Uma Abordagem de Agrupamento Automático de Dados Baseada na Otimização por Busca em Grupo Memética

Luciano D. S. Pacífico[1], **Teresa B. Ludermir**[2]

[1]Departamento de Computação – Universiade Federal Rural de Pernambuco
R. Dom Manuel de Medeiros, S/N – 52.171-900 – Recife – PE – Brazil

[2]Centro de Informática – Universidade Federal de Pernambuco
Av. Jornalista Aninal Fernandes, – 50740-560 – Recife – PE – Brazil

luciano.pacifico@ufrpe.br, tbl@cin.ufpe.br

Abstract. *As one of the most primitive pattern organization tasks, clustering is a hard grouping problem in exploratory data analysis. Most standard clustering algorithms are easily trapped into local minima points from the problem search space, once such models lack good global search capabilities. In this work, a memetic Swarm Intelligence (SIs) algorithm is presented, based on Group Search Optimization and K-Means, called MGSO, that attempts both finding the best number of final clusters and the best distribution among patterns in clusters, simultaneously. The proposed MGSO showed to be able to find good global solutions through a testing bed of nine real-world data sets, in comparison to other SIs and Evolutionary Algorithms from the literature.*

Resumo. *Uma das tarefas mais primitivas em organização de padrões, a Análise de Agrupamentos, é um problema difícil em análise exploratória de dados. Muitos dos algoritmos de agrupamento são facilmente presos em mínimos locais, por não possuírem bons operadores de busca global. Neste trabalho, um algoritmo de Inteligência de Enxames (SIs) memético é apresentado, baseado na Otimização por Busca em Grupo e no K-Means, chamado MGSO, que tenta encontrar o melhor número de agrupamentos, assim como a melhor distribuição dos dados nesses agrupamentos, simultaneamente. O MGSO mostrou-se capaz de encontrar boas soluções globais quando testado em nove problemas reais, em comparação a outros SIs e Algoritmos Evolucionários da literatura.*

1. Introdução

A enorme quantidade de dados gerada diariamente requer sistemas computacionais cada vez mais precisos e robustos, que consigam realizar a descoberta de padrões e tendências existentes em enormes quantidades de dados rapidamente, gerando informações úteis nas mais diversas áreas de aplicações. Nesse contexto, a Análise de Agrupamentos se destaca como uma das áreas mais fundamentais em reconhecimento de padrões, consistindo em um importante mecanismo para a análise exploratória de dados.

Os algoritmos de agrupamento buscam separar um conjunto de observações (base de dados) em grupos (ou agrupamentos), de modo que observações pertencentes a um mesmo grupo sejam mais semelhantes entre si, de acordo com seus conjuntos de características, do que observações pertencentes a grupos diferentes. O problema de agrupamento de dados é considerado um problema NP-Difícil, sendo seu estudo de grande

interesse para a área de otimização. Dentre as abordagens de agrupamento de dados tradicionais, os algoritmos particionais possuem grande destaque, sendo o K-Means [MacQueen et al. 1967] uma das abordagens de agrupamento mais populares, em decorrência de sua fácil implementação e de sua rápida taxa de convergência. Porém, os algoritmos particionais tradicionais possuem várias limitações, como a inexistência de mecanismos de otimização global, e sua sensibilidade à inicialização.

Nesse contexto, Algoritmos Evolucionários (EAs) e de Inteligência de Enxames (SIs) têm sido cada vez mais empregados na tarefa de agrupamento de dados, em decorrência de suas capacidades de busca por ótimos globais, evitando pontos ótimos locais do espaço de busca do problema. Nos EAs e SIs, um conjunto de soluções (população) é mantido e melhorado ao longo de um processo geracional (iterativo), no qual operadores evolucionários são executados a essas soluções em uma tentativa de melhorar o valor de uma função objetivo (função de *fitness*). Em EAs, como os Algoritmos Genéticos [Holland 1992], Evolução Diferencial [Storn and Price 1995] e a Otimização por Busca com *Backtracking* [Civicioglu 2013], os mecanismos de busca simulam processos biológicos, tais como a mutação, recombinação e seleção. Já nos SIs, como a Otimização por Enxame de Partículas [Kennedy and Eberhart 1995] e a Otimização por Busca em Grupo [He et al. 2009], os operadores buscam simular comporamentos de animais sociais, como a busca por recursos, movimentos em bando, etc.

Os EAs e SIs têm sido adotados com sucesso ao contexto de algoritmos particionais nas últimas décadas, apresentando resultados promissores. Porém, tais algoritmos, em decorrência de suas naturezas estocásticas, tendem a ser mais lentos para convergir que algoritmos particionais tradicionais. Uma solução comumente encontrada na literatura é a combinação entre EAs ou SIs com modelos particionais tradicionais, resultando em modelos híbridos, que tentam combinar as boas capacidades de otimização global oferecidas pelos modelos evolucionários, com as rápidas taxas de convergência dos algoritmos paticionais [Latiff et al. 2016, Pacifico and Ludermir 2019, Pacifico and Ludermir 2020, Pacifico and Ludermir 2021].

Um outro problema relacionado aos algoritmos particionais tradicionais diz respeito à necessidade de fornecimento *a priori* do número final de agrupamentos a serem formados, o que requer um entendimento inicial do contexto do problema a ser resolvido, o que pode limitar a aplicação desses modelos a problemas pouco conhecidos ou novos. Como uma tentativa de eliminar essa limitação, abordagens de Análise Automática de Agrupamentos têm sido propostas, que tentam realizar tanto a otimização dos agrupamentos finais gerados, quanto à estimação do número final de grupos a serem formados [Das et al. 2007, José-García and Gómez-Flores 2016, Elaziz et al. 2019, Pacifico and Ludermir 2020].

Neste trabalho, um algoritmo memético é apresentado, que combina a capacidade de otimização global da Otimização por Busca em Grupo (GSO) com a velocidade de convergência do K-Means: o MGSO. O MGSO é implementado como um algoritmo particional de Análise Automática de Agrupamentos, buscando a otimização simultânea do número final de grupos estimados, assim como a melhor distribuição dos padrões nesses grupos. O GSO é escolhido por ter apresentado boas performances quando comparado a outros algoritmos de Computação Evolucionária, como o PSO e o GA [He et al. 2009].

O trabalho está dividido como segue. A próxima seção (Seção 2) apresenta os modelos que servem como base para a abordagem proposta (ou seja, o K-Means e o GSO), seguida pela apresentação do MGSO (Seção 3). A avaliação experimental é apresentada na Seção 4, seguida pelas conclusões e tendências para pesquisas futuras (Seção 5).

2. Preliminares

As próximas seções apresentarão uma breve descrição do algoritmo K-Means (Seção 2.1) e do GSO (Seção 2.2), respectivamente.

2.1. K-Means

O K-Means é um algoritmo de agrupamento particional para dados contínuos, que agrupa vetores de dados reais em um número pré-definido de agrupamentos (parâmetro de entreda do algoritmo). Considere uma partição P_C de um conjunto de dados com N_O padrões (cada padrão é representado por um vetor $\mathbf{o}_j \in \Re^m$, onde $j = 1, 2, ..., N_O$) em C agrupamentos. Cada agrupamento é representado pelo seu vetor centroide $\mathbf{g}_c \in \Re^m$ (onde $c = 1, 2, ..., C$).

No K-Means, os agrupamentos são formados pelo uso de uma medida de dissimilaridade, a distância Euclidiana (eq.(1)). A cada iteração do algoritmo (até que um número máximo de t_{max} iterações ou algum outro critério de término seja atingido), um novo vetor centroide é calculado, para cada agrupamento, como o vetor médio dos padrões atualmente associados ao agrupamento (eq. (2)). Após a determinação dos novos vetores centroides, os padrões da base de dados são redistribuídos nos agrupamentos, de acordo com o critério de proximidade ao vetor centroide.

$$d(\mathbf{o}_j,\mathbf{g}_c)=\sqrt{\sum_{k=1}^{m}(o_{jk}-g_{ck})^2} \tag{1}$$

$$\mathbf{g}_c=\frac{1}{n_c}\sum_{\forall j\in c}\mathbf{o}_j \tag{2}$$

onde n_c é o número de padrões atualmente associados ao agrupamento c.

O algoritmo K-Means é apresentado no Algoritmo 1.

Algorithm 1 K-Means

$t \leftarrow 0$.
Inicialização: Obtenha aleatoriamente C padrões da base de dados como os vetores centroides iniciais $\mathbf{g}_c$. Depois disso, aloque cada padrão $\mathbf{o}_j$ ao agrupamento mais próximo do mesmo.
enquanto $(t < t_{max})$ **faça**
  Determinação dos novos centroides: para cada agrupamento *c*, atualize seu centroide $\mathbf{g}_c^t$ usando eq.(2).
  Determinação da nova partição: para cada padrão $\mathbf{o}_j$, determine seu novo agrupamento de acordo com a proximidade ao centroide $\mathbf{g}_c^t$.
  $t \leftarrow t + 1$.
fim_enquanto
Retorne a partição final $P_C^{t_{max}}$.

2.2. Otimização por Busca em Grupo (GSO)

O GSO é um SI baseado em uma tentativa de simulação do comportamento de busca de animais sociais, e na teoria dos grupos vivos (*living group theory*). O GSO segue o modelo *Producer-Scrounger* (PS), proposto inicialmente por Barnard e Sibly [Barnard and Sibly 1981], como um *framework* para a análise comportamental das estratégias de busca por recursos empregadas por animais que vivem em grupos. No

GSO, a população G de S indivíduos é chamada de *grupo*, enquanto os indivíduos da população são chamados de *membros*. A busca desempenhada por cada membro no GSO leva em consideração o campo de varredura visual do mesmo. Em um problema definido por um espaço de busca n-dimensional, o i-ésimo membro na t-ésima iteração da busca terá uma posição atual definida pelo vetor $\mathbf{X}_i^t \in \Re^n$ e um ângulo de cabeça definido pelo vetor $\alpha_i^t \in \Re^{n-1}$. A direção da busca do i-ésimo membro, que é um vetor $\mathbf{D}_i^t(\alpha_i^t) = (d_{i1}^t, \ldots, d_{in}^t)$, pode ser calculada por α_i^t através de uma transformação polar para coordenadas cartesianas dada por:

$$\begin{aligned} d_{i1}^t &= \textstyle\prod_{q=1}^{n-1} \cos(\alpha_{iq}^t), \\ d_{ij}^t &= \sin(\alpha_{i(j-1)}^t) \textstyle\prod_{q=1}^{n-1} \cos(\alpha_{iq}^t)(j{=}1,\ldots,n{-}1), \\ d_{in}^t &= \sin(\alpha_{i(n-1)}^t) \end{aligned} \tag{3}$$

Um grupo no GSO consiste de três tipos de membros: *producers*, *scrougers* e *rangers*, sendo os *rangers* uma modificação ao modelo PS original proposta pelo GSO. Durante cada geração do GSO, o membro do grupo que encontrou a melhor reserva de recursos (melhor valor de *fitness*), é escolhido como *producer* [Couzin et al. 2005]. O *producer* executará uma estratégia de varredura baseada em seu campo de visão. Essa estratégia recebe o nome de *producing*. Na t-ésima geração de uma execução do GSO, o *producer* $\mathbf{X}_p^t$ varrerá o espaço de busca do problema através da observação de três pontos aleatórios em seu campo visual: um ponto a zero grau ($\mathbf{X}_z$), um ponto no hipercubo à sua direita ($\mathbf{X}_r$), e um ponto no hipercubo à sua esquerda ($\mathbf{X}_l$), de acordo com eq. (4).

$$\mathbf{X}_z{=}\mathbf{X}_p^t{+}r_1 l_{max}\mathbf{D}_p^t(\alpha_p^t),\ \ \mathbf{X}_r{=}\mathbf{X}_p^t{+}r_1 l_{max}\mathbf{D}_p^t(\alpha_p^t{+}\tfrac{\mathbf{r}_2\theta_{max}}{2}),\ \ \mathbf{X}_l{=}\mathbf{X}_p^t{+}r_1 l_{max}\mathbf{D}_p^t(\alpha_p^t{-}\tfrac{\mathbf{r}_2\theta_{max}}{2}) \tag{4}$$

onde $r_1 \in \Re$ é um número aleatório obtido através de uma distribuição normal com média 0 (zero) e desvio padrão 1 (um), $\mathbf{r}_2 \in \Re^{n-1}$ é uma sequência uniforme obtida aleatoriamente no intervalo (0, 1), $\theta_{max} \in \Re^{n-1}$ é o ângulo máximo de busca e $l_{max} \in \Re$ é a distância máxima de busca, definida pela equação eq. (5):

$$l_{max}{=}\|\mathbf{U}{-}\mathbf{L}\|{=}\sqrt{\textstyle\sum_{k=1}^{n}(U_k{-}L_k)^2} \tag{5}$$

onde U_k e L_k denotam o limite superior e o limite inferior da k-ésima dimensão do problema, respectivamente.

Se o *producer* for capaz de encontrar uma posição melhor que a atual, ele se dirigirá para esse ponto; caso contrário, o *producer* permanecerá em sua posição atual, movimentando sua cabeça para uma posição definida por um novo ângulo, de acordo com (eq. (6)):

$$\alpha_p^{t+1}{=}\alpha_p^t{+}\mathbf{r}_2\beta_{max} \tag{6}$$

onde $\beta_{max} \in \Re$ é o ângulo máximo de retorno. Se após $a \in \Re$ gerações o *producer* não for capaz de encontrar uma posição melhor que sua posição atual, ele retornará sua cabeça para a posição de ângulo zero (eq.(7)).

$$\alpha_p^{k+a}{=}\alpha_p^k \tag{7}$$

Todos os *scroungers* tentarão alcançar as reservas já encontradas pelo *producer*, sendo essa estratégia denominada *scrounging*. O operador de *scrounging* no GSO é dado por (eq. (8)):

$$\mathbf{X}_i^{t+1}=\mathbf{X}_i^t+\mathbf{r}_3\circ(\mathbf{X}_p^t-\mathbf{X}_i^t) \tag{8}$$

onde $\mathbf{r}_3 \in \Re^n$ é uma sequência uniforme aleatória obtida no intervalo (0, 1), e $\circ$ é o produto de Hadamard ou Schur, que calcula o produto interno entre dois vetores.

Os *rangers* irão executar buscas aleatórias no espaço do problema, sendo essa estratégia denominada *ranging* [Higgins and Strauss 2004]. O operador de *ranging* é apresentado abaixo (eq. (9)).

$$\mathbf{X}_i^{t+1}=\mathbf{X}_i^t+l_i\mathbf{D}_i^t(\alpha_i^{t+1}),\ \ l_i=ar_1l_{max} \tag{9}$$

O GSO define uma estratégia de contenção aos membros que, em decorrência da atualização de suas posições pelos operadores evolucionários da estratégia, acabam sendo posicionados fora dos limites do espaço de busca do problema. No GSO tradicional, quando um membro escapa dos limites do espaço de busca após a atualização de sua posição, o mesmo será reconduzido à sua posição anterior dentro dos limites do espaço de busca do problema [Dixon 1959]. A importância do tratamento de indivíduos da população que extrapolam os limites do espaço de busca do problema está relacionada ao fato de que soluções inválidas (que não se enquadrariam às restrições do problema tratado) poderiam ser geradas e consideradas como soluções possíveis para o problema. O GSO é apresentado no Algoritmo 2.

Algorithm 2 GSO

$t \leftarrow 0$.
Inicialize aleatoriamente as posições dos membros $\mathbf{X}_i{}^{(0)} \in G$. Determine aleatoriamente os ângulos de cabeça $\alpha_i^{(0)}$ de cada membro $\mathbf{X}_i^{(0)} \in G^{(0)}$
Calcule $f(\mathbf{X}_i^{(0)})$ para cada membro $\mathbf{X}_i^{(0)}$.
enquanto (*as condições de término não forem satisfeitas*) **faça**
  Use o melhor membro do grupo ($\mathbf{X}_p^t$) para a execução do *producing* (eq. (4)).
  Escolha um percentual dos membros restantes para a execução da estratégia de *scrounging*.
  Ranging: Os membros restantes executarão *ranging* por buscas aleatórias no espaço do problema.
  Aplique o operador de controle do GSO para os membros que escapam das dimensões do espaço de busca do problema.
  Calcule o novo *fitness* $f(\mathbf{X}_i^{t+1})$ para cada membro $\mathbf{X}_i^{t+1}$.
  $t \leftarrow t+1$.
fim_enquanto
retorne $\mathbf{X}_p^{t_{max}}$.

3. Abordagem Proposta: MGSO

Nesta seção, o algoritmo de agrupamento automático memético proposto, o MGSO, será descrito. No MGSO, a busca global executada pelo GSO é complementada pelas rápidas buscas locais oferecidas pelo K-Means, em uma tentativa de prever tanto o melhor número de agrupamentos finais, quanto a separação dos dados em análise da melhor forma possível nesses agrupamentos.

Considere uma partição P_C da base de dados, que contém N_o padrões $\mathbf{o}_j \in \Re^m$ ($j = 1, 2, ..., N_o$), em no máximo C_{max} agrupamentos. Cada agrupamento será representado por seu centroide $\mathbf{g}_c \in \Re^m$ ($c = 1, 2, ..., C_{max}$). Cada membro $\mathbf{X}_i \in \Re^n$ (onde $n = C_{max}+C_{max}\times m$) no grupo G representa C_{max} valores de ativação e C_{max} centroides

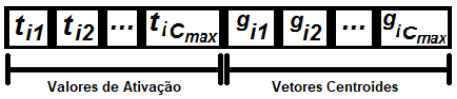

Figura 1. Representação dos membros: as primeiros C_{max} características representam os valores de ativação para cada agrupamento candidato, enquanto os $C_{max} \times m$ valores seguintes representam os C_{max} m-dimensionais centroides dos agrupamentos candidatos.

de agrupamentos ao mesmo tempo [Das et al. 2007, Elaziz et al. 2019], como ilustrado na Fig. 1.

Na t-ésima geração, o membro $\mathbf{X}_i^t$ terá seu valor de *fitness* avaliado considerando apenas os agrupamentos **ativados**, ou seja, os agrupamentos cujos valores de ativação sejam tais que $t_{ic}^t \geq 0.5$. A função de *fitness* escolhida deve ser apropriada para a realização da tarefa de Análise de Agrupamentos Automática, de modo a otimizar, simultaneamente, o número de agrupamentos finais, e também os vetores centroides [José-García and Gómez-Flores 2016].

A inicialização é realizada através da escolha aleatória de C_{max} padrões da base de dados em análise para compor os centroides iniciais de cada um dos membros $\mathbf{X}_i^{(0)}$, e, de modo semelhante, os valores de ativação $t_{ic}^{(0)}$ (onde $c = 1, 2, \ldots, C_{max}$) são obtidos aleatoriamente a partir de uma distribuição uniforme no intervalo $U(0, 1)$.

Após a inicialização e a avaliação de cada membro em $G^{(0)}$, o processo geracional do MGSO tem início de modo semelhante ao GSO. Na t-ésima geração, o membro mais promissor $\mathbf{X}_p^t$ é escolhido para a execução do operador de *producing* (eq. (4)). Em seguida, um percentual dos $\mathbf{X}_i^t$ membros restantes são escolhidos para executarem a estratégia de *scrounging* (eq. (8)). Por fim, os membros restantes serão escolhidos como *rangers*, executando o *ranging* (eq. (9)).

No MGSO, em cada geração t, o processo memético ocorrerá após a execução dos operadores do GSO tradicional. Esse operador será implementado pelo refinamento de cada membro $\mathbf{X}_i^{t+1}$ do novo grupo G^{t+1} pela execução de uma etapa completa do K-Means.

Se a qualquer momento da busca um membro $\mathbf{X}_i$ representar menos de dois agrupamentos ativos, o mesmo será reinicializado aleatoriamente, através de um processo semelhante ao adotado na etapa de inicialização do grupo.

O MGSO é representado no Algoritmo 3.

4. Análise Experimental

Nesta seção, o algoritmo memético proposto (o MGSO) é testado e comparado a cinco outros EAs e SIs da literatura, através da avaliação de nove bases de dados reais, obtidas através do *UCI Machine Learning Repository* [Asuncion and Newman 2007]. As bases de dados selecionadas, assim como suas características, são apresentadas na Tabela 1. Tais bases apresentam diferentes níveis de dificuldade, como classes desbalanceadas, sobreposição entre classes, dentre outros.

Os EAs e SIs selecionados para a análise comparativa são: O Algoritmo Genético (GA), a Evolução Diferencial (DE), a Otimização por Enxame de Partículas (PSO), a

Algorithm 3 MGSO

$t \leftarrow 0$.
Inicialização: para cada membro $\mathbf{X}_i{}^{(0)} \in G^{(0)}$, escolha aleatoriamente C_{max} padrões da base de dados como os centroides de agrupamento iniciais $\mathbf{g}_{ic}(c = 1, 2, \ldots, C_{max})$. Determine aleatoriamente os valores de ativação $t_{ic}^{(0)}$ e os ângulos de cabeça $\alpha_i^{(0)}$ de cada membro $\mathbf{X}_i^{(0)} \in G^{(0)}$. Para cada membro $\mathbf{X}_i{}^{(0)}$, encontre o agrupamento correspondente à cada padrão $\mathbf{o}_j$ pelo critério de menor distância.
Calcule o valor do *fitness* ($f(\mathbf{X}_i^{(0)})$) para cada membro $\mathbf{X}_i^{(0)}$.
enquanto (*as condições de término não forem satisfeitas*) **faça**
  Use o melhor membro do grupo ($\mathbf{X}_p^t$) para a execução do *producing* (eq. (4)). Para cada ponto gerado, determine a partição formada pelo mesmo pela atribuição de cada padrão da base de dados ao agrupamento mais próximo.
  Escolha um percentual dos membros restantes para a execução da estratégia de *scrounging*.
  Ranging: Os membros restantes executarão *ranging* por buscas aleatórias no espaço do problema.
  Aplique o operador de controle do GSO para os membros que escapam das dimensões do espaço de busca do problema.
  Execute uma única etapa do K-Means para refinar cada membro do novo grupo G^{t+1}.
  Reinicialize aleatoriamente cada membro no novo grupo G^{t+1} que contenha menos que dois agrupamentos ativos.
  Calcule o valor do *fitness* ($f(\mathbf{X}_i^{t+1})$) de cada membro no novo grupo G^{t+1}.
  $t \leftarrow t + 1$.
fim_enquanto
Retorne $\mathbf{X}_p^{t_{max}}$.

Tabela 1. Bases de Dados Reais.

Base de Dados	N. de Padrões	N. de Características	N. de Classes
Banknote Authentication	1372	4	2
Cancer	699	9	2
Diabetes	768	8	2
Heart	270	13	2
Ionosphere	351	34	2
Iris	150	4	3
Page Blocks Classification	5473	10	5
Seeds	210	7	3
Waveform	5000	21	3

Otimização por Busca com *Backtracking* (BSA) e o GSO tradicional. Tais algoritmos representam algumas das melhores abordagens do estado da arte em Computação Evolucionária e Inteligência de Enxames, tendo sido aplicados na solução de vários problemas práticos [Preetha 2021, Shi et al. 2020, Ye and Zheng 2021, Jin and Yin 2020, Pacífico 2020]. Neste trabalho, todos os EAs e SIs adotados para comparações foram adaptados ao contexto de Análise de Agrupamentos Automática, de modo semelhante ao processo apresentado para o MGSO (vide Seção 3). Os hiper-parâmetros para cada um dos algoritmos testados são apresentados na Tabela 2, tendo sido obtidos da literatura.

Quatro métricas de comparação da área de Análise de Agrupamentos são empregadas: o Índice de Calinski-Harabasz (CH) [Caliński and Harabasz 1974], o Índice de Rand Corrigido (CR) [Hubert and Arabie 1985], o Índice de Davies-Bouldin (DB) [Davies and Bouldin 1979], e o Índice de Jaccard [Halkidi et al. 2002]. As métricas de avaliação representam três méticas de maximização (CH, CR e JI, situação representada por $^{\uparrow}$), e uma métrica de minimização (DB, situação representada por $^{\downarrow}$). O CH foi adotado como função de *fitness* para todos os algoritmos avaliados.

A análise experimental inclui uma avaliação empírica baseada no valor médio obtido para cada uma das métricas adotadas em relação a trinta execuções independentes dos experimentos para cada base de dados, assim como um sistema de *ranks* elaborado através da aplicação de testes de hipóteses do tipo *Teste de Friedman* [Friedman 1937] aos resultados. O teste de Friedman é um teste de hipóteses não-paramétrico que calcula valores de *ranks* para os algoritmos para cada base de dados separadamente. Se a hipótese nula de que os *ranks* não são significativamente diferentes for rejeitada, o teste

Tabela 2. Hiper-parâmetros para os EAs e SIs.

Algoritmo	Parâmetro	Valor
Todos os EAs e SIs	t_{max}	200
	S	100
	C_{max}	20
GA	Taxa de Recombinação	0.8
	Taxa de Mutação	0.1
	Taxa de Seleção	0.8
DE	F	0.8
	Taxa de Recombinação	0.9
PSO	c_1	2.0
	c_2	2.0
	w	0.9 até 0.4
BSA	$mixrate$	1
	F	$3N(0,1)$
GSO e MGSO	Taxa de *Scroungers*	0.8
	θ_{max}	π/a^2
	α_0	$\pi/4$
	β_{max}	θ_{max}/2

de Nemenyi [Nemenyi 1962] é adotado com um teste *post hoc* para o teste de Friedman. De acordo com o teste de Nemenyi, a performance de dois algoritmos é considerada significativamente diferente se a diferença entre seus valores médios de *rank* for ao menos maior que uma *diferença crítica* dada por:

$$CD = q_a \sqrt{\frac{n_{alg}(n_{alg}+1)}{6n_{bases}}} \tag{10}$$

onde n_{bases} representa o número de bases de dados (ou seja, de imagens), n_{alg} representa o número de algoritmos comparados e q_a são valores críticos baseados em estatísticas nos limites do modelo t de *Student* divididas por $\sqrt{2}$ [Demšar 2006]. Os melhores *ranks* apontados pelo teste de Friedman-Nemenyi apresentarão valores altos, para as métricas de maximização (CH, CR e JI), e valores baixos para métricas de minimização (DB).

Os resultados experimentais obtidos são apresentados na Tabela 3. A análise empírica em relação aos valores médios obtidos para cada métrica que a solução proposta de abordagem memética entre o GSO e o K-Means resultou em um modelo mais estável que o GSO tradicional e os demais algoritmos de comparação. Também podemos observar que o MGSO foi capaz de encontrar o melhor valor para a função de *fitness* na maioria dos cenários avaliados (oito entre nove bases de dados, tendo sido o segundo melhor modelo para o caso da base de dados *Page Blocks Classification*), o que demonstra sua robustez. Os algoritmos estudados foram capazes de encontrar o número real de agrupamentos para seis dos nove casos avaliados (exceto para o algoritmo PSO), o que pode ser considerado um bom desempenho, compatível com outros trabalhos da literatura [Das et al. 2007, Tam et al. 2017], e mesmo para os três casos nos quais não houve um casamento perfeito entre o valor esperado e o obtido, os valores estimados pelos modelos foram bastante próximos aos valores reais.

A avaliação geral obtida pelo sistema de *ranks* proporcionados pelos testes de Friedman-Nemenyi (Tabela 4) indica que o MGSO proposto foi capaz de obter os melhores resultados globais, levando-se em conta cada uma das métricas de avaliação, tendo

Tabela 3. Resultados experimentais para as bases de dados reais (Média ± Desvio Padrão).

Base	Algoritmo	$CH^{\uparrow}$	$CR^{\uparrow}$	$DB^{\downarrow}$	$JI^{\uparrow}$	C
Banknote Authentication	GA	1423.4 ± 0.202	0.0487 ± 0.0015	0.8709 ± 0.0012	0.3803 ± 0.0008	**2 ± 0**
	DE	1423.6 ± 0.153	0.0485 ± 0.0006	0.8704 ± 0.0009	0.3804 ± 0.0006	**2 ± 0**
	PSO	1387.4 ± 107.6	**0.0647 ± 0.0420**	0.8863 ± 0.0378	0.3573 ± 0.0522	2.7667 ± 2.063
	BSA	1423.5 ± 0.2572	0.0491 ± 0.0010	0.8708 ± 0.0008	**0.3805 ± 0.0007**	**2 ± 0**
	GSO	**1423.7 ± 0.0486**	0.0487 ± 0.0003	**0.8702 ± 0.0004**	**0.3805 ± 0.0003**	**2 ± 0**
	MGSO	**1423.7 ± 0**	0.0486 ± 0	**0.8702 ± 0**	**0.3805± 0**	**2 ± 0**
Cancer	GA	1038.9 ± 1.979	0.8320 ± 0.0090	0.7618 ± 0.0006	0.8599 ± 0.0067	**2 ± 0**
	DE	1038.9 ± 2.572	0.8344 ± 0.0084	0.7618 ± 0.0006	0.8618 ± 0.0062	**2 ± 0**
	PSO	1029.3 ± 65.95	0.8372 ± 0.0121	0.7873 ± 0.1429	0.8633 ± 0.0116	2.0333 ± 0.1826
	BSA	1040.1 ± 1.165	0.8351 ± 0.0083	0.7615 ± 0.0003	0.8623 ± 0.0062	**2 ± 0**
	GSO	**1041.4 ± 0.0918**	0.8385 ± 0.0029	**0.7612 ± 0.0001**	0.8647 ± 0.0022	**2 ± 0**
	MGSO	**1041.4 ± 0**	**0.8391 ± 0**	**0.7612 ± 0**	**0.8651 ± 0**	**2 ± 0**
Diabetes	GA	1139.1 ± 2.102	0.0443 ± 0.0036	0.6646 ± 0.0042	0.3789 ± 0.0041	**3 ± 0**
	DE	1140.0 ± 2.251	0.0450 ± 0.0025	0.6651 ± 0.0032	0.3793 ± 0.0026	**3 ± 0**
	PSO	996.17 ± 187.6	**0.0501 ± 0.0164**	0.8084 ± 0.2226	0.3277 ± 0.0969	4.6667 ± 2.928
	BSA	1136.5 ± 3.586	0.0453 ± 0.0046	**0.6638 ± 0.0037**	**0.3806 ± 0.0050**	**3 ± 0**
	GSO	1141.8 ± 2.930	0.0451 ± 0.0010	0.6673 ± 0.0044	0.3783 ± 0.0017	**3 ± 0**
	MGSO	**1142.6 ± 0**	0.0452 ± 0	0.6681 ± 0	0.3781 ± 0	**3 ± 0**
Heart	GA	**206.95 ± 0.0036**	0.0295 ± 0.0012	0.9875 ± 0.0006	0.3606 ± 0.0009	**2 ± 0**
	DE	**206.95 ± 0**	**0.0302 ± 0**	**0.9871 ± 0**	**0.3611 ± 0**	**2 ± 0**
	PSO	206.84 ± 0.0995	0.0250 ± 0.0037	**0.9871 ± 0.0014**	0.3591 ± 0.0012	**2 ± 0**
	BSA	**206.95 ± 0**	**0.0302 ± 0**	**0.9871 ± 0**	**0.3611 ± 0**	**2 ± 0**
	GSO	**206.95 ± 0.0041**	0.0301 ± 0.0005	0.9873 ± 0.0007	0.3610 ± 0.0006	**2 ± 0**
	MGSO	**206.95 ± 0**	**0.0302 ± 0**	**0.9871 ± 0**	**0.3611 ± 0**	**2 ± 0**
Ionosphere	GA	115.65 ± 1.198	0.1464 ± 0.0132	1.5341 ± 0.0111	0.4190 ± 0.0064	**2 ± 0**
	DE	115.48 ± 1.601	0.1427 ± 0.0158	1.5367 ± 0.0143	0.4175 ± 0.0074	**2 ± 0**
	PSO	116.13 ± 9.484	**0.1791 ± 0.0214**	1.5375 ± 0.0895	0.4317 ± **0.0084**	2.0667 ± 0.258
	BSA	117.27 ± 0.9134	0.1564 ± 0.0151	1.5206 ± 0.0094	0.4233 ± 0.0075	**2 ± 0**
	GSO	118.43 ± 0.3889	0.1697 ± 0.0091	1.5158 ± 0.0052	0.4298 ± 0.0043	**2 ± 0**
	MGSO	**118.83 ± 0**	0.1776 ± 0	**1.5134 ± 0**	**0.4336 ± 0**	**2 ± 0**
Iris	GA	561.58 ± 0.256	0.7302 ± 0.0001	0.6622 ± 0.0013	0.6958 ± 0.0003	**3 ± 0**
	DE	**561.63 ± 0**	0.7302 ± 0	**0.6620 ± 0**	**0.6959 ± 0**	**3 ± 0**
	PSO	560.80 ± 2.540	0.7301 ± 0.0004	0.6636 ± 0.0047	0.6956 ± 0.0007	**3 ± 0**
	BSA	**561.63 ± 0**	0.7302 ± 0	**0.6620 ± 0**	**0.6959 ± 0**	**3 ± 0**
	GSO	561.37 ± 0.8113	**0.7316 ± 0.0040**	0.6627 ± 0.0023	0.6971 ± 0.0037	**3 ± 0**
	MGSO	**561.63 ± 0**	0.7302 ± 0	**0.6620 ± 0**	**0.6959 ± 0**	**3 ± 0**
Page Blocks Classification	GA	14395.2 ± 567.3	0.0070 ± 0.0154	**0.5250 ± 0.0342**	0.6044 ± 0.0893	5.5 ± 0.509
	DE	**16343.2 ± 778.1**	0.0003 ± 0.0129	0.6159 ± 0.0311	0.5195 ± 0.0745	7.5 ± 0.861
	PSO	13372.5 ± 1920.9	0.0109 ± 0.0059	0.5307 ± 0.0318	0.6634 ± 0.0300	**4.7000 ± 0.8769**
	BSA	15007.2 ± 529.4	0.0057 ± 0.0156	0.5626 ± 0.0523	0.6031 ± 0.0735	5.9667 ± 0.8087
	GSO	12456.9 ± 1436.9	**0.0110 ± 0.0108**	0.5364 ± 0.0267	0.6667 ± 0.0232	4.3667 ± 0.5561
	MGSO	14985.8 ± 2446.4	-0.0016 ± 0.0010	0.5794 ± 0.0319	0.5930 ± 0.0600	6 ± 1.4622
Seeds	GA	375.31 ± 0.7548	**0.7178 ± 0.0086**	0.7535 ± 0.0010	**0.6827 ± 0.0081**	**3 ± 0**
	DE	372.38 ± 2.3840	0.7106 ± 0.0209	0.7564 ± 0.0041	0.6763 ± 0.0194	**3 ± 0**
	PSO	375.73 ± 0.2892	0.7159 ± 0.0028	0.7535 ± 0.0007	0.6808 ± 0.0026	**3 ± 0**
	BSA	370.66 ± 5.5973	0.6988 ± 0.0274	0.7603 ± 0.0081	0.6656 ± 0.0243	**3 ± 0**
	GSO	375.68 ± 0.3881	0.7153 ± 0.0040	0.7535 ± 0.0007	0.6803 ± 0.0037	**3 ± 0**
	MGSO	**375.81 ± 0**	0.7166 ± 0	**0.7533 ± 0**	0.6815 ± 0	**3 ± 0**
Waveform	GA	2518.7 ± 11.88	0.3473 ± 0.0112	1.3783 ± 0.0036	0.4374 ± 0.0067	2 ± 0
	DE	2544.2 ± 8.190	0.3597 ± 0.0057	1.3734 ± 0.0021	0.4450 ± 0.0035	2 ± 0
	PSO	2552.6 ± 58.11	0.3669 ± 0.0213	1.3705 ± 0.0047	0.4480 ± 0.0209	**2.0333 ± 0.1826**
	BSA	2536.2 ± 7.4291	0.3537 ± 0.0093	1.3745 ± 0.0027	0.4413 ± 0.0056	2 ± 0
	GSO	2546.3 ± 10.52	0.3608 ± 0.0065	1.3733 ± 0.0027	0.4456 ± 0.0040	2 ± 0
	MGSO	**2563.3 ± 0**	**0.3713 ± 0**	**1.3696 ± 0**	**0.4521 ± 0**	2 ± 0

Tabela 4. Avaliação Global: *Ranks* médios para o Teste de Friedman-Nemenyi, com $CD = 2.5132$.

Algoritmo	$CH^{\uparrow}$	$RI^{\uparrow}$	$DB^{\downarrow}$	$JI^{\uparrow}$
GA	63.2796	74.6204	102.5111	77.9130
DE	84.0389	79.4926	107.3889	78.0444
PSO	94.8167	99.5981	**74.6333**	94.4741
BSA	67.1463	83.7889	104.4574	86.4574
GSO	100.6370	101.1222	80.4278	99.7852
MGSO	**133.0815**	**104.3778**	**73.5815**	**106.3259**

seus resultados sido considerados superiores, com diferenças estatísticas significativas em relação aos demais modelos (exceto para o Índice de Davies-Bouldin, no qual seus resultados foram considerados estatisticamente equivalentes aos do PSO), inclusive quando comparado ao GSO original. Os resultados experimentais indicam que a combinação do GSO com o K-Means foi capaz de proporcionar uma maior estabilidade ao modelo, assim como melhorar seu desempenho na realização da tarefa de agrupamento automático de dados.

5. Conclusões

Neste trabalho, um algoritmo de agrupamento automático particional memético é apresentado, o MGSO, que combina a capacidade de realização de buscas globais do GSO com a velocidade de convergência do K-Means, para a otimização simultânea do número final de agrupamentos e da partição da base de dados formada. O MGSO é proposto como uma forma de eliminar a dependência existente nos algoritmos de agrupamento particionais tradicionais do fornecimento *a priori* do número de agrupamentos finais desejado.

O algoritmo proposto é comparado a cinco algoritmos do estado da arte em Computação Evolucionária e Inteligência de Enxames, adaptados ao contexto da Análise de Agrupamentos Automática: GA, DE, PSO, BSA e GSO. A análise experimental levou em consideração quatro métricas de agrupamentos de dados, assim como um teste de hipóteses estatístico, sendo realizada pela aplicação dos modelos selecionados a nove bases de dados *benchmark* reais.

Os resultados experimentais revelaram a robustez e estabilidade do MGSO, que obteve os melhores valores globais para as métricas de agrupamento estudadas em todos os casos, de acordo com uma avaliação estatística, tendo sido capaz de superar o desempenho do GSO tradicional. Como trabalhos futuros, pretendemos estender nossa avaliação pela introdução de novos problemas à análise, tanto pela aquisição de bases de dados obtidas em cenários controlados (bases de dados sintéticas), quanto pela aplicação do MGSO a problemas reais práticos. O MGSO será também combinado a outros EAs e SIs, em tentativas de melhorias de seu desempenho através da proposta de abordagens híbridas.

Referências

Asuncion, A. and Newman, D. (2007). Uci machine learning repository.

Barnard, C. and Sibly, R. (1981). Producers and scroungers: a general model and its application to captive flocks of house sparrows. *Animal Behaviour*, 29(2):543–550.

Caliński, T. and Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27.

Civicioglu, P. (2013). Backtracking search optimization algorithm for numerical optimization problems. *Applied Mathematics and computation*, 219(15):8121–8144.

Couzin, I. D., Krause, J., Franks, N. R., and Levin, S. A. (2005). Effective leadership and decision-making in animal groups on the move. *Nature*, 433(7025):513–516.

Das, S., Abraham, A., and Konar, A. (2007). Automatic clustering using an improved differential evolution algorithm. *IEEE Transactions on systems, man, and cybernetics-Part A: Systems and Humans*, 38(1):218–237.

Davies, D. L. and Bouldin, D. W. (1979). A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30.

Dixon, A. (1959). An experimental study of the searching behaviour of the predatory coccinellid beetle adalia decempunctata (l.). *The Journal of Animal Ecology*, pages 259–281.

Elaziz, M. A., Nabil, N., Ewees, A. A., and Lu, S. (2019). Automatic data clustering based on hybrid atom search optimization and sine-cosine algorithm. In *2019 IEEE Congress on Evolutionary Computation (CEC)*, pages 2315–2322. IEEE.

Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association*, 32(200):675–701.

Halkidi, M., Batistakis, Y., and Vazirgiannis, M. (2002). Cluster validity methods: part i. *ACM Sigmod Record*, 31(2):40–45.

He, S., Wu, Q. H., and Saunders, J. R. (2009). Group search optimizer: an optimization algorithm inspired by animal searching behavior. *IEEE Transactions on Evolutionary Computation*, 13(5):973–990.

Higgins, C. L. and Strauss, R. E. (2004). Discrimination and classification of foraging paths produced by search-tactic models. *Behavioral Ecology*, 15(2):248–254.

Holland, J. H. (1992). Genetic algorithms. *Scientific american*, 267(1):66–72.

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

Jin, Y.-F. and Yin, Z.-Y. (2020). Enhancement of backtracking search algorithm for identifying soil parameters. *International Journal for Numerical and Analytical Methods in Geomechanics*, 44(9):1239–1261.

José-García, A. and Gómez-Flores, W. (2016). Automatic clustering using nature-inspired metaheuristics: A survey. *Applied Soft Computing*, 41:192–213.

Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948. IEEE.

Latiff, N. A., Malik, N. N. A., and Idoumghar, L. (2016). Hybrid backtracking search optimization algorithm and k-means for clustering in wireless sensor networks. In *2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*, pages 558–564. 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*, volume 1, pages 281–297. California, USA.

Nemenyi, P. (1962). Distribution-free multiple comparisons. In *Biometrics*, volume 18, page 263. INTERNATIONAL BIOMETRIC SOC 1441 I ST, NW, SUITE 700, WASHINGTON, DC 20005-2210.

Pacífico, L. (2020). Agrupamento de imagens baseado em uma abordagem híbrida entre a otimização por busca em grupo e k-means para a segmentação automática de doenças em plantas. In *Anais do XVII Encontro Nacional de Inteligência Artificial e Computacional*, pages 152–163. SBC.

Pacifico, L. and Ludermir, T. (2020). Backtracking group search optimization: A hybrid approach for automatic data clustering. In *Brazilian Conference on Intelligent Systems*, pages 64–78. Springer.

Pacifico, L. D. and Ludermir, T. B. (2019). Hybrid k-means and improved self-adaptive particle swarm optimization for data clustering. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE.

Pacifico, L. D. and Ludermir, T. B. (2021). An evaluation of k-means as a local search operator in hybrid memetic group search optimization for data clustering. *Natural Computing*, 20(3):611–636.

Preetha, V. (2021). Data analysis on student's performance based on health status using genetic algorithm and clustering algorithms. In *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, pages 836–842. IEEE.

Shi, X., Zhang, X., and Xu, M. (2020). A self-adaptive preferred learning differential evolution algorithm for task scheduling in cloud computing. In *2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)*, pages 145–148. IEEE.

Storn, R. and Price, K. (1995). Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces. international computer science institute, berkeley. Technical report, CA, 1995, Tech. Rep. TR-95–012.

Tam, H.-H., Ng, S.-C., Lui, A. K., and Leung, M.-F. (2017). Improved activation schema on automatic clustering using differential evolution algorithm. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pages 1749–1756. IEEE.

Ye, L. and Zheng, D. (2021). Stable grasping control of robot based on particle swarm optimization. In *2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*, pages 1020–1024. IEEE.