

Image Clustering Based on a Hybrid Group Search Optimization and K-Means Approach for Automatic Plant Disease Segmentation

Luciano D. S. Pacifico¹

¹Departamento de Computação (DC) – Universidade Federal Rural de Pernambuco (UFRPE) – Recife – PE – Brazil

luciano.pacifico@ufrpe.br

Abstract. *In this work, a hybrid partitional clustering algorithm between K-Means and Group Search Optimization is employed to deal with automatic disease segmentation in leaf images, as part of an automatic plant disease classification system: the GSO-KM. An unsupervised image clustering approach is adopted to deal with the problem. GSO-KM is compared to five algorithms from data clustering literature through twelve images, which present variant degrees of disease, using four evaluation metrics. Experimental analysis showed that GSO-KM is able to solve the image segmentation problem properly for the evaluated scenario.*

Resumo. *Neste trabalho, um algoritmo de agrupamento particional híbrido entre o K-Means e a Otimização por Busca em Grupo é usado para a tarefa de segmentação automática de doenças em imagens de folhas, como parte de um sistema automático de classificação de doenças em plantas: o GSO-KM. Uma abordagem não supervisionada de agrupamento de imagens é adotada no intuito de lidar com o problema. O GSO-KM é comparado a cinco algoritmos da literatura de agrupamento de dados através do uso de doze imagens, que apresentam diferentes graus de doença, e pelo uso de quatro métricas de avaliação. A análise experimental revelou que o GSO-KM é capaz de realizar, de forma satisfatória, a tarefa de segmentação das doenças nas imagens avaliadas.*

1. Introdução

Uma das áreas estratégicas da economia de países emergentes, como o Brasil, é a do agronegócio, que mesmo em épocas difíceis, como no cenário de pandemia atual, continua a apresentar índices crescentes de produção. O agronegócio tem sido bastante impulsionado pelo rápido desenvolvimento das Tecnologias de Informação e Comunicação (TICs) nas últimas décadas, sendo cada vez mais empregadas tecnologias de ponta para a automação das atividades desenvolvidas nesse setor econômico.

Um dos principais problemas que ocorrem em lavouras é o surgimento de doenças e pragas, sendo o rápido diagnóstico desses problemas de alto interesse econômico para os produtores [Barbedo et al. 2018]. Doenças em plantas geralmente manifestam seus primeiros sintomas observáveis através das folhas, e muitos estudos focam na análise das folhas para a identificação dessas patologias [Barbedo et al. 2018, Sawarkar and Kawathekar 2018]. Pelo uso de técnicas avançadas de Aprendizagem de Máquina e Visão Computacional é possível não só a identificação dessas doenças,

mas também a análise do grau do dano causado pelas mesmas [Tanmayee 2017, Solanke et al. 2018]. Uma das etapas fundamentais para a identificação e análise do dano causado por patologias em plantas é a segmentação das imagens das folhas, porém, muitas das propostas existentes na literatura realizam essa tarefa manualmente [Lee et al. 2015, Barbedo et al. 2018, Pacifico et al. 2019], o que requer um alto grau de esforço humano.

Neste trabalho, um método automático de segmentação de imagens de plantas é apresentado, baseado em uma abordagem de aprendizagem não supervisionada híbrida entre o algoritmo de agrupamento particional K-Means [MacQueen et al. 1967] e um Algoritmo Evolucionário (EA), a Otimização por Busca em Grupo (GSO): o GSO-KM. Os EAs são meta-heurísticas inspiradas na seleção natural das espécies, que executam buscas globais no espaço do problema tratado, através de mecanismos que simulam processos biológicos, como mutação, recombinação e seleção [Eiben and Smith 2010]. Os EAs mantêm um conjunto (*população*) de soluções candidatas (*indivíduos*) do problema a cada iteração (*geração*) de suas buscas, sendo a busca realizada por cada indivíduo da população de um EA guiada com o objetivo de otimizar uma função critério (função de *fitness*). Entre os EAs mais populares estão o Algoritmo Genético (GA) [Holland 1992], a Evolução Diferencial (DE) [Storn and Price 1995], a Otimização por Enxame de Partículas (PSO) [Kennedy and Eberhart 1995], a Otimização por Busca em Grupo (GSO) [He et al. 2009] e a Otimização por Busca com *Backtracking* (BSA) [Civicioglu 2013]. Tais algoritmos variam principalmente em relação à forma como seus operadores evolucionários conduzirão a população no espaço de busca do problema.

O trabalho está organizado como segue. O algoritmo K-Means (Seção 2) e o GSO (Seção 3) são apresentados na sequência. Em seguida (Seção 4), o GSO-KM será introduzido. Após isso (Seção 5), a análise experimental e discussão dos resultados é feita, e, por fim, são apresentadas as conclusões e linhas para trabalhos futuros (Seção 6).

2. 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 entrada do algoritmo). Considere uma partição P_C de um conjunto de dados com N_O padrões (cada padrão é representado por um vetor $\mathbf{o}_j \in \mathbb{R}^m$, onde $j = 1, 2, \dots, N_O$) em C agrupamentos. Cada agrupamento é representado pelo seu vetor centroide $\mathbf{g}_c \in \mathbb{R}^m$ (onde $c = 1, 2, \dots, C$).

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.

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

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

onde n_c é o número de padrões atualmente associados ao agrupamento c . O K-Means visa a minimização da Soma dos Erros Quadráticos dos Agrupamentos (eq. (3)) a cada iteração de sua execução.

$$J(P_C) = \sum_{c=1}^C \sum_{\forall j \in c} d(\mathbf{o}_j, \mathbf{g}_c)^2 \quad (3)$$

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.

Calcule o valor de $J(P_C^0)$ (eq.(3)).

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 .

Calcule o novo valor de $J(P_C^t)$ (eq.(3)).

$t \leftarrow t + 1$.

fim_enquanto

Retorne a partição final $P_C^{t_{max}}$.

3. Otimização por Busca em Grupo (GSO)

O GSO é um EA 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 $\mathbf{X}_i^t \in \mathbb{R}^n$ e um ângulo de cabeça definido pelo vetor $\alpha_i^t \in \mathbb{R}^{n-1}$. A direção da busca do i -ésimo membro, que é um vetor $\mathbf{D}_i^t(\alpha_i^t) = (d_{i1}^t, \dots, d_{in}^t)$, pode ser calculada por α_i^t através de uma transformação polar para coordenadas cartesianas dada por:

$$\begin{aligned} d_{i1}^t &= \prod_{q=1}^{n-1} \cos(\alpha_{iq}^t), \\ d_{ij}^t &= \sin(\alpha_{i(j-1)}^t) \prod_{q=1}^{n-1} \cos(\alpha_{iq}^t) (j=1, \dots, n-1), \\ d_{in}^t &= \sin(\alpha_{i(n-1)}^t) \end{aligned} \quad (4)$$

Um grupo no GSO consiste de três tipos de membros: *producers*, *scroungers* 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. (5).

$$\mathbf{X}_z = \mathbf{X}_p^{t+r_1 l_{max}} \mathbf{D}_p^t(\alpha_p^t), \quad \mathbf{X}_r = \mathbf{X}_p^{t+r_1 l_{max}} \mathbf{D}_p^t(\alpha_p^t + \frac{\mathbf{r}_2 \theta_{max}}{2}), \quad \mathbf{X}_l = \mathbf{X}_p^{t+r_1 l_{max}} \mathbf{D}_p^t(\alpha_p^t - \frac{\mathbf{r}_2 \theta_{max}}{2}) \quad (5)$$

onde $r_1 \in \mathfrak{R}$ é um número aleatório obtido através de uma distribuição normal com média 0 (zero) e desvio padrão 1 (um), $\mathbf{r}_2 \in \mathfrak{R}^{n-1}$ é uma sequência uniforme obtida aleatoriamente no intervalo (0, 1), $\theta_{max} \in \mathfrak{R}^{n-1}$ é o ângulo máximo de busca e $l_{max} \in \mathfrak{R}$ é a distância máxima de busca, definida pela equação eq. (6):

$$l_{max} = \|\mathbf{U} - \mathbf{L}\| = \sqrt{\sum_{k=1}^n (U_k - L_k)^2} \quad (6)$$

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. (7)):

$$\alpha_p^{t+1} = \alpha_p^t + \mathbf{r}_2 \beta_{max} \quad (7)$$

onde $\beta_{max} \in \mathfrak{R}$ é o ângulo máximo de retorno.

Se após $a \in \mathfrak{R}$ 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.(8)).

$$\alpha_p^{k+a} = \alpha_p^k \quad (8)$$

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. (9)):

$$\mathbf{X}_i^{t+1} = \mathbf{X}_i^t + \mathbf{r}_3 \circ (\mathbf{X}_p^t - \mathbf{X}_i^t) \quad (9)$$

onde $\vec{r}_3 \in \mathfrak{R}^n$ 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. (10)).

$$\mathbf{X}_i^{t+1} = \mathbf{X}_i^t + l_i \mathbf{D}_i^t(\alpha_i^{t+1}), \quad l_i = ar_1 l_{max} \quad (10)$$

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 $\mathbf{X}_i^{(0)} \in G$.
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. (5)).
 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.
 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 $\mathbf{X}_p^{t_{max}}$.

4. GSO-KM

Esta seção apresenta o modelo híbrido entre GSO e K-Means [Pacífico and Ludermir 2018]: o GSO-KM. Considere novamente uma partição P_C de uma base de dados com N_O padrões, onde cada padrão é representado por um vetor $\mathbf{o}_j \in \mathbb{R}^m$, em C agrupamentos, sendo cada agrupamento representado por seu vetor centroide $\mathbf{g}_c \in \mathbb{R}^m$.

Cada membro do grupo G do GSO-KM será representado por um vetor $\mathbf{X}_i \in \mathbb{R}^n$, onde $n = m \times C$, correspondendo a C centroides, um para cada agrupamento da partição. Desta forma, cada indivíduo codificará uma partição $\mathbf{X}_i.P_C$ da base de dados, que representa a solução final oferecida pelo mesmo. Nessa partição, cada padrão da base de dados estará associado ao grupo $\mathbf{X}_i.\mathbf{g}_c$, ($c = 1, \dots, C$) ao qual sua distância seja mínima.

No GSO-KM, a inicialização do grupo será dada pela escolha aleatória de C padrões da base de dados em análise para compor os vetores centroides iniciais de cada membro \mathbf{X}_i . A função de *fitness* do GSO-KM será a Soma dos Erros Quadráticos dos Agrupamentos, dada pela eq. (3).

Após a inicialização e o cálculo do valor do *fitness* do grupo inicial $G^{(0)}$, o processo geracional do GSO-KM terá início, sendo cada membro modificado através dos operadores de *producing*, *scrounging* ou *ranging*. Após o processo geracional do GSO-KM, a melhor solução final encontrada $\mathbf{X}_p^{t_{max}}$ é melhorado pela execução do algoritmo K-Means, resultando na solução final \mathbf{X}_p^{new} fornecida pelo modelo.

O algoritmo GSO-KM é apresentado no Algoritmo 3.

No contexto de segmentação de imagens, cada pixel (em RGB) será considerado um **padrão**, de modo que $m = 3$, e cada imagem será considerada uma **base de dados**.

5. Avaliação Experimental

Nesta seção, os resultados experimentais obtidos pelo GSO-KM são apresentados. Cinco algoritmos de agrupamento são comparados ao GSO-KM, sendo dois algoritmos de agrupamento tradicionais e três algoritmos evolucionários: o K-Means, o Fuzzy C-Means [Bezdek et al. 1984], a Evolução Diferencial (DE-KM), a Otimização por Enxame de Partículas PSO e a Otimização por Busca com *Backtracking* (BSA-KM). Os EAs selecionados são adaptados ao contexto de agrupamento particional de modo semelhante ao processo adotado para o GSO-KM (vide Seção 4). Os hiperparâmetros

Algorithm 3 GSO-KM

$t \leftarrow 0$.

Inicialização: Para cada membro $\mathbf{X}_i^{(0)}$, escolha aleatoriamente C padrões da base de dados como seus vetores centroides de agrupamentos iniciais $\mathbf{X}_i^{(0)} \cdot \mathbf{g}_c$. Após isso, associe os demais padrões \mathbf{o}_i da base de dados ao agrupamento mais próximo.

Calcule o valor inicial da função de adequação de cada um dos membros $\mathbf{X}_i^{(0)}$.

enquanto (*condições de término não forem satisfeitas*) **faça**

Escolha o melhor membro do grupo como o *producer* \mathbf{X}_p^t para a iteração atual.

Execute o operador de *producing* (apenas o \mathbf{X}_p^t) de acordo com a eq. (5).

Escolha um percentual dos membros do grupo (exceto o \mathbf{X}_p^t) para executar o operador de *scrounging*, de acordo com a eq. (9).

Ranging: Os demais membros irão executar a estratégia de *ranging* (eq. (10)).

Determine as novas partições representadas por cada membro \mathbf{X}_i^{t+1} pela associação de cada padrão da base de dados ao agrupamento mais próximo.

Calcule o novo valor da função de adequação para cada membro \mathbf{X}_i^{t+1} , de acordo com a eq. (3).

$t := t + 1$.

fim_enquanto

Execute o algoritmo K-Means para melhorar a melhor solução $\mathbf{X}_p^{t_{max}}$.

retorne \vec{X}_p^{new} .

Tabela 1. Parâmetros fixos para cada algoritmo.

Algoritmo	Parâmetro	Valor
Todos os EAs	S e t_{max}	20 e 100
	C	5
GA	Taxa de elitismo	20%
	Taxa de <i>crossover</i> e Taxa de mutação	0.9 e 0.01
DE	Taxa de <i>crossover</i> e F	0.9 e 0.8
PSO	w	0.9 a 0.4
	c_1 e c_2	2.0
GSO	θ_{max} e α_0 Percentual de <i>scroungers</i> e β_{max}	π/a^2 e $\pi/4$ 80% e $\theta_{max}/2$

para cada um dos EAs são apresentados na Tabela 1 [He et al. 2009, Abdel-Kader 2010, Pacifico and Ludermitr 2018, Pacifico and Ludermitr 2019].

Os experimentos foram conduzidos através da escolha de 12 imagens (Fig. 1) da base de dados *Rice Leaf Diseases* [Prajapati et al. 2017] do UCI Machine Learning Repository [Asuncion and Newman 2007]. As imagens selecionadas possuem diferentes níveis de lesões ocasionadas pelas doenças, o que acarreta variação no grau de dificuldade das mesmas. Todas as imagens foram selecionadas da classe de doenças *Blight*, e redimensionadas ao tamanho de 309x90 pixels.

Os algoritmos selecionados são avaliados através do uso de quatro métricas bem estabelecidas da área de análise de agrupamentos: a Soma dos Erros Quadráticos dos Agrupamentos (eq. (3)), o Erro de Quantização Ponderado (eq. (11)), a Distância Intra-Cluster (eq. (12)) e a Separação Inter-Cluster (eq. (13)). Neste trabalho, todos os EAs fizeram uso da Soma dos Erros Quadráticos dos Agrupamentos como **função de fitness**.

$$J_{e_2}(PC) = \sum_{c=1}^C [(\sum_{\forall \mathbf{o}_j \in c} d(\mathbf{o}_j, \mathbf{g}_c)/n_c) \times (n_c/N_O)] \quad (11)$$

$$D_{max}(PC) = \max_{c=1, \dots, C} \{ \sum_{\forall \mathbf{o}_j \in c} d(\mathbf{o}_j, \mathbf{g}_c)/n_c \} \quad (12)$$

$$D_{min}(PC) = \min_{\forall c_1, c_2, c_1 \neq c_2} \{ d(\mathbf{g}_{c_1}, \mathbf{g}_{c_2}) \} \quad (13)$$

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

A análise experimental inclui uma avaliação baseada no valor médio obtido para cada uma das métricas adotadas em relação a dez execuções independentes dos experimentos para cada imagem, assim como um sistema de *ranks* elaborado através da

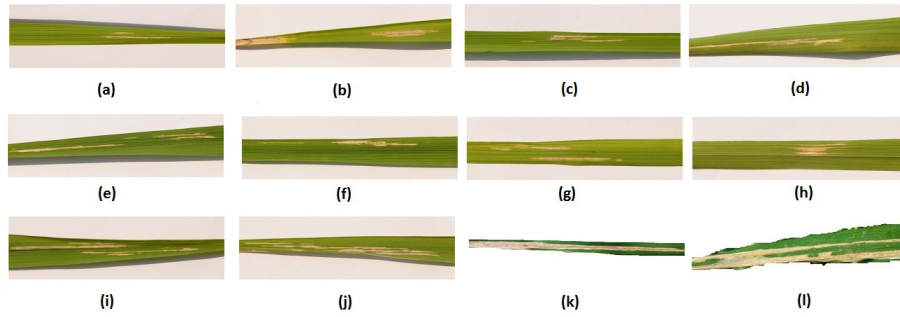


Figura 1. Imagens selecionadas [Asuncion and Newman 2007, Prajapati et al. 2017]: (a) DSC_0367 (Blight01), (b) DSC_0373 (Blight02), (c) DSC_0378 (Blight03), (d) DSC_0379 (Blight04), (e) DSC_0389 (Blight05), (f) DSC_0395 (Blight06), (g) DSC_0399 (Blight07), (h) DSC_0400 (Blight08), (i) DSC_0402 (Blight09), (j) DSC_0406 (Blight10), (k) DSC_0701 (Blight11), (l) DSC_0702 (Blight12).

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 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:

$$CD = q_{\alpha} \sqrt{\frac{n_{alg}(n_{alg}+1)}{6n_{bases}}} \quad (14)$$

onde n_{bases} representa o número de bases de dados (ou seja, de imagens), n_{alg} representa o número de algoritmos comparados e q_{α} são valores críticos baseados em estatísticas nos limites do modelo *t* de *Student* divididas por $\sqrt{2}$ [Demšar 2006].

Os resultados experimentais são apresentados na Tabela 2 e na Tabela 3. De acordo com uma análise empírica, é possível observar que o GSO-KM foi capaz de obter o melhor valor em relação à otimização da função de *fitness* em todos os casos, tendo sido o melhor modelo isoladamente em 6 dos 12 casos. O GSO-KM também foi o algoritmo que em média apresentou maior estabilidade nos experimentos, o que ressalta sua confiabilidade.

Em uma avaliação global, promovida pelo uso do sistema de *ranks* obtido pela aplicação do Teste de Friedman a cada uma das métricas de avaliação adotadas, podemos observar que o GSO-KM foi o responsável pelas melhores otimizações em relação a três das quatro métricas de avaliação (J , J_{e_2} e D_{max}), tendo o algoritmo sido responsável pela formação dos agrupamentos finais mais próximos aos seus respectivos vetores centroides (de acordo com o índice J), com a melhor distribuição média entre os padrões nos grupos (J_{e_2}), e com os agrupamentos mais compactos (D_{max}), porém o DE-KM foi responsável pela geração dos agrupamentos com melhor espaçamento entre si (de acordo com o D_{min}).

Em relação ao tempo médio de execução de cada algoritmo (Tabela 5), em decorrência de suas naturezas estocásticas, os algoritmos evolucionários possuem um custo

Tabela 2. Resultados Experimentais: valor médio (*Média*) e o desvio padrão (*Std*).

Imagem	Algoritmo	J^{\downarrow}		$J_{e_2}^{\downarrow}$		D_{max}^{\downarrow}		D_{min}^{\uparrow}	
		Média	Std.	Média	Std.	Média	Std.	Média	Std.
Blight01	K-Means	5.57x10 ⁶	8.12x10 ⁵	200.36	29.19	1560.75	11.857	270.83	490.2
	Fuzzy C-Means	5.76x10 ⁶	8.68x10 ⁵	207.29	31.212	1593.91	47.907	249.07	459.45
	BSA-KM	4.22x10 ⁶	6.00x10 ⁵	151.79	21.609	1545.20	1.9069	1085.58	366.88
	DE-KM	4.65x10 ⁶	9.98x10 ⁵	167.25	35.889	1561.89	27.248	2626.61	2294.24
	PSO-KM	4.41x10 ⁶	8.06x10 ⁵	158.71	28.991	1546.94	3.2037	968.20	492.00
	GSO-KM	4.03x10⁶	0	144.96	0	1544.99	0	1201.75	0
Blight02	K-Means	6.35x10⁶	281.50	228.47	0.0101	1361.08	0.9643	2375.91	6.795
	Fuzzy C-Means	7.97x10 ⁶	8.18x10 ⁵	286.85	29.439	1440.47	39.316	514.50	841.00
	BSA-KM	6.35x10⁶	0.3707	228.47	0.00001	1360.80	0	2380.40	0.2070
	DE-KM	6.82x10 ⁶	1.00x10 ⁶	245.54	36.003	1381.77	44.218	2633.79	534.17
	PSO-KM	6.35x10⁶	0.4943	228.47	0.00001	1360.80	0	2380.46	0.2761
	GSO-KM	6.35x10⁶	0.4943	228.47	0.00001	1360.80	0	2380.46	0.2761
Blight03	K-Means	7.66x10 ⁶	8.26x10 ⁵	275.76	29.73	1645.45	108.23	437.44	410.83
	Fuzzy C-Means	8.79x10 ⁶	63338.1	316.20	2.2775	946.51	131.66	199.15	0.8547
	BSA-KM	6.70x10 ⁶	9.38x10 ⁵	241.10	33.762	1605.43	64.706	1247.02	658.48
	DE-KM	7.00x10 ⁶	1.20x10 ⁶	251.96	43.475	1500.57	286.18	1723.52	1216.51
	PSO-KM	6.47x10 ⁶	5.93x10 ⁵	232.67	21.350	1552.32	223.21	1312.67	596.77
	GSO-KM	6.29x10⁶	3.13x10⁵	226.36	11.280	1539.66	216.19	1412.86	455.00
Blight04	K-Means	7.00x10⁶	3006.6	252.03	0.1081	1430.15	3.9739	1806.91	68.993
	Fuzzy C-Means	7.72x10 ⁶	2308.61	277.75	0.0830	1565.48	0.0062	1011.53	0.2283
	BSA-KM	7.01x10⁶	3011.45	252.12	0.1082	1427.48	4.0293	1749.57	69.143
	DE-KM	7.00x10⁶	3097.47	252.06	0.1113	1429.02	4.0931	1785.52	71.013
	PSO-KM	7.01x10⁶	1262.09	252.17	0.0453	1425.09	1.6870	1714.01	22.732
	GSO-KM	7.01x10⁶	2925.44	252.09	0.1051	1427.90	3.8616	1764.13	65.643
Blight05	K-Means	7.50x10 ⁶	7.18x10 ⁵	269.78	25.825	1537.71	18.357	354.01	418.52
	Fuzzy C-Means	8.43x10 ⁶	1.26x10 ⁶	303.40	45.630	1722.11	125.10	232.42	240.15
	BSA-KM	6.65x10 ⁶	8.22x10 ⁵	239.15	29.568	1511.82	24.376	856.26	480.31
	DE-KM	6.65x10 ⁶	8.22x10 ⁶	239.15	29.568	1507.91	26.297	862.61	484.58
	PSO-KM	6.66x10 ⁶	6.91x10 ⁵	239.57	24.864	1495.01	25.944	1176.22	401.64
	GSO-KM	6.14x10⁶	17.803	220.78	0.0006	1491.84	6.7198	1163.65	9.4660
Blight06	K-Means	5.81x10 ⁶	8.78x10 ⁵	208.96	31.590	1324.04	13.001	537.14	378.29
	Fuzzy C-Means	5.63x10 ⁶	1.05x10 ⁶	202.80	37.957	1468.20	138.62	600.61	439.01
	BSA-KM	5.32x10 ⁶	8.78x10 ⁵	191.59	31.604	1334.43	96.017	1172.10	159.29
	DE-KM	5.53x10 ⁶	9.65x10 ⁵	199.02	34.728	1350.69	39.745	1056.41	326.06
	PSO-KM	5.15x10 ⁶	7.66x10 ⁵	185.19	27.578	1343.19	28.666	904.44	313.91
	GSO-KM	4.78x10⁶	4339.74	172.13	0.1560	1342.09	30.923	1048.24	71.648

Tabela 3. Resultados Experimentais (Cont.).

Imagem	Algoritmo	J^{\downarrow}		$J_{e_2}^{\downarrow}$		D_{max}^{\downarrow}		D_{min}^{\uparrow}	
		Média	Std.	Média	Std.	Média	Std.	Média	Std.
Blight07	K-Means	3.56x10⁶	296.81	128.26	0.0106	876.22	43.116	1156.12	128.36
	Fuzzy C-Means	4.23x10 ⁶	2.23x10 ⁵	152.24	8.0326	1001.15	11.237	89.450	211.14
	BSA-KM	3.56x10⁶	395.74	128.26	0.0142	889.86	57.488	1115.53	171.14
	DE-KM	3.65x10 ⁶	2.75x10 ⁵	131.39	9.9026	894.77	65.262	1167.44	140.70
	PSO-KM	3.65x10 ⁶	2.74x10 ⁵	131.48	9.8767	906.61	67.251	1145.34	148.03
	GSO-KM	3.56x10⁶	334.80	128.26	0.0120	877.81	42.850	1153.90	127.77
Blight08	K-Means	3.75x10 ⁶	3.83x10 ⁶	135.02	13.781	1233.76	44.360	470.76	236.01
	Fuzzy C-Means	3.60x10 ⁶	8.9258	129.70	0.0003	1248.67	0.0073	461.90	0.0302
	BSA-KM	3.76x10 ⁶	4.07x10 ⁵	135.45	14.668	1231.58	48.380	705.22	255.76
	DE-KM	3.76x10 ⁶	4.07x10 ⁵	135.45	14.668	1231.58	48.380	715.22	275.51
	PSO-KM	3.67x10 ⁶	3.05x10 ⁵	131.97	10.999	1220.79	35.718	649.83	208.55
	GSO-KM	3.57x10⁶	205.03	128.49	0.0073	1208.31	7.4950	584.91	4.6025
Blight09	K-Means	6.43x10⁶	7.7312	231.46	0.0002	1262.87	2.5368	1578.47	0.5384
	Fuzzy C-Means	9.14x10 ⁶	1.36x10 ⁶	328.71	49.241	1828.73	294.33	390.60	531.89
	BSA-KM	6.43x10⁶	15.374	231.46	0.0005	1264.45	3.3615	1578.46	1.1314
	DE-KM	7.32x10 ⁶	1.42x10 ⁶	263.39	51.415	1258.41	8.6864	1838.18	418.22
	PSO-KM	6.43x10⁶	9.6409	231.46	0.0003	1263.46	3.3655	1578.57	0.6956
	GSO-KM	6.43x10⁶	3.3644	231.46	0.0001	1262.28	0.8776	1577.96	0.5835
Blight10	K-Means	6.65x10 ⁶	2.14x10 ⁶	239.31	77.066	904.38	165.02	520.55	839.79
	Fuzzy C-Means	5.57x10 ⁶	3.13x10 ⁵	200.36	11.283	932.49	2.9523	250.80	253.24
	BSA-KM	5.09x10 ⁶	5.50x10 ⁵	183.11	19.810	775.89	58.207	1590.80	992.53
	DE-KM	5.18x10 ⁶	5.51x10 ⁵	186.47	19.843	786.22	58.774	1397.18	1065.88
	PSO-KM	5.05x10 ⁶	4.83x10 ⁵	181.60	17.384	772.74	53.464	1336.64	998.13
	GSO-KM	4.76x10⁶	2.96x10⁵	171.50	10.645	741.73	32.635	1916.56	611.79
Blight11	K-Means	4.25x10⁶	0	153.06	0	1176.15	0	3565.95	0
	Fuzzy C-Means	4.29x10 ⁶	0.0720	154.28	0.000002	1243.84	0.0001	3221.94	0.0005
	BSA-KM	4.25x10⁶	17.127	153.06	0.0004	1174.64	3.1825	3562.23	7.8450
	DE-KM	4.63x10 ⁶	7.88x10 ⁵	166.51	28.368	1245.70	150.63	4468.34	2055.75
	PSO-KM	4.25x10⁶	43.879	153.06	0.0015	1175.41	2.3937	3562.79	6.8004
	GSO-KM	4.25x10⁶	48.163	153.06	0.0017	1175.06	2.3745	3564.72	3.9080
Blight12	K-Means	8.96x10⁶	170.63	322.51	0.0061	1305.20	2.2866	3539.19	43.911
	Fuzzy C-Means	9.28x10 ⁶	4.4417	333.99	0.0001	1088.67	0.0019	1472.28	0.0014
	BSA-KM	8.96x10⁶	282.14	322.51	0.0101	1307.37	3.7340	3579.31	69.745
	DE-KM	9.62x10 ⁶	1.38x10 ⁶	346.22	49.981	1266.58	81.836	4440.65	1894.51
	PSO-KM	8.96x10⁶	0	322.51	0	1304.48	0	3525.30	0
	GSO-KM	8.96x10⁶	0	322.51	0	1304.48	0	3525.30	0

Tabela 4. Avaliação Global: Rank médio do teste Friedman/Nemenyi por métrica, com uma Distância Crítica $CD = 2.1765$. Valores baixos de rank para J , J_{e_2} e D_{max} indicam melhores soluções (métricas de minimização), enquanto valores altos de rank para o D_{min} indicam melhores soluções (métrica de maximização).

Algoritmo	Rank $^{\downarrow}_J$	Rank $^{\downarrow}_{J_{e_2}}$	Rank $^{\downarrow}_{D_{max}}$	Rank $^{\uparrow}_{D_{min}}$
K-Means	30.8417	30.8417	31.9667	26.8250
Fuzzy C-Means	52.2833	52.2833	43.1083	7.7917
BSA-KM	25.6625	25.6625	28.5417	37.0917
DE-KM	28.2583	28.2583	28.8250	39.8333
PSO-KM	24.8042	24.8042	26.4500	34.3917
GSO-KM	21.1500	21.1500	24.1083	37.0667

Tabela 5. Tempo Médio de Execução de Cada Algoritmo (em segundos).

Imagem	K-Means	Fuzzy C-Means	BSA-KM	DE-KM	PSO-KM	GSO-KM
Blight01	0.2849	0.9120	37.491	26.991	34.063	32.241
Blight02	1.2117	0.7665	37.888	29.747	38.001	35.086
Blight03	0.2374	0.7562	37.161	30.106	37.616	34.532
Blight04	0.5998	1.4909	37.942	30.319	38.253	34.941
Blight05	0.3023	1.3381	37.449	30.154	37.639	34.449
Blight06	0.5601	0.9758	37.462	30.156	37.954	34.602
Blight07	0.6738	0.8944	37.537	30.399	37.974	34.789
Blight08	0.6433	1.3367	37.879	30.729	38.403	35.148
Blight09	0.9397	1.3249	35.863	30.366	38.051	34.636
Blight10	0.3424	0.8368	33.914	30.158	36.328	34.631
Blight11	0.7961	1.3105	32.909	30.391	34.353	31.679
Blight12	1.1645	1.5144	30.663	30.811	33.587	31.939

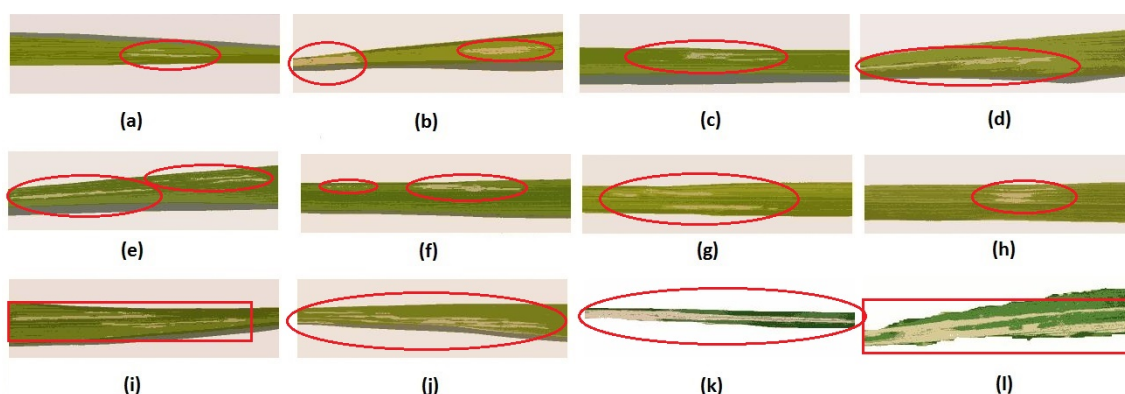


Figura 2. Resultados da segmentação obtidos pelo GSO-KM: (a)-(l) Blight01-Blight12. Regiões lesionadas em destaque.

computacional maior do que as técnicas de agrupamento tradicionais, como esperado. Porém, o custo médio de execução do GSO-KM é completamente compatível com o custo dos demais EAs testados, o que mostra que sua compatibilidade com o estado da arte dessas técnicas.

A Figura 2 apresenta os resultados da segmentação realizada pelo GSO-KM. Os resultados mostram que o GSO-KM foi capaz de identificar automaticamente as principais lesões nas imagens das plantas (regiões destacadas nas imagens), mesmo no caso de lesões relativamente pequenas em relação às dimensões das imagens.

6. Conclusões

Neste trabalho, um algoritmo híbrido entre o K-Means e a Otimização por Busca em Grupo é empregado na tarefa de segmentação automática de doenças em imagens de plantas: o GSO-KM. O GSO-KM é apresentado como um algoritmo de agrupamento particional, de modo a realizar a tarefa de segmentação sem a necessidade do fornecimento de qualquer tipo de conhecimento prévio sobre as imagens.

Como forma de validação do GSO-KM, 12 imagens foram obtidas através da base de dados *Rice Leaf Diseases* do UCI Machine Learning Repository, e quatro diferentes métricas de avaliação foram empregadas. Como algoritmos de comparação, dois algoritmos de agrupamento particionais da literatura (K-Means e Fuzzy C-Means) e três algoritmos evolucionários adaptados como algoritmos de agrupamento particionais (BSA, DE

e PSO) foram selecionados. Os resultados experimentais demonstram a capacidade do GSO-KM na realização da tarefa de sementação automática das lesões nas plantas, tendo tal modelo obtido os melhores valores nos experimentos para três (J , J_{e_2} e D_{max}) das quatro métricas avaliadas.

Como trabalhos futuros, pretendemos estender o GSO-KM ao contexto de Análise Automática de Agrupamentos (*Automatic Clustering*), de modo que o algoritmo seja capaz de identificar automaticamente a quantidade ideal de agrupamentos finais. O trabalho será também estendido pelo estudo de imagens de doenças em outras espécies de plantas. Por fim, pretendemos desenvolver um sistema completo e automático de categorização das doenças em imagens de plantas.

Referências

- Abdel-Kader, R. F. (2010). Genetically improved pso algorithm for efficient data clustering. In *Machine Learning and Computing (ICMLC), 2010 Second International Conference on*, pages 71–75. IEEE.
- Asuncion, A. and Newman, D. (2007). Uci machine learning repository.
- Barbedo, J. G. A., Koenigkan, L. V., Halfeld-Vieira, B. A., Costa, R. V., Nechet, K. L., Godoy, C. V., Junior, M. L., Patricio, F. R. A., Talamini, V., Chitarra, L. G., et al. (2018). Annotated plant pathology databases for image-based detection and recognition of diseases. *IEEE Latin America Transactions*, 16(6):1749–1757.
- 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.
- Bezdek, J. C., Ehrlich, R., and Full, W. (1984). Fcm: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2-3):191–203.
- 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.
- 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.
- Eiben, A. E. and Smith, J. E. (2010). *Introduction to evolutionary computing*, volume 2. Springer Berlin.
- 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.
- He, S., Wu, Q. H., and Saunders, J. (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.
- Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948. IEEE.
- Lee, S. H., Chan, C. S., Wilkin, P., and Remagnino, P. (2015). Deep-plant: Plant identification with convolutional neural networks. In *2015 IEEE international conference on image processing (ICIP)*, pages 452–456. 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.
- Pacifico, L. D. S., Britto, L. F. S., Oliveira, E. G., and Ludermir, T. B. (2019). Automatic classification of medicinal plant species based on color and texture features. In *2019 8th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 741–746. IEEE.
- Pacifico, L. D. S. and Ludermir, T. B. (2018). Hybrid k-means and improved group search optimization methods for data clustering. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.
- Pacifico, L. D. S. 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.
- Prajapati, H. B., Shah, J. P., and Dabhi, V. K. (2017). Detection and classification of rice plant diseases. *Intelligent Decision Technologies*, 11(3):357–373.
- Sawarkar, V. and Kawathekar, S. (2018). A review: Rose plant disease detection using image processing. *IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN*, pages 2278–0661.
- Solanke, S., Mehare, P., Shinde, S., Ingle, V., and Zope, S. (2018). Iot based crop disease detection and pesting for greenhouse-a review. In *2018 3rd International Conference for Convergence in Technology (I2CT)*, pages 1–4. 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.
- Tanmayee, P. (2017). Rice crop monitoring system—a lot based machine vision approach. In *2017 International Conference on Nextgen Electronic Technologies: Silicon to Software (ICNETS2)*, pages 26–29. IEEE.