Clusterização de Componentes de Indústria de Caminhões por meio de Metaheurísticas Bio-inspiradas.

Marcio Trindade Guerreiro[1], Diego Solak Castanho[1], Marcella Martins[1],
Fernanda Correa[1], Flávio Trojan[1], Hugo Siqueira[1]
[1]Universidade Tecnológica Federal do Paraná - UTFPR
Ponta Grossa-PR, Brasil
E-mails: marcio.guerreiro@daftrucks.com, diegocastanho@alunos.utfpr.edu.br, marcella@ utfpr.edu.br,
fernandacorrea@ utfpr.edu.br, trojan@ utfpr.edu.br, hugosiqueira@ utfpr.edu.br

***Resumo*—Este trabalho foi desenvolvido com o intuito de se realizar uma avaliação comparativa de performance de algoritmos de clusterização para mineração de dados. Na indústria, tais informações podem auxiliar na formação de estratégias para abertura de projetos, com o foco na redução de custos de composição dos automóveis em questão. Neste caso, a comparação é relativa aos preços de diferentes peças em um banco de dados de uma indústria automobilística, utilizando-se de métodos bio-inpirados: Evolução Diferencial (DE), Algoritmo Genético (GA) e Otimização por Enxame de Partículas para Clustering (PSOC). Como método comparativo implementou-se o K-Means. Durante esta avaliação das performances foi possível identificar que duas peças com mais de 90% de similaridade no design, peso e outras características, apresentavam diferença de preço na ordem de dez vezes. Os resultados demonstraram que os métodos bio-inspirados alcançam melhores desempenhos, superando a proposta clássica K-means, sobretudo o PSOC e o GA.**

***Palavras Chave—Clusterização; Otimização bio-inspirada; indústria automobilística*.**

I. INTRODUÇÃO

O desempenho na fabricação de produtos em empresas automotivas é de extrema importância para gerar redução de custos. O sucesso no gerenciamento de suas finanças e competências está ligado diretamente à capacidade de adaptação a um mercado aberto, de livre concorrência, produtos inovadores com prazos para entregas cada vez menores, alta qualidade e flexibilidade e manutenção de preços competitivos. Estes são desafios cada vez mais presentes na indústria e que impactam principalmente na capacidade de resposta da empresa, bem como na versatilidade e adaptabilidade para mudanças na produção devido a mercados turbulentos [1].

Este cenário provoca um aumento na velocidade de novos lançamentos e configurações que envolvem baixos volumes de produção e, consequentemente eleva os custos. Isso induz a uma descentralização no setor engenharia por família de produtos, que gera componentes com pequenas diferenciações de design, porém que podem desencadear um incremento significativo no custo de manufatura [2].

Com isso e com o avanço sistemas de informação e de gerenciamento na cadeia produtiva, a quantidade de dados gerados, coletados e armazenados nas várias aplicações aumenta significativamente [3]. Desta forma, ferramentas para executar automaticamente descoberta de conhecimento em grandes conjuntos de dados, como técnicas de mineração de dados, estão se tornando opções viáveis e cada vez mais eficientes [4].

Um conjunto de técnicas fundamentais e eficientes na mineração de dados é o agrupamento de dados, ou clustering [3]. Um problema de clusterização pode ser encarado de fato como uma tarefa de otimização. Em 1967, MacQuenn propôs o K-means, até hoje o mais conhecido algoritmo de agrupamento particional, o qual introduziu a ideia de centroide para representar um grupo. No entanto, com as aplicações modificadas para vários domínios, vários pesquisadores desenvolveram aprimoramentos nessa ideia inicial e novos algoritmos [5] [3].

Dentre os diversos métodos, tem ganhado destaque na literatura as propostas bio-inspiradas, que possuem como base algoritmos inicialmente desenvolvidos para otimização de funções reais. Neste sentido, este trabalho propõe abordar a Otimização por Enxame de Partículas para Clustering (PSOC), Algoritmo Genético (GA), Evolução Diferencial (DE) e o próprio K-means como comparativo.

Os referidos algoritmos foram aplicados a uma base de dados de uma indústria automobilística de modo que um primeiro recorte com dados rotulados é desenvolvido e um segundo, com mais instâncias e sem rótulos, é utilizado. Tais dados mostram em suas dimensões as características de cada peça. O objetivo é agrupá-las tendo em vista a grande similaridade, já que, apesar de muitas serem parecidas, apresentam custo até 10 vezes mais elevado para aquisição direta de um fornecedor.

Neste estudo, na Seção II são apresentados descritivos de cada técnica utilizada; Na seção III é discutida a metodologia aplicada, enquanto a Seção IV apresenta a base de dados, os resultados e análises. As conclusões estão na Seção V.

II. BASES TEÓRICAS

A. O Algoritmo K-means

O algoritmo K-means é muito conhecido pela sua capacidade de aplicação em grandes conjuntos de dados para resolver problemas de clusterização. Este algoritmo é amplamente utilizado porque pode ser facilmente implementável e também apresenta os resultados rapidamente. No entanto, o usuário deve especificar o número de clusters a priori [10].

O método divide as amostras em K grupos de variância igual, com N_y elementos, minimizando um critério conhecido como inércia ou soma de quadrados dentro do cluster. Esse algoritmo requer que o número de grupos seja especificado tendo a capacidade de adaptar-se a um grande número de amostras, e que pode ser aplicado na solução de problemas nas mais diferentes áreas do conhecimento. O ponto médio de cada agrupamento é comumente chamado de centroide [15]. Este é um vetor artificial gerado aleatoriamente e que possui o mesmo número de dimensões dos dados a serem agrupados.

O algoritmo procura minimizar o somatório da distância interna – distância intra-cluster (SSW - *sum of squares within clusters*) entre os dados e os centroides. A ideia por trás do K-means inicia-se com a geração aleatória dos centros. Em seguida alocam-se os dados em cada grupo, de modo que uma determinada amostra pertencerá ao cluster ao qual ele tem a menor distância ao respectivo centroide. Terminada esta fase, recalcula-se a nova posição dos centroides, para que ele se desloque para o centro geométrico do cluster [3], seguindo a equação (1).

$$m_k = \frac{1}{N_k}\sum_{i=1}^{N_k} z_i^k \tag{1}$$

em que N_k é o número de objetos alocados no cluster m_k na atual iteração e z_i é o i-ésimo objeto deste cluster.

O algoritmo se reinicia recalculando-se os grupos aos quais os dados pertencem e repete-se o processo até atingir algum critério de parada. O número de iterações e a estagnação dos centroides (ocorre quando os centroides não mudam de posição) são frequentemente usados como critério de parada.

As principais desvantagens do K-means são a sensibilidade para a inicialização, a necessidade de definir o número de grupos previamente e a dificuldade em separar dados sobrepostos [11]. Por conta disso, o método pode convergir para pontos bem distintos e não atua bem com dados que apresentam sobreposição [5].

Uma das formas de tentar minimizar os efeitos da inicialização é utilizar o *K-medoids*. Neste caso, em vez da geração aleatória de centroides, toma-se como ponto de partida k dados que pertencem ao grupo de amostras [15].

B. Otimização por enxame de particulas para Clustering

O algoritmo de Otimização por enxame de partículas (PSO) foi introduzido em 1995 por Kennedy e Eberhart, inspirado no comportamento social de um rebanho de aves. Aqui as soluções candidatas são chamadas de agentes ou partículas [23]. Além das aplicações em otimização com dados reais, o PSO tem sido amplamente utilizado em problemas de otimização binária, combinatória e em clusterização [3].

A codificação mais comumente utilizada para clusterização particional considera uma partícula como uma solução candidata completa. Neste caso, o agente conterá concatenados em um vetor as coordenadas espaciais de todos os k centroides. A aplicação do PSO pode ser feita de forma direta, utilizando as equações de posição e velocidade já conhecidas de acordo com (2) e (3).

$$x_p^{t+1} = x_p^t + v_p^{t+1} \tag{2}$$

$$v_p^{t+1} = \omega v_p^t + c_1 r_1 \cdot \left(pbest_p^t - x_t^p\right) + c_2 r_2 \cdot \left(nbest_p^t - x_p^t\right) \tag{3}$$

nas quais ω é o peso de inércia, x é a posição da partícula p, v é a velocidade, *pbest* é a melhor posição já encontrada pela partícula, *gbest* a melhor posição encontrada pelo grupo, c_1 e c_2 duas constantes previamente definidas e r_1 e r_2 dois números aleatoriamente gerados no intervalo [0,1].

O algoritmo é frequentemente inicializado espalhando aleatoriamente partículas sobre o espaço de busca. O mesmo processo é usado para gerar as velocidades iniciais, mas em alguns outros casos na literatura, estas são inicializadas com o valor igual a zero. O peso da inércia ω é geralmente menor que 1 e é usado para evitar a divergência da resposta. Também é comum limitar a velocidade das partículas a um intervalo [*-vmax; +vmax*] [4].

Uma definição essencial em qualquer problema que envolva algoritmos bio-inspirados é a definição de uma função de fitness, a qual avalia a qualidade da resposta de uma solução candidata. O fitness também é utilizado para definir *pbest* e *gbest*. Aqui a métrica utilizada é a soma das distâncias internas (SSW), definida na Seção III.

A seguir apresentamos o Algoritmo 1 com pseudocódigo para aplicação do PSO para clustering (PSOC) [3].

Algoritmo 1: Pseudocódigo PSOC

```
1 Inicialize cada partícula com um número de centroides
e avalie cada um deles
2 Atualize a melhor posição atual de cada partícula
3 Atualize a melhor partícula do enxame
4 enquanto critério de parada não for atingido faça
5     Atualize velocidade e posição
6     Avalie cada partícula
7     Atualize melhor posição de cada partícula
8     Atualize a melhor partícula do enxame
9 fim
```

C. Algoritimo genético (GA) para clusterização

O Algoritmo Genético (GA), é uma técnica de busca e otimização inspirada no princípio da evolução biológica. Há décadas o método tem sido aplicado na resolução de vários problemas em difer entes campos da ciência [25].

O GA é uma metaheurística populacional em que um conjunto de agentes (cromossomos ou indivíduos) interagem entre si e com o ambiente para otimização de uma função custo. A avaliação da função de fitness, solução populacional codificação e decodificação, seleção, reprodução e convergência são os princípios básicos do GA [25].

No Algoritmo 2 pode ser visualizado o pseudocódigo de funcionamento do GA. Da mesma forma que o PSO, um indivíduo será um vetor com os centroides candidatos. Entretanto, enquanto as partículas mudam de posição, aqui novas gerações são formadas para substituir as anteriores por meio de operadores genéticos.

Algoritmo 2: Pseudocódigo Algoritmo Genético

1 Inicialize a população de cromossomos com um número de centroides definido
2 Avaliação dos indivíduos
3 Verifique critério de parada
4 **enquanto** critério de parada não for atingido **faça**
5 Faça seleção dos pais
6 Aplique crossover nos pais selecionados
7 Aplique mutação nos filhos
8 Substitua a população pelos filhos gerados
9 **fim**

Inicialmente gera-se uma população aleatória com *n* indivíduos. Destes seleciona-se 2 "pais", os quais participam do crossover, ou troca de genes. O crossover acontece com probabilidade *Pc*. Esta seleção pode ser feita por diversos métodos sendo o da roleta o mais usual. Tal processo é repetido até a nova população de filhos ser do mesmo tamanho da inicial. Por fim, aplica-se a mutação, ou a perturbação gaussiana aleatória em um percentual pré-definido de genes. Tais etapas devem ser repetidas até ser atingido o critério de parada [13].

D. Evolução diferencial (DE) para clustering

A Evolução Diferencial (DE) é outra técnica inspirada na evolução das espécies. Na literatura, tem se mostrado uma candidata importante para problemas de otimização e *clustering*. Neste caso, os agentes são denominados vetores e são gerados como no PSO e GA [18].

Na DE, usa-se uma população de *N* soluções candidatas indicadas como $\mathbf{x}_{i,G}$, em que *i* denota cada agente e *G* representa a geração atual. Assim como no GA, os operadores são: crossover, mutação, seleção [14].

O processo de otimização é iniciado com a seleção de um vetor, nomeado *target vector*, e mais 3 outros escolhidos de forma aleatória $\mathbf{x}_{r1}$, $\mathbf{x}_{r2}$ e $\mathbf{x}_{r3}$. Estes geram um novo vetor mutado $\mathbf{v}_i$, através da Equação (4):

$$\mathbf{v}_{i,G+1} = \mathbf{x}_{r_1,G} + F \cdot (\mathbf{x}_{r_3,G} - \mathbf{x}_{r_2,G}) \quad (4)$$

em que *F* é definido pelo usuário.

De posse do vetor mutado e o *target vector* é então aplicado o crossover, que gera o *trial vector* $\mathbf{u}_i$. Este último é montado com base nos dois primeiros e na probabilidade r_j. Para o primeiro gene, sorteia-se um valor entre [0,1] e, se este for maior que r_j, o gene virá do vetor mutado. Caso contrário, do *target vector*. Tal processo se repete até um novo indivíduo das mesmas dimensões dos demais ser formado. Por fim, é feita uma seleção gulosa entre $\mathbf{v}_i$ e $\mathbf{u}_i$ [18].

A seguir apresenta-se o pseudocódigo da DE no Algoritmo 3.

Algoritmo 3: Pseudocódigo Evolução Diferencial

1 Inicialize os vetores
2 Avaliação da solução
3 Verifique critério de parada
4 **enquanto** critério de parada não for atingido **faça**
5 Aplique diferenciação vetorial (mutação)
6 Aplique recombinação vetorial (crossover), gerando o *trial vector*
7 Aplique seleção gulosa entre o *target* e o *trial vectors*
8 Avaliação dos novos vetores gerados e seleção
9 **fim**

III. METODOLOGIA

Nesta seção, discutimos a aplicação dos algoritmos supracitados em um banco de dados composto por peças que fazem parte do produto final de uma indústria automobilística de montagem. Inicialmente, há disponível 72 itens, previamente rotulados pela engenharia de manufatura e engenharia de produto, com 6 dimensões cada, sendo elas: classe de engenharia, peso, distância de transporte, consumo anual, classificação NCM (Nomenclatura Comum Mercosul) e ocorrências de qualidade. Em seguida, uma versão expandida com 321 amostras, mas sem rótulos será utilizada.

Neste trabalho são propostas como métricas de avaliação medidas que levam em consideração a distância interna, externa e outras definidas com base nestas: SSW (*sum of squares within clusters*) [6], SSB (*sum of squares between clusters*) [7], CH (*Calinski-Harabasz*) [8] e WB (*sum-of-squares based method*) [9].

O SSW é calculado conforme equação abaixo (4):

$$SSW_k = \sum_{i=1}^{n_k} dist(x_i, c_k) \quad (4)$$

sendo n_k o número de amostras e *dist* a distância euclidiana dada por (5):

$$dist(x_i, c_k) = \sqrt{\sum_{d=1}^{D} (x_{i,d} - c_{k,d})^2} \quad (5)$$

na qual $\mathbf{x}_i$ e $\mathbf{c}_k$ são referentes à amostra *i* e o controide *k*, respectivamente.

O SSB, que calcula a distância entre os centroides e deve ser o maior possível, é calculado conforme expressão (6):

$$SSB = \sum_{k=1}^{K-1} \sum_{l=k+1}^{K} dist(c_k, c_l) \quad (6)$$

A métrica CH (*Calinski-Harabasz*) é calculada conforme (7):

$$CH = \left(\frac{SSB/(m-1)}{SSW/(n-m)} \right) \quad (7)$$

no qual *n* é o número de elementos a serem agrupados e o *m* é o número de centroides.

O WB proposto por Xu, é dado por (8):

$$WB = m \cdot \left(\frac{SSW}{SSB} \right) \quad (8)$$

Com relação ao número de centroides para cada clusterizador, este foi variado de 2 a 10 no primeiro caso. Os algoritmos bio-inspirados foram rodados 30 vezes com 60 agentes cada e 200 iterações máximas. No PSO, as variáveis escolhidas foram ω=0,5 e c_1=c_2=2. O GA teve crossover com probabilidade de 80% e taxa de mutação de 30% para elevar o potencial de busca e evitar que a busca fique presa em um mínimo local. A DE foi escolhida com r_j = 20% e F=0,8. Todos esses valores de parâmetros foram determinados através de testes preliminares.

IV. Resultados e Discussões

Com aplicação dos 4 métodos na base rotulada com 72 elementos, foi possível gerar as Figuras de 1 a 4, que mostram o comportamento dos algoritmos em relação às métricas descritas para cada número de centroides selecionado, considerando o *m*. Observa-se que SSW e WB devem ser minimizadas, bem como SSB e CH maximizadas.

Pela análise dos resultados, percebe-se que em termos do SSW a partir de 6 centroides há uma estagnação no seu valor para todos os clusterizadores, assim como o WB. Como esperado, o SSB sempre é reduzido uma vez que quanto mais centroides, menor vai ser a soma de suas distâncias. Já o CH apresenta dois "joelhos" proeminentes para 5 e 6 grupos. Dessa forma, é possível adotar 6 como número adequado de grupos, considerando todas as métricas.

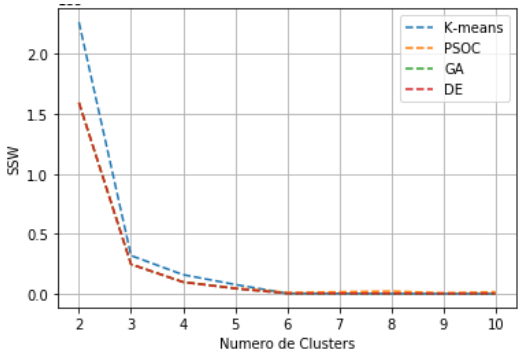

Figura 1 – Comportamento SSW variando a quantidade de centroides com 72 itens.

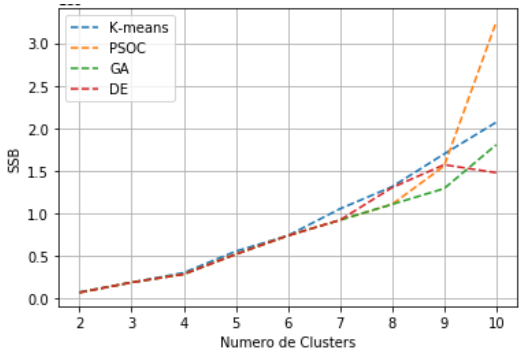

Figura 2 – Comportamento SSB variando a quantidade de centroides com 72 itens.

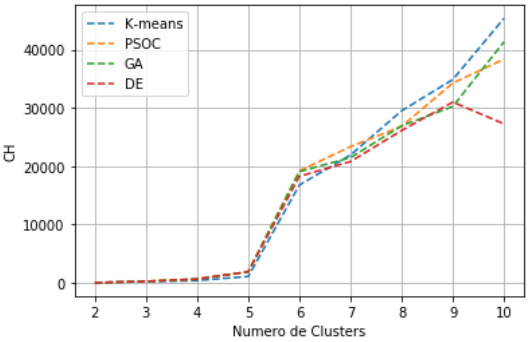

Figura 3 – Comportamento CH variando a quantidade de centroides com 72 itens.

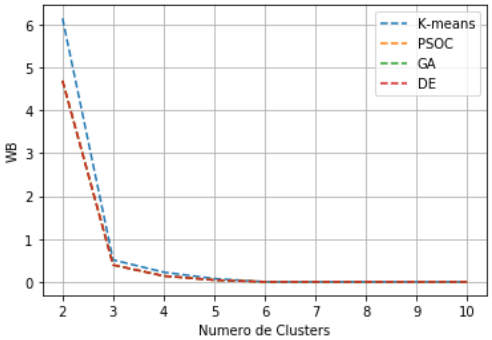

Figura 4 – Comportamento WB variando a quantidade de centroides com 72 itens.

Os resultados computacionais considerando as métricas e algoritmos estão sumarizados na Tabela 1.

Tabela 1: SSW, SSB, CH e WB do K-means, PSOC, GA e DE na base rotulada de 72 itens.

	Comparativo *K-means*, PSOC, GA e DE (6 centroides)			
	1º	2º	3º	4º
	PSOC	GA	DE	K-means
SSW (min)	506660	510110	532600	580122
SSB (max)	737.140.000	737.140.000	737.140.000	737.167.282
CH (max)	19204	19075	18269	16773
WB (min)	0,0041	0,0042	0,0043	0,0047
Assertividade	100%	100%	100%	100%

Todos os algoritmos propostos obtiveram uma assertividade de 100% dos valores quando comparado com o agrupamento manual (rótulo) definidos pela área de engenharia da referida indústria automobilística. Entretanto, considerando as métricas envolvidas, para o SSW percebe-se uma vantagem de performance para os algoritmos bio-inspirados em relação ao K-means, com destaque para o PSOC. Comportamento similar se observa para o CH e WB. Considerando o SSB, houve um empate entre as métricas bio-inspiradas, as quais superaram, novamente, o K-means.

O gráfico boxplot foi aplicado aos resultados para mostrar a dispersão dos mesmos nas 30 execuções e considerando o SSW, como mostrado na Figura 1. Observa-se que o PSO apresentou dispersão maior que os demais, embora o melhor resultado geral pertença a ele.

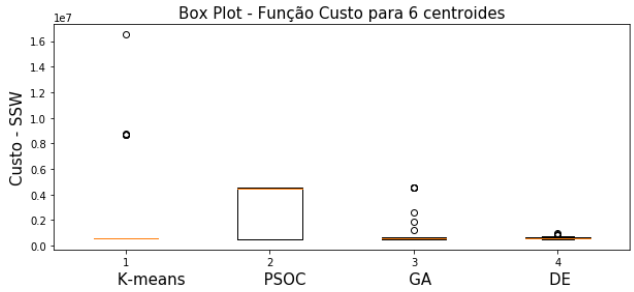

Figura 5 - Boxplot da função custo SSW

A seguir, apresentamos os resultados para a base de dados de 321 itens, em que não estão disponíveis os rótulos. Nas Figuras 6 a 9, podemos verificar o comportamento de cada um dos algoritmos propostos nas mesmas métricas e também com 6 dimensões.

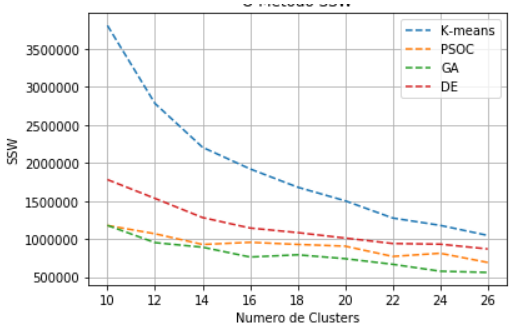

Figura 6 – Comportamento SSW variando a quantidade de centroides com 321 itens.

A base de dados atual é mais complexa não só pelo número de dados, mas também pela diversidade de peças. Neste caso, até a definição do número de grupos se torna mais difícil. Aqui observando o primeiro "joelho" vemos para SSW e CH ser 16 o número mais adequado. Ressaltamos que a própria indústria indica que valores de grupos muito altos não tem sentido prático.

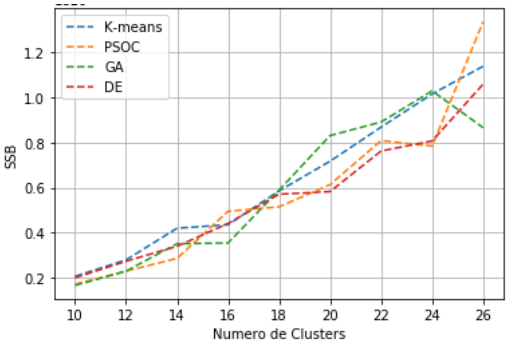

Figura 7 – Comportamento SSB variando a quantidade de centroides com 321 itens.

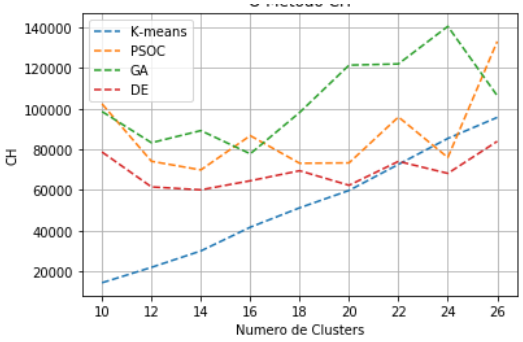

Figura 8 – Comportamento CH variando a quantidade de centroides com 321 itens.

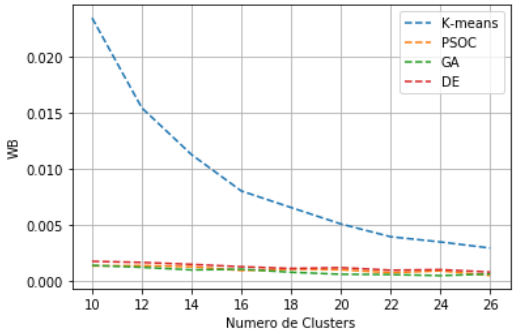

Figura 9 – Comportamento WB variando a quantidade de centroides com 321 itens.

Na Tabela 2 são mostrados os valores das métricas atingidas. Pode-se notar que o GA atingiu os melhores desempenhos para SSW e o PSOC para o SSB, CH e WB. Este resultado é intrigante tendo em vista que o GA foi ao mesmo tempo o melhor e o pior se consideramos métricas distintas e mais ainda se lembrarmos que a função custo a ser otimizada é o SSW. Os resultados desta tabela evidenciam a dificuldade de solução do problema.

Tabela 2: SSW, SSB, CH e WB do K-means, PSOC, GA e DE na base não-rotulada de 321 itens.

	Comparativo K-means, PSOC, GA e DE (16 centroides)			
	1º	2º	3º	4º
	GA	PSOC	DE	K-means
SSW (min)	763680	956800	1144800	1961319
SSB (max)	3.547.000.000	4.953.500.000	4.411.400.000	4.814.269.735
CH (max)	77797	86717	64545	40367
WB (min)	0,0011	0,0010	0,0013	0,0081

Por fim apresentamos o boxplot do SSW para 16 centroides, evidenciando que o GA atingiu a menor dispersão e o K-means a maior.

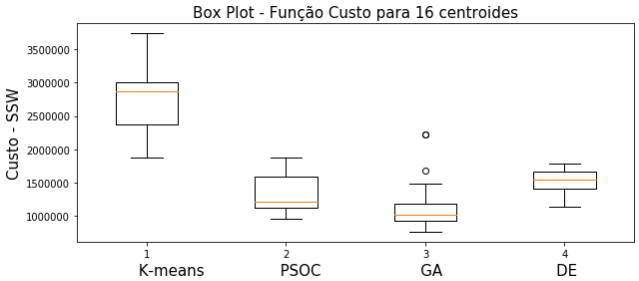

Figura 10 - Boxplot da função custo SSW

V. Conclusão

Este trabalho propôs a aplicação de modelos bio-inspirados de clusterização para agrupamento de peças de uma indústria do setor automobilístico: Otimização por Enxame de Partículas para Clustering (PSOC), Algoritmo Genético (GA) e Evolução Diferencial (DE). Como modelo comparativo foi utilizado o tradicional K-means.

Duas bases de dados foram abordadas: uma rotulada com 72 itens e outra expandida com 321 amostras e sem rótulos. Em ambas, as peças possuem 6 dimensões que as caracterizam. Pode-se observar que todas as técnicas convergiram para a quantidade de agrupamentos definida pelo time de engenharia da indústria para o primeiro no caso. Entretanto, considerando as métricas de qualidade abordadas, o PSOC alcançou grupos mais bem definidos. Na segunda tarefa, o GA foi o otimizador que alcançou menor SSW, enquanto o PSOC foi o melhor para as demais métricas.

Estes resultados mostram que os modelos bio-inspirados são ferramentas promissoras para a área de clusterização e que podem trazer ganhos importantes de desempenho para a indústria. Além disso, tal agrupamento leva a uma queda direta no custo de produção da referida manufatura.

Ficou evidenciado que dependendo do valor inicial de um centroide, algumas vezes o mesmo fica sem nenhum elemento. Este fenômeno se agrava quanto maior for o número de centroides, o que pode ser minimizado com a utilização da inicialização com medoides.

Para trabalhos futuros recomenda-se a expansão da base de dados, bem como a implementação de outros algoritmos como o Fuzzy C-Means, adequado a dados com sobreposição.

References

[1] P. Holtewert, T. Bauernhansl. “Optimal configuration of manufacturing cells for high flexibility and cost reduction by component substitution”. 48th CIRP Conference on MANUFACTURING SYSTEMS - CIRP CMS, 2015.

[2] J. O. Hansen, A. Kampker, J. Triebs, “Approaches for flexibility in the future automobile body shop: results of a comprehensive cross-industry study”, 51st CIRP Conference on Manufacturing Systems, 2018

[3] P.Santos, M.Macedo, E. Figueiredo, C. J. Santana Jr, F. Soares, H. Siqueira, A. Maciel, A. Gokhale, C. J. A. Bastos-Filho. “Application of PSO-BASED Clustering Algorithms on Educational Databases”. Learning and Nonlinear Models - Journal of the Brazilian Society on Computational Intelligence (SBIC), Vol. 15, Iss. 1, pp. 26-40, 2018.

[4] S. C. M. Cohen and L. N. de Castro, “Data clustering with particle swarms,” in IEEE Congress on Evolutionary Computations, 2006, p. 1792-1798.

[5] A. C. Benabdellah, A. Benghabrit, I. Bouhaddou, “A survey of clustering algorithms for an industrial context”. Second International Conference on Intelligent Computing in Data Sciences (ICDS 2018).

[6] G.H. Ball, L.J. Hubert, “ISODATA, A novel method of data analysis and pattern classification” (Tech. Rep. NTIS No. AD 699616), Menlo Park, CA: Standford Research Institute, 1965.

[7] L. Xu, “Bayesian Ying-Yang. Machine, clustering and number of clusters.” Pattern Recognition Letters, 18, p. 1167-1178, 1997.

[8] T. Calinski, and J. Harabasz. “A dendrite method for cluster analysis”. Communication in statistics, 3, pp. 1-27, 1974.

[9] Q Zhao, M Xu, PFränti. “Sum-of-Square Based Cluster Validity Index and Significance Analysis”. Department of Computer Science, University of Joensuu, Box 111, Fin-80101 Joensuu, Finland, 2009.

[10] W. Mohd, A.H. Beg, T. Herawan, K. F. Rabbi, “An Improved Parameter less Data Clustering Technique based on Maximum Distance of Data and Lioyd K-means Algorithm”, Procedia Technology 1, 2012, p. 367 – 371.

[11] H. Khoshdel and B. Saman, “A new hybrid learning-based algorithm fordata clustering,” in The 16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP 2012), 2012, pp. 095– 100.

[12] J. Kennedy, R. Eberhart, “Particle swarm optimization”, in: Proceeding of IEEE International Conference on Neural Networks, 4, Perth (Australia), 1944, pp. 1942–1948, 1995.

[13] A.M. Aibinu, H. B. Salau, N. A. Rahman, M.N. Nwohu, C.M. Akachukwu, “A novel Clustering based Genetic Algorithm for route optimization”, Engineering Science and Technology, an International Journal 19 (4), 2016, p. 2022-2034.

[14] M Ramadas, A Abraham, S Kumar, "FSDE-Forced Strategy Differential Evolution used for data clustering", Journal of King Saud University – Computer and Information Sciences 31, 2019, p. 52–61.

[15] P. Arora, D. Deepali, S. Varshney. "Analysis of K-Means and K-Medoids Algorithm For Big Data", International Conference on Information Security & Privacy (ICISP2015), 2015, pp. 11-1.

[16] L. Lorena, J. C. Furtado, "Constructive Genetic Algorithm for Clustering Problems." Evolutionary Computation 9 (3), 2001, pp. 3009-327.

[17] A. Banerjee, I. Abu-Mahfouz, "Evolutionary Clustering Algorithms for Relational Data, Procedia Computer Science 140, 2018, pp. 276-283.

[18] S. Paterlini, K. Thiemo, “High performance clustering with differential evolution.” In: Congress on Evolutionary Computation, CEC2004, 2, 2004, pp. 2004–2011.

[19] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth. “From Data Mining to Knowledge Discovery in Databases”, AI magazine, 17 (3), 1996, p.37-54.

[20] M.Macedo, E. Figueiredo, F. Soares, H. Siqueira, A. M. A. Maciel, A.Gokhale, C. J. A. Bastos-Filho. “Clustering Students Based on Grammatical Errors for On-line Education”. Learning and Nonlinear Models, 15 (1), 2018., pp. 26-40,.

[21] Hartigan, J.A. “Clustering algorithms”, New York, NY: Wiley, 1975.

[22] K. Compapong, S. Kasemvilas. “A comparison of effectiveness of risk data clustering method in Psychiatric Patient Service”. in: International Conference on Information Technology and Electrical Engineering (ICITEE), 2013. p. 2-7.

[23] R. Eberhart , J. Kennedy, “A new optimizer using particle swarm theory”, in: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995. p. 39-43.

[24] R. Chouhan, A. Purohit. "An approach for document clustering using PSO and K-means algorithm", in: 2nd International Conference on Inventive Systems and Control (ICISC), 2018. P.1380-1384.

[25] S. Ding, L. Xu, C. Su, H. Zhu, Using Genetic Algorithms to optimize artificial neural networks, Journal of Convergence Inf. Technol., 2010, p. 54-62.