

ALGORITMO BRKGA APLICADO AO COMITÊ DE AGRUPAMENTOS

Augusto Pizano Vieira Beltrão

Instituto de Computação da Universidade Federal Fluminense
Niterói, 24210-310, RJ, Brasil.
apizano@id.uff.br

Luiz Satoru Ochi

Instituto de Computação da Universidade Federal Fluminense
Niterói, 24210-310, RJ, Brasil.
satoru@ic.uff.br

José André de Moura Brito

Escola Nacional de Ciências Estatísticas
Rio de Janeiro, 20231-050, RJ, Brasil.
jambrito@gmail.com

RESUMO

O Comitê de Agrupamentos (CA) tem, por fundamento, a combinação de diferentes partições (soluções de agrupamento) de uma base de dados em uma partição consenso. De acordo com a literatura, esta partição é menos sensível a ruídos e apresenta maior qualidade, quando comparada às partições base. Este trabalho traz a proposta de um algoritmo de CA, denominado GA-CE, que combina a metaheurística BRKGA como função consenso e o uso de três diferentes estratégias de geração das partições base. O GA-CE foi comparado com 6 relevantes algoritmos da literatura em 20 bases de dados, considerando dois índices de validação externa conhecidos na literatura do CA: $\mathcal{NMI}$ e $\mathcal{AR}$. A partir dos experimentos computacionais realizados, observou-se, em geral, que o algoritmo GA-CE produziu de partições consenso de boa qualidade, quando comparado aos principais algoritmos da literatura.

PALAVRAS CHAVE. Comitê de Agrupamentos, Análise de Cluster, Otimização, BRKGA.

ABSTRACT

The Cluster Ensemble (CE) is based on the combination of different partitions (clustering solutions) of a dataset into a consensus partition. According with literature, this partition is less sensitive to noise and presents higher quality, when compared to base partitions. This work proposes a CE algorithm, called GA-CE, which combines the BRKGA metaheuristic as a consensus function and the use of three different strategies for generating the base partitions. GA-CE was compared with 6 relevant algorithms from the literature in 20 datasets considering two external validation indices known in CE literature: $\mathcal{NMI}$ and $\mathcal{AR}$. From the experiments carried out, it was observed, in general, that the GA-CE produced good quality consensus partitions, when compared to the main algorithms in the literature.

KEYWORDS. Cluster Ensemble. Cluster Analysis. Optimization. BRKGA.

1. Introdução

Em sua concepção, algoritmos de agrupamento são aplicados para segmentar os objetos de uma base de dados de acordo com uma medida de distância (euclidiana, manhattan etc), calculada a partir de seus atributos quantitativos, qualitativos ou mistos [Jain, 2010], de modo que objetos de um mesmo grupo tenham alta similaridade entre si, enquanto objetos de grupos distintos tenham baixa similaridade [Boongoen e Iam-On, 2018].

Algoritmos de agrupamento apresentam diferentes limitações de acordo com a estratégia utilizada, muitas vezes com objetivos diferentes, e não apenas uma diferença na função objetivo, geralmente por serem dependentes de certos pressupostos acerca da base de dados [Boongoen e Iam-On, 2018]. Por exemplo, enquanto a estratégia do *k-means* é associar objetos mais próximos a cada centroide, o que resulta em grupos (hiper)esféricos de tamanhos semelhantes, o DBSCAN analisa objetos vizinhos de outro objeto com uma estratégia baseada em densidade, o que possibilita obter grupos de formatos variados. Assim sendo, o algoritmo não hierárquico *k-means* é mais adequado para determinar grupos de formato hiperesférico, enquanto o DBSCAN pode produzir grupos de diversos formatos, como alongados e espirais, e grupos aninhados. Pode-se perceber que, a partir do modo como um algoritmo constrói uma partição (grupos), determinados algoritmos podem ser mais adequados em detrimento a outros, considerando bases de dados específicas.

De forma a lidar com tais limitações/particularidades e produzir partições de boa qualidade, o Comitê de Agrupamentos (CA) constitui-se como uma boa alternativa, possibilitando melhorar a qualidade de partições produzidas por diferentes algoritmos de agrupamento e/ou diferentes configurações de seus parâmetros de entrada [Vega-Pons e Ruiz-Shulcloper, 2011]. Mais especificamente, a ideia central do CA diz respeito a combinar múltiplas partições, denominadas partições base, produzidas por diferentes algoritmos de agrupamento em uma única partição consenso, mediante a aplicação de uma função consenso. Esta partição resultante possui, geralmente, algumas propriedades interessantes como, por exemplo, ser menos sensível a ruídos nos dados quando comparada às partições base. O comitê de agrupamentos, por sua eficácia em produzir soluções de boa qualidade, tem encontrado aplicação em diversas áreas e temas, dentre as quais destacamos, transportes [Xiao et al., 2016], processamento de imagens [Liu et al., 2018] e cidades inteligentes [Kenger et al., 2023].

Tendo por objetivo produzir partições consenso de boa qualidade, o presente trabalho traz a proposta de um algoritmo de comitê que combina uma nova função consenso, baseada em índice de validação relativo, e a metaheurística BRKGA [Resende, 2011; Brito et al., 2022; Londe et al., 2024]. O algoritmo foi comparado com outras abordagens da literatura em cenário controlado, mais especificamente, utilizando as mesmas partições base produzidas a partir de 3 estratégias diferentes, mediante a aplicação de um conjunto de algoritmos de agrupamento hierárquicos e não hierárquicos, com o intuito de avaliar quais são as melhores partições no que concerne à maximização da qualidade¹ em termos de índices de validação de agrupamento para diversas bases de dados. Os resultados dos experimentos demonstraram a capacidade do algoritmo proposto de produzir soluções de boa qualidade (à luz de duas medidas de validação) para diferentes bases de dados, a partir de diversos comitês.

Além da introdução, este trabalho está dividido da seguinte forma: A Seção 2 traz a descrição do problema de comitê de agrupamentos e uma revisão da literatura com os principais trabalhos. A seção 3 apresenta o algoritmo proposto, doravante denotado por GA-CE e a Seção 4 traz os resultados de um conjunto de experimentos computacionais. Por fim, na Seção 5 são

¹A qualidade de uma partição pode ser medida comparando-a com as classes (rótulos) dos objetos em bases de dados de classificação quando disponíveis [Fred e Jain, 2005]. Estes rótulos também são referidos como *ground-truth partition* ou *true underlying partition* [Topchy et al., 2004].

apresentadas as conclusões e trabalhos futuros.

2. Comitê de Agrupamentos

O presente capítulo traz a descrição do Problema de Comitê de Agrupamento, assim como um panorama do problema. Adicionalmente, na revisão de literatura, são apresentados os trabalhos mais recentes e relevantes relacionados ao uso de Algoritmos Genéticos no âmbito do Comitê de Agrupamentos.

2.1. Definição do Problema

Considere $X = \{x_1, \dots, x_i, \dots, x_n\}$ uma base de dados composta por n objetos e $\Pi = \{\pi_1, \dots, \pi_g, \dots, \pi_m\}$ um comitê de agrupamentos composto por m partições base, sendo cada partição base $\pi_g = \{C_1, C_2, \dots, C_{k_g}\}$ correspondente a uma partição de X , formada por k_g grupos, onde o número de grupos de cada partição base pode ser diferente. Basicamente, para cada $x_i \in X$, $\psi^g(x_i)$ denota o rótulo do grupo da g -ésima partição base do qual x_i faz parte. Ao resolver o Problema de Comitê de Agrupamento (PCA), o objetivo é encontrar uma nova partição, denominada partição consenso, $\pi^* = \{C_1^*, \dots, C_k^*\}$ de X que resume a informação de Π [Boongoen e Iam-On, 2018]. Em linhas gerais, a resolução do PCA pode ser dividida em duas etapas, a saber: (i) produção das partições base (geração do comitê) e (ii) aplicação de uma função consenso nas partições base para produção da partição final. Essa função, por sua vez, é responsável pela combinação do conjunto de partições base (comitê Π) em uma única partição final π^* . Em linhas gerais, a resolução do PCA pode ser dividida em duas etapas, a saber: (i) produção das partições base (geração do comitê) e (ii) aplicação de uma função consenso nas partições base para produção da partição final. Essa função, por sua vez, é responsável pela combinação do conjunto de partições base (comitê Π) em uma única partição final π^* .

2.2. Revisão de literatura

No trabalho de Boongoen e Iam-On [2018] são apresentadas diversas propostas de função consenso distribuídas em quatro categorias, sejam elas: (i) abordagem direta [Topchy et al., 2004]; (ii) abordagem baseada em rótulos - matriz de rótulos (b) Filkov e Skiena [2004]; (iii) abordagem baseada em semelhança entre pares - matriz de coassociação (c) [Fred e Jain, 2005; Iam-On et al., 2008; Iam-On e Boongoen, 2013; Semaan et al., 2013] e (iv) abordagem baseada em grafos / hipergrafos [Yu et al., 2007; Semaan et al., 2014].

O trabalho de Huang et al. [2018a] faz duas propostas de função consenso, a saber: *locally weighted evidence accumulation* (LWEA) e *locally weighted graph partitioning* (LWGP). Ambas as propostas têm, por base, uma matriz de co-associação (CO) aprimorada, que atribui maior (ou menor) importância a cada grupo de acordo com a incerteza associada a ele. Depois de criar a matriz CO aprimorada, o LWEA utiliza o algoritmo hierárquico de ligação média para produzir a partição consenso, já o LWGP utiliza um algoritmo de particionamento de grafos. O trabalho de Huang et al. [2018b] considera não apenas a co-ocorrência de objetos (conexões diretas), mas a informação em níveis de granularidade maiores. Em Huang et al. [2019] apresenta-se um *framework* envolvendo dois estágios: U-SPEC (gera partições) e U-SENC (combina partições). O *framework* foca na escalabilidade, por meio de amostragem dos dados, para lidar com grandes bases de dados e é baseado em *Spectral Clustering*. O trabalho de Mahmoudi et al. [2021] utiliza uma matriz de similaridades (entre grupos) construída a partir de uma métrica nova para formar k meta grupos (*meta clusters*)². Esta matriz indica as similaridades entre os grupos das partições base e cada meta grupo representa um grupo da partição consenso. Então, cada objeto é alocado a um meta

²Um metagruppo é basicamente um conjunto de grupos das partições base, ou seja, um *cluster* de grupos. Todos os grupos de cada partição base são tratados como unidades a serem agregadas, mais especificamente, uma matriz de distâncias é construída a fim de “clusterizar” estes grupos.

grupo segundo a similaridade entre objetos e grupos. Em Huang et al. [2021] é utilizada uma ampla gama de métricas diversificadas, subespaços aleatórios e grupos (*clusters*) ponderados para resolver o CA para conjuntos de dados (*datasets*) de alta dimensão (muitas colunas). Em Zhou et al. [2021] propõe-se o algoritmo TRCE, que transforma o problema de CA em um problema de grafos (*multiple graph learning problem*) para construir uma partição de consenso considerando robustez em 3 níveis: partição base, grafo e robustez no âmbito da instância.

Os autores de Xu e Ding [2021] combinaram RTS (*Rough Set Theory*) com o CA para propor uma nova matriz de co-associação ponderada que é baseada na confiabilidade do *cluster* global e na similaridade entre pares de amostra local. Em Zhou et al. [2022] utiliza-se o SHL (*structured hypergraph learning*) para representar o comitê. Em vez de contruir um hipergrafo diretamente das partições base, o hipergrafo é dinamicamente construído a partir das partições base otimizando uma função objetivo. Em Jia et al. [2023] são utilizadas, simultaneamente, as informações da matriz HC (esparsa baseada em informações de alta confiança de pares de objetos) e da matriz de co-associação original para produzir uma matriz de co-associação aprimorada.

Por fim, destacam-se trabalhos relevantes envolvendo a utilização de Algoritmos Genéticos no CA propostos por Dai e Sheng [2019]; Canuto et al. [2020]; Zhu et al. [2020]; Yang et al. [2021]. Neste último trabalho os autores propuseram um algoritmo genético, denominado HGMCE, que utiliza uma função objetivo, baseada na similaridade dentro dos grupos e entre grupos, para combinar as partições base. Experimentos foram realizados em 10 bases de dados, com número de objetos variando entre 178 e 932, onde o HGMCE, quando comparado com 4 algoritmos da literatura, demonstrou resultados melhores nos experimentos, considerando o $\mathcal{NM}$ e a acurácia.

3. Algoritmo GA-CE

O GA-CE é aplicado em duas etapas conforme representado na Figura 1, a saber: Na primeira etapa são construídas m partições base $\{\pi_1, \pi_2, \dots, \pi_m\}$, mediante a aplicação de algoritmos de agrupamento que podem ser não-hierárquicos e/ou hierárquicos. Em seguida, a partir destas partições, uma matriz de co-associação D_{CO} (uma para cada estratégia) é calculada. Na segunda etapa, dividida em duas fases, aplica-se um algoritmo BRKGA, usado para selecionar, a partir de X , o melhor conjunto de k objetos representativos (medoides) [Kaufman e Rousseeuw, 1990] para os grupos. Neste caso, cada solução produzida pelo algoritmo corresponde à partição formada quando se aloca cada objeto de X ao grupo do objeto representativo mais próximo em função dos valores da matriz D_{CO} . Adicionalmente, a qualidade da solução é avaliada pela função objetivo associada ao índice de Silhueta Média calculado a partir da matriz de distâncias de coassociação, conforme descrito na Seção 3.2. Por fim, na fase 2, os melhores conjuntos de k objetos representativos (de acordo com a Silhueta Média) são submetidos ao procedimento iterativo (descrito na Seção 3.4). Neste procedimento, cada um dos $(n - k)$ objetos restantes da base é alocado iterativamente a um dos k objetos representativos (grupos), até que todos os objetos estejam associados a algum grupo. Depois de produzir algumas partições a partir de diversas aplicações do procedimento iterativo a solução escolhida como partição consenso é aquela que apresenta maior valor do índice de Silhueta Média.

3.1. Cromossomo e Decodificador: BRKGA

No BRKGA proposto cada cromossomo tem tamanho $\eta = k$ e cada entrada do cromossomo está associada a um grupo, sendo cada grupo representado por um objeto de X . Assim sendo, o cromossomo, que é um vetor de chaves aleatórias, pode ser mapeado em um conjunto de k objetos de X por meio do decodificador mediante a conversão de cada entrada do cromossomo (número real no intervalo $[0, 1)$) em um identificador correspondente a algum dos objetos de X (número inteiro pertencente ao conjunto $\{1, \dots, n\}$), conforme descrito a seguir.

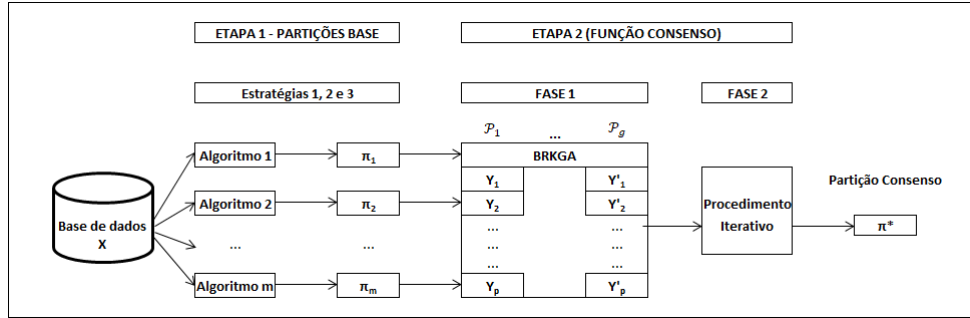


Figura 1: Funcionamento do GA-CE.

Criar conjunto de tamanho n contendo todos os objetos de X enumerados por um identificador $id = 1, \dots, n$. Este decodificador é aplicado de maneira iterativa selecionando-se (sem reposição) um objeto para representar um dos grupos em cada uma das k iterações. Inicialmente (iteração 1), multiplica-se a primeira entrada do cromossomo por n e aplica-se a função teto. Este número inteiro obtido corresponde ao identificador do primeiro objeto medoide. Assim, o conjunto de objetos a serem considerados na iteração 2 tem tamanho $(n - 1)$. O segundo objeto representativo é selecionado a partir da segunda entrada do cromossomo, multiplicando por $(n - 1)$ e aplicando a função teto. Este número corresponde ao identificador do segundo objeto representativo. Este processo é repetido até que sejam selecionados k objetos de X . Segue um exemplo de mapeamento de um cromossomo $Y = \{0.19, 0.87, 0.23\}$ de tamanho $k = 3$ para uma base de dados com $n = 10$ objetos.

- Primeiro objeto representativo: multiplica-se a primeira entrada de Y por n e aplica-se a função teto: $\lceil 0.19 \times 10 \rceil = 2$. Deste modo, deve-se escolher o 2º objeto de $x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}$. Assim, o objeto representativo do primeiro grupo é o objeto x_2 .
- Segundo objeto representativo: multiplica-se a segunda entrada de Y por $(n - 1)$ e aplica-se a função teto: $\lceil 0.87 \times 9 \rceil = 8$. Deste modo, deve-se escolher o 8º objeto de $x_1, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}$. Assim, o objeto representativo do segundo grupo é o objeto x_9 .
- Terceiro objeto representativo: multiplica-se a terceira entrada de Y por $(n - 2)$ e aplica-se a função teto: $\lceil 0.23 \times 8 \rceil = 2$. Deste modo, deve-se escolher o 2º objeto de $x_1, x_3, x_4, x_5, x_6, x_7, x_8, x_{10}$, o que implica que o objeto representativo do terceiro grupo é x_3 .

3.2. Função Objetivo: BRKGA

Inicialmente, cada um dos $(n - k)$ objetos restantes de X é alocado ao grupo do objeto representativo mais próximo (menor distância segundo a matriz D_{CO}). Deste modo, uma solução para o problema corresponde a uma partição dos n objetos de X em k grupos. A função objetivo (*fitness*) a ser maximizada corresponde ao índice de Silhueta Média desta partição, calculada a partir da matriz D_{CO} , tal que $D_{CO}(x_i, x_j) = 1 - CO(x_i, x_j)$ (matriz de coassociação CO). A dissimilaridade entre um par de objetos x_i e x_j corresponde ao percentual de vezes que eles não pertencem ao mesmo grupo nas m partições base. Deste modo, cada entrada de D_{CO} pertence ao intervalo $[0, 1]$. O cálculo desta matriz é feito depois da Etapa 1 (geração das partições base) e antes da Etapa 2 (construção da partição consenso).

3.3. Operadores Genéticos

Os operadores genéticos do GA-CE são os mesmos do BRKGA tradicional. Inicialmente, o BRKGA possui uma população composta de p cromossomos (vetores de chaves aleatórias) de

tamanho k e cada cromossomo da população é mapeado (pelo decodificador) em um conjunto de k medoides, sendo a função objetivo calculada para cada um deles. Assim, todos os cromossomos da população são ordenados (do melhor para o pior) e os p_e melhores são copiados para a população da próxima geração. Depois, são gerados p_m cromossomos mutantes para população da próxima geração. Por fim, diversos novos cromossomos filhos são gerados (para a população da próxima geração) a partir da combinação dos cromossomos da população atual mediante aplicação do cruzamento uniforme. A cada geração, os três operadores genéticos são aplicados para produzir a população da próxima geração e assim por diante. Na última geração, o melhor cromossomo da população é selecionado e mapeado no melhor conjunto de k objetos representativos para a segunda fase.

3.4. Procedimento iterativo

Este procedimento, associado à fase 2, é efetuado em 2 passos. No primeiro aloca-se cada objeto ao medoide mais próximo desde que a distância seja pequena entre eles, ou seja, trata-se de fazer as alocações mais confiáveis e triviais para que cada grupo tenha um conjunto de objetos além do medoide e no segundo passo alocam-se os demais objetos (que não foram alocados no primeiro passo) a um dos k grupos.

Inicialmente, cada um dos k grupos ($C_1, C_2, \dots, C_k$) é formado apenas por seu respectivo medoide, enquanto os $n - k$ objetos restantes de X não estão alocados a nenhum grupo. No primeiro passo, cada objeto deve ser alocado ao medoide mais próximo, desde que ele apresente uma distância menor ou igual a um determinado limite (*threshold*), que é dado pelo parâmetro de entrada σ do GA-CE, tal que $\sigma \in [0, 1]$. No segundo passo, todos os objetos que ainda não foram alocados são ordenados pela distância do medoide (primeiro os objetos mais próximos, depois os mais distantes). Para cada objeto x_i não alocado, deve-se calcular a média das distâncias entre x_i e cada um dos k grupos. O objeto x_i é alocado ao grupo que apresentar menor distância média (calcula a partir de todos os objetos já alocados ao grupo). Este procedimento se repete até que todos os objetos estejam alocados a um dos k grupos. Além disso, durante o cálculo da distância média entre x_i e o grupo C_j , são levados em consideração apenas os η objetos mais próximos de x_i , sendo tal número correspondente a um parâmetro de entrada do GA-CE.

Este procedimento iterativo é aplicado na última geração do BRKGA em todo conjunto elite e a solução escolhida como partição consenso é aquela que apresenta maior valor do índice de Silhueta Média.

4. Experimentos Computacionais

Nesta Seção são apresentados e analisados os resultados provenientes dos experimentos computacionais realizados com o GA-CE. Todos os experimentos foram realizados em um computador dotado de um processador Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz e com 8 GB RAM, no Sistema Operacional Windows 10.

O algoritmo GA-CE foi comparado com seis algoritmos, e algumas de suas variantes, relevantes na literatura de Comitê de Agrupamentos e frequentemente utilizados em diversos trabalhos como base de comparação de resultados, sejam eles: EAC [Fred e Jain, 2005], SRS [Iam-On et al., 2008], CTS [Iam-On et al., 2008], ASRS [Iam-On e Boongoen, 2013], WEAC [Huang et al., 2015] e LWEA [Huang et al., 2018a]. O algoritmo GA-CE e mais 4 algoritmos da literatura foram implementados em R³.

³Os algoritmos BRKGA-CE, GA-CE, EAC (EAC-SL, EAC-AL, EAC-CL), WEAC (WEAC-SL, WEAC-AL, WEAC-CL) e LWEA (LWEA-SL, LWEA-AL, LWEA-CL) foram implementados em R. Os algoritmos SRS (SRS-SL, SRS-AL, SRS-CL), CTS (CTS-SL, CTS-AL, CTS-CL), ASRS (ASRS-SL, ASRS-AL, ASRS-CL) estão implementados no pacote 'diceR' [Chiu e Talhouk, 2018].

Para os experimentos, foram consideradas 20 bases de dados de classificação de *benchmark*, com diferentes características em termos de número de objetos (observações) e número de atributos (dimensões). Algumas dessas bases correspondem a bases de dados do mundo real e outras são dados artificiais. A Tabela 1 apresenta as bases de dados, em que na primeira coluna está o nome de cada base, na segunda o número de objetos (n), na terceira o número de atributos (d) e na quarta o número de classes (ou seja, o número k de grupos). A quinta coluna indica a origem de cada base de dados: (i) bases de dados artificiais do conjunto *Shape sets*⁴ [Fränti e Sieranoja, 2018], e (ii) bases de dados do mundo real do *UCI Machine Learning Repository*⁵.

Tabela 1: Descrição das bases de dados de *benchmark*.

Número	Bases de Dados	n	d	Classes	Origem
D1	Iris	150	4	3	UCI MLR
D2	Wine	178	13	3	UCI MLR
D3	Sonar	208	60	2	UCI MLR
D4	Glass	214	9	6	UCI MLR
D5	New-thyroid	215	5	3	UCI MLR
D6	Seeds	218	7	3	UCI MLR
D7	Flame	240	2	2	dados artificiais
D8	Heart failure	299	12	2	UCI MLR
D9	Pathbased	300	2	3	dados artificiais
D10	Spiral	312	2	3	dados artificiais
D11	Ecoli	336	7	8	UCI MLR
D12	Ionosphere	351	33	2	UCI MLR
D13	Libras	360	90	15	UCI MLR
D14	Jain	373	2	2	dados artificiais
D15	Compound	399	2	6	dados artificiais
D16	WDBC	569	30	2	UCI MLR
D17	Synthetic control	600	60	6	UCI MLR
D18	R15	600	2	15	dados artificiais
D19	Aggregation	788	2	7	dados artificiais
D20	Yeast	1484	8	10	UCI MLR

Os parâmetros do GA-CE foram calibrados utilizando o *irace* (*Iterated racing for automatic algorithm configuration*) López-Ibáñez et al. [2016], cuja implementação está disponível no pacote 'irace' do R. As 6 bases de dados usadas na calibração foram Iris, Flame, Spiral, Ecoli, Jain, WDBC (3 artificiais e 3 do mundo real). Foram 6 parâmetros calibrados ao longo de 1000 iterações do *irace* considerando o $\mathcal{NMZ}$ para avaliação da qualidade das soluções [Fred e Jain, 2005; Huang et al., 2018a]. Os parâmetros são: $p \in \{50, 100, 150\}$ (tamanho da população), $p_e \in \{0.1p, 0.2p, 0.25p\}$ (número de cromossomos elite), $p_m \in \{0.1p, 0.2p, 0.3p\}$ (número de cromossomos mutantes), $\rho \in \{0.6, 0.7, 0.8\}$ (probabilidade no cruzamento), $\sigma \in \{0.3, 0.5, 0.7\}$ (*threshold*, valor considerado para o limite máximo de distância) e $\eta \in \{10, 20, 30\}$ (vizinhos mais próximos). Realizada a calibração, o melhor conjunto de valores encontrado para os parâmetros foi: $p = 100$, $p_e = 0.25$, $p_m = 0.1$, $\rho = 0.8$, $\sigma = 0.7$, $\eta = 20$. O número de gerações foi fixado em 200, uma vez que, em experimentos preliminares, nenhuma melhoria ou contribuição significativa (ganhos na função objetivo) foi observada com número de gerações superior a 200.

As partições base dos comitês foram geradas adotando-se três diferentes estratégias, que contemplam a aplicação de alguns algoritmos clássicos de agrupamento, conforme exposto a seguir:

- Estratégia 1: algoritmos não hierárquicos: (i) baseados no *k-means*, considerando três versões: Hartigan-Wong, Lloyd, MacQueen. Implementadas no pacote base do R. (ii) baseados no *k-medoids*: PAM, CLARA e CLARANS. Implementados no pacote 'fastkmedoids' do R.

⁴<http://cs.uef.fi/sipu/datasets/>

⁵<https://archive.ics.uci.edu/ml/index.php>

- Estratégia 2: métodos hierárquicos aglomerativos: ligação simples, média, completa, Ward, mediana, centroide e *Mcquitty*. Método hierárquico divisivo: DIANA. Os algoritmos hierárquicos aglomerativos estão implementados no pacote base (*hclust*), enquanto o algoritmo DIANA no pacote 'cluster' do R.
- Estratégia 3: considera todos os algoritmos hierárquicos e não hierárquicos das estratégias 1 e 2.

Para garantir diversidade nas partições base, o valor de k foi aleatoriamente selecionado de $K = \{2, 3, \dots, \lfloor \sqrt{n} \rfloor\}$ em cada execução de cada algoritmo; conjunto também adotado em Huang et al. [2018a,b]. As partições base utilizadas são iguais para todos os algoritmos, para garantir uma comparação equânime. Os testes foram feitos considerando 30 execuções repetidas de cada algoritmo em cada base de dados considerando cada uma das 3 estratégias (ou seja, um total de $30 \times 20 \times 3 = 1800$ execuções por algoritmo) e todas as execuções foram feitas com tamanho do comitê $m = 10$ (cada comitê é gerado por uma das estratégias). Muitos trabalhos da literatura utilizam valores entre 10 e 50 para o tamanho comitê [Fred e Jain, 2005; Iam-On et al., 2008; Iam-On e Boongoen, 2013; Huang et al., 2018a,b]. Neste trabalho, assim como em Huang et al. [2018a], optou-se por comitês de tamanho 10. Foram utilizados dois índices de validação externa para calcular a qualidade das partições consenso: $\mathcal{AR}$ (*Adjusted Rand index*) e $\mathcal{NMI}$ (*Normalized Mutual Information*), também considerados em diversos trabalhos de Comitê de Agrupamentos [Huang et al., 2018a].

Foram aplicados testes de hipótese (considerando nível de significância de 5%) para validação estatística dos resultados, conforme a Tabela 2. O teste de *Friedman* foi utilizado para ordenar os algoritmos conforme a qualidade de seus respectivos resultados e verificar se existe diferença significativa entre eles. Este teste foi aplicado 6 vezes (2 medidas $\times$ 3 estratégias) para avaliar se existe diferença significativa entre os ranks médios⁶. Depois, o teste de *Wilcoxon* (*Wilcoxon signed ranks test*) foi aplicado para verificar se existe diferença significativa comparando os algoritmos dois a dois, considerando a correção de Bonferroni. Analisando-se Tabela 2, pode-se perceber que todos os resultados são significativos, o que sugere que a diferença entre os algoritmos comparados é estatisticamente significativa em todos os cenários (2 métricas $\times$ 3 estratégias). Pode-se perceber que o GA-CE, em geral, ocupa as primeiras posições, considerando diversos cenários (as oito melhores soluções por cada algoritmo e índice estão assinaladas em cinza).

O teste de *Friedman* indica se existe diferença significativa entre os ranks médios dos algoritmos comparados, no entanto, não apresenta quais algoritmos foram responsáveis pela diferença significativa. Deste modo, de forma a consolidar a análise, o teste dos sinais de *Wilcoxon* foi aplicado para verificar se existia diferença significativa entre os resultados do GA-CE e o algoritmo de melhor rank médio em cada cenário, segundo a Tabela 2. Segundo os testes realizados⁷, não existe diferença significativa entre o GA-CE e o melhor rank médio em nenhum caso. Este resultado corrobora com a evidência de que o GA-CE performa bem independente do cenário adotado.

A Figura 2 ilustra a comparação entre o GA-CE e os algoritmos da literatura, conforme os resultados da Tabela 2 (algoritmos ordenados de acordo com os valores do $\mathcal{AR}$ na estratégia 2).

⁶Um rank de um algoritmo corresponde ao valor que indica a posição daquele algoritmo quando ordenado frente aos demais, segundo alguma medida de qualidade. O rank médio do algoritmo corresponde à média das posições que ele ocupa considerando todas as bases de dados utilizadas nos experimentos. Quanto maior o rank, melhor.

⁷Seguem resultados (p-valores ajustados no R pela função *p.adjust* considerando a correção de Bonferroni) do teste de *Wilcoxon* comparando GA-CE com o melhor resultado (Tabela 2) em todos os 6 cenários.

Estratégia 1: (i) $\mathcal{AR}$: p-valor 0.34, (ii) $\mathcal{NMI}$: p-valor 0.07. Estratégias 2 e 3 : (i) $\mathcal{AR}$: p-valor 1.00, (ii) $\mathcal{NMI}$: p-valor 1.00.

Tabela 2: Resultados do teste de *Friedman* com nível de significância de 5% e 18 graus de liberdade. Apresenta os ranks médios de cada algoritmo (considerando as 20 bases de dados).

	Estratégia 1		Estratégia 2		Estratégia 3	
	$\mathcal{AR}$	$\mathcal{NMZ}$	$\mathcal{AR}$	$\mathcal{NMZ}$	$\mathcal{AR}$	$\mathcal{NMZ}$
χ^2_f	71.557	52.071	72.255	50.181	99.265	69.401
p-valor	5.06E-08	6.46E-05	3.86E-08	0.0001233	7.27E-13	1.16E-07
EAC-SL	7.80	8.75	8.00	8.30	6.00	7.60
EAC-AL	12.15	11.55	9.85	9.00	11.40	11.80
EAC-CL	7.00	8.25	8.80	8.85	7.60	7.65
CTS-SL	8.40	9.10	6.32	6.72	6.57	7.22
CTS-AL	12.40	12.55	7.25	7.50	9.55	9.80
CTS-CL	9.85	9.15	10.70	9.90	10.80	9.60
SRS-SL	10.40	10.95	8.62	9.97	8.17	9.28
SRS-AL	13.10	12.75	11.20	11.00	13.00	13.10
SRS-CL	10.40	9.45	12.20	11.25	12.35	11.20
ASRS-SL	8.25	7.65	7.27	7.12	6.00	6.12
ASRS-AL	12.30	11.45	10.30	9.70	11.70	10.60
ASRS-CL	11.45	10.05	10.40	10.10	11.55	10.60
WEAC-SL	9.80	10.05	8.40	9.70	7.97	9.52
WEAC-AL	12.60	12.85	12.05	11.60	13.63	13.60
WEAC-CL	7.50	8.70	9.78	10.35	10.15	10.20
LWEA-SL	5.05	4.90	8.65	9.72	5.20	5.65
LWEA-AL	13.75	13.25	14.00	13.15	14.45	13.45
LWEA-CL	6.95	8.40	11.80	12.35	10.35	10.10
GA-CE	10.85	10.20	14.40	13.70	13.55	12.90

O GA-CE também foi comparado com o HGCEA e a diversos outros algoritmos da literatura utilizados no artigo de Yang et al. [2021]. As partições base usadas nesse caso foram geradas pelo k – means, k – medoids, fuzzy c – means (FCM), expectation maximization with Gaussian mixture (EMGM) e affinity propagation (AP) com um comitê de tamanho 50 de acordo com o trabalho original. O GA-CE foi executado 30 vezes e a média do $\mathcal{NMZ}$ é apresentada para comparação na Tabela 3⁸. O GA-CE apresentou os melhores resultados em 2 dos 3 datasets comparados. No dataset “Pima”, o GA-CE perdeu apenas para o HGCEA. A Tabela 3 traz a comparação entre as partições base e as partições consenso, segundo o $\mathcal{NMZ}$. Pode-se perceber que, em 2 dos 3 casos, as partições consenso apresentaram melhor qualidade que as partições base. O caso em que as partições base apresentaram qualidade maior que as consenso são para o dataset “Pima” que, de modo geral, todas as soluções têm baixa qualidade com média abaixo de 10% para o $\mathcal{NMZ}$.

Tabela 3: Resultados dos experimentos com o GA-CE comparando com o artigo Yang et al. [2021]. A medida considerada é o $\mathcal{NMZ}$.

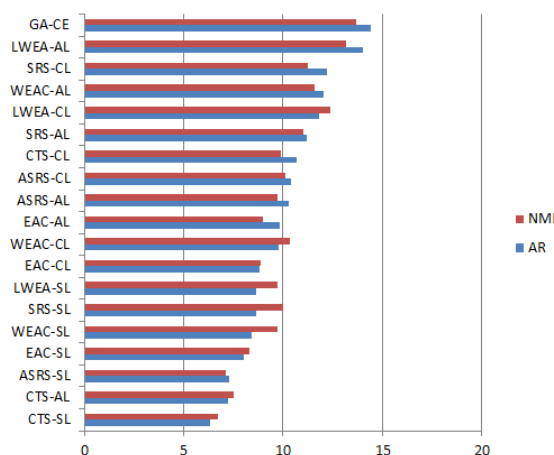
Dataset	Partições base	GA-CE	HGCEA	MDEC-HC	MDEC-SC	CGE	USPEC
Wine (n = 178, d = 13)	0.65	0.87	0.83	0.77	0.82	0.40	0.82
Pima (n = 768, d = 8)	0.09	0.07	0.74	0.00	0.01	0.00	0.00
Brea (n = 699, d = 9)	0.47	0.76	0.46	0.42	0.46	0.26	0.02

5. Conclusão e trabalhos futuros

Um novo algoritmo (GA-CE) foi proposto para o PCA, desenvolvido a partir do estudo da metaheurística BRKGA. Tal algoritmo tem por objetivo maximizar o índice de Silhueta Média, considerando a matriz de coassociação e representando a solução por meio de objetos representativos de cada grupo. Além disso, o GA-CE é baseado em fases que contemplam a geração de partições base (considerando 3 estratégias diferentes), a definição dos objetos representativos dos grupos e, por fim, a construção da partição consenso. Foram conduzidos diversos experimentos, com três diferentes estratégias de geração do comitê, contemplando diversos algoritmos de agrupamento hierárquicos e não hierárquicos e dois índices de validação externa, comumente utilizados na

⁸ Alguns datasets que não apresentaram dados faltantes foram selecionados para comparação

Figura 2: Gráfico contendo resultados da estratégia 2 na Tabela 2. Ordenados pelos valores de $\mathcal{AR}$.



literatura: $\mathcal{AR}$ e $\mathcal{NMI}$. Os resultados dos experimentos demonstraram a capacidade do GA-CE de produzir soluções de boa qualidade (à luz das medidas de validação) para diferentes bases de dados, a partir de diversos comitês. Complementarmente, diversas análises estatísticas foram feitas para avaliar/validar os resultados, ratificando a qualidade da abordagem proposta frente às principais abordagens da literatura. Estas análises correspondem ao teste de hipótese de *Friedman*, uma comparação dos valores de $\mathcal{AR}$ da estratégia 2, alguns testes de hipótese não paramétricos, indicando diferença não significativa entre os resultados do GA-CE e os melhores resultados do experimento. Os melhores resultados comparativos foram obtidos quando foi utilizada a estratégia 2, indicando, consequentemente, que a estratégia de geração do comitê tem impacto nas soluções finais associadas à partição consenso. Outro experimento realizado compara o GA-CE com os resultados do artigo Yang et al. [2021]. Neste experimento o GA-CE apresentou os melhores resultados em 2 das 3 bases de dados comparadas. O GA-CE perdeu apenas em 1 das bases de dados para 1 algoritmo.

Como trabalhos futuros, pretende-se testar outros índices de validação, além do Silhueta Média e matrizes de consenso já propostas na literatura [Iam-On et al., 2008; Huang et al., 2015, 2018a]. Outra possibilidade é testar novas estratégias de geração de comitê, visto que a escolha da estratégia pode afetar a partição consenso.

Referências

- Boongoen, T. e Iam-On, N. (2018). Cluster ensembles: A survey of approaches with recent extensions and applications. ISSN 15740137.
- Brito, J. A., Semaan, G., e Fadel, A. (2022). The effective brkga algorithm for the k-medoids clustering problem. *RAIRO-Operations Research*, 56(4):3137–3153.
- Canuto, A., Neto, A. F., Silva, H. M., Xavier-Júnior, J. C., e Barreto, C. A. (2020). Population-based bio-inspired algorithms for cluster ensembles optimization. *Natural Computing*. ISSN 15729796.
- Chiu, D. S. e Talhouk, A. (2018). dicer: an r package for class discovery using an ensemble driven approach. *BMC bioinformatics*, 19(1):1–4.

- Dai, H. e Sheng, W. (2019). A multi-objective clustering ensemble algorithm with automatic k-determination. In *2019 IEEE 4th International Conference on Cloud Computing and Big Data Analytics, ICCCBDA 2019*. ISBN 9781728114095.
- Filkov, V. e Skiena, S. (2004). Integrating microarray data by consensus clustering. *International Journal on Artificial Intelligence Tools*, 13(04):863–880.
- Fränti, P. e Sieranoja, S. (2018). K-means properties on six clustering benchmark datasets. *Applied Intelligence*, 48(12):4743–4759.
- Fred, A. L. e Jain, A. K. (2005). Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. ISSN 01628828.
- Huang, D., Lai, J. H., e Wang, C. D. (2015). Combining multiple clusterings via crowd agreement estimation and multi-granularity link analysis. *Neurocomputing*. ISSN 18728286.
- Huang, D., Wang, C. D., e Lai, J. H. (2018a). Locally Weighted Ensemble Clustering. *IEEE Transactions on Cybernetics*, 48(5):1460–1473. ISSN 21682267.
- Huang, D., Wang, C.-D., Lai, J.-H., e Kwoh, C.-K. (2021). Toward multidiversified ensemble clustering of high-dimensional data: From subspaces to metrics and beyond. *IEEE Transactions on Cybernetics*, 52(11):12231–12244.
- Huang, D., Wang, C.-D., Peng, H., Lai, J., e Kwoh, C.-K. (2018b). Enhanced ensemble clustering via fast propagation of cluster-wise similarities. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.
- Huang, D., Wang, C.-D., Wu, J.-S., Lai, J.-H., e Kwoh, C.-K. (2019). Ultra-scalable spectral clustering and ensemble clustering. *IEEE Transactions on Knowledge and Data Engineering*, 32(6):1212–1226.
- Iam-On, N. e Boongoen, T. (2013). Pairwise similarity for cluster ensemble problem: link-based and approximate approaches. In *Transactions on Large-Scale Data-and Knowledge-Centered Systems IX*, p. 95–122. Springer.
- Iam-On, N., Boongoen, T., e Garrett, S. (2008). Refining pairwise similarity matrix for cluster ensemble problem with cluster relations. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. ISBN 3540884106.
- Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666.
- Jia, Y., Tao, S., Wang, R., e Wang, Y. (2023). Ensemble clustering via co-association matrix self-enhancement. *IEEE Transactions on Neural Networks and Learning Systems*.
- Kaufman, L. e Rousseeuw, P. J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley. ISBN 978-0-47031680-1.
- Kenger, Ö. N., Kenger, Z. D., Özceylan, E., e Mrugalska, B. (2023). Clustering of cities based on their smart performances: A comparative approach of fuzzy c-means, k-means, and k-medoids. *IEEE Access*, 11:134446–134459.

- Liu, H., Zhang, Q., e Zhao, F. (2018). Interval fuzzy spectral clustering ensemble algorithm for color image segmentation. *J. Intell. Fuzzy Systems*, 35(5):5467–5476.
- Londe, M. A., Pessoa, L. S., Andrade, C. E., e Resende, M. G. (2024). Biased random-key genetic algorithms: A review. *European Journal of Operational Research*.
- López-Ibáñez, M., Dubois-Lacoste, J., Cáceres, L. P., Birattari, M., e Stützle, T. (2016). The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3:43–58.
- Mahmoudi, M. R., Akbarzadeh, H., Parvin, H., Nejatian, S., Rezaie, V., e Alinejad-Rokny, H. (2021). Consensus function based on cluster-wise two level clustering. *Artificial Intelligence Review*, 54(1):639–665.
- Resende, M. (2011). Introdução aos algoritmos genéticos de chaves aleatórias viciadas. *Simpósio Brasileiro de Pesquisa Operacional*, p. 3680–3691.
- Semaan, G. S., Rodrigues, W., de Moura Brito, J. A., e Ochi, L. S. (2014). Método baseado em combinação de soluções com particionamento de grafos para o problema de agrupamento automático. *Learning & Nonlinear Models*.
- Semaan, G. S., Torres, C. d. C., Brito, J. A. d. M., e Ochi, L. S. (2013). Um método baseado em combinação de soluções com coassociação para o problema de agrupamento automático. *Revista brasileira de estatística*, 74(239):43–68.
- Topchy, A. P., Law, M. H., Jain, A. K., e Fred, A. L. (2004). Analysis of consensus partition in cluster ensemble. In *Fourth IEEE International Conference on Data Mining (ICDM'04)*, p. 225–232. IEEE.
- Vega-Pons, S. e Ruiz-Shulcloper, J. (2011). A survey of clustering ensemble algorithms. *International Journal of Pattern Recognition and Artificial Intelligence*. ISSN 02180014.
- Xiao, W., Yang, Y., Wang, H., Li, T., e Xing, H. (2016). Semisupervised hierarchical clustering ensemble and its application. *Neurocomputing*, 173:1362–1376.
- Xu, L. e Ding, S. (2021). Dual-granularity weighted ensemble clustering. *Knowledge-Based Systems*, 225:107124.
- Yang, W., Zhang, Y., Wang, H., Deng, P., e Li, T. (2021). Hybrid genetic model for clustering ensemble. *Knowledge-Based Systems*, 231:107457.
- Yu, Z., Wong, H. S., e Wang, H. (2007). Graph-based consensus clustering for class discovery from gene expression data. *Bioinformatics*. ISSN 13674803.
- Zhou, P., Du, L., Shen, Y.-D., e Li, X. (2021). Tri-level robust clustering ensemble with multiple graph learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, p. 11125–11133.
- Zhou, P., Wang, X., Du, L., e Li, X. (2022). Clustering ensemble via structured hypergraph learning. *Information Fusion*, 78:171–179.
- Zhu, S., Xu, L., e Goodman, E. D. (2020). Evolutionary multi-objective automatic clustering enhanced with quality metrics and ensemble strategy. *Knowledge-Based Systems*. ISSN 09507051.