

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO RIO GRANDE
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM COMPUTACIONAL

**UM ALGORITMO PARA AGRUPAMENTO DE DADOS
UTILIZANDO INTERAÇÃO ENTRE AGENTES**

por

Lutiele Machado Godois

Dissertação para obtenção do Título de
Mestre em Modelagem Computacional

Rio Grande, fevereiro, 2018


Lutiele Machado Godois

“ Um algoritmo para agrupamento de dados utilizando interação entre agentes ”


Dissertação apresentada ao Programa de Pós Graduação em Modelagem Computacional da Universidade Federal do Rio Grande - FURG, como requisito parcial para obtenção do Grau de Mestre. Área concentração: Modelagem Computacional.

Aprovada em

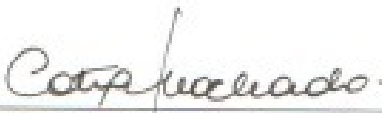
BANCA EXAMINADORA



Prof. Dr. Leonardo Ramos Echimendorfer
Orientador - FURG



Prof. Dr. Diana Francisca Adamatti
Coorientadora - FURG



Prof. Dr. Catia Maria dos Santos Machado
FURG



Prof. Dr. Marilton Sanchotene de Agular
UFPEL

Ficha catalográfica

S586p Godois, Lutiele Machado.

Um algoritmo para agrupamento de dados utilizando interação entre agentes / Lutiele Machado Godois. – 2018.
56 p.

Dissertação (mestrado) – Universidade Federal do Rio Grande – FURG, Programa de Pós-graduação em Modelagem Computacional, Rio Grande/RS, 2018.

Orientador: Dr. Leonardo Ramos Emmendorfer.

1. Algoritmos de clustering 2. Sistemas multiagente 3. Validação de agrupamento I. Emmendorfer, Leonardo Ramos II. Título.

CDU 004.891

Dedico este trabalho a minha família,
que não mediu esforços para que eu concluísse
mais uma etapa da minha vida acadêmica.

AGRADECIMENTOS

Agradeço primeiramente a Deus, por estar sempre ao meu lado, me dando sabedoria e discernimento.

Aos meus pais amados, pelo amor, apoio e compreensão da minha ausência física em muitas datas importantes durante estes dois anos.

Aos meus irmãos, sobrinha, avós e tios pelo incentivo e confiança nas minhas decisões.

Aos meus orientadores, Prof^a. Diana Francisca Adamatti e Prof. Leonardo Ramos Emendorfer, pela amizade, contribuição para o meu crescimento acadêmico, paciência e pela disponibilidade sempre quando precisava.

Aos meus amigos, principalmente a Ana Paula, Mariely e Nitiele, pela convivência, companhia e conselhos em todos os momentos.

Agradeço também, a CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pelo apoio financeiro, colaborando assim para a realização da pesquisa, e a Universidade Federal do Rio Grande (FURG) pelo oportunidade de cursar o Mestrado em Modelagem Computacional.

RESUMO

Os Algoritmos de *Clustering* possuem como um de seus principais objetivos a formação de grupos de forma que os objetos de dados pertencentes a esse grupos são semelhantes entre si. Muitas técnicas de agrupamento foram propostas em trabalhos encontrados na literatura nas áreas de Mineração de Dados e Estatística, a maioria delas se baseia em informações *a priori* para obter os resultados, como o número desejado de grupos. Assim, este trabalho apresenta a implementação e avaliação de um Algoritmo de *Clustering* baseado nas características de agentes, que detecta o número de grupos para um determinado conjunto de dados. Os grupos formados durante o processo de agrupamento são, assim, padrões emergentes da interação entre agentes. Dessa forma, o algoritmo é testado para diferentes conjuntos de dados, além de sua comparação com algoritmos de agrupamento *K-means* e *DBSCAN*, e seus resultados validados utilizando as seguintes formulações matemáticas definidas para esse fim: *Silhouette*, *Davies Bouldin*, *Dunn*, *Dunn Generalizado* e *DBC*.

Palavras-chaves: Algoritmos de *Clustering*, Sistemas Multiagente, Validação de Agrupamentos.

ABSTRACT

The Clustering Algorithms have as one of their main objectives the formation of groups so that the data objects belonging to these groups are similar to each other. Many grouping techniques have been proposed in works found in the literature in the areas of Data Mining and Statistics, most of them are based on a priori information to obtain the results, such as the desired number of groups. Thus, this work presents the implementation and evaluation of a Clustering Algorithm based on the characteristics of agents, which detects the number of groups for a given set of data. The groups formed during the clustering process are thus emerging patterns of agent interaction. Thus, the algorithm is tested for different data sets, in addition to its comparison with clustering algorithms *K-means* and *DBSCAN*, and its results validated using the following mathematical formulations defined for this purpose: *Silhouette*, *Davies Bouldin*, *Dunn*, *Dunn Generalized* e *DBC*V.

Keywords: Clustering Algorithms, Multi-Agent System, Cluster Validation.

ÍNDICE

1	INTRODUÇÃO	11
1.1	Objetivos	12
1.1.1	Objetivo Geral	12
1.1.2	Objetivos Específicos	12
1.2	Estrutura do trabalho	12
2	REFERENCIAL TEÓRICO	13
2.1	Mineração de Dados	13
2.2	Agrupamento de Dados	14
2.3	Técnicas de Agrupamentos de Dados	15
2.3.1	<i>K-means</i>	15
2.3.2	<i>DBSCAN</i>	17
2.4	Índices de Validação de Agrupamentos	19
2.4.1	<i>Silhouette</i>	19
2.4.2	<i>Davies-Bouldin</i>	20
2.4.3	<i>Dunn</i>	20
2.4.4	<i>Dunn Generalizado</i>	21
2.4.5	<i>DBC</i> V	24
2.4.6	F1-Score	25
2.5	Sistemas Multiagente	26
2.6	Trabalhos Relacionados	29
3	ALGORITMO DESENVOLVIDO	33
3.1	O Algoritmo <i>DL3</i>	33
3.2	Metodologia Adotada	35
4	ANÁLISE DOS RESULTADOS	40
4.1	Comparação com o <i>K-means</i>	41
4.2	Comparação com o <i>DBSCAN</i>	47
5	CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS	53
6	REFERÊNCIAS	54

LISTA DE FIGURAS

Figura 2.1: Algoritmo <i>K-means</i>	16
Figura 2.2: Exemplo do comportamento do Algoritmo <i>K-means</i>	17
Figura 2.3: <i>DBSCAN</i>	18
Figura 2.4: Algoritmo <i>DBSCAN</i> em um conjunto de 3000 dados e duas dimensões. . .	18
Figura 2.5: Exemplos de Grafos.	21
Figura 2.6: Grafos de vizinhança relativa (RGN).	22
Figura 2.7: Grafos de Gabriel (GG).	22
Figura 2.8: Árvore Geradora Mínima (MST).	23
Figura 2.9: Representação genérica de agente.	28
Figura 3.1: Representação de como cada objeto analisa os setores de visão.	34
Figura 3.2: <i>DL3</i>	35
Figura 3.3: Etapas do trabalho	36
Figura 3.4: Interface do NetLogo	37
Figura 3.5: Conjuntos de dados utilizados no trabalho.	39
Figura 3.6: Exemplo do comportamento do <i>DL3</i> em sua aplicação no conjunto de dados <i>Lsun</i>	40
Figura 4.1: Agrupamentos gerados para os conjuntos de dados <i>Aggregation</i> e <i>Jain</i> , e os seus respectivos valores de <i>F1-Score</i> , considerando os algoritmos <i>K-means</i> e <i>DL3</i>	44
Figura 4.2: Agrupamentos gerados para os conjuntos de dados <i>Lsun</i> e <i>Target</i> , e os seus respectivos valores de <i>F1-Score</i> , considerando os algoritmos <i>K-means</i> e <i>DL3</i>	45
Figura 4.3: Agrupamentos gerados para os conjuntos de dados <i>Face</i> e <i>Flame</i> , e os seus respectivos valores de <i>F1-Score</i> , considerando os algoritmos <i>K-means</i> e <i>DL3</i>	46
Figura 4.4: Agrupamentos gerados para os conjuntos de dados <i>Aggregation</i> e <i>Jain</i> , e os seus respectivos valores de <i>F1-Score</i> , considerando os algoritmos <i>DBSCAN</i> e <i>DL3</i>	49
Figura 4.5: Agrupamentos gerados para os conjuntos de dados <i>Lsun</i> e <i>Target</i> , e os seus respectivos valores de <i>F1-Score</i> , considerando os algoritmos <i>DBSCAN</i> e <i>DL3</i>	50
Figura 4.6: Agrupamentos gerados para os conjuntos de dados <i>Face</i> e <i>Flame</i> , e os seus respectivos valores de <i>F1 - Score</i> , considerando os algoritmos <i>DBSCAN</i> e <i>DL3</i>	51

LISTA DE TABELAS

Tabela 1:	Alguns Critérios de Validação de Clusters	27
Tabela 2:	Tabela resumo de trabalhos relacionados	31
Tabela 3:	Valores para os índices	43
Tabela 4:	Valores para o índice <i>DBC</i> V	48

1 INTRODUÇÃO

A ideia primitiva de agrupamento surge da união de objetos que apresentam algum tipo de semelhança, sendo desde os primórdios uma das habilidades básicas dos seres humanos. Passando para o viés científico se tem o conceito de Mineração de Dados, sendo um processo de obtenção de conhecimentos a partir de uma base de dados de modo que o agrupamento de dados é uma das suas tarefas mais importantes, com muitos estudos nas áreas da computação e estatística, se caracterizando assim, como uma área propícia para pesquisas (Tan; Steinbach; Kumar, 2006). Dessa forma, *Clustering* ou Agrupamento de Dados tornou-se um tema cada vez mais importante nos últimos anos, devido ao grande número de dados utilizados em diversas áreas como biologia, medicina, psicologia e processamento de imagens.

Os algoritmos de agrupamentos de dados particionam objetos de dados (padrões, entidades, instâncias, observações, unidades) em um número de *clusters* (grupos, subconjuntos ou categorias) (Xu; Wunsch, 2009). Devido a esse grande número de aplicações, dificilmente se encontrará uma técnica de *Clustering* aplicável de forma satisfatória a todos os tipos de dados, nos diferentes contextos. Logo, existem muitos algoritmos para este fim, sendo exemplos as técnicas descritas nos trabalhos de MacQueen (1967) e Ester et al. (1996).

É importante salientar que o uso dessas diferentes técnicas em um mesmo conjunto de dados podem gerar resultados finais discordantes. Uma justificativa para essa ocorrência de soluções diversas é o fato que cada algoritmo possui procedimentos de agrupamento próprios. Outro fator influenciador, na geração de agrupamentos de alguns algoritmos, é a definição de parâmetros de entrada, como o número desejado de grupos ou a fixação de objetos centrais para caracterizar cada grupo da amostra. Logo, esses fatores prejudicam a formalização de uma única metodologia capaz de realizar essa tarefa.

Diante das afirmações anteriores, existe a necessidade de desenvolvimento de novas técnicas com um número pequeno de parâmetros, pois algoritmos com muitos dados de entrada pode gerar alto tempo computacional ate encontrar a melhor solução de agrupamento. Assim, muitas pesquisas focam em apresentar métodos para encontrar valores “ótimos” para essas informações prévias. Nesse trabalho, por sua vez, se propõe a investigação, implementação e a avaliação de um algoritmo de agrupamento que parte dessas justificativas. Para o cumprimento dos objetivos propostos essa nova técnica se baseia, resumidamente, na utilização de um Sistema Multiagente para solucionar problemas típicos em agrupamento de dados.

Os Sistemas Multiagente são compostos por múltiplos elementos computacionais que interagem, conhecidos como agentes. Eles, por sua vez, possuem a capacidade de ação autônoma, de decidir por si mesmos o que precisam fazer para satisfazer seus objetivos e a capacidade de interagir com outros agentes (Wooldridge, 2002). Assim, é justamente essa possibilidade de interação, cooperação e troca de informações entre agentes, e principalmente de autonomia que a técnica de agrupamento proposta se baseia, no desenvolvimento de um algoritmo capaz de realizar a auto-organização de diferentes conjuntos de objetos de dados.

No algoritmo implementado, cada agente (objeto de dado) realiza certa tarefa com alguns critérios definidos para formar grupos com outros agentes, baseado apenas em sua localização espacial e sua capacidade de “enxergar” seus vizinhos. O critério principal utilizado é que o agente escolha o grupo com o maior número de agentes para se agrupar de acordo com um raio de visão definida.

As principais etapas do trabalho desenvolvido são: desenvolvimento do algoritmo e a sua implementação, testes utilizando diferentes conjuntos de dados; e por fim a fase de avaliação usando alguns critérios de validação de *clusters* encontrados na literatura e a comparação com outros algoritmos populares em realizar também a tarefa de *Clustering*.

1.1 Objetivos

1.1.1 Objetivo Geral

Desenvolver um algoritmo de agrupamento de dados de baseado nas características de agentes, apresentando-o sua dinâmica e progresso, e avaliando seu desempenho com algoritmos conhecidos e validação dos resultados obtidos através de diferentes formulações matemáticas encontradas na literatura.

1.1.2 Objetivos Específicos

- Desenvolver um algoritmo de agrupamento de dados de forma automatizada com poucos parâmetros.
- Investigar a nova técnica, demonstrando a eficiência do algoritmo em problemas de agrupamento.
- Comparar o algoritmo desenvolvido com as técnicas populares em Mineração de Dados, *K-means* e *DBSCAN* (*Density Based Spatial Clustering of Applications with Noise*).
- Validar os agrupamentos gerados pelo algoritmo através do cálculo dos índices, *Silhouette*, *Davies Bouldin*, *Dunn*, *Dunn Generalizado* e *DBC*.

1.2 Estrutura do trabalho

A dissertação está estruturada da seguinte maneira: no primeiro Capítulo apresentam-se os objetivos do trabalho e a justificativa para realização da pesquisa; no Capítulo 2 é exposto o referencial teórico e trabalhos relacionados com o tema; no Capítulo 3 se tem a descrição da metodologia seguida para o cumprimento dos objetivos fixados. No Capítulo 4 são apresentados os resultados finais após a execução de todas as atividades previstas. E por fim, no Capítulo 5 são relatados as considerações finais abrangendo as vantagens e desvantagens da nova técnica; e os trabalhos futuros, destacando possíveis modificações e outras aplicações para o algoritmo.

2 REFERENCIAL TEÓRICO

Neste capítulo, são abordados conceitos relacionados ao desenvolvimento do trabalho, com o intuito de facilitar a compreensão dos principais temas contemplados pela proposta apresentada.

2.1 Mineração de Dados

Historicamente, a noção de encontrar padrões úteis em dados tem recebido uma variedade de nomes, incluindo mineração de dados, extração de conhecimento, descoberta de informações, colheita de informações, arqueologia de dados e processamento de padrões de dados. O termo mineração de dados tem sido usado principalmente por estatísticos, analistas de dados e as comunidades de Sistemas de Informação de Gestão (MIS - *Management Information Systems*). Também ganhou popularidade no campo do banco de dados (Fayyad; Piatetsky-Shapiro; Smyth, 1996).

A mineração de dados é definida como o processo de descobrir padrões novos, perceptivos e relevantes, bem como modelos preditivos e descritivos a partir de dados em larga escala (Zaki; Meira JR., 2014). De acordo com essa definição, a mineração de dados apresenta os dois objetivos primários na prática: a previsão e a descrição. A previsão envolve o uso de algumas variáveis ou campos no banco de dados para prever valores desconhecidos ou futuros de outras variáveis de interesse; e a descrição se concentra em encontrar padrões descrevendo os dados e é seguida pela apresentação ao usuário para a realização da interpretação (Fayyad; Piatetsky-Shapiro; Smyth, 1996).

Os pontos de vista algébricos, geométricos e probabilísticos dos dados desempenham um papel fundamental na mineração de dados. Dado um conjunto de dados de n pontos em um espaço d -dimensional, a análise fundamental e as tarefas de mineração abordadas incluem: a análise exploratória de dados, a mineração de padrões frequentes, agrupamento de dados ou *Clustering* e modelos de classificação. A seguir são definidas essas tarefas, de acordo com Zaki; Meira JR. (2014):

- **Análise exploratória de dados:** Tem como objetivo explorar os atributos numéricos e categóricos dos dados individualmente ou em conjunto para extrair características-chaves da amostra de dados através de estatísticas que fornecem informações sobre a centralidade e dispersão, por exemplo.
- **Mineração de padrões frequentes:** Refere-se à tarefa de extrair padrões informativos e úteis em conjuntos de dados maciços e complexos. Padrões compreendem conjuntos de valores de atributos co-ocorrentes, chamados conjuntos de itens, ou padrões mais complexos, tais como sequências, que consideram relações de precedência explícitas (posicionais ou temporais) e gráficos que consideram relações arbitrárias entre pontos. O objetivo principal é descobrir tendências e comportamentos ocultos nos dados para entender melhor as interações entre os pontos e atributos.

- **Agrupamento de dados ou *Clustering*:** É a tarefa de particionar os pontos em grupos naturais chamados *clusters*, de tal forma que os pontos dentro de um grupo são muito semelhantes, enquanto que os pontos em *clusters* distintos são tão dissimilares quanto possível.
- **Classificação:** A tarefa de classificação é prever o rótulo ou a classe para um determinado ponto não marcado. Formalmente, um classificador é um modelo ou função M que prediz o rótulo de classe \hat{y} para um dado exemplo de entrada x , isto é, $\hat{y} = M(x)$, onde $\hat{y} \in \{c_1, c_2, \dots, c_k\}$ e cada c_i é um rótulo de classe (um valor de atributo categórico). Para construir o modelo precisamos de um conjunto de pontos com seus rótulos de classe corretos, que é chamado de conjunto de treinamento. Depois de aprender o modelo M , podemos prever automaticamente a classe para qualquer novo ponto. Muitos tipos diferentes de modelos de classificação foram propostos, como árvores de decisão, classificadores probabilísticos e máquinas de vetores de suporte.

Neste trabalho, a ênfase é na técnica de Agrupamento de Dados. Desta forma, na próxima seção, o conceito é abordado mais detalhadamente.

2.2 Agrupamento de Dados

Provavelmente, agrupamento de dados é um dos problemas mais amplamente estudados pelas comunidades de mineração de dados e aprendizagem de máquina por causa de suas numerosas aplicações. O problema básico de agrupamento de dados pode ser definido da seguinte forma : *Dado um conjunto de pontos de dados, divida-os em um conjunto de grupos que são tão semelhantes quanto possível* (Aggarwal; Reddy, 2014).

Porém, essa definição de problema pode sofrer algumas alterações dependendo do modelo específico e/ou do tipo de dado utilizado. Por exemplo, um modelo generativo pode definir similaridade com base em um mecanismo generativo probabilístico, enquanto que uma abordagem baseada em distância utilizará uma função de distância tradicional para quantificação (Aggarwal; Reddy, 2014).

Basicamente, o objetivo dos algoritmos dessa tarefa de mineração de dados é agrupar dados em um certo número de *clusters* (que podem ser grupos, subconjuntos ou categorias), sendo que os padrões no mesmo grupo devem ser semelhantes entre si, enquanto os padrões em diferentes clusters não devem ser. Tanto a semelhança como a dissimilaridade devem ser investigadas de forma clara e significativa (Xu; Wunsch, 2009).

A seguir são expostas algumas descrições matemáticas simples de vários tipos de agrupamento, com base nas discussões de Hansen; Jaumard (1996).

Considere uma amostra $O = \{O_1, O_2, \dots, O_N\}$ de N entidades entre as quais os *clusters* devem ser encontrados. Logo, os tipos de agrupamento de dados que podem ser identificados pelos algoritmos são:

- (i) Subconjunto C de O ;

(ii) Partição $P_M = \{C_1, C_2, \dots, C_M\}$ de O em M grupos:

$$(a) C_j \neq \emptyset \quad j = 1, 2, \dots, M;$$

$$(b) C_i \cap C_j = \emptyset \quad i, j = 1, 2, \dots, M \text{ e } i \neq j$$

$$(c) \bigcup_{i=1}^M C_j = O$$

(iii) Empacotamento (Packing) $Pa_M = \{C_1, C_2, \dots, C_M\}$ de O com M grupos como (ii) mas sem considerar (c);

(iv) Cobertura $Co_M = \{C_1, C_2, \dots, C_M\}$ de O por M clusters como (ii) mas sem considerar (b);

(v) Hierarquia $H = \{P_1, P_2, \dots, P_q\}$ de $q \leq N$ partições de O .

Conjunto de partições P_1, P_2, \dots, P_q de O tal que $C_i \in P_k, C_i \in P_l$ e $k > l$ implica $C_i \subset C_j = \emptyset$ ou $C_i \cap C_j = \emptyset$ para todo $i, j \neq i, k, l = 1, 2, \dots, q$.

Entre esses tipos de agrupamentos os mais usados são os particionais e hierárquicos. Além dos citados ainda, existem os métodos de agrupamento fuzzy, baseados na Teoria dos Conjuntos Fuzzy (Zadeh, 1965), que permitem que os objetos pertençam a vários grupos simultaneamente, com diferentes graus de associação, ou seja, os objetos nos limites entre várias classes não são forçados a pertencer completamente a uma das classes, mas sim são atribuídos graus de associação entre 0 e 1 indicando sua associação parcial.

Xu; Wunsch (2009) descrevem o procedimento de análise de agrupamento de dados em quatro passos básicos: Seleção ou extração de recursos, Projeto ou seleção do algoritmo de agrupamento; Validação dos grupos; Interpretação dos resultados. É necessário enfatizar que a análise de agrupamento não é um processo de direção única. Em muitas circunstâncias, é necessário uma série de testes e repetições.

Nas próximas seções são apresentados alguns algoritmos de agrupamento de dados mais comumente encontrados na literatura e que servirão de comparação com a nova proposta descrita neste trabalho.

2.3 Técnicas de Agrupamentos de Dados

Os algoritmos de agrupamentos dados são utilizados em muitas aplicações de diferentes áreas, de forma que as heurísticas desenvolvidas normalmente consideram apenas alguns problemas específicos. Assim, se torna difícil encontrar uma técnica genérica para todas as aplicações.

Apresenta-se aqui, duas dessas técnicas bastante tradicionais em agrupamento de dados, utilizados para fins de comparação com o algoritmo descrito nesse trabalho.

2.3.1 K-means

O Algoritmo *K-means*, foi primeiramente empregado por MacQueen (1967) e é um dos mais conhecidos algoritmos de agrupamento de dados. O *K-means* trata-se de uma técnica de

agrupamento parcial que tenta encontrar um número de grupos k , especificado pelo usuário, que são representados por seus centroides (Tan; Steinbach; Kumar, 2006).

Para o cálculo da medida de similaridade, utilizado pelo método, pode ser utilizada a tradicional Distância Euclidiana, ou outra, como as Distâncias de Manhattan e de Chebyshev. Essa medida de similaridade, entre o centroide e o dado, define a qual grupo o respectivo dado pertencerá. Um dos critérios de parada para o algoritmo que pode ser considerado é a não variação dos dados entre os grupos, em outras palavras, quando o valor dos centroides não variarem mais, então o método encontrou um resultado convergente.

Logo, o algoritmo *K-means* visa minimizar uma função objetivo, nessa conjuntura uma função de erro quadrático. Na equação (2.1) tem-se a função objetivo a ser minimizada.

$$\min = \sum_{i=1}^k \sum_{j=1}^n d(X_i, C_j)^2 \quad (2.1)$$

Onde $d(X_i, C_j)$ é a distância entre um i -ésimo dado (X_i) e o centroide do j -ésimo cluster (C_j), n é o número de instâncias e k é o número de grupos considerados no algoritmo que deverá ser escolhido antes da sua execução.

Existem vários algoritmos que representam o método *K-means*, a Figura 2.1 apresentam uma dessas representações encontrado em Tomasini (2015). Além disso, a Figura 2.2 mostra um exemplo do comportamento do algoritmo:

Entrada: Um conjunto de dados $X_{n \times d}$ e o Número de *clusters* k desejado.

Saída: Uma partição X em k grupos.

Escolher aleatoriamente k valores para centroides dos grupos.

Início

repita

para cada $x_i \in X$ e *clusters* C_j , $j = 1, \dots, k$ **faça**

 | Calcular a distância entre x_i e o centroide do grupo $\bar{x}^j : d(x_i, \bar{x}^j)$

fim

para cada objeto x_i **faça**

 | Associar x_i com o centroide mais próximo

fim

para cada grupo C_j , $j = 1, \dots, k$ **faça**

 | Recalcular o centroide

fim

até;

Não haver mais alteração na associação.

Fim

Figura 2.1: Algoritmo *K-means*

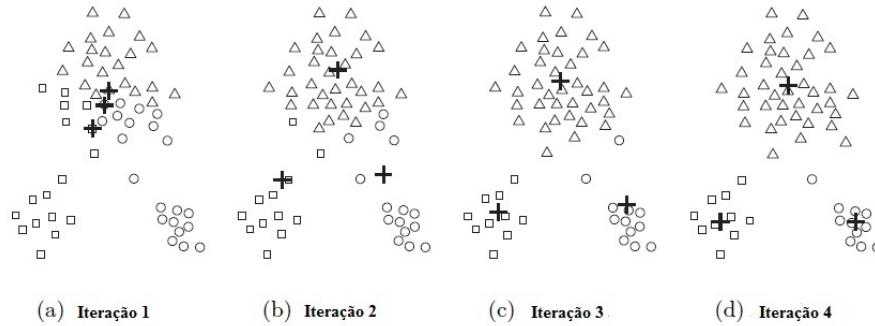


Figura 2.2: Exemplo do comportamento do Algoritmo K -means.
(Tan; Steinbach; Kumar, 2006)

A popularidade do algoritmo K -means é explicada pela baixa complexidade, agilidade de desenvolvimento e a fácil compreensão das propriedades matemáticas, possuindo um tempo de função linear em função do tamanho do conjunto de padrões. Porém, esse método também possui algumas desvantagens, como a necessidade de estabelecer *a priori* o número de classes. Dessa forma, o K -means produz uma solução baseada na escolha inicial dos centroides que representam cada grupo. Em relação ao tempo de execução, ele está relacionado de forma direta com o conjunto de dados em questão e a escolha do parâmetro k sendo um fator importante para o sucesso do desempenho final do algoritmo.

2.3.2 DBSCAN

O *DBSCAN* - *Density Based Spatial Clustering of Applications with Noise* (Agrupamento Espacial Baseada em Densidade de Aplicações com Ruído) fundamenta-se em uma noção baseada em densidade de grupos, que é projetada para descobrir *clusters* de forma arbitrária (Ester et al., 1996). Nesse contexto, a densidade se define como o número de pontos dentro de um raio específico, Eps , que é um parâmetro do algoritmo, além do $MinPts$ que é o número mínimo de pontos. Baseia-se na abordagem centralizada, onde a densidade é estimada para um ponto específico no conjunto de dados contando o número de pontos dentro do raio especificado Eps . Isso permite classificar um ponto como central, de fronteira e ou ruído. Resumidamente, a ideia principal do algoritmo é que para cada ponto de um *cluster* a vizinhança de acordo com um raio (Eps) pré-determinado, deve conter pelo menos um número mínimo de pontos ($MinPts$) (Ahmed; Razak, 2016).

A representação do algoritmo *DBSCAN*, apresentado na Figura 2.3, é encontrada em Tomasi (2015) e a Figura 2.4 exemplifica a aplicação do *DBSCAN* em um conjunto de dados. Desta forma, o *DBSCAN* produz um agrupamento particional, em que o número de grupos é determinado automaticamente pelo algoritmo, sendo que pontos em regiões de baixa densidade são omitidos, pois são classificados como ruídos. Assim, o *DBSCAN* não produz um agrupamento completo (Tan; Steinbach; Kumar, 2006).

Entrada: Um conjunto de dados X e os parâmetros Eps e $MinPts$.

Saída: Uma partição X em k grupos.

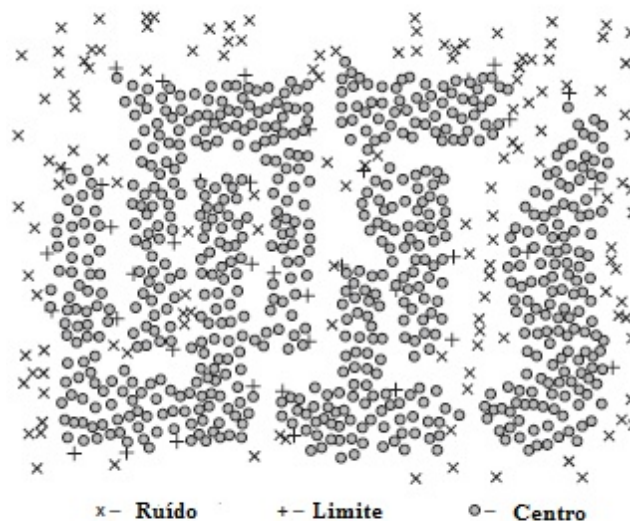
Início

1. Classificar todas as instâncias como objetos do tipo: centro, limite ou ruído;
2. Eliminar os objetos rotulados como ruído;
3. Colocar uma aresta entre todos os objetos de centro que estejam dentro do Eps uns dos outros;
4. Tornar cada grupo de Objetos de centro um grupo separado;
5. Atribuir cada objeto limite a um dos grupos dos seus objetos de centro associados;

Figura 2.3: *DBSCAN*



(a) Clusters encontrados pelo DBSCAN



(b) Pontos de Ruídos, Limites e Centros.

Figura 2.4: Algoritmo DBSCAN em um conjunto de 3000 dados e duas dimensões.
Tan; Steinbach; Kumar (2006)

2.4 Índices de Validação de Agrupamentos

O problema principal na tarefa de *Clustering* é a definição do número “ótimo” de grupos e a análise do resultado final. Existem alguns algoritmos que procuram a definição automática desse número de grupos. Porém, muitas vezes dependem do ajustamento de um parâmetro limiar. Dessa forma, é comum validar os resultados gerados por algum algoritmo de agrupamento, dado um conjunto de dados. A validação consiste em quantificar a qualidade desses resultados, considerando em muitos casos as distâncias intra e inter-grupos, com a pretensão de encontrar compactabilidade e separabilidade nos grupos.

Existem três abordagens para analisar a validação de um agrupamento. A primeira é baseada em critérios externos. Isso implica que avalia-se os resultados de um algoritmo de agrupamento baseado em uma estrutura pré-especificada, que é imposta a um conjunto de dados e reflete a uma intuição sobre a estrutura de agrupamento do conjunto de dados considerado. A segunda abordagem baseia-se em critérios internos, a qual avalia os resultados de um algoritmo de agrupamento em termos de quantidades que envolvem os vetores do próprio conjunto de dados (por exemplo, a matriz de proximidade). A terceira abordagem é baseada em critérios relativos, sendo a ideia básica a avaliação de uma estrutura de agrupamento, comparando-a com outros esquemas de agrupamento, resultante do mesmo algoritmo mas com diferentes valores de parâmetro (Halkidi; Batistakis; Vazirgiannis, 2001).

Na sequência são apresentadas algumas abordagens de índice para medir a validade dos grupos formados por algoritmos de agrupamento propostas na literatura e que serviram de referência para analisar os resultados desta pesquisa.

2.4.1 *Silhouette*

Proposto por Rousseeuw (1987) o Índice *Silhouette* de validação de consistência no interior de agrupamentos de dados é dado pela formulação da equação (2.2).

$$s(v_i) = \frac{d(v_i, C_h) - d(v_i, C_j)}{\max(d(v_i, C_j), d(v_i, C_h))} \quad (2.2)$$

Onde:

- $d(v_i, C_h)$ é a dissimilaridade mínima do objeto v_i em relação a todos os outros objetos do grupo mais próximo C_h (vizinho do objeto v_i).
- $d(v_i, C_j)$ é a dissimilaridade média do objeto v_i em relação a todos os outros objetos do grupo C_j , que contém v_i .
- $s(v_i)$ varia entre o intervalo fechado $[-1, 1]$, de tal forma que mais próximo de 1, melhor a alocação do objeto no grupo.

Após esse cálculo para os pontos de dados do agrupamento, é necessário calcular a média para o grupo (S_j) e para o agrupamento como um todo (GS), as equações são apresentadas na equações (2.3) e (2.4).

$$S_j = \frac{\sum_{i=1}^{N_j} s(v_i)}{N_j} \quad (2.3)$$

$$GS = \frac{\sum_{j=1}^k S_j}{K} \quad (2.4)$$

Onde:

- N_j é o número de pontos do grupo j .
- K é o número de grupos

2.4.2 *Davies-Bouldin*

O Índice *Davies-Bouldin* (Davies; Bouldin, 1979) avalia a separação dos clusters formados pelas técnicas agrupamentos de dados, através da equação (2.5).

$$DB = \frac{1}{K} \sum_{i=1}^K \max_{j \neq i} \left[\frac{\text{diam}(C_i) + \text{diam}(C_j)}{d(C_i, C_j)} \right] \quad (2.5)$$

Onde:

- $\text{diam}(C_i)$ distância média dentro do grupo C_i .
- $\text{diam}(C_j)$ distância média dentro do grupo C_j .
- $d(C_i, C_j)$ distância inter-clusters desses grupos.

Nesse caso, quanto menor o valor do índice, melhor o resultado, significando, baixas medidas de dispersão intragrupo e grandes distâncias intergrupo.

2.4.3 *Dunn*

Introduzido por Dunn (1973), o Índice *Dunn* valida os resultados dos algoritmos de agrupamento de dados a partir da equação (11).

$$Dn = \min_{1 \leq i \leq q} \left\{ \min_{1 \leq j \leq q \text{ e } j \neq i} \left(\frac{d(C_i, C_j)}{\max_{1 \leq h \leq q} \{\text{diam}(C_h)\}} \right) \right\} \quad (2.6)$$

Onde:

- q é o número de grupos no agrupamento.
- $d(C_i, C_j)$ representa a distância entre os grupos C_i e C_j .
- $diam(C_h)$ é o diâmetro de um grupo, o que pode ser considerado como uma medida de dispersão de agrupamento.

O índice se caracteriza na comparação das distâncias intergrupos com o tamanho do grupo mais disperso. O valor resultante está no intervalo $[0, \infty)$, de tal modo que quanto maior o valor obtido mais separados e compactos são os grupos.

2.4.4 *Dunn Generalizado*

Pal; Biswas (1997) apresentam uma generalização do Índice *Dunn* usando algumas estruturas de grafos. Ao contrário do Índice de *Dunn* tradicional, a reformulação não é sensível a pontos ruidosos e pode ser aplicável a *clusters* hiperesféricos (induzidos quando a medida de distância utilizada pelo algoritmo é a Norma Euclidiana) e estruturais.

Para a explicação do Índice *Dunn Generalizado* é importante salientar, primeiramente, alguns conceitos da Teoria de Grafos que são utilizados na sua formulação. Tais conceitos são:

- **Grafos:** Seja V um conjunto finito e não vazio, e E uma relação binária sobre V . Os elementos de V são representados por pontos. O par ordenado $(v, w) \in E$, também podendo ser denotado como vw , de forma que $v, w \in V$, é representado por uma linha que liga v a w . Tal representação de um conjunto V e uma relação binária sobre o mesmo é denominada um grafo $G(V, E)$ (Rabuske, 1992). Os elementos de V são denominados vértices e os pontos ordenados de E denominam-se arestas dos grafos. A Figura 2.5 mostra dois exemplos de grafos (a) e (b), sendo que em (a) se tem $V = \{a, b, c, d\}$ e $E = \{ab, ac, ad, bc, bd, cd\}$, e em (b), tem-se $V = \{a, b, c, d\}$ e $E = \{ad, ad, db, dc, cc\}$.

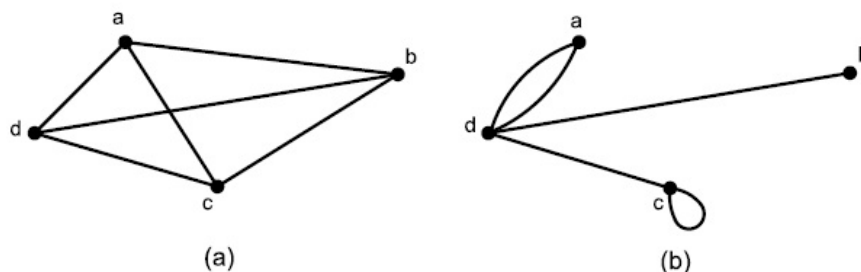


Figura 2.5: Exemplos de Grafos.
Costa (2011)

- **Grafo de Vizinhaça Relativa (RNG):** Dois pontos x_i e x_j são conectados em RNG se $d(x_i, x_j) \leq \max\{d(x_i, x_k), d(x_j, x_k)\}$ para todo $k, k \neq i; k \neq j$, isto é, x_i e x_j estão

conectados em RGN se nenhum outro ponto cair em $LUNE(x_i, x_j)$, onde $LUNE(x_i, x_j)$ é a intersecção de dois discos de raios $d(x_i, x_j)$ e com centros em x_i e x_j . Na Figura 2.6, (a) mostra $LUNE(x_i, x_j)$ e (b) apresenta o RGN de (c).

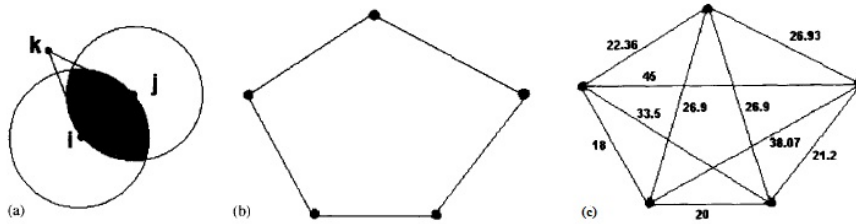


Figura 2.6: Grafos de vizinhança relativa (RGN).
Pal; Biswas (1997)

- **Grafo de Gabriel:** Dois pontos x_i e x_j são conectados em GG se $d^2(x_i, x_j) < d^2(x_i, x_k) + d^2(x_j, x_k)$ para todo $k, k \neq i; k \neq j$ onde $d(x_i, x_j)$, é a distância Euclidiana entre x_i e x_j , isto é, x_i e x_j estão conectados em GG se não houver outros pontos em $DISK(x_i, x_j)$, onde $DISK(x_i, x_j)$ é o disco com diâmetro $d(x_i, x_j)$ centrado no ponto médio de x_i e x_i . Na Figura 2.7, (a) mostra $DISK(x_i, x_j)$ para um par (x_i, x_j) e (b) apresenta o GG de (c).

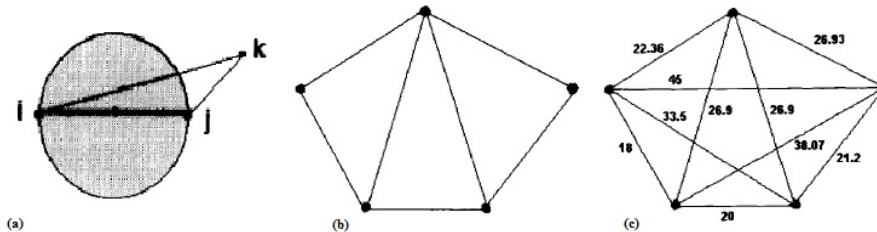


Figura 2.7: Grafos de Gabriel (GG).
Pal; Biswas (1997)

- **Árvore Geradora Mínima (MST):** Uma árvore geradora em um grafo G é um subgrafo mínimo que conecta todos os vértices de G . Se G é um grafo ponderado (isto é, há um número real associado a cada extremidade de G), então o peso da árvore geradora T de G é definido como a soma dos pesos de todos os ramos em T . Em geral, diferentes árvores geradoras de G terão pesos diferentes. Uma árvore geradora com o menor peso em um grafo ponderado é chamado de **Árvore Geradora Mínima (MST)**. Na Figura 2.8, (b) representa um MST do grafo (a).

Logo, os métodos propostos são descritos da seguinte maneira:

Suponha um conjunto de dados $X \in R^p$ agrupado em c classes, X_1, X_2, X_c . Para definir índices de validade dos *clusters*, é necessário definir o diâmetro de uma classe e a separação entre classes. Assim, estes índices serão definidos em termos de arestas dos três tipos de grafos mencionados anteriormente. A seguir, os diferentes índices são definidos em termos de GG, mas os índices podem ser escritos usando MST e RNG de forma similar.

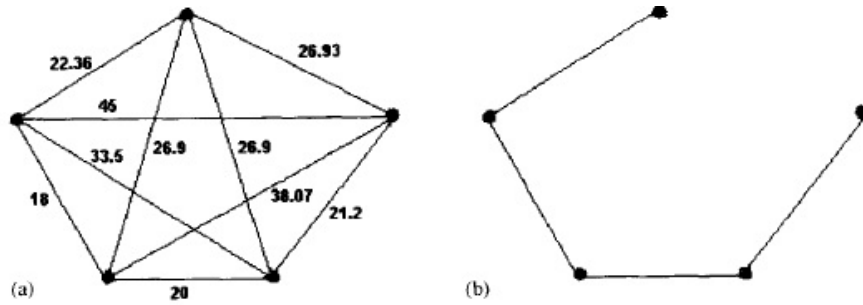


Figura 2.8: Árvore Geradora Mínima (MST).
Pal; Biswas (1997)

Seja $E_i^{GG} = \{e_{ij}^{GG}, 1 \leq j \leq l_i, \text{ onde } l_i \text{ é } |E_i|; 1 \leq i \leq c\}$ o conjunto de arestas do GG calculado em X_i .

O diâmetro d_i^{GG} do *cluster* X_i é definido conforme equação (2.7).

$$d_i^{GG} = \max_j \{e_{ij}^{GG}, j = 1, 2, \dots, l_i\}. \quad (2.7)$$

O máximo de todos os diâmetros representa a extensão máxima possível D de todos os *clusters* na divisão, isto é, $D^{GG} = \max\{d_i, 1 \leq i \leq c\}$.

É notável que para uma boa partição, D terá um valor menor em comparação com uma partição ruim. Se dois *clusters* separados forem mesclados em uma partição, então D será maior para essa partição. O diâmetro de um conjunto como definido no índice *Dunn* tradicional é fortemente influenciado pela presença de pontos ruidosos, enquanto não há muito efeito de pontos ruidosos na equação (2.7).

A separação $d_{ij} = \text{dist}(X_i, X_j)$ entre dois *clusters* X_i e X_j é definida pela distância entre os centros dos *clusters* das classes i e j , conforme equação (2.8).

$$d_{ij} = \|v_i - v_j\| \quad (2.8)$$

Pode-se definir um índice de validação de *clusters* considerando a equação (2.9):

$$Dn_{GG} = \min_{1 \leq i \leq c} \left\{ \min_{1 \leq j \leq c \text{ e } j \neq i} \left\{ \frac{d_{ij}}{\max_{1 \leq k \leq c} \{d_k^{GG}\}} \right\} \right\} \quad (2.9)$$

Da mesma forma, são definidos dois outros índices considerando as estruturas geométricas de grafos RNG e MST (equações 2.10 e 2.11).

$$Dn_{RGN} = \min_{1 \leq i \leq c} \left\{ \min_{1 \leq j \leq c \text{ e } j \neq i} \left\{ \frac{d_{ij}}{\max_{1 \leq k \leq c} \{d_k^{RGN}\}} \right\} \right\} \quad (2.10)$$

$$Dn_{MST} = \min_{1 \leq i \leq c} \left\{ \min_{1 \leq j \leq c \text{ e } j \neq i} \left\{ \frac{d_{ij}}{\max_{1 \leq k \leq c} \{d_k^{MST}\}} \right\} \right\} \quad (2.11)$$

2.4.5 DBCV

O trabalho proposto por Moulavi et.al (2014) apresenta um índice de validação relativo para *clusters* baseados em densidade, arbitrariamente formados, denominado *Density Based Clustering Validation (DBCV)*. O índice avalia a qualidade do agrupamento com base na conexão de densidade relativa entre pares de objetos. Para esse fim, emprega o conceito do modelo de árvores de contorno de densidade proposto em Hartigan (1975). Nesse modelo, se os objetos são pontos distribuídos em um espaço N -dimensional (uma dimensão para cada variável), os *clusters* são definidos como regiões de alta densidade separadas de outras regiões através das regiões de baixa densidade.

Para a formulação do *DBCV* é necessário definir as seguintes ideias: $a_{ptscoredist}$ é definido como a inversa da densidade de cada objeto em relação a todos os outros objetos dentro de seu *cluster*, através disso é definida uma distância de alcance simétrica que é então empregada para construir uma Árvore Geradora Mínima (MST) - como discutido na seção 2.4.4 - dentro de cada *cluster*. O uso da MST detecta a forma e a densidade de cada *cluster*, pois é construído sobre o espaço transformado com a distância de alcance simétrica.

Assim, considerando a entrada de uma partição, calcula-se $a_{ptscoredist}$ sendo a inversa da densidade do objeto x_i em seu *cluster* a partir da equação (2.12).

$$a_{ptscoredist}(x_i) = \left(\frac{\sum_{j=2}^{n_i} \left(\frac{1}{d(x_i, x_j)} \right)^M}{n_i - 1} \right)^{-\frac{1}{M}} \quad (2.12)$$

Onde:

- M é a dimensionalidade dos dados;
- n_i é o tamanho do i -ésimo *cluster*;
- $d(x_i, x_j)$ é a distância entre os objetos x_i e o seu j -ésimo vizinho mais próximo no respectivo *cluster*.

Uma vez que todos os objetos têm seu $a_{ptscoredist}$ calculado, se constrói uma matriz de acessibilidade mútua dos objetos do *cluster* usando a definição de distância de acessibilidade mútua (equação (2.13)).

$$d_{mreach}(x_i, x_j) = \max(a_{ptscoredist}(x_i), a_{ptscoredist}(x_j), d(x_i, x_j)) \quad (2.13)$$

A partir da matriz d_{mreach} de cada *cluster*, um MST_{MRD} é construído para capturar a estrutura subjacente dos dados. Considerando um MST_{MRD} como a Árvore Geradora Mínima construída usando $a_{ptscoredist}$ considerando os objetos do *cluster* considerado. Assim, a partir do MST_{MRD} é possível calcular a Escassez de Densidade do Cluster (DSC), que é definida como o peso máximo de borda entre todas as bordas internas em MST_{MRD} do *cluster*. Também se pode definir a Separação de Densidade de um Par de *Clusters* (DSPC) como a distância de alcance mínimo entre os nós internos dos MST_{MRD} entre dois *clusters*. Utilizando essas definições se pode calcular a validade de um *cluster*, considerando C_i e C_j dois *clusters* distintos, através da equação (2.14).

$$VC(C_i) = \frac{\min_{1 \leq j \leq l, j \neq i}(DSPC(C_i, C_j) - DSC(C_i))}{\max(\min_{1 \leq j \leq l, j \neq i}(DSPC(C_i, C_j)), DSC(C_i))} \quad (2.14)$$

Por fim é definido o índice de validade da solução para todo o agrupamento C como sendo a média ponderada do índice de validade de todos os *clusters* em C (equação (2.15)).

$$DBCVC(C) = \sum_{i=1}^l \frac{|C_i|}{|N|} VC(C_i) \quad (2.15)$$

Onde:

- l é o número de *clusters*;
- N é o número total de objetos em avaliação, incluindo os ruídos.

O índice $DBCVC$ gera valores entre -1 e +1, sendo que valores maiores indicam melhores soluções de *clustering* baseadas em densidade.

2.4.6 F1-Score

O *F1-Score* deriva da média harmônica da **precisão** e **revocação**. O objetivo desse critério externo de validação é avaliar a eficácia dos algoritmos de agrupamento de dados quando aplicados nos conjuntos. Desta forma, é necessário conceituar primeiramente precisão e revocação.

Precisão representa a proporção de itens classificados como corretos, neste caso rótulos que o algoritmo de agrupamento de dados retorna, que realmente estão corretos, considerando os resultados adequados para os conjuntos de dados, encontrados na literatura, descartando qualquer coisa que possa não estar correta. É definido conforme a equação (2.16):

$$P(L_r, G_i) = \frac{|L_r \cap G_i|}{|G_i|} \quad (2.16)$$

Onde:

- L_r é a informação externa sobre o conjunto de dados considerado, ou seja, a resposta encontrada na literatura;
- G_i é a predição feita pelo algoritmo de *clustering* para L_r .

Revocação indica quanto de todos os itens que deveriam ter sido classificados como corretos, foram realmente classificados como corretos. Nenhum item correto é deixado de fora. A revocação é definido pela equação (2.17) e utiliza as mesmas variáveis da precisão.

$$R(L_r, G_i) = \frac{|L_r \cap G_i|}{|L_i|} \quad (2.17)$$

Logo, define-se *F1-Score* como apresentada na equação (2.18).

$$F(L_r, G_i) = \frac{2 * P(L_r, G_i) * R(L_r, G_i)}{P(L_r, G_i) + R(L_r, G_i)} \in [0, 1] \quad (2.18)$$

Conforme o algoritmo consegue reconstruir a informação dos grupos predeterminadas para um conjunto, o valor de *F1-Score* se aproxima de 1. Caso contrário, a *F1-Score* é próximo do valor 0. A Tabela 1 sumariza os critérios de validação de *clusters*, gerados a partir de algoritmos de agrupamentos de dados, apresentados anteriormente.

2.5 Sistemas Multiagente

Antes de conceituar um Sistemas Multiagente, é importante apresentar, primeiramente, o que é um agente dentro dessa perspectiva. Wooldridge; Jennings (1995) apresentam dois usos gerais do termo agente, o primeiro considerado fraco e um pouco incontestável e o segundo mais forte, e talvez mais ambíguo.

A noção fraca de agente é usada para denotar um hardware ou, mais usual, um software baseado em sistema de computador que possui as seguintes propriedades:

- **Autonomia:** capacidade dos agentes em operar sem a intervenção direta de seres humanos ou de outros, possuindo assim algum tipo de controle sobre suas ações e estado interno.
- **Habilidade social:** capacidade dos agentes em interagir com outros agentes e talvez com seres humanos também por intermédio de algum tipo de linguagem de comunicação de agentes.
- **Reatividade:** os agentes são capazes de perceber seu ambiente e responder de forma adequada às mudanças que ocorrem.

Tabela 1: Alguns Critérios de Validação de Clusters
Critérios de Validação de Clusters

Nome do Índice	Descrição
<i>Silhouette</i>	Esse índice determina a qualidade dos agrupamentos baseado na proximidade entre os objetos de um determinado grupo, além da proximidade desses objetos ao grupo que está mais próximo. Resulta em valores que variam entre $[-1, 1]$. Quanto mais próximo de 1 melhor o objeto está alocado no grupo.
<i>Davies-Bouldin</i>	O cálculo do índice está em função da razão entre a soma da dispersão interna dos agrupamentos e a distância entre dos referidos agrupamentos. Logo, como resultados desejáveis se tem que menores valores desse índice correspondem a agrupamentos compactos e com os centroides distantes entre si.
<i>Dunn</i>	O índice é calculado pela razão entre a menor distância intergrupo e a maior distância intragrupo. O resultado varia no intervalo $[0, 1)$, de tal forma que quanto maior o valor resultante mais compactos e bem separados são os grupos.
<i>Dunn Generalizado</i>	Introduzido por Pal; Biswas (1997) reescreve o Índice Dunn em função de três estruturas geométricas de grafos (GG, RNG e MST).
<i>DBC</i>	É um índice de validação relativo para agrupamentos baseados em densidade, arbitrariamente formados, ou seja, avalia a qualidade de agrupamento com base na conexão da densidade relativa entre pares de objetos.
<i>F1 - Score</i>	É um critério externo de validação que avalia a eficácia dos algoritmos de agrupamento de dados quando aplicados em conjuntos. O resultado varia no intervalo $[0, 1]$, de tal forma que quanto maior o valor resultante mais correto é considerado o agrupamento predito pelo algoritmo.

- **Pró-atividade:** capacidade dos agentes de exibir o comportamento orientado por algum objetivo e tomada de iniciativa.

Por sua vez, a noção mais forte compreende o agente como um sistema de computador que, além de possuir as propriedades citadas acima, na noção fraca, também abrange conceitos que são mais comuns em seres humanos como, por exemplo, as noções mentalísticas: conhecimento, crença, intenção e obrigação.

Além das propriedades citadas, os autores apontam também outros atributos: a capacidade de mobilidade do agente, a veracidade de suas informações utilizada na comunicação, a benevolência com que lhe é pedido e a racionalidade como suposição de que um agente agirá para atingir seus objetivos.

Em Russell; Norvig (1995) encontra-se a seguinte definição de agente como qualquer entidade que pode perceber seu ambiente através de sensores e agindo sobre esse ambiente por meio de efetores. A Figura 2.9 mostra uma representação genérica de agente produzida pelos autores.

Nos últimos anos, a tendência em computação é de sistemas de computador cada vez mais globais e interligados. O desenvolvimento de paradigmas de software que são capazes de explorar o potencial de tais sistemas é talvez o maior desafio na computação neste século e a noção de agentes parece ser uma forte candidata a esse paradigma (Wooldridge, 2002).

Com base nos conceitos apresentados, define-se Sistemas Multiagente como sistemas com-

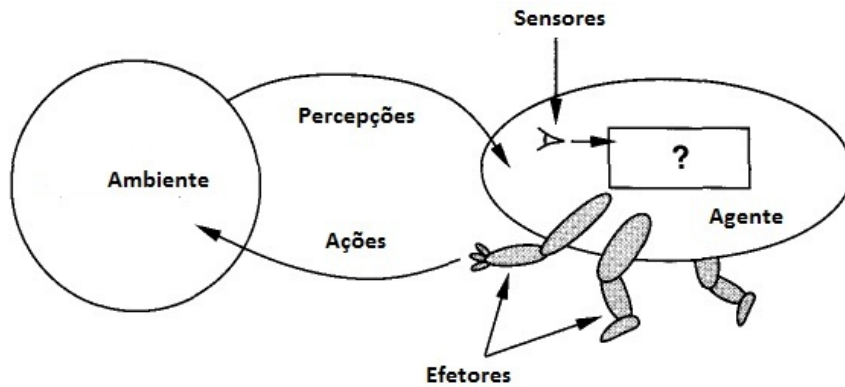


Figura 2.9: Representação genérica de agente.
Russell; Norvig (1995)

postos por elementos computacionais chamados agentes. Logo, consiste em agentes que se interagem uns com os outros tipicamente através de trocas de mensagens. Para interagir com sucesso, esses agentes terão as capacidades de cooperar, coordenar e negociar uns com os outros, da mesma forma as pessoas realizam no cotidiano (Wooldridge, 2002).

Stone; Veloso (2000) citam as seguintes razões para o uso de Sistemas Multiagente: Exigência de alguns domínios; Paralelismo; Robustez; Escalabilidade; Programação mais simples; Estudo da inteligência; Distribuição geográfica; e custo-benefício. Além disso, o campo dos sistemas multiagente é altamente interdisciplinar, com aplicações em áreas diversas como: economia, filosofia, lógica, ecologia e ciências sociais. Possuindo assim, muitas visões diferentes como: Agentes como um paradigma para engenharia de software e Agentes como uma ferramenta para compreender as sociedades humanas (Wooldridge, 2002).

A informação coletiva que atinge os sensores dos agentes em um Sistema Multiagente é tipicamente distribuída: os agentes podem observar dados que diferem espacialmente (aparecem em locais distintos), temporalmente (chegam em tempos diferentes), ou mesmo semanticamente (requerem interpretações diferentes). Isso, automaticamente torna o estado de mundo parcialmente observável para cada agente, apresentando consequências nas suas tomadas de decisões. Também é possível a fusão de sensores, ou seja, os agentes podem combinar suas percepções de forma otimizada para aumentar seu conhecimento coletivo a cerca do estado atual (Stone; Veloso, 2000). Assim, em Sistemas Multiagente, múltiplos agentes existem de maneira simultânea em um mesmo ambiente, onde eles se cooperam mutuamente para atingir uma determinada meta.

Baseando-se no que foi descrito até aqui, um dos objetivos deste trabalho é a utilização uma espécie de sistema de agentes para resolver problemas de agrupamento de dados, ou seja, a formulação do algoritmo de *clustering* se apropria de algumas características desses elementos computacionais.

2.6 Trabalhos Relacionados

Para colaborar com o desenvolvimento e fundamentação da pesquisa, realizou-se uma busca, em plataformas digitais, por trabalhos publicados que envolvessem concomitantemente os respectivos tópicos: Algoritmos de agrupamento de dados, Sistemas Multiagente e comportamento coletivo.

Entre os trabalhos encontrados destacam-se aqueles que se apropriam de comportamentos coletivos típicos de populações, conhecidos também como algoritmos bioinspirados. Algoritmos inspirados em colônia de formigas, colônia de abelhas, otimização por enxame de partículas - PSO, bando de pássaros, estão entre os mais comuns na literatura e possuindo diversas adaptações.

Deneubourg et al. (1991) apresenta um modelo inspirado no agrupamento de corpos mortos da espécie de formiga *Pheidole pallidula*. Nesse modelo, as formigas se movem de forma randômica, e baseia-se em duas probabilidades, uma relacionada a formiga pegar um objeto e outra para ela deixá-lo, sendo que a de pegar é maior quando o respectivo objeto está isolado e a de deixar é mais elevada quando o objeto está em um local com outros objetos em suas proximidades. Alicerçadas nesse trabalho, outras pesquisas em computação, com o intuito de agrupamento de objetos foram desenvolvidas. Lumer; Faieta (1994), por exemplo, descrevem o Algoritmo Simples de *Clustering* por Formiga (*Standart Ant Clustering Algorithm - SACA*) optando apenas pela utilização de uma função de similaridade contínua para avaliar a similaridade entre diferentes objetos. Outras alterações no SACA foram propostas, como em Monmarché (1999), Handl (2003), Hartmann (2005), Vizine et al. (2005), Aranha; Iba (2006).

Colônia de abelhas também inspiraram a implementação de algoritmos de *clustering*, os trabalhos de Karaboga; Ozturk (2009) e de Santos (2009) são exemplos. O primeiro chamado Algoritmo *Artificial Bee Colony - ABC*, basea-se nos comportamentos das abelhas para a coleta de alimentos e serve como base de outras pesquisas realizadas (Zhang; Ouyang; Ning (2010), Andrade; Cunha (2011) e Ranjbar; Azami; Rostammi (2015)). O segundo, por sua vez, nomeado *Bee Clustering*, a principal inspiração para a formação de novos grupos é o movimento realizado durante a “dança” das abelhas para recrutamento de novos agentes.

A Otimização por Enxame de Partículas - PSO tem como um dos “pontapés” iniciais o trabalho de Kennedy; Eberhart (1995), que descrevem comportamentos sociais entre indivíduos, como em um bando de pássaros e em cardumes de peixes. A pesquisa de Reynolds (1987) foi uma referência para os autores durante a simulação computacional do movimento de indivíduos pertencentes a um bando de pássaros. Destacam-se também os trabalhos de Merwe; Engelbrecht (2003) e Cohen; de Castro (2003), como exemplos do uso de PSO em *clustering*.

Em um contexto diferente de algoritmos bioinspirados, a utilização de Multiagente na tarefa de agrupamento de dados também são descritos nos trabalhos de Agogino; Tumer (2010) e Chaimontree; Atkinson; Coenen (2011). Agogino; Tumer (2010) apresentam um método baseado em agente com o objetivo de combinar muitas bases de agrupamentos em um único agrupamento unificado de “consenso” que é robusto contra vários tipos de falhas e não requer

sincronização espacial ou temporal. Nesta abordagem, os agentes processam agrupamentos provenientes de fontes separadas e agrupam-nas para produzir um consenso unificado. Os resultados dessa pesquisa mostram que o método alcança um desempenho melhor ou equivalente aos métodos tradicionais de agrupamento de consenso não-agente em condições sem falhas e permanece eficiente em uma ampla gama de cenários de falha que prejudicam o desempenho de métodos tradicionais.

Chaimontree; Atkinson; Coenen (2011) descrevem um *framework* para agrupamento com vários agentes é descrito, onde cada agente individual representa um *cluster* individual. A partir da criação de uma configuração inicial do *cluster*, os agentes podem negociar com o intuito de melhorar esse agrupamento inicial. O *framework* é usado nos seguintes paradigmas de agrupamento: *K-means* e KNN. A avaliação relatada, pelos autores, demonstra que a negociação pode servir para melhorar uma configuração de *cluster* inicial.

Em relação a comportamento coletivo e *clustering*, enfatiza-se o trabalho de dissertação de Gueleri (2013). Na pesquisa, o autor desenvolveu técnicas de agrupamento baseadas em comportamento coletivo e auto-organização. Nessas técnicas cada objeto do conjunto de dados corresponde a um indivíduo do sistema, os quais interagem um com os outros, de forma que os grupos apareçam a partir de uma organização promovida por próprios. Os objetos são mantidos fixos em seu espaço de atributos, mas carregam certo tipo de “energia”. Tal energia será trocada gradualmente entre eles. Logo os grupos serão formados por objetos que possuem energias semelhantes.

Também foram encontrados trabalhos que utilizam algoritmos de *clustering* para a otimização da distribuição de agentes e o emprego dessas duas estratégias em algumas aplicações, como em imagens digitais (Kubalík et al. (2010), Minden; Youn; Khan (2012), Xin; Sagan (2016)). Por fim, a Tabela 2 sumariza os algoritmos de agrupamentos de dados que foram mencionados anteriormente.

Tabela 2: Tabela resumo de trabalhos relacionados
Principais Trabalhos Relacionados

Inspiração	Principais trabalhos	Descrição
<i>Colônia de formigas</i>	Deneubourg et al. (1991), Lumer; Faieta (1994), Monmarché (1999), Handl (2003), Hartmann (2005), Vizine et al. (2005) e Aranha; Iba (2006).	Esses trabalhos apresentam modelos matemáticos que descrevem o comportamento em colônia de formigas, principalmente aquele ocasionado pela formação de cemitérios. Baseados nesses modelos vários algoritmos de agrupamento de dados foram desenvolvidos, o principal deles é o SACA, que motivou outros trabalhos nessa mesma perspectiva.
<i>Colônia de abelhas</i>	Karaboga; Ozturk (2009), Santos (2009), Zhang; Ouyang; Ning (2010), Andrade; Cunha (2011) e Ranjbar; Azami; Rostammi (2015)	Colônias de abelhas também serviram de inspiração para novos algoritmos de agrupamentos de dados. Os principais comportamentos modelados são os realizados para a coleta de alimentos e a dança para o recrutamento de abelhas.
<i>Otimização por enxame de partículas - PSO</i>	Kennedy; Eberhart (1995), Reynolds (1987), Merwe; Engelbrecht (2003) e Cohen; de Castro (2003)	Os algoritmos de <i>clustering</i> inspirados em PSO, utilizam os modelos que descrevem os comportamentos em bandos de passáros.
<i>Multiagente</i>	Agogino; Tumer (2010), Chaimontree; Atkinson; Coenen (2011), Kubalík et al. (2010), Minden; Youn; Khan (2012) e Xin; Sagan (2016)	São nesses trabalhos que são efetivamente utilizado o conceito de multiagente em <i>clustering</i> ou o uso de um algoritmo para algumas aplicações que baseiam-se em agentes.
<i>Troca de energia e coletividade</i>	Gueleri (2013)	Nesse trabalho, são propostos três algoritmos de agrupamento de dados baseando-se na coletividade e auto-organização, através de troca de energia entre os objetos considerados.

Após essas considerações, a proposta apresentada nesse trabalho relata o desenvolvimento e avaliação de um novo algoritmo de agrupamento de dados, baseando-se em algumas propriedades de agentes como a autonomia, a capacidade de reação e troca de informações que são características já testada por alguns dos trabalhos relatados anteriormente. Porém, ao contrário de algumas propostas, a nova técnica não se baseia em algum comportamento bioinspirado, as características para cada agente, que representa um objeto de dado, são formuladas de maneira artificial e os grupos serão formados a partir da capacidade visual de cada agente e de suas movimentações durante o processo de agrupamento.

3 ALGORITMO DESENVOLVIDO

Este trabalho apresentou inicialmente um estudo amplo do problema de agrupamento de dados e alguns índices de validação, passando também por outros temas relevantes para a nova proposta, como a definição de agentes e pesquisas que possuem objetivos semelhantes.

Durante essa revisão teórica percebe-se algumas desvantagens de determinados algoritmos de *clustering*, tais como a definição *a priori* de quantos grupos a serem formados ou a inserção de centroides e estruturas que forcem a estabilidade do agrupamento. Tendo em vista essas deficiências, busca-se o desenvolvimento de um algoritmo apto para auto-organizar seus objetos de dados com o intuito de formarem grupos bem definidos. Para este fim, são utilizadas algumas propriedades importantes do conceito de agentes, como sua capacidade de executar ações autônomas e de comunicação com outros agentes pertencentes ao sistema. Resumidamente, o algoritmo segue a ideia exposta a seguir:

Dado um conjunto de dados X com n objetos, ou seja $X = \{x_1, x_2, x_3, \dots, x_n\}$, de forma que cada objeto x_i , para $i = \{1, 2, 3, \dots, n\}$ está associado a um agente e um alcance de visão único α para todos os agentes. O algoritmo é iniciado com cada agente apresentando um rótulo distinto, por exemplo, x_1 possui um rótulo c_1 , x_2 possui um rótulo c_2 e assim por diante, de modo que todos os c_i são diferentes. Assim é correto afirmar que, inicialmente cada agente x_i é o seu próprio grupo, ou seja, o conjunto X possui n grupos na primeira iteração. São esses rótulos que poderão ser propagados durante o processo de agrupamento. Além disso, cada x_i é o centro de uma circunferência de raio igual ao alcance α . Essa circunferência é dividida em 8 setores circulares que representam as possíveis opções de direção de movimentação para os agentes. Essa movimentação, de velocidade constante, se faz necessária para a exploração local. O objetivo de cada agente é encontrar o setor que possui mais agentes. São com os agentes desse setor que serão realizadas as trocas de informações, que consiste na alteração do estado dos rótulos dos agentes envolvidos.

A Figura 3.1 ilustra esse comportamento que representa o processo de agrupamento do algoritmo. Cada “círculo colorido” representa um agente diferente, sendo que cada cor distinta dos objetos indica um grupo. Na ilustração, o agente x_1 centrado em uma circunferência de raio qualquer optando por se mover em direção do Setor 1 (ângulo de 0°) que é o de maior número de outros agentes (5 agentes no total). Um pseudocódigo da implementação realizada é descrito na próxima seção, denomina-se o algoritmo de *DL3*.

3.1 O Algoritmo *DL3*

Considerando a Figura 3.2, as principais etapas do processo de agrupamento do *DL3* são:

- **Etapa Inicial:** Primeiramente é necessário a entrada de um conjunto de dados, sendo que cada objeto de dado será definido como um agente do sistema e um número que determina o alcance da visão dos agentes, de modo que este valor seja o mais adequado

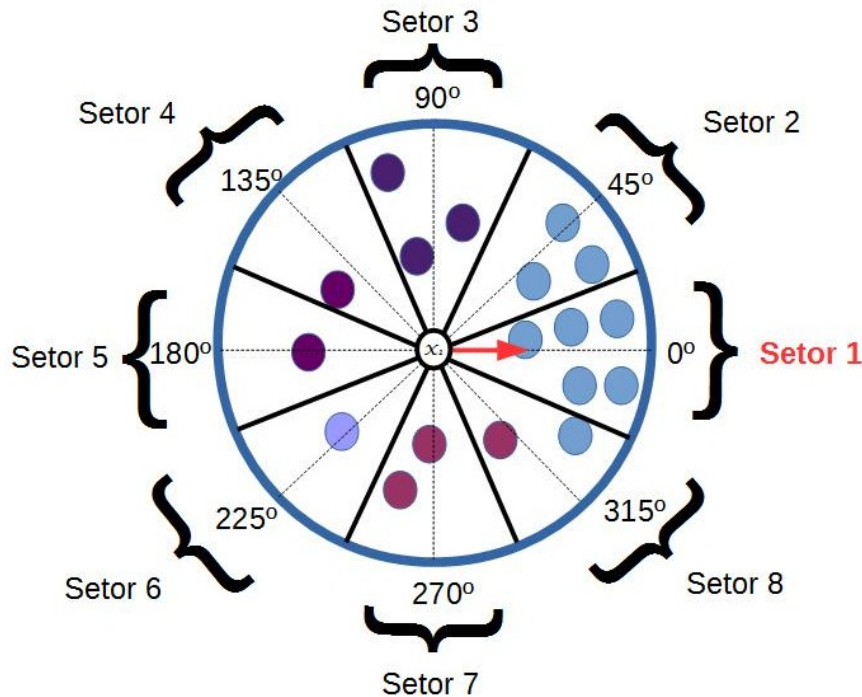


Figura 3.1: Representação de como cada objeto analisa os setores de visão.

para cada conjunto de dados. Inicialmente, cada ponto dado possuirá um rótulo distinto e característico.

- Etapa de Agrupamento:** Cada agente possui um campo de direção de 360° discretizado em 8 setores, cada um cobrindo uma faixa de 45° e raio igual ao alcance definido *a priori* pelo usuário. Em cada iteração, a direção para onde os agentes se movem é definida usando o critério baseado no número de outros agentes percebidos por cada agente x_i , sendo que eles se movem em uma velocidade constante. Se houver uma melhor direção percebida para se mover, x_i se move para onde o número de vizinhos é maior. Caso contrário, se não houver agentes em qualquer setor ou em caso de empates uma direção aleatória é definida, ou seja, quando não houver agentes em qualquer setor, o agente escolhe aleatoriamente uma das 8 direções para se mover, e se for encontrado mais de um setor com mais agentes, então aleatoriamente os agentes escolhem um desses setores para se mover. O rótulo do *cluster* de cada agente, que foram considerados diferentes inicialmente, será alterado quando após x_i mover-se para a direção com mais agentes. Assim, x_i herda o rótulo do x_a mais próximo pertencente ao *cluster* que foi direcionado. Nesse caso, ambos os agentes estão configurados para o mesmo cluster, pois o rótulo de x_i está configurado para ser o mesmo rótulo de x_a .
- Critério de Parada:** O critério de parada do algoritmo é estabilidade dos agrupamentos, ou seja, verificar quando o número de grupos não se modifica significativamente em relação ao alcance de visão adotado para os agentes.

Com essa nova formulação de algoritmo de agrupamento espera-se que o número e a dis-

tribuição de grupos surjam como resultado da interação entre agentes. Nenhuma informação centróide é necessária ou mantida, na verdade, somente os agentes são responsáveis pelo seu agrupamento. Assim, os *clusters* resultantes são dependentes da distribuição dos pontos de dados e do alcance de visão assumido.

Entrada: Um conjunto de dados X e um alcance de visão α para os agentes.

Saída: Uma partição de X em k grupos.

Início

Cada $x_i \in X$, para $i \in \mathbb{N}^*$, representa um agente diferente no plano.

para cada $x_i \in X$ **faça**

 | $c_i = i$
 | // cada agente está no seu próprio grupo e possui um rótulo individual.

fim

repita

 | **para** cada $x_i \in X$ **faça**
 | | Cada x_i é o centro de uma circunferência de raio igual ao alcance α ;
 | | A circunferência é dividida em 8 setores circulares;
 | | Encontrar o setor que possui mais agentes.
 | | **se** *Houver mais de um setor com o mesmo número de agentes* **então**
 | | | Mover o agente aleatoriamente na direção de um dos setores que possuem o
 | | | mesmo número de agentes.

 | | **fim**

 | | **senão**

 | | | Mover o agente na direção do setor com mais vizinhos.

 | | **fim**

 | **fim**

 | **para** cada $x_i \in X$ **faça**

 | | $c_i = c_a$ // x_i herda o rótulo do x_a mais próximo pertencente ao setor que é
 | | direcionado.

 | **fim**

até *Não haver mais alteração na associação de rótulos para os agentes;*

Fim

Figura 3.2: *DL3*

3.2 Metodologia Adotada

A Figura 3.3 mostra as quatro etapas realizadas para atingir os objetivos definidos previamente. A primeira etapa compreende a implementação do algoritmo em uma linguagem para simulações multiagente. Na etapa seguinte, foram selecionados seis conjuntos de dados para o teste do algoritmo. Com o término dos testes, caracterizou-se a terceira etapa com a validação desses resultados com o auxílio de oito índices. E por fim, a análise final definida como a comparação dos resultados gerados pelo *DL3* com outros dois algoritmos de *clustering* tradicionais. A seguir, são melhor explicadas essas etapas.

Para realizar uma análise experimental, programou-se uma versão do algoritmo utilizando o software NetLogo. Esse ambiente de programação é adequado para a modelagem de sistemas

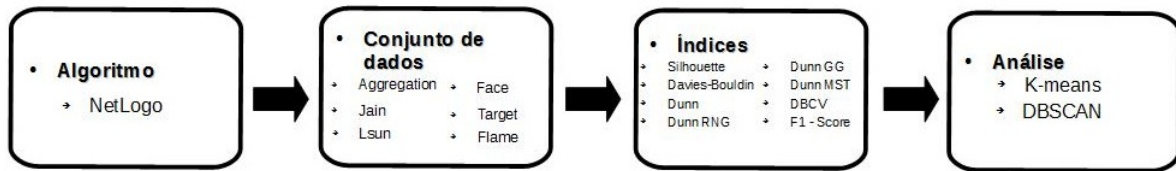


Figura 3.3: Etapas do trabalho

complexos e possuem um desenvolvimento ao longo do tempo. O NetLogo foi criado em 1999 por Uri Wilensky e está em desenvolvimento contínuo desde então no Center for Connected Learning and Computer-Based Modeling na Universidade Northwestern, localizada em Illinois nos Estados Unidos. É um software livre e de código aberto escrito em Scala e em Java e roda na máquina virtual do Java. Ele permite passar instruções a milhares de agente com comportamentos independentes. Tornando-se dessa maneira uma ferramenta para a análise comportamental em iterações sucessivas.

A figura 3.4 mostra a interface criada no NetLogo e que foi usada para as simulações. Os itens numerados, a seguir, explicam cada elemento mostrado na interface como, as variáveis de entrada, e os botões responsáveis pelas ações necessárias durante a execução das etapas do algoritmo de agrupamento implementado.

1. **Input nome-arquivo:** Lê um arquivo do formato txt.
2. **Input num-pontos:** Responsável pela entrada no número de objetos do conjunto de dados, que serão mostrados em 10 da Figura 3.3.
3. **Input visao-fixa:** Determina o número relacionado ao alcance de visão de cada agente.
4. **Button setup:** Plota os objetos de forma aleatória, ou seja, não é necessário entrar com a coordenadas de cada agente.
5. **Button abre:** Abre e plota os objetos de dados de um determinado conjunto com as coordenadas definidas em um arquivo de formato txt.
6. **Button go:** Realiza a etapa de agrupamento dos objetos, segundo o que está descrito em Algoritmo 5.
7. **Button voltar-pontos:** Retorna os objetos para suas posições iniciais. porém, com suas novas rotulações, para os conjuntos formados aleatoriamente pelo programa.
8. **Button voltar-abre:** Retorna os objetos para suas posições iniciais. porém, com suas novas rotulações, para os conjuntos carregados a partir de um arquivo txt.
9. **Monitor Número de Grupos:** Mostra o número final de grupos formados pelo agrupamento dos agentes.

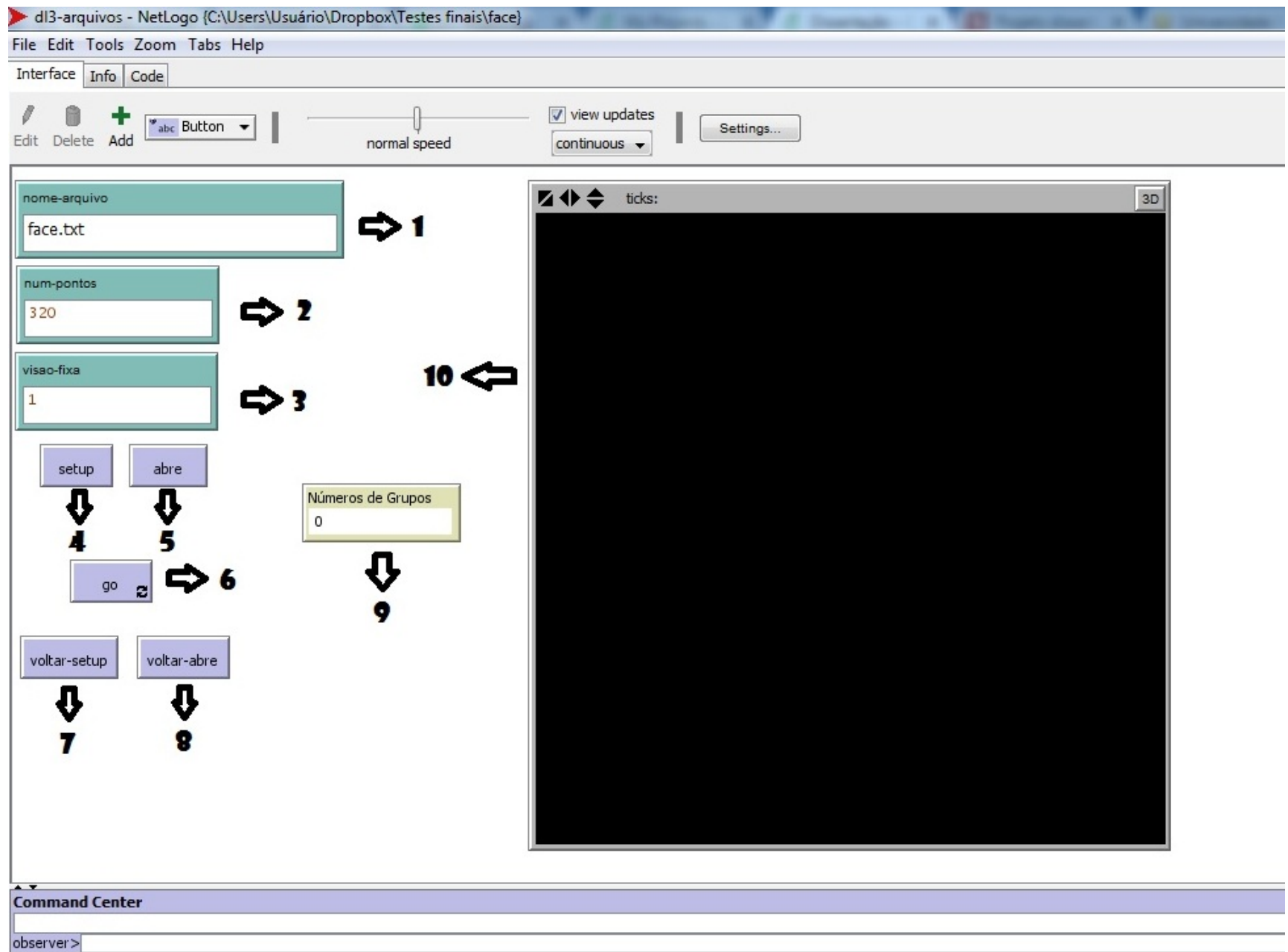


Figura 3.4: Interface do NetLogo

Esse algoritmo foi aplicado nos seguintes conjuntos de dados de duas dimensões ¹: *Aggregation* (Gionis; Mannila; Tsaparas (2007)), *Jain* (Jain; Law (2006)), *Lsun* (Utsch (2005)), *Target* (Utsch (2005)), *Face* (Ilc (2013)), *Flame* (Fu; Medico (2007)). A Figura 3.5, mostra esses conjuntos de dados utilizados e a Figura 3.6 tem-se um exemplo que demonstra o comportamento do *DL3* quando aplicado em um desses conjuntos. Na Figura 3.6 fica claro as movimentações realizadas por cada agente nas sucessivas iterações, necessária para as trocas de rótulos entre os agentes, até a parada do algoritmo quando os agentes retornam as suas posições iniciais, agora com a identificação corresponde ao *cluster* que se agrupou.

Na próxima seção apresenta-se os resultados dessas aplicações, porém alguns resultados preliminares do desempenho do algoritmo podem ser vistos em Godois et. al (2017a), considerando apenas conjuntos de dados gerados de forma aleatória, e Godois et. al (2017b) que utiliza alguns dos conjuntos de dados usados nessa dissertação, mas validando os resultados com o uso de apenas um índice, o *Silhouette*. Logo, antes dos resultados desse trabalho, é importante expor os principais problemas encontrados nesses conjuntos de dados selecionados.

Aggregation é um conjunto de dados sintético de pontos bidimensionais. Um agrupamento intuitivamente bom para este conjunto de dados consiste nos sete grupos perceptualmente distintos, conforme apresentado na Figura 3.5. De fato, o conjunto de dados contém recursos que são conhecidos por criar dificuldades para os algoritmos de *clustering*, como ligações estreitas entre *clusters*, grupos de tamanho desigual, etc.

O conjunto de dados *Jain*, por sua vez, possui o formato de duas meia luas. Visualmente é perceptível que o conjunto possui dois grupos bem definidos e bem separados no plano bidimensional. A curvatura apresentada por esse conjunto se caracteriza como a principal dificuldade para o agrupamento.

O *Lsun*, é um conjunto que por inferência visual apresenta três grupos claramente separados no plano bidimensional, dois conjuntos são da forma de retângulos alongados que formam a letra L e um conjunto de forma circular formando algo que se aproxima de um sol. O seu principal problema são as diferentes variações entre os *clusters*.

No conjunto *Target*, visualmente se tem seis grupos no plano *xy*. Os *clusters* apresentam os seguintes formatos: um com a forma de *bullseye* (centro de um alvo de tiro), um que contém o primeiro e também circular e quatro pequenos aglomerados colocados como *outliers*. A dificuldade dos algoritmos de *clustering* em detectar os formatos desse conjunto de dados fica por conta da presença dessas grupos definidos como *outliers*.

No conjunto de dados denominado *Face*, tem como resposta quatro grupos. Os formatos dos *clusters* são os seguintes: uma meia lua, dois *clusters* circulares e um com forma não bem definida. Os quatro *clusters* produzem uma espécie de rosto no plano de duas dimensões. Logo, devido a existência de três formas distintas, esse conjunto exige que um algoritmo consiga identificá-las corretamente.

¹Os conjuntos de dados se encontram nos seguintes links: [https : //cs.joensuu.fi/sipu/datasets/](https://cs.joensuu.fi/sipu/datasets/), [https : //www.uni-marburg.de/fb12/arbeitsgruppen/datenbionik/data?language_sync = 1](https://www.uni-marburg.de/fb12/arbeitsgruppen/datenbionik/data?language_sync=1) e [https : //www.mathworks.com/matlabcentral/fileexchange/42199-generalized-dunn-s-index](https://www.mathworks.com/matlabcentral/fileexchange/42199-generalized-dunn-s-index)

Por fim, tem-se o conjunto de dados *Flame*, possuindo 240 objetos de dados bidimensionais, que pertencem a dois clusters. Os clusters estão alinhados um perto do outro, e apresentam características geométricas interessantes, composta por uma distribuição não - esférica de elementos e outra concentração no formato de meia lua.

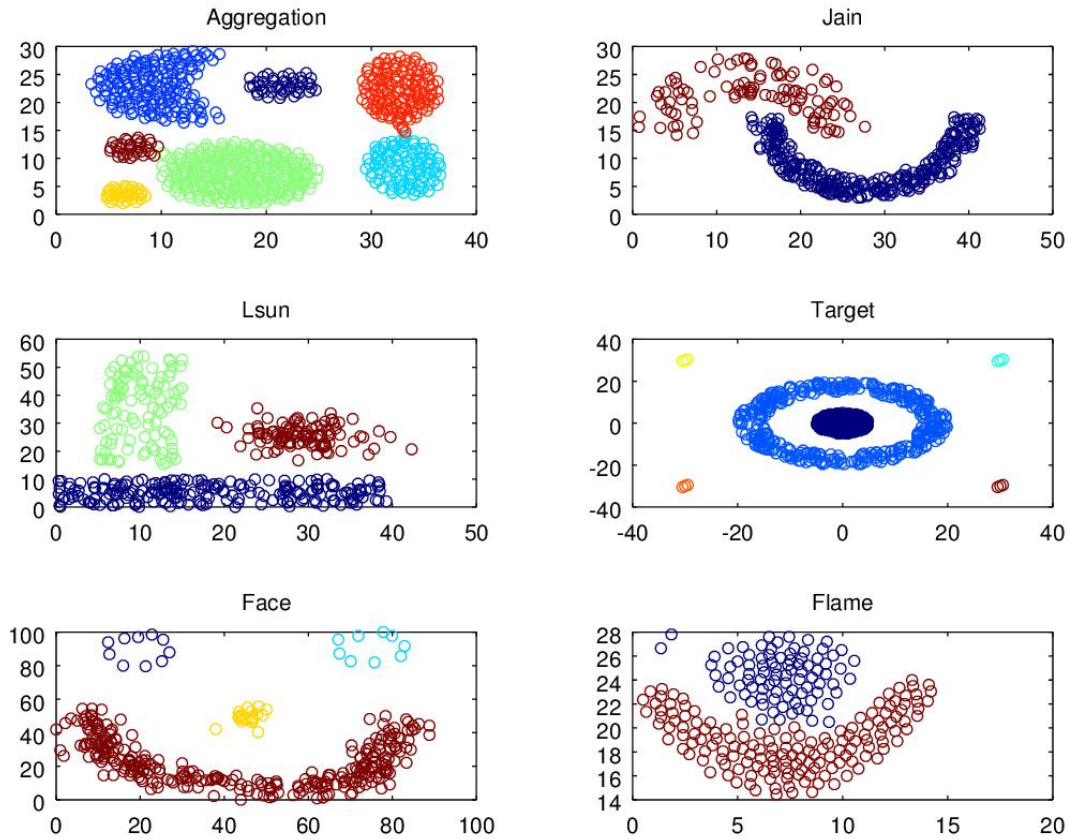


Figura 3.5: Conjuntos de dados utilizados no trabalho.

Como o algoritmo proposto define grupos que não são conhecidos *a priori*, a partição final de dados irá requerer algum tipo de avaliação. Assim, após as respectivas simulações e testes, se realizou a etapa de validação do agrupamento, a qual se refere aos procedimentos que avaliam os resultados da análise de grupo de forma quantitativa e objetiva partindo da premissa que os problemas de validade de grupo são inerentemente estatísticos (Jain; Dubes, 1988). Para essa validação dos resultados serão usados alguns critérios apropriados, como os apresentados no Capítulo 2. Por fim, os resultados finais obtidos para o algoritmo foram comparados com as técnicas, *K-means* e *DBSCAN*, destacando as vantagens e desvantagens dessa nova proposta.

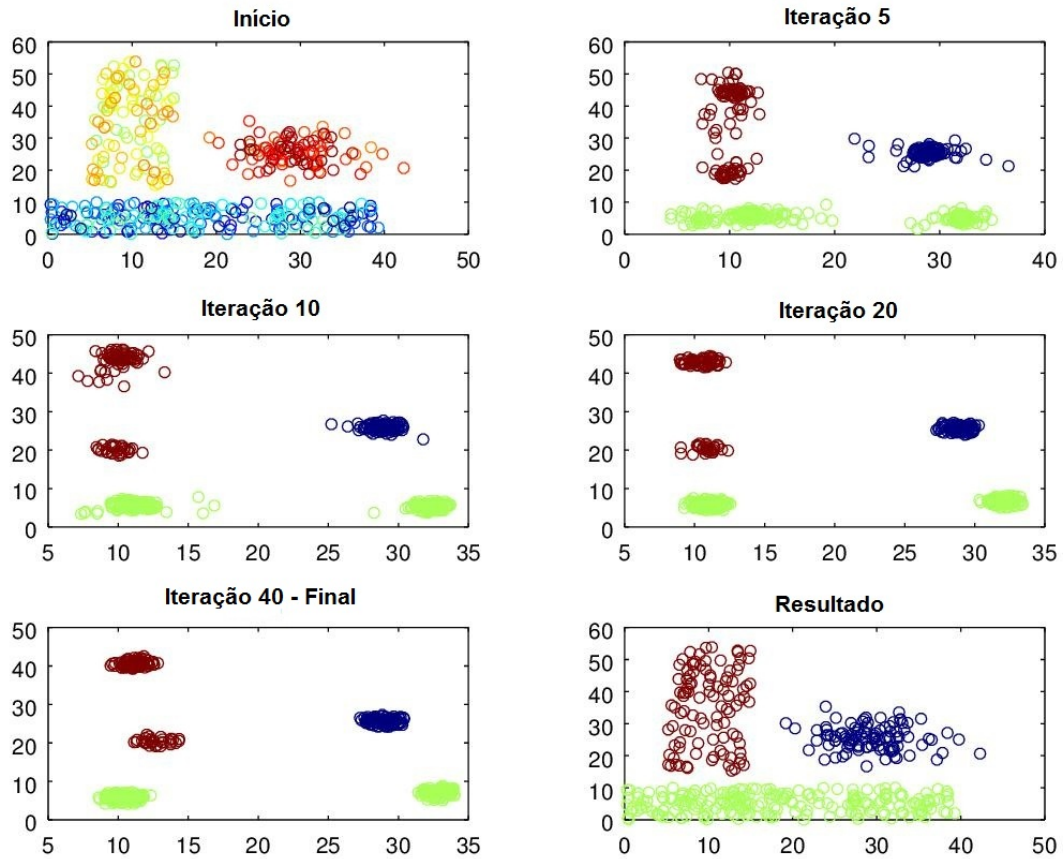


Figura 3.6: Exemplo do comportamento do $DL3$ em sua aplicação no conjunto de dados $Lsun$.

4 ANÁLISE DOS RESULTADOS

Após a implementação do Algoritmo $DL3$ na plataforma NetLogo, foram realizadas simulações considerando os conjuntos de dados em duas dimensões de diferentes tamanhos e formatos, apresentados na Figura 3.5, considerando também ajustes no parâmetro de entrada α . Para todos os conjuntos de dados utilizou-se $0,1 \leq \alpha \leq max(\alpha)$ para determinar o alcance de visão dos agentes. De modo que, a cada nova simulação ao valor de α é acrescido uma quantidade de 0,1 e $max(\alpha)$ representa o alcance de visão que gera um único grupo como resultado final do agrupamento.

Concluída essa fase, optou-se por encontrar os melhores agrupamentos gerados, a partir da aplicação da nova proposta de algoritmo nos conjuntos de dados, através dos cálculos de índices de validação: *Silhouette*, *Davies-Bouldin*, *Dunn*, *Dunn RNG*, *Dunn GG*, *Dunn MST* e *DBCW*. Esses índices foram escolhidos pois, possuem fundamentações matemáticas distintas e podem gerar diferentes interpretações de qual agrupamento é o mais adequado, de forma que alguns métodos são mais sensíveis em detectar alguns formatos de *clusters*.

Com os resultados dos cálculos dos índices concluídos e descobertos os melhores agrupa-

mentos produzidos pelo *DL3*, realizou-se procedimento semelhante para outros algoritmos de *clustering*, ou seja, descobrir o melhor agrupamento através dos valores dos índices para o *K-means* e *DBSCAN* e assim, realizar as comparações entre eles. Como o *DBSCAN* é um algoritmo baseado em densidade, para a comparação, optou-se apenas em utilizar o índice *DBC*V que possui uma perspectiva de densidade também.

Ainda para auxiliar a análise entre as técnicas, foram calculadas as porcentagens de quanto os resultados do *DL3* melhoraram ou pioraram em relação aos valores dos índices dos outros algoritmos. Além disso, como a literatura disponibiliza um resultado considerado “ideal” para os conjuntos de dados, foram realizados os cálculos do *F1 - Score*. Nas próximas seções são apresentados esses resultados. Primeiramente é realizada a comparação com o *K-means* e na sequência com o *DBSCAN*.

4.1 Comparação com o *K-means*

Como visto, anteriormente, o *K-means* é um dos mais algoritmos de agrupamento de dados mais tradicionais e serve de comparação para novas técnicas. No caso do *DL3*, foram utilizados 6 (seis) índices de validação para comparar os seus resultados com os gerados pelo *K-means*, além disso o número *k* de grupos considerado para os dois algoritmos é o apontado como “ótimo” na literatura para o conjunto. A Tabela 3 mostra os valores obtidos através desses cálculos para situação.

Para o conjunto de dado *Aggregation*, buscou-se um número *k* de grupos igual a 7 (sete). O agrupamento gerado pelo algoritmo *DL3* obteve resultados superiores em 5 (cinco) índices, chegando a ser superior a 93,55% no índice *Dunn GG*, que é baseado no Grafo de Gabriel. O *K-means* apresenta desempenho superior apenas segundo o índice *Silhouette*, porém não muito alto, cerca de 1,85%.

Jain é conjunto de dados caracterizado, na literatura, por possuir 2 (dois) grupos. Nesse caso, o *DL3* possui um melhor resultado segundo 4 (quatro) índices, principalmente nos fundamentados pela Teoria de Grafos. Novamente, o índice *Silhouette* aponta o agrupamento gerado pelo algoritmo *K-means* superior, assim como o índice *Dunn* que destaca, uma inferioridade de 74,6% do *DL3*.

Os valores para o *Lsun* indicam superioridade do *K-means* através do *Silhouette* e *Davies - Bouldin*, 8,17% e 7,06% respectivamente. Para o índice *Dunn* e os índices de *Dunn Generalizado* o resultado do *DL3*, é considerado melhor, chegando a melhorar 77,3%. Dessa forma, 4 (quatro) critérios afirmam que o agrupamento do novo algoritmo é o mais indicado para esse conjunto de dados.

Target foi o conjunto de dados que apresentou o mesmo número de índices, 3 (três), apontando melhor desempenho para os algoritmos. A maior porcentagem para o *DL3* é apontado pelo *Dunn*, cerca de 98,38% superior que o valor identificado para o agrupamento do *K-means*. Já o índice *Silhouette* destaca-se ao mostrar a performance do *K-means* 89,61% mais correta que a do *DL3*.

Para o conjunto *Face*, o agrupamento estabelecido pelo *DL3* é considerado mais adequado pelos índices *Dunn* e generalizações, como no caso do *Lsun*. O *Dunn* tradicional é o que apresenta maior porcentagem de superioridade, 93,38%. Já os resultados dos índices *Silhouette* e *Davies - Bouldin* ressaltam uma inferioridade da nova proposta, principalmente no primeiro critério.

O último problema testado é o encontrado no conjunto *Flame*, nesse caso os valores dos índices para os resultados gerados pelos algoritmos ficaram mais próximos do que os obtidos nas situações anteriores. O destaque de melhor desempenho do *DL3* é expressado no índice *Dunn*, 21,88% e o sua pior performance pelo índice *Dunn GG*, 13,77%.

Com esses resultados, é possível notar que o índice *Silhouette* sempre aponta os agrupamentos do algoritmo *K-means* como mais correto. Um dos motivos pode ser o uso da distância euclidiana em suas formulações, o que faz esse índice conseguir identificar melhor os formatos de *clusters* gerados pelos *K-means*. Já nos índices *Dunn RGN* e *Dunn MST* todos os resultados dos testes apontam o desempenho do *DL3* como mais correto. Como o índice *Dunn GG*, também baseado em grafos, mostrou algo semelhante na maioria dos casos, pode-se concluir que esses índices melhor identificam os agrupamentos ocasionado pelo *DL3*.

Para melhor avaliar o desempenho do novo algoritmo, ainda utilizou-se o critério de validação externo *F1-Score*. Vale ressaltar que o uso desse índice é possível, pois são conhecidas as respostas predeterminadas para os conjuntos de dados. Caso fossem considerados apenas problemas reais, não seria possível o uso desse índice.

Nas Figuras 4.1, 4.2 e 4.2 tem-se os gráficos dos agrupamentos gerados para os pelos algoritmos *K-means* e *DL3* e os resultados do *F1-Score* para cada caso. Percebe-se que o *DL3* obteve valor máximo para o *F1-Score* para quatro dos seis conjuntos de dados considerados. Os outros dois conjuntos, também obtiveram valores significativos, acertando mais de 98% dos rótulos. Já o *K-means* não conseguiu ter desempenho semelhante ao algoritmo proposto, em nenhum dos casos obteve 100% de acerto dos rótulos. O melhor resultado do *K-means* foi no conjunto *Jain*, com um *F1-Score* aproximado de 0,87820, nesse mesmo conjunto o *DL3* obteve *F1-Score* igual a 1.

Com os resultados obtidos pelos cálculos do *F1-Score* é mais fácil observar o melhor desempenho do algoritmo *DL3* em comparação ao *K-means*, o que não foi muito evidente nos resultados dos outros índices. Desta forma, é possível afirmar que nem sempre um índice de validação de agrupamento gerará um resultado coerente ao conjunto de dados, o que leva a motivação de futuros estudos nessa área.

Tabela 3: Valores para os índices *Silhouette* (o máximo valor encontrado nas simulações), *Davies-Bouldin* (o mínimo valor encontrado nas simulações), *Dunn* (o máximo valor encontrado nas simulações), *Dunn RNG* (o máximo valor encontrado nas simulações), *Dunn GG* (o máximo valor encontrado nas simulações), *Dunn MST* (o máximo valor encontrado nas simulações) e dos respectivos parâmetros (o k número de grupos para o *K-means* e o alcance α para o *DL3*). O número k de grupos considerado para os dois algoritmos é o apontado como “ótimo” na literatura para o dataset. O melhor resultado está destacado em negrito e os valores em verde e vermelho, representam respectivamente, a melhora ou piora do desempenho do *DL3* no respectivo conjunto de dados em comparação com o outro algoritmo.

<i>Conjunto de dados</i>	Índices	<i>K-means</i>	k	<i>DL3</i>	%	α
Aggregation	Silhouette	0,65566	7	0,64377	-1,85	2
	Davies-Bouldin	0,60867		0,38506	+58,07	
	Dunn	0,03254		0,03260	+0,18	
	Dunn RNG	4,96025		26,91242	+81,57	
	Dunn GG	1,39575		21,65246	+93,55	
	Dunn MST	4,96025		32,76156	+84,86	
Jain	Silhouette	0,67217	2	0,55036	-22,13	6
	Davies-Bouldin	0,78307		0,74325	+5,36	
	Dunn	0,01870		0,01071	-74,60	
	Dunn RNG	3,91528		4,91659	+20,37	
	Dunn GG	3,32437		4,17456	+20,37	
	Dunn MST	47,05870		59,09361	+20,37	
Lsun	Silhouette	0,65900	3	0,60921	-8,17	6
	Davies-Bouldin	0,65253		0,70208	-7,06	
	Dunn	0,03352		0,14769	+77,30	
	Dunn RNG	5,58416		23,35697	+76,09	
	Dunn GG	4,67072		9,48877	+50,78	
	Dunn MST	9,35970		23,35697	+59,93	
Target	Silhouette	0,49363	6	0,26034	-89,61	9
	Davies-Bouldin	0,75412		6,66221	-88,68	
	Dunn	0,00411		0,25330	+98,38	
	Dunn RNG	0,03640		0,06531	+44,27	
	Dunn GG	0,03640		0,02292	-58,81	
	Dunn MST	0,03640		0,09177	+60,34	
Face	Silhouette	0,65275	4	0,05293	-1133,23	10
	Davies-Bouldin	0,70938		0,75771	-6,38	
	Dunn	0,01503		0,22697	+93,38	
	Dunn RNG	0,98798		10,37251	+90,47	
	Dunn GG	0,69684		4,66256	+85,04	
	Dunn MST	2,85110		12,06393	+76,37	
Flame	Silhouette	0,53001	2	0,52774	-0,43	2,2
	Davies-Bouldin	1,12186		1,11958	+0,20	
	Dunn	0,03303		0,04228	+21,88	
	Dunn RNG	4,86674		4,88606	+0,39	
	Dunn GG	3,36189		2,95488	-13,77	
	Dunn MST	4,86674		4,88606	+0,39	

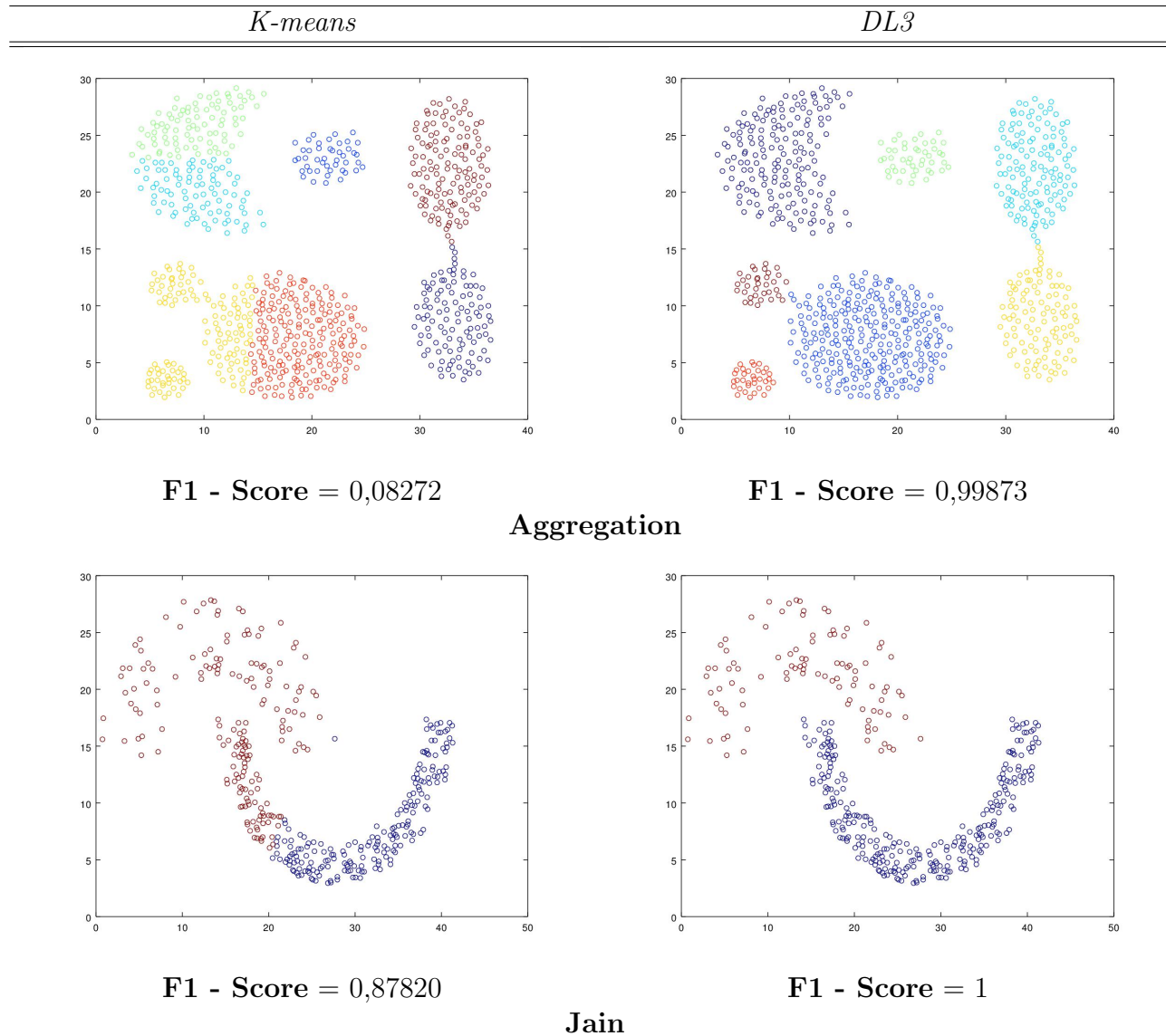


Figura 4.1: Agrupamentos gerados para os conjuntos de dados *Aggregation* e *Jain*, e os seus respectivos valores de *F1-Score*, considerando os algoritmos *K-means* e *DL3*.

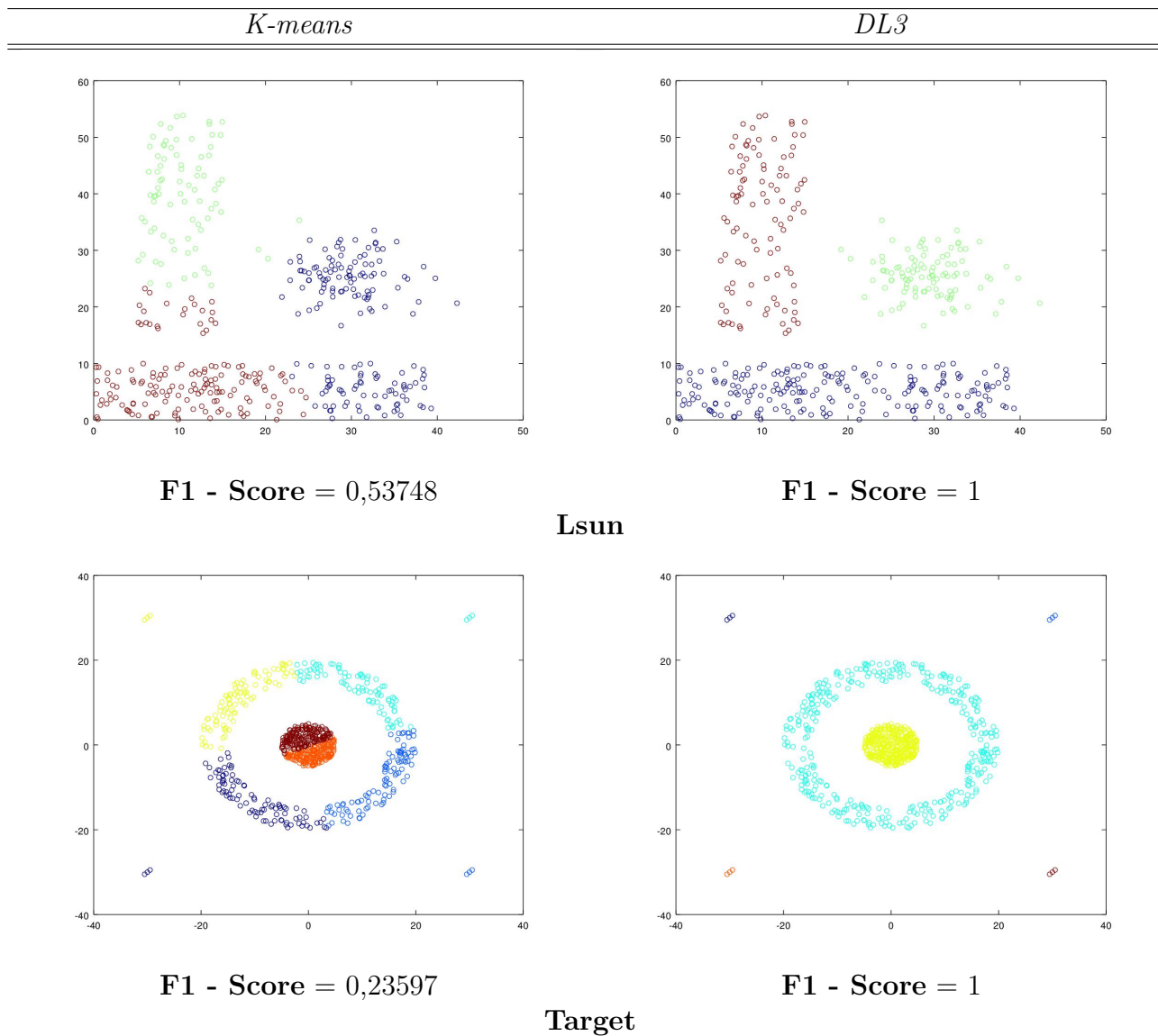


Figura 4.2: Agrupamentos gerados para os conjuntos de dados *Lsun* e *Target*, e os seus respectivos valores de *F1-Score*, considerando os algoritmos *K-means* e *DL3*.

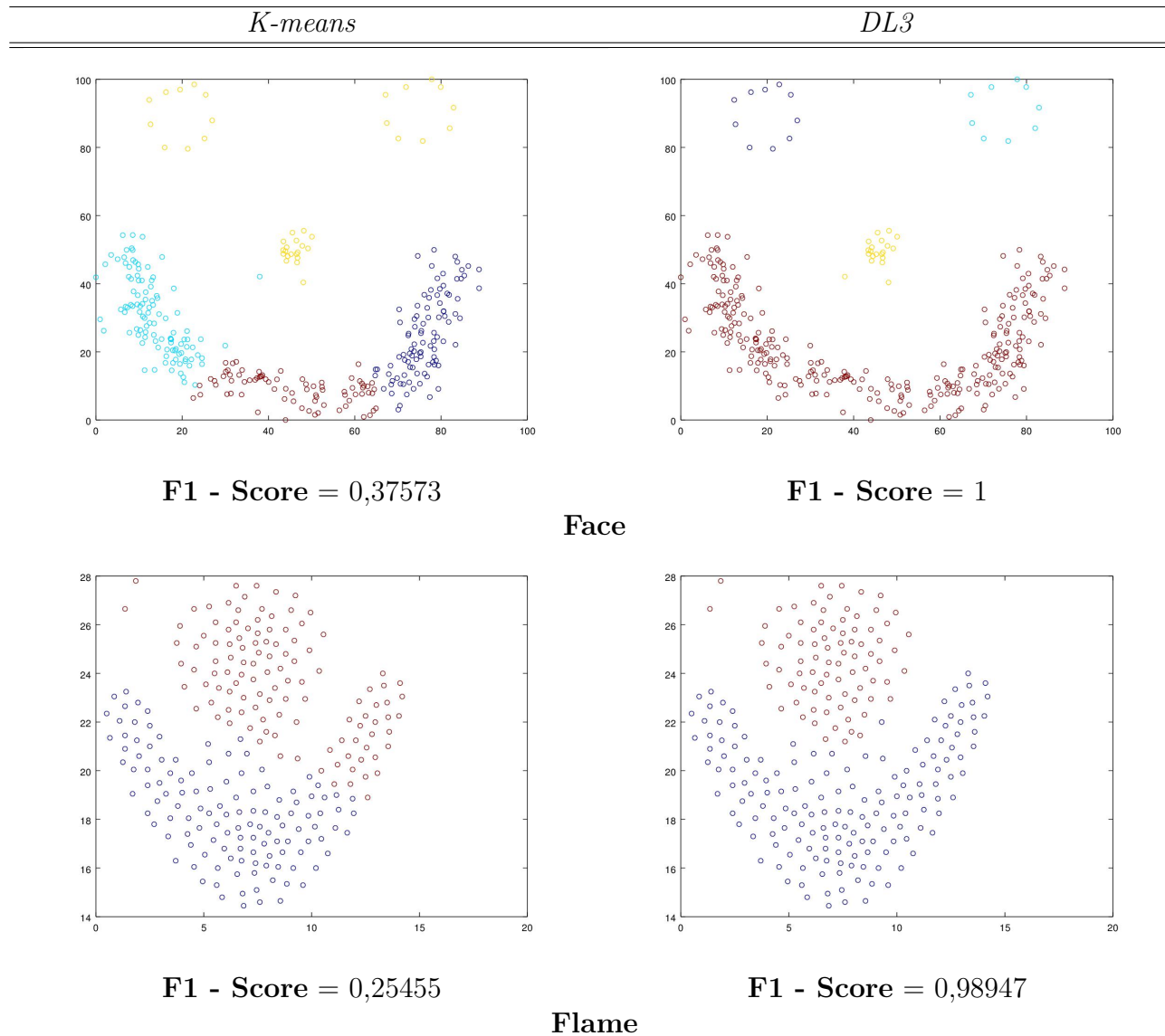


Figura 4.3: Agrupamentos gerados para os conjuntos de dados *Face* e *Flame*, e os seus respectivos valores de *F1-Score*, considerando os algoritmos *K-means* e *DL3*.

4.2 Comparação com o *DBSCAN*

O *DBSCAN* é outro algoritmo de agrupamento de dados considerado para comparar com o novo algoritmo. Como já mencionado anteriormente, para o *DL3* utilizou-se $0, 1 \leq \alpha \leq \max(\alpha)$ para determinar o alcance de visão dos agentes. De modo que, a cada nova simulação ao valor de α é acrescido uma quantia de $0, 1$ e $\max(\alpha)$ representa o alcance de visão que gera um único grupo como resultado final do agrupamento. Para o *DBSCAN*, os parâmetros *Minp* e *Eps*, foram definidos da seguinte forma: os valores do *Minp* variou dentro do intervalo $[1,20]$ (considerando apenas os números inteiros pertencentes a esse intervalo) e os valores do *Eps* variou de $[1,10]$, e acréscimo de $0,1$ a cada nova atualização desse parâmetro. Além disso, os conjuntos de dados permanecem os mesmo utilizados na comparação com o *K-means*.

Após a realização das referidas simulações, a comparação entre os resultados dos dois algoritmos foi realizada através da utilização do índice de validação de agrupamento *DBC*V, caracterizado na literatura como indicado para agrupamentos baseados em densidade. A Tabela 4 apresenta os melhores resultados obtidos pelo cálculo do índice para cada algoritmo, além dos parâmetros desse melhor caso.

Com os valores obtidos pelo índice, o algoritmo *DL3*, apresentou resultados melhores em quatro dos três dos conjuntos de dados considerados (*Aggregation*, *Lsun* e *Flame*), sendo que o melhor desempenho é no conjunto *Flame*, onde o valor do índice para o agrupamento gerado pelo *DL3* é aproximadamente 8,95% maior que do agrupamento ocasionado pelo *DBSCAN*. Além disso, somente no conjunto *Jain* que o *DBSCAN* obteve melhor desempenho, porém o percentual de superioridade é pequeno (0,08%). Nos outros dois conjuntos restantes, o índice *DBC*V aponta o mesmo resultado para ambos os algoritmos.

Outro ponto a destacar é o número de grupos, k , visto que em nenhum dos algoritmos é necessário informá-lo *a priori*. Para o conjunto de dados *Aggregation* em nenhum dos melhores resultados apontados pelo índice *DBC*V é obtido $k=7$ como esperado, para o algoritmo *DBSCAN* tem-se $k=6$ e no algoritmo *DL3* é $k=5$. Outro caso semelhante é apresentado no conjunto *Lsun*, nesse cenário o maior índice *DBC*V para o *DBSCAN* é um agrupamento com quatro grupos, sendo que o buscado é $k=3$.

Para os demais casos os valores de k são iguais aos indicados na literatura, porém isso não significa que os formatos dos *clusters* estão também de acordo. Por isso, utilizou-se o mesmo meio considerado na comparação com o *K-means*, a construção dos gráficos e o cálculo do *F1-Score*. Os resultados podem ser vistos nas Figuras 4.4, 4.5 e 4.6.

Através dos gráficos dos agrupamentos gerados por cada caso e apontados pelo índice como mais adequado, é possível observar que nem sempre o formato dos *clusters* são os esperados. Considerando os cálculos do *F1-Score*, essa conclusão fica mais evidente, destaca-se os resultados obtidos para os conjuntos *Jain* e *Flame*. Para o índice *DBC*V o agrupamento do algoritmo *DBSCAN* é superior ao melhor resultado do algoritmo *DL3*, mas através do *F1-Score*, considerando a resposta da literatura, percebe-se que o agrupamento resultante do *DL3* conseguiu acertar mais que o dobro de rótulos corretos se comparado com o valor alcançado para o agrupa-

mento do *DBSCAN*. Para o conjunto *Flame* algo parecido ocorre, o agrupamento do algoritmo *DBSCAN* obteve valor do *F1-Score* maior que o do *DL3*, sendo que o índice *DBC*V apontou o agrupamento do *DL3* melhor que o do *DBSCAN*.

Os conjuntos *Target* e *Face*, que alcançaram o mesmo melhor valor para o índice *DBC*V para os dois algoritmos, tiveram 100% de acerto de acordo com o *F1-Score*, assim é possível afirmar que o índice *DBC*V conseguiu encontrar o melhor agrupamento para esses conjuntos de dados segundo a literatura, sendo, dessa forma, mais sensível a esses formatos.

Com os resultados obtidos é possível observar que o *DL3* conseguiu um desempenho mais próximo ao *DBSCAN* do que ao *K-means*, isso fica evidente quando comparamos os valores dos índices, em muitos casos o *DL3* foi absurdamente muito superior que o *K-means*, caso que não acontece na comparação com o *DBSCAN*.

Tabela 4: Valores para o índice *DBC*V (o máximo valor encontrado nas simulações), os respectivos parâmetros (*MinP* e *Eps* para o *DBSCAN*, respectivamente e o alcance α para o *DL3*) e o k que representa o número dos grupos formados. O melhor resultado está destacado em negrito e os valores em verde e vermelho, representam respectivamente, a melhora ou piora do desempenho do *DL3* no respectivo conjunto de dados em comparação com o outro algoritmo.

Conjunto	DBC	<i>Minp</i>	<i>Eps</i>	k	DBC	%	α	k
Aggregation	0,82011	8	1,8	6	0,87710	+6,51	3	5
Jain	0,78799	9	3,7	2	0,78733	-0,08	4,8	2
Lsun	0,78716	14	5,9	4	0,84144	+6,45	7,4	3
Target	0,94770	1	10	6	0,94770	-	9	6
Face	0,95738	1	7,8	4	0,95738	-	10	4
Flame	0,75943	17	1,8	2	0,83406	+8,95	2,6	2

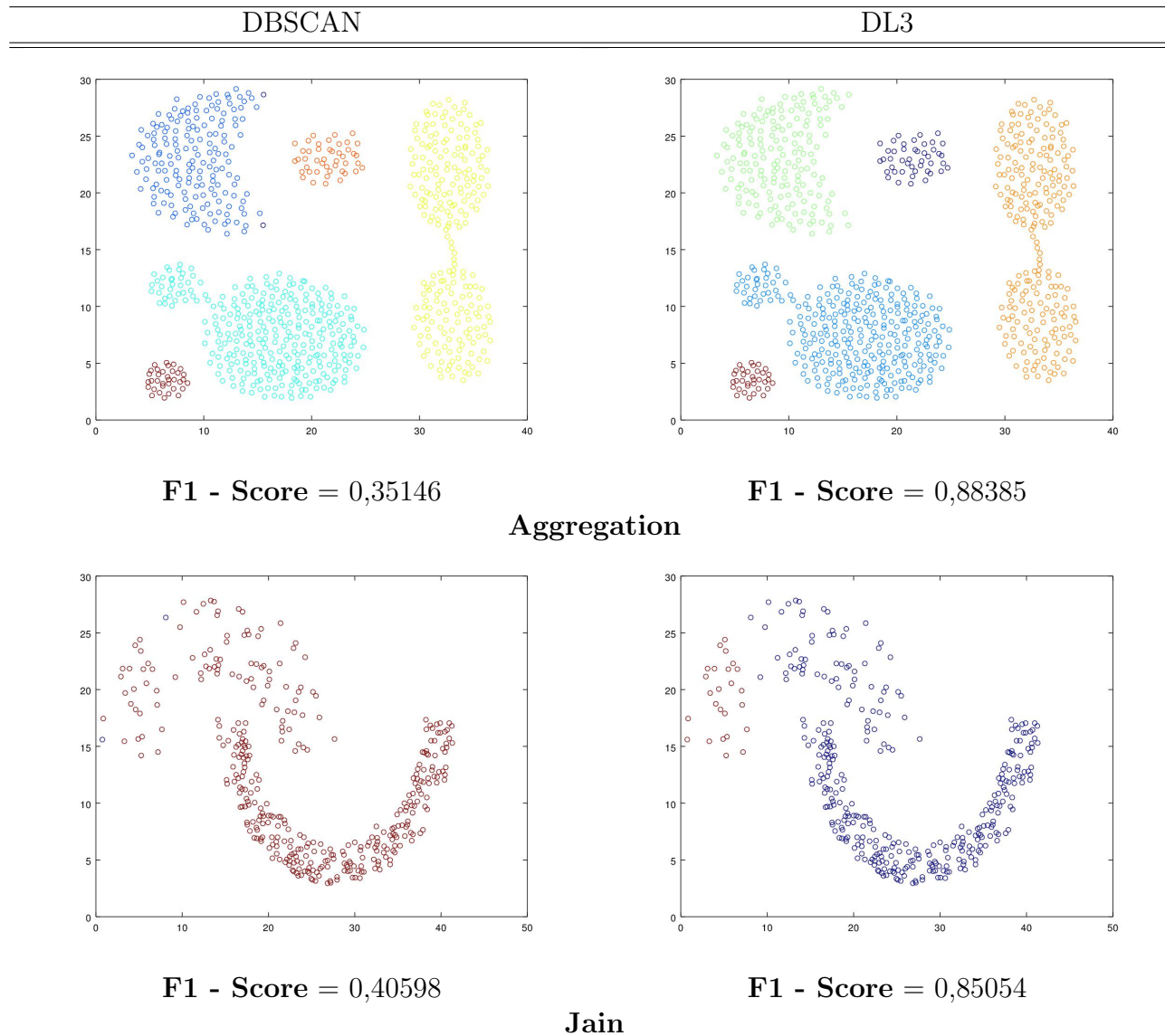


Figura 4.4: Agrupamentos gerados para os conjuntos de dados *Aggregation* e *Jain*, e os seus respectivos valores de *F1-Score*, considerando os algoritmos *DBSCAN* e *DL3*.

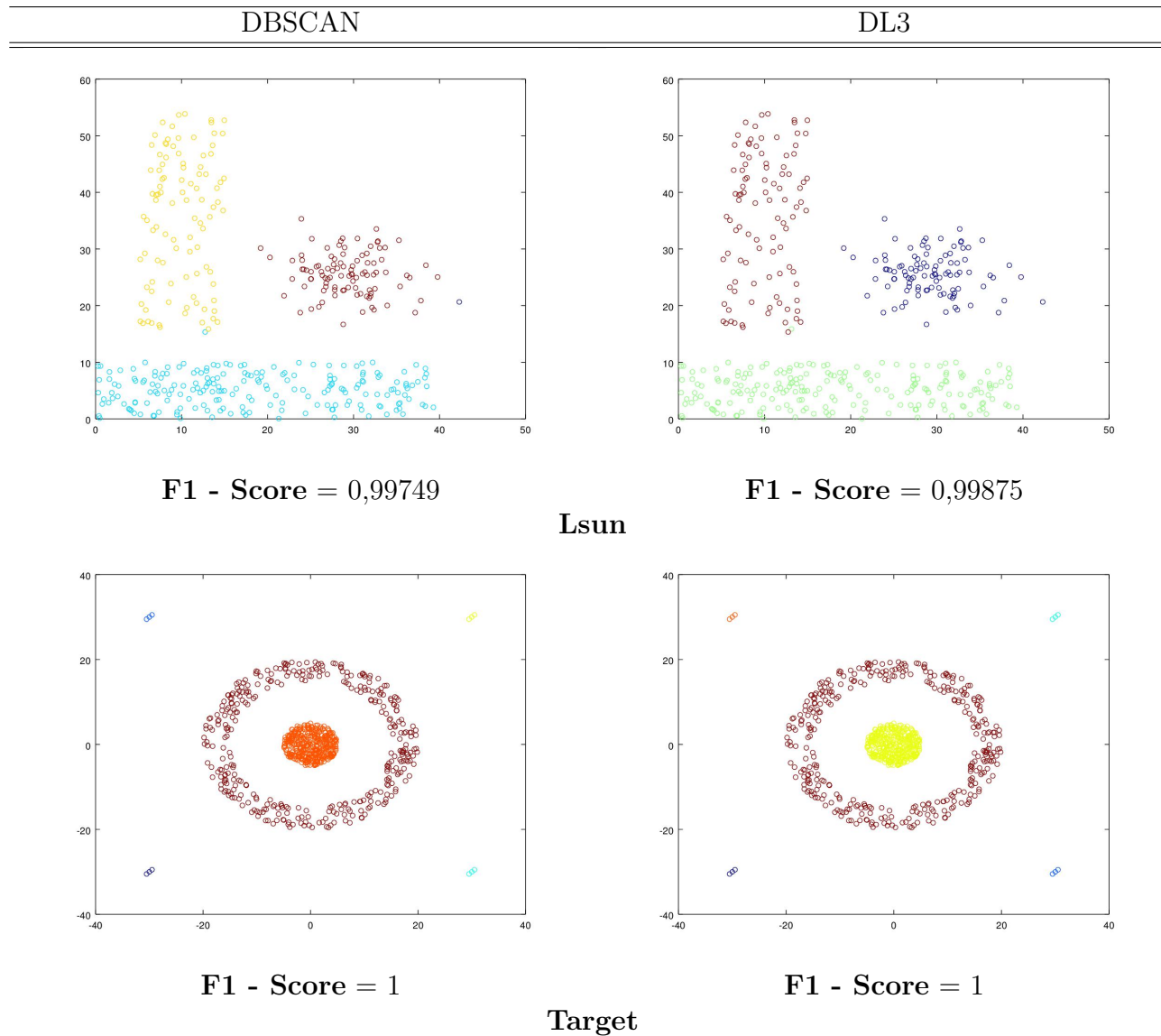


Figura 4.5: Agrupamentos gerados para os conjuntos de dados *Lsun* e *Target*, e os seus respectivos valores de *F1-Score*, considerando os algoritmos *DBSCAN* e *DL3*.

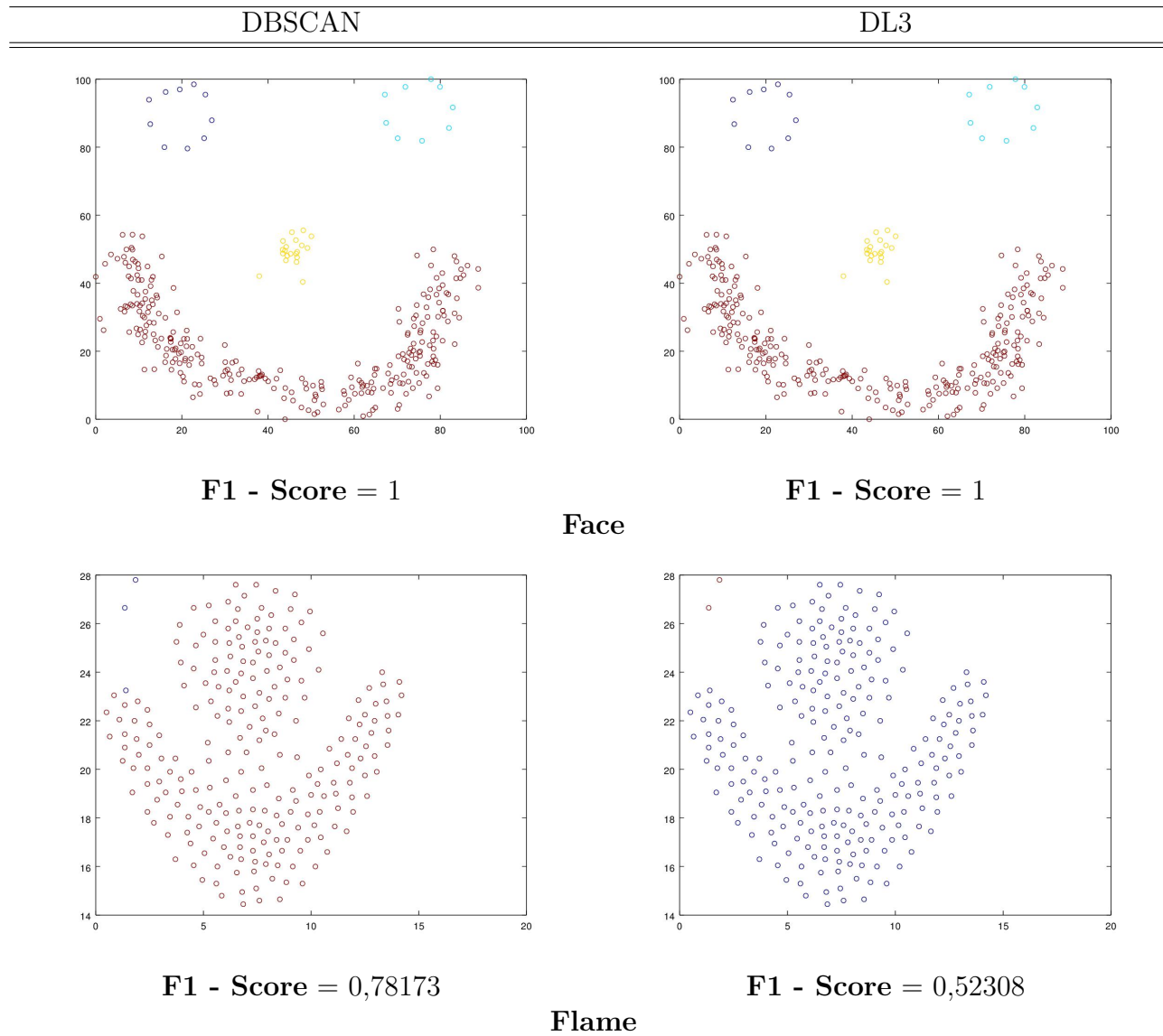


Figura 4.6: Agrupamentos gerados para os conjuntos de dados *Face* e *Flame*, e os seus respectivos valores de *F1 - Score*, considerando os algoritmos *DBSCAN* e *DL3*.

Por fim, o trabalho pretendeu testar o algoritmo *DL3* e apresentá-lo de uma forma bem clara. Os experimentos realizados tiveram como propósito analisar o algoritmo em alguns conjuntos de dados. Os resultados obtidos mostraram-se bons, levando a conclusão que sistema de agentes de comportamento simples desenvolvido pode auxiliar na tarefa complexa de agrupamento de dados.

5 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Com base nos resultados obtidos para diferentes conjuntos de dados, acredita-se que o algoritmo *DL3*, baseado em uma perspectiva de agentes, pode ser usado para vários problemas. Desta forma, não se limita a um único problema ou solução específica. Além disso, o usuário não precisa definir centroides, como alguns algoritmos tradicionais, para definir grupos, pois eles se organizam, formando seus próprios grupos sem informações iniciais sobre o número desejado. O algoritmo também segue algumas propriedades do agente, como a autonomia que contribui para a tomada de decisão durante o processo de agrupamento. No entanto, ele é dependente de determinados parâmetros, como o alcance de visão do agente (parâmetro α). Desta forma, trabalhos futuros podem se dedicar a formalizar maneiras de encontrar o valor ideal para o alcance.

Como pode ser visto, o uso dessa perspectiva de agentes para agrupamento de dados obteve bons resultados, superando em todos os casos o algoritmo *K-means* e alguns agrupamentos gerados pelo *DBSCAN* e chegando a empatar em duas situações, de acordo com o índice *F1-Score*. Além desse índice, outros foram testados, visto que nem sempre será possível utilizar o *F1-Score*, principalmente em conjunto de dados reais, os quais não se sabe previamente a distribuição correta dos *clusters*. Os resultados desses índices, mostraram algumas deficiências na detecção de alguns formatos de grupos, instigando, assim, possíveis reflexões posteriores sobre esse assunto.

Novos estudos, análises e propostas para melhorar o desempenho e estabilidade podem ser realizados, com o intuito de tornar o algoritmo *DL3* uma boa ferramenta para a mineração de dados. Pelos testes no *NetLogo*, verifica-se um desempenho rápido em sua execução, ou seja, consegue convergir (finaliza) com poucas iterações, não apresentando problemas de performance. Em trabalhos futuros, poderá haver mais ênfase nesse aspecto.

Além disso, a utilização de outros conjuntos de dados de dimensões maiores ou de dados reais, o desenvolvimento de novos índices de validação de *cluster* que melhor identifique os agrupamentos gerados pelo *DL3*, a comparação com outras técnicas de *clustering* não consideradas nesse trabalho, além do uso do algoritmo em algumas aplicações, como na segmentação de imagens, podem ser explorados em trabalhos futuros.

6 REFERÊNCIAS

- Aggarwal, C. C.; Reddy, C. K. *Data Clustering: Algorithms and Applications..* CRC Press, 2014, 652 p.
- Agogino, A.; Tumer, K., *A Multiagent Coordination Approach to Robust Consensus Clustering.* Advances in Complex Systems, v.13, n°. 2, p.165–198, 2010.
- Ahmed, K. N.; Razak, T. A. *An Overview of Various Improvements of DBSCAN Algorithm in Clustering Spatial Databases.* International Journal of Advanced Research in Computer and Communication Engineering, Thirumullaivoyal, Chennai, India , v.5, p.360–363, 2016.
- Andrade, L. A. C. G.; Cunha, C. B., *Algoritmo de Colônia Artificial de Abelhas para um Problema de Clusterização Capacitado.* XLIII SBPO - Simpósio Brasileiro de Pesquisa Operacional, 2011, Ubatuba. Anais do XLIII SBPO - Simpósio Brasileiro de Pesquisa Operacional, p. 1-12, 2011
- Aranha, C.; Iba, H., *The effect of using evolutionary algorithms on ant clustering techniques.* Proceedings of the Third Asian-Pacific workshop on Genetic Programming, p.24-34, 2006.
- Chaimontree, S.; Atkinson, K.; Coenen, F., *A Multi-agent Based Approach to Clustering: Harnessing the Power of Agents.* International Workshop on Agents and Data Mining Interaction ADMI 2011: Agents and Data Mining Interaction, Springer, p.16-29, 2011.
- Cohen, S. C. M.; de Castro, L.N., *Data Clustering with Particle Swarms.* IEEE Congress on Evolutionary Computation, CEC 2006, Vancouver, BC, Canada, p.1792-1798, 2003.
- Costa, P. P. *Teoria de Grafos e suas Aplicações.* 2011. Programa de Pós-Graduação – Mestrado Profissional em Matemática Universitária — Universidade Estadual Paulista Júlio de Mesquita Filho, Rio Claro.
- Davies, D. L.; Bouldin, D. W., *A Cluster Separation Measure.* IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Computer Society , PAMI-1, p.224–227, 1979.
- Deneubourg, J.L.; Goss, S.; Franks, N.; Sendova, F. A.; Detrain, C.; Chrétien, L. *The Dynamics of Collective Sorting Robot-Like Ants and Ant-Like Robots.* From Animals to Animats: Proc. of the 1st Int. Conf. on Simulation of Adaptive Behaviour, MIT Press Cambridge, MA, USA, p.356-363, 1991.
- Dunn, J. C.; Bouldin, D. W., *A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters.* Journal of Cybernetics, Taylor Francis , v.3, p.32–57, 1973.

- Ester, M.; Kriegel, H., P.; Sander, J.; Xu, X., *Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*. KDD-96 Proceedings, Portland, Oregon: AAAI Press, p.226–231, 1996.
- Fayyad, U.; Piatetsky-Shapiro, G.; Smyth, P., *From Data Mining to Knowledge Discovery in Databases*. AI Magazine, Palo Alto, California, USA: AAAI Press, v.17, n° 3, p.37–54, 1996.
- Fu, L., Medico, E., *FLAME, a novel fuzzy clustering method for the analysis of DNA microarray data*. BMC Bioinformatics, BioMed Central Ltd., v.8, n°.3, 2007.
- Gionis, A.; Mannila, H.; Tsaparas, P., *Clustering Aggregation*. ACM Transactions on Knowledge Discovery from Data, ACM New York, NY, USA , v. 1, n°. 1, Artigo 4, 2007.
- Godois, L. M.; Marco, L. C.; Adamatti, D. F., Emmendorfer, L. R., *Algoritmo DL3: Uma abordagem de Clustering baseado em Auto-organização*. Proceedings of WESAAC 2017, São Paulo -SP, Brasil, p. 90-100, 2017.
- Godois, L. M.; Marco, L. C.; Adamatti, D. F., Emmendorfer, L. R., *Um Algoritmo para Agrupamento de Dados usando interações entre Agentes*. Anais do Encontro Nacional de Inteligência Artificial e Computacional (ENIAC'2017), Uberlândia-MG, 2017.
- Gueleri, R. A., *Agrupamento de dados baseado em comportamento coletivo e auto-organização*. 2013. Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional - Mestrado em Ciências da Computação e Matemática Computacional — Universidade de São Paulo, São Carlos.
- Halkidi, M.; Batistakis, Y.; Vazirgiannis, M., *Cluster analysis and mathematical programming*. Journal of Intelligent Information Systems, Springer Berlin Heidelberg , v.17, p.107–145, 2001.
- Handl, J. *Ant-based methods for tasks of clustering and topographic mapping: extensions, analysis and comparison with alternative techniques*. 2003. Masters Thesis, Universität, Erlangen-Nürnberg, Erlangen, Germany.
- Hansen, P.; Jaumard, B., *Cluster analysis and mathematical programming*. Math. Program., Springer Berlin Heidelberg, v.79, p.191–215, 1997.
- Hartigan, J. A. *Clustering Algorithms*. John Wiley Sons, 1975, 366 p.
- Hartmann, V. *Evolving agents warms for clustering and sorting*. Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, Washington DC, p.217-224, 2005.
- Ilc, V. *Modified & Generalized Dunn's index*. Site MathWorks. Disponível em <<https://www.mathworks.com/matlabcentral/fileexchange/42199-modified—generalized-dunn-s-index>>. Acesso em 08/12/2017.

- Jain, A.; Dubes, R. *Algorithms for clustering data*. Prentice Hall, 1988, 304 p.
- Jain, A. K.; Law, M. H. C., *Data Clustering: A User's Dilemma*. PReMI 2005, LNCS 3776, Springer-Verlag Berlin Heidelberg, p. 1–10, 2005.
- Karaboga, D.; Ozturk, C., *A novel clustering approach: Artificial Bee Colony (ABC) algorithm*. Applied Soft Computing, Elsevier, p.652–657, 2009.
- Kennedy, J. ; Eberhart, R. *Particle swarm optimization*. Proceedings IEEE International Conference on Neural Networks, Perth, WA, Australia, p.1942–1948, 1995.
- Kubalik, J. ; Tichy, P.; Sindelar, R. ; Staron, R. J., *Clustering Methods for Agent Distribution Optimization*. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), v.40, n^o.1, p.78 - 86, 2010.
- Lumer, E. D.; Faieta, B., *Diversity and adaptation in populations of clustering ants*. Proceedings of the Third International Conference on Simulation of Adaptive Behaviour, MIT Press Cambridge, MA, USA, p.501–508, 1994.
- MacQueen, J. B., *Some Methods for classification and Analysis of Multivariate Observations*. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, Calif.: University of California Press, p.281–297, 1967.
- Merwe, D. W. van der ; Engelbrecht, A. P., *Data clustering using particle swarm optimization*. The 2003 Congress on Evolutionary Computation, CEC '03., Canberra, ACT, Australia, Australia, IEEE, p.215-220, 2003.
- Minden, V. L.; Youn, C. C. ; Khan, U. A., *A distributed self-clustering algorithm for autonomous multi-agent systems*. 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, USA, p.1445-1448 2012.
- Monmarché, N., *On data clustering with artificial ants*. AAI Technical Report WS-99-06 , Orlando, Florida, p.23-26, July 18, 1999.
- Moulavi, D.; Jaskowiak, P. A.; Campello, R. J. G. B.; Zimek, A.; Sander, J., *Density-Based Clustering Validation*.. Proceedings of the 14th SIAM International Conference on Data Mining, Society for Industrial and Applied Mathematics, Philadelphia, USA, p.839–847, 2014.
- Pal, N.; Biswas, J. *Cluster validation using graph theoretic concepts*. Pattern Recognition, Elsevier , v.30, p.847–857, 1997.
- Rabuske, M. A. *Introdução à Teoria dos Grafos*. Editora da UFSC, 1992, 173 p.
- Ranjbar, M.; Azami, M.; Rostammi, A. S., *Fuzzy Artificial Bee Colony for Clustering* . Journal of Agricultural Science and Engineering, v. 1, n^o. 2, , p. 46-53, 2015.

- Reynolds, C. W, *Flocks, herds and schools: a distributed behavioral model*. Computer Graphics, Elsevier, v.21, n^o.4, p.25-34, 1987.
- Rousseeuw, P. J., *Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis*. Computational and Applied Mathematics, Springer International Publishing , v.20, p.53–65, 1987.
- Russell, S. ; Norvig, P. *Artificial Intelligence*. Prentice Hall, 1995, 1016 p.
- Santos, D. S. *Bee clustering: um Algoritmo para Agrupamento de Dados Inspirado em Inteligência de Enxames*. 2009. Programa de Pós-Graduação em Computação - Mestrado em Computação — Universidade Federal do Rio Grande do Sul, Porto Alegre.
- Stone, P.; Veloso, M. *Multiagent Systems: A Survey from a Machine Learning Perspective*. Autonomous Robotics, Springer US, v.8, p.345–383, 2000.
- Tan, P. N.; Steinbach, M.; Kumar, V. *Introduction to Data Mining*. Pearson Addison-Wesley, 2006, 769 p.
- Tomasini, C. *Seleção automática de índices internos de validação de agrupamento..* 2015. Programa de Pós-Graduação em Computação - Mestrado em Engenharia de Computação — Universidade Federal do Rio Grande, Rio Grande.
- Utsch, A. *Clustering with SOM: U*C*. Proceedings Workshop on Self-Organizing Maps, Paris, France, p.75–82, 2005.
- Vizine, A. L, Castro, L. N., Hruschka, E. R., Gudwin, R. R, *Towards Improving Clustering Ant: An Adaptative Ant Clustering Algorithm*. Informatica, Slovenian Society Informatika, v.29, n^o2, p.143-154, 2005.
- Wooldridge, M. *An Introduction to Multiagent Systems*. JOHN WILEY SONS, LTD, 2002, 484 p.
- Wooldridge, M.; Jennings, N. R., *Intelligent Agents: Theory and Practice*. The Knowledge Engineering Review, Cambridge University Press, v.10, No2, p.115–152, 1995.
- Xin, P.; Sagan, H., *Digital Image Clustering Algorithm based on Multi-agent Center Optimization*. Journal of Digital Information Management, v.14, n^o.1, p.8-14, 2016.
- Xu, R.; Wunsch, D. C. *Clustering*. Wiley-IEEE Press, 2009, 368 p.
- Zadeh, L., *Fuzzy sets..* Inf. Control, Elsevier Inc. , v.8, p.338–353, 1965.
- Zaki, M. J. ; Meira JR., W. *Data Mining and Analysis: Fundamental Concepts and Algorithms..* Cambridge University Press, 2014, 562 p.
- Zhang, C.; Ouyang, D.; Ning, J., *An artificial bee colony approach for clustering*. Expert Systems with Applications, Elsevier, v. 37, p.4761–4767, 2010.