10.1007/978-3-319-05542-8_11-2.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53–65.

Santi, É., Aloise, D., & Blanchard, S. J. (2016). A model for clustering data from heterogeneous dissimilarities. *European Journal of Operational Research*, *253*(3), 659–672.

Saxena, A., Prasad, M., Gupta, A., Bharill, N., Patel, O. P., Tiwari, A., et al. (2017). A review of clustering techniques and developments. *Neurocomputing*, *267*, 664–681. http://dx.doi.org/10.1016/j.neucom.2017.06.053, URL https://www.sciencedirect.com/science/article/pii/S0925231217311815.

Sayın, S. (2000). Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming. *Mathematical Programming*, *87*(3), 543–560.

Smith, S. A. (2009). Optimizing retail assortments for diverse customer preferences. In N. Agrawal, & S. A. Smith (Eds.), *Retail Supply Chain Management: Quantitative Models and Empirical Studies* (pp. 183–205). Boston, MA: Springer US, http://dx.doi.org/10.1007/978-0-387-78902-6_8.

Steuer, R. E. (1986). *Multiple criteria optimization: Theory, computation, and application, vol. 233*.

Stolfo, S., & Lee, W. (1999). Kdd'99 classifier learning contest. URL http://kdd.ics.uci.edu/databases/kddcup99/task.html.

Strehl, A., & Ghosh, J. (2002). Cluster ensembles—A knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, *3*(Dec), 583–617.

Takane, Y. (1980). Analysis of categorizing behavior by a quantification method. *Behaviormetrika*, *7*(8), 75–86.

Tomar, D., & Agarwal, S. (2013). A survey on data mining approaches for healthcare. *International Journal of Bio-Science and Bio-Technology*, *5*(5), 241–266.

Topchy, A., Jain, A. K., & Punch, W. (2004). A mixture model for clustering ensembles. In *Proceedings of the 2004 SIAM international conference on data mining* (pp. 379–390). SIAM.

Topchy, A. P., Law, M. H., Jain, A. K., & Fred, A. L. (2004). Analysis of consensus partition in cluster ensemble. In *Fourth IEEE international conference on data mining* (pp. 225–232). IEEE.

Wang, H., & Liu, G. (2018). Two-level-oriented selective clustering ensemble based on hybrid multi-modal metrics. *IEEE Access*, *6*, 64159–64168.

Wolpert, D. H., Macready, W. G., et al. (1995). *No free lunch theorems for search*: *Technical report*, Technical Report SFI-TR-95-02-010, Santa Fe Institute.

Xiao, W., Yang, Y., Wang, H., Li, T., & Xing, H. (2016). Semi-supervised hierarchical clustering ensemble and its application. *Neurocomputing*, *173*, 1362–1376. http://dx.doi.org/10.1016/j.neucom.2015.09.009, URL https://www.sciencedirect.com/science/article/pii/S092523121501317X.

Yan, D., Chen, A., & Jordan, M. I. (2013). Cluster forests. *Computational Statistics & Data Analysis*, *66*, 178–192. http://dx.doi.org/10.1016/j.csda.2013.04.010, URL https://www.sciencedirect.com/science/article/pii/S0167947313001400.

Yang, F., Li, T., Zhou, Q., & Xiao, H. (2017). Cluster ensemble selection with constraints. *Neurocomputing*, *235*, 59–70.

Yu, Z., Li, L., Wong, H.-S., You, J., Han, G., Gao, Y., et al. (2014). Probabilistic cluster structure ensemble. *Information Sciences*, *267*, 16–34. http://dx.doi.org/10.1016/j.ins.2014.01.030, URL https://www.sciencedirect.com/science/article/pii/S0020025514000504.

Yu, Z., Wong, H.-S., & Wang, H. (2007). Graph-based consensus clustering for class discovery from gene expression data. *Bioinformatics*, *23*(21), 2888–2896.