UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

MARCIO TRINDADE GUERREIRO

ANÁLISE DE MÉTODOS DE AGRUPAMENTO DE DADOS PARA DETECÇÃO DE ANOMALIAS NA PRECIFICAÇÃO E CATEGORIZAÇÃO DE PEÇAS DA INDÚSTRIA AUTOMOTIVA

DISSERTAÇÃO

PONTA GROSSA

2021

MARCIO TRINDADE GUERREIRO

ANÁLISE DE MÉTODOS DE AGRUPAMENTO DE DADOS PARA DETECÇÃO DE ANOMALIAS NA PRECIFICAÇÃO E CATEGORIZAÇÃO DE PEÇAS DA INDÚSTRIA AUTOMOTIVA

Analysis of data clustering methods to detect anomalies in the pricing and categorization of automotive industry parts

Dissertação apresentada como requisito para obtenção do título de Mestre em Ciência da Computação da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador: Prof. Dr. Hugo Valadares Siqueira.
Coorientador: Prof. Dr. Flávio Trojan.

PONTA GROSSA

2021

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Ponta Grossa

MARCIO TRINDADE GUERREIRO

ANÁLISE DE MÉTODOS DE AGRUPAMENTO DE DADOS PARA DETECÇÃO DE ANOMALIAS NA PRECIFICAÇÃO E CATEGORIZAÇÃO DE PEÇAS DA INDÚSTRIA AUTOMOTIVA

Trabalho de pesquisa de mestrado apresentado como requisito para obtenção do título de Mestre Em Ciência Da Computação da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Sistemas E Métodos De Computação.

Data de aprovação: 05 de Fevereiro de 2021

Prof Hugo Valadares Siqueira, Doutorado - Universidade Tecnológica Federal do Paraná

Prof Carmelo Jose Albanez Bastos Filho, Doutorado - Universidade de Pernambuco (Upe)

Prof Flavio Trojan, Doutorado - Universidade Tecnológica Federal do Paraná

Prof Lourival Aparecido De Gois, Doutorado - Universidade Tecnológica Federal do Paraná

Prof Sergio Luiz Stevan Junior, Doutorado - Universidade Tecnológica Federal do Paraná

Documento gerado pelo Sistema Acadêmico da UTFPR a partir dos dados da Ata de Defesa em 05/02/2021.

AGRADECIMENTOS

Agradeço a minha família por todo o apoio e incentivo para a realização deste trabalho, especialmente minha esposa Eliana, a minha mãe Benedita, meu pai Antonio, ao meu irmão Fabio, minha irmã Elaine. Que sempre encontraram formas criativas me ajudando nas dificuldades e incentivando a seguir em diante.

Agradeço a todos os professores que durante esses tempo repassaram e compartilharam seu conhecimento, aconselharam e principalmente tiveram paciência e discernimento com minhas dificuldades. Em especial ao meu orientador Prof. Dr. Hugo Valadares Siqueira pela confiança que sempre depositou no meu trabalho, na minha capacidade, por toda a dedicação e horas de esforço empregadas para a realização dessa dissertação, muito mais que um orientador, e um guitarrista de primeira, um amigo que levo para a vida. Também um agradecimento especial ao meu coorientador Prof. Dr. Flávio Trojan, diga-se de passagem, que é um excelente baixista e vocalista.

Muito obrigado a todos os amigos que dividiram comigo momentos de felicidade, de angustia e ansiedade no decorrer deste trabalho. Aos companheiros do LICON, especialmente o Jonatas e o Solak serei eternamente grato pelo aprendizado diário, pelo trabalho em conjunto, pela amizade, pelas risadas, e pelo café.

A todos os colegas de trabalho que apoiaram e contribuíram com o desenvolvimento desta dissertação, não vou citar nomes aqui pois certamente cometeria alguma injustiça, eu simplesmente não tenho palavras para descrever tudo o que vocês fizeram por mim, não esquecerei jamais que ninguém chega a lugar algum sozinho, levarei vocês em meu coração por toda a vida.

Agradeço a Universidade Tecnológica Federal do Paraná (UTFPR) por me aceitar no processo seletivo e por ser uma fonte de inspiração e conhecimento para a realização deste trabalho.

RESUMO

GUERREIRO, Marcio Trindade. **Análise de métodos de agrupamento de dados para detecção de anomalias na precificação e categorização de peças da indústria automotiva**. 2021. 83 f. Dissertação de mestrado em Ciência da Computação, Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2021.

O desempenho de fabricação de uma empresa para redução de custos é uma ação de extrema importância para garantir a competitividade e evitar desperdício de recursos. As variáveis de processos e composição de produtos na indústria automotiva, geram diariamente uma quantidade de combinações de configurações de dados e de cenários que tornam inviável seu processamento de forma manual. Com isso, boa parte do conhecimento gerado acaba não sendo utilizado diretamente em itens similares, acarretando, por muitas vezes, em grandes diferenças de custos por pequenas diferenças estruturais e de design do produto. O objetivo desse trabalho é comparar o desempenho de algoritmos de clusterização e munido de etapas de pré-processamento para o agrupamento de peças, considerando características físicas de fabricação. Em seguida, é feita uma comparação de eficiência de custo de componentes similares, auxiliando na tomada de decisão para formação de estratégias para alcançar o ponto ótimo relativo aos custos desses componentes. Foi realizado o agrupamento através dos seguintes algoritmos: *K-Means*, *K-Medoids*, *Fuzzy C-Means* - FCM, Hierarquico, Agrupamento por Densidade Espacial em Aplicações com Ruido (*Density Based Spatial Clustering of Applications with Noise* - DBSCAN), Mapas Auto-Organizáveis (*Self Organizing Maps* - SOM), Otimização por Enxame de Partículas (*Particle Swarm Optmization* - PSO), algoritmo Genético (*Genetic Algorithm* - GA) e Evolução Diferencial (*Differential Evolution* - DE). Como métrica de comparação utilizou-se os seguintes índices: Soma dos Erros Quadráticos (*Sum of Squared Errors* - SSE), Soma das Distâncias Internas (*Sum of Squares Within Clusters* - SSW), Soma das Distâncias Externas (*Sum of Squares Between Clusters* - SSB), (*Calinski-Harabasz* - CH), o índice WB e Silhouette. O algoritmo hierárquico foi o que obteve os melhores resultados práticos, quando verificada a métrica SI e no resultado geral pontuando-se todas diferentes métricas aplicadas.

Palavras-chaves: Clusterização de Dados, Indústria automotiva, Redução de Custo.

ABSTRACT

GUERREIRO, Marcio Trindade. **Anomaly detection in pricing and classification of parts of an automotive industry using grouping methods (Clustering)**. 2021. 83 p. Masther's degree thesis in Computer Science, Federal Technology University - Paraná. Ponta Grossa, 2021.

The manufacturing performance of a company to reduce costs is an extremely important action to ensure competitiveness and avoid wasting resources. The variables of processes and composition of products in the automotive industry generates a daily number of combinations of data configurations and scenarios that make their manual processing unfeasible. As a result, much of the knowledge generated ends up not being used directly in similar items, resulting in large differences in costs due to small structural and product design differences. The objective of this work is to compare the performance of clustering algorithms and provided pre-processing steps for the grouping of parts, considering physical manufacturing characteristics. Then, a cost-efficiency comparison of similar components is made, assisting in the decision making for the formation of strategies to reach the optimum point regarding the costs of these components. Grouping was performed using the following algorithms: *K-Means*, *K-Medoids*, *Fuzzy C-Means* - FCM, *Hierarchical*, *Density Based Spatial Clustering of Applications with Noise* - DBSCAN, *Self Organizing Maps* - SOM, *Particle Swarm Optmization* - PSO, *Genetic Algorithm* - GA and *Differential Evolution* - DE. As a comparison metric, the following indices were used: *Sum of Squared Errors* - SSE, *Sum of Squares Within Clusters* - SSW, *Sum of Squares Between Clusters* - SSB, *Calinski-Harabasz* - CH, WB and Silhouette index. The hierarchical algorithm was the one that obtained the best practical results, when checking the SI metric and in the general result scoring all different applied metrics.

Keywords: Data Clustering, Automotive Industry, Cost Reduction.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

A eficiência na gestão de custos de fabricação em empresas automotivas é de extrema importância para gerar produtos competitivos. O sucesso financeiro e gerencial está ligado diretamente à capacidade de adaptação a um mercado aberto, de livre concorrência. Produtos inovadores com prazos reduzidos de entregas são cada vez mais requisitados, de forma que a alta qualidade percebida pelo consumidor, flexibilidade e preços competitivos tornaram-se pré-requisitos. Estes são desafios cada vez mais presentes na indústria e que impactam principalmente na capacidade de resposta de uma empresa, bem como na versatilidade, velocidade e adaptabilidade para mudanças na produção (HOLTEWERT; BAUERNHANSL, 2016)(KRAPPE; ROGALSKI; SANDER, 2006)(ARGONETO; RENNA, 2013).

Tal cenário leva à necessidade de constantes lançamentos de novos veículos, com novas configurações e opcionais. Estes envolvem, em muitos casos, baixos volumes de produção em configurações específicas e, consequentemente, provocam uma pressão na elevação dos custos. Isto também leva a uma descentralização no setor engenharia por família de produtos, o que por muitas vezes induz a elaboração de componentes com pequenas diferenciações de *design*, mas que podem desencadear um incremento significativo no custo de fabricação (HANSEN; KAMPKER; TRIEBS, 2018)(ELMARAGHY, 2008)(GAMEROS *et al.*, 2017).

Com o avanço da tecnologia de sistemas de informação e de gerenciamento na cadeia produtiva, há um aumento significativo da quantidade de dados gerados, coletados e armazenados nas várias aplicações, a cada fração de tempo (SANTOS *et al.*, 2017)(FIGUEIREDO *et al.*, 2019). Desta forma, ferramentas para executar automaticamente descobertas de conhecimento se mostram muito úteis em grandes conjuntos de dados e estão se tornando uma crescente e eficiente opção (COHEN; CASTRO, 2006)(ALAM *et al.*, 2014)(JOSÉ-GARCÍA; GÓMEZ-FLORES, 2016).

Este trabalho aborda a análise dados reais de uma montadora automotiva, a qual, devido a questões de confidencialidade, não pode ser nomeada. Nos dias atuais, há uma necessidade imediata de expandir o foco da empresa para incluir a análise de grandes conjuntos de amostras, os quais podem possuir grande dimensionalidade. Entretanto, a tarefa é desafiadora devido ao aumento incremental no volume de dados e a complexidade em termos de sua heterogeneidade.

Neste sentido, um desafio para identificação de anomalias de precificação de peças é monitorar se as classificações (rótulos) disponíveis nos dados cadastrais destes materiais são suficientemente adequadas para o correto agrupamento das peças, as quais deveriam possuir custos similares de fabricação. No caso específico aqui abordado, duas classificações previamente existentes no banco de dados foram avaliadas, a NCM (Nomenclatura Comum do Mercosul) e o número de classe. A sigla NCM é um código hierárquico de categorias. A cada código de NCM estão atreladas descrições e alíquotas de ICMS (BATISTA, 2017)(MACEDO, 2005)(FATALLA, 2016). Já o número de classe é um dado criado pela engenharia de produto no momento de cria-

ção de novas peças. A definição deste número é de livre escolha pelo engenheiro de produto e por este motivo apresenta uma caráter subjetivo e suscetível a diferentes interpretações. Ambos os rótulos apresentaram incoerências durante as análises amostrais. Dessa forma, serão investigadas diferentes técnicas de clusterização para tratar a identificação de anomalias de precificação em peças, usando as suas especificações como base.

O agrupamento (*clustering*) é uma ferramenta usada em muitas frentes de pesquisa, incluindo na mineração de dados, para que o conhecimento significativo seja extraído de amostras aparentemente não estruturadas (PANDOVE; GOEL; RANI, 2018)(HANCER; KARABOGA, 2017)(NANDA; PANDA, 2014). Clusterizar é uma maneira de agrupar dados e identificar padrões de forma coerente e não supervisionada. Dentre as diferentes técnicas, a literatura aponta que as técnicas de inteligência computacional colocam-se como alternativa aos métodos tradicionais como *K-Means* em problemas reais. Entretanto, o atual conhecimento sobre esta tarefa não é suficiente para definir o melhor algoritmo para todos os casos (FIGUEIREDO *et al.*, 2019)(PANDOVE; GOEL; RANI, 2018)(MUKHOPADHYAY *et al.*, 2013).

Com base no disposto, diversas abordagens de clusterização foram aplicadas, a saber: *K-means* e *K-medoids*, identificadas servindo de base de comparação para as demais técnicas e verificada em diversas publicações; Agrupamento por Densidade Espacial em Aplicações com Ruido (*Density Based Spatial Clustering of Applications with Noise* - DBSCAN), por se tratar de uma técnica de busca espacial comumente encontrada em trabalhos similares; *Fuzzy C-means* - FCM, sendo uma técnica difusa de agrupamento de dados que em casos específicos possuem resultados superiores ao *K-means* e *K-medoids*; Hierárquico, técnica que não necessariamente precisa-se definir a quantidade de centroides como parâmetro de entrada; Mapas Auto-Organizáveis (*Self Organizing Maps* - SOM), técnica de rede neural artificial que não necessita de dados de treinamento; Otimização por Enxame de Partículas (*Particle Swarm Optmization* - PSO), Algoritmo Genético (*Genetic Algorithm* - GA) e Evolução Diferencial (*Differential Evolution* - DE), por serem algoritmos bio-inspirados comumente encontrado na literatura, possuindo resultados relevantes na literatura por sua capacidade intrínseca de otimização local e global.

Como métrica de avaliação e comparação os seguintes índices foram abordados: Soma dos Erros Quadráticos (*Sum of Squared Errors* - SSE), Soma das Distâncias Internas (*Sum of Squares Within Clusters* - SSW), Soma das distâncias externas (*Sum of Squares Between Clusters* - SSB), (*Calinski-Harabasz* - CH), índice WB e Silhouette. Estas foram as métricas mais utilizadas no levantamento bibliográfico realizado para o desenvolvimento deste trabalho para agrupamentos com dados quantitativos.

Neste estudo, a seção 2 apresenta a metodologia de pesquisa aplicada, na seção 3 apresenta métodos e métricas de agrupamento de dados, a seção 4 uma explanação dos algoritmos de agrupamento utilizados, na seção 5 etapas de pré-processamento aplicadas na base da indústria automotiva em questão, na seção 6 é o estudo de caso. As conclusões estão na seção 7.

1.1 OBJETIVOS

As Seções 1.1.1 e 1.1.2 descrevem respectivamente, o objetivo geral deste trabalho e os objetivos específicos, além das etapas que foram realizadas para concretizá-lo.

1.1.1 Objetivo Geral

O objetivo principal desse trabalho é a análise de métodos de agrupamento de dados para detecção de anomalias na precificação e categorização de peças da indústria automotiva.

1.1.2 Objetivos Específicos

Para atingir o objetivo geral, os seguintes objetivos específicos serão estabelecidos, considerando uma base de dados real de uma indústria automotiva:

- Pré-processamento dos dados através da avaliação de distribuição dos dados, normalização, padronização e aplicação da técnica de Análise de Componentes Principais (*Principal Component Analysis* - PCA);
- Aplicação dos algoritmos *K-means*, *K-medoids*, DBSCAN, FCM, Hierárquico, Mapas Auto-Organizáveis, PSO, GA e Evolução Diferencial;
- Verificação de performance através da utilização das métricas SSE, SSW, CH, WB e Silhouette;
- Aplicação de técnicas de seleção da quantidade de centroides (*Elbow Curve*);
- Determinação da quantidade de rodadas para os algoritmos com comportamento estocástico utilizados nesta pesquisa.
- Comparação dos algoritmos aplicados e definição do mais apropriado.

1.2 JUSTIFICATIVA

O desempenho para redução de custos de fabricação de uma empresa é uma ação de extrema importância, pois sua saúde financeira está ligada diretamente a sua capacidade e velocidade de adaptação ao mercado. Desta forma, as chances de se aumentar os lucros é ampliada caso sejam otimizados os custos de manufatura de maneira rápida e eficaz.

O universo de análise bem como as variáveis de processos e composição de produtos na indústria automotiva geram diariamente uma quantidade de combinações, configurações e cenários que torna inviável seu processamento de forma manual. Como grandes volumes de dados são gerados, boa parte do potencial conhecimento alcançado acaba não sendo aplicado diretamente em itens similares, acarretando, por muitas vezes, em uma grande diferença de custos por pequenas diferenças estruturais e de design do produto (HOLTEWERT; BAUERNHANSL, 2016)(KRAPPE; ROGALSKI; SANDER, 2006)(ARGONETO; RENNA, 2013).

Estudos manuais realizados dentro da empresa onde está sendo desenvolvido este trabalho têm comprovado que através da análise das características de cada peça é possível agrupar e comparar o custos de fabricação por similaridade, possibilitando o cálculo de um índice de eficiência de custo. Isto permite a definição e priorização de projetos de redução de custo baseando-se em características como: material, processo de fabricação, peso, tipos de tratamento, pintura, dentre outras.

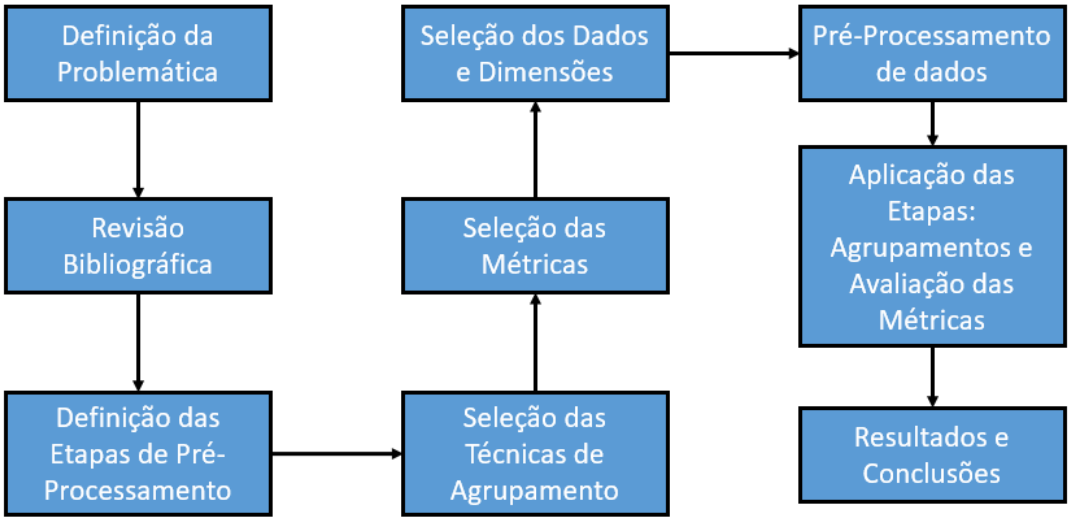
Esta análise permitiu identificar, por exemplo, que um batente da suspensão dianteira de dois produtos com capacidades de carga distintas possuíam mais de 90% de similaridade no design / peso e outras características. Entretanto, em relação aos custos de compra foi detectada uma diferença de aproximadamente 10 vezes entre as referidas peças.

Todo o processo aqui abordado visa auxiliar as organizações nas tomadas de decisão gerenciais para formação de estratégias mais eficientes em busca do ponto ótimo de custos de fabricação.

2 METODOLOGIA

Neste capítulo são apresentadas as etapas metodológicas desenvolvidas em direção aos objetivos específicos e geral. A Figura 1 sumariza os passos seguidos, os quais serão discutidos nas seções seguintes.

Figura 1 – Fluxograma Metodológico da Pesquisa.



Fonte: Autoria própria.

2.1 DEFINIÇÃO DA PROBLEMÁTICA

O primeiro passo deste trabalho foi a definição de qual é o problema a ser resolvido e que é esperado do projeto, etapa fundamental para a identificação e definição dos próximos passos. Basicamente procurou-se as respostas para os seguintes questionamentos:

- Qual é o problema a ser resolvido?
- Esse problema tem relevância financeira e é viável?
- Qual time irá trabalhar no projeto?
- Qual a técnica a ser aplicada?

O objetivo deste levantamento foi identificar a viabilidade e quais possíveis técnicas deveriam ser utilizadas para o desenvolvimento da investigação. Após esta etapa, foi escolhido o agrupamento (clusterização) de peças com similaridades físicas para posterior comparação de custos. Este problema apresenta um grande impacto financeiro com retorno anual esperado

próximo de 600.000,00 reais por ano. O time de projeto envolvido é multi-funcional, sendo composto por engenharia de produto, engenharia de manufatura, compras e logística, e as técnicas a serem aplicadas serão de clusterização de dados

De posse dessas definições, a próxima etapa iniciada foi a revisão bibliográfica.

2.2 REVISÃO BIBLIOGRÁFICA

Para a realização de uma revisão bibliográfica, é necessária uma decomposição em três fases principais: planejamento, condução e documentação da revisão (BRERETON *et al.*, 2007). Cada uma é a combinação de outros procedimentos mais simples, como mostra a listagem abaixo:

1. **Plano de Revisão**

   Questões especificas da pesquisa

   Desenvolvimento do protocolo de revisão

   Validação do protocolo de revisão

2. **Condução da Revisão**

   Identificar pesquisas relevantes

   Selecionar estudos primários

   Avalie a qualidade do estudo

   Extraia os dados necessários

   Sintetizar dados

3. **Fase de documentação**

   Escrever relatório de revisão

   Valide o relatório

De acordo com Kitchenham *et al.* (2009), o planejamento de uma revisão consiste em seis definições:

- Definir perguntas de pesquisa;
- Definir processo de pesquisa;
- Definir critérios de inclusão e exclusão;
- Definir avaliação da qualidade;

- Definir coleta de dados;
- Definir análise de dados.

2.2.1 Questões da pesquisa

Neste estudo, foi planejado investigar questões específicas relacionadas ao agrupamento de dados e, com as respostas, contribuir para outras pesquisas com o cenário quantitativo e qualitativo atual, além de indicar estudos futuros sobre esse assunto. As principais questões abordadas neste artigo são:

Q1 - como as abordagens de agrupamento podem ser classificadas?

Q2 - quais são as abordagens mais comuns de agrupamento que podem ser encontradas recentemente na literatura? E porque?

Q3 - quais são as funções de avaliação mais comuns (*fitness*) que podem ser encontradas?

Q4 - quais são as principais aplicações para essas abordagens?

Relacionada às questões acima, a revisão sistemática de Figueiredo *et al.* (2019), bem como as pesquisas publicadas por: Nanda e Panda (2014), Alam *et al.* (2014), Pandove, Goel e Rani (2018), José-García e Gómez-Flores (2016), Mukhopadhyay *et al.* (2013), fornecem uma grande contribuição qualitativa e quantitativa para este trabalho. Para indicar estudos adicionais, mais informações analíticas podem ser encontradas nos próximos sub-tópicos desta dissertação.

2.2.2 Metodologia de seleção de artigos

Uma revisão bibliográfica sobre um tópico, neste caso do estudo de agrupamento (*clustering*), deve identificar e destacar fontes específicas sobre o assunto. Estudos relacionados podem ser publicados em periódicos e conferências relacionados à aprendizado de máquina. Nesse contexto, alguns editores são dominantes, como a *Association for Computing Machinery (ACM)*, a *IEEE Computer Society (IEEE)*, a *Springer* e a *Science Direct* podem ser destacados, pois fornecem várias revistas e conferências, em áreas de interesse relacionadas, com "Otimização Bioinspirada", "Inteligência enxame"("*Swarm Intelligence*") e a "Computação evolutiva"("*Evolutionary Computation*") e Técnicas de *Clustering* no contexto de agrupamento de dados.

O procedimento de busca visou à identificação de estudos que seriam incluídos ou excluídos do conjunto final dos estudos de revisão, com base na conexão com a área de interesse. O plano de pesquisa envolveu a investigação automatizada nas quatro bibliotecas digitais já mencionadas. Além disso, antes da observação manual dos estudos, a filtragem automática era empregada para remover duplicatas como primeira etapa; para esse fim, o *Mendeley Desktop*,

versão 1.19.4, foi utilizada como ferramenta de gerenciamento. O conjunto de artigos retornado da consulta mencionada consistia em 853 papers. No entanto, a maioria destes foi identificada marginalmente relacionada à agrupamento de dados. A exclusão de artigos irrelevantes foi realizada manualmente.

2.3 ETAPAS DE PRÉ-PROCESSAMENTO

Para se evitar a polarização dos grupos formados pela má distribuição dos dados e diferença da grandeza entre as dimensões escolhidas, as etapas de pré-processamento do tipo normalização e padronização são de grande importância antes da aplicação dos respectivos algoritmos de aprendizado de máquina.

A normalização e padronização são abordagens em que os dados são escalados ou transformados para fazer uma contribuição igual de cada dimensão. O sucesso dos algoritmos de aprendizado de máquina depende da qualidade dos dados para obter um modelo generalizado do problema. A importância destes métodos foi comprovada previamente (SINGH; SINGH, 2020). A padronização deve ser aplicada posteriormente a etapa de normalização pois o efeito desta implica em distorções dos resultados quando utilizados de forma inversa. Técnicas de padronização exigem uma distribuição dos dados próxima de uma curva normal.

A última etapa de pré-processamento utilizada neste trabalho foi o PCA (*Principal Component Analysis*) que é um método baseado na projeção dos dados em um sub-espaço de menor dimensão. Este visa a possibilidade de se reduzir a quantidade de dimensões mas mantendo a sua significância, influência, neste sub-espaço projetado. As técnicas de redução dimensional reduzem o custo computacional em problemas multidimensionais, sem prejuízo ao resultado final devido a manutenção da significância das dimensões originais. Este é uma um método linear que captura da variabilidade do processo (PEARSON, 1901).

2.4 TÉCNICAS DE AGRUPAMENTO

Uma etapa crucial dentro dos objetivos propostos é a escolha dos algoritmos de clusterização para serem aplicados a base de dados da indústria automotiva. Durante a pesquisa bibliográfica várias técnicas foram encontradas e, como ponto de partida, observou-se que os algoritmos *K-means* e *K-medoids* serviam de base de comparação para as demais técnicas como sendo o algoritmo a ser batido. Em um segundo momento, verificou-se uma alta frequência de algoritmos evolutivos que apresentam em outros estudos resultado superior aos primeiros escolhidos. Foram abordados: Otimização por Enxame de Partículas (*Particle Swarm Optmization* - PSO), algoritmo Genético (*Genetic Algorithm* - GA) e Evolução Diferencial (*Differential Evolution* - DE).

Nota-se que os métodos acima descritos são técnicas particionais. Entretanto, a literatura apresenta outras abordagens, como a difusa e hierárquica. A técnica difusa encontrada com destaque foi o *Fuzzy C-means* - FCM; Já o algoritmo hierárquico é uma técnica determinística, ou seja, não possui o efeito da inicialização randômica.

Por fim, outras técnicas foram adicionadas como o Agrupamento por Densidade Espacial em Aplicações com Ruido (*Density Based Spatial Clustering of Applications with Noise* - DBSCAN), por se tratar de um modelo de busca espacial e que também apresentava características relevantes como não requerer como dado de entrada a quantidade de centroides. Por fim, escolheu-se os Mapas Auto-Organizáveis (*Self Organizing Maps* - SOM), técnica de rede neural artificial.

2.5 MÉTRICAS DE AVALIAÇÃO

Uma vez escolhido os algoritmos, a próxima etapa foi a definição das métricas de avaliação da qualidade dos grupos formados por cada um dos 9 algoritmos de clusterização definidos. Como ponto de partida observou-se que as mais utilizadas na literatura abordada foram: Soma das Distâncias Internas (*Sum of Squares Within Clusters* - SSW) e Soma das distâncias externas (*Sum of Squares Between Clusters* - SSB). Como o próprio nome define, o primeiro é baseado na soma das distâncias entre os elementos de um mesmo grupo e o segundo na soma das distância entre os centros de cada grupo, ou seja, um define o quão coeso são os grupos finais e o outro o quão distintos são os estes. A SSW deve ser minimizado e o SSB a maximizado.

Assim sendo, consequentemente identificou-se outros dos índices relevantes: (*Calinski-Harabasz* - CH) e índice WB. Estes possuem na sua representação matemática uma razão entre SSW e SSB. O que os diferencia é que um utiliza a quantidade de dados e quantidade de centroides em sua equação e o outro apenas a quantidade de centroides.

Já durante a fase experimental prévia a aplicação nos dados reais, notou-se que para a definição da curva de inflexão dos índices, o Soma dos Erros Quadráticos (*Sum of Squared Errors* - SSE) possuía uma característica de auxiliar nesta interpretação gráfica. Por fim, a métrica Silhouette, muito usual na literatura, também foi adicionada.

2.6 SELEÇÃO DOS DADOS DA INDÚSTRIA AUTOMOTIVA

A etapa de seleção dos dados foi iniciada através de uma reunião com um time multifuncional da empresa alvo deste estudo contemplando especialistas de múltiplos departamentos: engenharia de manufatura, engenharia de produto, compras, qualidade, materiais, logística, produção, desenvolvimento de fornecedores, etc. Neste evento foi explicada a problemática em conjunto com o time de desenvolvimento de novos negócios com o objetivo de agrupamento de

peças por similaridade. Assim o time chegou aos seguintes passos de seleção de dados:

Passo 1 - Carregar a base de dados completa do sistema SAP;

Passo 2 - Selecionar dados quantitativos contínuos;

Passo 3 - Eliminar componentes com dados faltantes;

Passo 4 - Considerar NCM como limite inferior da quantidade de grupos formados;

Passo 5 - Considerar a quantidade de diferentes números de classe de engenharia como o limite superior da quantidade de centroides.

O primeiro carregamento das informações sistêmica levantou 4267 itens com 14 dimensões, e modo que que dentre as dimensões levantadas as seguintes foram selecionadas:

(a) Peso

(b) Comprimento

(c) Largura

(d) Altura

(e) Volume

(f) Densidade

Esta seleção das respectivas dimensões foram feitas com o foco de agrupar peças por características físicas similares. No julgamento do time multi-funcional, estas características seriam as mais adequadas para execução de tal tarefa dentre as informações coletadas.

Desta forma o 1º e 2º passos foram concluídos. No 3º passo foi identificado uma série de dados faltantes o que causou a redução do carregamento inicial de 4267 para 2765 peças com 6 dimensões ou características. A Tabela 1 apresenta um resumo estatístico das dimensões escolhidas:

Tabela 1 – Resumo estatístico do banco de dados da industria automotiva.

Objeto	**(a)**	**(b)**	**(c)**	**(d)**	**(e)**	**(f)**
Média	7306	37485	19198	11408	483	576
Desvio P.	55966	51225	38980	29991	12315	14226
Mínimo	2,20	35	25	8	0,01	0,01
25%	57,32	10000	7000	1300	0,06	7
50%	416,67	15800	10000	5000	0,84	25
75%	2226,67	46500	20000	10000	7,03	74
Máximo	1565283	835000	963000	810000	493736	594300

Fonte: Autoria própria.

Algumas dimensões tais como descrição das peças, número de identificação, problemas de qualidade, localização do fornecedor, consumo anual e preço, além de alguns outros dados

que são categóricos, como classe de engenharia e classificação NCM (Nomenclatura Comum do Mercosul), precisam ser retirados para aplicação dos algoritmos.

Assim como orientado pelo time multi-funcional no 4º e 5º passos, 200 diferentes códigos NCM e 621 de classe de engenharia foram utilizados como referencia para definição da varredura de diferentes quantidades de grupos a serem formados pelos algoritmos de clusterização.

Com este pano de fundo o esperado é que este estudo mostre uma quantidade ideal de centroides entre 200 e 621. Por este motivo, foi feita uma varredura do número de centroides com a seguintes quantidades pré estabelecidas: 60, 160, 260, 360, 460, 560, 660, 760 e 860. Propositalmente não foram inclusos os valores de 200 e 621 pois em análises manuais, de forma amostral, foram encontradas incoerências em ambos os grupos.

3 MÉTODOS E MÉTRICAS DE AGRUPAMENTO DE DADOS

A extração de conhecimento (também conhecido como processo de KDD, do inglês *knowledge-discovery in databases*), é o processo de obter-se informações implícitas em grupos de dados, através de relações não triviais. O KDD busca padrões ocultos em um grande volume de informação utilizando alguma metodologia específica, que é útil para várias aplicações (ALAM *et al.*, 2014)(KARYPIS; KUMAR; STEINBACH, 2000)(FIGUEIREDO *et al.*, 2019).

O processo de agrupamento de dados, também conhecido como clusterização, contribui significativamente em muitas aplicações da extração do conhecimento. O objetivo desta classe de métodos é dividir os dados para maximizar a homogeneidade dentro de cada grupo e a heterogeneidade entre grupos distintos (ALAM *et al.*, 2014). Um problema fundamental deste processo é determinar a melhor estimativa do número de grupos (JOSÉ-GARCÍA; GÓMEZ-FLORES, 2016)(ALAM *et al.*, 2014).

Devido à falta de conhecimento prévio do domínio, é difícil escolher um número apropriado de grupos (*clusters*), pois parte das técnicas de aprendizado de máquina utilizadas nestes casos são não supervisionadas, já que os dados não possuem rótulos. Isto é particularmente difícil, especialmente quando os dados têm muitas dimensões, quando os grupos diferem amplamente em forma, tamanho e densidade e quando existe sobreposição entre dimensões distintas. No final dos anos 90, o problema do agrupamento automático deu origem a um novo ramo com a aplicação de meta-heurísticas bioinspiradas, que na ciências da computação é uso de computadores para sintetizar formas, comportamentos e padrões similares aos naturais (simulação da natureza), no desenvolvimento de novas técnicas para solução de problemas complexos (computação inspirada na natureza) (JOSÉ-GARCÍA; GÓMEZ-FLORES, 2016)(ZHONG; ZHANG; ZHANG, 2013)(RANA; JASOLA; KUMAR, 2011).

Essa divisão dos *clusters* pode ser realizada de várias maneiras (ALAM *et al.*, 2014). De acordo com a literatura, os métodos de agrupamento podem ser classificados como abordagens particionais, hierárquicas, sobrepostas e baseadas em grafos (FIGUEIREDO *et al.*, 2019)(NANDA; PANDA, 2014)(JOSÉ-GARCÍA; GÓMEZ-FLORES, 2016), os quais são descritos a seguir.

3.1 AGRUPAMENTO PARTICIONAL

O agrupamento particional divide um conjunto de dados em vários grupos com base em determinado critério conhecido como medida de adequação. Esta afeta diretamente a natureza da formação dos (*clusters*). Depois que uma medida apropriada é selecionada, a tarefa de particionamento é convertida em um problema de otimização, baseado na minimização da distância ou maximização da correlação entre padrões, otimizando sua densidade no espaço dimensional. Tais técnicas são populares em vários campos de pesquisa devido à sua capacidade de agrupar

grandes conjuntos de dados (NANDA; PANDA, 2014)(FIGUEIREDO *et al.*, 2019)(ALAM *et al.*, 2014). Os grupos são formados com base em semelhanças e diferenças entre os dados dos elementos dos grupos. As medidas de similaridade diferem de aplicação para aplicação, mas as mais comuns são baseadas em distância, padrão e densidade (ALAM *et al.*, 2014).

Sem dúvida, a técnica de partição mais proeminente e popular é o algoritmo *K-Means*, proposto há mais de 50 anos (JOSÉ-GARCÍA; GÓMEZ-FLORES, 2016)(MACQUEEN *et al.*, 1967). Ainda é amplamente utilizado para lidar com bancos de dados de alta dimensão devido à simplicidade, convergência rápida e baixa custo computacional (NANDA; PANDA, 2014). No entanto, diversos autores já identificaram possíveis desvantagens deste método, como a tendência a convergência para mínimos locais, dificuldade em determinação do número de grupos e alto enviesamento e dependência da inicialização. Ao longo dos anos, muitas propostas para aumentar seu desempenho foram desenvolvidas, como meios K-harmônicos, *Kernel K-Means* e *K-Medoids* (ALAM *et al.*, 2014)(NANDA; PANDA, 2014).

3.2 MÉTODOS HIERÁRQUICOS

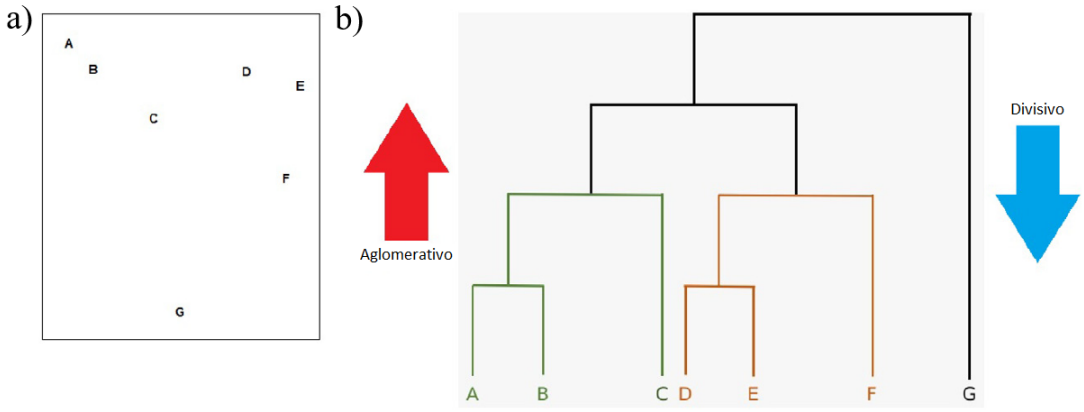
Os algoritmos de clusterização hierárquicos são os quais o processo de agrupamento atual é criado com base no nível hierárquico anterior. Diferentemente da abordagem particional, o *clustering* hierárquico não precisa especificar o número de grupos com antecedência (ALAM *et al.*, 2014)(JOSÉ-GARCÍA; GÓMEZ-FLORES, 2016). Essa abordagem é mais adequada para problemas envolvendo conjuntos de dados relativamente pequenos. O princípio básico envolve verificar a proximidade dimensional dos pontos, mas esta definição é subjetiva e existem muitas definições de medidas de similaridade e distância (PANDOVE; GOEL; RANI, 2018).

Tais abordagens criam uma estrutura chamada dendrograma, que consiste em uma estrutura em árvore, a qual representa a sequência hierárquica de partições aninhadas do conjunto de dados (JOSÉ-GARCÍA; GÓMEZ-FLORES, 2016). Eles podem ser de dois tipos: aglomerativos e divisivos, como exemplifica a Figura 2.

No primeiro caso (seta vermelha), cada amostra é inicialmente um *cluster*. Posteriormente, durante as seguintes etapas iterativas, os *clusters* mais semelhantes são reunidos com base na menor distância (dissimilaridade) usando algum método de ligação (ALAM *et al.*, 2014), que é uma forma de calcular a distância entre dois grupos. É uma estratégia de baixo para cima (*botton-up*). Na abordagem de divisão, todos os dados pertencem inicialmente a um *cluster* único, o qual é dividido em grupos menores, de acordo com um critério de distância predefinido (ALAM *et al.*, 2014)(JOSÉ-GARCÍA; GÓMEZ-FLORES, 2016). Pode ser classificado como um método descendente (*top-down*).

Considere um conjunto bidimensional de amostras na Figura 2. Após criar o dendrograma, destaca-se como podem ser as abordagens aglomerativas e divisivas. Por fim, dependendo do nível escolhido, o agrupamento final pode apresentar três ou dois grupos, como mostra

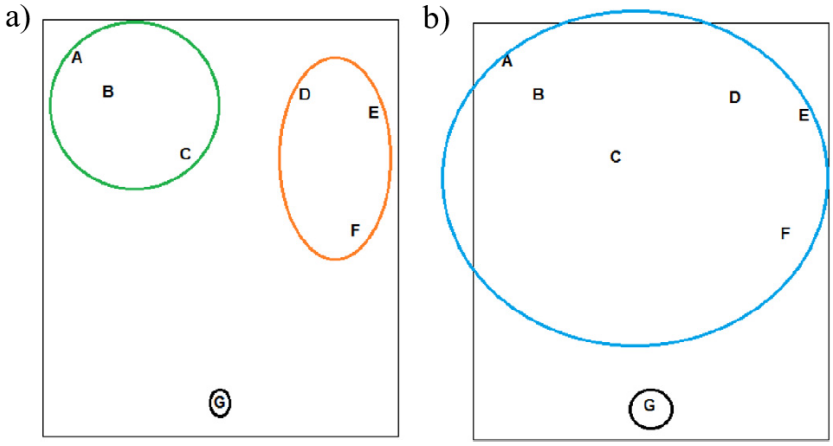
Figura 2 – Operação do método hierárquico: (a) conjunto de dados 2D; (b) dendrograma. (FIGUEIREDO *et al.*, 2019)



Fonte: Adaptado de Figueiredo *et al.* (2019).

a Figura 3.

Figura 3 – *Clusters* no método hierárquico: (a) 3 grupos; (b) 2 grupos. (FIGUEIREDO *et al.*, 2019)



Fonte: Adaptado de Figueiredo *et al.* (2019).

3.3 MÉTODOS DE SOBREPOSIÇÃO

Os métodos de sobreposição (*overlapping*) consideram que cada amostra pode ter semelhanças com diferentes grupos. Como consequência, ele pode pertencer a mais de um *clusters*. Esses algoritmos podem ser classificados em *soft* ou difuso (*fuzzy*) (FILHO *et al.*, 2015).

Métodos difusos são aqueles em que cada objeto pertence inteiramente a um ou mais *clusters* e a associação parcial não é permitida. Nos métodos difusos, um objeto pertence a todos os *clusters* de alguma forma, mas com diferentes graus de associação (ALAM *et al.*, 2014).

Os métodos de sobreposição mais utilizados na literatura seguem a abordagem difusa. As notas de associação são atualizadas durante o processo de agrupamento para que eles possam atribuir cada objeto a um único grupo usando a classificação de associação mais alta para

permitir o particionamento dos dados. O *Fuzzy C-Means* (FCM) é um algoritmo típico dessa classe (NANDA; PANDA, 2014).

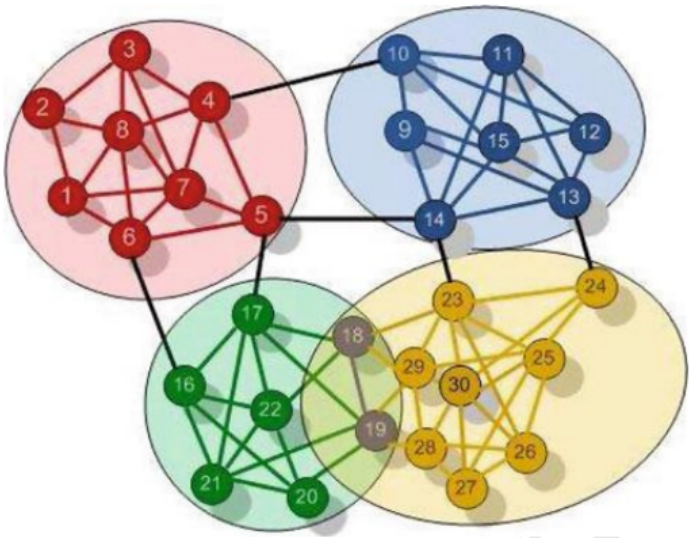
O FCM é uma alternativa interessante ao *K-Means*, pois apresenta características importantes como:

- O número de grupos pode ser definido automaticamente;
- Funciona bem com dados sobrepostos;
- É robusto para inicialização, ruído e que são valores atípicos (*outliers*).

O FCM define uma matriz de associação $U = \{u_{ij}\}_{i,j}^{N,n_k}$, em que $u_{ij} \in [0, 1]$ é o grau de pertinência. Observe que cada padrão é nomeado i para cada *cluster* j.

Um ponto importante, no entanto, é a definição pelo usuário do limite de interromper o processo iterativo, pois afeta a seleção do número de *clusters* (FILHO *et al.*, 2015). Um exemplo de agrupamento por sobreposição pode ser verificado na Figura 4 que mostra uma rede com 4 grupos com *overlap*, (BINESH; REZGHI, 2018).

Figura 4 – Nós com multi *Clusters*.



Fonte: Binesh e Rezghi (2018).

Veja que os dados 18 e 19 podem pertencer a mais de um grupo. Entretanto, a matriz de pertinência formada irá dizer a qual *cluster* tais dados terão maior similaridade.

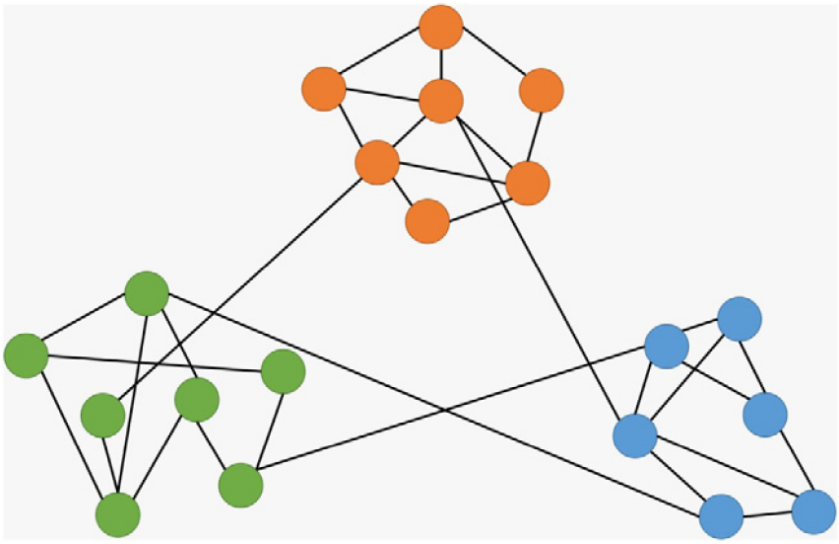
3.4 AGRUPAMENTO BASEADO EM GRAFOS

No agrupamento baseado em grafos, um grafo ponderado é criado conectando os dados de acordo com alguma medida de similaridade. O problema é reformulado usando a semelhança. Nesse sentido, o objetivo é encontrar a partição em que as arestas entre os diferentes grupos

tenham pesos pequenos e arestas dentro de um grupo apresentam pesos altos (HRUSCHKA *et al.*, 2009) (LUXBURG, 2007).

Existem várias maneiras de transformar um determinado conjunto de dados em um grafo. No grafo vizinhança, todas as amostras que apresentam distâncias aos pares menores que um limite estão conectados. No modelo de x-vizinho mais próximo, os dados mais próximos de um ponto específico estão conectados. No grafo totalmente conectado, todos os pontos de semelhança positiva estão ligados entre si e são ponderados pelo valor da similaridade (WU; ZHU; JI, 2014)(LUXBURG, 2007). Na Figura 5 é apresentado um exemplo de um processo de agrupamento usando este método, em que são destacadas as arestas e os grupos formados.

Figura 5 – Operação do método baseado em grafos (FIGUEIREDO *et al.*, 2019).



Fonte: Figueiredo *et al.* (2019).

No entanto, essa classificação em particional, hierárquico, sobreposição e grafos, não é única e definitiva. Alguns autores não consideram a abordagem de grafos como uma proposta diferente (NANDA; PANDA, 2014), enquanto outros definem a sobreposição como um caso particular de agrupamento particional (JOSÉ-GARCÍA; GÓMEZ-FLORES, 2016). Como mencionado, os algoritmos de *clustering* são desenvolvidos como um esquema não supervisionado. Metodologias supervisionadas são melhor definidas como métodos de classificação (NANDA; PANDA, 2014).

3.5 MÉTRICAS DE AVALIAÇÃO

Um ponto crucial na aplicação de abordagens de agrupamento é a definição de métricas de avaliação adequadas ao problema. Durante a revisão teórica um grande número delas foram detectadas e as seguintes foram selecionadas: Soma dos Erros Quadráticos (*Sum of Squared Errors* - SSE), Soma das Distâncias Internas (*Sum of Squares Within Clusters* - SSW), Soma

das distâncias externas (*Sum of Squares Between Clusters* - SSB), (*Calinski-Harabasz* - CH), o índice WB e Silhouette. Esta seleção leva em consideração a frequência de utilização na literatura estudada, bem como resultados previamente obtidos em estudos autorais publicados.

(a) A Soma dos Erros Quadráticos (*Sum of Squared Errors* - SSE) é uma medida de compactação dos grupos. É um índice de minimização, ou seja, quanto menor o valor mais compacto são os grupos (SU; DY, 2004)(THINSUNGNOENA *et al.*, 2015)(FRÄNTI; SIERANOJA, 2018).

Seja $\mathbf{X} = \{x_1, x_2, ..., x_n\}$ um conjunto de dados com n amostras. Suponha que as amostras em $\mathbf{X}$ têm rótulos rígidos que os marcam como representantes de k *clusters* sem sobreposição, ou $K = \{c_1, c_2, ..., c_k\}$, representando os centroides. O algoritmo de agrupamento busca encontrar a partição ideal $P = \{P_1, P_2, ..., P_m\}$, posicionando iterativamente os k centroides (ZHAO; XU; FRÄNTI, 2009). Essa métrica pode ser data pela equação 3.1:

$$SSE = \sum_{k=1}^{K} \sum_{i=1}^{n_q} dist(\mathbf{x_i} - \mathbf{c_k})^2 \tag{3.1}$$

em que $\mathbf{c_k}$ é o centroide do *cluster* k e $\mathbf{x_i}$ é o i-ésimo objeto do k-ésimo *cluster* (FIGUEIREDO *et al.*, 2019).

Note que com esta equação é calculada as distâncias entre os membros de cada grupo até o respectivo centroide ao quadrado. O objetivo da clusterização é a minimização do SSE nos *clusters* (SU; DY, 2004)(THINSUNGNOENA *et al.*, 2015)(FRÄNTI; SIERANOJA, 2018), de modo a deixá-los compactos e coesos.

(b) A Soma das Distâncias Internas (*Sum of Squares Within Clusters* - SSW) é uma medida de compactação dos grupos e também um índice de minimização, ou seja, quanto menor o valor mais compacto são os grupos. O que a diferencia do SSE é a não elevação ao quadrado do resultado da equação somatória (FIGUEIREDO *et al.*, 2019)(ALAM *et al.*, 2014)(JOSÉ-GARCÍA; GÓMEZ-FLORES, 2016). Tal medida é data pela equação 3.2:

$$SSW = \sum_{k=1}^{K} \sum_{i=1}^{n_q} dist(\mathbf{x_i} - \mathbf{c_k}) \tag{3.2}$$

em que $\mathbf{c_k}$ é o centroide do *cluster* k e $\mathbf{x_i}$ é o i-ésimo objeto do k-ésimo *cluster* (FIGUEIREDO *et al.*, 2019).

Esta métrica é a mais utilizada como função objetivo em agrupamento particional (HANCER; KARABOGA, 2017).

(c) A Soma das Distâncias Externas (*Sum of Squares Between Clusters* - SSB) mensura quão delineados estão os grupos. É um índice de maximização, ou seja, quanto maior o valor

mais distintos são os grupos formados pelos respectivos algoritmos (OZTURK; HANCER; KARABOGA, 2015)(ZHAO; XU; FRÄNTI, 2009). Trata-se de um critério de medição de separação dos grupos, e pode ser calculado pela equação 3.3:

$$SSB = \frac{1}{2} \sum_{k=1,l=1,l\neq k}^{K} dist(\mathbf{c_l} - \mathbf{c_k}) \tag{3.3}$$

onde $\mathbf{c_k}$ é o centroide do *cluster* k e $\mathbf{c_l}$ os demais *clusters* (FIGUEIREDO *et al.*, 2019).

(d) O índice *Calinski-Harabasz* - CH é uma composição entre as métricas SSW e SSB, que considera também a quantidade de centroides e de dados. É um índice de maximização (OZTURK; HANCER; KARABOGA, 2015)(JOSÉ-GARCÍA; GÓMEZ-FLORES, 2016)(CALIŃSKI; HARABASZ, 1974)(ZHAO; XU; FRÄNTI, 2009).

Dois métodos de análise de *clustering* podem diferir na escolha de uma medida de homogeneidade e de heterogeneidade. Isto impacta no procedimento de aplicação desta medida em pontos de agrupamento. Uma relação funcional escolhida como uma medida da homogeneidade dentro do *cluster* (ou a heterogeneidade entre *cluster*) geralmente reflete na relativa conveniência do agrupamento e depende da natureza do problema (CALIŃSKI; HARABASZ, 1974). A equação 3.4 mostra a relação dos elementos exposto acima conforme estabelecido para *Calinski-Harabasz CH*:

$$CH = \frac{SSB/(k-1)}{SSW/(n-k)} \tag{3.4}$$

Baseado na observação acima a proposta da métrica *Calinski-Harabasz CH*, a qual cria uma relação entre SSW, SSB, n , que é a quantidade de elementos total da amostra, e k que é a quantidade de centroides (ZHAO; XU; FRÄNTI, 2009).

(e) O índice WB também é uma composição entre os índices SSW e SSB, adicionando uma relação entre estas métricas e a quantidade de centroides. É uma métrica de minimização (OZTURK; HANCER; KARABOGA, 2015)(JOSÉ-GARCÍA; GÓMEZ-FLORES, 2016)(CALIŃSKI; HARABASZ, 1974)(ZHAO; XU; FRÄNTI, 2009).

Os métodos de soma dos valores mostrados nos itens (a), (b) e (c) desta seção são todos baseados na propriedade do SSW e SSB. As tendências de SSW normalizado e SSW / SSB são quase os mesmos, o que indica que o fator do numerador tem um efeito mais importante na divisão. Em outros índices do tipo WB, exceto no índice de Xu, descobrimos que eles aumentam / diminuem monotonamente ou precisam de um método adicional de detecção de joelhos, como diferenças sucessivas para obter número ideal de clusters. O índice de Xu tem um ponto mínimo claro de inflexão (ZHAO; XU; FRÄNTI, 2009).

A equação 3.5 apresenta a relação dos elementos exposto acima conforme estabelecido por Zhao, Xu e Fränti (2009):

$$WB = k \times SSW/SSB \tag{3.5}$$

em que k é a quantidade de centroides.

Neste caso, fica enfatizado o efeito de SSW com a multiplicação do número de *clusters*. As vantagens do método proposto são que ele determina o número de *clusters* por valor mínimo sem método de detecção do ponto de inflexão, e é bastante simples e eficiente (ZHAO; XU; FRÄNTI, 2009).

(f) O Índice Silhouette (SI) é uma métrica usada para avaliar a validade de agrupamento. Seu valor pode variar de (-1 < SI < 1) e mede o quanto um dado observado é similar às demais observações em seu próprio grupo, comparada às observações alocadas no grupo mais próximo a ele. Valores de SI próximos a -1 indicam que o dado foi erroneamente inserido no grupo de destino. Valores de SI próximos a zero mostram que o dado poderia estar tanto no seu grupo de destino quanto em algum outro. Valores do SI próximos a 1 indicam que dado está corretamente alocado (ROUSSEEUW, 1987)(FIGUEIREDO *et al.*, 2019).

A equação 3.6 apresenta a fórmula para o cálculo do Índice SI:

$$SI = \frac{1}{N}\sum_{i=1}^{N}\frac{b_i - a_i}{max(b_i, a_i)} \tag{3.6}$$

em que a_i é dado pela equação 3.7:

$$a_i = \frac{1}{N_k}\sum_{\mathbf{x_j}\in\mathbf{C_k}} dist(\mathbf{x_i}, \mathbf{x_j}) \tag{3.7}$$

e b_i é a dado pela 3.8 :

$$b_i = min_{h\in\{1,\ldots,K\},h\neq k}(\frac{1}{N_h}\sum_{\mathbf{x_j}\in\mathbf{C_h}} dist(\mathbf{x_i}, \mathbf{x_j})) \tag{3.8}$$

O SI é um índice de maximização, $SI \in [-1, 1]$, em que a é a distância média de um ponto $\mathbf{x_i}$ aos demais pertencentes ao mesmo grupo $\mathbf{c_k}$, b é a mínima distância média de um ponto $\mathbf{x_i}$ pertencente a algum outro grupo $\mathbf{c_h}$ (JOSÉ-GARCÍA; GÓMEZ-FLORES, 2016)(FIGUEIREDO *et al.*, 2019).

As principais diferenças entre as métricas selecionadas é que SSB é a soma das distancias entre os centroides de todos os grupos formados e o SSW é a soma das distancias de cada ponto ao seu respectivo centroide centroide. O SSE é similar ao SSW com uma elevação ao quadrado dentro do somatório que acentua a diferença de cada passo da varredura. O WB e o CH é uma razão entre SSB e SSW, emquanto o *Silhouette* possui uma formulação em que o resultado pode variar de [1,-1].

3.6 *CLUSTERING* AUTOMÁTICO - *ELBOW CURVE*

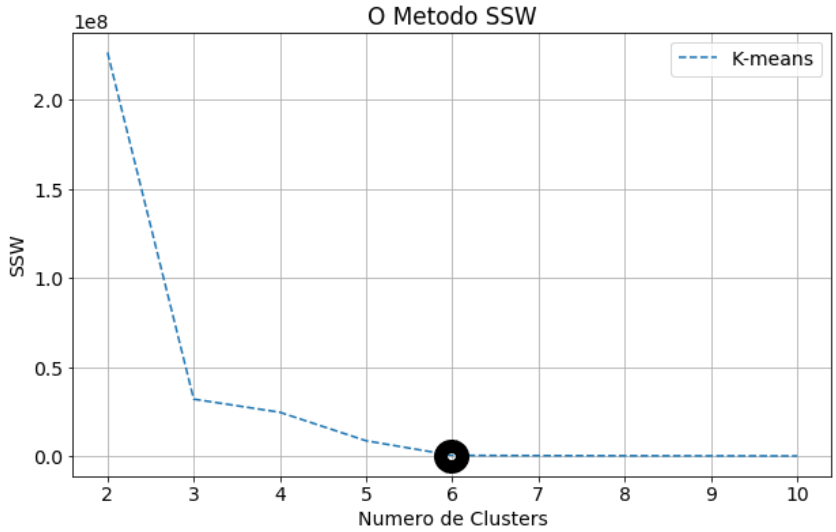
Na análise de clusterização, um problema fundamental é determinar a melhor estimativa do número de grupos. Devido à falta de conhecimento prévio do domínio, é difícil escolher um número apropriado, especialmente quando os dados têm muitas dimensões, quando os *clusters* diferem amplamente em forma, tamanho e densidade ou quando existe sobreposição entre os grupos (JOSÉ-GARCÍA; GÓMEZ-FLORES, 2016).

Além disso, para se encontrar uma solução ideal, temos um problema classificado como NP difícil para $n_k > 3$. Na teoria da complexidade computacional, esta classe é considerada como inerentemente difícil e sua solução requer recursos significativos, ou seja, não solúvel através de técnicas por força bruta em tempo aceitável (FALKENAUER, 1998). Portanto, mesmo para agrupamentos de tamanho moderado, a tarefa pode ser computacionalmente proibitiva (COWGILL; HARVEY; WATSON, 1999).

No método proposto, a localização de um valor k ideal é realizada pelo método conhecido como $Elbow$. suponha que o agrupamento é realizado pelo algoritmo *K-Means*, em que uma série de quantidade de centroides é experimentada e depois plotada em um gráfico no qual se torna possível avaliar o ponto em que o aumento deste número traz um benefício pequeno para métricas como SSW. Os resultados da simulação mostram que, no final da execução, em algum momento o ganho marginal cairá drasticamente, o que resulta um ângulo no gráfico. O valor de k correto, isto é, o número de *clusters* mais adequado, é escolhido neste ponto, daí o critério de cotovelo - *elbow curve* (BHOLOWALIA; KUMAR, 2014).

Na Figura 6 pode-se visualizar o ponto de curvatura (inflexão) em que é assumida a quantidade ideal de grupos, neste caso igual a 6.

Figura 6 – Ponto de curvatura *elbow curve* onde é determinado a quantidade ideal de *clustering* dos dados selecionados



Fonte: Autoria Própria.

4 ALGORITMOS DE AGRUPAMENTO

Assim como exposto anteriormente, na análise de agrupamento, um problema fundamental é determinar a quantidade ideal do número de *clusters*. Devido à falta de conhecimento prévio do domínio, é difícil escolher um número apropriado, pois parte das técnicas desenvolvidas são não supervisionadas já que os dados não possuem rótulos. Isto é especialmente desafiador quando: os dados têm muitas dimensões; *clusters* diferem amplamente em forma, tamanho e densidade; existe sobreposição entre os grupo (JOSÉ-GARCÍA; GÓMEZ-FLORES, 2016).

Neste trabalho serão abordados diferentes algoritmos e funções objetivos, com vistas a melhor resposta para o problema de agrupamento de componentes de uma indústria automotiva. Os escolhidos são: *K-Means*, *K-Medoids*, *Fuzzy C-Means - FCM*, algoritmo Hierárquico, Agrupamento por Densidade Espacial em Aplicações com Ruido (*Density Based Spatial Clustering of Applications with Noise* - DBSCAN), Mapas Auto-Organizáveis (*Self Organizing Maps* - SOM), Otimização por Enxame de Partículas (*Particle Swarm Optmization* - PSO), algoritmo Genético (*Genetic Algorithm* - GA) e Evolução Diferencial (*Differential Evolution* - DE).

Nas próximas seções serão discutidas de maneira mais detalhada cada um dos algoritmos citados.

4.1 *K-MEANS*

O algoritmo *K-Means* é muito conhecido pela sua capacidade de aplicação em grandes conjuntos de dados para resolver problemas de *clustering* (MACQUEEN *et al.*, 1967). Este algoritmo é amplamente utilizado porque pode ser facilmente implementado e também apresenta os resultados rapidamente. No entanto, o usuário deve especificar o número de *Clusters* a priori (MOHD *et al.*, 2012)(NANDA; PANDA, 2014).

O ponto médio de cada agrupamento é comumente chamado de centroide (ARORA; VARSHNEY *et al.*, 2016). Este é um ponto artificial gerado aleatoriamente e que possui o mesmo número de dimensões dos dados a serem agrupados. O método divide as amostras em K grupos de variância igual, com n_k elementos, minimizando um critério conhecido como inércia ou a soma do quadrado das distâncias de cada ponto ao centro do agrupamento. Em outras palavras, o algoritmo procura minimizar o somatório da distância interna (*Sum of Squares Within Clusters* - SSW) entre os dados e os centroides.

A ideia por trás do *K-Means* inicia-se com a geração aleatória da posição dos centros. Em seguida alocam-se os dados em cada grupo, de modo que uma determinada amostra pertencerá ao *cluster* ao qual ele apresenta 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* formado na iteração corrente (SANTOS *et al.*, 2017).

O algoritmo procura minimizar o somatório da distância interna (SSW) entre os dados e os centroides, pela equação 3.2. Nesta, a medida $dist$ é a distância euclidiana é dada pela equação 4.1:

$$dist(\mathbf{x_i}, \mathbf{c_k}) = \sqrt{\sum_{d=1}^{D}(x_{i,d} - c_{k,d})^2} \tag{4.1}$$

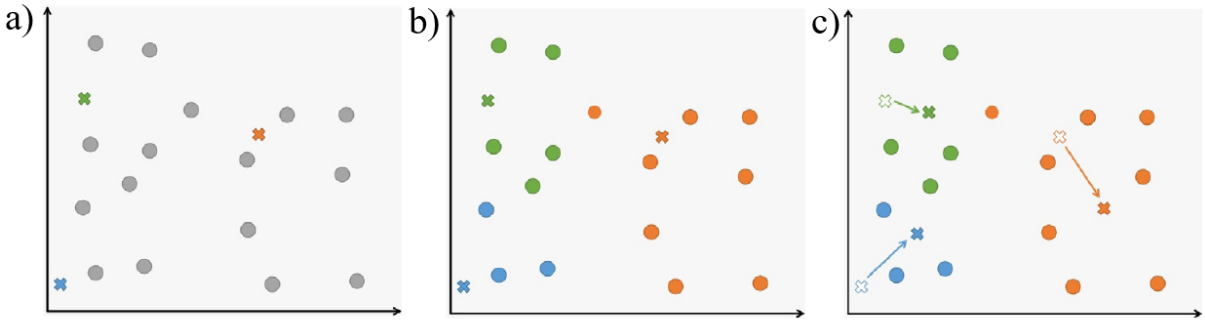
Em seguida, cada amostra é alocada no grupo do centro mais próximo e, posteriormente, as posições dos centroides são atualizadas pela equação 4.2:

$$c_k = \frac{1}{n_k}\sum_{i=1}^{n_k} {x_i}^k \tag{4.2}$$

Este processo é repetido até que um critério de parada seja alcançado. Os mais utilizados são o número de iterações ou um limite predefinido de uma medida de similaridade.

A Figura 7 mostra um exemplo de como o método funciona, no qual as esferas são os dados e as cruzes são os centroides.

Figura 7 – Exemplo de resultado da aplicação do K-Means: (a) geração aleatória inicial dos centroides; (b) alocação das amostras em cada grupo; (c) atualização da posição dos centroides (FIGUEIREDO *et al.*, 2019)



Fonte: Adaptado de Figueiredo *et al.* (2019).

Inicialmente gera-se aleatoriamente os centroides dentro do espaço de busca representados pelos "x"na Figura 7 a); em seguida, é identificado a qual centroide cada dado esta mais próximo Figura 7 b); a próxima etapa é adequar os centroides no ponto médio dos dados a que pertence, como mostra a Figura 7 c). Este processo é repetido iterativamente até que se alcance açgum critério de parada.

São bem conhecidos alguns inconvenientes do método. O primeiro é o alto grau de dependência da inicialização. Além disso, geralmente não funciona bem ao lidar com dados sobrepostos (FIGUEIREDO *et al.*, 2019).

Embora se possa provar que o procedimento sempre será encerrado, o algoritmo *K-Means* não encontra necessariamente a configuração ideal, correspondente ao mínimo global da função objetivo. O *K-Means* é um simples algoritmo que foi adaptado a muitos domínios desafiadores (PARK; LEE; JUN, 2006)(VELMURUGAN; SANTHANAM, 2010).

No pseudocódigo 1 pode-se verificar o funcionamento do algoritmo *K-Means*.

Pseudocódigo 1: *K-Means.*

```
1  início
2      parâmetros de entrada: dados, quantidade de centroides e critério de parada
3      geram-se os centroides aleatoriamente dentro do espaço dos dados
4      alocam-se os dados com menor distância a cada centroide formando os grupos
5      para cada grupo formado faça
6          recalcula-se o centro geométrico de cada grupo, gerando um novo
             centroide
7          aloca-se os dados a cada centroide formando os grupos
8          verifica-se critério de parada
9      fim
10     retorna-se os centroides e grupos formados
11 fim
```

Fonte: Autoria própria.

4.2 *K-MEDOIDS*

Um outro método de agrupamento particional é o *K-Medoids* (SOOD; BANSAL, 2013). O *K-Medoids* é um dos algoritmos de *clustering* mais populares, no qual um conjunto de dados de **X** é armazenado em K *clusters* fornecido pelo usuário. Este algoritmo trabalha com o princípio de minimizar as diferenças entre cada objeto em um *cluster* e seu objeto representativo.

Inicialmente, os centroides precisam ser definidos, e, neste caso, este é conhecido como Medóide (*Medoids*). A forma mais comum de proceder com esse processo é definir como centroides n_k amostras do grupo de dados, sorteadas de forma aleatória. Isto é feito para minimizar os efeitos da inicialização aleatória do *K-Means* (ARORA; VARSHNEY *et al.*, 2016).

Assim, espera-se encontrar uma única partição dos dados nos K *clusters*, de modo que cada um tenha um ponto mais representativo, ou seja, um ponto que seja o ponto mais centralmente localizado em relação a alguma medida, por exemplo, distância euclidiana(SINGH; CHAUHAN, 2011).Tal ideia reduz ruídos e discrepâncias, deixando o método mais robusto que o *K-Means* (ARORA; VARSHNEY *et al.*, 2016).

O método iterativamente encontra os k centroides e atribui cada objeto ao mais próximo centroide, em a coordenada de cada centroide é a média das coordenadas dos objetos no grupo. Infelizmente, sabe-se que o agrupamento é sensível à *outliers*, embora seja bastante eficiente em

termos de custo computacional. As etapas de execução são similares ao caso do *K-Means*.

No pseudocódigo 2 pode ser verificado o funcionamento do algoritmo *K-Medoids*

Pseudocódigo 2: *K-Medoids.*

```
início
    gera-se aleatoriamente a posição dos centroides sendo coordenadas dos dados
    encontra-se os Medoids mais próximos calculando a distância entre os pontos
      de dados x e os Medoids k e o mapa destes objetos.
    para cada Medoid k e cada ponto x associado ao Medoid k faça
        Troca-se k e x para calcular o custo total da configuração, e selecione o
          Medoids com o menor custo de configuração
        Se não houver alteração nas atribuições, repita as etapas 3 e 4 como
          alternativa.
    fim
    retorna-se os centroides e grupos formados
fim
```

Fonte: (ARORA; VARSHNEY *et al.*, 2016).

4.3 *FUZZY C-MEANS* - FCM

O *Fuzzy C-Means* - (FCM) tornou-se um importante método com muitas aplicações em agrupamento para problemas do mundo real. Entre os métodos de agrupamento difuso, o FCM é um dos mais conhecidos por sua simplicidade e eficiência, embora mostre algumas fraquezas, particularmente sua tendência em cair em mínimos locais.

Este algoritmo trata os *clusters* como grupos flexíveis para qual cada objeto de dados possui um grau de associação. Esses graus são avaliados entre 0 e 1, com um alto valor representando um alto grau de semelhança entre o objeto em estudo e o grupo (BEZDEK; EHRLICH; FULL, 1984).

O método desenvolvido por Bezdek, Ehrlich e Full (1984) tem como principal motivação abordar a deficiência no trabalho com grupos sobrepostos, dificuldade esta mostrada pelo algoritmo *K-Means*. Segundo Filho *et al.* (2015), o FCM apresenta características importantes como: o número de grupos pode ser definido automaticamente, funciona bem com dados sobrepostos e é robusto para inicialização, ruído e *outliers*.

O FCM define uma matriz de associação $U = \{u_{ij}\}_{i,j}^{N,n_k}$, em que $u_{ij} \in [0,1]$ é o grau de pertinência, $\sum_{j=1}^{n_k} u_{ij} = 1, \forall i$, e $0 < \sum_{i=1}^{N} u_{ij} < N$. Observe que cada padrão é nomeado i para cada *cluster* j.

O objetivo do método é minimizar a função de custo J_m descrita na equação 4.3:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{n_k} u_{ij}^m \parallel x_i - c_j \parallel^2 \tag{4.3}$$

sendo $\| \ . \ \|$ geralmente a norma euclidiana e m é o coeficiente de imprecisão fornecido pelo usuário.

Depois disso, o grau de pertinência e as posições centrais são calculados a partir das equações 4.4 e 4.5:

$$u_{ij}^{m} = \frac{1}{\sum_{k=1}^{C} \frac{\|x_i - c_j\|^{\frac{2}{m-1}}}{\|x_i - c_k\|}} \tag{4.4}$$

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m}.x_i}{\sum_{i=1}^{N} u_{ij}^{m}} \tag{4.5}$$

No pseudocódigo 3 pode-se verificar o funcionamento do algoritmo *Fuzzy C-Means*.

Pseudocódigo 3: *Fuzzy C-Means.*

início
  configure $\varepsilon > 0$. Fixe c, $2 \leq c \leq n$, $1 < m < \infty$; Fixe T (máximo número de iterações); inicialização randômica $u_{ik}(i = 1, ..., c$ e $k = 1, ..., n)$ do objeto k do grupo i, tal que $u_{ij} \in [0, 1]$, $0 < \sum_{k=1}^{n} u_{ik} < n$ e $\sum_{i=1}^{c} u_{ik} = 1$, para todo $k \in \Omega$;
  $t \leftarrow 0$;
  $J(t) \leftarrow 0$;
  $J(t+1) \leftarrow \sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik}^{m}) d_{ik}$;
  while $\mid J(t) - J(t+1) \mid \ > \varepsilon$ ***e*** $t < T$ **do**
    atualize a matriz protótipo K: Fixe o grau dos membros u_{ik}, atualize os protótipos
    atualize o grau dos membros matriz U: Fixe protótipo i, atualize o grau dos membros
    $J(t) \leftarrow J(t+1)$
    $J(t+1) \leftarrow \sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik}^{m}) d_{ik}$
    $t \leftarrow t + 1$
  end
  retorne as matrizes K e U
fim

Fonte: Autoria própria.

4.4 AGRUPAMENTO HIERÁRQUICO

Os algoritmos de *clustering* hierárquicos geram grupos em níveis sucessivos, nos quais o processo de agrupamento atual é criado com base no nível hierárquico anterior. Diferentemente da abordagem particional, o hierárquico não precisa especificar o número de *clusters* com antecedência (ALAM *et al.*, 2014)(JOSÉ-GARCÍA; GÓMEZ-FLORES, 2016).

No entanto, o custo computacional necessário para executar estes métodos é alto, o que pode levar a que um modelo inviável e que se torne um bloco para um conjunto de dados maior.

As abordagens hierárquicas criam uma estrutura chamada dendrograma, a qual consiste em uma estrutura em árvore e que representa a sequência hierárquica de partições aninhadas do conjunto de dados (JOSÉ-GARCÍA; GÓMEZ-FLORES, 2016).

O procedimento cria níveis de N *cluster* successivos, nos quais o grupo atual se baseia na solução obtida no nível anterior. Portanto, o método não requer conhecimento a priori sobre o número de clusters. No entanto, os grupos obtidos são estáticos porque os objetos atribuídos a um determinado *cluster* não podem ser movidos para outro (JOSÉ-GARCÍA; GÓMEZ-FLORES, 2016).

No pseudocódigo 4 pode-se verificar o funcionamento do algoritmo hierárquico (NGUYEN *et al.*, 2019).

Pseudocódigo 4: Hierárquico

```
1  início
2      entra-se os objetos do conjunto de dados, encontre dois objetos / clusters com
         a distância mais próxima e agrupe eles
3      o valor dos atributos no novo Ccuster é a média dos atributos de dois antigos
         objetos / clusters
4      com base nos centróides iniciais do cluster, calcule a média dos atributos dos
         membros dos clusters gerados nas etapas (1) a (4);
5      repita
6          se o objeto já apareceu na etapa (2)
7          em seguida, o objeto permanece no cluster original
8          caso contrário, calcule as distâncias entre o objeto e os Clusters existentes
9          se a menor distância abaixo do limite
10         em seguida, os objetos são atribuídos ao cluster mais próximo
11         senão, o objeto pertence ao cluster menor
12     até percorrer todos objetos;
13     atualiza-se o valor do atributo centroide
14     até que nenhum membro mude o cluster pertencente
15 fim
```

Fonte: (NGUYEN *et al.*, 2019)

Existem dois tipos de algoritmos hierárquicos: o divisivo e o aglomerativo (ALAM; DOBBIE; REHMAN, 2015)(ANDERBERG, 1973). No aglomerativo, inicialmente todo e qualquer elemento é um *cluster* individual. Uma estrutura semelhante a uma árvore é criada por mescla dos dois grupos mais próximos em gerações sucessivas. Este processo continua até que todos os dados sejam mesclados em um único grupo. No caso divisivo, o processo começa com todos elementos de dados em um único *cluster*, que é então dividido em agrupamentos menores com base na proximidade, até que os critérios relacionados ao número total de *cluster* seja obtido (ALAM; DOBBIE; REHMAN, 2015). É uma estratégia de cima para baixo (*top-down*).

Um número de questões relacionadas aos dados de entrada, as técnicas de agrupamento, a natureza do método de agrupamento e a qualidade da saída foram observados e comparados em diversas pesquisas encontradas na literatura, dificuldades estas que desafiam a maioria das

técnicas de agrupamento com eficiência de execução e qualidade da saída dos *clusters* formados (ALAM; DOBBIE; REHMAN, 2015).

Nas Figuras 2 e 3 apresentadas no capítulo 3 pode-se verificar visualmente os resultados simulados dos métodos divisivos e aglomerativos (FIGUEIREDO *et al.*, 2019). Através da analise destas figuras, principalmente a que apresenta dendrograma, também nota-se que a quantidade de centroides pode ser escolhida pelo usuário como um critério de parada.

4.5 AGRUPAMENTO POR DENSIDADE ESPACIAL EM APLICAÇÕES COM RUIDO (*DENSITY BASED SPATIAL CLUSTERING OF APPLICATIONS WITH NOISE* - DBSCAN)

Algoritmos de *clustering* baseado em densidade são muito usuais na literatura. O mais conhecido é o método de Agrupamento Espacial Baseado em Densidade em Aplicações com Ruído (*Density-Based Spatial Clustering of Applications with Noise* - DBSCAN)(ESTER *et al.*, 1996). Esta categoria pode encontrar agrupamentos de formato arbitrário, detectando-os através de esferas de alta densidade e, em seguida, mesclando as esferas próximas, para formar os *clusters*.

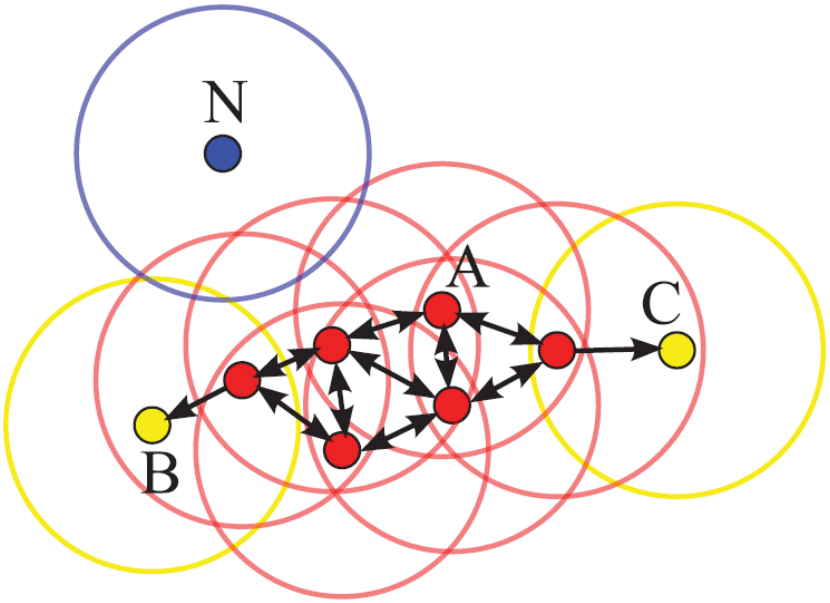
O modelo introduzido pelo DBSCAN utiliza uma estimativa simples do nível de densidade mínima, com base em um limiar para o número de vizinhos, *minPts*, dentro do raio ε (com uma distância arbitrária). Objetos com mais *minPts* vizinhos dentro desse raio (incluindo o ponto de consulta) são considerados um ponto central.

A intuição do DBSCAN é encontrar as áreas que satisfazem essa densidade mínima e que são separadas por áreas de menor densidade. Por razões de eficiência, o DBSCAN não realiza estimativa de densidade entre os pontos. Em vez disso, todos os vizinhos o raio ε de um ponto central são considerados parte do mesmo *cluster* (chamado densidade direta alcançável). Se algum desses vizinhos for novamente um ponto central, seus bairros serão incluído transitivamente (densidade alcançável). Pontos não essenciais neste conjunto são chamados de pontos de fronteira e todos os pontos no mesmo conjunto estão conectados à densidade. Pontos que não são acessíveis por densidade qualquer ponto principal é considerado ruído e não pertence a nenhum grupo (SCHUBERT *et al.*, 2017).

A Figura 8 ilustra o principio de funcionamento do de DBSCAN e os elementos formados a partir da aplicação deste algoritmo. O parâmetro *minPts* é 4 e o raio ε é indicado pelos círculos. Note que N é um ponto de ruído, A é um ponto central e B e C são pontos de fronteira (SCHUBERT *et al.*, 2017).

O DBSCAN é um algoritmo para calcular *clusters* de acordo com o modelo acima, exceto pontos de borda pertencentes a vários *clusters*, que são atribuídos apenas a um deles. Nesse algoritmo, o banco de dados é verificado linearmente em busca de objetos que ainda não foram processados. Pontos não essenciais são atribuídos ao ruído e, quando um ponto principal

Figura 8 – Elementos formados na aplicação do algoritmo DBSCAN



Fonte: Schubert *et al.* (2017).

é descoberto, seus vizinhos são iterativamente expandidos e adicionado ao *cluster*.

Este algoritmo básico tem uma abordagem padrão para calcular a relação fechamento transitiva, com uma modificação mínima em que apenas pontos centrais são expandidos. No entanto, isso pode gerar um algoritmo razoavelmente eficiente se um bom índice de banco de dados for usado (GUAN; YUEN; COENEN, 2019).

De acordo com Guan, Yuen e Coenen (2019) o DBSCAN tem três desvantagens:

- Primeiro, os parâmetros livres são difíceis de definir;
- Segundo, o número de *clusters* não pode ser controlado pelos usuários;
- Terceiro, o DBSCAN não pode ser usado diretamente como um classificador.

O DBSCAN não necessariamente usa a distância euclidiana ou a pontos em $\Re^d$, mas foi projetado para ser usado também com dados geográficos, polígonos e outros tipos de dados, como visto em (ESTER *et al.*, 1996)(SANDER *et al.*, 1998)(SCHUBERT *et al.*, 2015). Para além do "DBSCAN original" há outras propostas que o utilizam como base. Por exemplo, o *scikit-learn 0.16* (PEDREGOSA *et al.*, 2011) inclui uma variante que primeiro materializa todas as vizinhanças (que produz memória quadrática no pior dos casos), depois executa a expansão do *cluster* de maneira "vetorizada" apenas nos pontos principais. A complexidade geral do tempo de execução não foi aprimorado, mas é mais eficiente para ser executado pelo ambiente de tempo de execução *Python / NumPy* (TAN *et al.*, 2006).

No pseudocódigo 5 pode-se verificar o funcionamento do algoritmo DBSCAN (SCHUBERT *et al.*, 2017).

Pseudocódigo 5: DBSCAN

```
início
    entre com:
    banco de dados (X)
    ε: raio
    minPts: limiar de densidade
    dist: função distância
    label: rótulos dos pontos, inicialmente indefinido
    para para cada ponto p do banco de dados faça
        se label(p) ≠ indefinido então
            vizinhança N ← RangeQuery(DB, dist, p, ε)
        fim
        se | N | < minPts então
            label(p) ← Noise
            continue
        fim
        c ← nextclusterlabel
        label(p)
    fim
    Seed set S ← N \
    para cada q em S faça
        se label(q) = Noise então
            label(q) ← c
        fim
        se label(q) ≠ indefinido então
            continue
        fim
        Vizinhança N ← RangeQuery(DB, dist, p, ε)
        label(q) ← c
        se | N | < minPts então
            continue
        fim
        S ← S ∪ N
    fim
fim
```

(SCHUBERT *et al.*, 2017)

4.6 MAPAS AUTO-ORGANIZÁVEIS (*SELF-ORGANIZING MAPS* - (SOM)

Os Mapas Auto-Organizáveis (*Self-Organizing Maps* - SOM) são arquiteturas de redes neurais artificiais que são ajustadas utilizando métodos de aprendizado não supervisionado

para analisar vários conjuntos de dados, incluindo aqueles com valores ausentes. Podem ser utilizados para analisar e visualizar dados complexos multivariáveis e com múltiplos parâmetros (JUNTUNEN *et al.*, 2013).

Um problema típico para os SOM são aqueles em que as soluções (rótulos) não são dadas e a rede aprende de forma inteligente a agrupar os dados, reconhecendo padrões diferentes (ALHONIEMI *et al.*, 1999) (KOHONEN, 2002). A rede possui vantagens sobre outras abordagens multivariadas porque é capaz de lidar com as não linearidades em um sistema, pode ser produzida usando os dados sem conhecimento mecanicista do sistema, é capaz de lidar com dados ruidosos, irregulares ou ausentes, pode ser facilmente e rapidamente atualizada e pode interpretar informações com muitas variáveis ou parâmetros, utilizando recursos de visualização (HONG; ROSEN, 2001)(LIUKKONEN *et al.*, 2011a)(LIUKKONEN *et al.*, 2011b). Além disso, o SOM combina os diferentes conjuntos de dados para uma única forma visual e facilmente compreensível (KOHONEN, 2002).

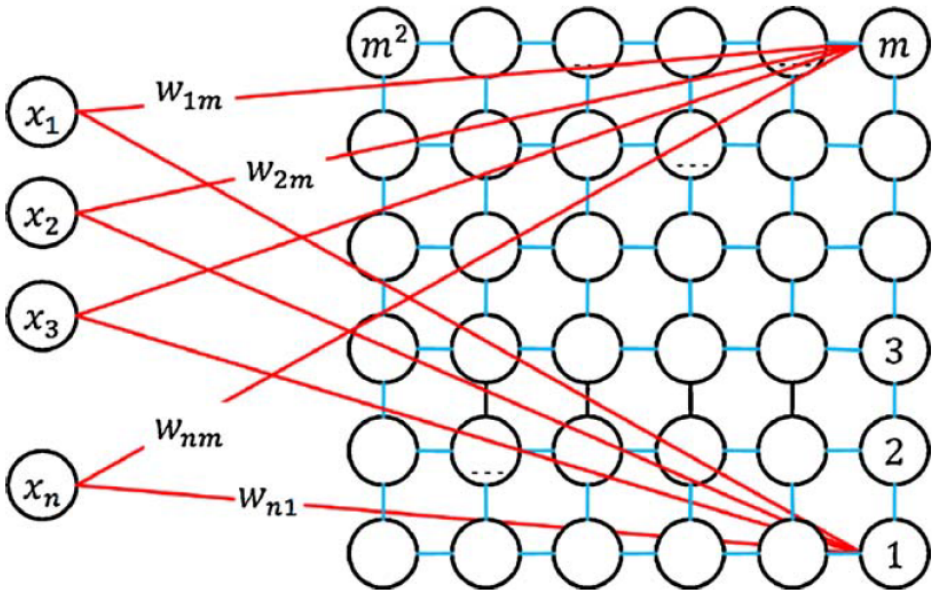
Em termos gerais, o método permite um diagnóstico eficiente em conexão com um processo, fornece uma ilustração clara de sua condição e oferece uma maneira aplicável de definir o melhor caminho para alcançar um processo mais eficiente, com melhor qualidade dos resultados. Além disso, a abordagem dos SOM é adequada para casos em que processos físicos não são bem conhecidos ou são altamente complexos, nos quais a entrada do conjunto de dados contém uma quantidade significativa de valores ausentes (JUNTUNEN *et al.*, 2013).

Como dito, a análise dos SOM é baseada em aprendizado não supervisionado. A inicialização foi utilizada para os vetores de referência preliminares, que é mais rápido e computacionalmente menos exigente que a clássica inicialização aleatória (KOHONEN, 2002). Conhecida também como a rede neural de Kohonen, esta técnica combina uma camada de entrada com uma camada competitiva de neurônios de processamento, que normalmente é organizado como uma grade bidimensional. A estrutura dos SOM é mostrada na Figura 9.

Como mostrado na Figura 9, a rede SOM é uma matriz de $\mathrm{M} = m \times m$ de neurônios artificiais para processamento. Se esses m^2 neurônios estão dispostos em uma grade ou em um plano, a rede é chamada bidimensional, já que esta mapeia vetores de entrada de alta dimensão para uma superfície bidimensional. Para uma determinada rede, o vetor de entrada x tem uma dimensão fixa n. Os n componentes do vetor de entrada $\mathbf{x}$ (ou seja, $x_1, x_2, ..., x_n$) são conectados a cada neurônio na matriz. Um peso sináptico w_{ij} é definido para uma conexão do *i-ésimo* componente do vetor de entrada o *j-ésimo* neurônio. Portanto, um vetor n-dimensional $\mathbf{w}_j$ de pesos sinápticos está associado a cada neurônio j (GHASEMINEZHAD; KARAMI, 2011).

As duas questões mais centrais do algoritmo de aprendizado de rede são: (1) o processo de adaptação do peso e (2) a ideia de uma vizinhança topológica de neurônios. A rede opera em duas fases: a fase de correspondência de similaridade e a fase de adaptação de peso. Inicialmente, o pesos são definidos para pequenos valores aleatórios e um padrão é apresentado para os nós de entrada da rede. Durante esta fase, as distâncias euclidianas entre as entradas e os pesos associados aos neurônios de saída são computados. Em seguida, a saída neurônio j,

Figura 9 – Estrutura básica da rede neural SOM



Fonte: Ghaseminezhad e Karami (2011).

que tem a distância mínima entre a saída M, é escolhido e declarado como um "vencedor". Na segunda fase, os pesos dos nós de entrada para o nó vencedor são modificados. Além disso, um bairro topológico N_j dos neurônios na vizinhança geográfica do neurônio vencedor deve ser determinado, e os pesos destes neurônios também são modificados. O algoritmo de aprendizado de rede SOM pode ser formulado conforme exposto no pseudocódigo 6.

Na Equação 4.6 pode-se visualizar a função da distância mais próxima do neurônio vencedor j.

$$D_{min}(t) = \min_i\{D_i(t)\} = \min_i \left\{\sum_j (\mathbf{x_j}(t) - \mathbf{w_{ij}}(t))^2\right\} \tag{4.6}$$

em que $\mathbf{w_{ij}}$ são os pesos que em $t = 0$ são pequenos valores gerados randomicamente, (t) é o número da iteração, e $\mathbf{x_j}$ é um vetor randomicamente selecionado dos dados de treinamento.

Na Equação 4.7 pode-se visualizar a função de atualização dos vetores de peso vencedor e de seu vizinho.

$$\mathbf{w_i}(t+1) = \mathbf{w_i}(t) + \alpha(t) \times (\mathbf{x} - \mathbf{w_i}(t)), \forall i \in N_j \tag{4.7}$$

sendo $\alpha(t)$ a função da taxa de aprendizado que decresce exponencialmente com as iterações e é calculado pela Equação 4.8.

$$\alpha(t) = \alpha_0 e^{\frac{-t}{3T}} \tag{4.8}$$

em que α_0 é a taxa de aprendizado inicial e T o número de iterações, ambos os dados definidos pelo usuário.

Na Equação 4.9 pode-se visualizar a função da ordem de vizinhança.

$$d(t) = \lfloor d_0 e^{\frac{-t}{3T}} \rfloor \tag{4.9}$$

sendo d_0 a vizinhança topológica inicial.

No pseudocódigo 6 pode-se verificar o funcionamento do algoritmo DBSCAN (NGUYEN *et al.*, 2019).

Pseudocódigo 6: SOM

```
1 início
2   configure a iteração de aprendizado t = 0 Então inicialize todos os pesos w_ij
      com pequenos valores randômicos (ou inicialize com pesos utilizando dados
      de entrada) configure a topologia de vizinhança inicial (d_0). Configure a taxa
      de aprendizado inicial (α_0) e configure o total do número de iterações (T).
3   while o número de iterações (t) é menor do que o número (T), repita as
      etapas (4) a (7) do
4     escolha um vetor de entrada x aleatoriamente no conjunto de treinamento
5     determine o neurônio j para que seu vetor de peso w_j seja o mais próximo
        do vetor de entrada e chame-o de neurônio vencedor. O neurônio j
        vencedor tem a distância mais próxima, D_min(t) para o padrão de entrada
        x(t), onde D_min(t) é dado pela equação 4.6:
6     atualize os vetores de peso do neurônio vencedor j e de seu vizinhos
        através da equação 4.7: onde α(t) é a função da taxa de aprendizado que
        diminui exponencialmente ao longo do tempo. Além do que, nós
        definimos uma função da ordem de vizinhança d(t) que diminui
        exponencialmente com o tempo. Nesso papel, as seguintes equações 4.8
        e 4.9 são usadas para α(t) e d(t): onde ⌊y⌋ denota o menor inteiro menor
        o igual a y
7     configure t = t + 1 e se t < T vá para o passo 3, e senão pare
8   end
9 fim
```

Fonte: (NGUYEN *et al.*, 2019)

4.7 OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS (*PARTICLE SWARM OPTIMIZATION* - PSO)

O algoritmo de Otimização por Enxame de Partículas (*Particle Swarm Optimization* - PSO) é mais usado, popular e famoso métodos baseado em enxames (FIGUEIREDO *et al.*, 2019). Foi inspirado no comportamento social de animais que se comportam com características de bando, simulando sua inteligência e comportamento coletivos (ALAM *et al.*, 2014). Foi introduzido em 1995 por Kennedy e Eberhart (EBERHART; KENNEDY, 1995). Além das aplicações em otimização com dados reais, o algoritmo PSO tem sido amplamente utilizado

em problemas de otimização binária, combinatória e em *clustering* (SANTOS *et al.*, 2017). Em problemas de otimização, as posições representam uma solução candidata para o problema. No entanto, em agrupamento, isto depende da codificação e dos parâmetros (NANDA; PANDA, 2014).

A principal característica do PSO é a maneira simples de compartilhar informações entre os agentes, com base em algumas equações. Aqui, um agente é chamado de partícula e um grupo deles forma um enxame. As partículas estão localizadas em um espaço de busca multidimensional e mudam de posição de acordo com a sua melhor posição alcançada até a iteração corrente (auto-experiência) e as posições do restante do enxame (experiência coletiva). Como de costume, uma função de condicionamento físico é usada para avaliar a qualidade dos agentes. Este conjunto de etapas permite o surgimento de um comportamento global complexo (SANTOS *et al.*, 2017).

O aprendizado das partículas vem de duas fontes, uma é de um própria experiência, chamada aprendizagem cognitiva, e a outra é o aprendizado combinado de todo o enxame, chamado aprendizado social. O primeiro caso é representado pela melhor posição da partícula ($\mathbf{pBest}$) até a iteração corrente e o aprendizagem social é representado pelo melhor posição alcançada considerando, por exemplo, toda a população ($\mathbf{gBest}$). Juntos, os aprendizados cognitivo e social são usados para calcular a velocidade das partículas e sua próxima posição (ALAM *et al.*, 2014).

A codificação mais comumente utilizada para clusterização particional considera uma partícula como uma solução candidata completa (FIGUEIREDO *et al.*, 2019). Neste caso, o agente conterá concatenados em um vetor as coordenadas espaciais de todos os n_k centroides. A aplicação do PSO pode ser feita de forma direta, utilizando as equações de posição e velocidade já conhecidas de acordo com as equações 4.10 e 4.11

$$\mathbf{x_i(t+1)} = \mathbf{x_i(t)} + \mathbf{v_i(t+1)} \tag{4.10}$$

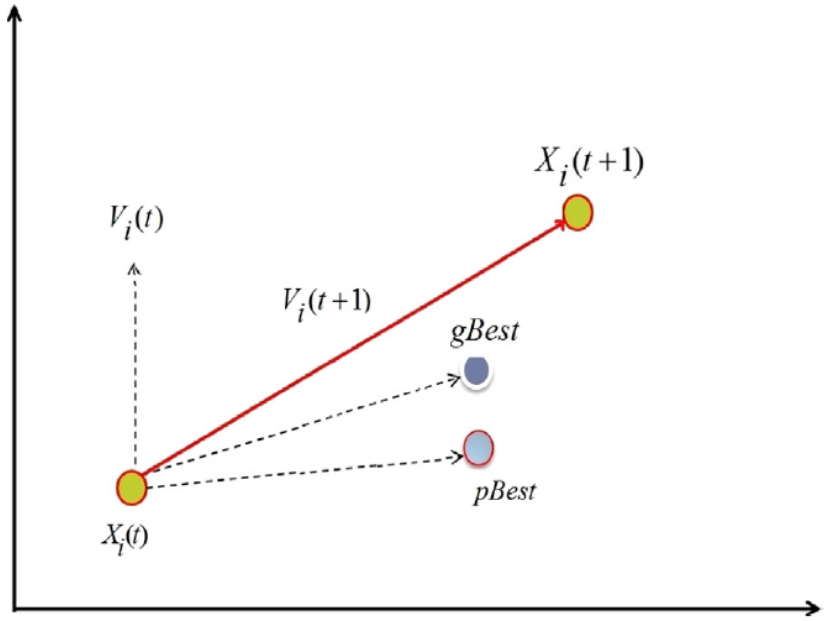
$$\mathbf{v_i(t+1)} = \omega \times \mathbf{v_i(t)} + q_1 \times r_1 \times (\mathbf{pBest_i(t)} - \mathbf{x_i(t)}) + q_2 \times r_2 \times (\mathbf{gBest(t)} - \mathbf{x_i(t)}) \tag{4.11}$$

nas quais ω é o peso de inércia, $\mathbf{x_i}$ é a posição da partícula, $\mathbf{v_i}$ é a velocidade, $\mathbf{pBest}$ é a melhor posição já encontrada pela partícula, $\mathbf{gBest}$ a melhor posição encontrada pelo grupo, q_1 e q_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 $[-\mathbf{vmax}; +\mathbf{vmax}]$ (COHEN; CASTRO, 2006).

O Figura 10 representa os vetores de deslocamento das partículas influenciado pelo pBest e o gBest.

Figura 10 – Movimentos das partículas influenciado pelo pBest e gBest (ALAM *et al.*, 2014).



Fonte: Alam *et al.* (2014)

Uma definição essencial em qualquer problema que envolva algoritmos bio-inspirados é a definição de uma função objetivo (*fitness*), a qual avalia a qualidade da resposta de uma solução candidata. Aqui a proposta utilizada foi minimizar o somatório de todos os pontos pertencentes a um determinado grupo com o seu respectivo centroide, a qual deve ser minimizada (SSW). Esta é a forma mais comumente adotada na literatura Figueiredo *et al.* (2019).

No pseudocódigo 7 pode-se verificar o funcionamento do algoritmo PSO com a função objetivo SSW.

Pseudocódigo 7: PSO

```
1  início
2      inicialize cada partícula com um número de centroides e avalie cada um deles
3      atualize a melhor posição atual de cada partícula
4      atualize a melhor partícula do enxame
5      while critério de parada não for atingido do
6          atualize velocidade e posição
7          avalie cada partícula
8          atualize melhor posição de cada partícula
9          atualize a melhor partícula do enxame
10     end
11 fim
```

Fonte: Autoria própria

Para o estudo de caso da industria automotiva em questão foi utilizada a topologia global em que as partículas recebem informações de todas as demais por ser a forma mais comum de utilização do PSO.

4.8 ALGORITMO GENÉTICO (*GENETIC ALGORITHM - GA*)

A computação evolutiva é o campo da pesquisa que extrai ideias da biologia evolutiva para desenvolver técnicas de busca e otimização com vistas a resolver problemas complexos. Os algoritmos evolucionários estão enraizados na teoria darwiniana da evolução das espécies. Darwin propôs que uma população de indivíduos capazes de se reproduzir e condicionados à variação (genética) seguida pela seleção natural resulta em novas populações de indivíduos cada vez mais adaptados ao seu ambiente. Esta proposta foi muito radical na época em que foi formalizada, no final da década de 1850, porque sugeria que um processo simples de reprodução com variação e seleção seria suficiente para produzir formas de vida complexas (CASTRO, 2006).

O algoritmo Genético (*Genetic Algorithm* - GA), é uma técnica de busca e otimização inspirada em tais principios. Há décadas o método tem sido aplicado na resolução de vários problemas em diferentes campos da ciência (PESHKO, 2007)(DING *et al.*, 2010).

Os GAs modelam a evolução genética, em que as características dos indivíduos são expressas usando genótipos. Os principais operadores são a seleção (para modelar a sobrevivência de o mais apto) e recombinação através da aplicação de um operador de *crossover* (para modelar reprodução), e a mutação. Note que apenas os dois últimos são considerados operadores genéticos.

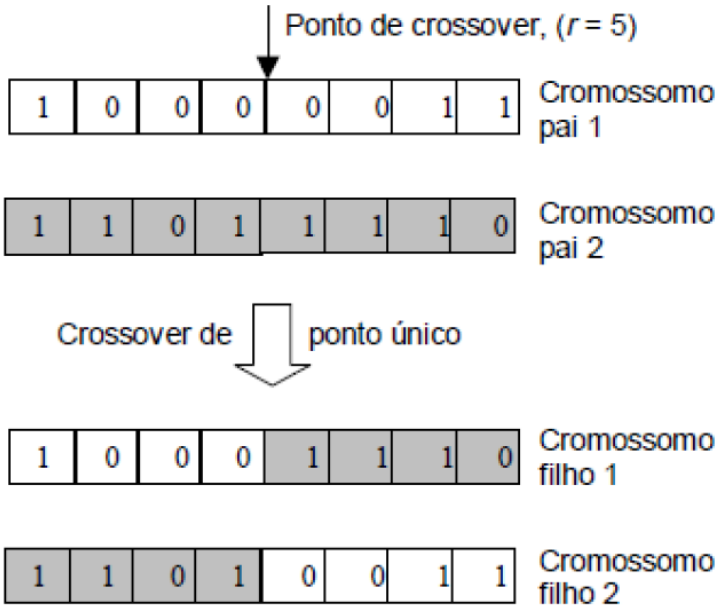
O objetivo da mutação é introduzir novo material genético em um indivíduo existente, isto é, adicionar diversidade às características genéticas da população (ENGELBRECHT, 2007). Da mesma forma que no PSO, um indivíduo, cromossomo ou agente apresenta uma solução candidata completa para determinado problema.

Em resumo, O algoritmo genético é uma meta-heurística 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 (BHATTACHARJYA, 2012).

Na Figura 11 pode-se visualizar o operador de cruzamento (*Crossover*) (CASTRO, 2006).

As diferenças entre os organismos são resultados dos processos evolutivos mutação (alteração ou desvio no material genético), recombinação ou cruzamento (troca de material genético entre cromossomos. Com exceção dos gametas, a maioria das células do mesmo euca-

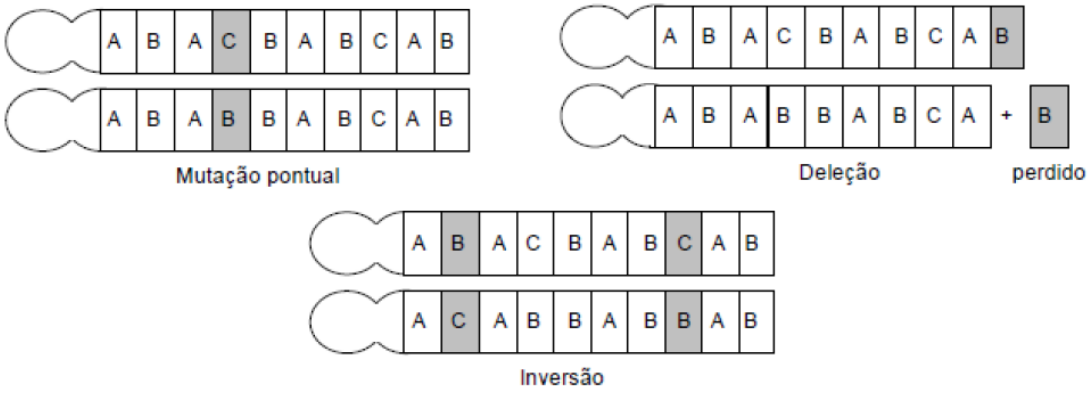
Figura 11 – Operador genético de cruzamento (*Crossing-over*)



Fonte: Adaptado de Castro (2006)

riota caracteristicamente, o organismo tem o mesmo número de cromossomos. Além disso, o organização e número de genes nos cromossomos são os mesmo de célula para célula. Essas características são iguais para todos os membros da mesma espécie. Mutações podem surgir espontaneamente ou ser induzidas por substâncias químicas ou mutagênicos por radiação. Existem vários tipos de mutação, por exemplo, mutação pontual, exclusão, translocação e inversão. Mutação pontual, deleção e inversão são ilustrado na Figura 12 (CASTRO, 2006).

Figura 12 – Operador genético de mutação



Fonte: Adaptado de Castro (2006)

Inicialmente gera-se uma população aleatória com n indivíduos. Destes, seleciona-se 2 "pais", os quais participam do *crossover*, ou troca de genes, que ocorre 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 (AIBINU *et al.*, 2016).

O pseudocódigo 8 pode-se verificar o funcionamento do algoritmo 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. Neste trabalho foi utilizado o operador genético de um ponto por ser o mais usual.

Pseudocódigo 8: GA

```
1  início
2      inicialize a população de cromossomos com um número de centroides definido
3      avaliação dos indivíduos
4      verifique critério de parada
5      while critério de parada não for atingido do
6          faça seleção dos pais
7          aplique crossover nos pais selecionado
8          aplique mutação nos filhos
9          substitua a população pelos filhos gerados
10     end
11 fim
```

Fonte: Autoria própria

4.9 EVOLUÇÃO DIFERENCIAL (*DIFFERENTIAL EVOLUTION* - DE)

A Evolução Diferencial (DE) é outra técnica inspirada na evolução das espécies. Na literatura, tem se mostrado um candidato importante para problemas de otimização e *clustering*. Neste caso, os agentes são denominados vetores e são gerados como no PSO e GA (SENTHILKUMAR; VANITHA, 2013).

As posições dos vetores fornecem informações valiosas sobre o cenário da aptidão. Desde que seja usado um bom método uniforme de inicialização aleatória para construir o população inicial, os indivíduos iniciais fornecerão uma boa representação da todo o espaço, com distâncias relativamente grandes entre os indivíduos. À medida que a busca avança, as distâncias entre os vetores diminuem, com todos convergindo (ENGELBRECHT, 2007).

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 (RAMADAS; ABRAHAM; KUMAR, 2016).

As distâncias entre os indivíduos são uma indicação adequada da diversidade da corrente população e da ordem de magnitude dos tamanhos dos passos que devem ser tomados para que a população convirja a um ponto. Se houver grandes distâncias entre agentes, é lógico que eles devem fazer grandes saltos para explorar o máximo possível do espaço de busca. Por outro lado, se as distâncias entre os indivíduos são pequenas, os tamanhos do saltos devem ser peque-

nos para explorar áreas locais. É esse comportamento que é alcançado pelo DE no cálculo da etapa de mutação, definindo tamanhos de saltos como diferenças ponderadas entre indivíduos selecionados aleatoriamente. O primeiro passo, portanto, é a mutação, realizada calculando-se um ou mais vetores de diferença e, em seguida, usando estes para determinar a magnitude e a direção dos tamanhos dos saltos (ENGELBRECHT, 2007).

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 $\mathbf{v_{i,G}}$, através da Equação 4.12:

$$\mathbf{v_{i,G}} = \mathbf{x_{r1,G}} + F \times (\mathbf{x_{r3,G}} - \mathbf{x_{r2,G}}) \tag{4.12}$$

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 seja formado. Por fim, é feita uma seleção gulosa entre $\mathbf{v_i}$ e $\mathbf{u_i}$ (SENTHILKUMAR; VANITHA, 2013).

No pseudocódigo 9 pode-se verificar o funcionamento do algoritmo DE.

Pseudocódigo 9: DE

```
1  início
2      inicialize os vetores
3      avaliação da solução
4      verifique critério de parada
5      while critério de parada não for atingido do
6          aplique diferenciação vetorial (mutação)
7          aplique recombinação vetorial (crossover), gerando o trial vector
8          aplique seleção gulosa entre o target e o trial vectors
9          valiação dos novos vetores gerados e seleção
10     end
11 fim
```

Fonte: Autoria própria

A seleção é aplicada para determinar quais indivíduos participarão da mutação para produzir um vetor de teste e determinar qual sobreviverá até a próxima geração. Com referência ao operador de mutação, vários métodos são utilizados. A seleção aleatória é geralmente usada para escolher os indivíduos dos quais os vetores de diferença são calculados. Em boa parte das implementações, o *target vetor* é selecionado aleatoriamente ou o melhor indivíduo é escolhido. Para construir a população para a próxima geração, a seleção determinística é usada: a prole substitui os progenitores se a aptidão do filho for melhor que o pai; caso contrário, o pai sobrevive para a próxima geração, garantindo a aptidão da população (ENGELBRECHT, 2007).

5 PRÉ-PROCESSAMENTO DE DADOS

O pré-processamento dos dados é uma etapa essencial para alcançar bom desempenho de algoritmos agrupamento. Esta etapa pode incluir discretização, remoção de *outliers* e ruído, integração de dados de várias fontes, transformação de dados para faixas dinâmicas comparáveis e, ainda, elaboração de uma metodologia para lidar com dados incompletos (DOUGHERTY, 2012).

Dentre estes, a normalização de dados é um passo essencial. Trata-se de uma etapa que envolve a transformação das características de cada objeto em valores que obedeçam uma distribuição normal. O principal objetivo é minimizar o impacto nos resultados daqueles dados cuja contribuição numérica é maior em classes de padrões discriminantes (GARCÍA; LUENGO; HERRERA, 2015). É muito útil para métodos de aprendizado estatístico, já que todos as variáveis contribuem igualmente para o processo de aprendizagem dentro de uma distribuição normal.

5.1 AVALIAÇÃO DE DISTRIBUIÇÃO NORMAL DOS DADOS

A normalização de dados é uma abordagem de pré-processamento em que estes são dimensionados ou transformados para induzir uma contribuição similar de cada variável, obedecendo uma curva aproximadamente normal de distribuição quando aplicado a um histograma. A importância desta tarefa foi apresentada em muitos estudos. Abordagens de seleção e ponderação de variáveis são uma tendência atual (SINGH; SINGH, 2019)(HAN; PEI; KAMBER, 2011)

Singh e Singh (2019) desenvolveram uma investigação levando em consideração a classificação de dados e o uso de métodos de normalização. Com base nisso, pode-se indicar três pontos importantes:

(a) Alguns métodos aumentam a complexidade dos dados após a sua aplicação mais do que não normalizados, prejudicando os resultados finais ;

(b) A média e o desvio padrão medidos são mais adequados para normalização em comparação ao *min-max* e mediana. Os melhores resultados foram obtidos com métodos de seleção de variáveis com ponderação ;

(c) A normalização muda a relevância das variáveis que causam resultados diferentes em termos de precisão e redução de dimensionalidade, ao trabalhar com seleção e ponderação.

Baseado nos pontos indicados para o estudo de caso da indústria automotiva, além da etapa de normalização foi aplicada uma etapa posterior de padronização para minimizar os efeitos negativos dessa etapa de pré-processamento.

5.2 PADRONIZAÇÃO DOS DADOS

No contexto de problemas de agrupamento a padronização é a etapa importante de pré-processamento para ajustar valores de variáveis ou atributos de diferentes faixas dinâmicas em uma faixa específica (MOHAMAD; USMAN, 2013).

Em geral, os algoritmos de aprendizado se beneficiam da padronização do conjunto de dados. Se alguns *outliers* estiverem presentes no conjunto, escaladores ou transformadores robustos são mais apropriados (MILLIGAN; COOPER, 1988). Desta forma, mitiga-se a distorção de resultados após aplicação dos algoritmos devido a diferença de escala das dimensões. Isto ficou muito evidente neste trabalho devido a diferença de escala de cada uma das dimensões selecionadas e o consequente impacto no resultado (detalharemos tal afirmação adiante).

Na prática, geralmente ignora-se a forma da distribuição e apenas transforma-se os dados para centralizá-los, removendo o valor médio de cada variável e, em seguida, escalonando-o dividindo as variáveis não constantes pelo desvio padrão. Por exemplo, muitos elementos usados na função objetivo de um método assumem que todas as variáveis estão centralizadas em torno de zero e têm variação na mesma ordem (MOHAMAD; USMAN, 2013). Se uma variável tiver uma variação maior que a das outras, ela poderá dominar a função objetivo e tornar o estimador incapaz de aprender com as outras variáveis corretamente (STEINLEY, 2004).

O *Z-score* é uma forma de padronização usada para transformar dimensões normais em forma de valorização padrão. Dado um conjunto de dados brutos $\mathbf{x}$, a fórmula de padronização do *Z-score* é definida como na equação 5.1:

$$z_j = Z(x_j) = \frac{x_j - \overline{x_j}}{\sigma_j} \tag{5.1}$$

em que, x_j é o dado de entrada, e $\overline{x_j}$ e σ_j são a média da amostra e o desvio padrão do *j-ésimo* atributo, respectivamente.

A dimensão transformada terá uma média de 0 e uma variação de 1. As informações de localização e escala da dimensão original foram perdidas (JAIN; DUBES, 1988).

Uma restrição importante da utilização do *Z-score* é que ele deve ser aplicada na padronização global e não apenas dentro do grupo (MILLIGAN; COOPER, 1988).

5.3 ANÁLISE DE COMPONENTES PRINCIPAIS

O estudo de caso utiliza dados de multidimensionalidade, o que pode atrapalhar o desempenho dos classificadores. Para amenizar estes impactos, é possível abordar uma técnica de redução de dimensionalidade, a Análise de Componentes Principais *Principal Components Analysis* - PCA (COVER; THOMAS, 1991) (SIQUEIRA *et al.*, 2013).

O PCA é um método baseado na projeção dos dados em um espaço de menor dimensão que caracteriza com precisão o estado do sistema. As técnicas de redução dimensional podem simplificar e melhorar significativamente os procedimentos de monitoramento de processos. Assim, a técnica produz uma representação de menor dimensão de maneira a preservar a correlação estrutural entre as variáveis. Trata-se de um método linear, sendo adequado em termos de captura da variabilidade do processo (PEARSON, 1901) (BOUHOUCHE; YAHI; BAST, 2011)(SIQUEIRA *et al.*, 2013).

O PCA transforma um espaço de dados em um subespaço menor das variáveis mais relevantes. O objetivo é projetar o espaço dimensional original *I* em um subespaço dimensional linear *I'*, onde *I'* < *I*, de modo que a variação nos dados seja explicada ao máximo dentro do subespaço transformado *I'*. Assim, componentes com pouca variação tendem a ser removidos. Os principais componentes de um conjunto são encontrados calculando a covariância (ou correlação) dos padrões e obtendo-se o conjunto mínimo de vetores ortogonais (os autovetores) que abrangem o espaço da matriz de covariância. Considerando este conjunto, qualquer vetor no espaço pode ser construído com uma combinação dos próprios (ENGELBRECHT, 2007)(SIQUEIRA *et al.*, 2013).

Assim, pode-se dizer que a primeira componente principal é a variável que leva à maior variância dos valores dimensionais, ou seja, aquela que preserva ao máximo o conteúdo a grandeza dimensional e a influência no agrupamento. A segunda componente principal é necessariamente ortogonal à primeira, e se associa ao segundo maior autovalor das variáveis estudadas. Logo, todas as componentes são ortogonais entre si e são ordenadas de acordo com seus respectivos autovalores (PEARSON, 1901)(SIQUEIRA *et al.*, 2013).

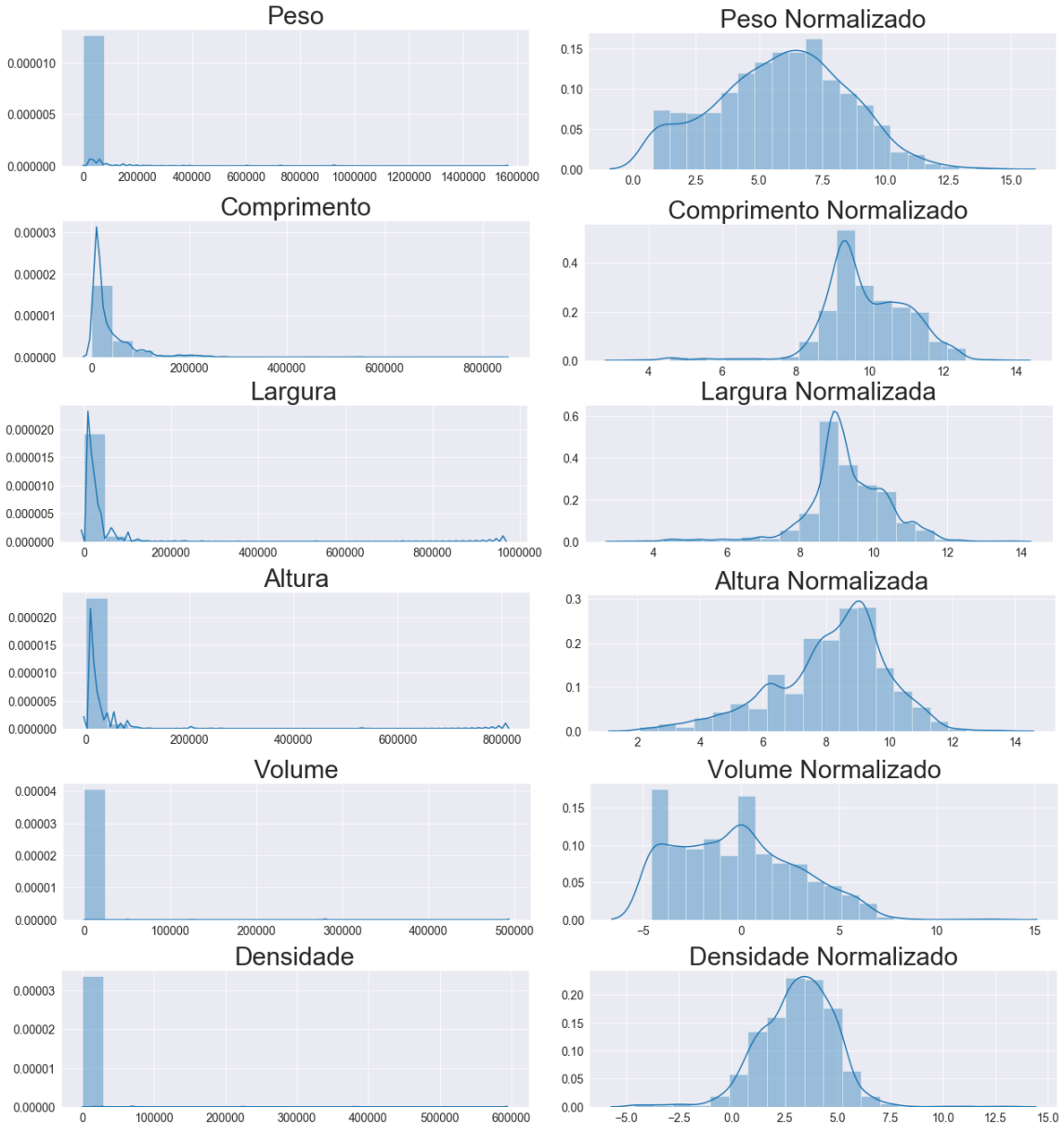
6 ESTUDO DE CASO

As etapas de pré-processamento, aplicação dos algoritmos e métricas e discussão dos resultados serão apresentados e discutidos a seguir.

6.1 NORMALIZAÇÃO DOS DADOS DA INDÚSTRIA AUTOMOTIVA

Na Figura 13 são apresentados os histogramas de cada uma das 6 dimensões selecionadas, sendo no lado esquerdo antes e lado direito depois da da normalização logarítmica.

Figura 13 – Correlação e distribuição das dimensões na base de dados da industria automotiva, antes e depois da normalização logarítmica.



Fonte: Autoria Própria

Após a transformação dos dados e das dimensões escolhidas, através da aplicação de uma escala logarítmica, observa-se uma melhor distribuição dos dimensões.

6.2 PADRONIZAÇÃO DOS DADOS APÓS NORMALIZAÇÃO

A padronização é um requisito comum para muitos estimadores de aprendizado de máquina implementados no *scikit-learn* (PEDREGOSA *et al.*, 2011). Eles podem se comportar mal se os recursos individuais não possuírem aproximadamente média zero e variância unitária. Na prática, geralmente ignora-se a forma da distribuição e apenas se transforma os dados para centralizá-los, removendo o valor médio de cada recurso e, em seguida, escalonando-o dividindo os recursos não constantes pelo seu desvio padrão.

Na Tabela 2 pode-se verificar uma análise estatística da base de dados depois da etapa de normalização para as dimensões escolhidas na base de dados da industria:

(a) Peso

(b) Comprimento

(c) Largura

(d) Altura

(e) Volume

(f) Densidade

Tabela 2 – Resumo estatístico do banco de dados normalizado da industria automotiva.

Objeto	**(a)**	**(b)**	**(c)**	**(d)**	**(e)**	**(f)**
Média	5,867	9,901	9,263	8,145	-0,222	3,098
Desvio P.	2,609	1,197	1,119	1,783	3,074	1,784
Mínimo	0,788	3,555	3,219	2,079	-4,605	-4,605
25%	4,049	9,210	8,854	7,170	-2,813	1,949
50%	6,032	9,668	9,210	8,517	-0,174	3,219
75%	7,708	10,747	9,903	9,210	1,950	4,301
Máximo	14,264	13,635	13,778	13,605	13,110	13,295

Fonte: Autoria própria.

Através da análise da Tabela 2, verifica-se que mesmo após a etapa de normalização a grandeza de cada uma das dimensões possuem uma significativa diferença quando comparamos seus valores originais, como média, desvio padrão, mínimos e máximos. Esta diferença pode ser prejudicial ao resultado final, pois tende a causar uma dominância de atração dos dados pelas dimensões com maior desvio padrão. Na Tabela 3 podemos verificar uma analise estatística depois

da etapa de normalização e padronização, descritos na seção 5.2 (PEDREGOSA *et al.*, 2011). Pode-se notar que a média tende a zero e o desvio padrão tende a 1 para todas as dimensões. Desta forma, é possível mitigar a condição em que uma das dimensões analisadas influencia o resultado.

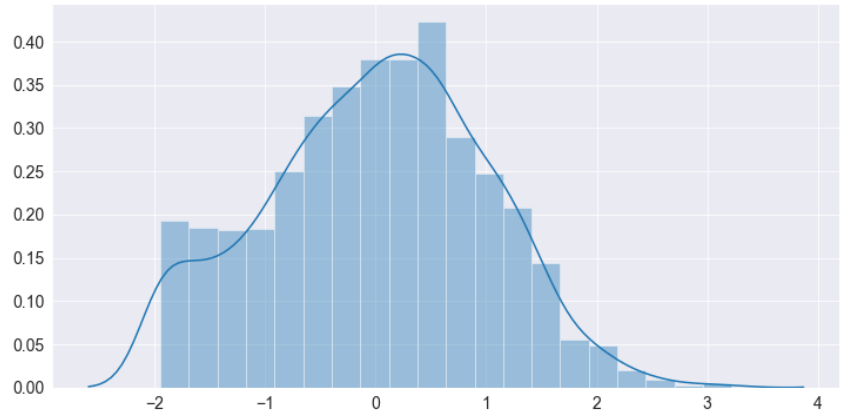
Tabela 3 – Resumo estatístico do banco de dados normalizado e padronizado da industria automotiva.

Objeto	(a)	(b)	(c)	(d)	(e)	(f)
Média	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Desvio P.	1,0002	1,0002	1,0002	1,0002	1,0002	1,0002
Mínimo	-1,9470	-5,3036	-5,4002	-3,4028	-1,4262	-4,3193
25%	-0,6971	-0,5775	-0,3659	-0,5467	-0,8432	-0,6444
50%	0,0634	-0,1952	-0,0472	0,2091	0,0155	0,0678
75%	0,7060	0,7069	0,5720	0,5980	0,7069	0,6744
Máximo	3,2192	3,1205	4,0334	3,0634	4,3381	5,7178

Fonte: Autoria própria.

A Figura 14 representa a curva de distribuição normal da dimensão Peso após aplicação de uma escala logarítmica e da padronização.

Figura 14 – Curva de distribuição normal da dimensão Peso após aplicação de escala logarítmica e da padronização.



Fonte: Autoria Própria

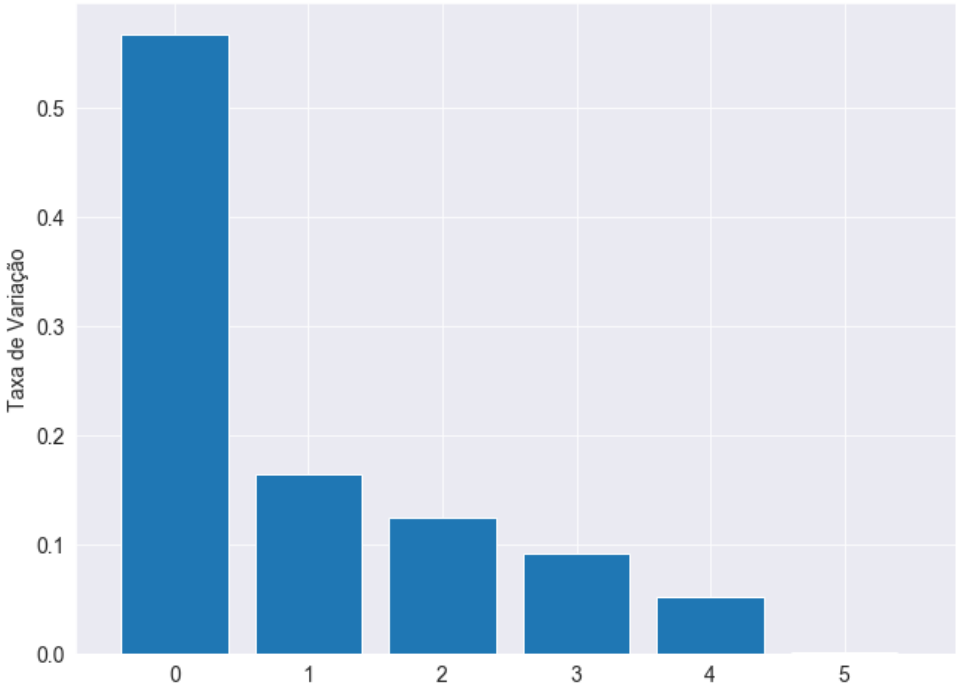
Pode-se observar que o formato da curva de distribuição permanece inalterada quando comparamos os gráficos da Figura 13 para a mesma dimensão de Peso Normalizado sem padronização com a Figura 14 que representa a dimensão Peso Normalizado e com padronização através do *zscore*.

6.3 ANÁLISE DE COMPONENTES PRINCIPAIS

Assim como apresentado na seção 5.3, para se amenizar os impactos da multidimensionalidade, foi aplicada uma técnica de redução de dimensionalidade conhecida como análise

de componentes principais (COVER; THOMAS, 1991) (SIQUEIRA *et al.*, 2013). A Figura 15 apresenta a taxa de variação das componentes principais após a aplicação do PCA nos dados normalizados e padronizado com *Z-scrore*.

Figura 15 – Aplicação do PCA no banco de dados da industria automotiva, taxa de variação.



Fonte: Autoria Própria

Pela análise dos resultados as 4 primeiras componentes principais representam uma significância na ordem de 94,72%, quantidade definida como entrada nos algoritmos aqui utilizados. Medições experimentais prévias realizadas em outras bases rotuladas, valores entre 90% a 98% causam pouca influência na qualidade dos grupos formados. Com isso pode-se reduzir de 6 para 4 as dimensões, diminuindo significativamente a quantidade de operações matemáticas realizadas por cada algoritmo.

6.4 DEFINIÇÃO DA QUANTIDADE DE RODADAS E CENTROIDES PARA ALGORITMOS ESTOCÁSTICOS

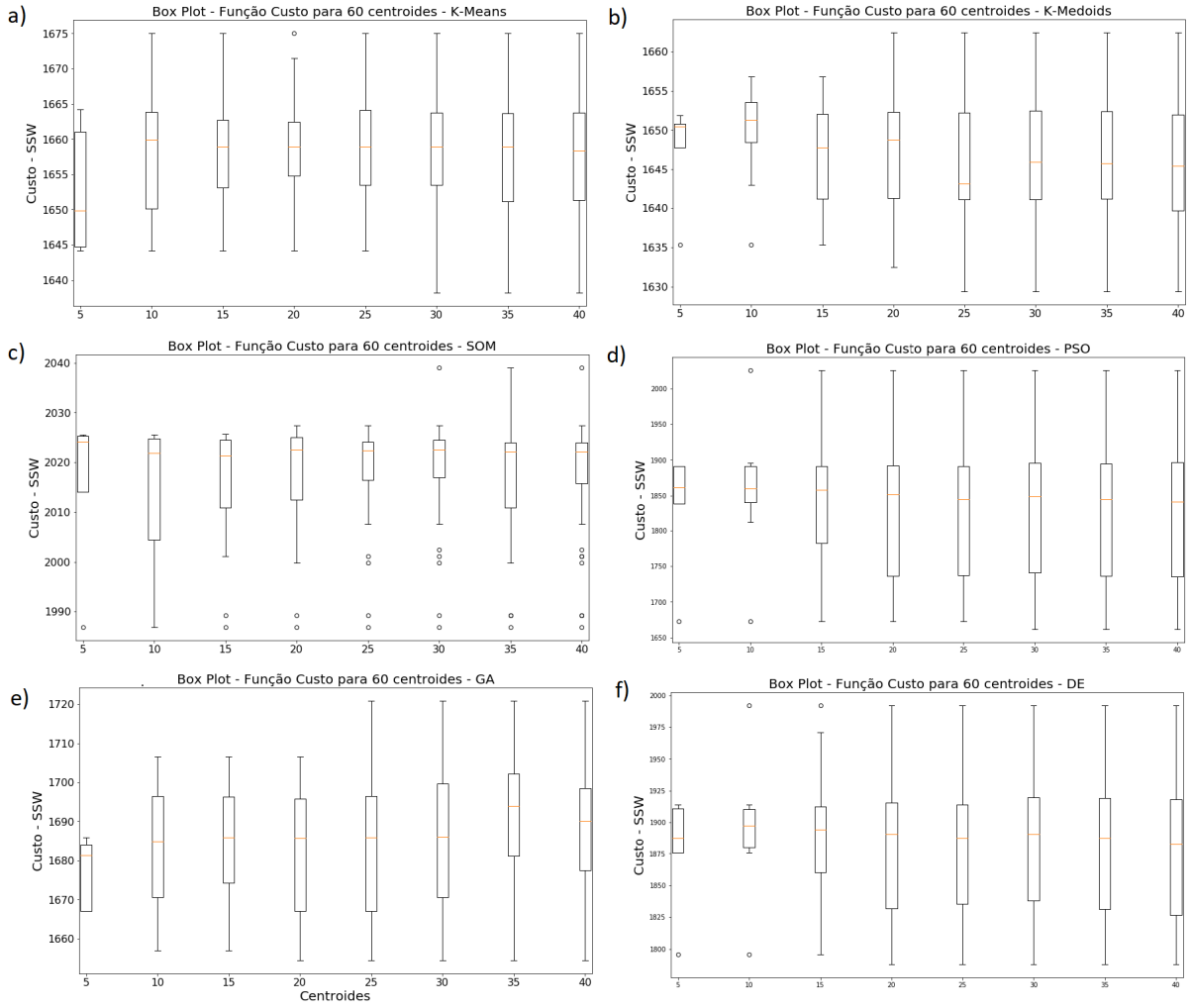
Um ponto a ser definido antes da quantidade de rodadas para os algoritmos estocásticos é a definição da quantidade de centroides, pois este é um parâmetro de entrada necessário para todos os métodos. No início da seção 6.1 apresentou-se que a base original da industria possui dados categóricos, como classe de engenharia e classificação NCM (Nomenclatura Comum do Mercosul). Estes dados podem ser considerados potenciais rótulos, mas na prática apresentam uma grande divergência. No caso da NCM são 200 códigos e no caso da classe de engenharia

são 621. É sabido pela indústria que alguns dos códigos da classe de engenharia poderiam ser agrupados sem prejuízo à análise.

Conforme exposto na seção 2.6 foi feita uma varredura do número de centroides de 60 a 860. Para análise de dispersão através da plotagem por *boxplot*, selecionou-se 60 centroides, pois é esperada a mesma dispersão para diferentes quantidades.

Assim como exposto anteriormente a literatura, é consistente em afirmar que o número ideal, de acordo com CASELLA e BERGER (2011) respeitando o teorema do limite central, é realizar 30 rodadas de cada algoritmo. Entretanto, como este estudo é direcionado a um banco de dados real, decidiu-se por se fazer uma análise empírica utilizando-se como entrada 60 centroides, variando-se a quantidade de rodadas de 5, 10, 15, 20, 25, 30, 35 e 40 e posteriormente plotando a dispersão dos resultados em gráficos *boxplot* (Na Figura 16), para os seguintes algoritmos:*K-Means*, *K-Medoids*, SOM com 100 iterações em cada rodada, PSO, GA e DE. EStes útimos consideram 1000 iterações em cada rodada.

Figura 16 – *Boxplot* aplicado no resultado do SSW para: (a) *K-Means*, (b) *K-Medoids*, (c) Mapas Auto-Organizáveis (Self Organizing Maps - SOM), (d) Otimização por Enxame de Partículas (Particle Swarm Optmization - PSO), (e) Algoritmo Genético (Genetic Algorithm - GA), (f) Evolução Diferencial (Differential Evolution - DE).



Fonte: Autoria Própria

Pela análise da dispersão de resultados pode-se observar que para o PSO e a DE a estabilização da dispersão já ocorre com 15 rodadas quando analisados os limites superiores e inferiores. Já para o GA e *K-Medoids*, essa estabilização ocorre a partir de 20 rodadas, enquanto para o *K-Means* e SOM são necessárias 30 rodadas. Portanto, para os algoritmos estocásticos desta pesquisa iremos estabelecer a quantidade de 30 rodadas para obtenção dos melhores resultados.

6.5 PARÂMETROS DE ENTRADAS DOS ALGORITMOS DE CLUSTERIZAÇÃO

Nesta seção serão apresentados os parâmetros de entrada para a aplicação dos algoritmos de *clustering* descritos no capítulo 4 no referido banco da dados da industria automotiva após a normalização, padronização (*Z-score*) e uso do PCA.

Realizou-se uma varredura do número de centroides k conforme definido na seção 2.6, em que foi executado 30 vezes os respectivos algoritmos estocásticos: *K-Means*, *K-Medoids*, Mapas Auto-Organizáveis (*Self Organizing Maps* - SOM), Otimização por Enxame de Partículas (*Particle Swarm Optmization* - PSO), Algoritmo Genético (*Genetic Algorithm* - GA), Evolução Diferencial *(Differential Evolution* - DE), para cada quantidade de centroide pré estabelecida, e apenas 1 vez para os algoritmos determinísticos: Hierarquico, DBSCAN e *Fuzzy C-Means*. Foram utilizados vetores com 4 dimensões ou as 4 principais componentes resultantes após aplicação do PCA.

Os resultados são relativos as métricas descritas no capítulo 3: Soma dos Erros Quadráticos (*Sum of Squared Errors* - SSE), Soma das Distâncias Internas (*Sum of Squares Within Clusters* - SSW), Soma das Distâncias Externas (*Sum of Squares Between Clusters* - SSB), (*Calinski-Harabasz* - CH), o índice WB e *Silhouette* - SI.

Nos algoritmos *K-Means*, *K-Medoids*, *Fuzzy C-Means* e Hierárquico o único parâmetro de entrada a ser fornecido é o número de centroides. Para o algoritmo DBSCAN, alguns parâmetros adicionais precisam ser definidos. Baseado em várias simulações e testes experimentais, definiu-se o Limiar de densidade $minPts$ = 1 e Raio ε = [0.847, 0.580, 0.4856, 0.4248, 0.3855, 0.3452, 0.3136, 0.29118, 0.268].

A rede neural SOM teve parâmetros definidos a partir de experimentação, sendo: malha de neurônios = [6x10, 16x10, 26x10, 36x10, 46x10, 56x10, 66x10, 76x10, 86x10], taxa de aprendizado inicial = 0,01 e quantidade de iterações por rodada = 100.

As metaheurísticas de otimização utilizadas (PSO, GA e DE) possuem uma quantidade maior de parâmetros de entrada a serem definidos. No caso do PSO os melhores resultados na aplicação das equações 4.10 e 4.11 foram obtidos através dos seguintes parâmetros:

- ω = 1 (peso de inércia);

- r_1 e r_2 dois números aleatoriamente gerados no intervalo $[0, 1]$;
- q_1 = 2 (coeficiente de aprendizado cognitivo - da partícula);
- q_2 = 2 (coeficiente de aprendizado social - da população);

A seguir é apresentado os parâmetros de entrada experimentais que obtiveram melhor desempenho na aplicação do GA:

- taxa de *crossover* = 80%;
- *crossover* de um ponto (simples), por ser o mais comumente utilizado;
- taxa de mutação = 30%, esta elevada taxa é um indicativo de operador de *crossover* não ser o mais adequado para esta aplicação;
- método de seleção da nova geração realizado por roleta viciada.

Da mesma forma como obtido para PSO e GA são apresentados abaixo os parâmetros de entradas específicos do algoritmo de Evolução Diferencial:

- taxa de *crossover* = 20%;
- *crossover* binário;
- limite inferior do fator de escala = 20%;
- limite superior do fator de escala = 80%.

Para os três algoritmos, considerou-se 10000 iterações por rodada, 30 rodadas para cada quantidade de centroides, população de 300 agentes e inicialização da população utilizando medoides. Chegou-se a estes números através da simulação de diferentes cenários e combinações de parâmetros e da convergência de resultados.

6.6 RESULTADOS OBTIDOS E PLOTAGEM DAS MÉTRICAS DE AVALIAÇÃO

Nesta seção são apresentados os resultados obtidos pela aplicação dos algoritmos acima expostos nas métricas selecionadas. Nesta dissertação foram utilizadas técnicas de *clustering* em dados numéricos não categóricos. Em outras palavras, dados que possuem uma ordem de grandeza em suas dimensões, dados quantitativos. Outro ponto importante de salientarmos é ausência de rótulos na base fornecida pela referida indústria automotiva.

A Tabela 4 mostra um panorama geral de resultados alcançados na métrica Soma dos Erros Quadráticos (*Sum of Squared Errors* - SSE), para os algoritmos *K-Means*, *K-Medoids*,

Fuzzy C-Means - FCM, Hierarquico, Agrupamento por Densidade Espacial em Aplicações com Ruido (*Density Based Spatial Clustering of Applications with Noise* - DBSCAN), Mapas Auto-Organizáveis (*Self Organizing Maps* - SOM), Otimização por Enxame de Partículas (*Particle Swarm Optmization* - PSO), algoritmo Genético (*Genetic Algorithm* - GA), Evolução Diferencial (*Differential Evolution* - DE), aplicado ao banco de dados da industria automotiva normalizado, padronizado (*Z-score*) e com Analise dos Componentes Principais (*Principal Components Analysis* - PCA).

Tabela 4 – Resultados de SSE aplicando os algoritmos: *K-means*, *K-medoids*, FCM, Hierárquico, DBSCAN, SOM, PSO, GA, DE.

algoritmo	60	160	260	360	460	560	660	760	860	Média
K-medoids	1206	555	351	243	179	135	104	81	63	324
Hierarchical	1343	590	358	241	174	130	98	75	59	341
K-means	1236	597	408	308	242	195	157	135	109	376
GA	1325	705	506	388	312	253	209	184	153	448
PSOC	1378	733	526	404	325	263	217	192	159	466
DE	1431	762	546	419	337	273	226	199	165	484
SOM	1861	990	710	545	438	355	293	259	215	629
FCM	2435	2048	1867	1658	1637	1450	1278	1233	1136	1638
DBSCAN	13055	11370	9502	8087	7053	6535	4969	3948	3300	7535
Média	2808	2039	1642	1366	1189	1065	839	700	595	

Fonte: Autoria própria.

A Tabela 4 mostra em ordem crescente pela coluna média dos resultados de toda varredura de centroides realizada para cada algoritmo. Nota-se uma vantagem de performance dos algoritmos *K-Medoids*, Hierárquico e *K-Means*. Ressalta-se que quanto menor o valor do SSE mais coeso são os grupos formados, tendo esta métrica que ser minimizada. A Figura 17 mostra a evolução dos resultados do SSE.

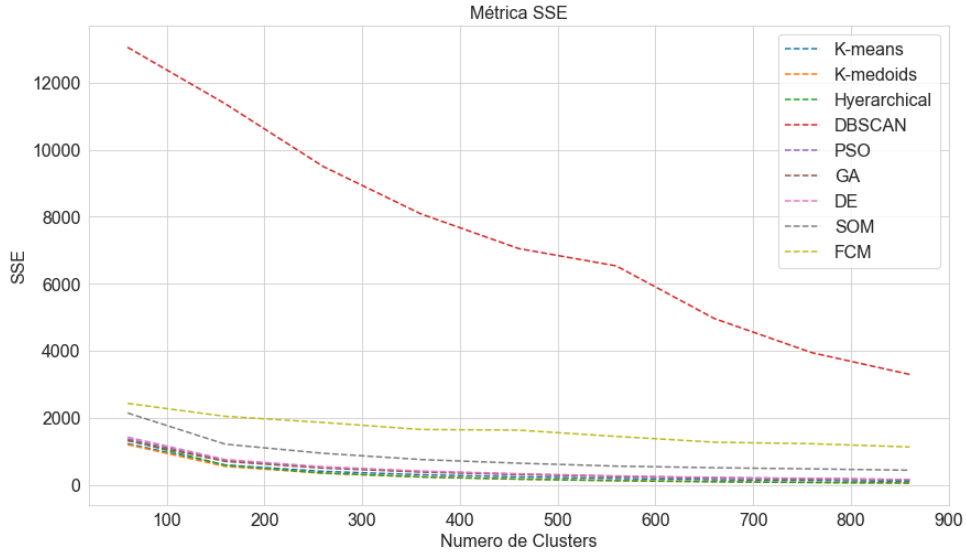
Através da análise gráfica nota-se uma clara desvantagem do algoritmo DBSCAN. Isso se deve ao fato dos parâmetros de entrada, os quais não devem ser tradados como ruído. Para tanto, mesmo que um referido dado fique isolado do raio pré-definido, este deve ser considerado como um grupo. Por conta da distribuição dos dados no espaço dimensional, muitos itens caíram nesta situação. Na análise dos grupos, ficou nítida a grande quantidade de um único item dentro de vários *clusters* gerados pelo DBSCAN.

Também analisando o gráfico percebe-se que o algoritmo *Fuzzy C-means* - FCM não obteve uma boa performance se comparado aos demais algoritmos. Novamente, pode-se atribuir este resultado a distribuição dos dados de entrada, que levou o algoritmo a cair em um mínimo local, prejudicando o resultado final.

Outro ponto a salientar é um significativo destaque positivo, o resultado das curvas para os algoritmos *K-Medoids*, Hierárquico e *K-Means* com um distanciamento das demais curvas e a correspondente proximidade entre elas.

A Tabela 5 mostra as melhores de 30 simulações para cada algoritmo estocástico, e

Figura 17 – Evolução do resultado da aplicação dos algoritmos estudados utilizando SSE variando a quantidade de centroides.



Fonte: Autoria Própria

a única rodada dos algoritmos determinísticos, considerando o Soma das Distâncias Internas - SSW.

Tabela 5 – Resultados de SSW aplicando os algoritmos: *K-means*, *K-medoids*, FCM, Hierárquico, DBSCAN, SOM, PSO, GA, DE.

algoritmo	**60**	**160**	**260**	**360**	**460**	**560**	**660**	**760**	**860**	**Média**
K-medoids	1636	1103	877	724	616	529	459	400	350	744
Hierarchical	1726	1137	888	726	613	528	456	396	346	757
K-means	1639	1120	906	775	677	603	540	484	438	798
GA	1758	1323	1123	978	875	782	717	662	613	981
PSOC	1829	1376	1168	1017	910	813	745	689	637	1020
DE	1899	1429	1213	1057	945	844	774	715	662	1060
SOM	1987	1469	1257	1131	1046	978	934	887	855	1172
FCM	1918	1644	1517	1393	1323	1224	1150	1066	987	1358
DBSCAN	5282	4793	4173	3671	3320	3084	2549	2191	1919	3443
Média	2186	1710	1458	1275	1147	1043	925	832	756	

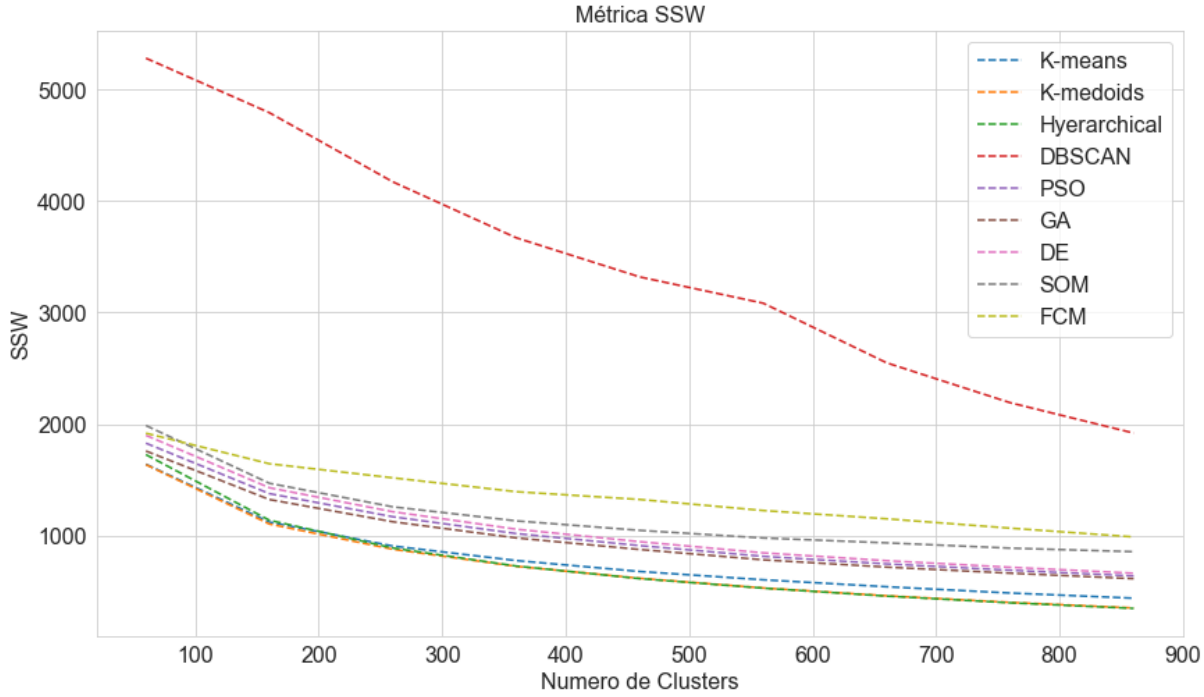
Fonte: Autoria própria.

Assim como na Tabela do SSE, o Tabela do SSW apresentada mostra os resultados em ordem crescente pela coluna de média. Quando se analisa as equações 3.1 e 3.2 é esperado um comportamento similar das curvas entre as duas métricas o que pode ser comprovado pelos valores encontrados na Tabela. Novamente nota-se uma nítida vantagem de resultados dos algoritmos *K-Medoids*, Hierárquico e *K-Means*, como esperado, já que estes valores são proporcionais ao SSE. Esta métrica também deve ser minimizada para um melhor resultado.

A Figura 18 mostra a evolução dos resultados do SSW a melhor rodada de todos so algoritmos acima citados.

Da mesma forma como observado na curva do SSE para a curva da métrica SSW nota-se

Figura 18 – Evolução do resultado da aplicação dos algoritmos estudados utilizando SSW variando a quantidade de centroides.



Fonte: Autoria Própria

um vantagem significativa dos algoritmos *K-Medoids*, Hierárquico e *K-Means*, e uma desvantagem dos algoritmos DBSCAN e FCM, pelos mesmos motivos já discorrido.

A Tabela 6 mostra as melhores de 30 simulações para cada algoritmo estocástico, e a única rodada dos algoritmos determinísticos, considerando o SSB.

Tabela 6 – Resultados de SSB aplicando os algoritmos PSOC, FCM, *K-means*, DE, GA, DBSCAN, *K-medoids*, *Hierarchical*, SOM - unidade (x10^3).

algoritmo	**60**	**160**	**260**	**360**	**460**	**560**	**660**	**760**	**860**	**Média**
DBSCAN	20	127	306	569	895	1288	1752	2294	2886	1126
K-medoids	14	106	274	518	836	1228	1692	2210	2794	1075
Hierarchical	14	100	266	505	818	1193	1654	2173	2759	1053
K-means	14	97	251	471	768	1119	1552	2058	2629	995
GA	13	91	248	459	744	1097	1520	1985	2543	967
PSOC	13	88	238	441	714	1053	1459	1905	2441	928
DE	12	84	228	423	684	1009	1398	1826	2339	889
SOM	11	81	208	400	653	953	1277	1624	2046	806
FCM	10	71	189	362	586	870	1191	1626	2096	778
Média	14	94	245	461	744	1090	1500	1967	2504	

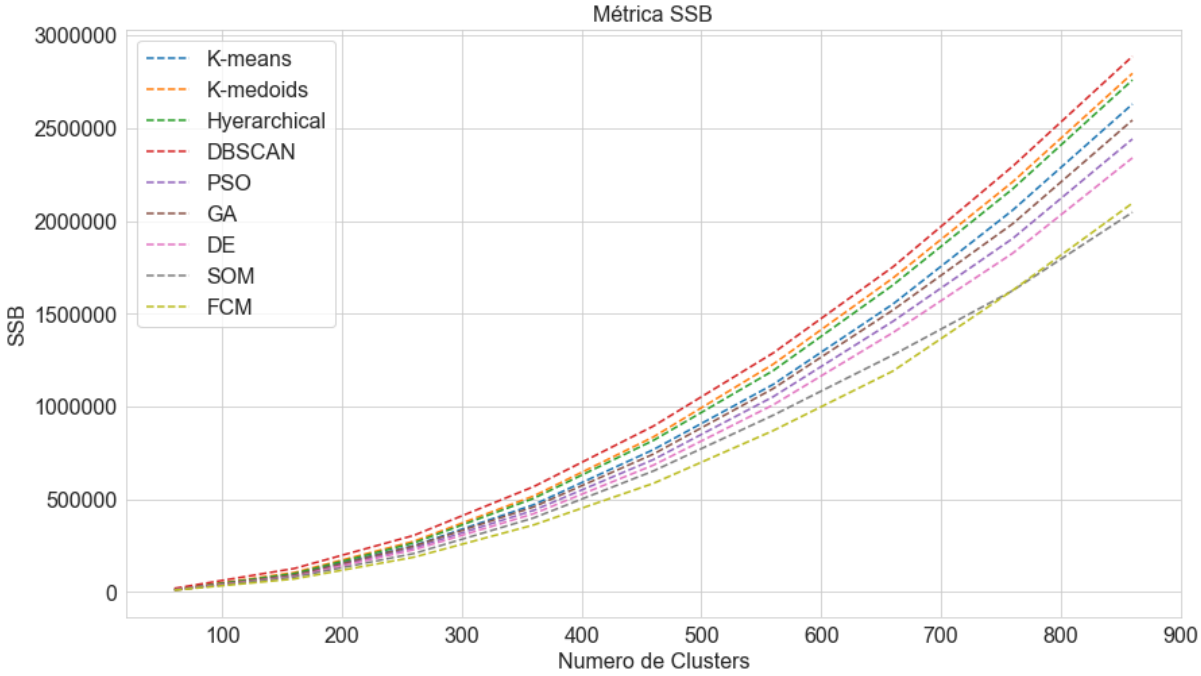
Fonte: Autoria própria.

Na Tabela 6 os resultados são apresentados em ordem decrescente para a coluna da média. Nota-se que os melhores são dos algoritmos: DBSCAN, *K-Medoids*, Hierárquico e *K-Means*.

A Figura 19 mostra a evolução dos resultados do SSB para as diferentes quantidades

de centroides.

Figura 19 – Evolução do resultado da aplicação dos algoritmos estudados utilizando SSB variando a quantidade de centroides.



Fonte: Autoria Própria

Importante salientar que quanto maior o SSB, melhor delineados ficam os grupos formados, pois significam que os centroides estão afastados. Em outras palavras, os algoritmos de *clustering* buscam maximizar o SSB.

A vantagem do DBSCAN é explicada pelo mesmo motivo da desvantagem encontrada no SSW para este algoritmo, ou seja, muitos centroides acabaram sendo formados com elementos únicos o que neste caso, aumentou o somatório das distancias euclidianas entre os grupos formados.

Na outra extremidade da mesma figura nota-se uma desvantagem do delineamento dos grupos formados na aplicação dos algoritmos SOM e FCM.

A Tabela 7 resume os resultados alcançados na métrica CH (*Calinski-Harabasz*).

A ordem de desempenho destaca o *K-Medoids*, *Hierarchical* e *K-Means*. Lembrando que quanto maior o CH, melhor a qualidade dos *clusters*, ou sseja, mais coesos e delineados são os grupos formados. Esta métrica é calculada pela equação 3.4, que representa uma relação entre SSB, SSW, quantidade de dados (n) e o número de centroides (k). Desta forma, o equacionamento de uma só vez a correlação na busca de maximizar o SSB e minimizar o SSW.

Assim como observado no SSB, o CH apresenta uma curva em que os algoritmos de clusterização buscam a maximização dos seus índices.

A Figura 20 mostra a evolução dos resultados do CH em relação a quantidade de grupos.

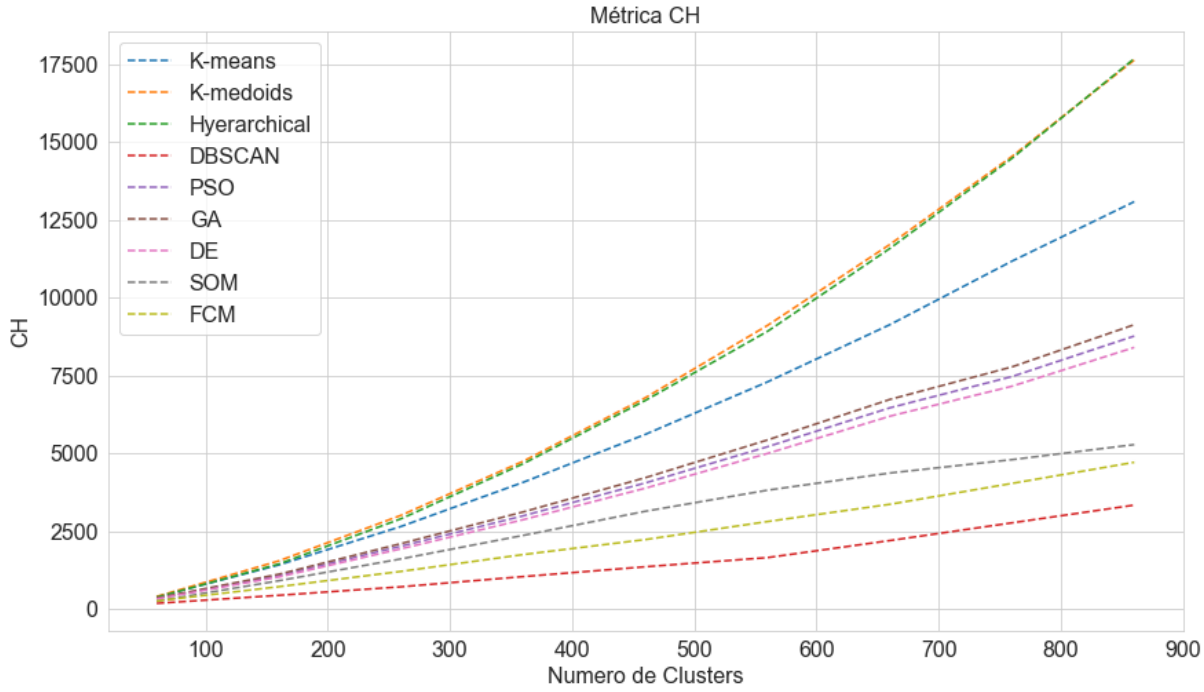
Apesar de ser mostrado uma correlação entre SSW e SSB, nesta métrica ainda não fica evidente o ponto de inflexão da curva, o que pode indicar que aumentos na quantidade de cen-

Tabela 7 – Resultados de CH aplicando os algoritmos PSOC, FCM, *K-means*, DE, GA, DBSCAN, *K-medoids*, *Hierarchical*, SOM.

algoritmo	**60**	**160**	**260**	**360**	**460**	**560**	**660**	**760**	**860**	**Média**
K-medoids	394	1539	3004	4742	6783	9108	11706	14537	17630	7716
Hierarchical	371	1439	2893	4662	6696	8917	11577	14476	17678	7635
K-means	382	1408	2646	4065	5605	7293	9129	11171	13081	6087
GA	344	1111	2095	3112	4213	5426	6726	7773	9128	4436
PSOC	330	1066	2011	2988	4044	5209	6457	7462	8763	4259
DE	316	1022	1927	2863	3876	4992	6188	7152	8398	4082
SOM	244	901	1603	2359	3134	3811	4362	4792	5275	2942
FCM	250	710	1202	1742	2224	2804	3357	4029	4707	2336
DBSCAN	175	433	706	1035	1350	1645	2192	2762	3331	1514
Média	312	1070	2010	3063	4214	5467	6855	8239	9777	

Fonte: Autoria própria.

Figura 20 – Evolução do resultado da aplicação dos algoritmos estudados utilizando CH variando a quantidade de centroides.



Fonte: Autoria Própria

troides podem ainda ser feitos até a estabilização dos resultados. Desta forma, as curvas não permitem que seja definida a quantidade de centroides a partir da qual haja uma certa estabilização da métrica.

A Tabela 8 mostra os resultados para o índice WB.

Tabela 8 – Resultados de WB aplicando os algoritmos PSOC, FCM, *K-means*, DE, GA, DBSCAN, *K-medoids*, *Hierarchical*, SOM.

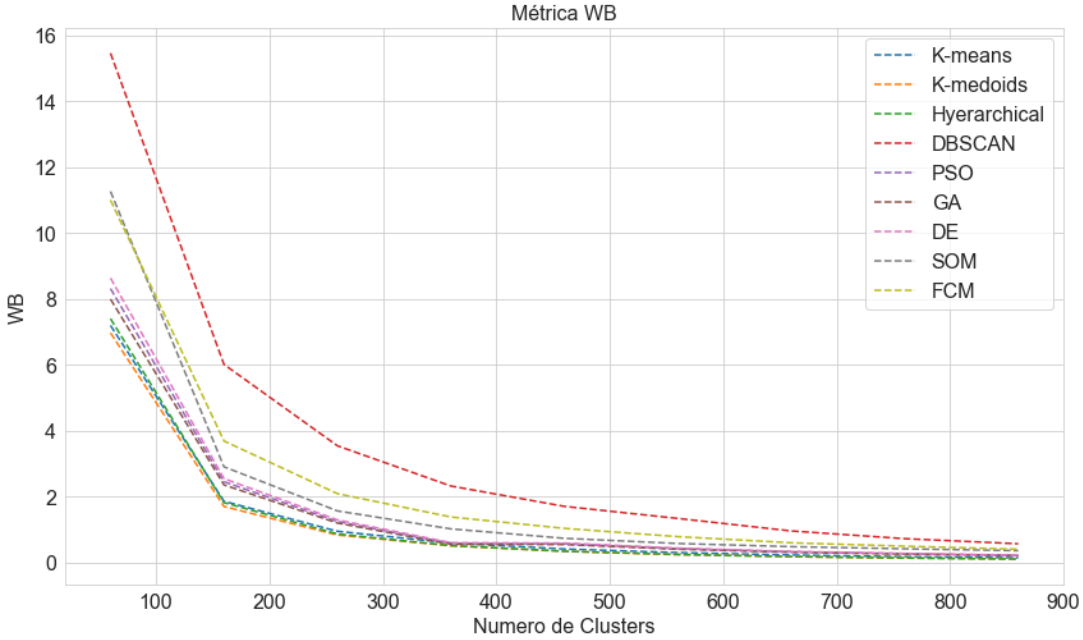
algoritmo	**60**	**160**	**260**	**360**	**460**	**560**	**660**	**760**	**860**	**Média**
K-medoids	6,973	1,703	0,837	0,509	0,341	0,243	0,180	0,138	0,108	1,226
Hierarchical	7,406	1,821	0,869	0,517	0,345	0,248	0,182	0,139	0,108	1,293
K-means	7,200	1,862	0,950	0,593	0,412	0,303	0,231	0,180	0,146	1,320
GA	7,999	2,360	1,200	0,560	0,548	0,407	0,313	0,258	0,209	1,539
PSOC	8,319	2,454	1,248	0,582	0,570	0,423	0,326	0,269	0,217	1,601
DE	8,639	2,549	1,296	0,604	0,592	0,440	0,339	0,279	0,226	1,663
SOM	11,271	2,908	1,569	1,023	0,737	0,580	0,483	0,419	0,362	2,150
FCM	11,006	3,692	2,093	1,384	1,038	0,788	0,601	0,498	0,405	2,390
DBSCAN	15,466	6,017	3,546	2,324	1,707	1,340	0,960	0,726	0,572	3,629
Média	9,364	2,818	1,512	0,900	0,699	0,530	0,402	0,323	0,261	

Fonte: Autoria própria.

Observa-se melhores resultados para *K-Medoids*, *Hierarchical* e *K-Means*, respectivamente. Tal métrica precisa ser minimizada e é calculada pela equação 3.5, que representa uma relação entre SSW, SSB, e o número de centroides (k), similar ao que acontece com o CH.

A Figura 21 mostra a evolução dos resultados do WB.

Figura 21 – Evolução do resultado da aplicação dos algoritmos estudados utilizando WB variando a quantidade de centroides.



Fonte: Autoria Própria

Nesta métrica, nota-se o único caso em que ficou relativamente evidente o ponto de inflexão da curva, com o número de 360 de centroides.

Já a Tabela 9, mostra os resultados para o índice SI.

Tabela 9 – Resultados de SI aplicando os algoritmos PSOC, FCM, *K-means*, DE, GA, DBSCAN, *K-medoids*, *Hierarchical*, SOM.

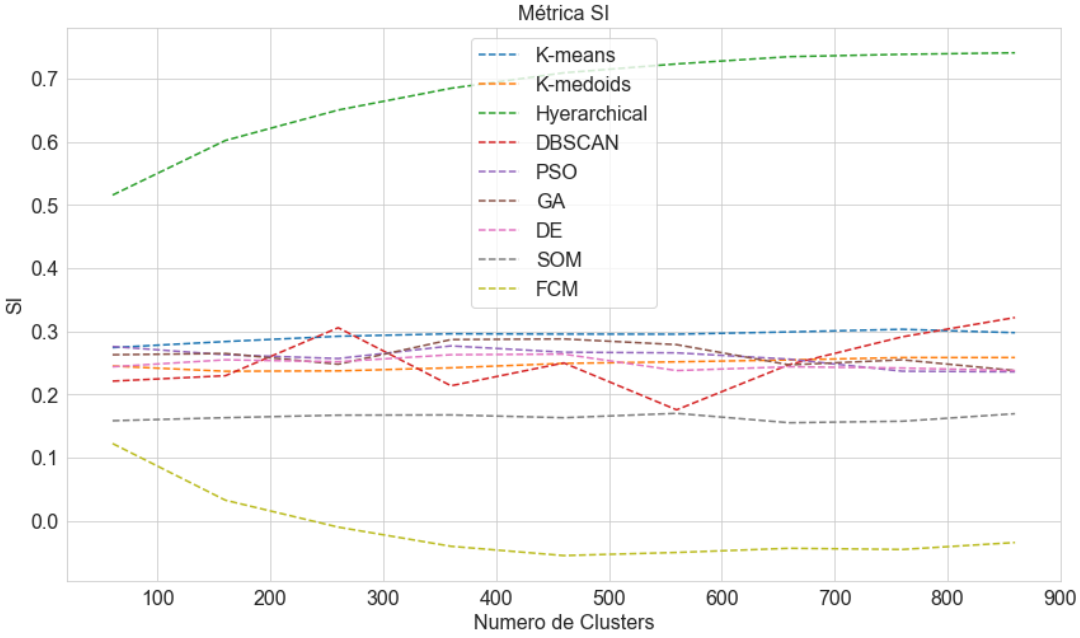
algoritmo	**60**	**160**	**260**	**360**	**460**	**560**	**660**	**760**	**860**	**Média**
Hierarchical	0,516	0,602	0,650	0,685	0,709	0,723	0,735	0,738	0,741	0,678
K-means	0,274	0,284	0,292	0,296	0,296	0,296	0,299	0,303	0,298	0,293
GA	0,263	0,265	0,248	0,287	0,288	0,279	0,247	0,255	0,238	0,263
PSO	0,276	0,263	0,257	0,277	0,267	0,266	0,256	0,237	0,236	0,259
DBSCAN	0,221	0,230	0,306	0,214	0,250	0,176	0,248	0,291	0,322	0,251
DE	0,244	0,255	0,252	0,263	0,264	0,238	0,244	0,242	0,238	0,249
K-medoids	0,245	0,237	0,237	0,242	0,249	0,252	0,255	0,259	0,259	0,248
SOM	0,159	0,163	0,167	0,168	0,163	0,170	0,155	0,158	0,170	0,164
FCM	0,123	0,033	-0,010	-0,040	-0,055	-0,050	-0,043	-0,045	-0,034	-0,013
Média	0,258	0,259	0,267	0,266	0,270	0,261	0,266	0,271	0,274	

Fonte: Autoria própria.

Pode-se destacar o algoritmo Hierárquico em relação aos demais, com valores crescentes assim que se aumenta a quantidade de centroides. Na outra extremidade pode-se observar o FCM com valores inferiores, o que fica mais visível a partir da plotagem a seguir.

A Figura 22 ilustra a evolução dos resultados do SI em relação a quantidade de grupos.

Figura 22 – Evolução do resultado da aplicação dos algoritmos estudados utilizando SI variando a quantidade de centroides.



Fonte: Autoria Própria

Assim como discorrido na revisão bibliográfica, no caso da métrica *Silhouette* o valor fica compreendido entre $[-1, 1]$ sendo que quanto mais próximo de 1, mais precisa é a formação dos grupos. Observa-se uma vantagem do algoritmo Hierárquico e um pior desempenho para o FCM. Os demais algoritmos apresentando um comportamento mais estabilizado em relação a esta métrica, obtendo valores intermediários. Nota-se uma grande diferença de resultados e uma definição da quantidade ideal de centroides inconclusiva.

6.7 DISCUSSÃO

No sentido de ser explorada outras condições de varredura. Baseado na escolha dos 3 melhores resultados globais, como pode ser observado na Tabela 10, foram escolhidos os algoritmos Hierárquico, *K-Medoids* e *K-Means* para esta exploração.

A pontuação da Tabela 10 foi elaborada através da colocação de cada algoritmo nas tabelas 4, 5, 6, 7, 8 e 9, em que para o primeiro lugar foi atribuído 9 pontos, para o segundo 8, e assim em diante até o último colocado que recebeu 1 ponto.

Tabela 10 – Pontuação baseada nos resultados do algoritmos e métricas.

algoritmo	SSE	SSW	SSB	CH	WB	SI	Total
Hierarchical	8	8	7	8	8	9	48
K-medoids	9	9	8	9	9	3	47
K-means	7	7	6	7	7	8	42
GA	6	6	5	6	6	7	36
PSO	5	5	4	5	5	6	30
DE	4	4	3	4	4	4	23
DBSCAN	1	1	9	1	1	5	18
SOM	3	3	2	3	3	2	16
FCM	2	2	1	2	2	1	10

Fonte: Autoria própria.

De forma empírica as seguintes condições de varredura de centroides foram aplicadas:

(a) varredura de 60 a 860 centroides com incremento de 100 em 100;

(b) varredura de 10 a 1410 centroides com incremento de 175 em 175;

(c) varredura de 160 a 1760 centroides com incremento de 200 em 200;

(d) varredura de 30 a 910 centroides com incremento de 110 em 110;

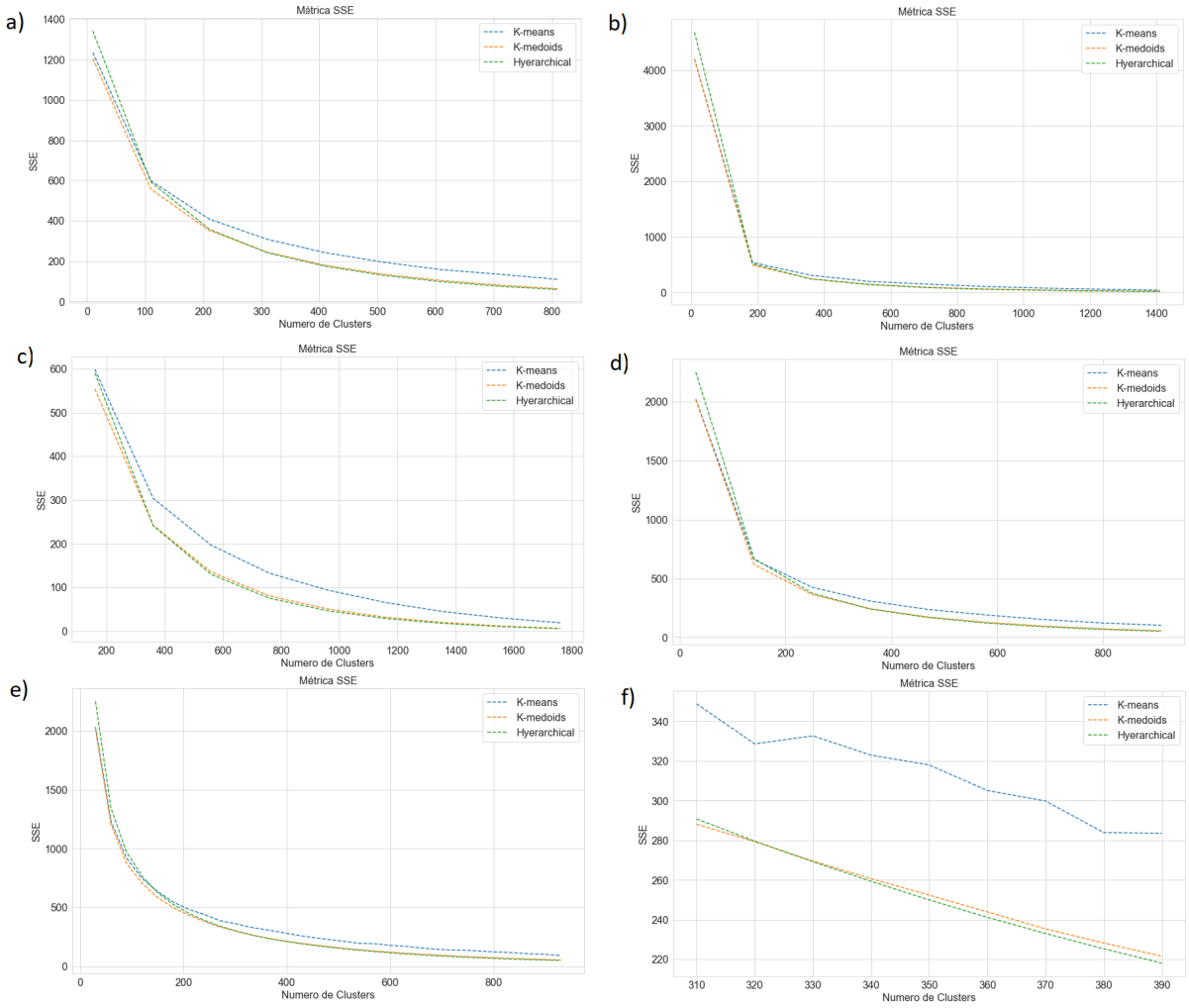
(e) varredura de 30 a 930 centroides com incremento de 30 em 30;

(f) varredura de 310 a 390 centroides com incremento de 10 em 10.

A Figura 23 mostra os resultados para as diferentes varreduras propostas, casos (a), (b), (c), (d), (e) e (f), considerando os algoritmos Hierárquico, *K-Medoids* e *K-Means* na métrica SSE, na base de dados da indústria automotiva, com as etapas de pre-processamento aplicadas, utilizando os 4 principais componentes do PCA.

Através da análise gráfica nota-se que há uma grande similaridade no comportamento das curvas para todos os cenários de número de centroides e também que na varredura do SSE de 310 a 390 centroides há uma significativa diferença entre *K-Means* e os demais, com vantagem destes por esta se tratar de uma curva de minimização.

Figura 23 – Resultado da aplicação dos algoritmos estudados utilizando SSE variando a quantidade de centroides: (a) 60 a 860, (b) 10 a 1410, (c) 160 a 1760, (d) 30 a 910, (e) 30 a 930, (f) 310 a 390.



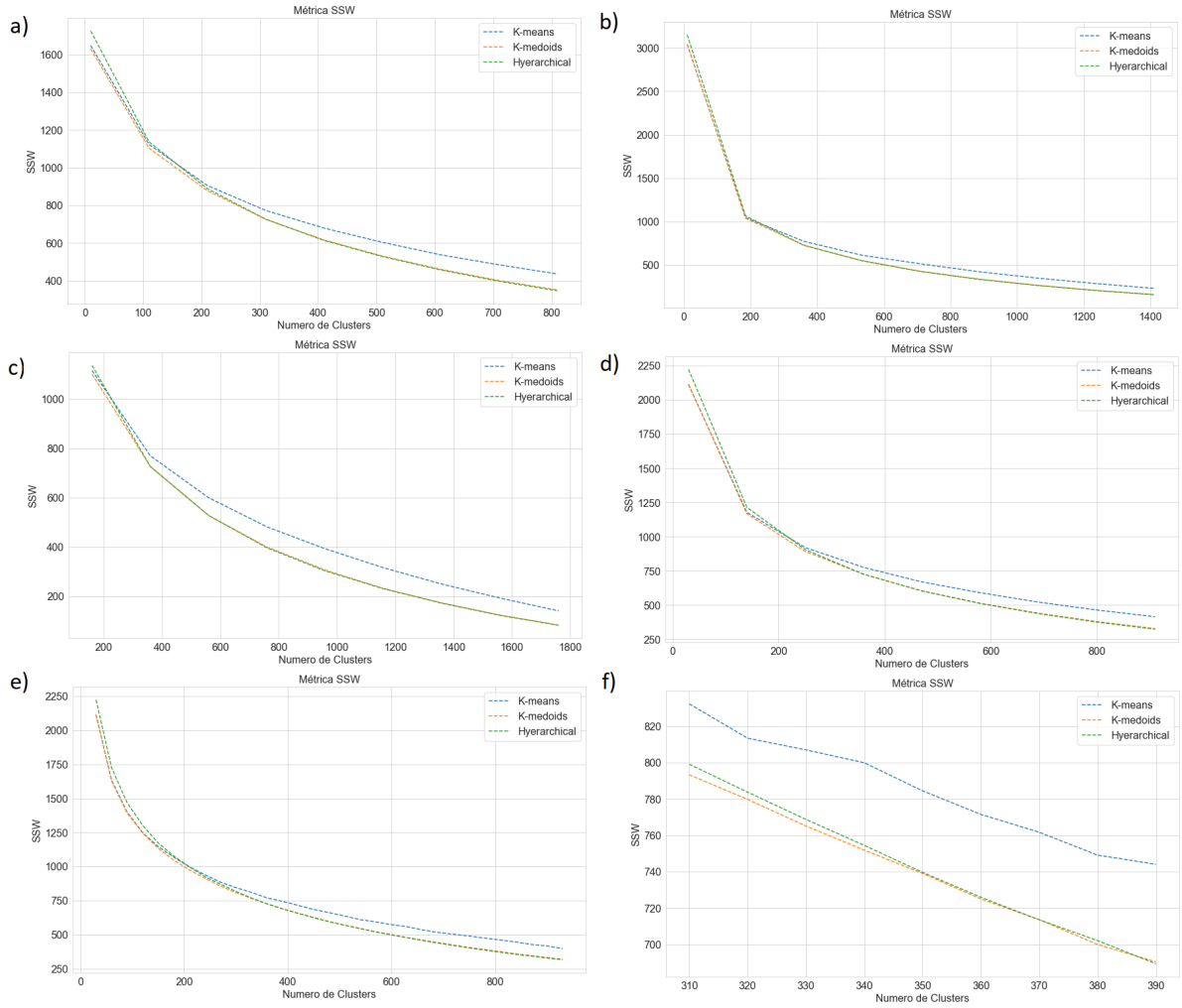
Fonte: Autoria Própria

Novamente analisando os gráficos (a), (c), (d) e (e), nota-se uma severa diminuição da velocidade de melhoria do indicador com o incremento de mais centroides próximo de 400. O gráfico (b) mostra esta tendência já próximo de 200 centroides. Isso pode ser explicado pelo fato de o gráfico de ser o único com um valor inicial de centroides começando em 10 e com incrementos de 175 em 175, fazendo com que a variação da primeira etapa seja muito maior.

Já a análise do gráfico (f) mostra uma maior incerteza em relação a quantidade ideal de centroides. O indicativo seria escolher entre 310 a 390, uma vez que estes incrementos de 10 em 10, continuam trazendo proporcionais ganhos no SSE.

A Figura 24 mostra os resultados para a métrica SSW, nas mesmas condições acima expostas.

Figura 24 – Resultado da aplicação dos algoritmos estudados utilizando SSW variando a quantidade de centroides: (a) 60 a 860, (b) 10 a 1410, (c) 160 a 1760, (d) 30 a 910, (e) 30 a 930, (f) 310 a 390.

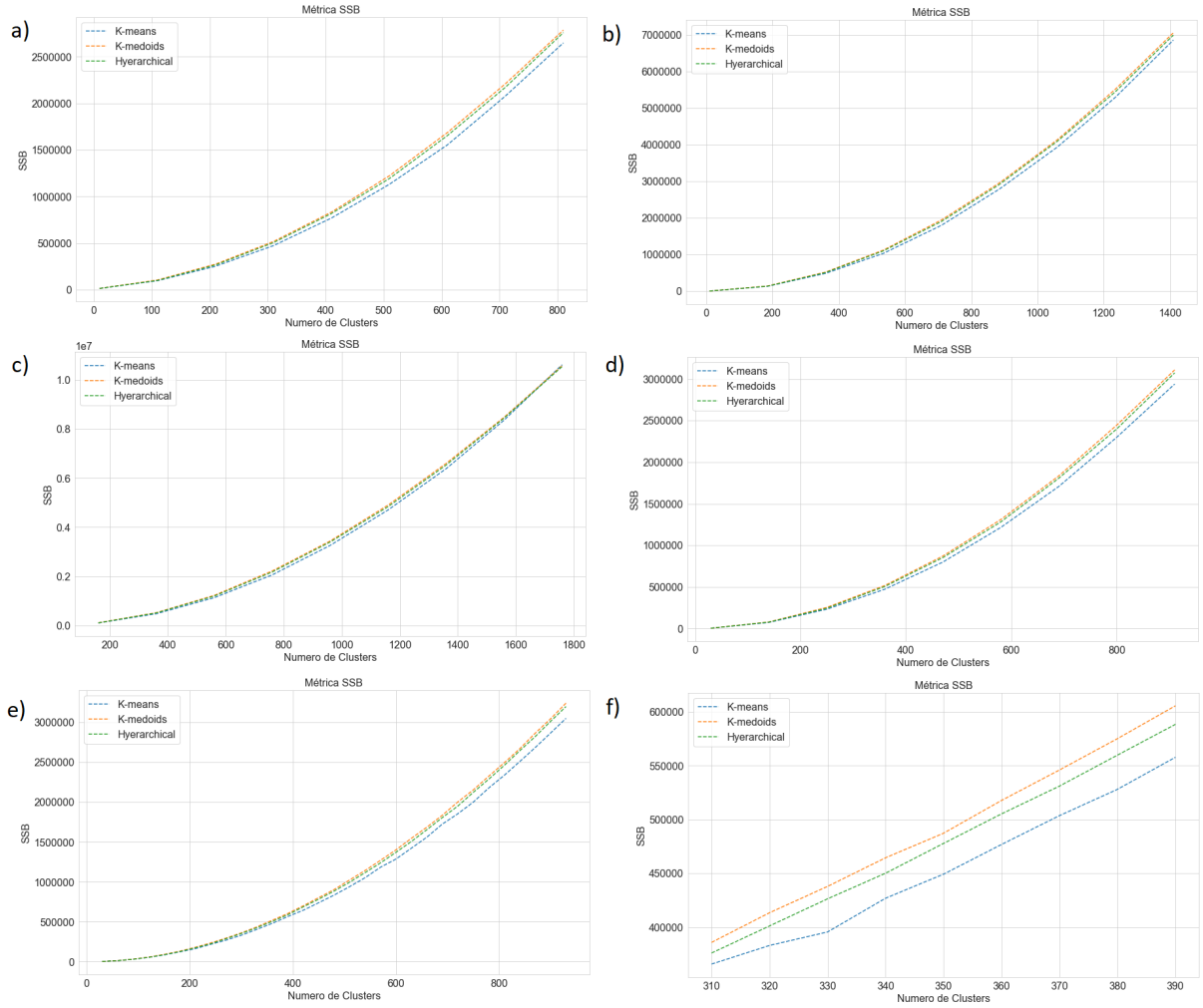


Fonte: Autoria Própria

Da mesma forma como no SSE percebe-se um comportamento similar das curvas nos diferentes cenários para a métrica SSW. Isso se deve pelas equações de cada uma das métricas serem basicamente as mesmas conforme exposto anteriormente. Também percebe-se aqui uma ligeira desvantagem do *K-Means* em relação ao *K-Medoids* e Hierárquico.

A Figura 25 mostra os resultados para a métrica SSB.

Figura 25 – Resultado da aplicação dos algoritmos estudados utilizando SSB variando a quantidade de centroides: (a) 60 a 860, (b) 10 a 1410, (c) 160 a 1760, (d) 30 a 910, (e) 30 a 930, (f) 310 a 390.

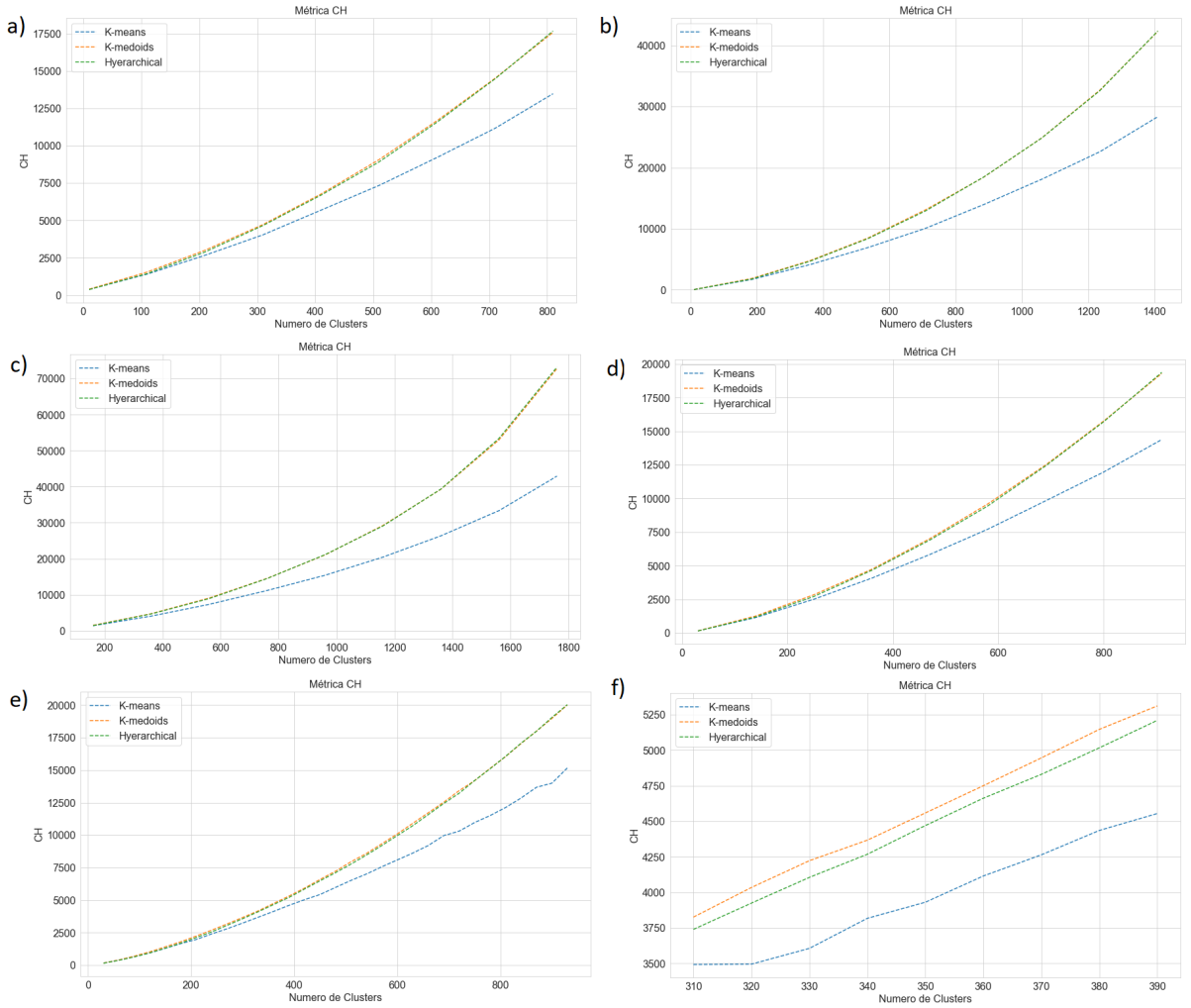


Fonte: Autoria Própria

Para o SSB observa-se um aumento deste indicador com o aumento da quantidade de centroides, assim como observado na base de dados Iris. As curvas possuem comportamento bem similares, em que a maior diferença pode ser observada no gráfico (f), que aplica uma varredura de 310 a 390 centroides com incremento de 10 em 10. Neste gráfico nota-se uma pequena vantagem do algoritmo *K-Medoids* se comparado aos algoritmos Hierárquico e *K-Means*, respectivamente.

A Figura 26 mostra os resultados para a métrica CH.

Figura 26 – Resultado da aplicação dos algoritmos estudados utilizando CH variando a quantidade de centroides: (a) 60 a 860, (b) 10 a 1410, (c) 160 a 1760, (d) 30 a 910, (e) 30 a 930, (f) 310 a 390.

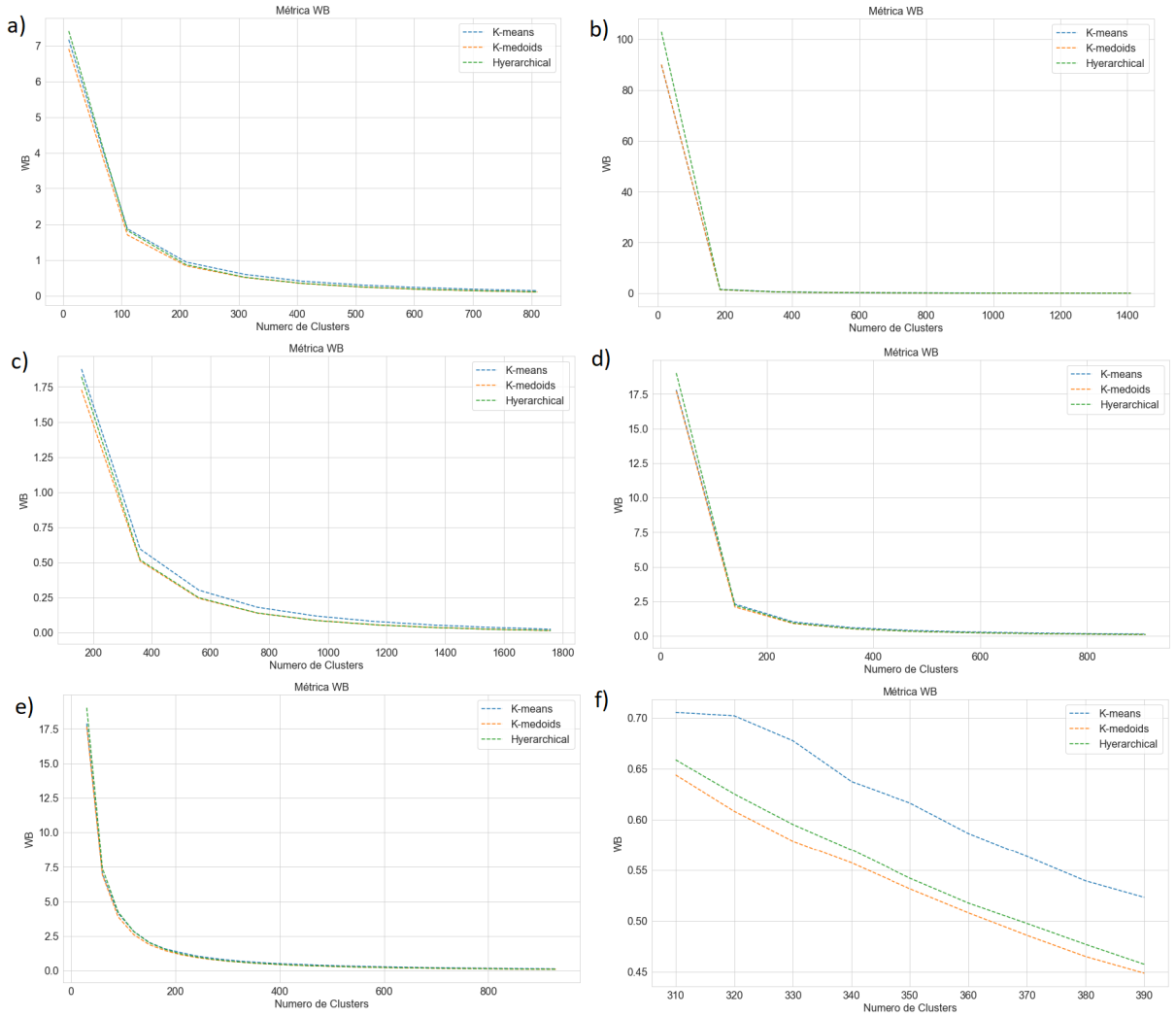


Fonte: Autoria Própria

Como abordado anteriormente esta métrica possui uma correlação diretamente proporcional entre SSB e SSW como pode ser verificado na revisão da equação 3.4. Portanto, trata-se de uma métrica em que busca-se a maximização dos resultados. Aqui também não fica evidente o ponto de inflexão da curva mesmo com as novas varreduras de quantidade de centroides estabelecidas. Importante salientar que os gráficos deixam evidente é que o aumento na quantidade de centroides trazem ganhos significativos na métrica CH, desta forma não permitindo que seja definida a quantidade de centroides a partir do qual haja uma certa estabilização da métrica, o que traga pequenos incrementos no indicador aumentando-se a quantidade de centroides.

A Figura 27 mostra os resultados para a métrica WB.

Figura 27 – Resultado da aplicação dos algoritmos estudados utilizando WB variando a quantidade de centroides: (a) 60 a 860, (b) 10 a 1410, (c) 160 a 1760, (d) 30 a 910, (e) 30 a 930, (f) 310 a 390.



Fonte: Autoria Própria

Pela equação 3.5 pode-se verificar que o WB é diretamente proporcional à razão de SSW e SSB, se tratando de uma métrica de minimização. As curvas (a), (d) e (e) convergem para melhorias significativas até próximo de 320 a 350 centroides. Já a curva (b) mostra uma tendencia de estabilização próximo de 200, basicamente porque, assim como apresentado anteriormente, é a única curva com um valor inicial de centroides começando em 10 e com incrementos de 175 em 175, fazendo com que a variação da primeira etapa seja muito maior.

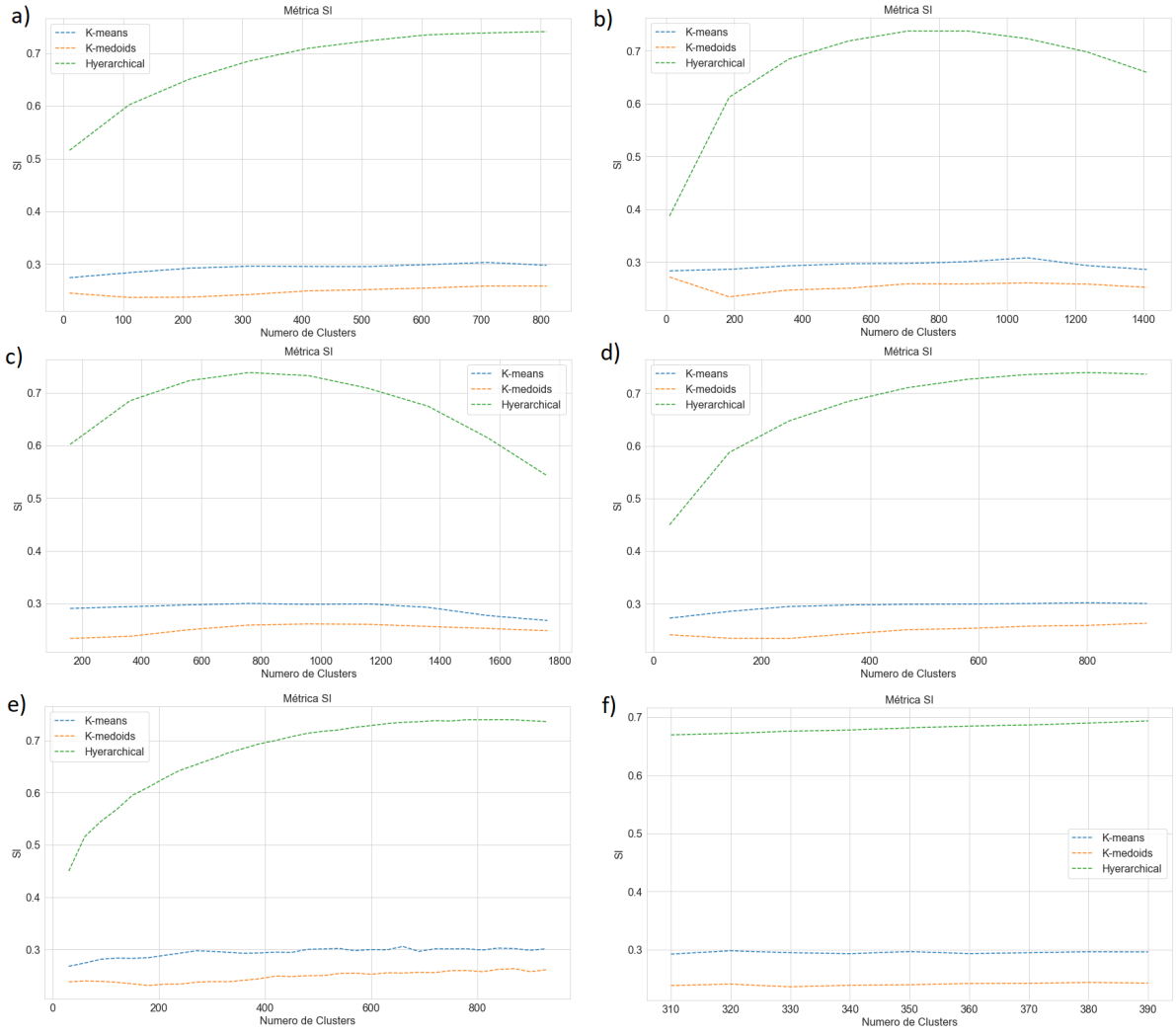
Com relação a curva (c) percebe-se que como se inicia com 160 centroides variando de 200 em 200 é obtida uma curva suavizada do indicador, o que permite inferir que a quantidade ideal de centroides é entre 800 e 1000 grupos.

No primeiro cenário estudado , a indicação era que a quantidade mais adequada de centroides era em torno de 360, e um questionamento foi levantado, por que não 320, ou 330, ou 380. O gráfico (f) apresentado comprova que quando fazemos uma aproximação na plotagem

da quantidade de centroides a evolução é quase que linear, não permitindo esta definição com exatidão a partir da plotagem do WB.

A Figura 28 mostra os resultados para a métrica SI.

Figura 28 – Resultado da aplicação dos algoritmos estudados utilizando SI variando a quantidade de centroides: (a) 60 a 860, (b) 10 a 1410, (c) 160 a 1760, (d) 30 a 910, (e) 30 a 930, (f) 310 a 390.



Fonte: Autoria Própria

Pela análise das curvas de SI nos diferentes cenários nota-se um comportamento linear e estabilizado de evolução desta métrica para os algoritmos *K-Medoids* e *K-Means* com uma pequena vantagem para este último. Já o algoritmo Hierárquico possui uma evolução significativa no indicador com um pico em aproximadamente em 800 centroides e contém os melhores resultados entre os três algoritmos.

Tenta-se que para o estudo de caso da referida indústria, os potenciais rótulos da base de dados que são: NCM e classe de engenharia, e possuem uma variação entre de 200 a 621 grupos respectivamente.

Outro ponto a destacar é que para os algoritmos estocásticos como o *K-Medoids* precisa

de pelo menos 30 rodadas do algoritmo, com isso, o custo computacional acaba sendo maior que o algoritmo Hierárquico. Cada rodada do algoritmo *K-Medoids* durou 11,47 segundos e o total as 30 rodadas consumiram 343,10 segundos. Já o Hierárquico, que é uma técnica determinísticas, levou 49,87 segundos até a convergência.

Dentre as métricas exploradas não houve uma que mostre o resultado mais adequado para as diferentes bases e técnicas aplicadas. Para a base da indústria, pela ausência de rótulos e pelo indicativo das dimensões de NCM e classe de engenharia de 200 a 621 grupos, a conclusão é de que a melhor métrica é a WB, com a qual indica-se aproximadamente 360 centroides como uma quantidade de grupos a serem formados. A aplicação de múltiplas métricas também auxilia a definição mais assertiva da quantidade de grupos. De certa forma, o conjunto de métricas atua complementarmente.

De posse dos resultados da clusterização e após a verificação do melhor desempenho dos algoritmos Hierárquico, *K-Medoids* e *K-Means*, foi realizada uma reunião com os times de engenharia de produto, engenharia de manufatura, controladoria e compras da referida empresa automotiva para uma análise crítica dos resultados. Em um dos casos, foi detectado que 3 assentos do motorista estava em um *cluster* enquanto outro estava em um grupo único. Quando retornamos a base de dados original descobrimos que uma de suas dimensões estava errada e isto distorceu o resultado.

Outros erros similares foram observados, principalmente em componentes pequenos os quais, por vezes, as dimensões básicas geométricas X, Y e Z correspondem na verdade a embalagem em si e não as dimensões reais da peça. Em outros casos, tratava-se do conjunto de peças e não de uma peça única. O mesmo ocorreu com o peso dos componentes. Uma vez os dados corrigidos fizeram com que os novos grupos formados fossem coerentes.

A base de coleta de dados esta em fase final de implementação para posterior criação de relatórios gerenciais de forma automatizada para identificação de oportunidades de redução de custos por similaridade.

Outro ponto que esta sendo cogitado pela empresa seria a aplicação das referidas técnicas não só para comparação de custos de peças já fabricadas como também para elaboração de predição de novos designs com um viés em eficiência de custo de fabricação.

7 CONCLUSÕES

Na indústria automotiva é muito comum que a concepção de novos produtos com características similares sejam propostas por diferentes equipes de desenvolvimento, o que acaba gerando um desafio ainda maior de comunização de componentes. Devido ao elevado volume de dados gerados por múltiplas equipes, as quais não usam ferramentas de mineração de dados, oportunidades de redução de custos acabam por serem perdidas, onerando a cadeia produtiva. Neste âmbito, o presente trabalho utilizou de ferramentas de *clustering* para organizar o reconhecimento de similaridade entre peças. A meta global é otimizar os custos de produtos correntes, com vistas a identificação oportunidades e a posterior comparação de custos.

Geralmente os algoritmos para clusterização são projetados com certas suposições e favorecem algum tipo de viés. Nesse sentido, não é adequado apontar um método "melhor"neste contexto, embora algumas comparações sejam possíveis. Estas são baseadas principalmente em algumas aplicações específicas, em determinadas condições e os resultados podem se tornar bastante diferentes se tais condições forem alteradas. Em resumo, clusterização é uma abordagem interessante e bastante útil em problemas desafiadores. Tem um grande potencial em aplicativos como reconhecimento de objetos, segmentação de imagens, filtragem e recuperação de informações. No entanto, é possível explorar esse potencial somente após várias escolhas de design.

Neste trabalho, inicialmente foi obtido melhor resultado de acerto na base Iris, que possuí rótulos, com o algoritmo genético. Para uma base de dados real da indústria automotiva, só é possível a comparação dos resultados através da análise das métricas selecionadas: SSE, SSW, SSB, CH, WB e SI. O melhor resultado ficou com o algoritmo Hierárquico.

Um outro ponto a salientar foi a constatação do custo computacional dos melhores algoritmos. É amplamente encontrado a literatura que o método Hierárquico possui um custo significativamente maior que o algoritmo *K-Medoids*. Tal constatação foi confirmada, quando considerando rodada única. Entretanto, devido a análise de dispersão realizada é necessário pelo menos 30 rodadas para uma significativa exploração dos resultados nos algoritmos estocásticos, fazendo com que o tempo das 30 rodadas seja maior que a única rodada do algoritmo Hierárquico, uma técnica determinística.

As soluções promissoras para determinação do número de grupos são úteis, pois não requerem informações a priori sobre o número real de *clusters* presentes no conjunto de dados. É importante ter em mente que o sucesso de uma metodologia depende muito de como ela foi projetada como seu esquema de codificação, função objetivo (ou índice de avaliação), métricas, medida de proximidade, etc. Pode-se observar na literatura que algoritmos multiobjetivos que consideram vários critérios de validade são adequados sobre algoritmos de *clustering* de objetivo único, porque fornecem a flexibilidade para selecionar a solução desejada a partir de um conjunto de soluções ideais. Além disso, abordagens multiobjetivos são capazes de lidar eficientemente com *clusters* não linearmente separáveis. Embora algoritmos multiobjetivos para agrupamento

automático já tenham sido utilizados, consideramos este tópico de pesquisa atualmente pouco explorado na literatura.

7.1 TRABALHOS FUTUROS

Apesar da grande quantidade de testes realizados por este trabalho nota-se que o desempenho apresentado por cada modelo de clusterização depende dos dados de entrada, das métricas e da técnica utilizada. Ntes trabalho foram utilizadas técnicas que consideram apenas dados quantitativos, excluindo-se da base de análise os dados qualitativos, como descrição dos itens, código NCM e classe de engenharia. Assim, faz-se necessário submeter a base aqui apresentada a modelos e técnicas que consideram dados quantitativos e qualitativos simultaneamente, como Distância *Gower* (*Gower Distance*). Também um possível estudo futuro pode ser a aplicação outras técnicas conhecidas de transformação de dados, como Box-Cox.

Pode-se encontrar também na literatura uma série de composições de algoritmos híbridos que abordam várias técnicas de clusterização, buscando a eliminação dos pontos fracos de cada método individualmente. Esta área tem demonstrado resultados interessantes com uma grande atenção dos pesquisadores na área de otimização. Pode-se citar algumas baseadas no PSO como: Clusterização PSO Hierárquica (*Hierarchical PSO for Clustering - HPSO-clustering*), clusterização PSO Evolucionária (*Evolutionary PSO for clustering - EPSO-clustering*), que podem ser aplicadas a estas bases.

REFERÊNCIAS

AIBINU, AM *et al.* A novel clustering based genetic algorithm for route optimization. **Engineering Science and Technology, an International Journal**, Elsevier, v. 19, n. 4, p. 2022–2034, 2016.

ALAM, Shafiq *et al.* Research on particle swarm optimization based clustering: a systematic review of literature and techniques. **Swarm and Evolutionary Computation**, Elsevier, v. 17, p. 1–13, 2014.

ALAM, Shafiq; DOBBIE, Gillian; REHMAN, Saeed Ur. Analysis of particle swarm optimization based hierarchical data clustering approaches. **Swarm and Evolutionary Computation**, Elsevier, v. 25, p. 36–51, 2015.

ALHONIEMI, Esa *et al.* Process monitoring and modeling using the self-organizing map. **Integrated Computer-Aided Engineering**, IOS Press, v. 6, n. 1, p. 3–14, 1999.

ANDERBERG, Michael R. The broad view of cluster analysis. In: **Cluster Analysis for Applications**. [S.l.]: Elsevier, 1973. p. 1–9.

ARGONETO, Pierluigi; RENNA, Paolo. Capacity sharing in a network of enterprises using the gale–shapley model. **The International Journal of Advanced Manufacturing Technology**, Springer, v. 69, n. 5-8, p. 1907–1916, 2013.

ARORA, Preeti; VARSHNEY, Shipra *et al.* Analysis of k-means and k-medoids algorithm for big data. **Procedia Computer Science**, Elsevier, v. 78, p. 507–512, 2016.

BATISTA, Rodrigo de Abreu. Classificação automática de códigos ncm utilizando o algoritmo naïve bayes. 2017.

BEZDEK, James C; EHRLICH, Robert; FULL, William. Fcm: The fuzzy c-means clustering algorithm. **Computers & Geosciences**, Elsevier, v. 10, n. 2-3, p. 191–203, 1984.

BHATTACHARJYA, Rajib Kumar. Introduction to genetic algorithms. **IIT Guwahati**, v. 12, 2012.

BHOLOWALIA, Purnima; KUMAR, Arvind. Ebk-means: A clustering technique based on elbow method and k-means in wsn. **International Journal of Computer Applications**, Citeseer, v. 105, n. 9, 2014.

BINESH, Neda; REZGHI, Mansoor. Fuzzy clustering in community detection based on nonnegative matrix factorization with two novel evaluation criteria. **Applied Soft Computing**, Elsevier, v. 69, p. 689–703, 2018.

BOUHOUCHE, Salah; YAHI, Mostepha; BAST, Jürgen. Combined use of principal component analysis and self organisation map for condition monitoring in pickling process. **Applied Soft Computing**, Elsevier, v. 11, n. 3, p. 3075–3082, 2011.

BRERETON, Pearl *et al.* Lessons from applying the systematic literature review process within the software engineering domain. **Journal of systems and software**, Elsevier, v. 80, n. 4, p. 571–583, 2007.

CALIŃSKI, Tadeusz; HARABASZ, Jerzy. A dendrite method for cluster analysis. **Communications in Statistics-theory and Methods**, Taylor & Francis, v. 3, n. 1, p. 1–27, 1974.

CASELLA, George; BERGER, Roger L. Inferência estatística-tradução da 2ª edição norteamericana. **Centage Learning**, 2011.

CASTRO, Leandro Nunes De. **Fundamentals of natural computing: basic concepts, algorithms, and applications**. [S.l.]: Chapman and Hall/CRC, 2006.

COHEN, Sandra CM; CASTRO, Leandro N de. Data clustering with particle swarms. In: IEEE. **2006 IEEE International Conference on Evolutionary Computation**. [S.l.], 2006. p. 1792–1798.

COVER, Thomas M; THOMAS, Joy A. Elements of information theory, 1991 john wiley & sons. **Inc. Print ISBN 0-471-06259-6 Online ISBN 0-471-20061-1**, 1991.

COWGILL, Marcus Charles; HARVEY, Robert J; WATSON, Layne T. A genetic algorithm approach to cluster analysis. **Computers & Mathematics with Applications**, Elsevier, v. 37, n. 7, p. 99–108, 1999.

DING, Shifei *et al.* Using genetic algorithms to optimize artificial neural networks. In: CITESEER. **Journal of Convergence Information Technology**. [S.l.], 2010.

DOUGHERTY, Geoff. **Pattern recognition and classification: an introduction**. [S.l.]: Springer Science & Business Media, 2012.

EBERHART, Russell; KENNEDY, James. A new optimizer using particle swarm theory. In: IEEE. **MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science**. [S.l.], 1995. p. 39–43.

ELMARAGHY, Hoda A. **Changeable and reconfigurable manufacturing systems**. [S.l.]: Springer Science & Business Media, 2008.

ENGELBRECHT, Andries P. **Computational intelligence: an introduction**. [S.l.]: John Wiley & Sons, 2007.

ESTER, Martin *et al.* A density-based algorithm for discovering clusters in large spatial databases with noise. In: **Kdd**. [S.l.: s.n.], 1996. v. 96, n. 34, p. 226–231.

FALKENAUER, Emanuel. **Genetic algorithms and grouping problems**. [S.l.]: John Wiley & Sons, Inc., 1998.

FATALLA, Fabio Campos. **Proposta de metodologia para classificação fiscal de mercadorias têxteis na Nomenclatura Comum do Mercosul**. Tese (Doutorado) — Universidade de São Paulo, 2016.

FIGUEIREDO, Elliackin *et al.* Swarm intelligence for clustering—a systematic review with new perspectives on data mining. **Engineering Applications of Artificial Intelligence**, Elsevier, v. 82, p. 313–329, 2019.

FILHO, Telmo M Silva *et al.* Hybrid methods for fuzzy clustering based on fuzzy c-means and improved particle swarm optimization. **Expert Systems with Applications**, Elsevier, v. 42, n. 17-18, p. 6315–6328, 2015.

FRÄNTI, Pasi; SIERANOJA, Sami. K-means properties on six clustering benchmark datasets. **Applied Intelligence**, Springer, v. 48, n. 12, p. 4743–4759, 2018.

GAMEROS, A *et al.* State-of-the-art in fixture systems for the manufacture and assembly of rigid components: A review. **International Journal of Machine Tools and Manufacture**, Elsevier, v. 123, p. 1–21, 2017.

GARCÍA, Salvador; LUENGO, Julián; HERRERA, Francisco. **Data preprocessing in data mining**. [S.l.]: Springer, 2015.

GHASEMINEZHAD, MH; KARAMI, A. A novel self-organizing map (som) neural network for discrete groups of data clustering. **Applied Soft Computing**, Elsevier, v. 11, n. 4, p. 3771–3778, 2011.

GUAN, Chun; YUEN, Kevin Kam Fung; COENEN, Frans. Particle swarm optimized density-based clustering and classification: Supervised and unsupervised learning approaches. **Swarm and evolutionary computation**, Elsevier, v. 44, p. 876–896, 2019.

HAN, Jiawei; PEI, Jian; KAMBER, Micheline. **Data mining: concepts and techniques**. [S.l.]: Elsevier, 2011.

HANCER, Emrah; KARABOGA, Dervis. A comprehensive survey of traditional, merge-split and evolutionary approaches proposed for determination of cluster number. **Swarm and Evolutionary Computation**, Elsevier, v. 32, p. 49–67, 2017.

HANSEN, Jan Ole; KAMPKER, Achim; TRIEBS, Johannes. Approaches for flexibility in the future automobile body shop: results of a comprehensive cross-industry study. **Procedia CIRP**, Elsevier, v. 72, p. 995–1002, 2018.

HOLTEWERT, Philipp; BAUERNHANSL, Thomas. Optimal configuration of manufacturing cells for high flexibility and cost reduction by component substitution. **Procedia CIRP**, Elsevier, v. 41, p. 111–116, 2016.

HONG, Yoon-Seok; ROSEN, Michael R. Intelligent characterisation and diagnosis of the groundwater quality in an urban fractured-rock aquifer using an artificial neural network. **Urban Water**, Elsevier, v. 3, n. 3, p. 193–204, 2001.

HRUSCHKA, Eduardo Raul *et al.* A survey of evolutionary algorithms for clustering. **IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)**, IEEE, v. 39, n. 2, p. 133–155, 2009.

JAIN, Anil K; DUBES, Richard C. Algorithms for clustering data. **Englewood Cliffs: Prentice Hall, 1988**, 1988.

JOSÉ-GARCÍA, Adán; GÓMEZ-FLORES, Wilfrido. Automatic clustering using nature-inspired metaheuristics: A survey. **Applied Soft Computing**, Elsevier, v. 41, p. 192–213, 2016.

JUNTUNEN, Petri *et al.* Cluster analysis by self-organizing maps: An application to the modelling of water quality in a treatment process. **Applied Soft Computing**, Elsevier, v. 13, n. 7, p. 3191–3196, 2013.

KARYPIS, Michael Steinbach George; KUMAR, Vipin; STEINBACH, Michael. A comparison of document clustering techniques. In: **TextMining Workshop at KDD2000 (May 2000)**. [S.l.: s.n.], 2000.

KITCHENHAM, Barbara *et al.* Systematic literature reviews in software engineering–a systematic literature review. **Information and software technology**, Elsevier, v. 51, n. 1, p. 7–15, 2009.

KOHONEN, Teuvo. Overture. In: **Self-Organizing neural networks**. [S.l.]: Springer, 2002. p. 1–12.

KRAPPE, H; ROGALSKI, Sven; SANDER, M. Challenges for handling flexibility in the change management process of manufacturing systems. In: IEEE. **2006 IEEE International Conference on Automation Science and Engineering**. [S.l.], 2006. p. 551–557.

LIUKKONEN, Mika *et al.* Quality-oriented optimization of wave soldering process by using self-organizing maps. **Applied Soft Computing**, Elsevier, v. 11, n. 1, p. 214–220, 2011.

____. Modeling of the fluidized bed combustion process and nox emissions using self-organizing maps: An application to the diagnosis of process states. **Environmental Modelling & Software**, Elsevier, v. 26, n. 5, p. 605–614, 2011.

LUXBURG, Ulrike Von. A tutorial on spectral clustering. **Statistics and computing**, Springer, v. 17, n. 4, p. 395–416, 2007.

MACEDO, Leonardo Correia Lima. **Direito tributário no comércio internacional**. [S.l.]: Edições Aduaneiras, 2005.

MACQUEEN, James *et al.* Some methods for classification and analysis of multivariate observations. In: OAKLAND, CA, USA. **Proceedings of the fifth Berkeley symposium on mathematical statistics and probability**. [S.l.], 1967. v. 1, n. 14, p. 281–297.

MILLIGAN, Glenn W; COOPER, Martha C. A study of standardization of variables in cluster analysis. **Journal of classification**, Springer, v. 5, n. 2, p. 181–204, 1988.

MOHAMAD, Ismail Bin; USMAN, Dauda. Standardization and its effects on k-means clustering algorithm. **Research Journal of Applied Sciences, Engineering and Technology**, v. 6, n. 17, p. 3299–3303, 2013.

MOHD, Wan Maseri Binti Wan *et al.* An improved parameter less data clustering technique based on maximum distance of data and lioyd k-means algorithm. **Procedia Technology**, Elsevier, v. 1, p. 367–371, 2012.

MUKHOPADHYAY, Anirban *et al.* A survey of multiobjective evolutionary algorithms for data mining: Part i. **IEEE Transactions on Evolutionary Computation**, IEEE, v. 18, n. 1, p. 4–19, 2013.

NANDA, Satyasai Jagannath; PANDA, Ganapati. A survey on nature inspired metaheuristic algorithms for partitional clustering. **Swarm and Evolutionary computation**, Elsevier, v. 16, p. 1–18, 2014.

NGUYEN, Hoang *et al.* A new soft computing model for estimating and controlling blast-produced ground vibration based on hierarchical k-means clustering and cubist algorithms. **Applied Soft Computing**, Elsevier, v. 77, p. 376–386, 2019.

OZTURK, Celal; HANCER, Emrah; KARABOGA, Dervis. Improved clustering criterion for image clustering with artificial bee colony algorithm. **Pattern Analysis and Applications**, Springer, v. 18, n. 3, p. 587–599, 2015.

PANDOVE, Divya; GOEL, Shivan; RANI, Rinkl. Systematic review of clustering high-dimensional and large datasets. **ACM Transactions on Knowledge Discovery from Data (TKDD)**, ACM, v. 12, n. 2, p. 16, 2018.

PARK, Hae-Sang; LEE, Jong-Seok; JUN, Chi-Hyuck. A k-means-like algorithm for k-medoids clustering and its performance. **Proceedings of ICCIE**, p. 102–117, 2006.

PEARSON, Karl. Principal components analysis. **The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science**, v. 6, n. 2, p. 559, 1901.

PEDREGOSA, Fabian *et al.* Scikit-learn: Machine learning in python. **Journal of machine learning research**, v. 12, n. Oct, p. 2825–2830, 2011.

PESHKO, Olesya. Global optimization genetic algorithms. **McMaster University Hamilton, Ontario ppt presentation**, v. 25, 2007.

RAMADAS, Meera; ABRAHAM, Ajith; KUMAR, Sushil. Fsde-forced strategy differential evolution used for data clustering. **Journal of King Saud University-Computer and Information Sciences**, Elsevier, 2016.

RANA, Sandeep; JASOLA, Sanjay; KUMAR, Rajesh. A review on particle swarm optimization algorithms and their applications to data clustering. **Artificial Intelligence Review**, Springer, v. 35, n. 3, p. 211–222, 2011.

ROUSSEEUW, Peter J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. **Journal of computational and applied mathematics**, North-Holland, v. 20, p. 53–65, 1987.

SANDER, Jörg *et al.* Density-based clustering in spatial databases: The algorithm gdbscan and its applications. **Data mining and knowledge discovery**, Springer, v. 2, n. 2, p. 169–194, 1998.

SANTOS, Pedro *et al.* Application of pso-based clustering algorithms on educational databases. In: IEEE. **2017 IEEE Latin American Conference on Computational Intelligence (LA-CCI)**. [S.l.], 2017. p. 1–6.

SCHUBERT, Erich *et al.* A framework for clustering uncertain data. **Proceedings of the VLDB Endowment**, VLDB Endowment, v. 8, n. 12, p. 1976–1979, 2015.

____. Dbscan revisited, revisited: why and how you should (still) use dbscan. **ACM Transactions on Database Systems (TODS)**, ACM, v. 42, n. 3, p. 19, 2017.

SENTHILKUMAR, P; VANITHA, N Suthanthira. A stride towards developing efficient approaches for data clustering based on evolutionary programming. 2013.

SINGH, Dalwinder; SINGH, Birmohan. Investigating the impact of data normalization on classification performance. **Applied Soft Computing**, Elsevier, p. 105524, 2019.

____. Investigating the impact of data normalization on classification performance. **Applied Soft Computing**, Elsevier, v. 97, p. 105524, 2020.

SINGH, Shalini S; CHAUHAN, NC. K-means v/s k-medoids: A comparative study. In: **National Conference on Recent Trends in Engineering & Technology**. [S.l.: s.n.], 2011. v. 13.

SIQUEIRA, Hugo Valadares *et al.* Máquinas desorganizadas para previsão de séries de vazões. [sn], 2013.

SOOD, Monica; BANSAL, Shilpi. K-medoids clustering technique using bat algorithm. **International Journal of Applied Information Systems**, Citeseer, v. 5, n. 8, p. 20–22, 2013.

STEINLEY, Douglas. Standardizing variables in k-means clustering. In: **Classification, clustering, and data mining applications**. [S.l.]: Springer, 2004. p. 53–60.

SU, Ting; DY, Jennifer. A deterministic method for initializing k-means clustering. In: IEEE. **16th IEEE International Conference on Tools with Artificial Intelligence**. [S.l.], 2004. p. 784–786.

TAN, Pang-Ning *et al.* Cluster analysis: basic concepts and algorithms. **Introduction to data mining**, Addison-Wesley Longman Publishing Co, Boston, MA, v. 8, p. 487–568, 2006.

THINSUNGNOENA, Tippaya *et al.* The clustering validity with silhouette and sum of squared errors. **learning**, v. 3, n. 7, 2015.

VELMURUGAN, T; SANTHANAM, T. Computational complexity between k-means and k-medoids clustering algorithms for normal and uniform distributions of data points. **Journal of computer science**, v. 6, n. 3, p. 363, 2010.

WU, Nuosi; ZHU, Zexuan; JI, Zhen. A growing partitional clustering based on particle swarm optimization. In: IEEE. **2014 IEEE Congress on Evolutionary Computation (CEC)**. [S.l.], 2014. p. 229–234.

ZHAO, Qinpei; XU, Mantao; FRÄNTI, Pasi. Sum-of-squares based cluster validity index and significance analysis. In: SPRINGER. **International Conference on Adaptive and Natural Computing Algorithms**. [S.l.], 2009. p. 313–322.

ZHONG, Yanfei; ZHANG, Shuai; ZHANG, Liangpei. Automatic fuzzy clustering based on adaptive multi-objective differential evolution for remote sensing imagery. **IEEE Journal of selected topics in applied earth observations and remote sensing**, IEEE, v. 6, n. 5, p. 2290–2301, 2013.

APÊNDICE

APÊNDICE A - ARTIGOS PUBLICADOS EM ANAIS DE CONGRESSOS

GUERREIRO, M. T.; CASTANHO, D. S.; MARTINS, M; TROJAN, F.; CORREA, F.; SIQUEIRA, H. V.**Clusterização de componentes de indústria de caminhões por meiode metaheurísticas bio-inspiradas**. In: FERNANDES, Bruno José Torres; Pereira Júnior,Antônio (Ed.).Anais do 14 Congresso Brasileiro de Inteligência Computacional. Curitiba,PR: ABRICOM, 2019. p. 1–7. ISBN 978-856997201-3